INFORMATION MINING FROM ONLINE REVIEWS FOR PRODUCT DESIGN

JIN, JIAN

Ph.D

The Hong Kong Polytechnic University

2013

The Hong Kong Polytechnic University

Department of Industrial and Systems Engineering

Information Mining from Online Reviews for Product Design

JIN Jian

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

August 2012

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

________________________(Signed)

________Jin Jian________(Name of student)

ABSTRACT

To improve product quality, designers should understand customers' needs. Traditionally, customers' needs are gathered through surveys. Nowadays, more and more e-commerce websites encourage customers to post online reviews and express their opinions. Consequently, online reviews become a valuable source of customers' needs. However, it is difficult for product designers to understand all the relevant customers' needs from a vast number of online reviews. Under this circumstance, an intelligent system should be designed to mine useful information from online reviews and help designers to improve their products. To understand how designers filter and digest the customers' needs, two exploratory case studies are conducted. The purpose of the first case study is to explore why some reviews are preferred by product designers, while the second is to understand how designers analyze online reviews.

From the first case study, two questions for the identification of design-preferred reviews are clarified. (1) The first question is how to identify helpful online reviews from designers' perspective. This question is formulated as a regression model. According to designers' arguments about why some reviews are helpful, the regression model is built from four categories of features which are extracted directly from the review contents. Closely associated with this question, another concern is whether the concept of helpfulness perceived by designers in one domain can be migrated to other domains. Different methods of feature selection are employed for this concern. (2) The second question is how to recommend rating values on online reviews by taking designers' personal preferences into consideration. This question is

formulated as a classification model. Utilizing the four categories of features, this classification model is considered from both the generic helpfulness aspect and the personal preference aspect from a designer. Various experiments suggest that design-preferred reviews can be identified by analyzing review content automatically.

From the second case study, two questions for building a design-centered knowledge base from online reviews are explored. (1) The first question is how to associate online reviews with product characteristics. A probabilistic approach, which utilizes the statistic information about keywords and context words in the online reviews, is proposed for this question. The impacts of context words are estimated in this probabilistic approach according to their distances to keywords. (2) The second question is how to prioritize product characteristics from online reviews. Based on the customer satisfaction on online reviews, an ordinal pairwise supervised classification approach is developed for this question. Also, an integer nonlinear programming optimization model is advised to make the pairwise-based results of this approach to be evaluated with standard classification and ranking evaluation metrics. The encouraging results validate the feasibility of the proposed methods.

Overall, in this research, a regression model is proposed to identify helpful online reviews; several feature selection methods are compared to explore whether the concept of helpfulness perceived by designers in one domain can be migrated to other domains; a classification model is suggested to recommend rating value on online reviews; a probabilistic approach is developed to connect online reviews with product characteristics; and finally, an ordinal pairwise supervised classification approach as well as the integer nonlinear programming optimization model is advised to prioritize product characteristics based on online reviews. The proposed techniques benefit product designers to improve their products and attract more customers. Future extensions can be conducted towards building intelligent applications to process online reviews for product designers.

ACKNOWLEDGMENTS

This research would not have been a success without the help of many people who gave me the greatest support.

First, I would like to express my great appreciation to my supervisor, Dr. Ping Ji. His continuous assistance and insight advice have cultivated me to finish this research work. His rich knowledge, valuable experience and analytical mind benefit me a lot about how to be a capable researcher.

I would like to convey my great gratefulness to my co-supervisor, Dr. Ying Liu, an assistant professor in the Department of Mechanical Engineering of National University of Singapore. He coached me as a researcher from scratch and provided me instructive suggestions about doing research. His hard working impresses me deeply and enlightens me the reason of his success.

Besides, my deep admirations would be given to my family. No matter where I am, they always care about my safety, my health, and my work. They give me the most selfless love.

Finally, the financial support from the General Research Fund of the Hong Kong SAR Government and the Department of Industrial and Systems Engineering of The Hong Kong Polytechnic University enables this project possible. I would also like to express my gratitude to several respectful teachers, many dear friends and those who helped me indeed but missed to thank previously.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACM	Association for Computing Machinery
AHP	Analytic Hierarchy Process
API	Application Programming Interface
CIKM	Conference on Information and Knowledge Management
CRF	Conditional Random Field
DCG	Discounted Cumulative Gain
DP	Document Profile
EM	Expectation Maximization
HoQ	House of Quality
HMM	Hidden Markov Model
IQ	Information Quality
IR	Information Retrieval
KNN	*K* Nearest Neighborhood
LDA	Latent Dirichlet Allocation
MAE	Mean Absolute Error
MAP	Mean Average Precision
MFS	Maximal Frequent Sequences
MLP	Multilayer Perceptron
NDCG	Normalized DCG
NLP	Natural Language Processing
PCA	Principal Component Analysis

PMCC	Pearson product Moment Correlation Coefficient
POS	Part of Speech
QFD	Quality Function Deployment
RMSE	Root Mean Squared Error
SIGKDD	Special Interest Group on Knowledge Discovery and Data Mining
SIGIR	Special Interest Group on Information Retrieval
SVM	Support Vector Machines
TF-IDF	Term Frequency - Inverse Document Frequency
UGC	User Generated Contents
VOC	Voice of the Customer
WSD	Word Sense Disambiguation

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Customer understanding plays an important part in market-driven product design. Conventionally, customer needs are obtained from questionnaires, call centers or customer services. After digesting all these data, designers conceive to improve their products. But it is usually time-consuming and labor-intensive to gain these data and conduct the analysis manually.

Nowadays, more consumers like to share their experiences online. They post a large number of product reviews online. Rich information about their opinions towards products is provided in these online reviews. Many potential consumers are influenced by online reviews in their purchase decision-making, and in the meantime, consumer preferences offered in online reviews are valuable for product designers [LLL07]. On the one hand, potential consumers are highly possibly affected by online reviews in terms of certain specific brands and models. Positive reviews might not necessarily bring in new customers, but strong negative reviews are possible to lead to sales decrease. On the other hand, product designers can gain insights from the analysis of online reviews potentially. Strategic adjustments as well as technical improvements can be made accordingly. Figure 1.1 illustrates the snapshot of a typical review on a color printer. As seen from this figure, this consumer gives 4-star to this printer. Many different aspects of printers are mentioned and both positive and negative sentiments of different product features are presented. For example, this consumer satisfies that “the actual printing is quiet”, but does not satisfy that “the paper tray feels a bit flimsy". After reading this printer review, "204 of 209" consumers found it helpful. Generally speaking, this online review provides valuable information about customer preferences and concerns and it is highly possible to help printer designers to improve their products.

204 of 209 people found the following review helpful:

Early Opinion 4.5 stars., September 23, 2009

By Steven Alpert

This review is from: **Epson Artisan 810 Wireless Touchscreen Color Inkjet All-in-One Printer (C11CA52201) (Electronics)**

I only have the printer for a few days, but so far I am very pleased.
I was a bit nervous before buying because I had read a lot of complaints about an earlier model, Artisan 800, having a lot of paper jams. But the paper handling on my 810 has been flawless, if a tiny bit noisy while it pulls in the paper. The actual printing is quiet, and of great quality. The paper tray feels a bit flimsy, but is easy to remove or insert, and there's no fuss to loading your paper in it. It can expand to hold legal size paper, and has a separate area for smaller sized papers, usually for photo printing paper. The package comes with 3 sheets of Epson high quality 4x6 glossy photo paper, so I printed three photos as a test and I could not be more pleased with the photo print quality. I have been using an HP Wireless printer up until now (model 5850) and the print quality of the Artisan 810 far outstrips that HP, with much deeper blacks and dark tones, and more rich looking color, which is probably due to superior ink quality more than the printer quality. Also, they promise that their inks are acid-free and will last 3 or 4 times longer than drug-store prints. Another pre-purchase worry was that there were complaints about the Artisan 800 being a big ink-hog. Too early to tell, but I've printed those 3 photos and a fourth on plain paper, and about 25 pages of text (some with graphics) and the ink levels have not budged yet. So, so far so good on ink usage. I was surprised and pleased too that the 810 printer comes with an TWO black ink cartridges, so you'll have an extra one when it runs out. Nice! Additionally, the inks provided are the same capacity as the refills, whereas some printers I have seen came with ink cartridges that have a lower capacity. A nice feature is that when you plug in a camera memory card in the front slots, not only can you see and even crop the photos on the printer's screen, but the photos on the card can also be remotely viewed by your computer on the same wifi! Very neat. It warns on the box that this feature may not work with Mac computers, but it worked fine with my IMac. Haven't tried printing from the iPhone as they claim you can do, though I suspect I won't use that feature much. I love the big touch-screen operation too, it is pretty slick. Made a few copies of some simple pencil drawings and they were very good, and it was not obvious to tell which was the original or copy, but I haven't really tested the copier out enough as yet to give a good opinion about its copy qualities.

Figure 1.1 One typical customer review

However, there exist hundreds or even thousands of online reviews for a product, especially for some popular products. They are distributed in different online shopping websites, such as Amazon.com, Epinions.com, etc. Also, some reviews contain only a few words but critical points towards important product engineering characteristics are reflected. Others may be rather lengthy with only few sentences containing valuable opinions. All of these unique properties of online reviews challenge both consumers and designers to process all of this helpful information manually.

From 2003-2004, there was an obvious increase in number of researchers studying and analyzing online reviews. These researchers mainly come from computer science domain and many publications are seen in several preeminent research communities, including ACM SIGKDD (Association for Computing Machinery, Special Interest Group on Knowledge Discovery and Data Mining), ACM SIGIR (Special Interest Group on Information Retrieval) and CIKM (Conference on Information and Knowledge Management) [DL07, HL04a]. It is now known as opinion mining. Opinion mining, or sentiment analysis, for online reviews refers to the application of techniques such as natural language processing (NLP), computational linguistics, and text analysis to identify and extract subjective information from online reviews, which aims to determine consumer attitudes with respect to the concerned products.

Various interesting research publications in opinion mining are categorized into three types, namely, sentiment identification, sentiment extraction, and opinion retrieval [DLY08, DLZ09, GAO+05, HL04b, JL08, KH04]. Nevertheless, there exist some limitations in the current study of opinion mining. They fail to step further and make a breakthrough towards making their innovations to be utilized directly by product designers. More specially, these findings are hard to be embedded in product design to facilitate designers to understand a large amount of online customer reviews efficiently. But to what extent that designers can understand customer needs and preferences in designing products is believed to be a decisive factor, especially in the stage of conceptual design.

Actually, in the design area, many researchers also dedicate to analyze customer needs. They proposed various approaches and models to interpret consumers' concern. One of the most famous tools to interpret customer needs or voice of the customer (VOC) is Quality Function Deployment (QFD). Originally introduced in Japan in the late 1960s [Aka90, Aka04], QFD has gained international acceptance in the design area. QFD facilitates designers in customer driven product design and manufacturing to save the production cost and time [Coh95]. QFD links customer needs to various engineering characteristics and, eventually, outputs the specifics of engineering requirements.

QFD starts from a planning matrix that connects customer needs to product engineering characteristics, product planning, part deployment and even manufacturing operations [FTTC03]. The planning matrix is called House of Quality (HoQ). Figure 1.2 shows the overall structure of HoQ.

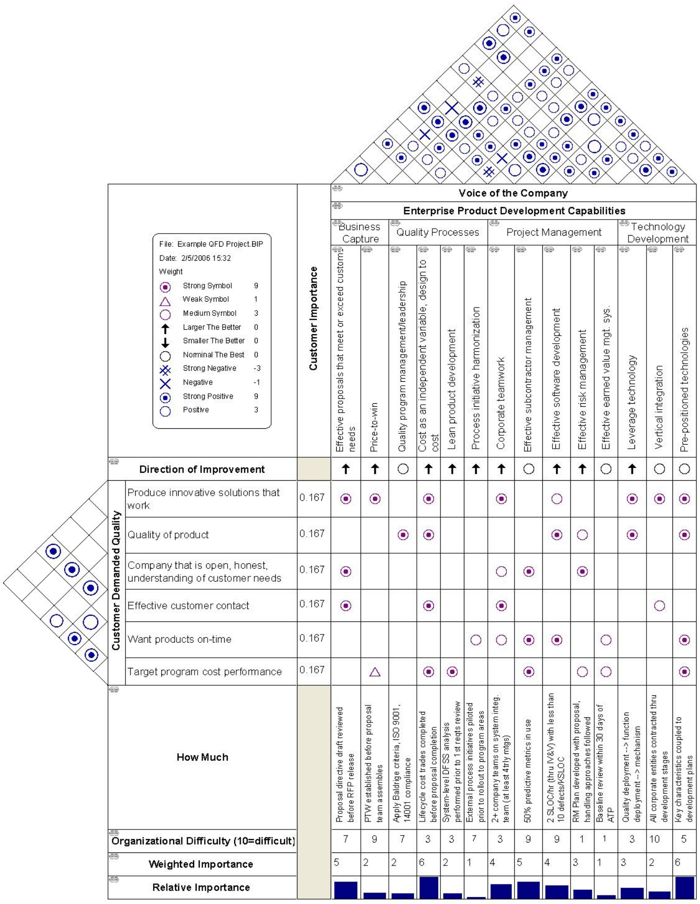


Figure 1.2 The overall structure of House of Quality [Coh95]

HoQ is a diagram, used for defining the relationship between customer needs and the product capabilities. It looks like a house with a correlation matrix as its roof, customer needs versus product engineering characteristics as the main part, competitor evaluation as the porch. The current researches of QFD methodology are basically concentrated on the following five areas:

- Identification of customer needs and their weightings
- Generation of product engineering characteristics and their weightings
- Finding association between customer needs and engineering characteristics
- Benchmarking with competitors
- Target values of product engineering characteristics

However, many of the state-of-the-art researches in the design area, including the publications in different areas of QFD, only take survey data as the input of customer analysis. Online reviews are generically different from survey data or data gathered from questionnaires or interviews. It is generally agreed that online reviews refer to the free texts written by consumers, rather than experts in specific domains. They are written entirely based on the willingness of consumers, out of their own interests, in their language and without any pre-defined questions to lead them.

In this circumstance, an intelligent system should be designed to mine useful information from online reviews and help designers to improve their products.

1.2 PROBLEM STATEMENT

Actually, there exist several research problems which must be solved before an intelligent system is designed to facilitate product designers.

For example, iPad designers want to improve the current model and they collect a large number of customer reviews. However, as a matter of fact, the quality of information available in a community is often inversely related to the size of its membership [Ott09]. So it is actually questionable to treat all these online reviews without bias, as the first step of QFD, in identification of customer needs and their weightings. One review of the new iPad from Amazon.com complains that "…I was hoping the third generation wouldn't be noticeably thicker and heavier than the iPad 2, but unfortunately it was. I could definitely tell the difference when reading ebooks, which I do a lot…. After reading some reviews with the latest iPad, I really miss the thinner and lighter iPad 2, and I decided to return the new iPad to my local Apple store and buy a new iPad 2." Generally speaking, it is a helpful product review since this consumer speaks out the opinion and provides some experience details, which might facilitate iPad designers to improve the current model. But, currently, there are no efficient strategies to filter a large number of online reviews and obtain valuable information from the perspectives of product designers. What aspects of customer reviews really make product designers regarding them helpful are particularly important and this is the first concern to build this intelligent system.

Another example also comes from the iPad design. The previous review complains about the weight and size problem and some reasons are provided by this consumer to support the arguments. It might be highly possible to be interpreted as a helpful one. However, this one might be also rated as “very unhelpful” by battery designers because they only care about these reviews which complain about whether iPad will drain the battery fast. Actually, it is how to recommend rating values on online reviews to meet the tastes of different product designers. To some online reviews, different designers may have different assessments about the helpfulness of online reviews from their own perspectives. Together with the previous question of identifying helpful online reviews, these two questions are relevant yet different. The previous one stresses how to filter out useful information from online reviews in a generic viewpoint, while this question targets at taking designers’ personal assessment into considerations. It is an interesting question, especially for some complex products which contain various features to be considered.

Now, suppose helpful online reviews are available, the challenging question is how to translate valuable customer concerns into QFD. More specifically, the question is how to connect online reviews to product engineering characteristics automatically. This question actually points to the association between customer needs and product engineering characteristics in QFD. For instance, in the previous example, this question is how to automatically connect the review, which complains iPad by the words “thicker and heavier”, with the weight and size problem through analyzing each sentence. Although the question of connecting consumer needs with product engineering characteristics has been studied by some researchers in the design area, they only focus on customer survey data, which often contain formatted tables and targeted interview questions. Also, these survey data do not contain much sentimental expressions which are one obvious characteristic that online reviews possess. Hence, online reviews can not be processed directly by the traditional models in the design area. It is not an easy task to connect online reviews with product engineering characteristics automatically.

The next step for product designers is to suggest a method to prioritize different product engineering characteristics when they conceive new models, which is also one important research area in QFD. For instance, when iPad designers plan to launch the next-generation product, they should balance the importance or weights of different product engineering characteristics, such as, battery, size, etc. Similarly, there also exist several relevant research efforts about how to prioritize product engineering characteristics. Still, only customer survey data are the focus in these researches and these models developed from the traditional customer survey are not applicable to online reviews. How to prioritize product engineering characteristics from online reviews for market driven design has never been investigated through yet. But it is one important step if online reviews are analyzed for product design.

Solving these questions properly and building a QFD model from online reviews provide interesting information to product designers. A unique opportunity is offered for researches on customer behaviors and market trends with a large amount of online reviews.

1.3 RESEARCH OBJECTIVES

In responding to these interesting questions, an intelligent system is proposed by identifying design-preferred online reviews from the perspective of product designers and building a design-centered knowledge base from online reviews.

- Identifying design-preferred online reviews from the perspective of product designers

The first objective of this research is to build a collection of helpful online reviews. The key question here is which online review is helpful. Several aspects of online customer reviews that are regarded by product designers regarding as helpful will be investigated in this research. These aspects will be utilized to predict the helpfulness of online reviews in the viewpoint of product designers. Also, whether the helpfulness of online reviews, as a concept being perceived by product designers in one domain, is able to be transferred to other domains will be examined.

The second objective of this research is to recommend rating values on online reviews with consideration of designers' assessments. The reasons why some reviews receive a divergence in judging the helpfulness will be investigated in this research. These reasons, together with several aspects of customer reviews that are regarded by designers as helpful, will facilitate this research to build a method for recommending rating values on online reviews from different product designers.

- Building a design-centered knowledge base from online reviews

The third objective of this research is to connect customer reviews with product engineering characteristics. The statistical information about various words in online reviews as well as their complex relationships with product engineering characteristics will be examined. Accordingly, a linguistic approach for connecting online reviews with product engineering characteristics automatically will be derived for product designers to ease them from analyzing all user generated contents (UGC) sentence by sentence.

The fourth objective of this research is to prioritize product engineering characteristics based on customer online reviews for improving the current product model. The customer sentiments of different product engineering characteristics as well as the overall customer satisfaction will be extracted from online reviews. Based on this, a method to prioritize product engineering characteristics will be developed.

Overall, the objectives of this research are:

- to build a collection of helpful online reviews;
- to recommend rating values on online reviews by taking designers' personal assessments into consideration;
- to connect customer reviews with product engineering characteristics;
- to prioritize product engineering characteristics based on online customer reviews for improving the current product model.

These four objectives form a guideline for this research.

1.4 THESIS SCOPE

The structure of the thesis is organized as follows.

In Chapter 2, an extensive literature review is conducted to demonstrate what have been studied on the quality evaluation of text documents, the recommendation systems, connecting customer needs with design quality, as well as prioritizing product engineering characteristics. Since opinion mining and QFD are fundamental concepts in this research, some state-of-the-art developments about these two research areas are also surveyed.

In Chapter 3, the framework of the proposed intelligent system is explained. The identification of design-preferred online reviews and the development of a design-centered knowledge base from online reviews are concerned in this intelligent system. Moreover, in order to understand how online reviews are examined in customer analysis from the perspective of product designers, two exploratory case studies are conducted.

In Chapter 4, the identification of design-preferred online reviews is centered. Two relevant questions are studied. First, four categories of features which are extracted from review documents are utilized to model the helpfulness of online reviews from the perspective of product designers. Whether the helpfulness of online reviews can be modeled as domain-free is also evaluated with feature analysis methods. Second, rating values on online reviews are recommended by combining both a generic perspective and a personal assessment perspective. The recommendation results from both perspectives are aggregated by a classification algorithm. Finally, the effectiveness of the two proposed approaches is verified by various categories of experiments.

In Chapter 5, the development of a design-centered knowledge base from online reviews is targeted. Also, two questions are investigated. At first, a linguistic approach is proposed to analyze how to connect customer reviews with product engineering characteristics. Two linguistic models, the Unigram model and the Bigram model, are derived accordingly. Moreover, a pairwise-based approach is devised in order to prioritize product engineering characteristics from online reviews for improving the current product model. An integer non-linear programming optimization model is then proposed to testify the performance of the pairwise-based approach with standard evaluation metrics for classification and ranking.

In Chapter 6, the achievements and the conclusions of this research are provided. Finally, some prospects for future work are suggested.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In Chapter 1, the four questions to be studied in this research were introduced. In this chapter, relevant literatures and some fundamental knowledge are reviewed.

This chapter is organized as follows. In Section 2.2, the highlight is the introduction of opinion mining and QFD. A limited number of researches about how online reviews are used in product design are also presented. From Section 2.3, several relevant yet fundamentally different researches are reviewed, including how the quality of text documents are evaluated, how to develop a recommendation system, how to connect customer needs with design quality, and how to prioritize product engineering characteristics. Finally, a summary concerning the reviews is made in Section 2.7 in order to distinguish this research from previous studies.

2.2 OPINION MINING AND QUALITY FUNCTION DEPLOYMENT

2.2.1 Opinion Mining

Opinion mining, or sentiment analysis, for online reviews refers to the application of techniques such as natural language processing (NLP), computational linguistics, and text analysis to identify and extract subjective information from online reviews, which aims to determine consumer attitudes with respect to the concerned products. Generally, existing research publications in opinion mining are categorized into three types, namely, sentiment identification, sentiment extraction, and opinion retrieval.

2.2.1.1 Sentiment Identification

Technically, several subtasks are included in sentiment identification, such as opinion identification (subjective or objective), polarity classification (positive, negative or neutral), and the identification of sentiment strength. These literatures are generally classified into different perspectives, namely, the word level, the feature level, the sentence level, and the document level.

- Sentiment identification at word level

The sentiment identification at the word level is to identify the sentiment polarity for words. The rationale behind is to utilize the similarities between words and the lexicon extensions from dictionaries or corpus.

Hu and Liu proposed a method to identify customer sentiments from online reviews [HL04b]. Part of speech (POS) tags, sets of synonyms, and sets of antonyms in WordNet were utilized in this method (See Figure 2.1).

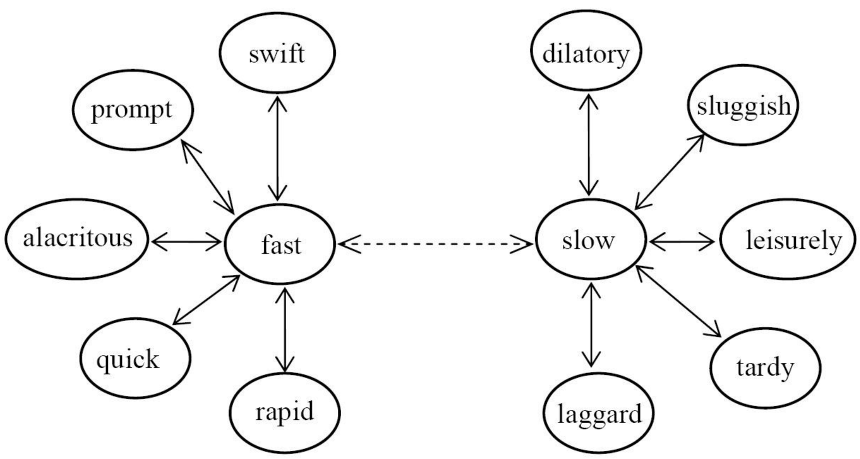


Figure 2.1 The bipolar adjective structure in WordNet [HL04b]

The basic idea is to identify a noun word and its nearest adjective which is often assumed as the opinion word only. In their later research, this method was utilized to make a comparison of opinions on different products. A visual comparison of opinions on two products is illustrated in Figure 2.2 [LHC05].

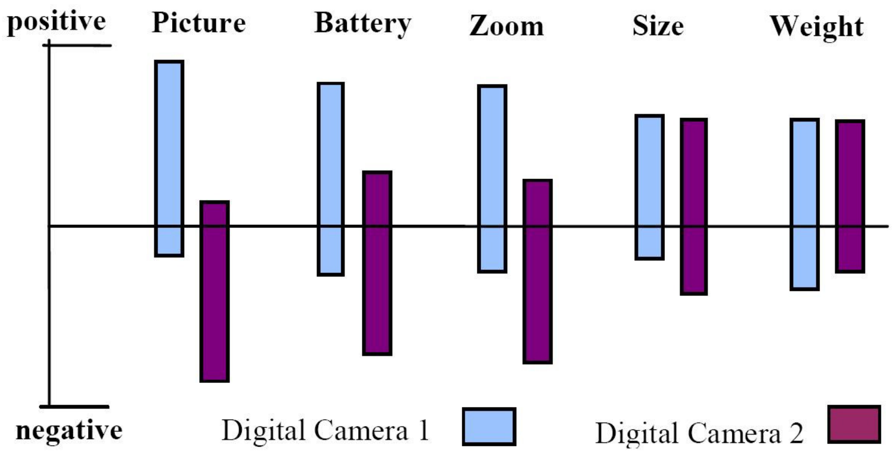


Figure 2.2 A visual comparison of opinions on two products [LHC05]

Similarly, both the random walk algorithm [HR10] and the shortest path method [KMM+04] were applied on WordNet to calculate the similarity between words. In these methods, the sentiment of an opinion word is judged by the similarity between itself and some training words. These training words are marked with manually labeled sentiments. Ding et al. proposed a holistic lexicon-based approach [DLY08], in which some external evidence and linguistic conventions were utilized to identify the sentiment. In their later research, this method was applied for entity discovery and entity assignment [DLZ09].

Different from several dictionary-based approaches, different corpora are also utilized for the sentiment identification. Web documents together with the mutual information between words and phrases [Tur02], as well as domain-oriented sentiment lexicons [DTCY10], were seen to be utilized to identify the sentiment. Also, a lexicalized learning framework based on a hidden Markov model (HMM) was reported [JH09].

- Sentiment identification at sentence level

Different approaches of the sentiment identification at the sentence level have been proposed, including corpus-based approaches, lexicon-based approaches, and combined approaches.

Pang utilized the Support Vector Machine (SVM) method, the Naive Bayes method and the Maximum Entropy method for the sentiment classification [PLV02]. The effectiveness of these classifiers with different text features, such as the Unigram, the Bigram, etc., was compared. The sentiment classification at the sentence level was also regarded as a sequence labeling problem [ZLW08]. A conditional random field (CRF) model was utilized to build a hierarchical structure with the original label set and some additional implicit labels. The sentiment relation between adjacent sentences in the context was measured by this method.

Different from corpus-based approaches, a lexicon-based method was employed in an unsupervised sentiment classification [ZC08]. Benefiting from the identification of sentence sentiments and the enlargement of the sentiment lexicon, an iterative training method was proposed to boost the classification accuracy.

A two-phase hybrid method was reported to utilize both a lexicon-based approach and a corpus-based approach [QZHZ09]. In the first phase, reviews were classified according to a sentiment dictionary. In the second phase, a supervised classifier was trained using a large number of training data. These train data included the reviews classified in the first phase and the reviews classified from a corpus-based supervised method. Similarly, a hybrid method was reported to linearly combine both the lexicon information and specific domain corpus [MGL09]. A matrix factorization model was also designed on the basis of both WordNet and labeled corpus [LZS09].

All these classifiers classify the sentiment the sentences (or words) into positive, negative and neutral.

- Sentiment identification at document level

The sentiment at the document level is usually considered from two perspectives: the contribution from different sentences and the contribution from different topics.

Pang et al. removed objective sentences through a graph-based method. Subjective sentences were then utilized to determine the polarity of document sentiment [PL04]. McDonald proposed a structured joint model for the sentiment identification at different levels, namely, the sentence level, the paragraph level and the document level [MHN+07]. The sentiment identification was considered as a sequential classification with constrained inference problem. A constrained Viterbi algorithm was then utilized to solve this problem.

Lin et al. argued that the sentiment at the document level is dependent on topics or domains [LH09]. A probabilistic modeling method was proposed based on Latent Dirichlet Allocation (LDA). Topics and related sentiments were detected to identify the sentiment at the document level. Similarly, an HMM-based model was also seen to identify the sentiment of documents from the topic aspect [MLW+07].

Other researches on sentiment identification also include cross-lingual sentiment classification and cross-domain sentiment classification.

- Cross-lingual sentiment classification

The lack of sentiment corpora in some particular languages was argued to limit the research progress on sentiment classification. However, many English corpora facilitate researchers in this area to explore this problem by cross-lingual sentiment classification.

Wan trained two classifiers for the sentiment classification of Chinese online reviews [Wan08, Wan09]. The first classifier was trained from English reviews. The second classifier was trained from the Chinese translations of labeled English reviews and online translation services. A co-training algorithm was then used to learn the parameters of the two classifiers. Finally, the results from the two classifiers were combined into a single classifier for sentiment classification. In the testing phase, unlabeled Chinese reviews were translated into English at first. The final sentiment classifier was then applied to predicting reviews as either positive or negative ones.

- Cross-domain sentiment classification

Due to the mismatch between domain-specific words, a sentiment classifier trained in one domain may not work well in other domains. Thus a cross-domain sentiment classification is highly desirable to reduce the domain dependency. Currently, the cross-domain sentiment classification was generally studied from the instance perspective and the feature perspective.

Jiang et al. analyzed the cross-domain sentiment classification from the instance perspective. They argued that the cross-domain sentiment classification usually suffers from two different factors, one for labeling adaptation and the other for instance adaptation [JZ07]. A method was proposed to solve the domain adaptation through instance weighting by considering both the labeling adaptation and the instance adaptation in two domains.

Blitzer et al. proposed a structural correspondence learning algorithm for the cross-domain sentiment classification from the feature perspective [BDP07]. In this algorithm, a set of pivot features which occur in both domains were chosen at first. The correlations between the pivot features and other features were then trained in the unlabeled data from both domains. Other different methods were employed in the cross-domain sentiment classification from the feature perspective, such as using domain independent words to align domain-specific words [PNS+10] and using the common topics with different words in different domains as the bridge to link the domain-specific features [LZ09].

2.2.1.2 Sentiment Extraction

Two categories of tasks are included in sentiment extraction: opinion target extraction and opinion holder extraction. Notice that, for customer reviews, the opinion target usually refers to product features. Statistical patterns of words and phrases in customer reviews are often utilized in relevant research.

Hu and Liu utilized an association mining algorithm to generate a set of frequent nouns or noun phrases [HL04a]. These nouns or noun phrases were regarded as possible product features. A pruning algorithm was then utilized to remove nouns or noun phrases that were unlikely features. Similarly, some heuristic language rules were seen to find words or phrases which match the rule patterns, and these words or phrases were considered as product features [LHC05]. A system utilizing the relaxation labeling was developed to find the product features and the word semantic orientation [PE05]. The relation between sentiment words and product features was also utilized to extract both sentiment words and product features iteratively [QLBC09]. Moreover, a semi-supervised method [ZWTZ09] and a supervised method [LHH+10] were seen to extract product features.

There also exist some limited researches contributing to the opinion holder extraction. For example, A Maximum Entropy ranking algorithm, which was trained from annotated corpora, was proposed to learn the syntactic features and find opinion holders [KH05].

2.2.1.3 Opinion Retrieval

Opinion retrieval refers to find the relevant documents which also contain the opinions on the query [ZJYM08]. Hence, the critical question is how to balance the topic relevance score and the opinion relevance score.

A word-based model was proposed for opinion retrieval [ZY08]. In this model, the Bayesian method was utilized to design a generic method to express the opinion relevance score, and the topic relevance score was linearly combined for opinion retrieval. Based on the query expansion from both the opinion aspect and the dictionary aspect, a unified relevance model was built [HC09]. Several problems were covered in this model, including the topic relevance score, the query-independent sentiment words expansion, the query-dependent sentiment words expansion, etc. Li et al. argued that word-based models failed to capture the association between the opinion and the corresponding target [LZFW10]. A unified graph model was introduced to achieve the sentence-based opinion retrieval.

In Chapter 4, some algorithms for the sentiment identification at word level are utilized in order to extract features from online reviews. These features are utilized to predict the helpfulness of online reviews and recommend the rating value on online reviews.

2.2.2 Quality Function Deployment (QFD)

As a widely used tool, QFD has plentiful applications in the product design area, such as conceptual design, process planning, project management, etc [CW02].

In this research, the objectives are to identify design-preferred online reviews from the perspective of product designers and to build a design-centered knowledge base from online reviews. Hence, the literature review illustrated here only concern about the initial several steps. Several important research publications about the identification of customer needs and their weightings are reviewed here, while existing efforts contributing to the generation of product engineering characteristics and their weightings as well as the association between customer needs and engineering characteristics will be described in the corresponding sections.

The identification of customer needs and their relative weights are the first focus in QFD. The simplest method to prioritize customer needs is based on a numeric scale [GH93]. However, much human interpretation is involved in this method. A pairwise method, which utilized conjoint analysis techniques, was proposed to compare customer needs and determine the relative weights [GG94]. A linear partial ordering approach was also suggested to prioritize customer attributes [HKC04].

In order to fulfill the fast changing of customer needs, the grey theory was seen to combine with QFD [WLW05]. A Markov chain model was also applied to QFD to update the technical measures timely and fulfill the fast changing of customer needs [WS06]. For the understanding of customer behavior, Chan and Ip argued that existing researches about the market-driven product design neglected various influencing factors about customer purchasing and customer value [CI11]. A decision support system was developed to predict the customer purchasing behavior and estimate the net customer value.

The analytic hierarchy process (AHP) was also introduced to calculate the relative importance of customer needs. AHP was originally developed for resource allocation and resource planning [Saa80]. In AHP, at first, the decision problem is decomposed into a hierarchy of sub-problems, which can be easily and independently analyzed. Several elements are then made from a pairwise comparison by individual assessments with concrete numerical values. These numerical values are utilized to rank possible solutions, which help to make the final decision.

In the product design area, a design concept hierarchy with several components is constructed at first. Customer needs, usually coming from survey data and described by a natural language, are then embedded in several components according to their correlations. Finally, AHP is utilized to prioritize various customer needs, which are the input of QFD for the new product design.

Fung et al. combined QFD, AHP and the fuzzy set theory in a hybrid system to prioritize the imprecise customer needs [FPX98]. Chuang also combined AHP and QFD for the location planning under some requirements [Chu01]. A relationship matrix was established to demonstrate the degrees of relationship between each pair of location requirements and location criteria for QFD. In this method, AHP was used to measure the relative importance weighting for each location requirement. An optimal location was finally suggested from the candidate locations. Moreover, a framework to prioritize customer needs in QFD was seen to improve industrialized housing design and manufacturing process [ACMS94]. Wang et al. compared the prioritization matrix method and AHP on several factors, such as, time, cost, difficulty, and accuracy [WXG98]. They concluded that if time, cost and difficulty are the major concerns in product design, the prioritization matrix method is preferred; where accuracy is the major requirement, the AHP method is a better choice.

Some researchers also associated Kano's Model with QFD to obtain and understand customer needs [MH98]. Kano's Model, shown in Figure 2.3 is a useful tool for the understanding of customer needs and their impacts on customer satisfaction. In Kano's Model, different customer needs are categorized based on how well they are able to achieve customer satisfaction, namely, must-be attributes, one dimensional attributes and attractive attributes.

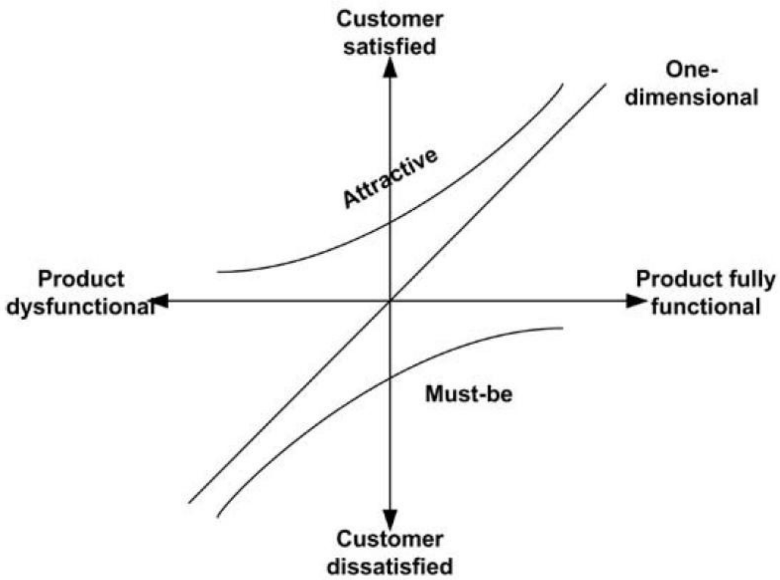


Figure 2.3 The Kano's Model diagram

Customer attributes, analyzed by Kano's Model, were utilized as the input of QFD to help designers to understand customer needs [STX00]. But Kano's Model was combined qualitatively into QFD with little quantitative analysis. Different from this method, quantitative customer satisfaction or dissatisfaction values were utilized to integrate Kano's Model into QFD by establishing a mathematical programming model to optimize product design [LTX07]. A Neuro fuzzy approach was also utilized to generate a customer satisfaction model [KWC09]. An example for notebook computer design was given to demonstrate that this model was better than a statistical regression approach.

2.2.3 Online Reviews for Product Design

Although online reviews are generally accepted as one important source to reflect consumers' sentiments, it just starts to attract researchers in the design area.

Various text mining technologies are utilized to discover valuable information from online reviews for product design.

A text mining system, where online reviews were integrated with the domain knowledge, was reported on knowledge discovery and management in product design [LLL07]. An automatic summarization approach was seen to analyze the topic structure of online reviews [ZLL09]. This approach was utilized to discover and assemble important topics in online reviews. The final summary of multiple reviews was then clustered by the topic structure, and different clusters were ranked according to the importance of different topics. An example of review summarization is shown in Figure 2.4.

Cluster 1 (4 reviews)

Sound - excellent polyphonic ringing tones are very nice (check cons) it also doubles as a radio, which is a nice feature when you are bored.

Cons: ring tones only come with crazy songs and annoying rings, there is only one ring that sounds close to a regular ring.

...

Cluster 2 (3 reviews)

Nice and small and excellent when it comes to downloading games, graphics and ringtones from www.crazycellphone.com I thought this was the ultimate phone when it comes to basic features, but I was disappointed when I saw that it was only a gsm comaptible phone.

...

Cluster 3 (17 reviews)

I've had an assortment of cell phones over the years (motorola, sony ericsson, nokia etc.) and in my opinion, nokia has the best menus and promps hands down.

No other color phone has the combination of features that the 6610 offers.

From the speakerphone that can be used up to 15 feet away with clarity, to the downloadable poly-graphic megatones that adds a personal touch to this nifty phone.

...

Figure 2.4 An example of review summarization [ZLL09]

An unsupervised, domain-independent method was also developed to generate a product-specific ontology for product engineering characteristics from customer reviews [Lee07a]. At first, customer reviews were parsed into phrase sequences, each of which referred to a single concept. All these phrases were clustered into initial concepts. A graph-based method was then utilized to find the maximal clique which defines the corresponding logical set of concepts. Finally, an ontology induction was formed based on the set of concepts. A graphical model was also adopted to extract the relationship between competing products from customer reviews [XLLS11]. A two-level Conditional Random Field (CRF) model with unfixed interdependencies was deployed to extract the dependencies between relations, entities and words of different product reviews. To obtain the rapid change of customer needs, a two-stage hierarchical process was built from online reviews [Lee07b]. At the first stage, the association rule algorithm was used to cluster related product attributes and customer needs into hyper-edges. At the second stage, hyper-rules were applied on hyper-edges to track consumer needs.

Research efforts contributing to the analysis on how online customer reviews influence product economical revenue were also been noticed. Using the rough set theory, the inductive rule learning, and several information retrieval (IR) methods, a system was developed to explore the relationship between the customer reviews and the review ratings [CT12]. Archak et al. decomposed online reviews into different product features [AGI07]. They then estimated the weights of product features, the customer evaluations of product features, and how these evaluations affected the revenue. The effects of negative reviews on consumer attitude were also examined

[LPH08]. Through categories of hypotheses and experiments, the effect of negative reviews was found to depend on the proportion, the quality, and the type of consumer involvements. The degree of high involvement consumers tending to conform to the perspectives of reviewers is found to increase with a higher proportion and a higher quality of negative online consumer reviews. Low involvement consumers are found not to care much about the quality of the negative reviews.

As seen from these research publications, online reviews are seldom evaluated, analyzed and applied directly in the process of product design. However, the objectives of this research are to build a collection of helpful online reviews, to recommend rating values on online reviews by taking designers' personal assessments into consideration, to connect customer reviews with product engineering characteristics, and to prioritize product engineering characteristics based on online customer reviews for improving the current product model. Hence, in the following four sections, some relevant research studies will be introduced, which correspond to the four research objectives of this research.

2.3 QUALITY EVALUATION OF TEXT DOCUMENTS

The first objective of this research is to build a collection of helpful online reviews. In order to distinguish this research, how the helpfulness of online reviews is defined and evaluated in the current researches will be presented in this section. In addition, some contributions towards evaluating the quality of requirements documents in software engineering are introduced. Although not directly relevant to the quality evaluation of text documents, feature selection will be applied in this research to analyze what aspects of online reviews really make product designers regarding them helpful. Hence some basic concepts about feature selection are also described.

2.3.1 Helpfulness Evaluation and Analysis of Online Reviews

The helpfulness of online reviews, concerned in a limited number of researches, is usually evaluated by the percentage of online helpful votes or evaluated from a pre-defined guideline.

2.3.1.1 The Helpfulness Evaluated by the Percentage of Online Helpful Votes

In general, online helpful votes refer to the voting ratio, x/y, that is, x out of y people find a particular review helpful, e.g., "204 out of 209 people found the following review helpful", as shown in Figure 1.1. In some research publications, the percentage of online helpful votes was also regarded as the golden criterion to define the helpfulness of product reviews. The question of helpfulness prediction was formulated as a binary classification [OS09, ZT10a], multiple classification or regression [DKK+09, LHAY08, MLD09, YLHA10] with several categories of features, such as sentiment features, user reputation or expertise features and information quality-based features.

Mahony regarded those reviews receiving more than 75% positive helpfulness votes as helpful ones [OS09]. A binary classification method was utilized to recommend helpful hotel reviews with reputation features, content features, social features and sentiment features. Zhang defined helpful reviews as those receiving more than 60% positive helpfulness votes [ZT10a]. The information gain-based approach was utilized to predict the helpfulness of online reviews.

Liu et al. investigated three important factors from observations, including reviewers' expertise, writing styles and timeliness [LHAY08]. Three observations about how the helpfulness of online reviews is influenced by these factors were presented. The arguments are "expertise might be well reflected through reviews they compose", "due to the large variation of the reviewers' background and language skills, the online reviews are of dramatically different qualities", and "the helpfulness of a review may significantly depend on when it is published" (See Figure 2.5). These factors were then combined linearly to estimate the percentage of online helpful votes. In their later research, this model was utilized to forecast the sales of a product [YLHA10].

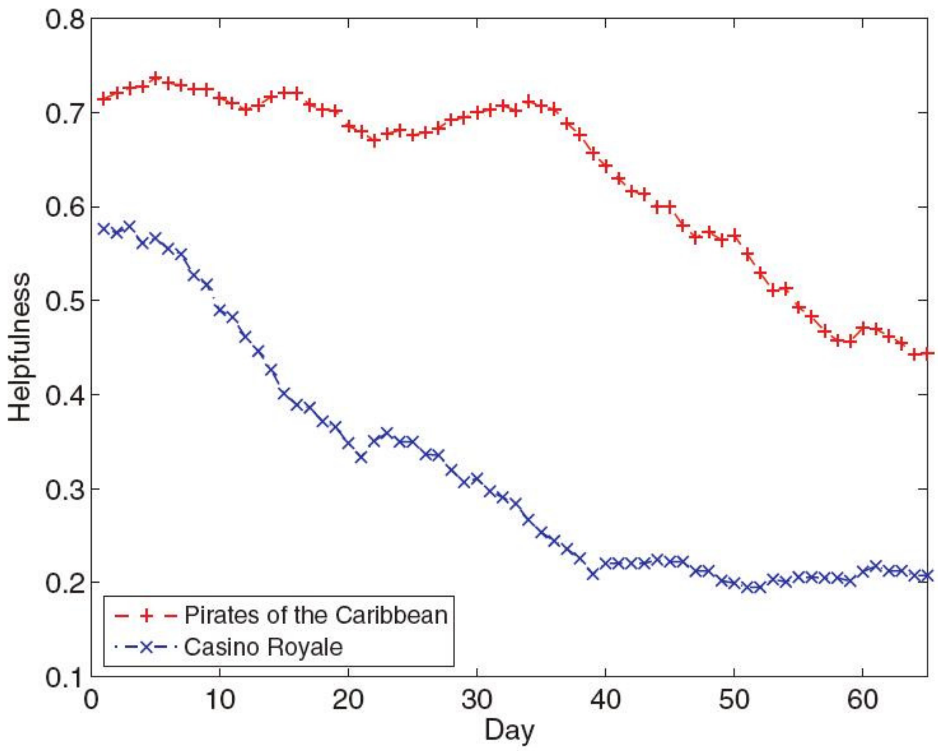


Figure 2.5 An example of review helpfulness vs. time of review [LHAY08]

Miao et al. also regarded the percentage of online helpful votes as the evaluation metrics for the helpfulness of online reviews [MLD09]. A linear combination of the helpfulness predicted from this model and the relevance model was developed to retrieve the sentiment information of online reviews. A sound analysis was conducted regarding several hypotheses that might influence the percentage of online helpful votes [DKK+09]. These hypotheses include the conformity hypothesis, the individual-bias hypothesis, the brilliant-but-cruel hypothesis and the straw-man hypothesis. Finally, the percentage of online helpful votes were said to not just depend on the content but also "on how the expressed evaluation score relates to other evaluation scores of the same product."

2.3.1.2 The Helpfulness Evaluated by a Pre-defined Guideline

Different from evaluation metrics using the percentage of online helpful votes, Liu et al. [LCL+07] argued that the helpfulness represented by the percentage of online helpful votes is not fair due to three kinds of biases: imbalanced vote bias, winner circle bias and early bird bias. The imbalanced vote bias was interpreted as "users tend to value others' opinions positively rather than negatively." Figure 2.6 shows that a half of reviews have the percentage of helpful votes bigger than 0.9. The winner circle bias can be explained by Figure 2.7. It illustrates that "the top two reviews hold more than 250 and 140 votes respectively on average, while the numbers of votes held by lower-ranked reviews decrease exponentially." The authors said that "the higher ranked reviews would attract more eyeballs and therefore gain more people's votes."

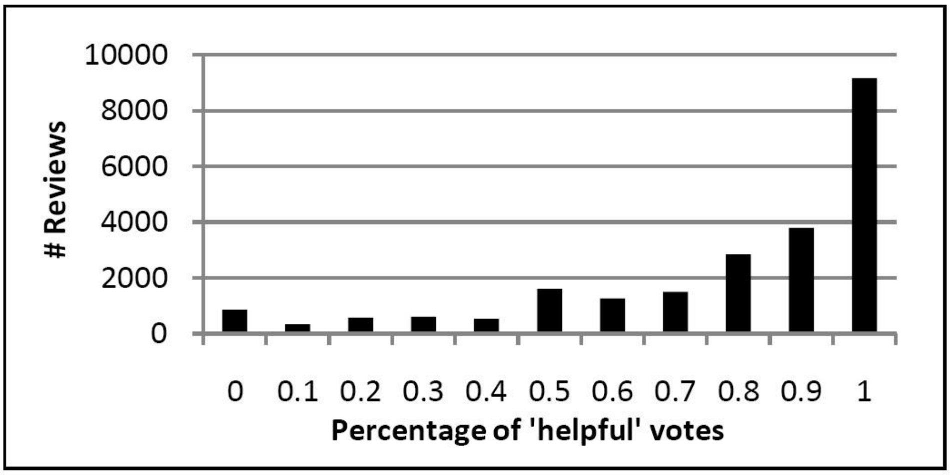


Figure 2.6 Imbalance vote bias [LCL+07]

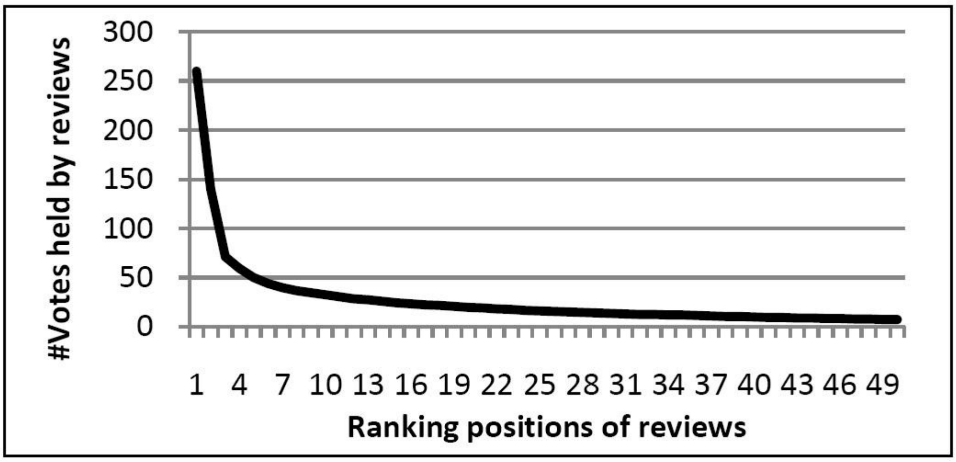


Figure 2.7 Winner circle bias [LCL+07]

The early bird bias, which says "the earlier a review is posted, the more votes it will get", can be found in Figure 2.8. Actually, a similar trend presented in Figure 2.8 can also be found in Figure 2.5. This phenomenon was explained as "reviews posted earlier are exposed to users for a longer time". Further, they concluded that the helpfulness votes are not necessarily strongly correlated with certain measures of review quality. In Chapter 3, this conclusion will be further confirmed by the proposed exploratory case study on review helpfulness.

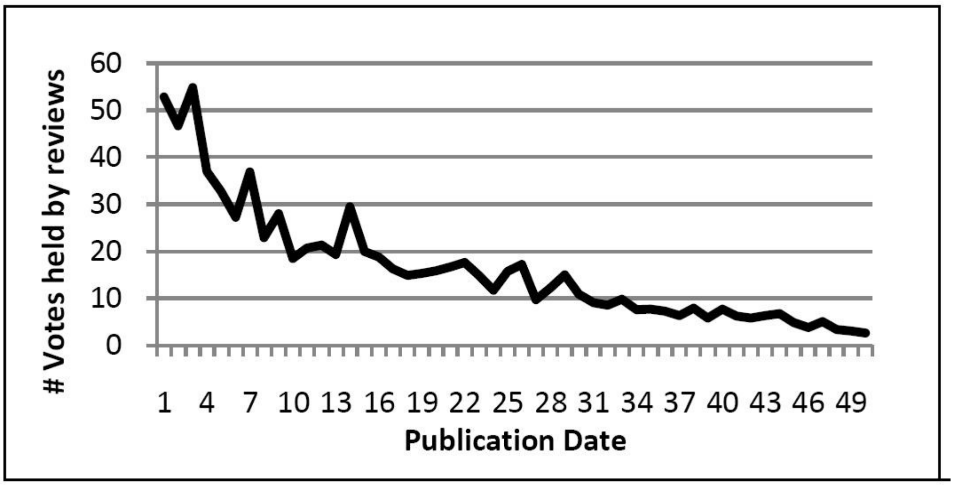


Figure 2.8 Early bird bias [LCL+07]

As introduced in [LCL+07], four human annotators were then hired to evaluate the helpfulness. The annotators' evaluations were found to be different greatly from the percentage of online helpful votes. They claimed that, given a guideline for the helpfulness evaluation, annotators achieve consistence on the helpfulness evaluation through a kappa statistic. Finally, the helpfulness prediction was modeled as a multiple classification. Several categories of features and SVM were employed to predict the helpfulness.

Inspired by the above research [LCL+07], a quality evaluation framework for online reviews was developed [CT10]. Different from previous efforts that reviews were only evaluated as helpful or not helpful, reviews were classified into five categories: "high-quality", "medium-quality", "low-quality", "duplicate" and "spam". Also, several groups of features and SVM were utilized to predict the quality of product reviews. Likewise, Li et al. argued that the annotated corpus for each domain of interest is infeasible [LYZW11]. A snippet-based unsupervised learning approach was devised to estimate the sentiment of online reviews. This approach was utilized to classify whether online reviews are recommended or not recommended. However, only reviews receiving the unanimous judgments of two annotators were employed. A probabilistic distribution model was also seen to judge the helpfulness of online reviews [ZT10b]. The Expectation Maximization (EM) algorithm was utilized to find the probability distribution of the helpfulness in a given training corpus. The bag-of-words model was used to represent the review text and to predict the helpfulness of online reviews. This method was reported to reach median correlation between the predicted helpfulness values and the human evaluation.

Although various algorithms are proposed to predict the helpfulness of online reviews, the helpfulness of online reviews is evaluated by a manually defined evaluation guideline. Also, it is easily to understand that the evaluation will be highly possible to achieve consistence, given a manually defined evaluation guideline. However, whether the helpfulness evaluation is reasonable for product designers is not explored. Hence, in order to better understand how the review helpfulness is perceived by product designers, an exploratory case study will be conducted in Chapter 3.

2.3.2 Quality of Requirement Documents in Software Engineering

Some contributions towards customer understanding on requirements documents can be found in the software engineering domain. Some researchers made efforts on the quality evaluation. An example of using QFD for the requirement analysis can be found in Figure 2.9. This method was utilized to solve the conflicting viewpoints and attitudes from different parties during the software requirements validation [RAR05].

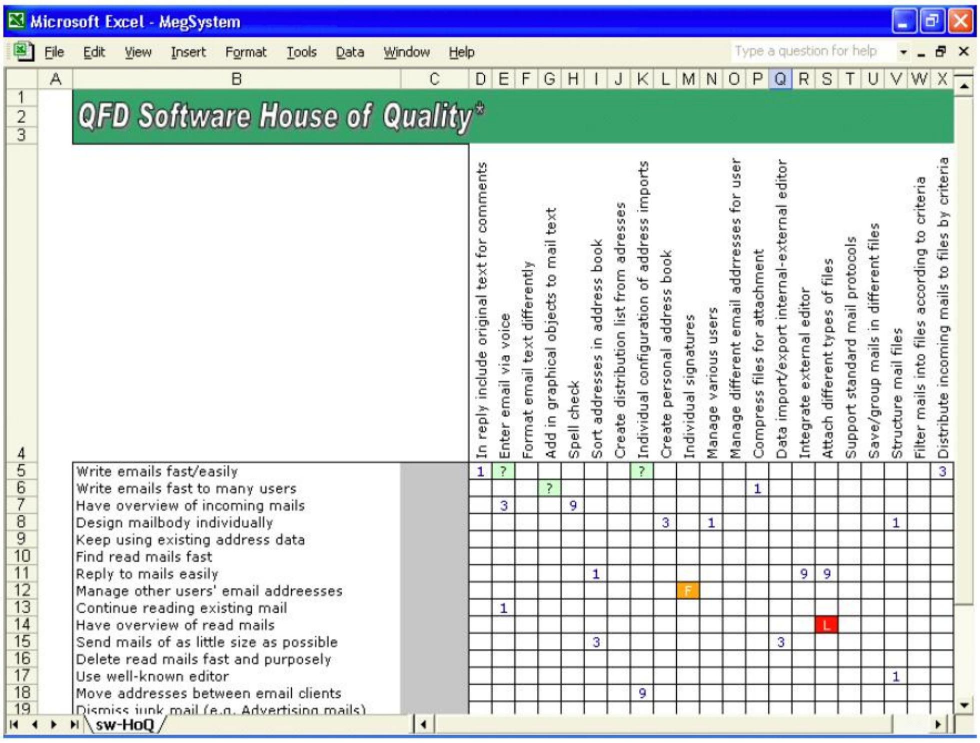


Figure 2.9 An example of using QFD in software requirement analysis [RAR05]

In some research efforts, the quality of requirement documents is seen to be evaluated by a quantitative approach [Ken96], by some natural linguistic patterns [FFGL00], etc. The quality of requirement documents was both evaluated by linguistic statistic indicators and the evaluations have to rely on manual efforts to a large extent. An automatic evaluation algorithm was proposed [MWG09]. Ten linguistic rules were utilized to extract functional requirements from software requirement specifications. A language was also proposed to describe the requirement specifications [VS06], as shown in Figure 2.10.

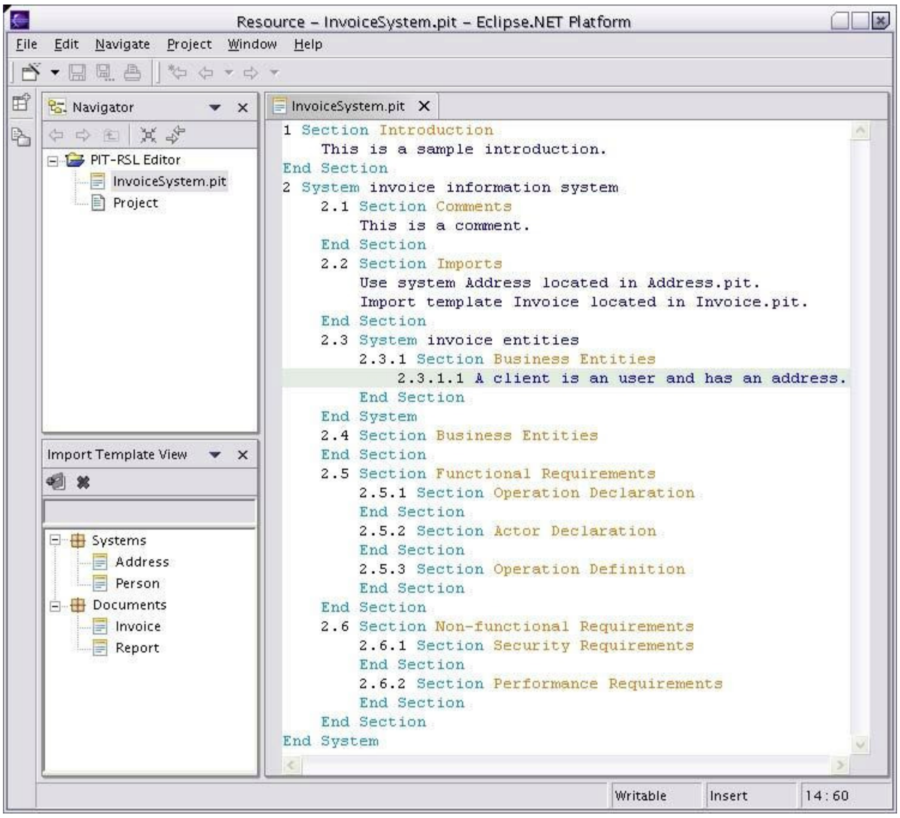


Figure 2.10 A language to describe the requirement specifications [VS06]

The language was formed by the identification of the most frequently used linguistic patterns in requirement documents, written in natural language. To guarantee the consistency of the written requirements, the requirements were analyzed by parsing tools, and validated according to the language syntactic and semantic rules.

Some machine learning algorithms were also introduced to evaluate whether the requirement specifications are ambiguous. A binary classification (ambiguous or unambiguous) using a decision tree algorithm was proposed at both the discourse-level and the sentence-level [HOK07]. Also, two classifiers were built to find similar sentences and reduce the ambiguity in the requirement specifications [PG08].

While such efforts also contribute to customer understanding at large, it differs greatly from online review evaluation. First, quality evaluation of requirement documents in software engineering focuses primarily on the effective gathering of explicit design requirements from designated user survey documents which often contain tables, diagrams and questionnaires. In contrast, online reviews are primarily free text largely written by consumers, which is one important characteristic of online reviews. Their evaluation by designers emphasizes on, for example, its value to alert designers to certain missing attributes that consumers would wish to have, but designers may not be aware of. Secondly, different from requirement documents in software engineering, online reviews contain a large number of sentences with either strong or weak sentiments. Such sentiments coupled with either existing or potential product attributes attract much more attention from designers. These sentimental expressions contain valuable inputs given by customers and often help designers when they envision new products or improve current models. Thirdly, comparing with a limited number of requirement documents provided in software engineering, designers are often overwhelmed by the sheer number of online reviews, not to mention other technical challenges.

2.3.3 Feature Selection

Feature selection is to select a subset of features, which helps people to acquire some important features, understand their relationship, and improve the performance of learning algorithms. A typical process of feature selection consists of four basic steps, namely, subset generation, subset evaluation, stopping criterion, and result validation (Figure 2.11).

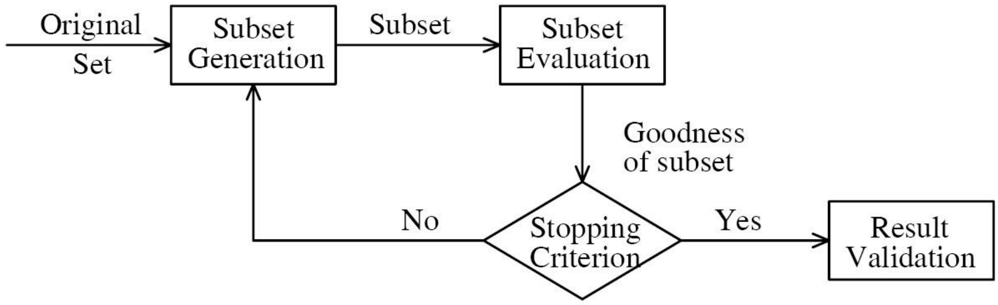


Figure 2.11 Four key steps of feature selection

The goodness of a subset is measured by an evaluation criterion. The number of features increases with the complexity of the problem. Finding the optimal feature subset is usually difficult and many problems related to feature selection have been proven to be NP-hard.

Feature selection is utilized in many areas of data mining, such as classification, regression, clustering, and association rules. The high-dimension data possibly lead to a lower accuracy and high computation cost. Hence, feature selection becomes an important step in many data mining applications.

Principal Component Analysis (PCA) is one of well known data preprocessing methods, which is often used as a feature selection method. The linear dependencies among features are captured and the most representative features are identified. The feature space is then reduced by these representative features with least possible lost information about the original data.

Three different variants of PCA, namely, the covariance matrix, the correlation matrix, and the feature matrix, were conducted to evaluate the classification performance and the runtime performance with various machine learning algorithms [JGDE08]. In this work, a mean shift of all features was performed at first in order to make the mean for each feature become zero. The performance of the above three different variants was then compared. The corresponding eigenvalues and eigenvectors of the covariance matrix, the correlation matrix, and the original feature matrix were normalized by their standard deviations. Features were ranked according to the correlation coefficients of the first principal component, and the top ranked features were chosen as the selected features.

Different feature selection algorithms are classified as three categories [GE03], namely, the filter model, the wrapper model, and the hybrid model.

The first category is the filter model [MG99]. The selection and evaluation of feature subsets is independent from learning algorithms. Several well-known similarity-based feature selection algorithms, such as cosine similarity and matching similarity, are all filter methods. The rationale behind these similarity-based feature selection algorithms is that one feature would be regarded as important if it has the maximum similarity with the target. In these similarity-based feature selection algorithms, the similarity between each feature and the target is calculated. The features are ranked according to the similarity. Yang et al. conducted a comparative research on several filter algorithms [YP97]. The information gain algorithm and the chi-square algorithm were found to be among the most effective methods for text classification.

The second category is the wrapper model. A predefined mining algorithm is usually utilized to evaluate the goodness of subsets of features [KJ97, LY05]. Each generated feature subset is evaluated in terms of which subset is more suitable for the mining algorithm. Hence, different mining algorithms lead to different feature selection results. Different search strategy functions for the mining algorithms can also produce different wrapper algorithms. Because mining algorithms are used to control the selection of feature subsets, the wrapper model tends to result in good performance of the feature subsets selection. But it tends to be more computationally expensive than the filter model.

The third category is the hybrid model [Das01]. It takes advantage of the above two models by exploiting their different evaluation criteria in different search stages. Both an independent measure and a mining algorithm are utilized to evaluate feature subsets. The independent measure is to decide the subsets for a given cardinality and the mining algorithm is to select the final best subset among the best subsets in different cardinalities. The best subset is found in each cardinality iteratively. If the best subset at current cardinality is better, the algorithm continues to find the best subset at the next cardinality. Otherwise, it stops and the current best subset is selected as the final best subset. The quality of selecting best subsets from a mining algorithm provides a natural stopping criterion in the hybrid model.

2.4 RECOMMENDATION SYSTEMS

The second question of this research is how to recommend rating values on online reviews with consideration of designers' assessments. This question will be explored based on existing researches of recommendation systems.

Recommendation systems are a subclass of information filtering system that seek to predict the 'rating' or 'assessment' that a user would give to an item or social element they had not yet considered, using a model built from the characteristics of an item or the user's social environment [RRS11]. Several state-of-the-art techniques of recommendation systems will be described in this section.

With the fast development of the Internet, recommendation systems become a top research topic for filtering the abundant information. Users' historic evaluations are utilized in recommendation systems to predict potentially further interests of its users. Some particular items are recommended to their users according to the personal assessment.

Recommendation systems assist their users to find interesting and valuable information about books, music, movies, etc. One of the first recommendation systems, Tapestry, was reported in 1992 [GNOT92]. In this recommendation system, the phrase, "collaborative filtering" was utilized. This term has been widely adopted regardless whether recommendation systems explicitly collaborate with recipients or not and whether they suggest items or not.

Generally, there are three categories of recommendation systems, namely, the content-based recommendations, the collaborative recommendations, and the hybrid approaches.

2.4.1 Content-based Recommendations

In content-based algorithms, the content information of items is utilized to make recommendations. The item similarity, which is calculated by comparing the content of items, is often utilized in these content-based algorithms. Also, the profile of the items that users have rated in the past is involved. Hence, the recommendation becomes to find items that are similar to those already preferred by target users.

The focus of content-based algorithms is on the recommendation of text documents, such as, websites or news messages. Therefore, the cosine similarity, a popular one in text mining, is utilized as the similarity function. In addition, some other techniques in text mining are also utilized in content-based algorithms. For example, the Rocchio algorithm was applied to averaging the content of items [Lan95, BS97].

For the weights of different words, the TF-IDF (term frequency - inverse document frequency) method, is often utilized. In the TF-IDF method, each document is represented as the weight vector of different words. The weight vector is then utilized by a classifier to estimate whether an item is recommended [PB97, MBR98].

2.4.2 Collaborative Recommendations

In collaborative recommendations, the items which are previously rated by other users are utilized to make the recommendation. Hence, past evaluations of a large group of users are required. In collaborative recommendations, the information about a lot of other users that have similar tastes is collected at first. The items which are preferred by other users with similar tastes are then recommended to target users.

Two types of collaborative recommendations are widely used, namely, the memory-based algorithms and the model-based algorithms.

2.4.2.1 Memory-based Collaborative Filtering

In memory-based collaborative filtering algorithms, recommended items are either those which are preferred by other users who share similar assessments with target users, or, those which are similar to other items preferred by target users.

The unknown rating for a specific user of an item is often aggregated from the ratings of other similar users or items. The aggregation can be either an average or a weighted sum. The weight can be a distance measure, such as the similarity, the correlation, etc. The more similar between users, the higher weight will gain in the prediction. However, different rating scales may be utilized by different users. Hence, an adjusted method is more widely used. Rather than using the absolute rating values, the deviation or bias from the average rating of the user or item is utilized.

There also exist several extensions to improve the recommendation. Memory-based approaches are found not to obtain a satisfactory result when there are relatively few ratings since the similarity is based on the intersection of the items [BHK98]. Some default values for the missing ratings were assigned to improve the recommendation accuracy. Different from several approaches that use the similarity of users, the similarities between items were utilized by both a correlation-based approach and a cosine-based approach [SKKR01]. The methods that use the similarity of items were also extended for the top item recommendations [DK04].

Methods of item-based similarities are claimed to have better computational performance than user-based collaborative methods.

2.4.2.2 Model-based Collaborative Filtering

In model-based collaborative filtering algorithms, items are recommended by models that are trained to identify patterns from the input data. Statistical information and the techniques of machine learning are utilized to recommend items.

The recommendation was regarded as a classification task by some researchers. A classification method using Bayesian belief nets was introduced into the recommendation systems [SK06]. In this method, ratings from different users were assumed to be independent. The probability of the rating for an item was calculated by this classification method. The rating with the highest probability was classified as the predicted class. However, Bayesian belief nets were argued that they do not directly maximize the classification accuracy [SSG+03, GSSZ05]. Extended logistic regression methods were proposed for recommendation in order to obtain high classification accuracy for both complete data and incomplete data.

Recommendation was also regarded as a regression task by some researchers. A probabilistic factor analysis model was proposed [Can02]. In this model, the items which are not rated by users were assigned with the average rating of other items. A regression model was then utilized as the initialization of Expectation Maximization (EM) algorithms to make a recommendation. A regression-based approach on numerical ratings was also proposed [VO05]. In this approach, the item similarity was utilized. A set of simple linear models was then combined to make a recommendation.

2.4.3 Hybrid Approaches

In hybrid methods, collaborative recommendation algorithms are combined with content-based recommendation algorithms or other collaborative algorithms. For example, the results of hybrid algorithms may come from different collaborative recommendation, collaborative recommendation with content-based recommendation, or collaborative recommendation with other recommendation systems.

In one of early hybrid method, through a weighted average approach, a content-based recommendation method was combined linearly with the collaborative recommendation method for newspapers [CGM+99]. Similarly, a majority voting approach was used to combine results from a content-based recommendation and a collaborative recommendation [Paz99]. In addition, for the items which are not rated by users, they were proposed to be filled by a content-based method [MMN02]. A weighted Pearson correlation-based collaborative algorithm was then utilized to make the recommendation for an item from a particular user. Burke reviewed some recommendation techniques [Bur02]. A weighted hybrid system, combining several recommendation techniques, was built for the recommendation of restaurants.

A hybrid method which combines both memory-based and model-based methods was proposed to make the recommendation [PHLG00]. In this method, a generative method was utilized for a user profile by sampling uniformly from other users and adding Gaussian noise to their ratings. Both the probability that a user has the same personality type as others and the probability that a user will like new items were then calculated. A probabilistic method that combines both the memory-based methods and the model-based methods was also reported [YST+04]. In this method, a mixture model was built from user profiles and the posterior distribution of user profiles was utilized to make the recommendation.

2.5 CONNECTING CUSTOMER NEEDS WITH DESIGN QUALITY

The third question of this research is how to connect customer reviews with product engineering characteristics. The corresponding researches in the design area are also concentrated on finding association between customer needs and product engineering characteristics, which is one of important researches of QFD methodology as illustrated in Chapter 1. Hence, in this section, how customer needs are translated into QFD will be presented at first. Moreover, some researchers in computer science are dedicated to explore how to identify the meaning of a word in a given context automatically, which is called word sense disambiguation (WSD). The third question in this research is similar to the objective of WSD. Thus, some state-of-the-art WSD techniques are also reviewed.

2.5.1 Translating Customer Needs into QFD

After successfully identifying customer needs, designers almost always consider how to translate customer needs into product design. Specially, the association between customer needs and product engineering characteristics is one important question in QFD. Several contributions are seen in this area.

Generally, all these researches have to cope with the inherent vagueness of human language and subjective judgment in the voice of the customers [KMDE00]. This problem is often seen to be analyzed by introducing the fuzzy set theory into QFD. For instance, to meet customer needs and facilitate information sharing between designers, a market-driven design system based on the fuzzy logic was developed [HPFO01]. This system was utilized to translate the market information into product specifications. Also, a fuzzy linear regression method was proposed to estimate the uncertainty in the functional relationship between customer needs and product engineering characteristics for product planning, which is one important process in new product design based on QFD [FCT06]. A fuzzy expert system was also proposed to identify important product engineering characteristics [KCBC07]. The fuzzy relationship between customer needs and product engineering characteristics as well as the fuzzy correlation among product engineering characteristics in QFD were analyzed by this fuzzy expert system. Moreover, linguistic variables and fuzzy numbers are found to be more appropriate to describe the inputs in QFD [CFT06]. This method is different from the previous efforts where the input data are assumed to be precise and treated as numerical data only [Aka90, GH93, GG94]. In an uncertain and vague environment, Kano's Model was reported to be integrated into QFD to quantify customer needs [MTCK08]. A fuzzy multi-objective model was then utilized to balance customer satisfaction and development cost.

2.5.2 Word Sense Disambiguation

Word sense disambiguation (WSD) is to identify the meaning of a word in a given context, which is an important research area in computational linguistics. According to the source of knowledge for sense disambiguation, different WSD algorithms can be classified into three categories, i.e., knowledge-based methods, supervised methods, and unsupervised methods [Nav09].

2.5.2.1 Knowledge-based Methods

In knowledge-based methods, without any corpus evidence, dictionaries or lexicon are relied on. Senses of words are provided in these knowledge resources, which contribute to the success of knowledge-based methods.

A method utilizing semantic relations in WordNet to expand the set of overlapped words was proposed [BP02]. This set of overlapped words was utilized for WSD. The word definition in the dictionary was also utilized in some researches [IMK08]. In this method, the definition-based similarity method was extended with different other similarity methods. The nearest neighbor method was then employed to compare the similarity of the target word for WSD [IMK08].

The graph-based methods were also introduced into WSD. For example, the semantic relations in WordNet were utilized to build chains of words [GM03]. Each chain was assigned with different weights according to the different semantic relationships. A disambiguation graph approach was then developed for WSD. Graphical structures for sequence data labeling, using random walks on encoding label dependencies, were also built. These graphical structures were then utilized in a graphical algorithm for unsupervised WSD [Mih05].

2.5.2.2 Supervised Methods

In supervised methods, annotated corpora and several classification algorithms in machine learning were utilized to predict the word sense. Different supervised learning methods for WSD utilized the annotated words [MC03, HMP+06].

Some supervised methods contributing to the domain specific WSD were reported. For example, through several categories of comparison experiments, extracting the predominant sense information from a mixed-domain corpus was found to be more accurate and the extraction of predominant sense information would perform well when the training and testing data are in the same domain [KMC05]. Using domain specific corpora as the input, a supervised method was proposed to adapt to different domains where manually labeled data were not always available [MKWC07].

Although supervised methods for WSD is one type of successful techniques for WSD, the core problem of supervised methods for WSD is that manual tagging is not always available. To build the training corpus, different possible contexts have to be considered for the word senses. An extensive analysis about different data characteristics, supervised algorithms, and parameters that might influence the performance of WSD techniques was conducted [YF02]. The similarities, differences, strengths and weaknesses of different algorithms were also summarized [YF02].

2.5.2.3 Unsupervised Methods

In unsupervised methods, word senses are derived directly from unlabeled corpora. Two categories are usually included, namely the type-based approach and the token-based approach [Ped06].

In the type-based approach, words are clustered according to the contexts in which they occur. For example, a clustering algorithm was developed for WSD [PL02]. In this algorithm, clusters of words were built according to their contextual similarities. Using the nearest neighbor method, each word was assigned to its most similar cluster and different clusters were used to represent different word senses.

In the token-based approach, words are clustered together according to the other words in each instance. For instance, a context-group clustering algorithm was reported [Sch98]. In this algorithm, word senses were induced from a corpus without labeled training instances or other external knowledge sources. Words, contexts, and senses were clustered according to the semantic similarity. The unsupervised clustering was then applied on both training and testing data. Word senses were interpreted as clusters of similar contexts of the ambiguous words.

2.6 PRIORITIZING PRODUCT ENGINEERING CHARACTERISTICS

The fourth objective in this research is to prioritize product engineering characteristics from online reviews to improve the current product model. Several state-of-the-art machine learning algorithms are examined in this research to verify whether they are applicable for this question. Learning to rank, as one of major concerns, is analyzed. Hence, for the better understanding of this argument, several learning to rank algorithms are reviewed. Also, in the design area, several researches are also conducted on the generation of product engineering characteristics and their weightings. Hence, in this section, some research publications towards weighting the importance of product engineering characteristics are introduced.

2.6.1 Importance Weighting of Product Engineering Characteristics

Due to the time and budget limitations, when designers conceive to improve current product models, it is usually unreasonable to consider product engineering characteristics without any bias. Importance weighting of product engineering characteristics, one important problem in QFD, becomes crucial for the resource allocation as well as the final decision-making.

A nonlinear programming model was proposed to prioritize product engineering characteristics in fuzzy environments [WC11]. Two numerical examples were shown to verify the availability of this model. A fuzzy weighted average method was also proposed to prioritize product engineering characteristics in fuzzy QFD [CFT06]. In this method, a discrete solution was obtained through changing the fuzzy weighted average problem to a pair of fractional programming problem for each product characteristic. Both the human perception and the customer heterogeneity are found to influence the importance of product engineering characteristics in QFD, but most of the relevant researches only center at one of them [KYCC11]. Accordingly, a fuzzy group decision-making method combing both a fuzzy weighted average method and an ordinal ranking was proposed to incorporate the two influential factors. This approach was argued to be better than the method proposed early by Chen, Fung and Tang [CFT06].

Some other researchers argued that the importance of product engineering characteristics in QFD can be evaluated from two aspects, namely, the needs of customer aspect and the needs of manufacturer aspect [GCXZ10]. From the perspective of customer needs, the analytic network process was utilized to estimate the initial importance of product engineering characteristics by considering the relationships of customer needs, product related engineering characteristics, and service related engineering characteristics. The fuzzy set theory was then applied in the analytic network process to deal with the uncertainty in decision-making. From the perspective of manufacturer needs, the data envelopment analysis was employed to adjust the initial weights of product related characteristics by considering both the business competition and the implementation difficulty.

Different models were also employed for importance weighting of product engineering characteristics. For example, QFD was regarded as a grey system [LZG09]. The relationships between customer needs and product engineering characteristics were then determined with the grey relational matrix. A grey method was utilized to prioritize product engineering characteristics. Kano's Model was also seen to be integrated with QFD to recognize the importance of product engineering characteristics [CJSM11].

2.6.2 Learning to Rank

Learning to rank is a type of supervised learning. The objective is to construct a ranking model from training data, which sorts new objects according to their degrees of relevance [JLLZ07]. A framework of learning to rank is shown in Figure 2.12.

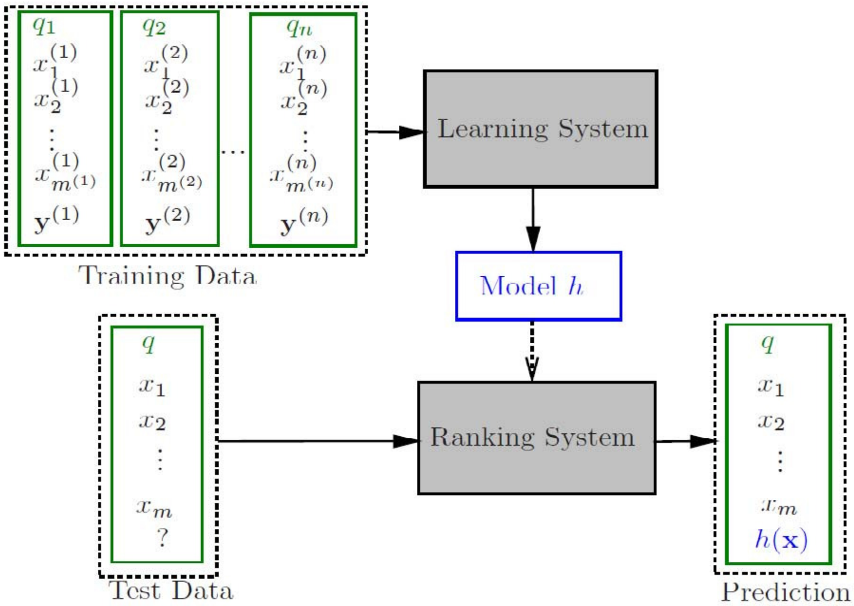


Figure 2.12 Learning to rank framework [JLLZ07]

The training set of learning to rank consists of n training queries q_i (i=1… n), their associated documents, and the corresponding relevance judgments. The associated documents are represented by a feature vector $x_{m^{(i)}}^{(i)}$, where $m^{(i)}$ is the number of documents associated with query q_i. A specific learning algorithm is then employed to learn a ranking model h. The output of the ranking model h is expected to be able to predict the ground truth labels for documents in training set as accurately as possible, in terms of a loss function. In the test step, when a new query

q comes, the ranking model h is applied to sorting the documents x_j (j=1...m) according to their relevance to the query q, and return the corresponding ranked list.

The algorithms of learning to rank are generally classified into three types, namely, pointwise approaches, pairwise approaches and listwise approaches. Different input and output spaces, hypothesis and loss functions are defined in different approaches.

2.6.2.1 Pointwise Approaches

The input space of a pointwise approach contains each single document, which is represented as a feature vector. For the output space, the pointwise approach contains the relevance degree of each single document. The hypothesis space of the pointwise approaches contains functions that take the feature vector of a document as input and predict the relevance degree of the document. The loss function of the pointwise approach examines the accurate prediction of the ground truth label for each single document.

Most of pointwise approaches apply existing machine learning algorithms directly. These approaches can be further divided into three subcategories, namely, regression-based algorithms, classification-based algorithms, and ordinal regression-based algorithms.

A regression method was utilized to predict the relevance of a single document and the square loss function was applied in this method [CZ06]. An SVM-based approach was proposed to transform a ranking problem into a classification on single document [Nal04]. A linear weighting function was utilized as the relevance scoring function.

Shashua and Levin regarded the ranking problem as an ordinal regression [SL03]. Two strategies, the fixed-margin strategy and the sum-of-margins strategy, through the generalization of SVM, were suggested to learn the thresholds and to maximize the margins for each category. In the fixed-margin strategy, each document is emphasized to be correctly classified into its target category (See Figure 2.13). The predicted score of each document, which is defined by several parameters, should be confined in a region with certain soft margins.

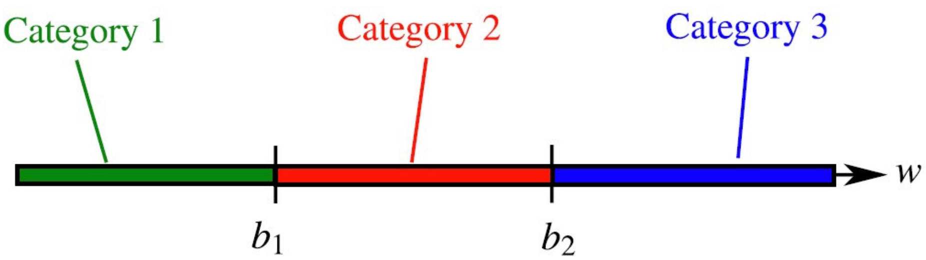


Figure 2.13 The fixed-margin strategy of the pointwise approach [SL03]

In the sum-of-margins strategy, the predicted score of each document is centered to be confined in a region which is defined by two model parameters (See Figure 2.14).

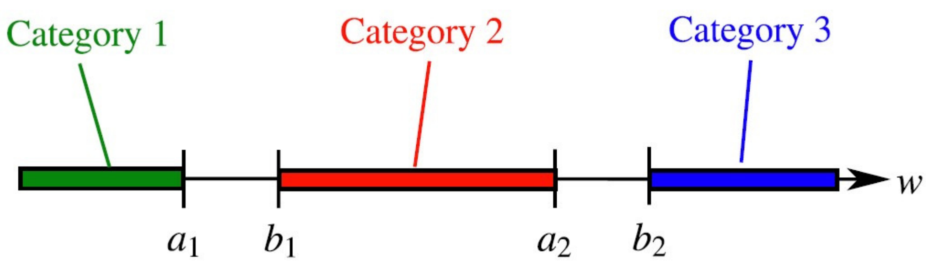


Figure 2.14 The sum-of-margins strategy of the pointwise approach [SL03]

The problem of the pointwise approach is that the input is a single document, and the relative order between documents is neglected. However, the ranking problem naturally cares about more on the relative order of documents.

2.6.2.2 Pairwise Approaches

The input space of a pairwise approach contains a pair of documents, which is also represented as feature vectors. For the output space, a pairwise approach contains the pairwise preference between each pair of documents. The hypothesis space of the pairwise approaches contains functions that take a pair of documents as input and output the relative order between them. The loss function of the pairwise approach measures the inconsistency between the predicted relationship and the ground truth label for the document pair.

The pairwise approaches do not target at accurately predicting the relevance degree of each single document. The relative order between two documents is cared about. An example of a pairwise approach, where the output takes values from $\{+1,-1\}$, is illustrated in Figure 2.15.

In the pairwise approaches, ranking is usually reduced to a classification on document pairs. The preference between document pairs is evaluated in these pairwise approaches.

A neural network was built to learn a preference function for all possible document pairs in training data [BSR+05, RPMS11]. A boosting approach on document pairs was also utilized to combine ranking functions in RankBoost

[FLSS03]. Based on SVM, RankSVM was proposed to perform the pairwise classification [HGO00, Joa02]. RankSVM differs from SVM at its constraint part and the loss function, which was built from document pairs.

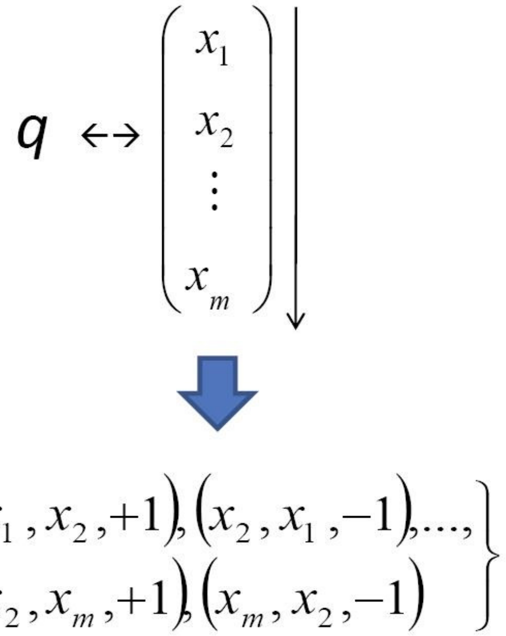


Figure 2.15 The pairwise approach for learning to rank [JLLZ07]

A single hyperplane in the feature space is employed by RankSVM, which is arguable to handle complex ranking problems [QZW+07]. Multiple hyperplanes were proposed to train a ranking model for document pairs as shown in Figure 2.16. Finally, the ranking results predicted by each ranking model were aggregated to produce the final ranking result.

Although pairwise approaches have their own advantages, the ignored fact is that ranking is a prediction question on a list of documents rather than on one document or a document pair.

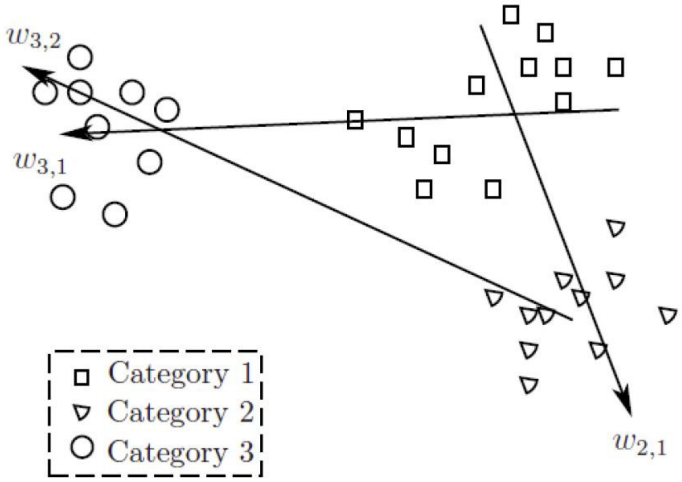


Figure 2.16 Training multiple hyperplanes in a pairwise approach [QZW+07]

2.6.2.3 Listwise Approaches

The input space of a listwise approach contains the entire group of documents associated with a query. For the output space, a listwise approach contains the ranked list of the documents. The hypothesis space of the listwise approaches contains functions that operate on a set of documents and predict their permutation. The loss function of the listwise approach includes two types, namely, measure-specific loss function and non-measure-specific loss function. The first type is explicitly related to the evaluation measures, while the second is not.

In the listwise approaches, the entire set of documents associated with a query in the training data are taken as the input and their ground truth labels are predicted. Since a permutation has a natural one-to-one correspondence with a ranked list, some researchers started to analyze how to apply the probability distributions on permutations for ranking problem. A list of documents was utilized as the training instances in ListNet [CQL+07]. The Plackett-Luce model, which is a famous model for permutation probability distributions, was utilized to define a listwise loss function. In their later research [XLW+08], the properties of related algorithms were described and, based on maximum likelihood, a new listwise approach, called ListMLE, was derived. Similarly, based on the cosine similarity between the predicted results and the ground truths, another listwise approach, which is called RankCosine, was proposed [QZT+08].

2.7 SUMMARY

Online reviews, as one important user generated contents, are extensively studied by many researchers. But most of them come from the computer science area. Their research focus of online reviews is mainly on several tasks of opinion mining, such as sentiment identification, sentiment extraction, and opinion retrieval. They neglect how to make the application of their findings to product design straightforward.

From the perspective of product designers, online reviews provide valuable information about customer preferences and customer needs. Customer needs are expected to be extracted automatically. In the design area, although, some efforts contribute to the identification of customer needs, such as, customer survey data. These data are fundamentally different from online reviews. Hence, the most of state-of-the-art techniques in design area are not applicable to online reviews.

Four objectives of this research are to build a collection of helpful online reviews, to recommend rating values on online reviews by taking designers' personal assessments into consideration, to connect customer reviews with product engineering characteristics, and to prioritize product engineering characteristics based on online customer reviews for improving the current product model. To distinguish this research with existing efforts, a large number of important publications in four relevant aspects have been reviewed in this chapter. A great advancement in different aspects has been reported. However, there is a visual research gap between the area of computer science area and the design area. It is that, although the value of online reviews is widely accepted for both consumers and product designers, they are not perceived, evaluated, analyzed and utilized in the perspective of product designers.

In this research, an intelligent system will be designed to mine useful information from online reviews and help designers to improve their products. It is a crucial task, especially for the customer analysis in market-driven design.

In the next chapter, the architecture of the intelligent system, to analyze online reviews for product design, will be described. Moreover, in order to understand how online reviews can be involved in product design, two exploratory case studies will be conducted.

CHAPTER 3

A FRAMEWORK OF MINING ONLINE REVIEWS FOR QFD

3.1 INTRODUCTION

In Chapter 2, several relevant yet different researches were reviewed. It is found that, although the value of online reviews is widely accepted, few efforts contribute to mining information from online reviews in the viewpoint of product designers.

In this chapter, the framework of an intelligent system is proposed. This intelligent system will be utilized to obtain valuable information about customer needs from online reviews from the perspective of product designers. Also, this intelligent system will facilitate designers to connect online reviews with product engineering characteristics and suggest how to prioritize product engineering characteristics from online reviews.

Moreover, in this chapter, for the sake of better understanding how online reviews can be involved in product design, two exploratory case studies are conducted.

3.2 SYSTEM FRAMEWORK

The framework of the proposed system is illustrated in Figure 3.1, which consists of three closely related parts.

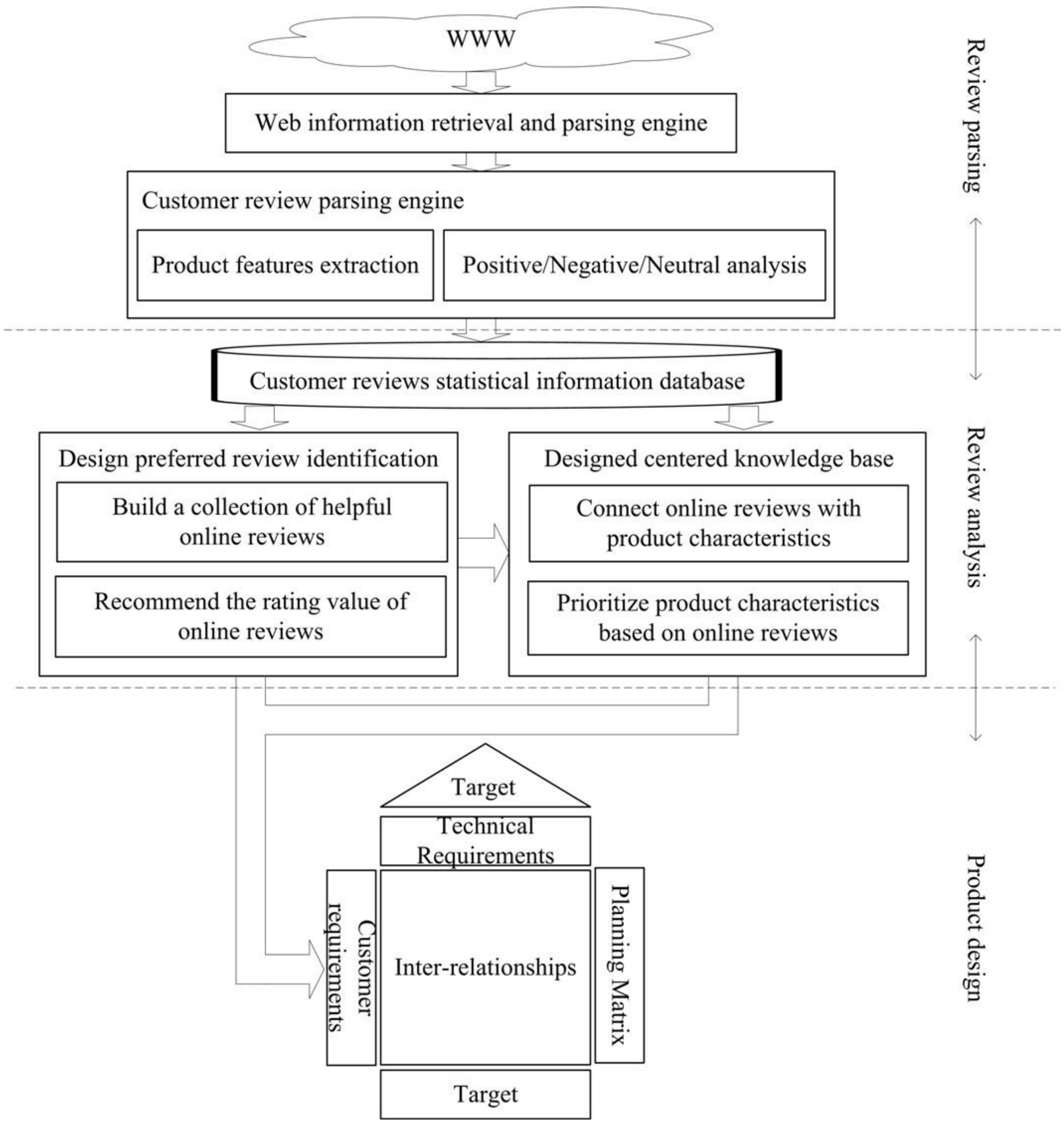


Figure 3.1 The framework of the proposed system

The first part is to gather a large number of online customer reviews and extract statistical information from parsing these reviews, such as product features and the corresponding sentiments of consumers.

In this part, at first, online customer reviews are collected by an information retrieval and parsing engine. These online reviews are utilized as the source to analyze customer needs by this system. State-of-the-art technologies in data mining, machine learning and natural language processing are then applied on these online reviews to extract some statistical information. For example, some methods of natural language processing are used to obtain the POS tag for each word in online reviews. These POS tags are helpful for the analysis of online reviews. Also, some opinion mining algorithms, which are introduced in Chapter 2, can be utilized to extract product features and analyze the corresponding sentiments implied in online reviews.

Online reviews, as well as the statistical information which is extracted in the first part, are stored in a database. With all these data, online reviews will be analyzed for product designers, which is the major task in the second part of the proposed system. Two problems are concerned, that is, Problem One: how to identify design-preferred online reviews from the perspective of product designers, and Problem Two: how to build a design-centered knowledge base from online reviews. As a matter of fact, these two problems are the focus of this research.

In Problem One, the identification of design-preferred reviews will be explained. Two research questions are included: (1) how to build a collection of helpful online reviews from the perspective of product designers, and (2) how to recommend rating values on online reviews by taking designers' personal assessments into consideration.

The first question in Problem One is to help product designers to find helpful online reviews from a large number of user-inputs, rather than taking all reviews as the input for customer analysis. The second question in Problem One is to facilitate a specific product designer to focus on those reviews which are relevant to his or her own design requirements. These two research questions are different. The former stresses how to filter out online reviews in a generic viewpoint for product design, while the latter targets at taking designers' personal assessment into considerations.

In this research, the two questions in Problem One will be analyzed and some innovated models will be elaborated for these two questions. With these models, design-preferred reviews are obtained from a large number of online customer reviews. These design-preferred online reviews still need to be further analyzed in order to facilitate designers directly. Hence, the next goal is how to translate this valuable customer reviews into product design.

For Problem Two, the objective is to build a design-centered knowledge base from online reviews. Two research questions are: (1) how to connect online reviews with product engineering characteristics, and (2) how to prioritize product engineering characteristics based on online reviews to improve current product model.

The first question in Problem Two points to the association between customer needs and product engineering characteristics. The second question in Problem Two is to generate the weights of product engineering characteristics. As explained in Chapter 1, both questions are important in QFD, and they are highly correlated. The output of the first question is to suggest how online reviews can be connected to product engineering characteristics. This connection will be then utilized as the input of the second question. Together with the customer satisfaction information which is reflected in online reviews, the weights of product engineering characteristics will be suggested in the second question.

Taking the findings of both Problem One and Problem Two as the input, the next process is to combine these results seamlessly with QFD for product design. This contributes to the future work of this research.

The emphasis in this research is how to make online reviews to be utilized from the perspective of product designers. For a better understanding of designers, two exploratory case studies of review analysis in the customer-driven product design paradigm are conducted.

3.3 AN EXPLORATORY CASE STUDY ON REVIEW HELPFULNESS

In order to better understand how the review helpfulness, as a concept, is perceived by designers, an exploratory case study is conducted using a number of real-world online reviews. As shown in Chapter 2, in the current research efforts, the helpfulness is either defined as the percentage of online helpful votes or evaluated by a pre-defined guideline. There is a visible gap that the review helpfulness is not perceived and defined from designers' point of view. But it is these designers who digest the reviews, understand customer needs and bring them into the design of new models. How online reviews are evaluated from the viewpoint of product designers has not been explored in the existing researches. For this reason, it enlightens the motivation to launch this exploratory case study, which has been never investigated in other existing research work.

In Figure 1.1, 204 out of 209 people found that review helpful. It is arguable to define the helpfulness of product reviews as the percentage of online helpful votes, which illustrates 204/209 as the helpfulness of this review. The purpose of this case study is not to verify whether the ratings given by designers are right or wrong, but to examine whether the helpfulness evaluations from product designers demonstrates a strong correlation with the percentage of online helpful votes. The objective is to figure out how online reviews are deemed as helpful from designers' evaluations. The helpfulness evaluation of online reviews was carried out by six full-time final-year undergraduates in product engineering who are familiar with the review topics, brands and models.

1,000 reviews of mobile phones of eight different brands from Amazon.com were randomly chosen. In these 1,000 reviews, on the average, there are 300.4 words and 16.5 sentences. However, they do not distribute evenly. Although the maximal number of words in a single review can reach 3,553, there also exist a large proportion of short reviews, such as "I absolutely love this phone. There is nothing else to say. I love it." Meanwhile, in terms of the number of sentences contained in each review, it also distributes unevenly with the highest number of 222.

In this case study, each designer had to read all these 1,000 reviews. Different from other researches, six designers were not instructed about how to conduct the helpfulness evaluation. No annotation guideline was given. The only evaluation concern is whether a review is helpful or not for the improvement of a product. A five-degree helpfulness evaluation was conducted using labels of "–2", "–1", "0", "1" and "2". "–2" means the "least helpful" and "2" means the "most helpful". Each review should be assigned with the most appropriate helpfulness label.

Finally, some simple policies were explained to the designers for the coding consistency purposes. They were required not to discuss with each other, and the helpfulness evaluation should be judged from one's own perspective according to their own knowledge, training and exposure in design engineering.

3.3.1 Evaluation Metrics

In this research, the average ratings over six designers are regarded as the ground truth of the helpfulness of the 1,000 online reviews. In terms of the evaluation, the focus is on the comparison between two random variables, i.e., the distance and the correlation between the percentage of online helpful votes and the average ratings.

The mean absolute error (denoted by *MAE*) and the root mean squared error (denoted by *RMSE*) are adopted to evaluate the distance. They are both frequently used metrics to measure the differences between values presented by a model and the values actually observed from the object being modeled. Also, the Pearson product moment correlation coefficient (denoted by *PMCC*) is utilized to evaluate the correlation between the percentage of online helpful votes and the actual values from designers.

- Mean absolute error

In statistics, *MAE* is a quantity used to measure how close predictions are to the eventual outcomes. *MAE* is an average of the absolute errors and is given by

$$MAE(real, predict) = \frac{1}{n}\sum_{i=1}^{n} | real_i - predict_i | \qquad (3.3.1)$$

$real_i$ is the true value for the sample i and $predict_i$ is the prediction value.

- Root mean squared error

In statistics, *RMSE* is also a frequently used measure of the differences between values presented by a model and the values actually observed from the thing being modeled. *RMSE* of an estimator *predict* with respect to the estimated parameter *real* is defined as the square root of the mean squared error. The formula is:

$$RMSE(real, predict) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(real_i - predict_i)^2} \tag{3.3.2}$$

- Pearson product moment correlation coefficient

In statistics, *PMCC* is a measure of the correlation (linear dependence) between two variables, giving a value between one and minus one inclusive. It is widely used as a measure of the strength of linear dependence between two variables.

$$\begin{aligned} &PMCC(real, predict) \\ &= \frac{\sum_{i=1}^{n} real_i \cdot predict_i - n \cdot \overline{real} \cdot \overline{predict}}{n \cdot s_{real} \cdot s_{predict}} \\ &= \frac{n \cdot \sum_{i=1}^{n} real_i \cdot predict_i - \sum_{i=1}^{n} real_i \cdot \sum_{i=1}^{n} predict_i}{\sqrt{n \cdot \sum_{i=1}^{n} real_i^2 - (\sum_{i=1}^{n} real_i)^2} \cdot \sqrt{n \cdot \sum_{i=1}^{n} predict_i^2 - (\sum_{i=1}^{n} predict_i)^2}} \end{aligned} \tag{3.3.3}$$

$\overline{predict}$, $\overline{real}$, $s_{predict}$ and s_{real} are predicted sample mean, real value mean, predicted sample standard deviation, and real value standard deviation, respectively.

3.3.2 Preliminary Results

- Number of Reviews vs. Number of Interest Voting

Among the 1,000 reviews, only 405 reviews received more than five votes. In Figure 3.2, the statistical results are provided by showing how many reviews receive votes for none, once, etc. The fact that only a small portion of reviews eventually are voted by sufficient users is confirmed. Hence, the percentage of online helpful votes which is directly utilized as the evaluation criterion for the helpfulness of reviews is fundamentally questionable.

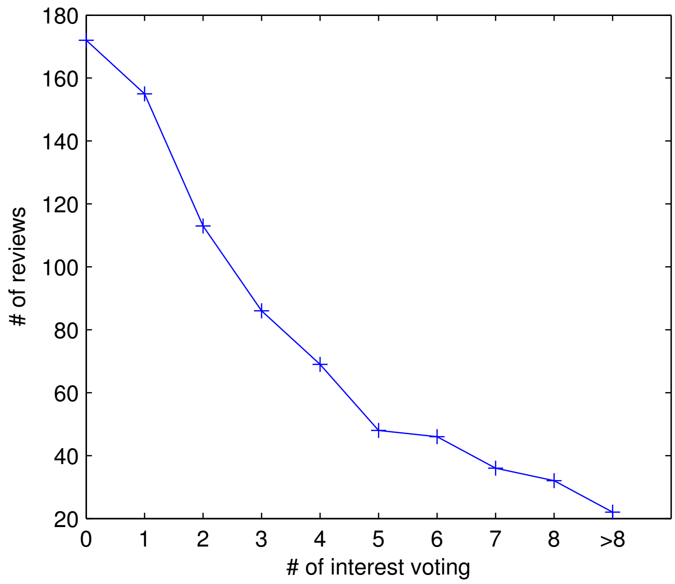


Figure 3.2 Helpfulness rating profile from six designers

- Percentage of Online Helpful Votes vs. Average Helpfulness Rating

The percentage of online helpful votes ranges from zero to one, but the helpfulness evaluated by designers is from minus two to positive two. Hence, the percentage of online helpful votes is scaled to the same region as the designer's rating in this evaluation in terms of three metrics. Table 3.1 shows the results of values over the 1,000 reviews.

Table 3.1 Percentage of online helpful votes vs. average designer rating

MAE	*RMSE*	*PMCC*
1.178	1.411	0.410

As seen from Table 3.1, the average designer rating presents a weak median correlation with the percentage of online helpful votes. The values of *MAE* and *RMSE* account for about 29.5% ($\frac{1.178}{2-(-2)}$) and 35.3% ($\frac{1.411}{2-(-2)}$) of the scale region, respectively. The previous assumption that designers' helpfulness rating might not present a strong correlation with the percentage of online helpful votes is confirmed by this phenomenon. There might be an unacceptable error between them.

- Designer's Helpfulness Rating

The percentage of online helpful votes is compared with all six designers' ratings. Likewise, the percentage of online helpful votes is also scaled to the region from minus two to positive two and they are also evaluated in terms of three metrics. The results from the 1,000 reviews are depicted in Figure 3.3.

The above argument that the percentage of online helpful votes does not necessarily behave in the same way as designer ratings is seen in Figure 3.3. Even for the smallest *MAE* and *RMSE* (designer 1), the two evaluation metrics still account for about 28.0% ($\frac{1.12}{2-(-2)}$) and 37.3% ($\frac{1.49}{2-(-2)}$) of the scale region respectively. For the *PMCC*, five of the six designers only present slightly higher than 0.25, which shows a weak correlation with the percentage of online helpful votes.

It is also found that the six designers presented different assessments on the helpfulness of reviews. One designer tended to display a strong positive attitude (the average helpfulness evaluation reached 1.03) towards the helpfulness rating on all the 1,000 reviews. Another two designers tended to give a neutral helpfulness rating (the average helpfulness evaluation are –0.011 and –0.033, respectively.)

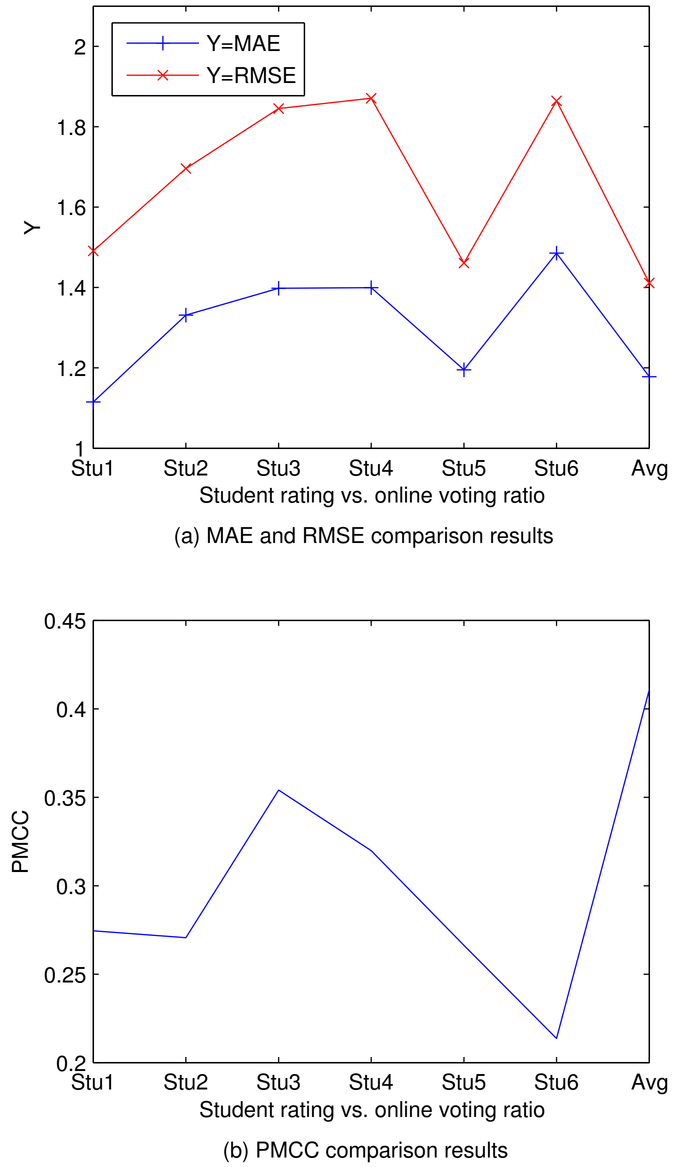


Figure 3.3 Designers' helpfulness ratings vs. percentage of online helpful votes

Another interesting observation is that the evaluation of the helpfulness varied among six designers' evaluation. In the 1,000 reviews, only 12 reviews, accounting for slightly higher than 1% of the dataset, received unanimous helpfulness labels from the six designers. On the other hand, many reviews received a large standard deviation of the helpfulness evaluation. For example, one review was labeled with two “2”s, two “–2”s, one “1” and one “–1”. The six annotators presented four different helpfulness evaluations. Another example is that one review was labeled with five “2”s. This same review was also labeled as “–2” by one designer. As seen from the result, there exists a large variance for some reviews.

All these interesting results reveal that different product designers may have different criteria in the evaluation of review helpfulness. The result triggers this research to investigate the reason why a particular review is helpful from the perspective of one designer, but not so helpful or not helpful for another designer.

3.3.3 Why a Review is Helpful

Naturally, the first objective is to investigate why some reviews receive the unanimous helpfulness evaluation. Especially, without giving instructions on the helpfulness evaluation, why did the designers assign the same polarity helpfulness label (i.e., assigning six “–2” labels or six “2” labels to some reviews)? A questionnaire for these six designers was initiated in this case study in order to understand the reasons behind this. According to the questionnaire, some reasons for reviews receiving all “2” labels include “a long review covers one’s preferences”, “mentions many different features”, “points out the like and dislike of the product”, “compares his E71 to Blackberrys”, etc. While the reasons for those receiving all “–2” labels include “did not mention anything good or bad about features”, “no information about the performance”, etc.

Second, it is also curious about what has happened to the reviews receiving a large divergence in labeling. So another questionnaire was made to investigate the reasons and perspectives of the designers. Since, in the case, each designer's labeling is different, six different questionnaires were distributed individually. Take the previous review receiving two "2"s, two "–2"s, one "1" and one "–1" for example. The two designers, giving the "most helpful" with label "2", commented that this review "reported that the phone didn't work properly since the front mask was a little open". This review also mentioned an "SD card detection problem (requires either OS or Hardware check)." Two designers regarded it helpful because several problems were pointed out and some potential reasons were provided. Other two designers, giving the "least helpful" with label "–2", indicated that this review "only focuses on the problem of the memory card" and "the problem mentioned should not be related to the phone." Another interesting example is one review with five "2"s and one "–2". The designer giving "–2" explained that "customer mainly talks about the usefulness of the phone, the third party applications and not much about the bad sides of the phone." These answers about why online reviews were rated as so facilitate this research to understand the perspective of product designers.

3.3.4 Discussions

The objective of this exploratory case study is to understand how the review helpfulness, as a concept, is perceived by designers. Several important judgment reasons regarding why some reviews are helpful are given by six designers in the follow-up answers of two questionnaires. It facilitates this research to explore the identification of design-preferred online reviews from the perspective of product designers.

Various efforts have been made towards defining and evaluating the helpfulness of online reviews, either by the percentage of online helpful votes or by a pre-defined evaluation guideline. In this case study, a weak median correlation between the average designer rating and the percentage of online helpful votes was presented. It might be unacceptable if the percentage of online helpful votes or a pre-defined evaluation guideline is utilized directly to predict the helpfulness of online reviews from the perspective of product designers. How product designers perceive, define and evaluate the helpfulness of online reviews is neglected.

The first interesting question under this concern is how to obtain helpful reviews from vast amount of online information about customer needs in the viewpoint of product designers. Also, some reasons towards what make online reviews helpful from the perspective of product designers are provided in the two follow-up questionnaires. These reasons contribute to discover helpful online reviews. In addition, some reasons are found not to be confined explicitly in a particular domain area. Especially, several reasons are domain-independent, such as, “a long review covers one’s preferences”, “mentions many different features”, etc. Hence, whether the helpfulness of reviews, evaluated by product designers in one domain area, is able to be transferred to other domains should be examined.

The second interesting question under this concern is how to recommend rating values on online reviews by taking personal assessments into consideration. For example, in the second questionnaire, several reasons illustrate that product designers have their own criteria of judgments on why some reviews are helpful. The rationale behind this may be that different product designers have different requirements. In order to satisfy the requirements of different product designers, a method is expected for recommending rating values on online reviews by taking designers' personal assessments into consideration. Some profiles about both online reviews and designers need to be investigated since they may facilitate to build this recommendation method.

In Chapter 4, the technical details for these two questions, how to build a collection of helpful online reviews and how to recommend rating values on online reviews by taking designers' personal assessments into consideration, will be described.

3.4 AN EXPLORATORY CASE STUDY ON REVIEW ANALYSIS

The objective of this case study is to understand how online reviews can be utilized by product designers in the analysis of customer needs. As shown in Chapter 2, in the current research efforts, customer survey data are utilized for the analysis of customer needs. But online reviews are also widely accepted to contain valuable consumers' information. However, they are not currently being evaluated and utilized in the analysis of customer needs. Conventional survey data mainly come from customer investigation, in which targeted questions for consumers are given. Consumers provide responses under the direction of with these questions. The responses are usually selected from a given answer list, although there exist some open questions with obvious intention. However, online reviews are different with these survey data. Online reviews are free texts. They are written by consumers, not directed by any questions, and generated from time to time. Online reviews are generally regarded as one valuable research source to analyze customers' sentiments. However, how online reviews can be utilized in product design directly, has not been fully explored, for example, how to connect customer reviews with product engineering characteristics and how to prioritize product engineering characteristics based on online customer reviews for improving the current product model. But the association between customer needs and product engineering characteristics as well as the weightings of product engineering characteristics are an important research area in QFD. For this reason, it enlightens the motivation to launch this exploratory case study.

In this case study, two annotators were hired to evaluate online reviews. Generally, it is almost impossible to gain the review evaluation from experienced product designers because of some confidentiality regulations in business. The review evaluation was carried out by two customer service clerks who acted as product designers. These customer service clerks were working in Epson Hong Kong and HP Hong Kong. They are very familiar with printers and they had a sound understanding about customer complaints and concerns. Also, they both had some experience with printer design using QFD, which contributed to build a high quality dataset.

In this case study, 770 reviews of four popular color printers (two Epson printers and two HP printers) were randomly selected as examples from Amazon.com and Epson.com. They are "Artisan810", "WorkForce610", "HP 6500", and “HP C309”. For short, “A810”, “W610”, “H6500”, and “C309” are used in the following Table 3.2, which shows the number of reviews for each model.

Table 3.2 Number of reviews

Printer	A810	W610	H6500	C309
Number of reviews	258	169	210	133

At first, a list of product engineering characteristics was collectively suggested by the two annotators, which is illustrated in Table 3.3.

Table 3.3 Product engineering characteristics

Printer Housing	Power Supply	Fax Setting	Brand
Wifi Integration	Ease of Setup	Ease of Use	Noise
Duplex Printing	Print Quality	Print Head	Package
Software Updated	Scan Software	LCD Panel	Outlooks
Auto Document Feeder	Printing Speed	Hopper Unit	Card Slot
Supplementary Software	Mac Compatible	Ink Longevity	Durability

The annotators were then asked to label the reviews. Each annotator began to read all reviews and distinguish the keywords in each review sentence. Here “keyword” is either a word or a phrase referring to product engineering characteristics. Sometimes, only one word is utilized to refer to a product characteristic, while, in some cases, a phrase is employed. If a sentence contains these keywords, they connected the keywords with the product engineering characteristics.

The linkage value is the customer sentiments on product engineering characteristics. It is denoted by “–2”, “–1”, “0”, “1” and “2”. “–2” means the least satisfied and “2” means the most satisfied. In Figure 3.4, an example of one Epson Artisan810 review labeling is shown.

Sentence	keyword	Auto Document Feeder	Card Slot	Consumable Replacement	Durability	Ease of setup	Ease of use	Fax Setting	Hopper Unit	Mac Compatible	Ink Longevity	LCD Panel	Noise	Power Supply	Print Quality	Printer Housing
I only have the printer for a few days, but so far I am very pleased.	-															
I was a bit nervous before buying because I had read a lot of complai	-															
But the paper handling on my 810 has been flawless, if a tiny bit noi	noisy												-2			
But the paper handling on my 810 has been flawless, if a tiny bit noi	pull								-1							
The actual printing is quiet, and of great quality.	quiet												1			
The actual printing is quiet, and of great quality.	quality														2	
The paper tray feels a bit flimsy, but is easy to remove or insert, a	tray								-2							
It can expand to hold legal size paper, and has a separate area for s	paper															
The package comes with 3 sheets of Epson high quality 4x6 glossy phot	print quality														2	
I have been using an HP Wireless printer up until now(model 5850)and	quality														2	
Also, they promise that their inks are acid-free and will last 3 or 4	ink										0					
Another pre-purchase worry was that there were complaints about the A	-															
Too early to tell, but I 've printed those 3 photos and a fourth on p	ink										1					
So, so far so good on ink usage.	ink										0					
I was surprised and pleased too that the 810 printer comes with an TW	ink			1												
Nice!	-															
Additionally, the inks provided are the same capacity as the refills,	ink			2												
A nice feature is that when you plug in a camera memory card in the f	card		2													
Very neat.	-															
It warns on the box that this feature may not work with Mac computers	mac									1						
Have n't tried printing from the iPhone as they claim you can do, tho	iphone															
I love the big touch screen operation too, it is pretty slick.	touch screen											2				
Made a few copies of some simple pencil drawings and they were very g	copy														-1	
Updated Review 03\/2010Here 's a 6 month update.	update															
The printer is getting a lot of work ; we print a lot of classroom ma	-				2											

Figure 3.4 Review labeling

In this example, the first sentence, “I only have the printer for a few days, but so far I am very pleased”, does not contain any keywords connected with product engineering characteristics, so the keyword in this line is labeled as “-”. The seventh line in Figure 3.4 is “the paper tray feels a bit flimsy, but is easy to remove or insert, and there’s no fuss to loading your paper in it.” This consumer actually complained about the “Hopper Unit” through the phrase “paper tray”, so the annotators wrote this phrase in the keyword column of the seventh line.

If more than one product engineering characteristics are mentioned in a sentence, the annotators copied this sentence, pasted it into the other line and labeled the second item in a new line. For example, in Figure 3.4, the fourth sentence is “the actual printing is quiet, and of great quality.” Two product engineering characteristics are mentioned with the word “quiet” connected with “Noise” and the phrase “great quality” connected with “Print Quality”. This sentence is repeated in order to unambiguously label the two keywords. A similar case is observed in the third and fourth line of Figure 3.4. The sentence is “but the paper handling on my 810 has been flawless, if a tiny bit...”

Finally, reviews were double checked by the annotators in order to avoid any mislabeling in such time-consuming and labor-extensive procedure.

3.4.1 Preliminary Results

- Words connected with different product engineering characteristics

Having carefully checked, a controversial relationship between keywords in online reviews and product engineering characteristics is found from the four datasets. Exactly, some particular keywords were labeled with one product characteristic only. But some keywords were also labeled with different product engineering characteristics.

Take the word “paper” as an example. One is “...It obviously needs a more absorbent paper because...” and the other is “...Very easy to swap in alternate papers, always easy to see if paper is left...” In the first sentence, the word “paper” is utilized to refer to “Supported Paper”, while the other sentence is referring to “Hopper Unit”.

The number of words which are connected with different product engineering characteristics is shown in Table 3.4. As seen from Table 3.4, there exist many keywords which are connected with different product engineering characteristics.

Table 3.4 Number of words connected with different product engineering characteristics

Printer	A810	W610	H650	C309
Number of words	33	29	24	36

The same keywords connected to different product engineering characteristics, which imply that a consumer topic in one review sentence can not be described by a single keyword. By intuition, the words which are near the keywords, or say, context words, might be utilized to describe consumer topics. These interesting phenomena trigger this research to explore how to connect online reviews with product engineering characteristics automatically.

- The frequency of product engineering characteristics

Once online reviews are connected with product engineering characteristics, an important task is to estimate the weights of different product engineering characteristics for new product design.

The prioritization of product engineering characteristics might be regarded as highly related with the frequency that they are mentioned in online reviews. In other words, those product engineering characteristics, which are frequently talked about by consumers, might be suggested to give higher weights. In Table 3.5, the top five frequently-mentioned product engineering characteristics are listed.

Table 3.5 Top five frequent product engineering characteristics

A810		W610	
Feature	Frequency	Feature	Frequency
Print Quality	0.727	Print Quality	0.488
Ease of Setup	0.495	Wifi Integration	0.377
Scan Software	0.485	Ease of Setup	0.364
Wifi Integration	0.475	Printing Speed	0.364
Hopper Unit	0.364	Noise	0.284
H6500		C309	
Ease of Setup	0.636	Print Quality	0.698
Wifi Integration	0.568	Ease of Setup	0.586
Print Quality	0.426	Wifi Integration	0.483
Scan Software	0.278	Printing Speed	0.302
Noise	0.272	LCD Panel	0.293

As seen from this table, more than 40% consumers prefer talking about “Print Quality”, “Ease of Setup” and “Wifi Integration”. It is easy to understand. For a printer, the print quality and the usability may always be the first concern for consumers. However, whether they should be assigned with a higher priority is unknown. This hypothesis will be examined in Chapter 5.

3.4.2 Discussions

The objective of this case study is to understand how online reviews are utilized in the customer requirement analysis from the perspective of product designers. Some interesting phenomena are observed and valuable manual labeled data are obtained in this case study.

As seen from this case study, two tasks are included in requirement analysis. At first, through reading online reviews, product designers connect online reviews with product engineering characteristics. After digesting all this information, they need to suggest how to prioritize product engineering characteristics for new product design. But, currently, the two tasks are conducted manually. It is time-consuming and labor-extensive. It triggers this research to explore the concern about how to build a design-centered knowledge base from online reviews.

The first interesting question under this concern is how to connect online reviews with product engineering characteristics automatically. Many keywords in online reviews are found to be connected with different product engineering characteristics. One single keyword alone is not able to judge the associated product engineering characteristics correctly. Knowledge mapping is expected to be learned from online reviews to indicate the relationship between customer needs and product engineering characteristics. A linguistic approach is targeted to connect online reviews with product engineering characteristics automatically and help designers to avoid the sentence by sentence analysis on online reviews.

The second interesting question under this concern is how to prioritize product engineering characteristics from online reviews for market-driven product design. How to weight product engineering characteristics is one important research area in QFD. Existing efforts contributing to this research area only care about customer survey data, while valuable customer information in online reviews is neglected. It is expected to develop a method to prioritize product engineering characteristics from online reviews. In a customer review for a specific product, both customer sentiments about product engineering characteristics and overall customer satisfaction or dissatisfaction may be conveyed. The above customer information can be utilized to build this prioritization method.

3.5 SUMMARY

In this chapter, the framework of the proposed system was explained. The focus of the proposed system is on identifying design-preferred online reviews from the perspective of product designers and building a design-centered knowledge base from online reviews.

The two questions in Problem One are how to build a collection of helpful online reviews and how to recommend rating values on online reviews by taking personal assessments into consideration. In order to understand how the review helpfulness is perceived by designers, an exploratory case study was conducted using a number of real-world online reviews.

The two questions in Problem Two are how to connect customer reviews with product engineering characteristics and how to prioritize product engineering characteristics from online customer reviews for improving the current product models. In order to understand how online reviews can be utilized by product designers in the analysis of customer needs, another exploratory case study was conducted.

Valuable data and interesting observations were obtained from the two exploratory case studies. All of these serve as the basis of this research and facilitate this research to build sound models and algorithms for mining valuable information from online reviews for product design. The next task is to find ways to solve these four questions explicitly.

In the next chapter, several novel techniques will be proposed to model Problem One, identifying design-preferred online reviews from the perspective of product designers.

CHAPTER 4

IDENTIFYING DESIGN-CENTERED ONLINE REVIEWS

4.1 INTRODUCTION

In Chapter 3, the framework of the system was explained. Two problems are included: how to identify design-preferred online reviews and how to build a design-centered knowledge base from online reviews. In addition, in order to understand how online reviews can be involved in product design, two exploratory case studies were conducted. Valuable data and instructive responses from designers were obtained from the two exploratory case studies. They contribute to develop sound models and algorithms.

In this chapter, the focus is on Problem One, how to identify design-preferred online reviews. As mentioned, two questions in Problem One are how to build a collection of helpful online reviews from the perspective of product designers and how to recommend rating values on online reviews by taking designers' personal assessments into consideration. In the first exploratory case study, the helpfulness evaluations from product designers do not demonstrate a strong correlation with the percentage of online helpful votes. In the follow-up two questionnaires of the first exploratory case study, several important judgment criteria regarding why some reviews are helpful are given by product designers. These criteria facilitate this research to model these two questions. Several technical details will be described in this chapter.

4.2 HELPFULNESS PREDICTION OF ONLINE REVIEWS

4.2.1 Overview of the Helpfulness Prediction Approach

An overview of the helpfulness prediction approach is shown in Figure 4.1. As seen from this figure, two phases are included.

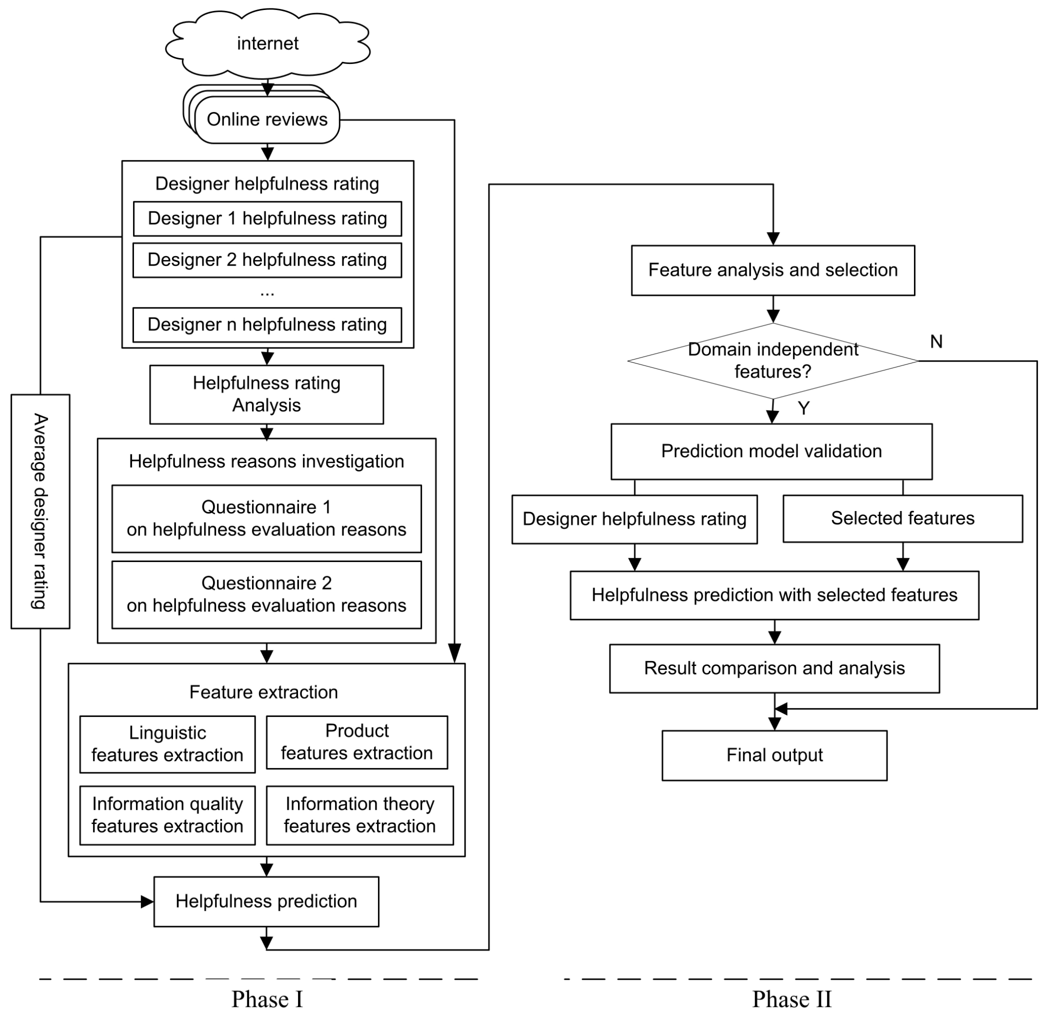


Figure 4.1 The approach overview

In Phase I, the average helpfulness rating from designers is assumed as the golden criterion, and the objective is to train a model for predicting the helpfulness of online reviews. Based on the understanding of how designers evaluate helpfulness, four categories of intrinsic features (i.e., linguistic features, product features, features extracted based on information quality, and features extracted using information theory) are proposed to support the modeling of helpfulness. The identification and extraction of these features are independent from designer ratings and other forms of external knowledge. The four categories of features are extracted entirely from the review content, which explain why they are called intrinsic features. Utilizing these features as input and the average helpfulness rating from designers as output, a regression is built to search an appropriate method for the learning of a possible underlying model. This concludes the research in Phase I.

These features are obtained entirely from the review content only. Whether the domain features (one of the four categories of features) possess a strong correlation with the helpfulness evaluation, will be explored in this research. In Phase II, feature selection and feature analysis is conducted to evaluate whether the accuracy of helpfulness prediction will be influenced significantly by the categories of domain-dependent features. The regression model learned from a specific product in Phase I is utilized to predict the helpfulness of unrated reviews, with or without the domain features, on different products in other domains. The question is whether a strong correlation is possessed by the domain features with the helpfulness being perceived. In other words, without domain features, if a significant loss will suffer in terms of the correlation between the predicted helpfulness and the designer ratings, then the modeling of helpfulness cannot be confidently migrated to another domain. These serve as the essential research content in Phase II.

Therefore, in Phase II, the focus is on a new and challenging question on helpfulness migration. With a limited number of manually rated reviews in one domain, the question is whether it is possible to model the helpfulness from the review texts alone, or whether such a model is generic enough to be migrated to other domain where the manually rated reviews may not be available. That is actually the primary reason why this approach is divided into two phases.

4.2.2 Why Four Categories of Features are Chosen

According to the exploratory case study, designers express their concerns about the helpfulness of online reviews. Some designers expect that more useful information can be learnt from long online reviews, primarily indicated by its numbers of words and sentences. Also, some customers share their comments by telling others their preferences or complaints on this particular product. It is valuable for product designers to know the reasons behind such sentiments, which are mainly expressed by adjectives or adverbs. Meanwhile, the six designers who were invited to evaluate the helpfulness of online reviews in the exploratory case study also complained that they might lose their interests to read and attempt to understand online reviews if there are many grammar errors (number of grammar errors), wrong spellings, and if there are many exceptionally long sentences (average number of words per sentence). All these interesting phenomena serve as the reasons to propose linguistic features as one category of features to model the helpfulness of online reviews.

In the meantime, some of the six product designers are also found to focus on whether key product features have been mentioned. For example, a few designers have given their lists of product features. Each list provides the most important product features that customers have talked about, and such product features are considered as crucial information carriers when designers conceive a new model. The conjecture is that the appearance of some particular product features might influence the helpfulness evaluation. Thus product features are proposed in this research as one primary category of features in the model building.

According to the two questionnaires, some designers replied that “this review mentions many product features”, while some argue that “many reviews shared the features he/she likes and dislikes”. Aggregating all these arguments, as a matter of fact, in a higher and abstract level, designers refer to information quality (IQ) in different aspects. For example, the first argument actually mentioned the information coverage and the second points to the information accuracy. All these have inspired this research to explore the possibility of extracting features from various perspectives of information quality.

Quite a few designers, in the exploratory case study, gave their most unhelpful labels to reviews with “no (concrete) information about user experience of the product concerned.” It has come to the attention that information can be presented differently from the designer perspectives in the reviews. For example, designers’ understanding actually will be greatly influenced by the sentiment of a product feature that deviates from the majority of sentiments provided in reviews, since more details are often provided with the reasons why a different or an opposite sentiment is given. Also, a review tends to be regarded as a helpful one if both pros and cons of a product have been mentioned. In the investigation, the two designers explicitly stressed that the appearance of “both pros and cons” in different aspects of products is an important factor for helpfulness evaluation. This phenomenon is often referred as the divergence of sentiments and so it appears in the modeling. In addition, a review stands for a higher chance of being rated as helpful if it has expressed a strong and sharp viewpoint towards certain product features with persuasive reasons. It sounds slightly similar to the first observation except that the sentiment expressed may or may not align with other consumers. Hence, such observations are proposed to be interpreted using information theory.

4.2.3 Modeling of Helpfulness

In this research, one objective is to explore whether the concept of helpfulness of online reviews, being perceived from a product designer’s perspective, can be modeled using features that are completely extracted from the texts of online reviews. These features are identified without the assistance of any domain knowledge, such as product structure, product ontology, knowledge rules, etc. These intrinsic features are entirely derived from the content of online reviews and serve mainly for the purpose of helpfulness modeling. In the following subsections, how the four categories of features are defined, based on the understanding from the exploratory case study, is explained.

4.2.3.1 Linguistic Features Extraction

According to the statements in Section 4.2.2, linguistic features are employed to model the helpfulness of online reviews. Table 4.1 gives the details of linguistic features.

Table 4.1 Linguistic features

Feature Alias	Description
L-NW	# of words
L-NS	# of sentences
L- ANWS	avg. # of words per sentence
L- NADJ	# of adjectives
L- NADV	# of adverbs

Several NLP techniques, for example, POS tagging, are employed in this step. POS tagging is the process of marking up the words in a text as corresponding to a particular POS (Part of Speech) such as noun, verb, pronoun, preposition, adverb, adjective or other lexical class markers to each word, based on its definition as well as its context, i.e., the relationship with adjacent and related words in a phrase, sentence or paragraph. Also, *LanguageTool*, an open source language and grammar checker for English, is utilized to check the number of grammar errors.

4.2.3.2 Product Features Extraction

POS tagging was also widely utilized in the extraction of product features. Once POS tagging is performed, linguistic rules are used to generate feature candidates. For example, "<*N*> [feature] usage" is a rule, which matches the segment "battery usage", to discover feature candidates. "*N*" is a POS tag that can match any word with that tag, and "usage" is a concrete word which can only match this particular word. After this process, phrases and words with the least possibility of being product features are removed. Clearly, many interventions are involved, and the performance depends on the completeness of the linguistic rules.

In this approach, a product feature list is generated based on a document profile (DP) model [LLL09]. In Figure 4.2, the process flow of a DP model is illustrated.

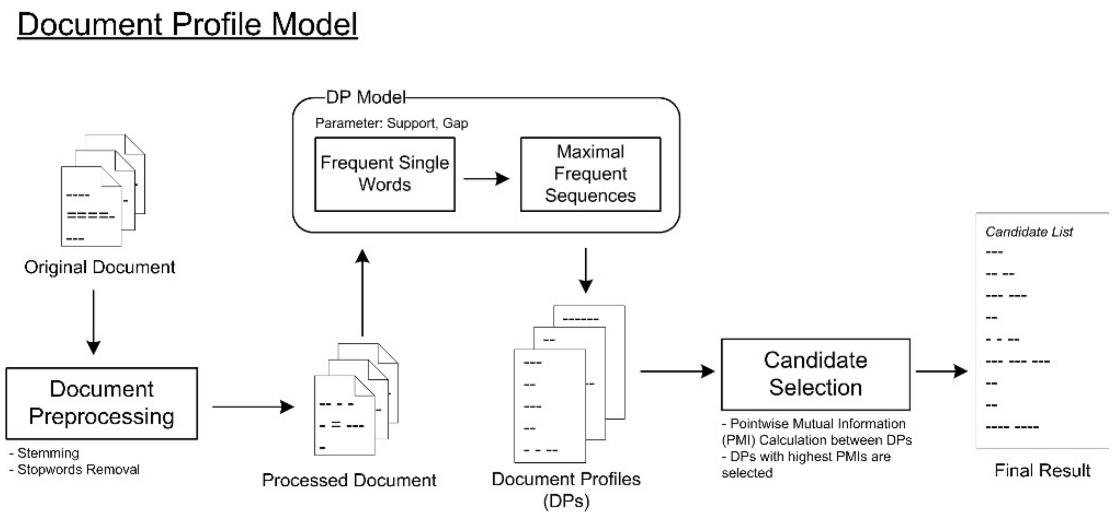


Figure 4.2 The process flow of a DP generation [LLL09]

The DP model focuses on the discovery of patterns of word frequency at the sentence level in reviews which is particularly important for short texts. Two parameters are required as its input, namely, the word gap g and support value s. The output of the DP model is a list of words and their corresponding frequencies, marked <w_i, v_i>, where w_i denotes the word and v_i denotes its occurrence value. The

DP model is concerned with how to capture single words and word sequences which often bear semantic meanings at the lexical level to represent documents. It extends the basic idea of using the pointwise mutual information to measure the strength of the semantic association based on the terms and Maximal Frequent Sequences (MFS) discovered. A simple metric called average pointwise mutual information is used to measure the average strength of the semantic association among a set of features.

After a list of words from the DP model is generated, some words with the highest frequency are obtained. Notice that product features are often expressed or depicted as nouns or noun phrases. In this research, nouns and noun phrases with the highest frequency are considered as candidates of the product features.

As expected, there still exists some noise, e.g., “phone”, “nokia”, etc., if all candidates are regarded as product features. Some pre-defined contextual words are removed with the help of domain knowledge from designers. In the DP model, words with low frequency are also listed, which are actually not product features. The words with low frequencies that might affect the result are pruned. In the experiment, this word list is pruned based on the natural frequency distribution of nouns or noun phrases in the dataset rather than cutting the word list with some random cut-off ratio. After this candidate selection process, the nouns or noun phrases left are considered as product features. The occurrence frequencies of each extracted product feature in product reviews are used as a feature for helpfulness prediction.

4.2.3.3 Features Extraction Based on Information Quality

Several researchers have attempted to identify the possible dimensions of information quality (IQ) that can be used to measure information quality. Various metrics were proposed to evaluate information quality. In this research, review quality is evaluated primarily in five aspects: information accuracy, information timeliness, information comparability, information coverage, and information relevance. The details of features extracted based on information quality are listed in Table 4.2.

Table 4.2 Features extracted using information quality

IQ Aspects	Feature Alias	Description
information accuracy	IQ-NSS	# of subjective sentences
	IQ-NOS	# of objective sentences
information timeliness	IQ-TIM	# of total elapsed days
information comparability	IQ-NRP	# of referred products
information coverage	IQ-NPF	# of product features
information relevance	IQ-NSPF	# of sentences referred product features
	IQ-RPFR	# of product features / # of sentences referred product
	IQ-RPFS	# of product features/ # of sentences
	IQ-RRS	# of sentences referred product features/# of sentences

As seen from Table 4.2, there are at least two tasks in order to extract all these features. The first task is to identify whether a sentence is subjective or objective for IQ-NSS (number of subjective sentences) and IQ-NOS (number of objective sentences). The second task is to identify whether a product name occurs in a review for IQ-NRP (number of referred products), which is needed to calculate the information quality dimensions quantitatively.

Notice that the techniques for the extraction of product features, which were explained in the previous section, is employed for calculating IQ-NPF (number of product features), IQ-NSPF (number of sentences referred product features), IQ-RPFR (number of product features / number of sentences referred product), IQ-RPFS (number of product features / number of sentences) and IQ-RRS (number of sentences referred product features / number of sentences).

- Identification of sentence sentiment orientation

Before the sentence sentiment is identified, it is necessary to know whether adjectival words, which reflect the author's sentiment, are included in a sentence since people tend to express the sentiment by adjectives (sometimes called opinion words). Although verbs can also be utilized to express their opinions (e.g., love or dislike), in this research, only adjectives are considered as opinion words.

At first, two sets of adjectives, the positive word set and the negative word set, are generated. The sentiment of an adjective indicates the direction that the word deviates from the norm. Adjectives that express a state of desirable sentiment (e.g., excellent or perfect) have a positive orientation, while those expressing undesirable states have a negative orientation (e.g., difficult or bad). Notice that the synonyms of an opinion word share a similar sentiment orientation and the antonyms have an opposite orientation of sentiment. The list of seed words is then expanded by WordNet. A word is added to the original positive list of words if it is a synonym of a positive adjective. Similarly, it is added to the original list of negative words if it is an antonym of a positive word.

To predict the sentiment of sentences, the method proposed in [HL04a] is utilized. In this method, counting the occurrences of opinion words was utilized to judge the orientation of a sentence. If the number of positive words is greater than negative words, the sentence is regarded as positive and vice versa. Also, the occurrence of the negative modifier word "not" is considered. The orientation of sentence sentiment is flipped if the word "not" appears.

- Identification of product model name

Notice that each product p_i possesses its own title $p_i(t)$. Also, each product has an exclusive model name. For example, "E71" is the unique model name for the "Nokia E71x Phone". In the observation, the unique model name is composed of a combination of digital numbers and characters with or without "-" or "/". Based on the above discussions, the algorithm of recognizing a model name consists of three steps as follows:

Step 1: Data preparation and pruning

In this step, two tasks are performed. Firstly, split the title $p_i(t)$ into a string array of individual words. Secondly, remove those words (e.g., "Nokia", "with", etc.) which may not be a product model name from the string array.

Step 2: Inverse index generating

In this step, the inverse index for each possible model name is established. The inverse index for possible model names includes: the individual word generated in Step 1 and a linked list which contains model names. As the unique nature for a model name, if the length of the linked list for this word is one, this word should be the model name.

Step 3: Candidate pruning and model name generating

In this step, the candidate model names with the least possibility of being a model name are pruned. Also, these words have to be kept if they cannot be decided whether it is a model name or not. For example, the title of a phone is “Blackberry Storm2 9550 Phone”. It is difficult to guess whether “Storm2” or “9550” is the model name, so both words have to be kept. After that, a candidate list for the model names is obtained. Given the model name, an open source project, called *Lucene*, is employed in this research. *Lucene* is a high-performance, full-featured text search engine library with features including fast indexing, ranked searching, extension APIs (Application Programming Interface), etc., to count the occurrences of the model name for each review.

4.2.3.4 Features Extraction Using Information Theory

In this subsection, information theory is used to estimate the information gained from three heuristic rules to extract features for the helpfulness prediction. Table 4.3 lists the features extracted by using information theory.

Before the value for the three aspects is calculated quantitatively, the sentiment for a product feature should be judged. This research improves the method proposed by Ding et al. [DLY08]. In this method, the sentiment for a product feature is predicted by considering the co-occurrence of a product feature and a sentiment word. Also, the sentiment of a product feature would be flipped once a negative word occurs.

Table 4.3 Features extracted using information quality

Feature Alias	Description
IT-SI	The self-information sum of product features
IT-DS	The divergence of sentiment sentences
IT-SS	The strength of sentiment sentences

To further improve this approach, a threshold for the sentiment value of a product feature is suggested. If the sentiment value is greater than a certain threshold, a positive sentiment on the product feature is considered. Similarly, a negative sentiment on the product feature is considered if sentiment value is smaller than a certain threshold. Otherwise, a neutral opinion is considered.

- The self-information of product features

Different sentiments of a product feature may be presented in different reviews. Intuitively, different information about helpfulness is provided to designers. Accordingly, for a product feature f_j, extracted as elaborated in the previous section, given a kind of sentiment (positive, negative or neutral), the probability of the product feature, $prob(f_j, sentiment)$, in a dataset is evaluated as:

$$prob(f_j, sentiment) = \frac{numOfsentence(f_i, sentiment)}{numofsentence(f_i)} \qquad (4.2.1)$$

numOfsentence(f_j, *sentiment*) denotes the total number of sentences which express a certain *sentiment* towards f_j and *numOfsentence*(f_j) denotes the total number of sentences that mention f_j. According to information theory, if a product feature f_j and the corresponding *sentiment* are given, the self-information gained, $SI(f_j, sentiment)$, is calculated as:

$$SI(f_j, sentiment) = -log(prob(f_j, sentiment)) \quad (4.2.2)$$

Due to the fact that different product features might occur in one review, the total self-information for a review $SI(review_i)$ is calculated as:

$$SI(review_i) = \sum SI(f_j, sentiment_{ji}) \quad (4.2.3)$$

$review_i$ denotes the i^{th} review, f_j denotes the j^{th} feature and $sentiment_{ji}$ denotes the sentiment for f_j in $review_i$. $SI(review_i)$ is the information gained for different sentiments for a product feature occurring in $review_i$.

- The divergence of sentiment sentences

Those reviews referring not only to the advantages but also the disadvantages satisfy both designers and potential consumers since a large mount of information is provided. For example, some sentences in one mobile phone review are "… Very good GPS, works very well with Google Maps. The built in GPS application is handy for calculating trip distance/current speeds etc. Very handy indeed. Nokia Maps are not very easy to use though…." In this review, both the advantage and the disadvantage about the GPS is mentioned. Valuable information is presented from online reviews.

Accordingly, the divergence of sentiments for one product feature is calculated, $DS(f_j)$, for product feature f_j as the sum of self-information, $SI(f_j$, sentiment), for three different sentiments (positive, negative and neutral) and it can be expressed as:

$$DS(f_j) = \sum_{s \in \{positive, negative, neutral\}} SI(f_j, s) \tag{4.2.4}$$

Also, due to the fact that different product features might occur in one review, the divergence information, $DS(\text{review}_i)$, for $review_i$ could be calculated as the sum of divergence information on different product features and can be expressed as:

$$DS(review_i) = \sum \mathrm{DS}(\mathrm{f}_{ij}) \tag{4.2.5}$$

f_{ij} denotes the j^{th} feature in $review_i$. $DS(review_i)$ is the information gained in $review_i$ referring to both the advantages and disadvantages of a product feature occurring in $review_i$.

- The strength of sentiment sentences

Reviews expressing a strong viewpoint towards some product features are usually preferred. Clearly, the strongest sentiment for a product feature might be positive, negative or neutral. The strength of sentiment is calculated, $SS(f_j)$, for product feature f_j as the maximum of self-information for three different sentiments and it can be expressed as:

$$SS(f_j) = max(SS(f_j,positive),SS(f_j,negative),SS(f_j,neutral)) \tag{4.2.6}$$

Accordingly, the sentiment strength for $review_i$, $SS(\text{review}_i)$, which mentions different product features, is calculated as:

$$SS(\text{review}_i) = \sum SS(f_{ij}) \tag{4.2.7}$$

$SS(\text{review}_i)$ is the information gained for designers if a strong and sharp viewpoint is expressed towards some product features.

4.2.4 Algorithm Description

4.2.4.1 Algorithms in Phase I

As expressed in Section 4.2.1, the regression model is applied in this research to predict the helpfulness of online reviews. The regression is initialized by introducing the bootstrap aggregating algorithm combined with a fast decision tree learner.

The fast decision tree learner builds a decision tree using information gained as the splitting criterion, and prunes it with reduced-error pruning. The bootstrap aggregating algorithm is a machine learning ensemble meta-algorithm to improve the classification and regression models in terms of stability and classification accuracy [Bre96]. Given a standard training set D of size N, the bootstrap aggregating algorithm generates M new training sets D_i, each of size $N' \geq N$, by sampling examples from D uniformly and with replacement. By sampling with replacement, which some examples are possible to be repeated in each D_i. This kind of samples is known as a bootstrap sample. The M models are fitted using the above M bootstrap samples and combined by averaging the output for regression or voting for classification.

Also, note that, a series of other experiments are actually conducted by using other prevailing algorithms such as the Multi-Layer perception neutral network (*MLP*), the simple linear regression (*SimpleLinear*), the sequential minimal optimization for training a support vector regression (*SMOreg*), and the decision tree algorithm (*REPTree*). They are all popular algorithms of machine learning, often being utilized in data regression to model complex relationships between various inputs and outputs. *MLP* is a mathematical model that is enlightened by the structure of biological neural networks. *SimpleLinear* is the least square estimator of a linear regression model with a single predictor variable. A straight line is fitted through a set of *n* points in a way that the sum of squared residuals of the model is made as small as possible. *SMOreg* globally replaces all missing values and transforms nominal attributes into binary ones. *REPTree* builds a decision or regression tree model using information gain reduction.

A tenfold cross-validation method is utilized for testing, and its results are reported based on the average rating of 1,000 reviews. Cross-validation is a technique to assess how the results of an analysis will generalize to an independent dataset [Gei93]. It is mainly used to estimate the accuracy of a predictive model in practice. One round of cross-validation involves partitioning data into complementary subsets, performing the analysis on one subset, and validating the analysis on the other subset. In the tenfold cross-validation, the data is randomly partitioned into ten sets of subsamples. A single set of subsample is retained for testing the model and the remaining sets are used as training data. The cross-validation process is then repeated in ten times, with each of the ten subsamples used exactly once as the validation data.

4.2.4.2 Algorithms in Phase II

In Phase II, the focus is whether the model learned in Phase I can be migrated to different products in other domain, which actually targets the impact delivered by certain features. Particularly, different algorithms of feature analysis and feature selection are chosen to identify the most informative and effective features. The algorithms of feature selection are:

- PCA-based feature selection schemes

  PCA on three variants of the feature matrix [JGDE08] are applied, namely, the original feature matrix normalized by its standard deviation, the eigenvalues and eigenvectors of the correlation matrix, and the eigenvalues and eigenvectors of the covariance matrix.

- Similarity-based feature selection schemes

  Three popular metrics are utilized: cosine similarity, Jaccard similarity and matching similarity. The Jaccard similarity and the matching similarity metrics are used to operate on sample sets. Before applying these metrics, the normalized values of each feature and instance target of a matrix are projected to a fixed number of groups. After that, these metrics are applied to the transformed matrix. Features are ranked accordingly, and the top ranked features are chosen as the selected features.

- Mutual information-based feature selection scheme

  Mutual information is a criterion commonly used in statistical language modeling of word associations and related applications. It is a famous method of feature selection, in which the mutual information between features and target values is estimated, and it was widely used in the field for years. Although Yang et al. [YP97] found that this feature selection scheme is not effective for text classification, no one then discussed about whether mutual information is an effective method for customer reviews' helpfulness evaluation. Inspired by this idea, the mutual information-based feature selection scheme is used.

4.2.5 Experimental Study and Discussions

4.2.5.1 Experiment Setup

In the experimental study, the 1,000 phone reviews collected from Amazon.com introduced in Section 3.3 are used. Since two different phases are proposed in Section 4.2.1, the performance is evaluated separately.

In order to evaluate the feasibility of Phase I, the average helpfulness value of the six designers was utilized as the helpfulness value of the online reviews. Altogether 83 features (including 6 linguistic features, 65 product features, 9 features using information quality and 3 features using information theory) were extracted. The bootstrap aggregating algorithm (-P 100 -S 1 -I 10) combined with a fast decision tree learner (-M 2 -V 0.001 -N 3 -S 1 -L 1) was utilized. The other comparable algorithms, like *MLP*, *SimpleLinear*, *SMOreg* and *REPTree*, were also tested, and it is found that the selected algorithm, i.e., bootstrap aggregating plus fast decision tree, performed better than other algorithms. A tenfold cross-validation method is utilized for testing and the results are reported based on the average of 1,000 repeated experiments.

In Phase II, the objective is to explore whether this model learned in Phase I is generic enough to be migrated to other products where the manually rated reviews may not be available. Various algorithms of feature selection are tested in this phase to examine the availability for this specific concern. These feature selection algorithms are: (1) three PCA-based feature selection schemes: PCA on the original feature-instance matrix (denoted by PCA), PCA on the correlation matrix of the original feature-instance matrix (denoted by PCACorr), and PCA on the covariance matrix of the original feature-instance matrix (denoted by PCACov), (2) three feature selection schemes based on different feature-instance similarity metrics, i.e., cosine similarity, Jaccard similarity and matching similarity, and (3) mutual information-based feature selection scheme.

For the datasets in other topics, 904 digital camera reviews and 1,026 shaver reviews were randomly chosen from Amazon.com. Due to the constraint of the budget allocated for this experiment, in Phase II, only two designers were assigned for each set of reviews. However, according to the previous research [LL07], it is found that "combining two best operators' results is able to achieve close-to-best results" in corpus building. The average of the two designers' helpfulness evaluation values is then taken as the golden criteria. The average of designer evaluations was compared with the helpfulness predicted from Phase I using the mobile phone dataset which is a different category of products. Hence, the robustness of the regression algorithm is evaluated, and the conjecture on whether the model learned in one domain can be transferred to other domain is examined.

All programs were implemented and tested in Java 1.6 and Weka 3.6 on a dual core 2.40GHz personal computer with 4GB memory. The interface of using the 1,000 mobile phone dataset is illustrated in Figure 4.3.

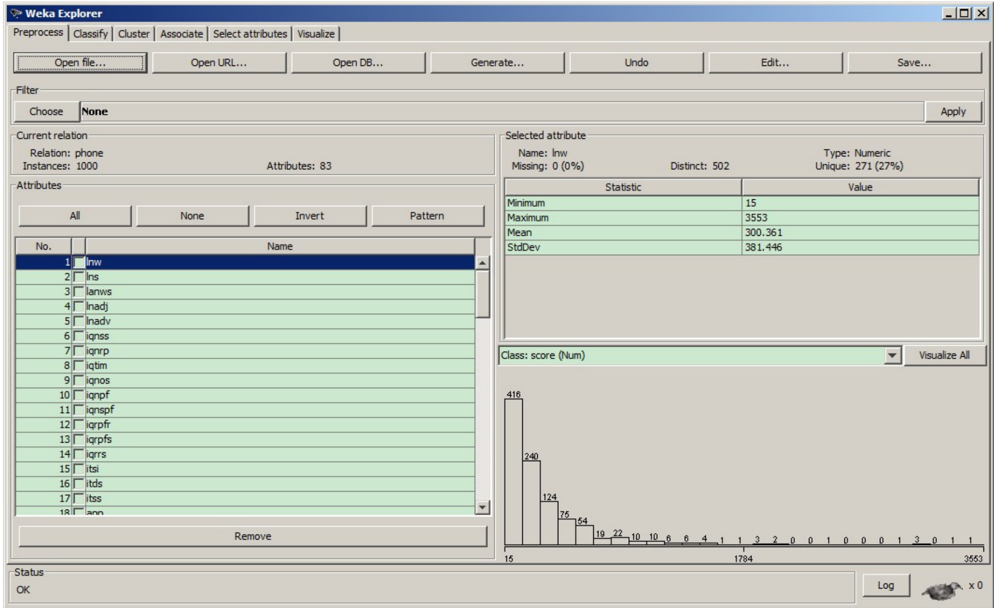


Figure 4.3 User interface for the 1,000 phone reviews dataset

4.2.5.2 Results and Discussions

- Phase I: Helpfulness Prediction with Domain Features

The result of the predicted helpfulness compared with designers' rating is demonstrated in Table 4.4.

Table 4.4 Predicted value results

MAE	*RMSE*	*PMCC*
0.599	0.487	0.795

As seen from Table 4.4, the predicted helpfulness indicates a strong *PMCC*, and small *MAE* and *RMSE*, 14.9% $(\frac{0.599}{2-(-2)})$ and 12.1% $(\frac{0.487}{2-(-2)})$ of the scale region, respectively, with designers' rating. Compared with Table 3.1, the proposed model performs much better than the method using the percentage of online helpful votes. The proposed model is shown to be better in interpreting the helpfulness in the designers' viewpoint.

For the algorithm selection in Phase I, more experiments with other algorithms, including *MLP*, *SimpleLinear*, *SMOreg* and *REPTree* were conducted. The performance with these algorithms is compared in Figure 4.4.

As seen from Figure 4.4, compared with these four algorithms, the selected algorithm, the bootstrap aggregating algorithm combined with the fast decision tree learner, performs better in all of the three evaluation metrics (higher *PMCC*, lower *MAE* and lower *RMSE*). Also, a better performance, compared with both the result of Phase I and the real average helpfulness rating, is achieved.

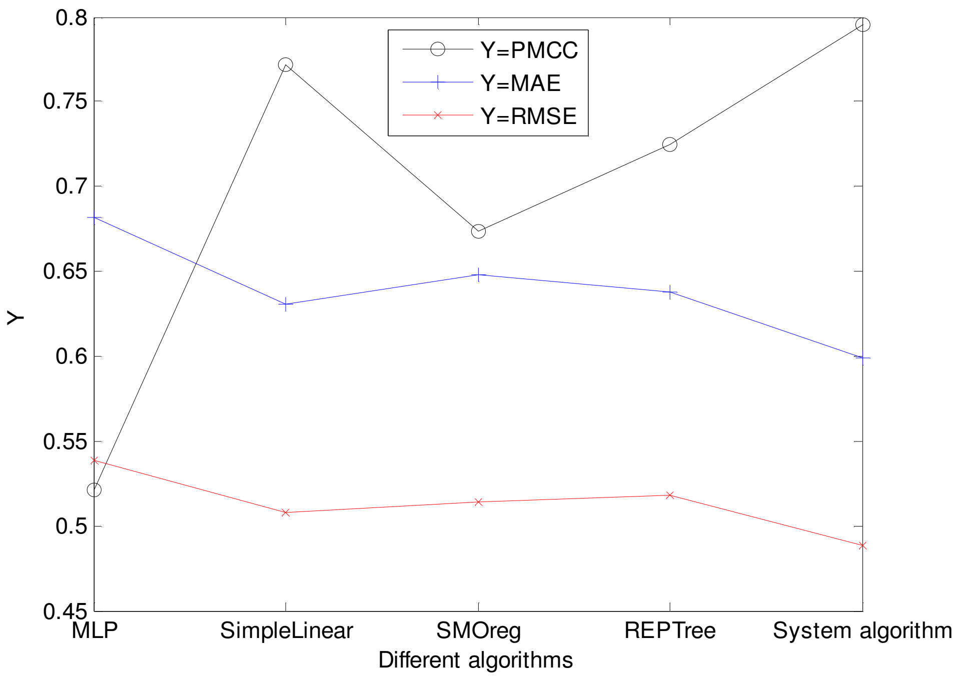
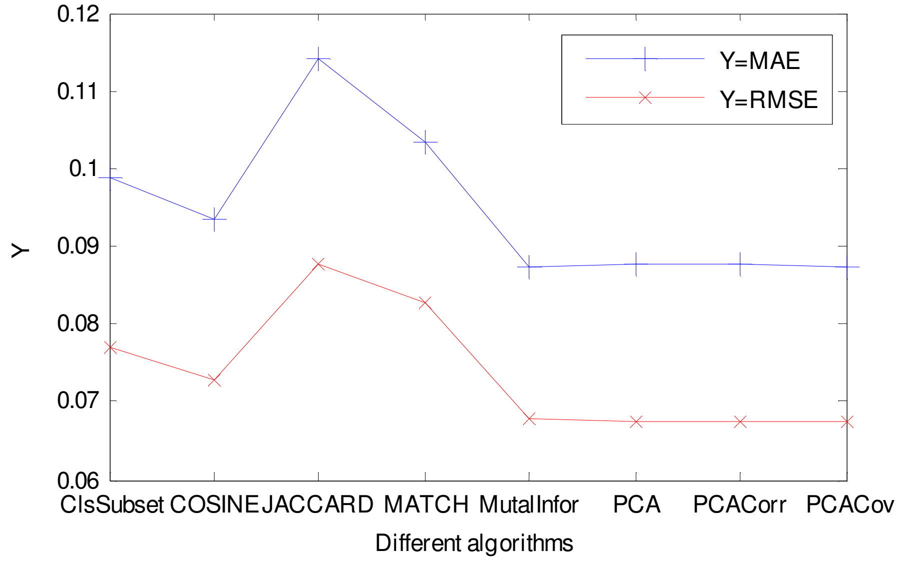


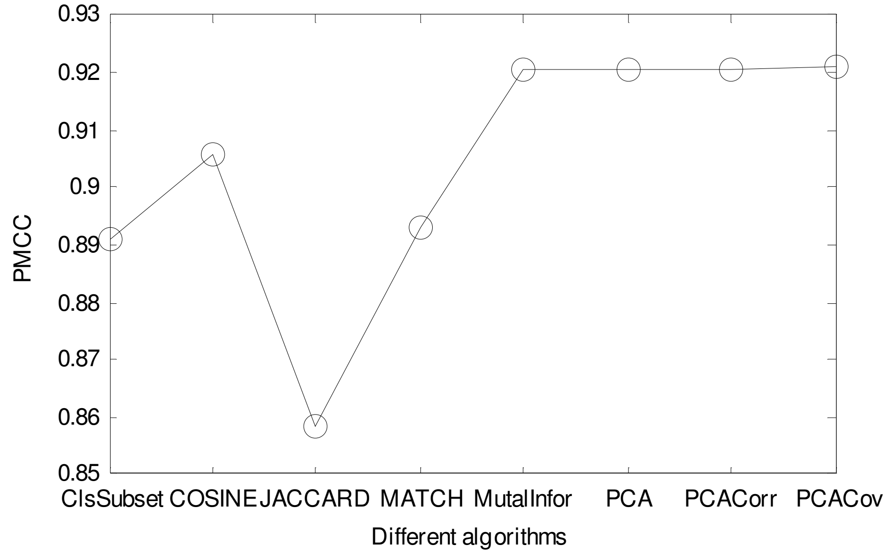
Figure 4.4 Comparisons on different algorithms in Phase I

- Phase II: Helpfulness Prediction without Domain Features

The prediction results of eight feature selection schemes about review helpfulness are compared in Figure 4.5. Top features are utilized for all these schemes. The mutual information and PCA perform best in all three datasets. For these two schemes, the predicted values reach small distances. Hence, they are chosen as the applied schemes.



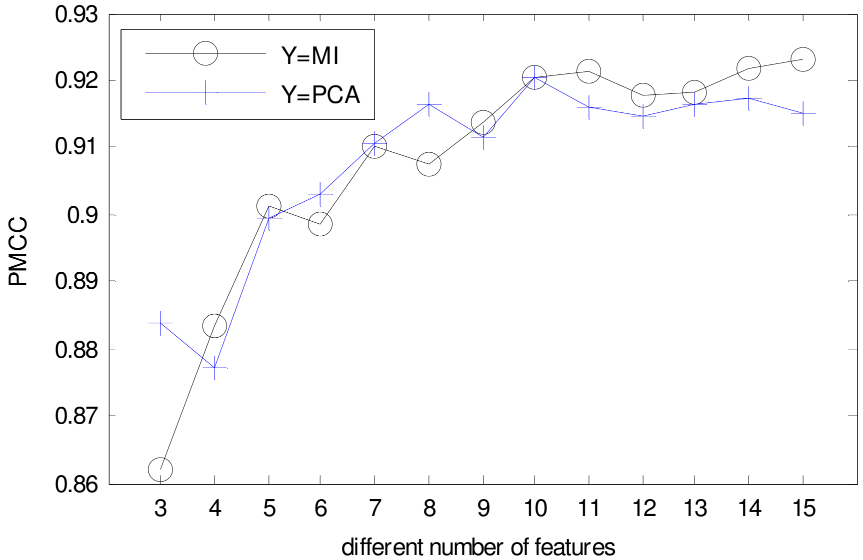
(a) *MAE* and *RMSE* results in phase II



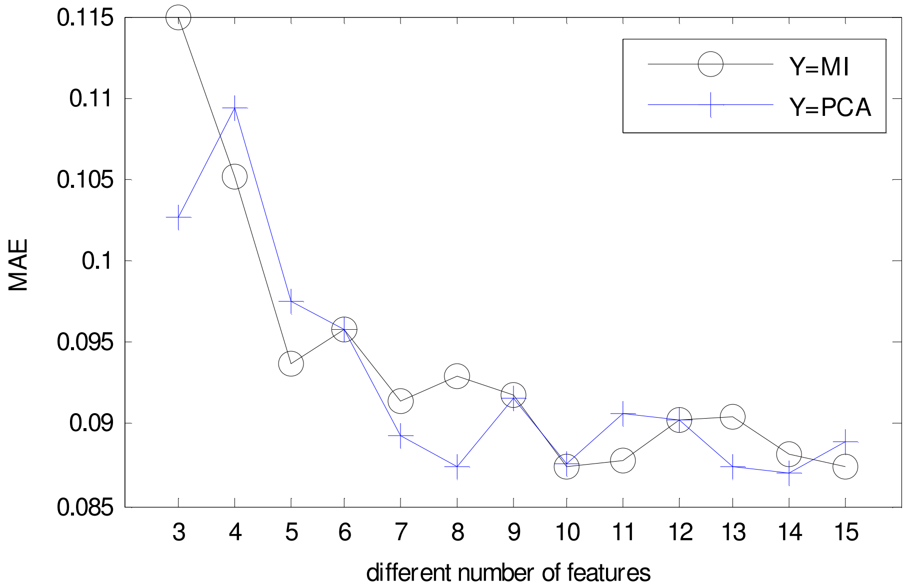
(b) *PMCC* results in phase II

Figure 4.5 Comparisons on different algorithms in Phase II

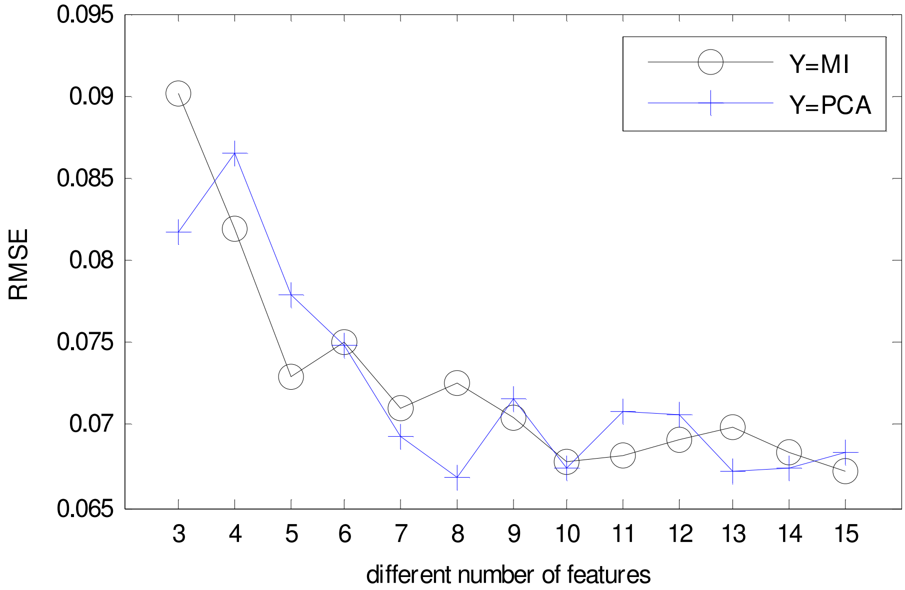
As seen from Figure 4.6, *PMCC* increases (*MAE* and *RMSE* decrease) with the increasing number of selected features. The curve reaches a relatively stable position if the number equals to nine or ten for both metrics of feature selection. It means that nine or ten important extracted features have the most influential impacts on the evaluation of review helpfulness.



(a) *PMCC* at different numbers of features



(b) *MAE* at different numbers of features



(c) *RMSE* at different numbers of features

Figure 4.6 Performance on different numbers of features

In Table 4.5, the features for both schemes are listed, respectively. As seen from these features, nearly all of them come from three categories of domain independent features.

As mentioned before, the objective is to further verify whether the three categories of domain independent features can be applied successfully to predict the helpfulness of online reviews without losing the prediction accuracy in other domains. Whether the predicted helpfulness of online reviews from Phase I demonstrate a strong correlation with the average rating from designers needs to be evaluated. It examines whether the helpfulness of online reviews is affected by the categories of domain dependent features.

Table 4.5 Selected features using mutual information and PCA

	MI	PCA
1st	L-NW	IQ-NSPF
2nd	L-NADJ	L-NW
3rd	L-NS	L-NADJ
4	L-NADV	L-NS
5	IQ-NSPF	IQ-NPF
6	IQ-NOS	IT-SI
7	IT-SI	LN-ADV
8	IQ-NPF	IT-DS
9	IQ-NSS	IQ-NOS
10	VIDEO	IQ-NSS

The performance in terms of other two different products was also evaluated. Altogether 904 digital camera reviews and 1,026 shaver reviews were employed to test the previous conjecture. Similarly, another four designers (two for digital camera reviews and two for shaver reviews) were invited to label online reviews according to the helpfulness labeling instruction in Section 3.3. The average scores from two designers are utilized as the helpfulness value of the online reviews. Notice that only three categories of domain independent features (including six linguistic features, nine features using information quality and three features using information theory) are utilized for these two datasets respectively in Phase II. Also, the bootstrap aggregating algorithm combined with a fast decision tree learner is still chosen as the selected algorithm.

As shown in Figure 4.7, the prediction performance by the three categories of domain independent features is illustrated. A strong correlation is shown between the predicted helpfulness and designers' evaluation. Also, the selected algorithm outperforms the other four benchmarking algorithms. The results demonstrate it is possible to model the helpfulness of online review from the perspective of product designers as domain-free.

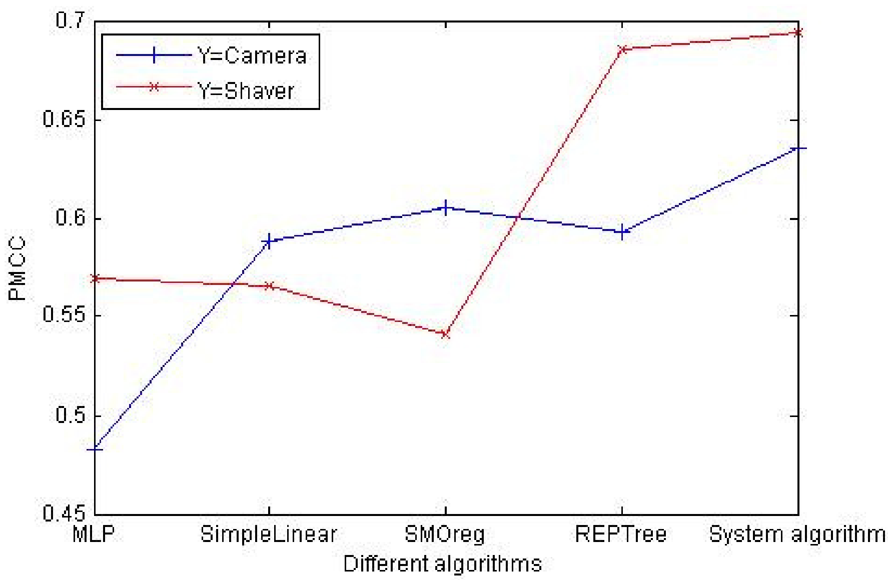


Figure 4.7 Performance using domain independent features

- Features Analysis

In the follow-up questionnaire of the case study, the key question was, i.e., "what are the contributing factors that make you regard this review as helpful?" For example, some designers mentioned that "it has many different features", "it points out the like and dislike of the product", etc. The underlying reason is that the presence of such features improves the helpfulness of online reviews. In other words, if some features are missing, it implies that the review may not be perceived as useful as others. As for feature analysis, the importance of different features is evaluated. For the feature vector formation for the data file of Weka, zero denotes the absence of the corresponding feature and a non-zero value denotes the feature's relative importance. As suggested from Table 4.5, the helpfulness prediction is found not to be influenced significantly by the absence of some product features, and different categories of features possess different impacts on the helpfulness value predicted.

It is observed that, from Table 4.5, some linguistic features (L-NW, L-NADJ, L-NADV, and L-NS) have significant influence on the helpfulness evaluation. A simple assumption for the helpfulness evaluation is justified as "a longer review tends to be more helpful". It can also be inferred that a long review tends to be written by an experienced consumer, no matter whether the product is preferred or not, and these kinds of reviews tend to be more helpful than short ones.

Still, it is noticed that some features (IQ-NSPF, IQ-NOS, IQ-NPF and IQ-NSS) extracted using IQ, also have important impacts on the helpfulness evaluation. IQ-NSPF and IQ-NPF confirm that the reason, "this review mentions many product features", is important for the helpfulness evaluation. IQ-NOS and IQ-NSS reflect the information accuracy of online reviews. It implies that information accuracy for reviews plays an important role in the helpfulness evaluation.

IT-SI is another important factor in the helpfulness evaluation. It illustrates that those reviews, containing a different sentiment that deviates from the majority sentiment, tend to be helpful. These reviews are highly possible to contain further details about this sentiment and explanations about the excuses. That would be what product designers prefer with their follow-up work.

Another observation is that many other comparable products might not necessarily be compared with in one helpful product review (IQ-NRP), which slightly contradicts some similar assumptions that "a helpful product review would mention many other different products". It could be guessed that different products are not necessarily mentioned in many online customer reviews and one or two similar products are only mentioned in a helpful review for preference comparisons. The helpfulness will not always be improved much with the increasing numbers of products mentioned.

The feature analysis, using mutual information-based feature selection scheme and the principal component analysis (PCA), exploits the utilization of features in terms of predicting the helpfulness. The applicability of some heuristic evaluation rules has been verified: "Helpful reviews tend to mention many product features trends", "Helpful reviews tend to be longer", "Helpful reviews tend to mention both the advantages and disadvantages of products", etc. It is also noted that features which are extracted based on information quality have a greater influence on the helpfulness prediction.

4.3 RATING VALUE RECOMMENDATION ON ONLINE REVIEWS

4.3.1 Problem Statement

As seen from the first exploratory case study in Chapter 3, some reviews receive a large divergence in terms of the helpfulness evaluation from the six designers. For example, as stated in the experiment of the case study, one review was labeled with two "2"s, two "–2"s, one "1" and one "–1". Four different helpfulness evaluations were presented by the six annotators. Moreover, according to the follow-up questionnaires, the viewpoints of different designers are not always aligned with each other.

The implication behind this phenomenon may be that each designer has his or her own perspective or criteria in understanding the helpfulness of online reviews. In order to fulfill each designer's requirement, the crucial question is how to recommend rating values on online reviews by taking his or her personal assessments into consideration.

To formally define the recommendation of rating values, some notations are introduced at first.

The set of designers is D and the set of reviews is R. The set of rating values is L (e.g., $L = \{-2, -1, 0, 1, 2\}$). A single designer $d \in D$ gives a rating $rating(d, r)$ for a particular review $r \in R$. The question is, how to recommend $rating(d, r)$ as $h(d, r)$, of a designer d on a review r. $h(d, r)$ is denoted as the recommended rating and $rating(d, r)$ is the real rating given by the designer d on review r.

This question can be formulated by a regression or a classification model in which the goal is to learn a model or function $f(d, r)$: $\mathrm{D} \times \mathrm{R} \rightarrow L$. This function is expected to estimate *rating*(d, r) as $h(d, r)$ of a designer d on a review r. Now, the technical consideration is whether a regression or classification method is more appropriate.

The choice between classification and regression largely depends on the rating scale. If the rating scale is continuous, then a regression is more appropriate. On the contrary, if the rating scale has some discrete values, then a classification is preferable. Another way to make a comparison is by considering the situation where all the instances are treated equally. Exactly, the rating value, predicted by the regression approach, tends to be the mean rating of all reviews. On the other hand, the classification will suggest the recommendation as the most frequent label [DK11]. For example, suppose a review r has rated as either helpful or unhelpful. The regression tends to make the safe decision as the average value, while the classification will suggest the most frequent label as either "helpful" or "unhelpful".

In this research, a five-discrete-graded-review rating system is adopted. Thus, a classification is chosen to recommend rating values on online reviews.

4.3.2 Overview of the Rating Value Recommendation on Online Reviews

4.3.2.1 Technical blueprint of the online review's rating value recommendation

An overview of the online review's rating value recommendation is given in Figure 4.8.

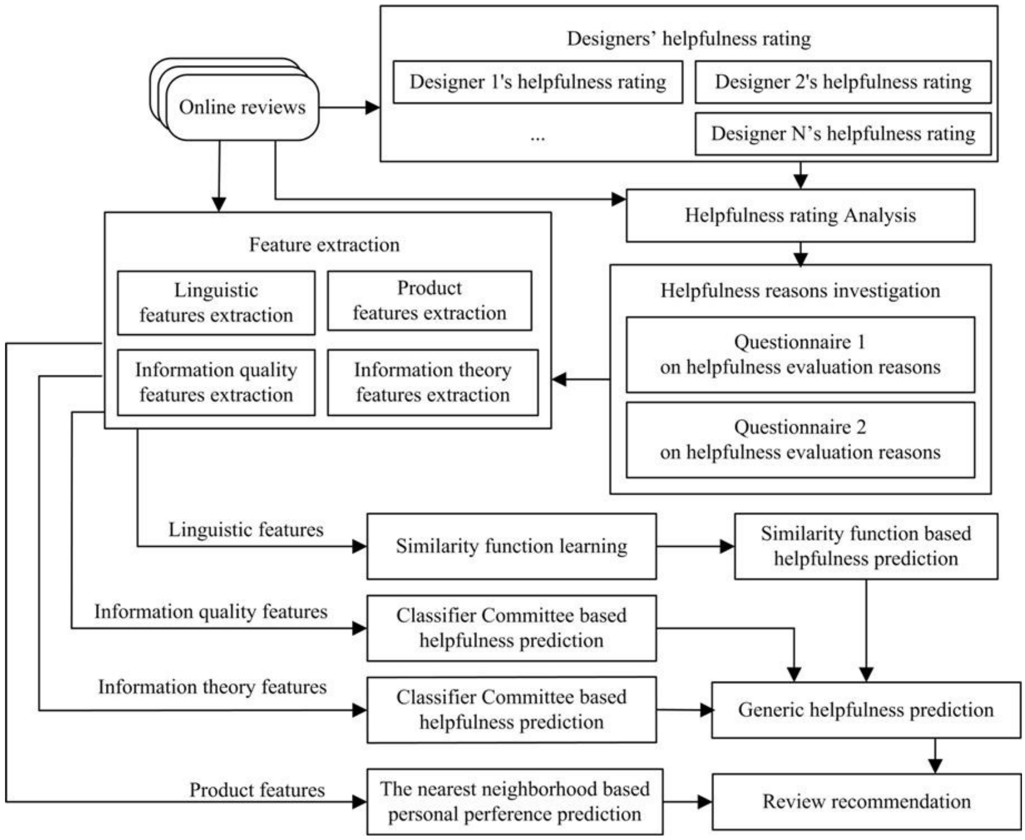


Figure 4.8 The technical blueprint

Based on the understanding of how designers perceive online reviews, the four categories of features are utilized to model how to recommend rating values on online reviews. As illustrated in the previous sections, the four categories of features are generated entirely from the review content. They are presented in the form of feature vectors, such as, linguistic features $x\_l(r)$, product features $x\_p(r)$, features extracted based on information quality $x\_qual(r)$, and features extracted using information theory to evaluate information quantity $x\_quan(r)$.

As illustrated in the exploratory case study, different designers presented different ratings towards the helpfulness of online reviews and they have their own criteria of judgments on why some reviews are helpful. The rationale is that different product designers have different requirements and the intuition may be that it is the occurrence of different product features that interest different designers to give various ratings towards the helpfulness of online reviews. In this research, the assumption is that, *rating*(d, r), a rating of a designer d on a review r, is evaluated from two different aspects, a generic aspect *gh*(d, r) and a personal assessments aspect *pp*(d, r). Specially, the question is, with the four categories of features, how to construct a classification model to make the recommendation, considering both the generic aspect *gh*(d, r) and the designer's personal assessments aspect *pp*(d, r). The assumption will be evaluated through the proposed model in the experimental study section.

In the generic aspect, *gh*(d, r) is evaluated from three domain-independent dimensions, namely, the linguistic dimension, the information quality dimension and the information quantity dimension. These three categories of features are all generated from the review content and they are all domain independent features. It interprets the reasons why they are utilized to interpret the generic aspect. The personal assessments aspect, *pp*(d, r) is evaluated from product feature dimension. Finally, the results from two aspects are combined through a classification method to recommend rating values on online reviews.

4.3.2.2 Technical Considerations

First, the generic helpfulness *gh*(d, r) intends to be evaluated from three dimensions. Different from the previous approach introduced in Section 4.2, three feature vectors, *x_l*(r), *x_qual*(r), and *x_quan*(r), are utilized to predict the generic helpfulness one by one. Specially, utilizing the linguistic features $x\_l(r)$, the rating value on an online review r is estimated as $gh\_l(d, r)$. Similarly, utilizing the features extracted based on information quality $x\_qual(r)$ and the features extracted from the information quantity dimension $x\_quan(r)$, the rating value on an online review r is $gh\_qual(d, r)$ and $gh\_quan(d, r)$, respectively. When three results $gh\_l(d, r)$, $gh\_qual(d, r)$ and $gh\_quan(d, r)$ are available, they are combined through a classification classifier to predict the generic helpfulness $gh(d, r)$.

By intuition, if similar ratings are given to two reviews, they may be literally similar to each other. Thus, in the linguistic dimension, $gh\_l(d, r)$ is predicted by a similarity function learning based method. The similarity function is to discover the similarity between review content in terms of review labeling. If the similarity function can be defined and learned, for an unlabeled review r, $gh\_l(r)$ can be predicted as the review label $l \in L$, which receives the maximum average similarity.

In the information quality dimension and the information quantity dimension, only a few features defined and extracted from review content (eight features in $x\_qual(r)$ only and three features in $x\_quan(r)$ only as illustrated in Section 4.2.2). Hence, since there might not be sufficient features, if a multiple classifier is directly used to make a five-discrete-level classification, it is highly possible that a satisfactory result might not be gained. For instance, if three features in $x\_quan(r)$ and a five-discrete-level classification classifier are applied directly, $gh\_quan(d, r)$ might not be good. A similar result might also occur for $gh\_qual(d, r)$. However, a well-known fact of computational learning theory is that, the more independent a set of classifiers is, the better they perform once assembled into a committee. Hence four individual classifiers (three binary classifiers and one ternary classifier) are combined into a committee, which is utilized to decide the result from this dimension.

In the personal assessments aspect, *pp*(*d*, *r*) is evaluated by product features and the occurrence of different product features is regarded as one important criterion in personal assessment. For instance, review *r* may receive the highest labeling from one designer since it is thought to point out some drawbacks about some features. Meanwhile, it might also receive the lowest labeling from another designer who argues that it does not talk about the product features which are relevant to one's concerns. On the other hand, intuitively, if designer d_u and designer d_v have similar assessments, or say, they rate other reviews in a similar way, the rating of d_u on a review r_i is likely to be similar to that of d_v. Likewise, designer d_u is likely to rate two reviews r_i and r_j in a similar fashion, if other users have given similar ratings to these two reviews. Therefore, the nearest neighborhood approach is used to predict *pp*(*d*, *r*).

Again, in order to clarify the recommendation of the online review's rating value for different product designers, several concerns that need to be considered in this research are:

(1) Evaluate reviews from the linguistic dimension as *gh_l*(*d*, *r*)

(2) Evaluate reviews from the information quality dimension as *gh_qual*(*d*, *r*)

(3) Evaluate reviews from the information quantity dimension as *gh_quan*(*d*,*r*)

(4) Estimate the generic helpfulness from the three dimensions as *gh*(*d*, *r*)

(5) Evaluate reviews from the personal assessment as *pp*(*d*, *r*)

(6) Recommend the rating value based on *gh*(*d*, *r*) and *pp*(*d*, *r*) as *h*(*d*, *r*)

4.3.3 Technical Approach of Review Recommendation

4.3.3.1 Similarity Learning Method

In this subsection, the focus is on how to learn a similarity function between customer reviews in the linguistic dimension.

Two reviews are supposed to receive similar helpfulness ratings if they are literally similar to each other. For example, there are four different reviews, r_a, r_b, r_c and r_d. Their corresponding helpfulness ratings are "2", "2", "–1" and "–2". In the first rule, the similarity between r_a and r_b should be greater than the similarity between r_a and r_c because both r_a and r_b receive the same "2" helpfulness rating, while r_c is labeled as "–1", which is different from the rating of r_a and r_b. From the second rule, the similarity between r_a and r_c should be greater than the similarity between r_a and r_d because the distance between r_a and r_c is closer than the distance from r_a and r_d. Formally, two rules are written as:

- Rule one

$$h(d,r_a) == h(d,r_b) \;\&\&\; h(d,r_a) \neq h(d,r_c) \Rightarrow sim(r_a, r_b) \geq sim(r_a, r_c) \quad (4.3.8)$$

- Rule two

$$h(d,r_a) \geq h(d,r_c) \;\&\&\; h(d,r_c) \geq h(d,r_d) \Rightarrow sim(r_a, r_c) \geq sim(r_a, r_d) \quad (4.3.9)$$

Given these rules, the objective is to find a function that maximizes the sum of similarities between reviews receiving the same ratings and the sum of similarities between reviews receiving the closer ratings. Accordingly, a similarity learning method is proposed, which can be defined as an optimization problem as shown in Model (4.3.10).

$$
\begin{aligned}
&\max \mu \sum_{\forall r_a, r_b, r_c \in c1} (sim(r_a, r_b) - sim(r_a, r_c)) + (1-\mu) \sum_{\forall r_a, r_b, r_d \in c2} (sim(r_a, r_b) - sim(r_a, r_d)) \\
&s.t. \\
&\quad sim(r_x, r_y) \in P \\
&\quad c1: sim(r_a, r_b) \geq sim(r_a, r_c) \\
&\qquad \forall h(d, r_a) = h(d, r_b) \;\&\&\, h(d, r_a) \neq h(d, r_c) \\
&\quad c2: sim(r_a, r_b) \geq sim(r_a, r_d) \\
&\qquad \forall h(d, r_a) \geq h(d, r_c) \;\&\&\, h(d, r_c) \geq h(d, r_d)
\end{aligned}
\tag{4.3.10}
$$

The goal is to find a similarity function $sim(r_x,r_y)$ in function set P that enables the objective of Model (4.3.10) to be maximized. The random variables of this optimization problem are parameters which define the similarity function. However, the formation of the similarity function is uncertain. In this research, the Taylor series method is employed to define the similarity function. It is denoted as:

$$sim(r_a,r_b) = \sum C(t, k)[f(r_a)]^t[f(r_b)]^k \tag{4.3.11}$$

$C(t, k)$ refers to the parameters to be learned. $f(r_a)$ is the linguistic features of review r_a and $[\cdot]^t$ is the power operator for linguistic features.

Given the training data, the optimization problem in Model (4.3.10) can be solved by an optimization method. An optimal solution of $C(t, k)$ will be obtained. With $C(t, k)$, the similar function is then utilized to estimate $gh\_l(d, r)$ by the linguistic features $x\_l(r)$.

Nevertheless, there are some other concerns. First, a large number of constraints make this optimization problem almost impossible to be solved within a reasonable time. Suppose there exist n reviews in the training set and there will be $\binom{n}{3} = \frac{n \times (n-1) \times (n-2)}{6}$ constraints. In an extreme case, it becomes unsolvable. Another consideration is that designer ratings for different online reviews might not always be consistent due to the vagueness or the uncertain judgments in the procedure of review evaluation. Thus, some compromises have to be made to derive the similarity function.

First, the question is changed to find a similarity function that minimizes the number of violations on the constraints, or maximizes the number of constraint satisfaction conditions. However, it is an NP hard problem. A further change has to be made. Starting from a random separation on the constraints, in the first step, the optimization problem can be solved with a smaller number of constraints. In the second step, different similarity functions, which is learned in one separation, is applied to the other separation. The similarity function with the smallest number of violations is chosen as the final similarity function. Finally, $gh\_l(d, r)$ is predicted as the rating which receives the maximal average similarity:

$$gh\_l(d,r) = \arg\max_h (\underset{h=h(d,r)}{avg} (sim(r, r_i))) \tag{4.3.12}$$

4.3.3.2 Build Classifier Committee

As mentioned in the previous section, a better performance may be obtained if multiple classifiers are assembled into one committee. In this research, four classifiers, including three binary classifiers φ_1, φ_2, φ_3 and one ternary classifier φ_4 are utilized to predict both $gh\_qual(d, r)$ and $gh\_quan(d, r)$.

For instance, a review r is given a label c, and the feature vector of r is x. A numerical score $s \in \{0, 1\}$ will be predicted by binary classifiers (φ_1, φ_2, or φ_3). "1" implies that review r belongs to class c and "0" stands for the review r does not belong to class c. In this case, the score s denotes the confidence that r belongs to the class c, and the three binary classifiers are defined as follows.

φ_1 attempts to classify whether a review is helpful (including "1" and "2") or not (including "0", "–1" and "–2").

φ_2 attempts to classify whether a review is unhelpful (including "–1" and "– 2") or not (including "0", "1" and "2").

φ_3 attempts to classify whether a review contains two helpful polarities (including "2" and "–2") or not (including "0", "1" and "–1").

All the numerical numbers from "–2" to "2" are the helpfulness ratings suggested by designers. Some rating values can be derived logically from φ_1, φ_2, φ_3. The suggested result from these three binary classifiers is shown in Table 4.6.

Table 4.6 The suggested result from three binary classifiers

φ_1	φ_2	φ_3	Result
Helpful	NOT Unhelpful	Polarity	2
Helpful	NOT Unhelpful	NOT Polarity	1
Helpful	Unhelpful	-	UNDEFINED
NOT Helpful	NOT Unhelpful	-	0
NOT Helpful	Unhelpful	Polarity	-2
NOT Helpful	Unhelpful	NOT Polarity	-1

For instance, there is a particular review r. r is classified as helpful from φ_1, not unhelpful from φ_2, and has helpful polarity from φ_3. Accordingly, r is labeled with "2" by the classifier committee.

However, only three binary classifiers are not sufficient to make a satisfactory separation on the five-graded rating levels. For example, if r is classified as helpful from φ_1, unhelpful from φ_2, and has helpful polarity from φ_3, then the rating of r still can not be derived logically. Therefore, a ternary classifier φ_4 is introduced.

φ_4 attempts to classify whether a review is helpful (including "2" and "1"), unhelpful (including "–2" and "–1") or receiving compromising rating (that is, "0").

Once the three existing binary classifiers, φ_1, φ_2, and φ_3, are not able to decide the rating, φ_4 is triggered to make the final decision. In the previous example, if review r is classified as helpful from φ_4, "2" will be labeled to r (two "–2"s, one "–1", two "1"s, three "2"s). On the other hand, if review r is classified as a compromising rating from φ_4, "0" will be labeled to r.

Finally, with the three binary classifiers and one ternary classifier, both *gh_qual*(*d, r*) and *gh_quan*(*d*, *r*) can be obtained accordingly.

4.3.3.3 The Nearest Neighbor Method

The proposed nearest neighbor method will be discussed, which enjoys a huge amount of popularity due to its simplicity and efficiency, to predict the personal assessments by product features.

The research motivation is that it will be utilized in the scenario that only a small number of online reviews are evaluated in terms of helpfulness and the rating values for other reviews are expected to be recommended. In such scenario, both the rating values of given online reviews from this designers and the rating values of other designers will be utilized to develop this method. Hence, by intuition, one designer may rely on the opinions of like-minded designers or other similar reviews to evaluate the rating of a review according to one's own assessments. The nearest neighbor method relies on the ratings of similar designers and the ratings of similar reviews to predict the personal assessments $pp(d, r)$ of designer d for review r.

This algorithm will be described by defining some equations step by step.

Suppose the average rating of the six designers for all reviews R is denoted as $avg(R)$. The average rating and the bias rating from designer d on all reviews are $avg(d)$ and $bias(d)$. The bias rating evaluates the distance deviation from the average rating for designers to the average rating from designer d, that is,

$$bias(d) = avg(d) - avg(R) \tag{4.3.13}$$

Similarly, the average rating and bias ratings from all designers for review r are $avg(r)$ and $bias(r)$, respectively. The bias for review r can be defined as:

$$bias(r) = avg(r) - avg(R) \tag{4.3.14}$$

Accordingly, the bias, $bias(d, r)$, can be defined as the observed rating deviations of designer d for review r.

$$bias(d, r) = avg(R) + bias(d) + bias(r) \tag{4.3.15}$$

Based on the definition of $bias(d, r)$, the residual, $residual(d, r)$, which indicates the observed rating distance of d for r, is utilized in many nearest neighbor method. It can be calculated as:

$$residual(d, r) = rating(d, r) - bias(d, r) \tag{4.3.16}$$

For example, using a similar definition of the residual, a collaborative filtering method with jointly derived neighborhood interpolation weights was proposed [KB11], which is actually a K nearest neighbor method (KNN).

KNN is a simple yet effective algorithm, where an object is classified by a majority vote of its neighbors, with the object being assigned to the most common class amongst its K nearest neighbors. In this collaborative filtering method, the interpolation weights θ is utilized as follows:

$$pp(d, r_i) = bias(d, r_i) + \sum_{r_j \in KNN(r_i)} \theta \cdot residual(d, r_j) \tag{4.3.17}$$

However, in this collaborative filtering method, only an item-based recommendation is given. Different from this item-based recommendation method, and many other user-based recommendation methods, in this research, the personal assessment is predicted with the nearest neighbor method from two points of view: the designer centered prediction *pp_d*(*d, r*) and the review-centered prediction *pp_r*(*d, r*).

With several equations defined previously, the two aspects which are combined in the hybrid recommendation model can be defined as:

$$pp\_d(d, r) = \frac{\sum_{d_a \in KNN(d)} sim(d, d_a) \cdot residual(d_a, r)}{\sum_{d_a \in KNN(d)} sim(d, d_a)} \tag{4.3.18}$$

$$pp\_r(d, r) = \frac{\sum_{r_a \in KNN(r)} sim(r, r_a) \cdot residual(d, r_a)}{\sum_{r_a \in KNN(r)} sim(r, r_a)} \tag{4.3.19}$$

Cosine similarity, cos(X, Y), is adopted to decide the contributions of each nearest neighborhood for both reviews and designers, which is defined as:

$$\cos(X, Y) = \frac{X \cdot Y}{\| X \| \| Y \|} = \frac{\sum_{i=1}^{n} X_i \times Y_i}{\sqrt{\sum_{i=1}^{n} (X_i)^2} \sqrt{\sum_{i=1}^{n} (Y_i)^2}} \tag{4.3.20}$$

Actually, some other similarity metrics, such as, the Pearson product moment correlation coefficient and Jaccard similarity, are also tested in this step. These algorithms do not perform well but involve much computation. Hence, the cosine similarity is chosen in this research.

Moreover, a linear combination is utilized for *pp*(*d*, *r*) to balance both *pp_d*(*d, r*) and *pp_r*(*d, r*) with three unknown parameters ω_1, ω_2 and ω_3.

$$
\begin{aligned}
& pp(d,r) \\
&= bias(d,r) + \omega_1 \cdot pp\_d(d,r) + \omega_2 pp\_r(d,r) + \omega_3 \\
&= bias(d,r) + \omega_1 \cdot \frac{\sum_{d_a \in KNN(d)} sim(d,d_a) \cdot residual(d_a, r)}{\sum_{d_a \in KNN(d)} sim(d,d_a)} \\
&\quad + \omega_2 \cdot \frac{\sum_{r_a \in KNN(r)} sim(r,r_a) \cdot residual(d, r_a)}{\sum_{r_a \in KNN(r)} sim(r,r_a)} + \omega_3
\end{aligned}
\tag{4.3.21}
$$

Three parameters can be derived from training data. Accordingly, using product features, personal assessments of designers are predicted by using this proposed nearest neighbor method.

4.3.4 Experimental Study and Discussions

4.3.4.1 Experiment Setup

In the experimental study, the 1,000 phone reviews introduced in Section 3.3 are used.

First, in this experiment study, the effectiveness of the learned similarity function is verified. It will be testified by the percentage violations of two heuristic rules, which is introduced in Section 4.3.3.1, for different rating values. Different similarity function will be compared, including the cosine similarity, the Pearson product moment correlation coefficient (*PMCC*), and, finally, the *p*-norm distance. The *p*-norm distance, $\|(X, Y)\|_p$, where p equals to "1", "2","3", are defined as followed:

$$\| (X,Y) \|_p = (\sum_{i=1}^{m} | X_i - Y_i |^p)^{\frac{1}{p}} \tag{4.3.22}$$

Also, in order to make a comparison between the classification and regression to recommend rating values, in this research, SVM is chosen as a fair framework. Though SVM is often utilized as a classification algorithm, it has been successfully extended into regression algorithms (like *epsilon-SVM*). *C-SVM* is applied as the classification algorithm and *epsilon-SVM* as the regression algorithm because they have a similar structural loss function and the optimization method.

In terms of evaluation metrics, both classification-based and regression-based performance metrics are examined.

Precision and *Recall* are used to evaluate the performance in terms of classification. *Precision* is the fraction of retrieved instances that are relevant and it is calculated as the number of correct results divided by the number of all returned results.

$$Precision = \frac{|\{relevant\ instances\} \cap \{retrieved\ instances\}|}{|\{retrieved\ instances\}|} \tag{4.3.23}$$

Recall is the fraction of relevant instances that are retrieved, which is calculated as the number of correct results divided by the number of results that should be returned.

$$Recall = \frac{\left|\{relevant\ instances\} \cap \{retrieved\ instances\}\right|}{\left|\{relevant\ instances\}\right|} \tag{4.3.24}$$

For the regression-based performance metrics, *MAE* and *RMSE* (see Equation 3.3.2 and Equation 3.3.1) are utilized.

The ten-fold cross validation approach was utilized in the experiments. All experiments were implemented and tested in Java 1.6 and Weka 3.6 on a dual core 2.40 GHz personal computer.

4.3.4.2 Results and Discussions

The comparison of similarity metrics on designer ratings can be seen in Figure 4.9.

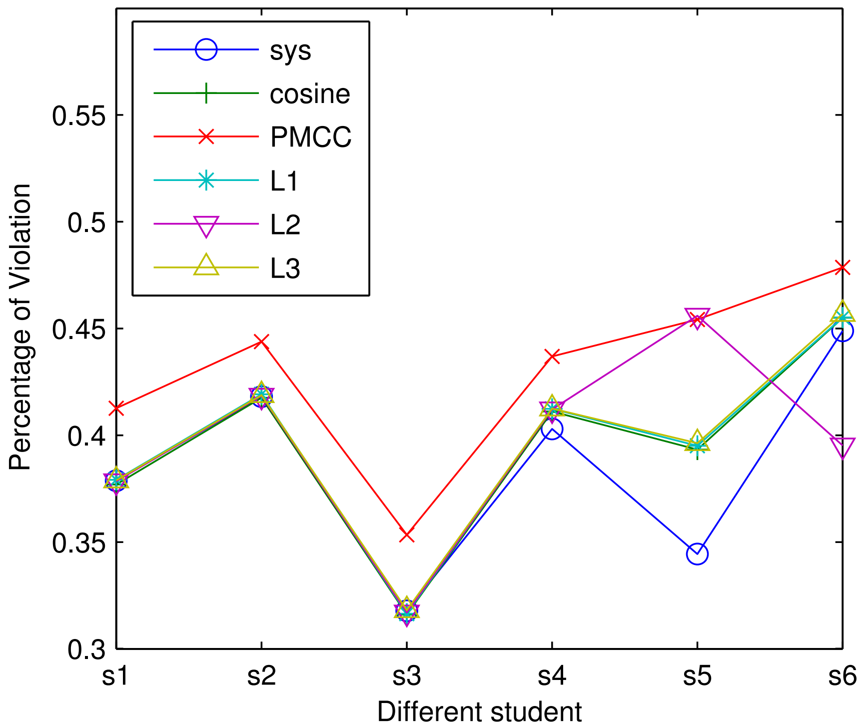


Figure 4.9 Percentage violations from different similarity functions

It shows the percentage violations of two heuristic rules for different rating values. In this figure, "*sys*" means the learned similarity function, "*cosine*" is the cosine similarity function, "*PMCC*" stands for the Pearson product moment correlation coefficient, and, finally, L1, L2 and L3 mean the *p*-norm distance between two sets of values, where *p* equals to one, two, and three.

Although the learned similarity function does not always improve the performance significantly, it performs best compared with other popular similarity metrics and it is the most suitable similarity function at the current stage.

The performance of recommending rating values on online reviews is shown in Table 4.7. As seen from this table, the system achieves a high performance in terms of four evaluation metrics. As for *Precision* and *Recall*, which are two classification metrics, they both achieve a higher than 0.9 results for the average of the six designers. In addition, for evaluations from the regression aspect, in terms of *MAE* and *RMSE*, results of less than 0.1 and 0.2 are gained.

Table 4.7 Performance of the six recommendations

Designer number	*Precision*	*Recall*	*MAE*	*RMSE*
S1	0.912	0.925	0.055	0.166
S2	0.915	0.913	0.080	0.186
S3	0.914	0.910	0.082	0.185
S4	0.912	0.907	0.082	0.190
S5	0.967	0.966	0.030	0.119
S6	0.904	0.900	0.090	0.198

For the comparison of both the classification-based algorithm and the regression-based algorithm, the performance of the recommendation approach is shown in Figure 4.10.

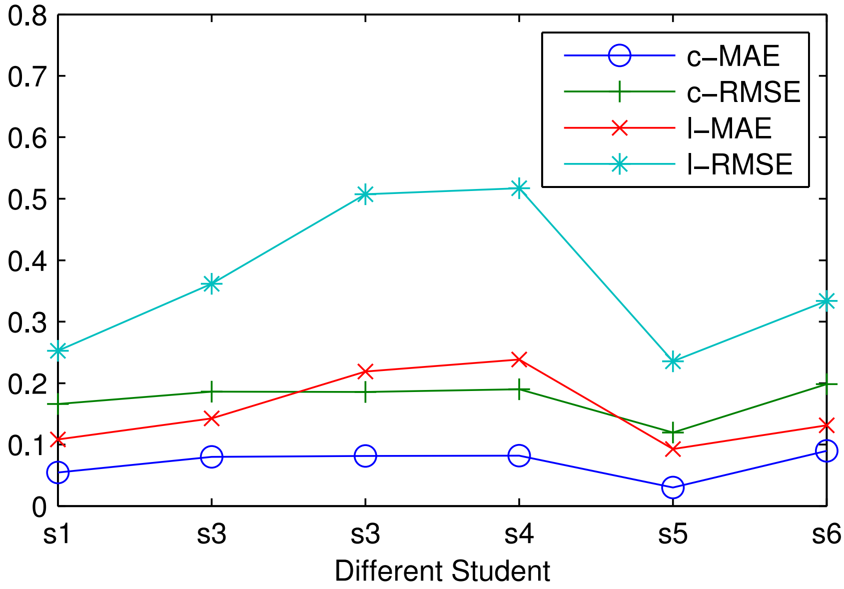
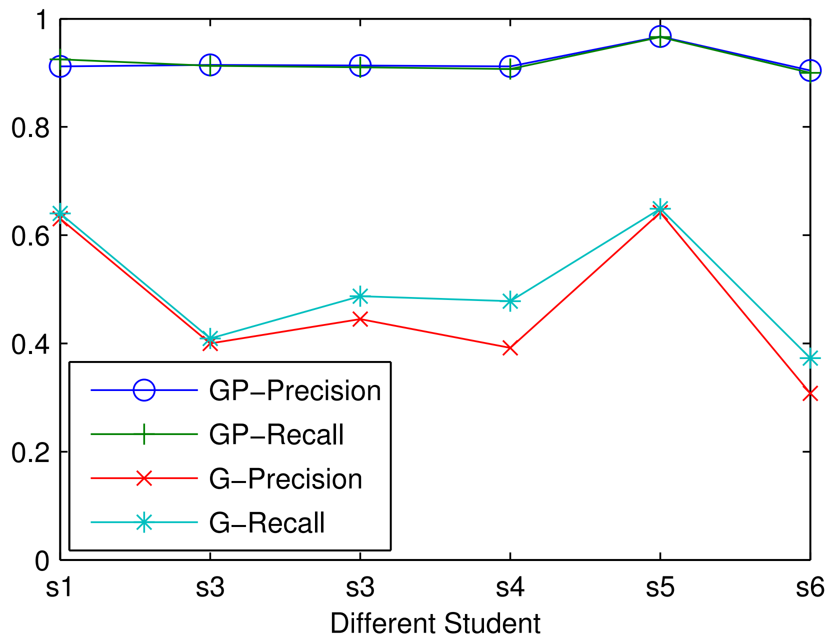


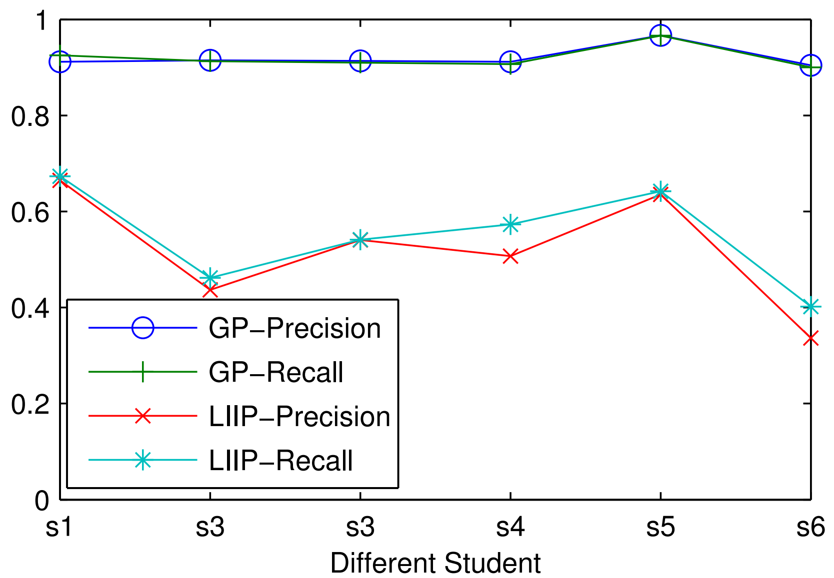
Figure 4.10 Classification vs. regression performance

The performance is evaluated in terms of *MAE* and *RMSE*. "*c-MAE*" and "*c-RMSE*" stand for the two metrics if the classification-based method is applied. "*l-MAE*" and "*l-RMSE*" are for the regression-based method. As seen from this figure, the classification-based algorithm performs constantly better than the regression-based method. It confirms that the classification-based algorithm is more appropriate to be applied here for the recommendation of rating values.

The performance comparison in terms of precision and recall is shown in Figure 4.11. "*GP-Precision*" and "*GP-Recall*" are the precision and recall using the proposed method for recommending rating values. "*G-Precision*" and "*G-Recall*" are the performance if three categories of domain-independent features are utilized only to recommend the rating values.



(a) The proposed algorithm (GP) vs. the generic helpfulness only algorithm (G)



(b) The proposed algorithm (GP) vs. the classification algorithm that directly utilizes four categories of features (LIIP)

Figure 4.11 The performance comparison on different algorithms

In Figure 4.11(a), the performance of the recommendation is shown to be boosted significantly by taking personal assessments into considerations. Different product designers are argued to have different opinion towards the importance of product features. Hence, product features are taken into consideration. A good performance is highly possible not to be gained by using domain-independent features only. It indicates that the recommendation of rating values for different designers can be improved by product features.

In Figure 4.11 (b), the performance of recommendation with different methods is compared. "*LIIP-Precision*" and "*LIIP-Recall*" stand for the evaluation metrics that the four categories of features are combined directly into a classification, such as *C-SVM*, to make the recommendation. "*GP-Precision*" and "*GP-Recall*" stand for the evaluation metrics of the proposed recommendation approach.

As seen from this figure, a good performance will not be obtained if all four categories of features are combined in a single feature vector and a single classifier is used to make the recommendation. The underlying reason is that the proposed recommendation approach stress more on the product features. The hypothesis that the recommendation of rating values can be influenced by product features is validated in these experiments.

4.4 SUMMARY

In this chapter, how to identify design-preferred online reviews from the perspective of product designers is described. The key question about how online reviews are actually perceived and evaluated from the perspective of product designers is explored.

The first question in this chapter is how to predict the helpfulness of online reviews in the viewpoint of product designers. Based on the investigation about why some reviews are helpful, a feasible approach for the prediction of the helpfulness was built through a regression method. Different from existing efforts, in this research, the ratings of product designers are regarded as the golden criterion to evaluate the helpfulness. Four categories of features are extracted directly from the content of online reviews. The effectiveness of the proposed method is verified by categories of experiments. Closely associated with the helpfulness prediction, another interesting question is explored. It is whether the helpfulness of online reviews can be modeled as domain-free. Various feature selection methods are conducted to testify whether domain-dependent features are contributing to the helpfulness of online reviews. As seen from the experiments, the helpfulness of online reviews from the perspective of product designers can be evaluated by domain-independent features, without a significant loss.

The second question in this chapter is how to recommend rating values on online reviews for different product designers. A classification-based recommendation approach was developed in this chapter. Four categories of crucial features are applied to the proposed recommendation approach. This approach is to recommend rating values from both a generic aspect and a personal assessment aspect. From the generic aspect, the recommendation is evaluated by three categories of domain-independent features. A similarity function is learned from the linguistic features and a classifier committee is built for the information quality dimension and information quantity dimension. From the personal assessment aspect, the recommendation is evaluated by product features. The results from both aspects are consolidated with a classification algorithm to recommend rating values on online reviews. The effectiveness of the proposed approach was evaluated by categories of comparative experiments.

The proposed models, together with the promising results, verify the possibility of identifying design-preferred online reviews from the perspective of product designers. In the next chapter, the concentration will be on how to build a design-centered knowledge base from online reviews, in order to benefit designers to improve the current product models directly.

CHAPTER 5

BUILDING A DESIGN-CENTERED KNOWLEDGE BASE

5.1 INTRODUCTION

In the previous chapter, how to identify design-preferred online reviews from the perspective of product designers is described. Two questions were concerned, how to predict the helpfulness of online reviews in the viewpoint of product designers and how to recommend the rating values on online reviews for different product designers. The models presented in the previous chapter, as well as the encouraging experimental results, explore the possibility of identifying design-preferred online reviews from the perspective of product designers.

In this chapter, the focus is Problem Two, how to build a design-centered knowledge base from online reviews. As mentioned, the two questions in Problem Two are how to connect customer reviews with product engineering characteristics and how to prioritize product engineering characteristics based on online customer reviews. In the second exploratory case study, one keyword only is seen to be not able to describe a consumer topic. It triggers this research to develop a linguistic approach to connect online reviews with product engineering characteristics automatically. Moreover, the objective of review analysis is to facilitate designers to improve their products and, in market-driven product design, it is unreasonable to treat all the product engineering characteristics without bias. Hence, an approach to prioritize product engineering characteristics is derived from online reviews. The technical details for the two questions are illustrated in the corresponding sections.

5.2 PROBLEM DEFINITION

One critical procedure of product design is to "transform customer needs or demands into design quality... and ultimately to specific elements of the manufacturing process" [Aka90]. In market-driven product design, the focus of product specifications is usually on fulfilling customer needs. Taking customer information as input, design efforts are targeted at particular product engineering characteristics, which yield maximum benefits for customer satisfaction. Exactly, in QFD, product designers are required to connect customer needs with product engineering characteristics and generate the weightings of these product engineering characteristics. In order to clarify the problem to be explored in this chapter, some notation will be defined step by step.

More consumers prefer to share their opinions about products at e-commerce websites. In these websites, each product has a series of reviews, r_1, r_2, ..., r_p. For a specific review r_i, it may contain several sentences s_{i1}, s_{i2}, ..., s_{iq}. These online reviews contain valuable information about customer needs. According to the customer needs, designers only target at some of product engineering characteristics due to the time limitation and the budget cost. It is unreasonable to consider all of them without bias. Typically, a list of product engineering characteristics includes $PC = <pc_1, pc_2... pc_n>$. One example of this list can be found in Table 3.3.

In order to digest customer needs, product designers have to digest a large number of online reviews sentence by sentence. One review r_i is assumed to mention one or more product engineering characteristics. In this research, the keyword W_T is assumed to be the most important word that makes product designers to understand that a certain product engineering characteristic pc_k is implied in one review. For example, in the exploratory case study shown in Section 3.4, the designers read customer reviews and distinguish the keywords in each sentence to indicate which product engineering characteristics are referred to.

As mentioned in the exploratory case study, the same keywords are connected to different product engineering characteristics. In this research, product designers are assumed to understand the underline meaning of the keyword W_T through reading the context W_c. The context W_c refers to a text window which includes the left N_L words of W_T and the right N_R words of W_T.

However, it is almost impossible for a product designer to read, label and analyze all these online reviews manually. An intelligent linguistic approach is targeted to connect online reviews with product engineering characteristics automatically.

- (1) How to connect online customer reviews with product engineering characteristics

As seen from the second exploratory case study, some keywords are labeled with one product engineering characteristic, but in other reviews, they are also labeled with other different product engineering characteristics. In Section 3.4, an example of this phenomenon is presented. One is "...It obviously needs a more absorbent paper because..." and the other is "...Very easy to swap in alternate papers, always easy to see if paper is left..." In the first sentence, the word "paper" is utilized to refer to "Supported Paper", while the other sentence is labeled as "Hopper Unit".

The word "paper" can be utilized to refer to "Supported Paper" or "Hopper Unit" in different sentences.

In this research, a probabilistic model will be developed for this question. Suppose a keyword W_T, in context W_c, is labeled as product engineering characteristic pc_p, rather than pc_q. Hence, in context W_c, if the possibility that W_T is connected with pc_p is defined as $P(W_c, pc_p)$, $P(W_c, pc_p)$ is argued to be bigger than $P(W_c, pc_q)$. In other words, in context W_c, the possibility that W_T is connected with pc_p is bigger than the possibility that W_c connected with pc_q. Mathematically, it can be denoted as,

$$P(W_c, pc_p) \geq P(W_c, pc_q) \tag{5.2.1}$$

With the probability defined on the keywords and associated product engineering characteristics, the probabilistic model for the analysis of keywords will be explored. This model will be utilized to connect online reviews with product engineering characteristics automatically. The technical details will be described in Section 5.3.

Once online reviews are connected to product engineering characteristics, the question for designers to solve is how to prioritize these product engineering characteristics according to the consumer sentiments in online reviews.

Specially, in a particular review r_i, this customer may be satisfied or unsatisfied with the product engineering characteristic pc_j. This information can be denoted as (pc_j, O_{ij}). O_{ij} is the associated opinion about the product engineering characteristic pc_j in review r_i. Hence, accordingly, one review r_i can be represented as a product engineering characteristic-opinion pair vector, $<(pc_1, O_{i1}), (pc_2, O_{i2}), ...,$

(pc_n, O_{in})>. For short, the vector can be represented as $O_i = <O_{i1}, O_{i2}, ..., O_{in}>$. A positive O_{ij} denotes that, in review r_i, the consumer is satisfied with pc_j, while, a negative value denotes that the consumer is unsatisfied with pc_j. Also, for each product engineering characteristics, in review r_i, there may be zero, one or more review sentences associated with it. If the consumer does not explicitly mention the product engineering characteristic pc_j in review r_i, in this research, the corresponding sentiment O_{ij} will be set to zero, which implies that this consumer is assumed to have a neutral sentiment for pc_j. For the case that there are more than one sentences discussing pc_j in review r_i, the average value of O_{ij} is taken as the consumer's final opinion on pc_j.

In addition, in these e-commerce websites, consumers are also encouraged to give an overall rating towards the overall satisfaction about the product. For example, in Figure 1.1, a four-star printer review is given by one consumer. In this research, the number of stars is assumed to be the overall satisfaction of one consumer. It can be denoted as "in review r_i, a rating cs_i is utilized to express the overall satisfaction about the product." cs_i usually ranges on an ordinal and discrete scale. This scale may be in the form of either an ordered set of numerical values (e.g., five to one "star") or an ordered set of non-numerical labels (e.g., Very Good, Good, Barely Acceptable, Poor, and Very Poor). The only difference between the two cases is that in the former the distances between consecutive scores are known, while this is not true in the latter.

According to the definition of product engineering characteristic-opinion pair vector O_i, together with the information about the overall satisfaction cs_i, review r_i

can be also denoted as (O_i, cs_i). Hence, for a customer review set, containing p reviews, it can be denoted as < (O_1, cs_1), (O_2, cs_2)... (O_p, cs_p)>. Now, based on the information about customer sentiments and the overall satisfaction, the central question is how to prioritize product engineering characteristics automatically.

- (2) How to prioritize product engineering characteristics based on customer reviews

Specially, it is about how to derive the weights W = <w_1, w_2, ..., w_n> for product engineering characteristics pc_1, pc_2, ..., pc_n from <(O_1, cs_1), (O_2, cs_2), ..., (O_p, cs_p)> by exploiting all p reviews. Here, the number of star is regarded as the customer satisfaction and the sentiment vector over different product engineering characteristics is regarded as feature vector. Mathematically, it can be described as:

$$cs_i = f(\sum_{j=1}^{n} w_j O_{ij}) \tag{5.2.2}$$

$f(x)$ is a function transforming the sum of weighted opinions on product engineering characteristics into the overall customer satisfaction.

It appears that merely a regression model is qualified to learn w_1, w_2... w_n. However, it is arguable to practice regression models to analyze this question. Regression models are utilized to analyze those questions with continuous values as the target, while cs_i is a discrete value, either in an ordinal discrete scale or an ordered non-numerical label.

The classification model might be more persuasive than the regression model. Even so, it is still questionable whether it is plainly formulated by a simple classification model or not. The inherent ranking information of cs_i will be neglected by simple classification models, since, no matter whether the discrete scale or non-numerical label is applied, the rating is ordered or ranked. For instance, in one review, customer assigns a five-star. Suppose that the review is predicted as a four-star by model one and a three-star by model two. In this scenario, model one is favored, rather than model two, since the result is closer to the original five-star rating.

Another potential technique for this question is learning to rank. Learning to rank is a task to construct a ranking model automatically by using training data, so that new objects can be sorted according to their degrees of relevance or importance [Liu09]. Nonetheless, learning to rank models neglect that objects can be possibly placed in the same position. Hence, learning to rank techniques also can not be utilized directly in this question to prioritize product engineering characteristics.

Particularly, in this question, both the classification and the ranking information should be taken into consideration. Therefore, an ordinal classification model will be built for this question and the technical details will be presented in Section 5.4.

5.3 CONNECTING CUSTOMER REVIEWS WITH PRODUCT ENGINEERING CHARACTERISTICS

5.3.1 A Probabilistic Keywords Analysis Model

In this section, the probabilistic model for the analysis of keywords is proposed, which is used to connect online reviews with product engineering characteristics. In order to clarify the idea clearly, this probabilistic model will described from a concrete example step by step.

For instance, there is a sentence, "...A nice feature is that when you plug in a camera memory card in the front ..." After reading this sentence, designers distinguish the word "card" to mark that the "card slot" of the printer, which is a product engineering characteristic in Table 3.3, is implied here. However, as illustrated in the previous sections, the consumer topic in this sentence can not be described only by the word "card" because, in the other sentence, "card" might be utilized to refer to "consumable replacement", which is another product engineering characteristic in Table 3.3.

By intuition, some context words might be utilized to describe consumer topics. Hence, in this research, the context W_c is defined as the text window around the keyword W_T, which includes the left N_L words of W_T, the right N_R words of W_T, and the keyword W_T itself. In this example, if only two words are considered in both sides of "card", say, both N_L and N_R equals to two, the context W_c will include five words, "camera", "memory", "card", "in,", "the".

On the other hand, in this context, the word "card" is labeled as "card slot", rather than "consumable replacement". Hence, in this context, the possibility that "card" is connected with "card slot" is bigger than the possibility that "card" connected with "consumable replacement". Following Equation 5.2.1, the above example can be denoted as

$$P(W_c, \text{“card slot”}) \geq P(W_c, \text{“consumable replacement”}) \qquad (5.3.3)$$

In addition, intuitively, not all the words in the context W_c affect designers equally when they analyze which product engineering characteristics are the keyword "card" referring to. It is assumed that the word closer to "card" should have a greater impact on the decision. The impact tends to be weaker for a relatively farther word. For example, compared with "camera" and "memory" which are near the keyword "card", two words "nice" and "feature" may have a relatively weaker impact for designers to understand consumers' concern "card slot". But notice that, some adverb words, such as, "only", "there", "very", frequently appear in customer reviews. These words are generally thought to have little effect for designers to connect online reviews with product engineering characteristics. Hence, after the stemming and the stop words removal on review sentences, these adverb words are chosen to be as effective words in both W_L and W_R.

If an impact function is available to be applied on review sentences, given a context, the consumer topic can be judged automatically. However, it is arguable to provide an arbitrary impact function which is only defined manually from the distance to the keyword. Hence, the question here is how to learn an impact function from labeled review sentences.

Generally, given a context W_c, if a keyword W_T is labeled as product engineering characteristic pc_p rather than product engineering characteristic pc_q, then $P(W_c, pc_p)$ is assumed to be bigger than $P(W_c, pc_q)$. The objective of this research is to build a parameter learning method for the impact function, which is able to estimate the weights of different words in context W_c.

If α, β, and γ are defined as the impact factor parameters for the left words of W_T in the context W_c, the right words, and W_T itself, respectively, then, according to Bayesian rules, the possibility of W_c connected with pc_p, $P(W_c, pc_p)$ can be equivalently derived as,

$$
\begin{aligned}
&P(W_c, pc_p) \\
&=P(pc_p)P(W_c \mid pc_p) \\
&=P(pc_p)P(W_L, W_R, W_T \mid pc_p) \\
&=P(pc_p)P(W_L \mid pc_p)^{\alpha}P(W_R \mid pc_p)^{\beta}P(W_T \mid pc_p)^{\gamma}
\end{aligned} \tag{5.3.4}
$$

$P(pc_p)$ can be interpreted as the probability that consumers are talking about a product engineering characteristic pc_p and, mathematically, it can be estimated by the percentage of training samples connecting with pc_p. Given the context that the keyword W_T is connected with the product engineering characteristic pc_p, $P(W_L \mid pc_p)$, $P(W_R \mid pc_p)$ and $P(W_T \mid pc_p)$, can be interpreted as, the occurrence possibilities for the left words of W_T, for the right words of W_T, and for W_T itself, respectively.

Accordingly, in the previous example, if $P(W_c, pc_p) > P(W_c, pc_q)$, it can be inferred that,

$$
\begin{aligned}
&P(W_c, pc_p) > P(W_c, pc_q) \\
&\sim \frac{P(pc_p)P(W_c \mid pc_p)}{P(pc_q)P(W_c \mid pc_q)} > 1 \\
&\sim \frac{P(pc_p)P(W_L \mid pc_p)^{\alpha}P(W_R \mid pc_p)^{\beta}P(W_T \mid pc_p)^{\gamma}}{P(pc_q)P(W_L \mid pc_q)^{\alpha}P(W_R \mid pc_q)^{\beta}P(W_T \mid pc_q)^{\gamma}} > 1 \\
&\sim \log\frac{P(pc_p)P(W_L \mid pc_p)^{\alpha}P(W_R \mid pc_p)^{\beta}P(W_T \mid pc_p)^{\gamma}}{P(pc_q)P(W_L \mid pc_q)^{\alpha}P(W_R \mid pc_q)^{\beta}P(W_T \mid pc_q)^{\gamma}} > 0 \\
&\sim \alpha\log\frac{P(W_L \mid pc_p)}{P(W_L \mid pc_q)} + \beta\log\frac{P(W_R \mid pc_p)}{P(W_R \mid pc_q)} + \gamma\log\frac{P(W_T \mid pc_p)}{P(W_T \mid pc_q)} + \log\frac{P(pc_p)}{P(pc_q)} > 0
\end{aligned} \tag{5.3.5}
$$

In Model (5.3.5), a computational model is derived, and, it intends to learn the impact factor α, β, and γ for the left words of W_T in the context W_c, the right words, and W_T. The computational model can be trained from the labeled customer reviews.

Exactly, $P(W_L \mid pc_p)$, $P(W_R \mid pc_p)$, $P(W_T \mid pc_p)$ and $P(pc_p)$ can be estimated from training data. By means of tuning up the impact factor α, β, and γ, the model is then expected to satisfy the inequality described in Model (5.3.5) for all the labeled customer reviews in training data. Once α, β, and γ are learned, given a context W_c, the probability that the keyword W_T is connected to a particular product engineering characteristic can be compared with the probability that W_T is connected to other product engineering characteristics. Finally, the customer topics, or, the connections from labeled keywords to product engineering characteristics can be predicted accordingly.

Actually, it is unnecessary to estimate $P(pc_p)$ from training data exactly, since only $\frac{P(pc_p)}{P(pc_q)}$ is required. It can be approximated as:

$$\log\frac{P(pc_p)}{P(pc_q)} = \log\frac{|pc_p|}{|pc_q|} \tag{5.3.6}$$

$|pc_p|$ is the count that consumers are talking about a product engineering characteristic pc_p in training data. Hence, the item $\log\frac{P(pc_p)}{P(pc_q)}$ contains no parameters and it is a determinant item, which can be derived from training data directly.

The objective here is that, by means of tuning up the impact factor α, β, and γ, the inequality described in Model (5.3.5) is expected to be satisfied for all the labeled customer reviews in training data. In other words, three parameter-dependent items,

$\alpha \log \frac{P(W_L \mid pc_p)}{P(W_L \mid pc_q)}$, $\beta \log \frac{P(W_R \mid pc_p)}{P(W_R \mid pc_q)}$, and $\gamma \log \frac{P(W_T \mid pc_p)}{P(W_T \mid pc_q)}$, according to training data, should be tuned as large as possible.

Accordingly, a function, *Ratio*, is defined for parameter-dependent items on training data:

$$Ratio(\alpha, \beta, \gamma) = \sum_{pc_p} \sum_{pc_q} \{\alpha \log \frac{P(W_L \mid pc_p)}{P(W_L \mid pc_q)} + \beta \log \frac{P(W_R \mid pc_p)}{P(W_R \mid pc_q)} + \gamma \log \frac{P(W_T \mid pc_p)}{P(W_T \mid pc_q)}\} \quad (5.3.7)$$

The question turns to how to tune α, β, and γ to maximize *Ratio* function,

$$max\ Ratio(\alpha, \beta, \gamma) \quad (5.3.8)$$

If *Ratio* function is maximized, the inequality in Model (5.3.5) can be satisfied by the labeled customer reviews in training data as many as possible. In other words, the optimal α, β, and γ define an optimal discriminant function that separates the most proper product engineering characteristic clearly from others.

The idea is actually borrowed from SVM, which proposes a marginal maximization approach. SVM insists on finding the maximum margin hyper-planes and leaving much room for the correct classification of the future data. Similarly, the objective of the model proposed here also intends to maximize the ratio between two product engineering characteristics. Specifically, the goal is to make W_T in the context W_c clearly and correctly being connected with pc_p.

Notice that, a normalization term is employed in the target function of SVM to avoid the parameters which define the hyper-planes being tuned to large. Likewise, the normalization terms should also be applied to the objective function to curb the overtraining phenomenon. Thus, in order to combine both *Ratio* and the normalization terms, a loss function is then defined as:

$$Loss(\alpha,\beta,\gamma) = -Ratio(\alpha,\beta,\gamma) + C_1\frac{\|\alpha\|^2}{2} + C_2\frac{\|\beta\|^2}{2} + C_0\frac{\|\gamma\|^2}{2} \quad (5.3.9)$$

$-Ratio(\alpha, \beta, \gamma)$ is applied to $Loss(\alpha, \beta, \gamma)$ since $max\ Ratio(\alpha, \beta, \gamma)$ is essentially equal to minimizing the minus one. The corresponding weights of the normalization terms are tuned by C_1, C_2 and C_0 for α, β, and γ. Note that γ is a scalar, while both α and β are vector parameters:

$$\begin{aligned}\alpha &= (\alpha_1, \alpha_2, ..., \alpha_{N_L})^T \\ \beta &= (\beta_1, \beta_2, ..., \beta_{N_R})^T\end{aligned} \quad (5.3.10)$$

In the previous discussion, the word which is closer to W_T is assumed to affect more on the judgment about the connection of the product engineering characteristic. Thus, given a word in the left side of W_T at distance i, the impact factor α_i of this word is expected to be larger than the word at distance $i+1$ with the impact factor α_{i+1}. Similarly, the impact factor β_j of the word in the right side of W_T at distance j should be bigger than β_{j+1}. These intuitive rules can be written as:

$$\begin{aligned}&\forall i \in [1, N_L - 1], \alpha_i > \alpha_{i+1} \\ &\forall j \in [1, N_R - 1], \beta_j > \beta_{j+1}\end{aligned} \quad (5.3.11)$$

Equally, two sets of constraints are mathematically denoted as Model (5.3.12), where the bold "**0**" denotes a zero vector:

$$\begin{aligned}&M \cdot \alpha \le \mathbf{0} \\ &M \cdot \beta \le \mathbf{0} \\ &M = \begin{bmatrix} -1 & 1 & 0 & ... & 0 & 0 \\ 0 & -1 & 1 & ... & 0 & 0 \\ 0 & 0 & 0 & ... & 0 & 0 \\ 0 & 0 & 0 & ... & -1 & 1 \end{bmatrix}\end{aligned} \quad (5.3.12)$$

Also, for all α_i, β_j, γ are the impact factors for the words to identify how a keyword is connected with a product engineering characteristic, and they are suggested to be nonnegative and less than or equal to one:

$$\begin{aligned} &\mathbf{0} \le \alpha \le \mathbf{1} \\ &\mathbf{0} \le \beta \le \mathbf{1} \\ &0 \le \gamma \le 1 \end{aligned} \tag{5.3.13}$$

The bold "**1**" denotes a vector composing of one in all the dimensions. Combining the loss function in Equation (5.3.9), the constraints in Equation (5.3.12) and the constraints in Equation (5.3.13), finally, the optimization model is,

$$\begin{aligned} &\min Loss(\alpha, \beta, \gamma) \\ s.t. \quad &M \cdot \alpha \le \mathbf{0} \\ &M \cdot \beta \le \mathbf{0} \\ &\mathbf{0} \le \alpha \le \mathbf{1} \\ &\mathbf{0} \le \beta \le \mathbf{1} \\ &0 \le \gamma \le 1 \end{aligned} \tag{5.3.14}$$

Next, the focus is on how to derive $P(W_L \mid pc_k)$ and $P(W_R \mid pc_k)$. Two models, a "unigram" model and a "bigram" model, are derived based on the *N*-gram method. An *N*-gram is a contiguous sequence of *N* items from a given sequence of text in the fields of statistical language modeling. The word "unigram" refers an *N*-gram of size one and the word "bigram" refers size two.

5.3.2 A Unigram Model

The unigram model is one of the most commonly used methods in information retrieval. In the unigram model, the probability of hitting an isolated word is calculated, without considering any influence from the words before or after the keywords. In this model, the probability of each word all depends on the word itself. An illustration of the unigram model for the left context words is shown here:

$$P(W_L \mid pc_k) = \prod_{i=1}^{N_L} P(W_{Li} \mid pc_k) \tag{5.3.15}$$

$P(W_{Li} \mid pc_k)$ is the occurrence possibilities for left words W_{Li}, given that the keyword W_T is connected with the product engineering characteristic, pc_k. $P(W_{Li} \mid pc_k)$ can be estimated as,

$$P(W_{Li} \mid pc_k) = \frac{c(W_{Li}, pc_k)}{\mid pc_k \mid} \tag{5.3.16}$$

$c(W_{Li}, pc_k)$ is the count that the word W_{Li} is mentioned in pc_k. $|pc_k|$ is the count that consumers are talking about pc_k in training data.

However, if $c(W_{Li}, pc_k)$ equals to zero, $P(W_{Li} \mid pc_k)$ will be zero accordingly, and, it will induce that $P(W_L \mid pc_k)$ also equals to zero. Finally, the model can not be utilized to predict the correct product engineering characteristic. In order to avoid the zero probability problem for $P(W_{Li} \mid pc_k)$, the Dirichlet Priors smoothing method is utilized. In the Dirichlet Priors smoothing method, the probability is parameterized with a prior probability base on the training data:

$$P(W_{Li} \mid pc_k) = \frac{c(W_{Li}, pc_k) + \mu P(W_{Li} \mid \boldsymbol{C})}{\mid pc_k \mid + \mu} \tag{5.3.17}$$

μ is a constant, which is utilized to tune the weight of the smoothing item. $P(W_{Li} \mid \boldsymbol{C})$ is the probability of word W_{Li} which occurs in training corpus $\boldsymbol{C}$.

Accordingly to Equation (5.3.17), $\alpha \log \frac{P(W_L \mid pc_p)}{P(W_L \mid pc_q)}$ and $\beta \log \frac{P(W_R \mid pc_p)}{P(W_R \mid pc_q)}$ can be written as:

$$\alpha \log \frac{P(W_L \mid pc_p)}{P(W_L \mid pc_q)} = \sum_{i=1}^{N_L} \alpha_i \log \frac{P(W_{Li} \mid pc_p)}{P(W_{Li} \mid pc_q)}$$
$$\beta \log \frac{P(W_R \mid pc_p)}{P(W_R \mid pc_q)} = \sum_{i=1}^{N_R} \beta_i \log \frac{P(W_{Ri} \mid pc_p)}{P(W_{Ri} \mid pc_q)} \quad (5.3.18)$$

The ratio function $Ratio_1(\alpha, \beta, \gamma)$ for the unigram model is then:

$$Ratio_1(\alpha,\beta,\gamma) = \sum_{pc_p}\sum_{pc_q}\{\sum_{i=1}^{N_L} \alpha_i \log \frac{P(W_{Li} \mid pc_p)}{P(W_{Li} \mid pc_q)} + \sum_{i=1}^{N_R} \beta_i \log \frac{P(W_{Ri} \mid pc_p)}{P(W_{Ri} \mid pc_q)} + \gamma \log \frac{P(W_T \mid pc_p)}{P(W_T \mid pc_q)}\} \quad (5.3.19)$$

α_i, β_i, γ are all scalars, and, thus, the corresponding loss function $Loss_1(\alpha, \beta, \gamma)$ for the unigram model can be defined as:

$$Loss_1(\alpha,\beta,\gamma) = -Ratio_1(\alpha,\beta,\gamma) + C_1 \frac{\sum_{i=1}^{N_L} \alpha_i^2}{2} + C_2 \frac{\sum_{i=1}^{N_R} \beta_i^2}{2} + C_0 \frac{\| \gamma \|^2}{2} \quad (5.3.20)$$

Then, the optimization problem (5.3.14) turns to be a quadratic programming problem.

5.3.3 A Bigram Model

A bigram model is also a frequently used model in information retrieval. Rather than calculating the probability of hitting an isolated word, the bigram considers the influence from the words before or after each word. In this model, the probability of each word depends on its own word and the nearby word. An illustration of the bigram model for the left context words is shown here:

$$P(W_L \mid pc_k) = P(W_{L1} \mid pc_k) \prod_{i=1}^{N_L - 1} P(W_{Li+1} \mid W_{Li}, pc_k) \quad (5.3.21)$$

Similarly, in the bigram model, the Dirichlet Priors smoothing method is utilized to avoid the zero probability problem for $P(W_{Li+1} \mid W_{Li}, pc_k)$:

$$P(W_{Li+1} \mid W_{Li}, pc_k) = \frac{c(W_{Li+1}, W_{Li}, pc_k) + \mu P(W_{Li+1} \mid \boldsymbol{C})}{c(W_{Li}, pc_k) + \mu} \tag{5.3.22}$$

$c(W_{Li+1}, W_{Li}, pc_k)$ is the count that the word W_{Li} and the word W_{Li+1} are both mentioned in pc_k. $c(W_{Li}, pc_k)$ is the count that the word W_{Li} is mentioned in pc_k.

According to Equation 5.3.22, $\alpha \log \frac{P(W_L \mid pc_p)}{P(W_L \mid pc_q)}$ and $\beta \log \frac{P(W_R \mid pc_p)}{P(W_R \mid pc_q)}$ can be written as:

$$\begin{aligned} \alpha \log \frac{P(W_L \mid pc_p)}{P(W_L \mid pc_q)} &= \alpha_0 \log \frac{P(W_{L1} \mid pc_p)}{P(W_{L1} \mid pc_q)} + \sum_{i=1}^{N_L-1} \alpha_i \log \frac{P(W_{Li+1} \mid W_{Li}, pc_p)}{P(W_{Li+1} \mid W_{Li}, pc_q)} \\ \beta \log \frac{P(W_R \mid pc_p)}{P(W_R \mid pc_q)} &= \beta_0 \log \frac{P(W_{R1} \mid pc_p)}{P(W_{R1} \mid pc_q)} + \sum_{i=1}^{N_R-1} \beta_i \log \frac{P(W_{Ri+1} \mid W_{Ri}, pc_p)}{P(W_{Ri+1} \mid W_{Ri}, pc_q)} \end{aligned} \tag{5.3.23}$$

The ratio function $Ratio_2(\alpha, \beta, \gamma)$ for the bigram model is then:

$$\begin{aligned} Ratio_2(\alpha, \beta, \gamma) = \sum_{pc_p} \sum_{pc_q} \{ & \alpha_0 \log \frac{P(W_{L1} \mid pc_p)}{P(W_{L1} \mid pc_q)} + \beta_0 \log \frac{P(W_{R1} \mid pc_p)}{P(W_{R1} \mid pc_q)} + \\ & \sum_{i=1}^{N_L-1} \alpha_i \log \frac{P(W_{Li+1} \mid W_{Li}, pc_p)}{P(W_{Li+1} \mid W_{Li}, pc_q)} + \\ & \sum_{i=1}^{N_R-1} \beta_i \log \frac{P(W_{Ri+1} \mid W_{Ri}, pc_p)}{P(W_{Ri+1} \mid W_{Ri}, pc_q)} + \\ & \gamma \log \frac{P(W_T \mid pc_p)}{P(W_T \mid pc_q)} \} \end{aligned} \tag{5.3.24}$$

α_i, β_i, γ are all scalars, and the corresponding loss function $Loss_2(\alpha, \beta, \gamma)$ for the bigram model is:

$$Loss_2(\alpha, \beta, \gamma) = -Ratio_2(\alpha, \beta, \gamma) + C_1 \frac{\sum_{i=1}^{N_L} \alpha_i^2}{2} + C_2 \frac{\sum_{i=1}^{N_R} \beta_i^2}{2} + C_0 \frac{\| \gamma \|^2}{2} \tag{5.3.25}$$

The optimization problem (5.3.14) also turns to be a quadratic programming problem.

In summary, the whole algorithm can be described as Algorithm 1.

Algorithm 1: Impact factor learning Algorithm

$\alpha, \beta, \gamma \leftarrow$ {Random numbers between zero and one}
for Each keywords W_T connecting with multiple product engineering characteristics **do**
  $S \leftarrow$ {All review sentences labeled with W_T}
  $PC \leftarrow$ {All possible product engineering characteristics for W_T}
  for $S_i \in S$ **do**
    Stemming, stop words removal on S_i
    POS tagging S_i and words filtering with certain POS
    $W_L \leftarrow$ {Left N_L words of W_T }, $W_R \leftarrow$ {Right N_R words of W_T }
    $pc_p \leftarrow$ {the product engineering characteristic that W_T relates in S_i }
    for $pc_q \in PC$ **do**
      Calculate $\log\frac{P(W_L \mid pc_p)}{P(W_L \mid pc_q)}$, $\log\frac{P(W_R \mid pc_p)}{P(W_R \mid pc_q)}$, and $\gamma\log\frac{P(W_T \mid pc_p)}{P(W_T \mid pc_q)}$
      Calculate $\log\frac{P(pc_p)}{P(pc_q)}$
      Save these results in a vector V_t , and append V_t in a matrix MT
    end for
  end for
end for
Using MT as the training data, solve the optimization problem as described in Equation 5.3.14
return α, β, γ

The performances of both the unigram model and the bigram model will be compared in the next Section.

5.3.4 Experimental Study and Discussions

5.3.4.1 Experiment Setup

In this experimental study, the four sets of printer reviews, together with the labeled keywords and product engineering characteristics from the two annotators, which are introduced in Section 3.4, are utilized.

Two models, the unigram model and the bigram model, are proposed to connect online reviews with product engineering characteristics. The performance of the two models is compared.

Several parameters are seen in these two models. For example, there are, the number of context words on both sides (N_L and N_R), the constant that tunes up the weights for the smoothing item (μ) in Equation 5.3.17, and the weights for the normalization terms (C_1, C_2 and C_0). Thus, in order to analyze how the performance can be influenced by these parameters, different categories of experiments are conducted.

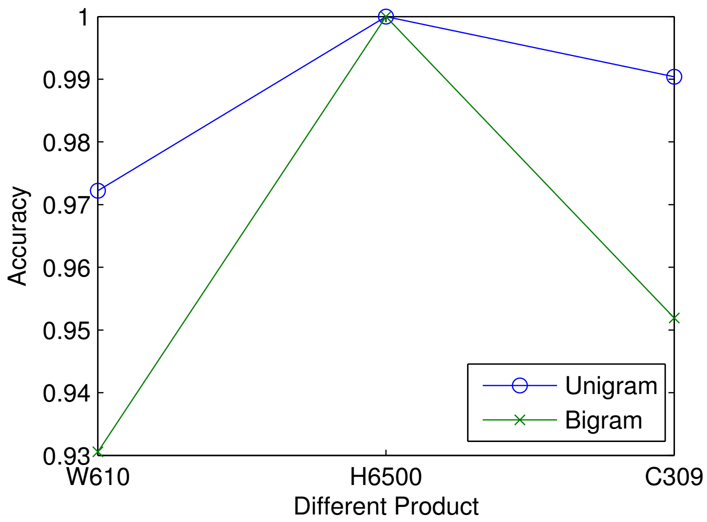
All programs were implemented and tested in Java 1.6 and Weka 3.6 on a dual core 2.40GHz personal computer with 4GB memory.

5.3.4.2 Results and Discussions

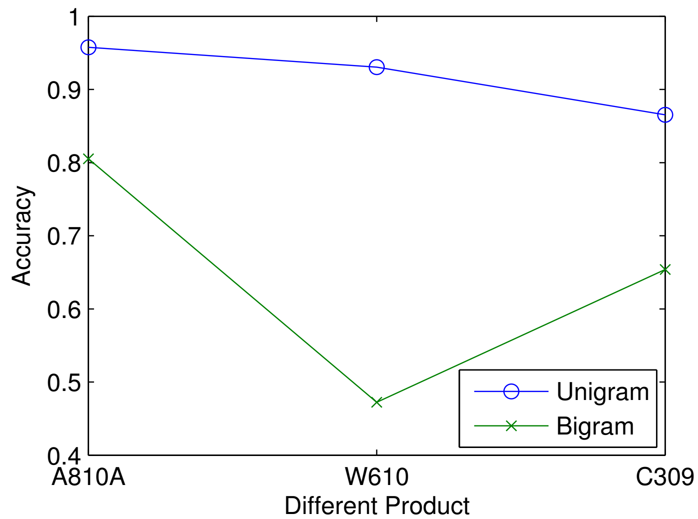
- Experiment 1: performance comparison on the unigram model and the bigram model

The performance of the unigram model and the bigram model is compared in Figure 5.1. Categories of experiments are conducted under the same parameter settings (μ equals to 500, C_1, C_2 and C_0 equal to 50, and N_L and N_R are equal to 25).

Notice that, if there are not sufficient words on either left hand side or right hand side the corresponding impact factors are set to zero. In Figure 5.1(a), the result is presented with 258 A810 reviews as the training data, and, in Figure 5.1(b), the result is with 210 HP6500 reviews as the training data.



(a) A810



(b) H6500

Figure 5.1 The unigram model and the bigram model

As shown in the two figures, the performance of the unigram model always outperforms that of the bigram model. Specially, in Figure 5.1(b), the performance of the unigram model shows much better than that of the bigram model with all reviews of three products as testing data. For the reviews of W610 as the testing data in Figure 5.1(b), about 50% correct results were obtained by the bigram model, while a higher than 90% correct results were obtained by the unigram model.

In Figure 5.2, the average accuracy of both the unigram model and the bigram model, in terms of different numbers of context words, is shown (μ equals to 500, C_1, C_2 and C_0 equal to 50). In these experiments, 256 A810 reviews are utilized as the training data and reviews of other three products are utilized as the testing data.

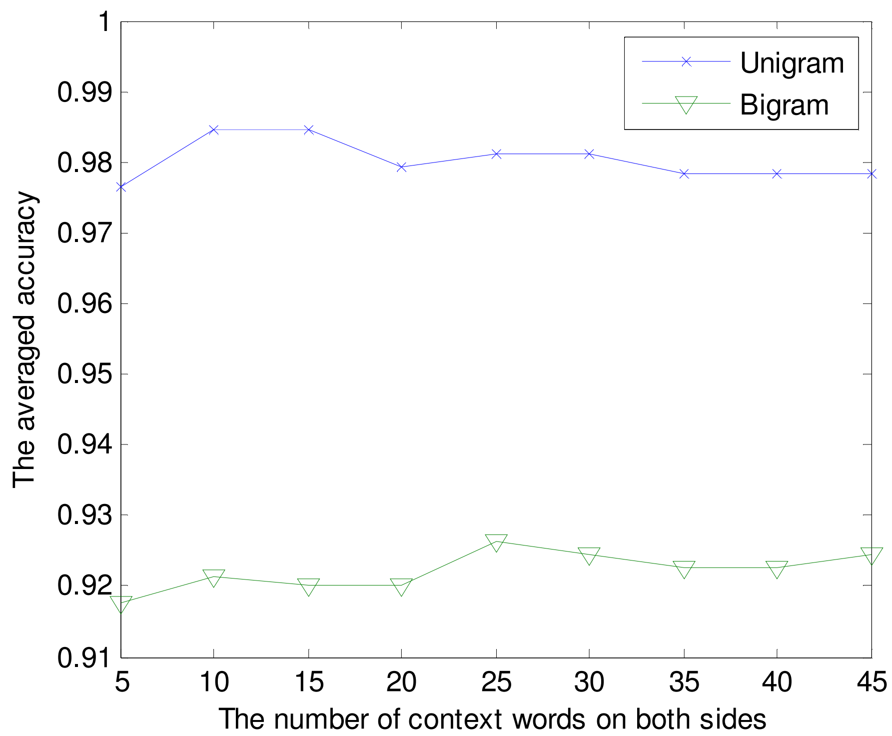


Figure 5.2 The average accuracy

First, the unigram model is shown to perform much better than the bigram model, which coincides with the phenomena in Figure 5.1. Although a well known experience is that the *N*-gram model might perform better with higher *N*, the models with larger *N* may involve much more computations. In addition, the success of a model with a higher *N* largely depends on how well it gets trained. In other words, in order to guarantee a good prediction result, it is necessary to prepare sufficient training data to train this model. In this research, only the words which are connected with different product engineering characteristics are utilized as the training data. However, as Table 3.4 shows, there are less than 40 words in each product reviews. The insufficient training might be one reason that the unigram model performs better than the bigram model, and thus, the unigram model alone will be applied in the following experiments.

Moreover, as seen from Figure 5.2, the performance of the models does not fluctuate much when more than 25 context words are chosen on both sides (N_L and N_R are equal to 25). Experiment 2 will make a further analysis towards the number of context words.

In Figure 5.3, the impact of different context words on both sides is illustrated, using A810 as the training data. In this figure, the impact α_i of the left i word of the keyword W_T, is denoted by the distance of $-i$. For example, "–5", in the horizontal line, denotes the left fifth word of the keyword W_T. Similarly, β_j, which illustrates the right j word of W_T, is denoted the distance of j. Finally, the impact of the keyword W_T, γ, is denoted by the distance of "0".

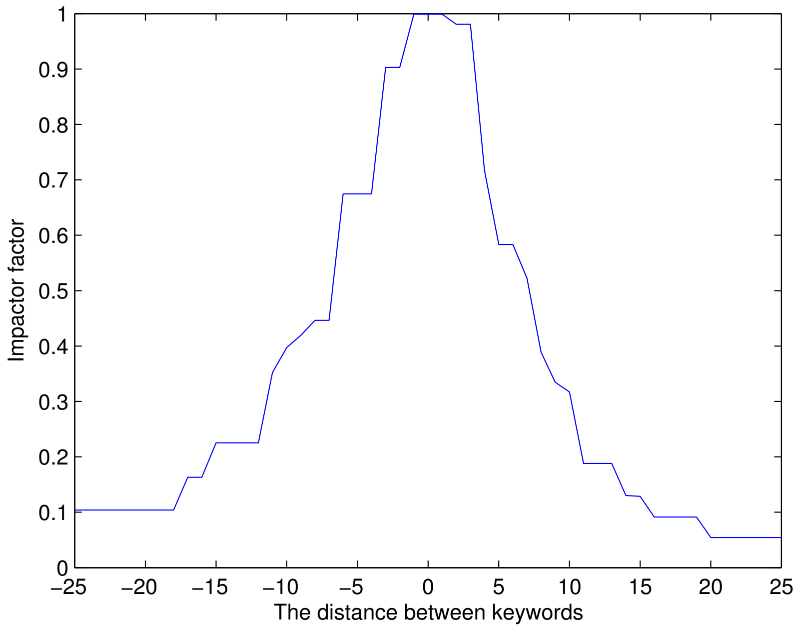
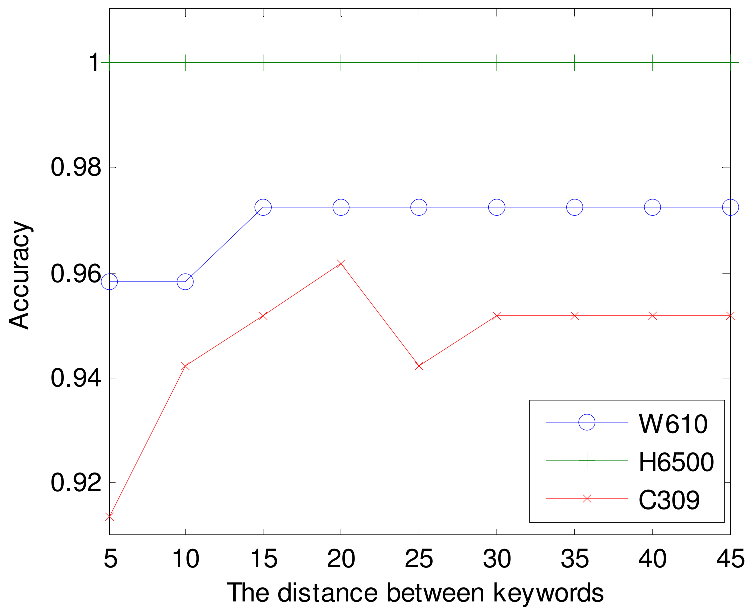


Figure 5.3 The impact factor

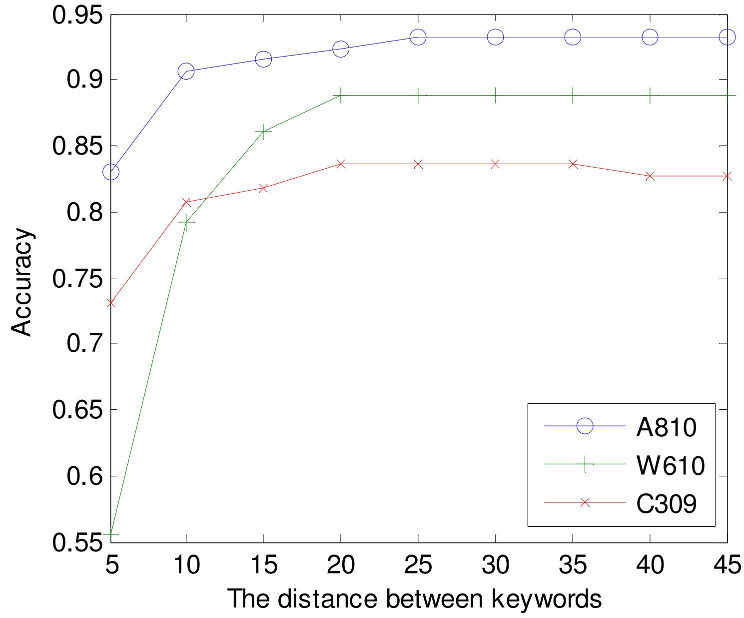
As seen from this figure, the impact of words on both sides is not symmetric and it declines significantly when the distance is small, say, less than 10 on both sides. However, the impact does not change much when the distance is bigger than 15 and they are observed to have only a little influence on predicting the connection to product engineering characteristics.

- Experiment 2: performance comparison on different numbers of context words

In Figure 5.4, the performance is compared based on different numbers of context words using A810 and HP6500 reviews as training data respectively. When only a small number of context words are chosen, the accuracy is relatively low. The accuracy is higher with more context words considered. It turns to be stable where the number of context words on each side is bigger than 25. The results presented in this figure coincide with the results shown in Figure 5.2, although the best results appear when the number of words is slightly less than 25 in both two figures. Therefore, in practice, about 25 context words are suggested to be chosen on each side, which means N_L = N_R = 25 may be sufficient to predict the meaning of the keyword W_T.



(a) A810



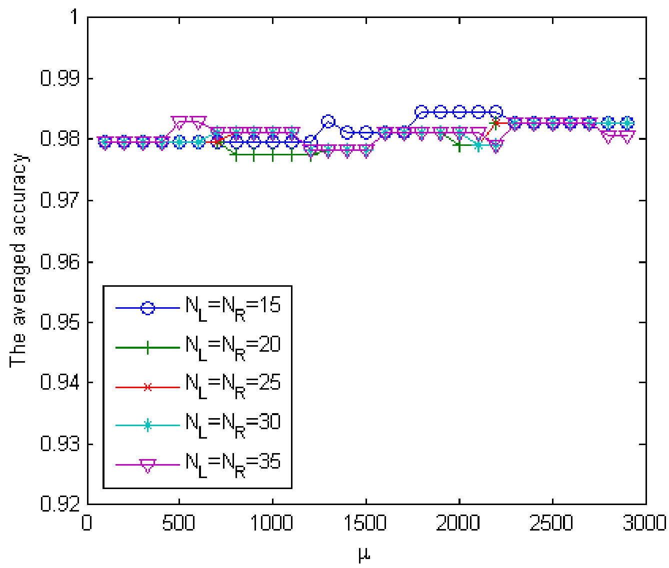
(b) HP6500

Figure 5.4 The performance comparison on different numbers of context words

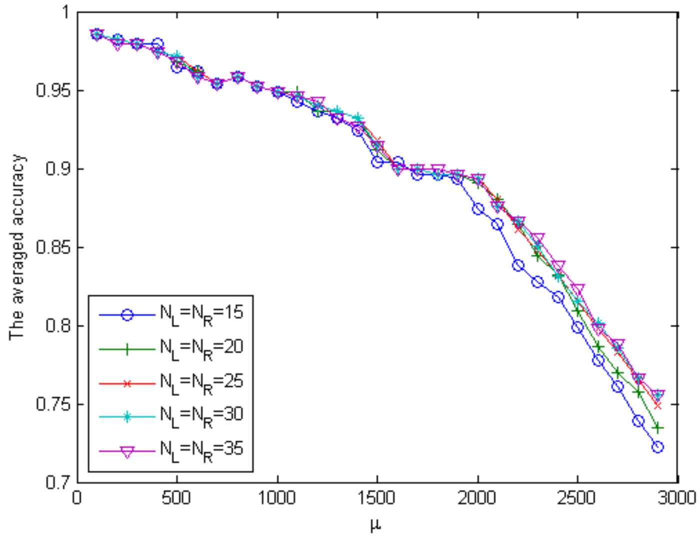
- Experiment 3: performance comparison on different μ

In Figure 5.5, the performance is compared in terms of different smoothing item μ in Equation 5.3.17. First, the average accuracy of three product reviews is shown in Figure 5.5(a), with A810 reviews are utilized as the training data. Also, some experiments with different numbers of context words (N_L and N_R) and different μ were conducted. As shown in Figure 5.5 (a), the average accuracy does not change much with different μ.

Similarly, some comparison experiments, utilizing HP6500 reviews as the training data, were performed. However, different phenomena were found in Figure 5.5 (b). The average accuracy is observed to drop gradually when μ is set to be larger than about 800. Similar trends can be found with different numbers of context words in this figure. Thus, a moderate μ is suggested to be chosen in practice. For example, in Experiment 1, μ is set to be 500.



(a) 810A



(b) H6500

Figure 5.5 The performance comparison on different μ

- Experiment 4: performance comparison on different weights of normalization terms

In Figure 5.6, the performance is compared with different weights of the normalization terms, C_1, C_2 and C_0. As stated in Section 5.3.1, the normalization terms are utilized to curb α, β, and γ from being tuned too large. Small weights of the normalization terms have little impact to control the above parameters α, β, and γ, and it tends to lead the over-fitting problem. But large weights might affect the performance of the proposed models.

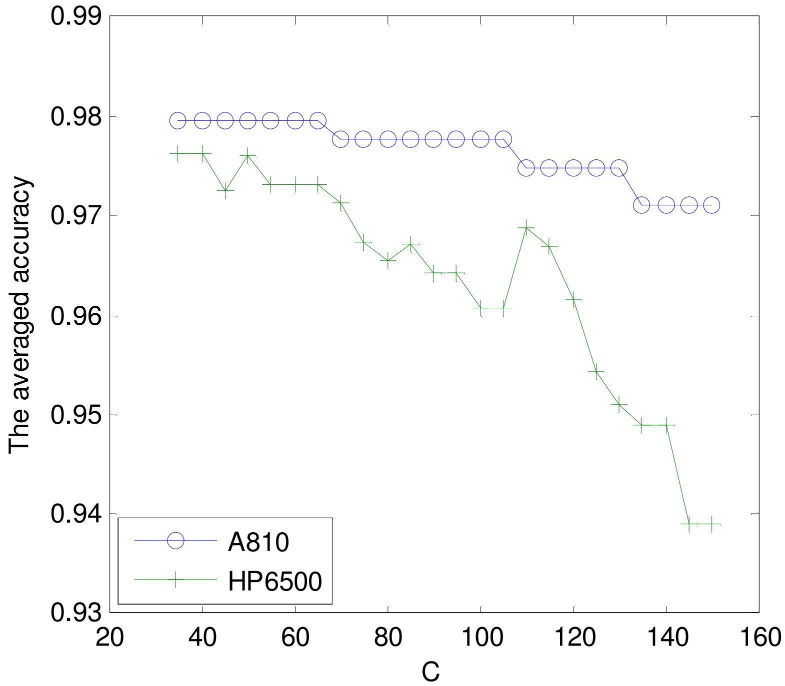


Figure 5.6 The performance comparison on different *C*

As seen in Figure 5.6, the average performance does not vary too much when the weights are smaller than 70 or 80, with either A810 or HP6500 being utilized as the training data. The average performance does not change too much when the weights are small. For example, when C is smaller than 60, the averaged accuracy is around 0.975 and 0.980, respectively. However, it begins to turn down gradually with a larger *C*. Specially, when the HP6500 reviews are applied as the training data, it drops dramatically if the weights are set to be larger than 100. Hence, the setting where C_1, C_2 and C_0 equal to 70 was employed in several experiments.

5.4 PRIORITIZING PRODUCT ENGINEERING CHARACTERISTICS BASED ON CUSTOMER REVIEWS

5.4.1 An Ordinal Classification Approach

In this chapter, another goal is placed on how to prioritize product engineering characteristics (*PC*) from online reviews. Exactly, it is how to learn the weight *W* of *PC* by exploiting the customer satisfaction on online reviews.

As presented in the previous section, several state-of-the-art techniques are not applicable for this question because both the discrete nature and the ranking information of the customer satisfaction should not be neglected. Thus, an ordinal classification approach is targeted for this question.

On the other hand, various learning to rank algorithms are proposed to tune the parameters of data with inherent ranking information. In these learning to rank algorithms, several of them are pairwise-based, such as RankSVM [Joa02] as well as several related methods [FLSS03, Joa06]. In the pairwise approach, the original distribution of the training examples *D* is expanded into a set of candidate pairs *P*, which include a set of document pairs. The learning procedure is conducted over this set of document pairs and, usually, outputs the values from $\{+1, -1\}$ to indicate the pairwise preference between each pair of documents. An example of the pairwise approach is shown in Figure 2.15. In RankSVM, the weight vector *W* of features is learnt from training data, which enables the distance between hyperplanes can be maximized. *W* is then utilized to predict the preference of the two documents in the testing data.

If different product engineering characteristics are regarded as features and the degree of customer satisfaction is regarded as the expected ranking position, the weights of features may be leant accordingly. But, in RankSVM, through the predicted preference, two documents are well separated. Thus, the visiual research gap is that, two documents can not be predicted to have the same preference. This is not true for the degree of customer satisfaction on different customer reviews since consumers may give the same rating to the product.

However, the idea of learning W and maximizing the distance between hyperplanes contributes in this research to solve this question. Hence, in this research, learning W of PC in the review space D is transformed to learn W in the review pair space P. But notice that the weight W refers to the weights of the associated opinions with the product engineering characteristics. Formally, the set of review pairs P is implied by review set D. P is the set of review pairs $((O_i, cs_i), (O_j, cs_j))$, which are drawn from review set D.

In order to illustrate the proposed classification approach clearly, a running example is introduced step by step. In this example, first, designers are assumed to concern six product engineering characteristics, pc_1, pc_2, ..., pc_6, which are regarded as important in product design. The corresponding weights of the six product engineering characteristics are needed to be estimated, w_1, w_2, ..., w_6. Also, there are nine related customer reviews, r_1, r_2, ..., r_9, collected from a commercial website for this product. Consumers can give a one-to-five star to a particular review to indicate their overall customer satisfaction for this product. It is denoted as $cs_i \in \{1, 2, 3, 4,$

5}. In Table 5.1, an example of cs_i for the nine reviews is shown, which denotes the overall customer satisfaction for the product.

Table 5.1 An example of the customer satisfaction of nine reviews

cs_1	cs_2	cs_3	cs_4	cs_5	cs_6	cs_7	cs_8	cs_9
5	2	3	4	5	2	4	1	3

In a particular review, different sentiments on different product engineering characteristics may be observed. The sentiments on different product engineering characteristics can be labeled manually by annotators, like what has been described in the second exploratory study in Chapter 3. Also, through several techniques of opinion mining or sentiment analysis, this information can be extracted automatically.

In this research, a five-degree metric, ranging from minus two to positive two, is utilized to evaluate the sentiment on product engineering characteristics in online reviews, where minus two stands for the least satisfied and the positive two stands for the most satisfied. It can be denoted as $O_{ij} \in \{-2, -1, 0, 1, 2\}$.

In this example, O_i =<O_{i1}, O_{i2}, ..., O_{i6}}>, is assumed to be the customer satisfaction over the six product engineering characteristics in review r_i. According to the settings, training examples in D can be denoted as:

$$(O_1, cs_1), (O_2, cs_2), ..., (O_9, cs_9) \in D \qquad (5.4.26)$$

According to the customer satisfaction information, the set of review pairs can be constructed. For two review r_i and r_j, if $cs_i > cs_j$, or equivalently, r_i is ranked better than r_j, ($O_i - O_j$, 1) is put into P. If $cs_i < cs_j$, or say, r_j is ranked better than r_i, then $(O_i - O_j, -1)$ is put into P. If cs_i is equal to cs_j, it means r_i is ranked equivalently to r_j, then $(O_i - O_j, 0)$ is put into P. So there are $\binom{9}{2} = \frac{9\times 8}{2} = 36$ review pairs:

$$((O_1, cs_1), (O_2, cs_2)), ((O_1, cs_1), (O_3, cs_3)), ...,$$
$$((O_2, cs_2), (O_3, cs_3)), ((O_2, cs_2), (O_4, cs_4)) ...,$$
$$...$$
$$((O_8, cs_8), (O_9, cs_9)) \tag{5.4.27}$$

For instance, according to the customer satisfaction information in Table 5.1, $cs_4 > cs_6$, then $(O_4 - O_6, 1)$ is put into P. $cs_4 < cs_5$, then $(O_4 - O_5, -1)$ is put into P. $cs_4 = cs_7$, then $(O_4 - O_7, 0)$ is put into P. Accordingly, all review pairs in P can be found in Table 5.2.

Table 5.2 Review pairs in P

$(O_1 - O_2, 1)$	$(O_1 - O_3, 1)$	$(O_1 - O_4, 1)$	$(O_1 - O_5, 0)$
$(O_1 - O_6, 1)$	$(O_1 - O_7, 1)$	$(O_1 - O_8, 1)$	$(O_1 - O_9, 1)$
$(O_2 - O_3, -1)$	$(O_2 - O_4, -1)$	$(O_2 - O_5, -1)$	$(O_2 - O_6, 0)$
$(O_2 - O_7, -1)$	$(O_2 - O_8, 1)$	$(O_2 - O_9, -1)$	$(O_3 - O_4, -1)$
$(O_3 - O_5, -1)$	$(O_3 - O_6, 1)$	$(O_3 - O_7, -1)$	$(O_3 - O_8, 1)$
$(O_3 - O_9, 0)$	$(O_4 - O_5, -1)$	$(O_4 - O_6, 1)$	$(O_4 - O_7, 0)$
$(O_4 - O_8, 1)$	$(O_4 - O_9, 1)$	$(O_5 - O_6, 1)$	$(O_5 - O_7, 1)$
$(O_5 - O_8, 1)$	$(O_5 - O_9, 1)$	$(O_6 - O_7, -1)$	$(O_6 - O_8, 1)$
$(O_6 - O_9, -1)$	$(O_7 - O_8, 1)$	$(O_7 - O_9, 1)$	$(O_8 - O_9, -1)$

With P defined, deriving the weight W for the associated opinion of product engineering characteristic turns to be a tri-classification problem. According the customer sentiments on different product engineering characteristics, it attempts to classify review pairs to $\{-1, 0, 1\}$.

Formally, one review pair in P can be denoted as (OP_k, cr_k), where OP_k equals to $O_i - O_j$. Here $O_i = <O_{i1}, O_{i2}, ..., O_{in}>$ and $<(pc_1, O_{i1}), (pc_2, O_{i2}), ..., (pc_n, O_{in})>$ is a product engineering characteristic-opinion pair vector for review r_i. O_i is the short form of the product engineering characteristic-opinion pair vector. Thus, OP_k can be represented as $<O_{i1} - O_{j1}, O_{i2} - O_{j2}, ..., O_{in} - O_{jn}>$ accordingly. It can be also denoted as $OP_k = <OP_{k1}, OP_{k2}, ..., OP_{kn}>$, where $OP_{ks} = O_{is} - O_{js}$. The corresponding cr_k, which represents the customer satisfaction relationship of review r_i and r_j, is a discrete value, where $cr_k \in \{-1, 0, 1\}$. Taking OP_k as the feature vector and cr_k as the target class, it is a tri-classification problem.

However, if a further step is taken, this question can be simplified to a binary classification. Notice that, when $cs_i < cs_j$, if the reverse subtraction of the opinion vector for r_i and r_j is taken, say, $(O_j - O_i, 1)$ is put into P, rather than $(O_i - O_j, -1)$, a binary classification will be shown. The step will not influence the value of W. But the problem is that the ranking relationship of two reviews is lost. Thus, in order to trace which review presents a higher degree of customer satisfaction, the information about the relationship of customer satisfaction $cs_i > cs_j$ or $cs_j > cs_i$ should be kept.

Now OP_k is either $O_i - O_j$ or $O_j - O_i$, and the customer satisfaction relationship of two reviews cr_k is "1" or "0". "1" denotes that two reviews do not rank equivalently or receive different labels of customer satisfaction (say, different numbers of stars), while "0" denotes two reviews should be predicted as receiving the same degree of customer satisfaction.

Taking the previous nine reviews as examples, accordingly, given the above transformation rules, all review pairs in *P* can be generated and they are shown in Table 5.3.

Table 5.3 Review pairs in *P* after applying transformation rules

$(O_1 - O_2, 1)$	$(O_1 - O_3, 1)$	$(O_1 - O_4, 1)$	$(O_1 - O_5, 0)$
$(O_1 - O_6, 1)$	$(O_1 - O_7, 1)$	$(O_1 - O_8, 1)$	$(O_1 - O_9, 1)$
$(O_3 - O_2, 1)$	$(O_4 - O_2, 1)$	$(O_5 - O_2, 1)$	$(O_2 - O_6, 0)$
$(O_7 - O_2, 1)$	$(O_2 - O_8, 1)$	$(O_9 - O_2, 1)$	$(O_4 - O_3, 1)$
$(O_5 - O_3, 1)$	$(O_3 - O_6, 1)$	$(O_7 - O_3, 1)$	$(O_3 - O_8, 1)$
$(O_3 - O_9, 0)$	$(O_5 - O_4, 1)$	$(O_4 - O_6, 1)$	$(O_4 - O_7, 0)$
$(O_4 - O_8, 1)$	$(O_4 - O_9, 1)$	$(O_5 - O_6, 1)$	$(O_5 - O_7, 1)$
$(O_5 - O_8, 1)$	$(\mathrm{O}_5 - O_9, 1)$	$(O_7 - O_6, 1)$	$(O_6 - O_8, 1)$
$(O_9 - O_6, 1)$	$(\mathrm{O}_7 - O_8, 1)$	$(O_7 - O_9, 1)$	$(O_9 - O_8, 1)$

Notice that, in either SVM or RankSVM, "–1" and "1" are utilized to denote two different classes. With "–1" and "1", the hyperplanes are defined clearly in SVM or RankSVM. Hence, for the sake of simplicity, rather than "0", "–1" is deployed to exemplify two instances which are assigned with the same label in the previous step. Now the class information for review pairs cr_k is either "–1" or "1", which can be denoted as $cr_\mathrm{k} \in \{-1, 1\}$.

Hence, the transformation rules from review set *D* to review pair set *P* is summarized as:

$$(O_i - O_j, 1) \rightarrow \mathrm{P}, \text{ if } cs_i > cs_j$$

$$(O_j - O_i, 1) \rightarrow \mathrm{P}, \text{ if } cs_i < cs_j$$

$$(O_i - O_j, -1) \rightarrow \mathrm{P}, \text{ if } cs_i = cs_j \tag{5.4.28}$$

According to these rules, the review pair set P from the nine reviews in D is shown in Table 5.4.

Table 5.4 The final review pairs in P

$(O_1 - O_2, 1)$	$(O_1 - O_3, 1)$	$(O_1 - O_4, 1)$	$(O_1 - O_5, -1)$
$(O_1 - O_6, 1)$	$(O_1 - O_7, 1)$	$(O_1 - O_8, 1)$	$(O_1 - O_9, 1)$
$(O_3 - O_2, 1)$	$(O_4 - O_2, 1)$	$(O_5 - O_2, 1)$	$(O_2 - O_6, -1)$
$(O_7 - O_2, 1)$	$(O_2 - O_8, 1)$	$(O_9 - O_2, 1)$	$(O_4 - O_3, 1)$
$(O_5 - O_3, 1)$	$(O_3 - O_6, 1)$	$(O_7 - O_3, 1)$	$(O_3 - O_8, 1)$
$(O_3 - O_9, -1)$	$(O_5 - O_4, 1)$	$(O_4 - O_6, 1)$	$(O_4 - O_7, -1)$
$(O_4 - O_8, 1)$	$(O_4 - O_9, 1)$	$(O_5 - O_6, 1)$	$(O_5 - O_7, 1)$
$(O_5 - O_8, 1)$	$(\mathrm{O}_5 - O_9, 1)$	$(O_7 - O_6, 1)$	$(O_6 - O_8, 1)$
$(O_9 - O_6, 1)$	$(\mathrm{O}_7 - O_8, 1)$	$(O_7 - O_9, 1)$	$(O_9 - O_8, 1)$

Now, the question turns to learn the weight W to classify review pairs in P into two classes ("–1" and "1"). It is quite similar to what has been done in SVM. In SVM, weight W is tuned to maximize the distance between the parallel hyperplanes. An example of SVM is illustrated in Figure 5.7.

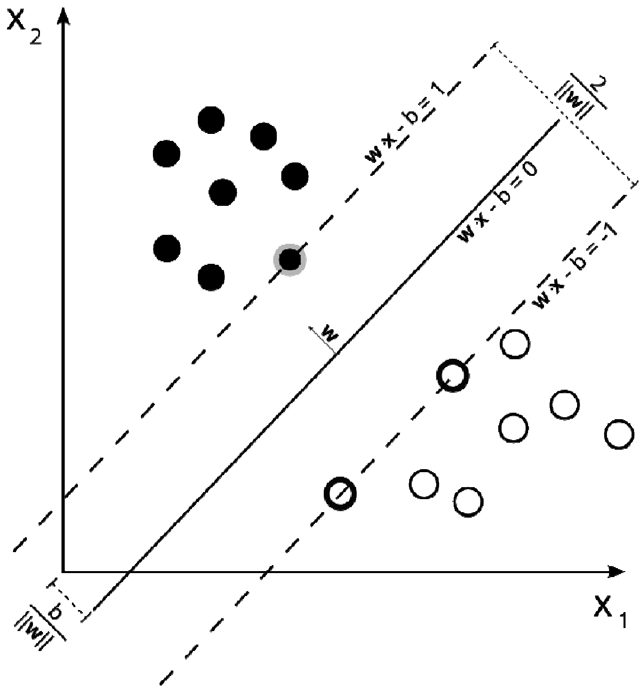


Figure 5.7 The linear separation of two classes with SVM

By borrowing the idea of margin maximization, an ordinal classification method based on SVM is devised to learn the weight *W* to classify review pairs into two classes.

The objective of SVM is to maximize the distance between hyper planes. But no ordinal information between different classes is considered in SVM. So in the first step, the ordinal classification problem is transformed into a binary classification. Moreover, in SVM, *W* denotes the weights of features and there are no additional constraints on *W*. But, in this question, *W* is defined as the priorities of the product engineering characteristic-associated opinion, or the weights of product engineering characteristics. It implies that *W* should be nonnegative since the sentiment polarities of different product features are considered. For example, "–2" means the most negative, while “2” illustrates the most positive. Hence, an additional constraint is added in this problem.

Also, if two classes cannot be separated by hyperplanes, a compromised idea of margin maximization is employed. A slack variable, ξ_k, is chosen in SVM. The slack variable is to measure the degree of compromise. This idea is also adopted to devise this ordinal classification method.

The complete model for this problem is shown as follows:

$$\begin{aligned} &\min \sum \xi_k + \frac{C}{2} W^T W \\ s.t.\ &\hat{cr}_k = \sum_{j=1}^{n} w_j OP_{kj} \\ &\hat{cr}_k cr_k \geq 1 - \xi_k \\ &W \geq \mathbf{0} \\ &\xi_k \geq 0 \end{aligned} \tag{5.4.29}$$

Model (5.4.29) furnishes the details about the method. ξ_k is the slack variable to estimate the degree of compromise. The coefficient C governs the relative importance of the regularization term compared with the sum of the degree of compromise term. $\frac{1}{2} W^T W$ is the regularization term, which is to control the over-fitting phenomenon. It is added to the error function in order to discourage the coefficients W from being tuned too large to induce the over-fitting phenomenon. The linear term, $\sum_{j=1}^{n} w_j OP_{kj}$, is to estimate the customer satisfaction relationship of two reviews $\hat{cr}_k$. Like SVM, the distance between two hyperplanes is still 2 – 2ξ, and the distance between the hyperplanes also intends to be made as large as possible, which makes the two classes to be easily discriminated. In the third constraint, the weight w_i of product engineering characteristic pc_i is restrained to be bigger than or equal to zero. Bold "**0**" denotes a zero vector, rather than a scalar zero.

Accordingly, the weights of the six product engineering characteristics, w_1, w_2, ..., w_6, for the nine reviews can be calculated by the optimization problem in Model (5.4.30):

$$\begin{aligned} & \min \sum_{k=1}^{36} \xi_k + \frac{C}{2} \sum_{t=1}^{6} W_t^2 \\ s.t.\ & \hat{cr}_k = \sum_{j=1}^{6} W_j OP_{kj} \\ & \hat{cr}_k cr_k \geq 1 - \xi_k \\ & W \geq \mathbf{0} \\ & \xi_k \geq 0 \end{aligned} \tag{5.4.30}$$

The proposed ordinal classification is a pairwise method. In this method, a pairwise comparison is made on the overall customer satisfaction. But notice that, although pairwise comparison results are obtained, perhaps this method can not be utilized directly in AHP. AHP makes a pairwise comparison, but AHP give multiple comparison values. The pairwise comparison results in this classification algorithm are only binary defined. More specifically, the relationship of the overall customer satisfaction is either equal or not. If this classification algorithm is utilized directly in AHP, the result will only tell which product engineering characteristics is more important. How to extend this classification method to make the pairwise comparison results be applied in AHP is one future work of this research.

5.4.2 Transforming the Results to the Original Customer Satisfaction Rating

The weights of product engineering characteristics *W* are learnt from the pairwise classification approach. At the same time, the customer satisfaction relationship of two reviews is predicted by a linear model. However, a new interesting question will come out if the pairwise approach is applied.

Suppose the proposed pairwise approach is expected to be evaluated by some classification metrics like *Precision*, *Recall* as well as *F-measure* and, possibly, it can be evaluated by ranking metrics like Mean Average Precision (*MAP*) and Normalized Discounted Cumulative Gain (*NDCG*). Hence, the predicted customer satisfaction for each review is required. However, due to various reasons, it is generally hard to train a classifier to separate well all data without any errors, and there are possibly some misclassified instances. In this pairwise approach, there must be also some misclassified review pairs. Thus, it is impossible to transform the pairwise-based results into the customer satisfaction rating faithfully. For example, there are three reviews, r_i, r_j, and r_k. The customer satisfaction relationship can be interpreted by Model (5.4.29) as $\hat{cs}_i > \hat{cs}_j$, $\hat{cs}_j > \hat{cs}_k$, and $\hat{cs}_k > \hat{cs}_i$. This result induces that the customer satisfaction for the three reviews can not be assigned to satisfy all the relationship. Hence, the new and interesting question then is how to assign the customer satisfaction for each review, in which the relationship is satisfied, or, in which the number of violations for the relationship is minimized.

In particular, there are two reviews r_i and r_j with $\hat{cs}_i > \hat{cs}_j$. The question is how to assign the customer satisfaction for these two reviews to satisfy the relationship. It can be mathematically formalized by Model (5.4.31):

$$\begin{aligned}
&\hat{cs}_i - \hat{cs}_j \geq 1 - M \cdot \alpha \\
&\hat{cs}_j - \hat{cs}_i \geq 1 - M \cdot (1-\alpha) \\
&\alpha \in \{0,1\} \\
&\hat{cs}_i, \hat{cs}_j \in \{1,2,3,4,5\}
\end{aligned} \tag{5.4.31}$$

In Model (5.4.31), α is either zero or one, denoting whether the relationship is satisfied or not by the assignment of customer satisfaction for r_i and r_j. If $\hat{cs}_i > \hat{cs}_j$ is satisfied, α equals to zero, otherwise α equals to one. M is a large number, for instance M equals to 10^3.

Likewise, if $\hat{cs}_i < \hat{cs}_j$, the equivalent model is as follows:

$$\begin{aligned}
&\hat{cs}_j - \hat{cs}_i \geq 1 - M \cdot \beta \\
&\hat{cs}_i - \hat{cs}_j \geq 1 - M \cdot (1-\beta) \\
&\beta \in \{0,1\} \\
&\hat{cs}_i, \hat{cs}_j \in \{1,2,3,4,5\}
\end{aligned} \tag{5.4.32}$$

If $\hat{cs}_i = \hat{cs}_j$, γ is symbolized whether the equation relationship is satisfied or not. The model is:

$$\begin{aligned}
&\hat{cs}_i - \hat{cs}_j \leq M \cdot \gamma \\
&\hat{cs}_j - \hat{cs}_i \leq M \cdot \gamma \\
&\gamma \in \{0,1\} \\
&\hat{cs}_i, \hat{cs}_j \in \{1,2,3,4,5\}
\end{aligned} \tag{5.4.33}$$

According to Model (5.4.31), Model (5.4.32), and Model (5.4.33), α, β, and γ are utilized to denote whether the corresponding relationship is satisfied or not. Hence, the sum of α, β, and γ represents the total number of the relation that are not dissatisfied by the assignment of customer satisfaction for all the review pairs. Hence, the question of transforming the pairwise-based results into the customer satisfaction rating faithfully turns to minimize the sum of α, β, and γ.

Combining Model (5.4.31), Model (5.4.32), and Model (5.4.33), the final Model to derive $\hat{c}s_i, \hat{c}s_j, \hat{c}s_k$ is:

$$
\begin{array}{llcl}
 & \min\{\sum \alpha_i + \sum \beta_j + \sum \gamma_k\} & & \\
s.t. & \hat{c}s_{ai} - \hat{c}s_{bi} & \geq & 1 - M \cdot \alpha_i \\
 & \hat{c}s_{bi} - \hat{c}s_{ai} & \geq & 1 - M \cdot (1 - \alpha_i) \\
 & \hat{c}s_{bj} - \hat{c}s_{aj} & \geq & 1 - M \cdot \beta_j \\
 & \hat{c}s_{aj} - \hat{c}s_{bj} & \geq & 1 - M \cdot (1 - \beta_j) \\
 & \hat{c}s_{ak} - \hat{c}s_{bk} & \leq & M \cdot \gamma \\
 & \hat{c}s_{bk} - \hat{c}s_{ak} & \leq & M \cdot \gamma \\
 & \alpha_i, \beta_j, \gamma_k & \in & \{0,1\} \\
 & \hat{c}s_{ai}, \hat{c}s_{bi}, \hat{c}s_{aj}, \hat{c}s_{bj}, \hat{c}s_{ak}, \hat{c}s_{bk} & \in & \{1,2,3,4,5\}
\end{array}
\tag{5.4.34}
$$

As seen from Model (5.4.34), it is an integer nonlinear programming optimization problem and it is solvable to obtain the optimal result.

Following the previous the example of nine reviews, there are *A* pairs for $\hat{c}s_i > \hat{c}s_j$, *B* pairs for $\hat{c}s_j > \hat{c}s_i$, and *C* pairs for $\hat{c}s_i = \hat{c}s_j$. Obviously, $A + B + C = 36$ (See Model (5.4.27)). Model (5.4.34) is then utilized to obtain the customer satisfaction for all nine reviews from the relationship of 36 review pairs. Specifically, the details of this procedure are illustrated in Model (5.4.35).

$$
\begin{array}{llcl}
 & \min\{\sum_{i=1}^{A} \alpha_i + \sum_{j=1}^{B} \beta_j + \sum_{k=1}^{C} \gamma_k\} & & \\
s.t. \forall i \in [1, A] & \hat{c}s_{ai} - \hat{c}s_{bi} & \geq & 1 - M \cdot \alpha_i \\
 & \hat{c}s_{bi} - \hat{c}s_{ai} & \geq & 1 - M \cdot (1 - \alpha_i) \\
\forall j \in [1, B] & \hat{c}s_{bj} - \hat{c}s_{aj} & \geq & 1 - M \cdot \beta_j \\
 & \hat{c}s_{aj} - \hat{c}s_{bj} & \geq & 1 - M \cdot (1 - \beta_j) \\
\forall k \in [1, C] & \hat{c}s_{ak} - \hat{c}s_{bk} & \leq & M \cdot \gamma \\
 & \hat{c}s_{bk} - \hat{c}s_{ak} & \leq & M \cdot \gamma \\
 & \alpha_i, \beta_j, \gamma_k & \in & \{0,1\} \\
 & \hat{c}s_{ai}, \hat{c}s_{bi}, \hat{c}s_{aj}, \hat{c}s_{bj}, \hat{c}s_{ak}, \hat{c}s_{bk} & \in & \{1,2,3,4,5\}
\end{array}
\tag{5.4.35}
$$

In Model (5.4.29) and Model (5.4.34), how to prioritize product engineering characteristics from online reviews and how to transform the pairwise-based results to the original customer satisfaction are shown. The performance of these methods will be discussed in the next section.

5.4.3 Experimental Study and Discussions

5.4.3.1 Experiment Setup

In Section 5.4.1 and Section 5.4.2, a pairwise classification approach was proposed to prioritize product engineering characteristics from online reviews, and how to transform the pairwise-based results into the original customer satisfaction rating for each review is discussed. In order to verify the performance of these methods, both classification-based and rank-based performance metrics are examined.

Precision, *Recall* and *F-measure* are usually employed to evaluate the performance in classification algorithms. The definition of *Precision* and *Recall* were given in Equation 4.3.23 and Equation 4.3.24. *F-measure* is a weighted average of the precision and recall. Here F_1 score is utilized, which is the harmonic mean of *Precision* and *Recall*. F_1 score reaches its best value at one and its worst score at zero.

$$F_1 = 2 \cdot \frac{|\, Precision \cdot Recall \,|}{|\, Precision + Recall \,|} \tag{5.4.36}$$

MAP and *NDCG* are popular evaluation metrics for ranking algorithms. *MAP* is a rank-based metric defined for relevant and non-relevant examples across a set of queries. It is based on the *P@n* metric and *AP*(*q*). *P@n* shows the precision achieved by considering only the top *n* examples in the ranked list. If there are r_n relevant documents in the top n examples, then $P@n = \frac{r_n}{n}$. $AP(q)$ averages the $P@n$ over possible values of n. Let r_q be the total number of relevant examples of query q, and $|Q|$ be the total number of examples in query q, and $r(n)$ be a function returning one if the n^{th} ranked example is relevant, and zero, otherwise.

$$\begin{aligned} AP(q) &= \frac{1}{r_q}\sum_{n=1}^{|Q|} P@n \cdot r(n) \\ MAP &= \frac{1}{|Q|}\sum_{q\in Q} AP(q) \end{aligned} \tag{5.4.37}$$

NDCG is to assess the quality of ranking when multiple levels of relevance are presented. The usefulness, or gain, of a document is measured based on its position in the result list. The gain is accumulated from the top of the result list to the bottom with the gain of each result discounted at lower ranks.

$$\begin{aligned} DCG &= \sum_{i=1}^{n} \frac{\log_2(1+i)}{2^{rel_i}-1} \\ NDCG &= \frac{DCG}{IDCG} \end{aligned} \tag{5.4.38}$$

rel_i is the graded relevance of the result at position i, such as not relevant, relevant, and extremely relevant. In this research, rel_i is the number of “star”, which stands for the degree of customer satisfaction, in the review at position i. *IDCG* is the normalization term of *DCG*, which ensures that the perfect *NDGC* score for the given set of examples is one. It means, in a faultless ranking algorithm, the *DCG* will be the same as the *IDCG*, which produces an *NDCG* of one.

All of the four printer datasets which were introduced in Section 3.4 were utilized as the experimental data. The programs were implemented and tested in Java 1.6 and Lingo 10.0 on a dual 2.40GHz personal computer.

5.4.3.2 Results and Discussions

The performance of Model (5.4.29) is illustrated in Figure 5.8. C is the regularization term that avoids the weights of product engineering characteristics, the parameters w_i, in Model (5.4.29) being tuned too large. As seen from Figure 5.8, except for the "W610" dataset, the predicted accuracy is all higher than 70% and, as expected, Figure 5.8 shows that the accuracy slopes down gradually with a higher C.

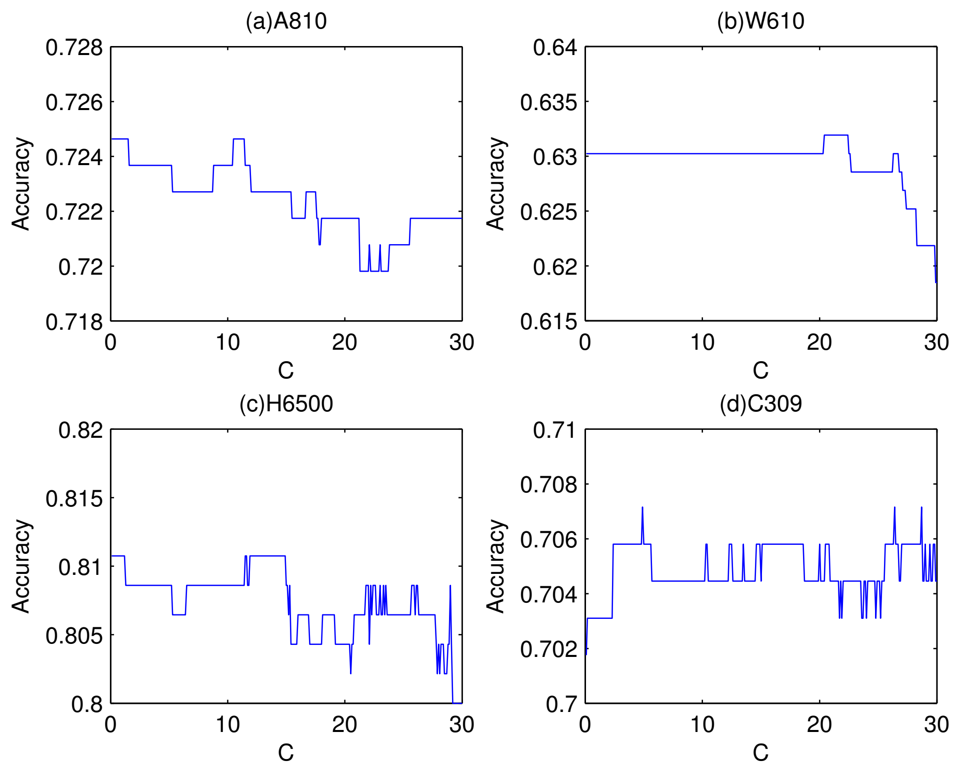


Figure 5.8 Accuracy vs. regularization term C

Another point is that the weight might be tuned too large if there is a proportion of zeros in some product engineering characteristics. In this research, a proportion of zeros illustrates that consumers do not express their sentiments or only leave a neutral opinion about the product engineering characteristics. However, it will induce that a higher weight will be tuned to these product engineering characteristics with zero value in Model (5.4.29). Nevertheless, it is unreasonable to suggest product designers to make more efforts on those engineering characteristics which receive little comments. Hence, in order to avoid this problem and obtain a relative fair priority for all product engineering characteristics, in all of the following experiments, product engineering characteristics which received less than 10% in product reviews were neglected.

The most important product engineering characteristics of the four printer datasets are listed in Table 5.5. In all these experiments, C equals to 15, where the performances are relatively stable in Figure 5.8. Compared with top frequent product engineering characteristics in Table 3.5, somewhat different yet interesting results are presented in Table 5.5.

Firstly, as seen from Table 5.5, those mentioned frequently product engineering characteristics in customer reviews are not necessarily predicted as important product engineering characteristics. For example, “Print Quality” is frequently discussed by consumers, according to Table 3.5. However, it does not appear in Table 5.5 in all of these four printer datasets. Generally speaking, “Print Quality” is a hot topic in printer reviews, but product designers do not necessarily pay more attentions on this product engineering characteristic when they launch a new printer. It illustrates that a high degree of “Print Quality” perhaps not necessarily lead to the same degree of customer satisfaction. From these experiments, when designers plan to design a new printer or improve the current model, which product engineering characteristics should be given more attention are suggested.

Table 5.5 Top five important product engineering characteristics

	A810	W610
1st	Ink Longevity	Fax Setting
2nd	Mac Compatible	Printing Speed
3rd	Wifi Integration	Wifi Integration
4th	Printing Speed	Ease of Use
5th	Ease of Use	Ease of Setup
	H6500	**C309**
1st	Ease of Setup	Ease of Setup
2nd	Ease of Use	Wifi Integration
3rd	Wifi Integration	LCD Panel
4th	Scan Software	Noise
5th	Printing Speed	Printing Speed

But it does not mean “Print Quality” is not important for printer design. Although “Print Quality” receives relative lower priority, generally, the high print quality is considered as a must when a printer is designed. It actually points to another relevant question, how to classify product engineering characteristics into different categories, such as, must-be, one dimensional and attractive attributes in Kano’s Model. This is one future work of this research.

Secondly, important product engineering characteristics may not be talked about by a large proportion of consumers. Take the “Ease of Use” in Table 5.5 as an example. This term appears three times in Table 5.5, but it does not appear in Table 3.5. It interprets that, although this item is not frequently mentioned by consumers, the overall customer satisfaction is impacted by this product engineering characteristic in a certain degree. Product designers need to pay more attention to improve the usability of printers, and the high degree of customer satisfaction depends on these product details. Another experiment will present more details to support this justification.

Thirdly, there are also some product engineering characteristics in both Table 5.5 and Table 3.5. For instance, “Wifi Integration” and “Ease of Setup” appear in the two tables. Admittedly, these two product engineering characteristics, especially “Wifi Integration”, are new creative product engineering characteristics for a printer. Without the “Wifi Integration”, a printer still can work very well. Similarly, with a little complex setting up for some amateur, a printer may still be a good product. However, with these creative product engineering characteristics, the user experience must be improved. These product engineering characteristics are preferred by many consumers, and there are a large number of comments on it. Customer satisfaction will be boosted with these novel product engineering characteristics.

Another objective is to explore what are the most important product engineering characteristics for those consumers who gave a five-star to the product, and whether these product engineering characteristics are aligned with the ones in Table 5.5. Thus, in this research, some similar experiments were conducted towards concerning five-star reviews only. The results are presented in Table 5.6. Compared with Table 3.5 and Table 5.5, Table 5.6 shows some different results.

Table 5.6 Top five important product engineering characteristics from five star reviews

	A810	W610
1st	Ink Longevity	Fax Setting
2nd	Auto Document Feeder	Wifi Integration
3rd	Consumable Replacement	Printing Speed
4th	Wifi Integration	Ease of Setup
5th	Ease of Use	Noise
	H6500	**C309**
1st	Mac Compatible	Ease of Setup
2nd	Ease of Use	Supplementary Software
3rd	Wifi Integration	Ink Longevity
4th	Printing Speed	LCD Panel
5th	Print Quality	Print Quality

As seen from Table 5.6, except for “Wifi Integration”, more product engineering characteristics are not frequently discussed by consumers. The important reasons for consumers to give a five-star rating are diversified, compared with the results shown in Table 5.5 when all the reviews are considered. For example, “Ink Longevity” is regarded as an important product engineering characteristic in “A810” and “C309” dataset which do not appear in Table 5.5 and Table 3.5. It illustrates that “Ink Longevity” is not frequently mentioned by many consumers and, generally, this product engineering characteristic is not regarded as an important one when designers conceive a new product.

Also, as seen from Table 5.6, different products present different advantages towards how to satisfy consumers to give a five-star rating. For instance, the "Fax Setting" in "W610" dataset is a decisive factor for consumers to make a five-star decision and, similarly, "Mac Compatible" is regarded as the most important product engineering characteristic in the "H6500" dataset. These phenomena further confirm that it is some details of the product that influence the overall customer satisfaction.

Finally, the performance of transforming the pairwise-based results into the original customer satisfaction rating with Model (5.4.34) is illustrated in Table 5.7. The final results are evaluated in terms of both classification (*Precision*, *Recall*, and F_1) and ranking metrics (*MAP* and *NDCG*). As seen from this table, a relative high performance is achieved by the ordinal classification algorithm in all of the four data sets. It proves the availability of the proposed ordinal classification approach.

Table 5.7 Performance of the ordinal classification algorithm

	Classification			Ranking	
	Precision	*Recall*	F_1	*MAP*	*NDCG*
A810	0.674	0.717	0.695	0.913	0.988
W610	0.600	0.629	0.614	0.889	0.975
H6500	0.710	0.742	0.725	0.953	0.988
C309	0.692	0.769	0.729	0.974	0.995

5.5 SUMMARY

In this chapter, how to build a design-centered knowledge base from online reviews is described. The key question about how online reviews are utilized in the viewpoint of product designers is explored.

The first question in this chapter is how to connect online reviews with product engineering characteristics. According to the second exploratory case study in Chapter 3, one keyword is not necessarily connected with the same product engineering characteristic. Thus, a one-to-one mapping can not be built between keywords in online reviews and product engineering characteristics. However, online reviews are expected to be connected with product engineering characteristics automatically to gain valuable consumer concern in the new product design. In this chapter, a probabilistic keywords analysis method was innovated for this concern. A unigram model and a bigram model were proposed to connect online reviews with product engineering characteristics. The impacts of keywords and context words were analyzed in the two models by an optimized weight-learning-method. Through categories of comparative experiments, the unigram model is shown to perform better than the bigram model.

The second question in this chapter is how to prioritize product engineering characteristics based on online reviews. Exactly, it is how to balance the weights of different product engineering characteristics according to the overall customer satisfaction of products and the customer sentiments of product engineering characteristics in online reviews. The weights of different product engineering characteristics can be utilized to facilitate designers when they conceive to improve the current product models. An ordinal classification approach was proposed for this concern. In this approach, a marginal maximization method was developed on review pairs by exploring the overall customer satisfaction of products and the customer sentiments of product engineering characteristics. Moreover, an integer nonlinear programming optimization model was advised to transform the pairwise-based results into the original customer satisfaction rating for each review, which many existing pairwise approaches do not explain clearly. Finally, the feasibility of the proposed approach was verified by several experiments.

In the previous chapter and this chapter, the technical details about how to identify design-preferred online reviews and how to build a design-centered knowledge base from online reviews were described, which are the focus of this research. In next chapter, the contributions and some prospects for future work will be discussed.

CHAPTER 6

CONCLUSIONS

6.1 ACHIEVEMENTS

Understanding the needs of the customers will help designers to conceive new products in today's market-driven product design. There are various ways to collect customer requirements. Conventionally, customer needs are obtained from customer survey or customer service data. These data are manually collected, sorted, and analyzed to gain customer needs for the new product design. However, it is time-consuming and labor-intensive to do this.

With the fast development of information technology, customers share their personal tastes and preferences by online reviews. Rich information about customer needs is provided in these online reviews. However, there exist a large number of online reviews and they are generated from time to time. It is impossible to be analyzed manually. As seen from the comprehensive literature reviews, online reviews are rarely seen to be utilized in the requirement analysis of product design, although they are widely accepted to be beneficial for product designers. Several relevant algorithms concerning online reviews which were proposed by researchers in computer science mainly focus on opinion mining, while those models developed by researchers in the product design area utilize customer survey data only. However, online reviews are fundamentally different from survey data. In this circumstance, an intelligent system is proposed to analyze a large number of online reviews for product design.

Before the technical details of the intelligent system are described, it is critical to understand how online reviews can be utilized from the perspective of product designers. Two exploratory case studies of review analysis in the customer-driven product design paradigm were conducted. The first case study is to explore how the review helpfulness perceived by designers. The second case study is to observe how online reviews can be utilized for the analysis of customer needs. Valuable data and interesting observations were obtained from the two exploratory case studies. All of these serve as the basis of this research to build sound models and algorithms for mining information from online reviews for product design.

In this research, two cohesive problems are investigated in the intelligent system. Problem One is how to identify design-preferred online reviews from the perspective of product designers. From the first exploratory case study, reasons about why some reviews are perceived as helpful are given by product designers. Accordingly, utilizing the four categories of features which are deemed as vital criteria for a helpful review, a regression model has been built to predict the helpfulness of online reviews. In addition, whether a model trained by customer reviews in one domain is able to be applied to reviews in the other domain is analyzed by several feature selection methods. Another similar yet different important question is then explored. It is how to recommend rating values on online reviews by taking personal assessments into consideration. Based on the four categories of features, both the generic aspect and the personal assessment aspect towards how to recommend rating values have been examined. Two aspects are then consolidated into an approach to recommend rating values on online reviews.

The two questions discussed in Problem One are, (1) how to build a collection of helpful online reviews from the perspective of product designers and (2) how to recommend rating values on online reviews for different designers.

Problem Two in this research is how to develop a design-centered knowledge base from online reviews. From the second exploratory case study, observations are that one keyword is not necessarily connected with the same product engineering characteristic, but online reviews are expected to be connected with product engineering characteristics automatically to gain valuable consumer concern in the new product design. According to some statistical information about online reviews, a probabilistic analysis model has been derived to estimate the contribution of keywords and different context words to product engineering characteristic judgments. This model is utilized to connect online reviews with product engineering characteristics automatically. Hence, the research effort turned to how to prioritize product engineering characteristics based on online customer reviews. An ordinal classification approach has been proposed for this concern. In this approach, a marginal maximization method has been developed on review pairs by exploring the overall customer satisfaction and the customer sentiments of product engineering characteristics. In addition, an integer nonlinear programming optimization model has been advised to transform the review pairs-based results into the customer satisfaction rating for each review.

The two questions discussed in this part are, (1) how to connect customer reviews with product engineering characteristics and (2) how to prioritize product engineering characteristics based on online customer reviews.

6.2 CONTRIBUTIONS

The ultimate aim of the proposed intelligent system is to facilitate designers to gain valuable information from online reviews and make this information to be utilized in product design. The focuses of this research are how to identify design-preferred online reviews from the perspective of product designers and how to build a design-centered knowledge base from online reviews. The major contributions of this research are several folds in two aspects.

(1) How to identify design-preferred online reviews from the perspective of product designers

How the helpfulness of online reviews is perceived by product designers has been investigated instead of defining a set of criteria on helpfulness arbitrarily, selecting representative reviews, and giving reviews to designers for rating. It is a user-centered method, in which an exploratory case study was conducted to understand how users actually behave.

According to this exploratory case study, four categories of features were then defined, which contributes to model the helpfulness of online reviews in the viewpoint of product designers. Secondly, the concern was explored towards whether the helpfulness of online reviews modeled in one domain can be successfully migrated to other domains. It is particularly desirable when the manually rated reviews are not steadily available for training in some particular domains.

Furthermore, a similar yet different question was described regarding how to recommend rating values on online reviews by considering the assessments of different designers. Two perspectives, namely, the generic aspect and the personal assessment, were combined to model this question. This question does not lose its practical significance, especially for designers with different requirements.

Compared with existing approaches, two intelligent approaches are proposed to analyze online reviews in terms of identifying customers' requirements. With the proposed method, customer requirements can be identified from the analysis of online reviews efficiently.

(2) How to build a design-centered knowledge base from online reviews

In order to understand how online reviews can be utilized by product designers, another exploratory case study was conducted. From the case study, online reviews were seen to connect product engineering characteristics, just like to build the association between customer needs and engineering characteristics in QFD. The annotated reviews are utilized to analyze the weights of different product engineering characteristics when designers conceive to improve current products.

In order to ease designers to annotate product engineering characteristics from online reviews entirely by hands, how to connect online reviews with product engineering characteristics was described. Through the analysis about the connection between online reviews with product engineering characteristics from the annotated data, the statistical information was extracted from online reviews. The statistical information, in this research, contributes to develop a probabilistic approach to connect online reviews with product engineering characteristics automatically.

Moreover, how to prioritize product engineering characteristics from online reviews in market-driven design was explored. The overall customer satisfaction of products and the customer sentiment of different product engineering characteristics were utilized from online reviews. Based on customer-related information, a pairwise model has been proposed to prioritize product engineering characteristics. In addition, an integer non-linear programming method has been built to transform the pairwise-based results into the original customer satisfaction rating for each review.

6.3 PROSPECTS FOR FUTURE WORK

One major limitation is that several proposed approaches in this research rely on, from exploratory case studies, supervised learning methods. In the two case studies, several annotators were hired either to label the helpfulness of online reviews or analyze the product engineering characteristics mentioned in online reviews. The annotated data cost several days to complete. For this reason, it is impossible to obtain a large number of training samples. Hence, a semi-supervised method may be more favorable, which can alleviate the burden of the corpus building.

In this research, only online reviews are taken into consideration. Fast development of information technology brings consumers novel chances to express their sentiments. Consumers also share their opinions through public discussion board, their personal blogs and social network websites. Booming size of tweets and microblogs talking about particular products are also found in various websites. Consumers share their experiences in different forms, no matter good or bad, which will unavoidably be seen by their friends or fans. Their sentiments will necessarily be strong recommendations or suggestions that will influence the decision of potential consumers. How to discover valuable information automatically from these channels for product designers will be certainly a hot research topic.

Another limitation is that the sentiments which are expressed in online reviews are regarded to reflect consumers' true opinion. However, some consumers are inactive online review writers. They may not prefer to waste their time to describe the advantages, and they only point out some drawbacks. Under this circumstance, the sentiment expressed in online reviews might not be aligned with their true opinion. Comparing with different sources of customer information, together with several conventional customer survey data, would be a good choice for designers to conduct customer analysis. How to combine different types of customer information and propose a holistic model for designers is a challenging question.

Some careful consumers may read several product reviews to identify the advantage and disadvantage about a particular product before their purchase. After that, these consumers also write their own judgments online. Their product reviews hence are greatly influenced by other reviews. How to utilize the similarity between helpful online reviews and how these online reviews influence the needs of potential consumers are also valuable for designers to understand different customers.

Another interesting example is that some positive reviews are possibly from product promoters. They submit several good words to describe their products in order to get a higher market share. On the one hand, it is important to make a rational justification towards how to recognize real consumers and extract their needs before make customer analysis in product design. On the other hand, however, the description of products in these reviews provides valuable chance for competitors to understand more about the product. Hence, how to identify the degree of overstate in these online reviews will also help competitors to make improvements of the product.

Also, in this research, customer reviews on different products are analyzed. However, reviews are focused on only one product in different models. Reviews of different products are not analyzed simultaneously. But, when QFD is utilized to make analysis for new product design, in practice, one important step is the benchmarking with competitors, which is also a hot research topic for researchers in the product design area to study about conventional customer survey data. How to make a comparison with different products from online reviews will definitely provide interesting results to product designers.

Four questions are described in this research and satisfactory results were obtained using online review data. The neglected point is whether these questions can be popularized to a general problem in the field of computer science or mathematics, with or maybe without domain specific knowledge. It is important in a way that several elaborate models will be derived from the solution of the general problem and these models contribute researchers and practitioners in different fields.

Ultimately, a fully automatic system should be developed. Although several approaches have been proposed for product designers and the feasibility of these approaches has been verified, they are not combined seamlessly into a fully automatic system. This fully automatic system is expected to be utilized directly by product designers. If this system can be utilized by product designers, more advices and suggestions to be utilized in product design from mining online reviews will be gained, which will definitely promote to develop more algorithms, models and applications.

APPENDIX I

QUESTIONNAIRES ABOUT WHY A REVIEW IS HELPFUL

QUESTIONNAIRE 1

Questionnaire

What are the reasons helping you to decide post No. 17 is helpful? Rank them from most important one to the least important one.

What are the reasons helping you to decide post No. 141 is helpful? Rank them from most important one to the least important one.

What are the reasons helping you to decide post No. 231 is helpful? Rank them from most important one to the least important one.

What are the reasons helping you to decide post No. 215 is NOT helpful? Rank them from most important one to the least important one.

What are the reasons helping you to decide post No. 936 is NOT helpful? Rank them from most important one to the least important one.

Reviews

No. 17

Have had the E71 for about a week and so far I love the phone. I really did my research on smartphones before I made the decision to go with the E71. Things I love about the phone:-Operating system is very easy to use and the phone is very fast. -Love the web browser, pages load quickly and clearly with vibrant color on both wifi and through ATT service -Love the voice activated features on the phone which I am still learning about. I really like how the phone can read your text messages to you and how the phone will say the name of the person calling so if I am driving and it's somebody I don't want or need to talk to I don't even have to pick up my phone. -The size of this phone is amazing with all of the features on this phone like the full

QWERTY keyboard and large screen that it is so small. My company uses Blackberrys or Motorola and in comparing the build quality, size, and features they don't really come close to the E71. The Blackberry phones are bulky and have a very unimpressive, cheap plastic build. With the slim dimensions of the E71 this phone fits in my pocket easier and more comfortably than my old Motorola C139. -Above all the call quality on the E71 is the best I have ever had in a cell phone and the sound quality is amazingly clear and reception is always good. I paired up my E71 with a Nokia bluetooth headset and even on Bluetooth and on the freeway people have commented that I don't even sounds like I am in a car with almost no background noise. I researched just about all smartphone brands for about 2 months before purchasing the E71 so I want to also give my opinions on some of the common ""dislikes"" or ""cons"" about this phone from the reviews I have read or seen and provide my opinions:- The phone is completely customizable which according to some reviews is good and bad. Bad if you are someone who isn't very ""tech savvy"" and good if you just want something like the IPHONE that is completely setup for you when you buy it. MY OPINION: I am not the most tech savvy person and with this being my first smartphone and scrolling through the different menus and setup screens I found the phone was very easy to setup the first day I got it. I like for example how there are multiple profiles you can setup for the E71 which include setups for ringtone, text tone, email tones, et cetera that you can program into a specific profile so if you are in a meeting and you need to silence your phone you can just switch to the ""meeting"" profile. With there being about 7 or 8 profiles you can setup, you can setup your phone to have a multitude of profiles depending on your setting. If you can't find an application or screen there is a handy help menu and you just simply type in what you are looking for and you get detailed instructions for either finding the application or setting it up. If you want to get back to the home screen, there is a home key that you click to get back to the home screen along with other handy keys next to the scroll key to get to commonly used screens quickly. Conclusion: The phone is very easy to navigate and setup. Non-standard 2.5mm headphone jack: This seems like more a minor annoyance than anything. I have had the phone about a week and used everything on it from calling, calendar, and MP3 player and I have not had any problems with this not having a standard 3.5mm jack. I don't see any problems here. Camera: The camera takes slightly above average photos in outdoor/sunlight but below average indoors with the flash with the 3.2MP camera. Bottom line, if you want camera quality on a cell phone comparable to your DSLR or point and shoot digital camera than you're going to spend a lot more $$ and have a phone much bulkier and thicker than the very compact E71. So in the end there is a tradeoff. If you want to also use a cell phone to take family pictures and other important events than you'd probably want to go with the Nokia N95 but again the N95 costs at least $100 more and 0.8"" thick (double the E71) so it costs more and isn't exactly pocketable like the E71. Fingerprints: The fingerprints show up on the back panel pretty easily and I consider this a minor annoyance. Nokia would have done better to make the entire backside of the phone the dark gray color as opposed to making the removable plate a chromed titanium. Summary: For the price and comparing this phone to other phones like the IPHONE and Blackberry Bold, the E71 doesn't have a couple of things the IPHONE does and EVERYTHING the Blackberry has for much cheaper than both. The E71 doesn't have the large touchscreen of the IPHONE but the IPHONE also doesn't have video recording like the E71. Considering the IPHONES and Blackberry Bold new are still $500+ and I was able to get the E71 for about $330, the E71 provides a much greater value.

No. 141

The n810 is a unit that takes time to learn to use well. The more time you spend researching it, the more it can do. Even now, over a month after receiving this, I know there's still more it's capable of. Being a stay-at-home type, I mostly find myself using this for midnight movies and web browsing, when I don't want to get out of bed and boot up my computer. It's also good during thunderstorms or when waiting in an office or restaurant, or when my desktop has other things to do. The n810 has the form factor of a hefty calculator, and has a refined feel to it - a third generation device that's had a lot of thought put into its design. Sadly, I suspect that Nokia isn't interested in continuing the internet tablet line, but prefers to stick with cell phones and their ongoing fees. (Edit: The n900 rover is nearing release. It sounds like a cross between an internet tablet and a mobile phone, priced like the latter.) Processor: 400Mhz Memory: 128MB DDR RAM Primary Storage: 256MB Internal Flash RAM Secondary Storage: 2GB Internal Flash Card, non-removable Tertiary Storage: Removable Mini or Micro SDHC Flash card, up to 32GB Battery: Nokia BP-4L Lithium-polymer (2-3 hours battery life at full usage) Display: 800x480, 16 bit color, 15:9 aspect ratio (odd size) OS: Maemo Linux CPU and memory: These are definitely the weak points of the n810. Oddly, that's a form of compliment. It means that Nokia has designed the n810 well enough that the available technology is the device's choke point. If I were to pick the first things to improve in the next generation, it would be this basic hardware. Data Storage: The 256MB of flash memory used as the device's primary storage seems to be a holdover from older designs. This is inconvenient because the applications are downloaded to and stored in this primary (and smallest) flash device. Those who are either very bold or knowledgable in Linux can reassign another device to be the n810's primary storage. However, I've yet to work up the courage to try. Different sources list the maximum size of the removable flash card as 8GB or 32GB, and users have reported that it is able to use the 16GB microSDHC cards. The 32GB microSDHC cards are not available yet. Ease of Use: The basic functions - music, web browsing, playing small videos, playing games, and downloading applications - are fairly easy to use. There's also a catalog of useful third-party applications online that are reasonably easy to use and install. However, if you want to get into more complicated things like third-party beta applications, converting videos for the n810, using the command-line interface, and partitioning your flash drives, then you hit a learning curve. On the negative side, explanations by Linux-users tend to assume you're fluent in Linux (I'm not). On the positive side, the n810 is -very- adaptable and rewards your efforts to understand it. Once you scratch the surface, it's more of a pocket-sized PC than a dedicated media player. Battery Life: Fully active, the original battery seems to have about 2-3 hours of life to it. I find myself recharging frequently, but at least the n810 can be active while it's charging. I've considered picking up a spare battery and charger, simply so I can go longer between recharges. It's interesting to note that turning the n810 on or off is also a big power-eater. Leaving it on in a low powered mode (touchscreen, wi-fi, and GPS unused) actually consumes less battery life. Although I haven't had the device long enough to encounter this, I've read that lithium batteries slowly age and deteriorate starting from the moment they're manufactured (this is in no way unique to the n810). Expect to replace the battery every few years. They cost about thirty dollars online. Installation is trivial: pop open the battery cover and swap batteries. Touchscreen: This thing picks up fingerprints very easily, so I tend to stick with the stylus. Items onscreen also tend to be a bit small for finger-tapping. Most of the time, the touchscreen is sensitive enough, but occasionally I have to press a bit harder than I'd like near the edges. Most of the time when the touchscreen seems insensitive, it's because the CPU is busy, but there's one or two spots I use consistently that aren't as responsive. Edit: Three months after purchase, the device is accumulating a small set of tiny scratch marks on the touchscreen surface. Even using the stylus is not proof against this. It's regrettable that Nokia used a soft plastic screen for this device.Keyboard: A bit small, but useful for two-thumb tapping. A significant improvement over virtual keyboards.Wi-fi: I haven't had any trouble with my home wi-fi network since I took a hammer to my old router and replaced it with a better quality second-hand router. The range seems to go out to somewhere in the backyard, so at least I can web-surf in the sun. The only wi-fi hotspot I tried out was at Borders, and that hit me up for a subscription of some sort so I left.Web Browsing: A bit slow, but acceptable and quite versatile. Flash and Youtube work, although it tends to choke a bit on the larger flash files. You can save files and images to the n810, just don't install or run any executables not specifically for Maemo Linux. Movies: The secret to quality movies on the n810 is finding a good video conversion program. It took several internet searches and trying three different programs, but I finally have one I'm almost completely satisfied with. The best resolution for playing videos on the n810 seems to be around 560x312 (or 520x312 if you want to crop to the native 15:9 aspect ratio). The media player included with the n810 only handles resolutions up to 352x288, despite the screen resolution of 800x480. The downloadable MPlayer app can handle resolutions up to 800x480, but any more than about half that and you can run into performance issues. It may seem a pain to have to convert movies myself, but experience suggests that all portable media players need their videos downsized for them. Doing it myself means I'm not at the mercy of someone else's selection and marketing schemes, but have full control over my own media library. Tip: MPlayer is controlled through the keyboard rather than the touchscreen. The keyboard commands are included in its instruction file. Music: Transferring files to the n810 is easy and no-fuss. Once the device is plugged into your computer, you can access its flash cards like any other storage device. Playback is simple. The hardest part is organizing the music files. The weak link here is the output device. Investing in a good set of headphones is recommended. Word Processing: I haven't gotten much out of this as a text-editor, mostly because the available text editing applications don't support Rich Text Files. I switched over to .rtf after Microsoft Office quit working in protest over my desktop upgrades. Bluetooth: I got a bluetooth dongle for my computer just to see what it can do. The dongle works, but bluetooth data transfer turned out to be slow and unreliable. The data transfer rate is about 4 MB/minute, and the connection is sometimes interrupted. Instant Messenger: I've managed to tie the n810 in to my MSN Instant Messenger account, so this is a go.Games: A while back, Nethack started working again as mysteriously as it stopped. Battle for Wesnoth is impressive, but the game itself is fairly difficult. Star Control 2 was a complicated, four part download that took a little familiarity with the linux mkdir and apt-get functions to pull off. The impressive thing about games available for the n810 is that they're full games, not just little applets you play for a few minutes and then get bored of. The difficulty of using games on the n810 lies with the tiny controls that make any real-time games awkward. With a download and an adapter, you can attatch a full-sized USB keyboard to the n810 using its USB host mode. Of course, the keyboards aren't nearly as portable, possibly excepting the plastic roll-up type. Built-in camera: It functions, but the images are rather poor quality, especially in low lighting. The camera is also oriented toward the person looking at the screen. Definitely not a substitute for a digital camera. Telephony: I really, really didn't care for Skype's terms of service, so I uninstalled this software rather than risk dealing with them. One of the main reasons I chose the n810 was to be relatively free of corporate paranoia (I have my own, thank you).Upgrades: Install Diablo operating system (tip: Attatch the n810 to your computer before running the update wizard.), Create a swap file (control panel -> memory), add a SDHC flash memory card (MiniSDHC or microSDHC with mini adapter), Partition the internal memory card and set it for use as the primary storage device, Load Applet (monitors CPU and memory usage, and allows you to kill running processes), Brightness and Volume fine adjustment. GPS: Untried.

No. 231

After looking through the many Droid reviews on Amazon, I saw a lot of talk about customizability but not enough examples. This review is intended to illustrate exactly why customizability is so important to many of us happy Droid owners. I also want to give people who are new to both smart phones and the Google experience an idea of what it was like to transition to that world. But mostly, I just want to add my perspective to the mix. Life Before Droid: I had a Verizon Moto Razr. Loved the light weight, hated the battery life (having to charge the phone every other day). Loved the voice and reception quality (at first, but reception deteriorated over the years). Hated the Verizon calendar and the lack of a useful home screen appointment reminder feature (and the fact that appointments would shift on their own by one hour--a bug Verizon never fixed). Hated the call reminder beep every five minutes (but refused to disable it because I needed that reminder). Liked notes but hated the limited notes features and most of all, hated the lack of backup of all my contacts and notes (My Verizon services came late to the game, and in any case I wanted better portability). Oh yeah--loved the voice dial. What I Wanted: My background is in IT and Windows, so I was a heavy user of Outlook. I wanted a ""Today"" screen. I also needed to be able to share emails with my work

Outlook, but not all of them. I did, however, want all of my personal emails to go to my phone. Here were my priorities:1) Better than average voice, speaker, and reception quality.2) Synch of all contacts data with an external, portable data storage area.3) A satisfying internet browsing experience.4) Emails and calendar with a Today screen.5) Better than average battery life for a smart phone.6) A decent notes app, a voice recorder, a password keeper, etc.7) A decent photo and video camera.What I Got: I tried the iPhone just long enough to determine that it failed my expectations for voice quality. I am a stickler that a smart phone is primarily a phone, so it better perform well as a phone! I also saw a friend's Palm Pre and the interface was annoyingly slow, so add snappy performance to the list. Finally bought the Droid. At first, I was annoyed by the physical keyboard. The weight didn't bother me at all (until I picked up a Droid Eris). The touch-lag on the screen was never an issue, as I found that the Droid was actually pretty snappy compared to other touch phones (until I compared it with an iPhone).With the Droid, I initially met all of the above objectives, except 6 and 7. That is, I would have to find the apps and customizations I needed and the photo camera was not great (videos looked ok, though).A Google calendar Today widget came already installed on the Droid. It only showed the next appointment, so I thought I might want to find a better widget from the app store (turns out later the stock widget was my preference after all). Clicking on the widget took me to my calendar or my task list, so I was pretty happy with it. After seeing how Blackberry phones were an absolute failure when it came to browsing, I was extremely happy with the Droid's Google Chrome browser. I know the phone is not as fast as the iPhone, but it has better overall speed and quality than just about anything else. The Droid display is second to none, and really adds to your browsing experience. In comparison with my friend's iPhone, you could easily read fine print on my display that you would have to zoom on his. The iPhone also had a strong blueish tint to its screen. My friend said my Droid had a yellowish tint, but it looks all white to me. ;-) Speaking of zooming, Droid handles automatic zooming and formatting of the text column very well (others say better than iPhone), and with Android v2.1, it has multi-touch. That's not to say there aren't some bonehead omissions on Google's part, such as the lack of zoom when reading emails. I hope they fix that soon. As for battery life... I had adjusted my expectations that I would need to plug-in the Droid every night and possibly during the day if I used some of its features (ahem, games) heavily. Setup: I knew from the reviews that synching with Outlook was problematic to impossible, so I opened a gmail account instead and setup my home Outlook client to download but not delete emails (my phone would be where I permanently deleted my mail). I then altered my email router settings (You don't use an email router service? You should!) so that my mail would transparently go to my gmail account instead of my ISP account. I exported all my personal Outlook contacts and imported them into Google. It was a piece of cake! I had used many Outlook folders before, so I had to find the equivalent in Google. Eventually, I managed to categorize all of my contacts. After backing-up my Outlook contacts, I then deleted them from Outlook and imported the ones from Google. It was seamless and, apart from some very minor data issues, proved to me that I could rely on Google as my main contact database and import them into Outlook whenever I needed to update my local computer. So now I was ready to setup my phone. After entering my gmail account in the phone, everything synched fine. All of my Google emails were pushed to the Droid and when I deleted them on the Droid, they were deleted on the Google database. All of my contacts, with thumbnail pictures, showed up on my phone. There, too, modifying a contact would propagate the change to the Google database, so the change was permanent. I tried forwarding an Outlook calendar appointment from my work computer to my Google account. The Droid received the email and I promptly deleted it. And within a few seconds I would see the appointment in my calendar and on my Today widget. Now I tried to setup the wireless interface. My wireless router uses encryption and I had a lot of problems getting everything working, but eventually I figured out that there is a flaw in my router when I use both wireless G/N (I have a D-Link router). I simply set it to G only and the Droid had no problem connecting. On the many public wireless sites, the Droid connects easily and performance is of course fantastic on wireless (although with a good 3G signal, performance is not too bad either).Life After Droid: Sometimes my Droid shows a koi pond as its wallpaper, with a realistic water animation--the rocks and koi waver and ripple as if it was a video. Other times it will show a stretch of wet sand at the beach, with a foamy wave that ripples up and down. It also randomly shows several other scenes that I found to work well with the animated water effect. This is thanks to an app I downloaded that runs on Android 2.1 [Waterpaper Live Wallpaper]. At one point however, I found the app to run a little choppy. And after adding Google Earth, I started to experience some serious home screen freezing and refresh delays of many seconds. But then I realized my Droid was starved for internal memory and was spending too much time clearing space to run the home screen. So I removed Google Earth and a few other apps, bringing my free internal memory up from 60MB to 100MB. That resolved the home issues and made everything perform snappier than ever. It really is too bad they don't store the apps on the SD card, as that would address this problem.When I want to voice-dial, I just hold down the camera button and it works (albeit a little slowly) the same as on my Moto Razr--but with deadly accuracy. The stock Droid was an epic fail when it came to voice dial, but I found a much more accurate voice dial app [Voice Dialer HF] and another app [Button Shortcut] to link any function to my camera button, and tadaaa!--one-click voice dial.When I miss a phone call or email, the LED light blinks (pink for phone call, green for email) and I get the notification ringtone alert once every 20 minutes for the first 2 hours, after which it shuts up. Niiiiice. This customizability was thanks to yet another app [Missed Reminder].I'm making a call to an acquaintance in China. As soon as I've dialed the number, the display assures me that the Google Voice service will be used for the international call instead of direct dial (which is the default action for all U.S. calls). I'm saving money and I didn't have to dial the Google Voice number first or use a special app interface to dial out--the feature worked transparently as soon as I configured the Google Voice app. And I don't ever worry about my phone going on standby during the call and having to unlock my screen just to get back to the dialer, even though I set the screen timeout to 30 secs. That's because the KeepScreen app prevents the timeout for any set of apps I specify (including the phone app).I'm stepping outside my car and need to quickly note the parking location in this huge parking garage. On my Droid, I just touch a widget and I'm instantly recording a voice note. The same app that does voice recording also keeps my text notes, drawings, checklists, and AES 128-bit encrypted passwords, in a tree-structured folder system. I highly recommend this app [Note Everything]! Oh, and I got used to the keyboard. ***Addendum 3/12 *** But I also downloaded a new keyboard app. It's a port of the HTC touch keyboard, and is available from several of the Android app library sites, but you may not find it in the official app store. It not only does calibration so you make fewer fat finger mistakes, but it allows voice-to-text input at any time in the middle of typing. In fact, I used it now to complete this paragraph. Actually, voice-to-text is now a standard keyboard feature in Android 2.1, but I still prefer this alternate keyboard. There's more--a lot more. I'll stop now. I love my Droid.

No. 215

I bought a Razr on Nov 30th; it quit on Dec 31st. They said it would have to be sent in for repairs that could take 12wks. I asked about buying a new phone they said $150+. DO NOT BUY A RAZR!!!

No. 936

I have bought for a friend of mine, i was so impressed with the phone. This a great phone. I'm a blackberry customer, but was very impressed with this phone

Responses

Student 1

What are the reasons helping you to decide post No. 17 is helpful? Rank them from most important one to the least important one.

The user had posted the features he likes most and he also shares his experience in using E71. Moreover, he compared some features with the other brands, such as Blackberrys or Motorola. It helps us to know what the user actually need or want. Through their comments, we know more about their habbits and understanding with the product. It's good for the product development in real market.

What are the reasons helping you to decide post No. 141 is helpful? Rank them from most important one to the least important one.

This user had written his feedbacks briefly, such as software, hardware, battery, keyboard and screen. He almost descibed all the features of the product which let product engineer know more about the voice of customer.

What are the reasons helping you to decide post No. 231 is helpful? Rank them from most important one to the least important one.

First of all, this user had shared what features he likes and dislikes of the phone. Secondly, he also shared the problems he faced when using the product and how he fix it. Thirdly, he shared some experience in using the product. These comments can help engineers to design or improve products which are more suitable for the customers.

What are the reasons helping you to decide post No. 215 is NOT helpful? Rank them from most important one to the least important one.

It's hard to understand the situation since it didn't show the complete case of the repair service.

What are the reasons helping you to decide post No. 936 is NOT helpful? Rank them from most important one to the least important one.

The user had only shared how they got the phone but he didn't state the reason why they love the phone.

Student 2

What are the reasons helping you to decide post No. 17 is helpful? Rank them from most important one to the least important one.

The customer has mentioned the good and bad of the product.

The customer has compared our product with the other product for some criteria for his opinion.

The customer has mentioned that what criterion is most important in his point of view.

What are the reasons helping you to decide post No. 141 is helpful? Rank them from most important one to the least important one.

The customer can point out the strength and weakness of the product.

Explain the good and the bad of the product clearly, such as how bad is the flash memory of the product, how good is the file transferring.

What are the reasons helping you to decide post No. 231 is helpful? Rank them from most important one to the least important one.

The customer has mentioned what criteria of the product he is most considered.

The customer has mentioned what features of the product is greater than the competitors.

What are the reasons helping you to decide post No. 215 is NOT helpful? Rank them from most important one to the least important one.

The customer did not mention any goof or bad about the product features.

What are the reasons helping you to decide post No. 936 is NOT helpful? Rank them from most important one to the least important one.

The customer only mentions that the phone is great, but he did not explain which parts or features of the phone are great. Therefore, we can not know which parts of features satisfied the customer.

Student 3

What are the reasons helping you to decide post No. 17 is helpful? Rank them from most important one to the least important one.

1. User did a research on smartphone before purchasing

2. Mention some cons and provide suggestion

3. User's standpoint is relatively neutral

What are the reasons helping you to decide post No. 141 is helpful? Rank them from most important one to the least important one.

1. 2-3 hrs battery life

2. 256MB of flash memory (should increase in next model)

3. Media player cannot support high resolution

4. Weak third-party support

What are the reasons helping you to decide post No. 231 is helpful? Rank them from most important one to the least important one.

1. Wifi connection problem

2. Droid was starved for internal memory

3. Synching with Outlook

What are the reasons helping you to decide post No. 215 is NOT helpful? Rank them from most important one to the least important one.

1. It is mainly a comment about maintenance department, instead of the phone itself.

What are the reasons helping you to decide post No. 936 is NOT helpful? Rank them from most important one to the least important one.

1. User mentions that everything is good, which cannot help in designing next model.

Student 4

What are the reasons helping you to decide post No. 17 is helpful? Rank them from most important one to the least important one.

Since the customer compares his E71 to Blackberrys and Motorola, what is the pros and cons of the phone, therefore I give this rank 2

What are the reasons helping you to decide post No. 141 is helpful? Rank them from most important one to the least important one.

Since the customer give us a brief feedback on the phone and show us what is good.

What are the reasons helping you to decide post No. 231 is helpful? Rank them from most important one to the least important one.

The customer gives us very detail comment and every single part of the phone which is very useful.

What are the reasons helping you to decide post No. 215 is NOT helpful? Rank them from most important one to the least important one.

This comment just talks about the price of the phone.

What are the reasons helping you to decide post No. 936 is NOT helpful? Rank them from most important one to the least important one.

The customer just buy the phone for his friend, the end user is not him.

Student 5

What are the reasons helping you to decide post No. 17 is helpful? Rank them from most important one to the least important one.

"My company uses blackberry & Motorola" (Can find out why companies are interested in them)

Bluetooth calling at speeds are good and clear (Since motorola is good in telecommunications, we can further develop there to improve call quality)

Camera Quality can be improved

3.5mm headset jack required

Meeting Profiles are important for commercial users (can further develop there)

What are the reasons helping you to decide post No. 141 is helpful? Rank them from most important one to the least important one.

"Unit takes time to learn "(meaning the OS is not user friendly)

Linux based (seems like customer do not like linux based systems)

Small keyboard

Camera can be improved

Larger resolution playback support (CPU must be revised in next version of phone)

Bluetooth transfer rate (can be improved)

What are the reasons helping you to decide post No. 231 is helpful? Rank them from most important one to the least important one.

****Outlook support (customer must do a lot and switch to gmail, also many companies are using outlook, if we can have support, we can gain a lot of new customers)*

Camera Quality

Phone not as fast as iphone (customers' thoughts)

Camera Button responsiveness

What are the reasons helping you to decide post No. 215 is NOT helpful? Rank them from most important one to the least important one.

There is no information about the phone's performance from the user.

What are the reasons helping you to decide post No. 936 is NOT helpful? Rank them from most important one to the least important one.

The customer bought the phone for his friend, and he was just impressed with it but no information about his usage experience.

Student 6

What are the reasons helping you to decide post No. 17 is helpful? Rank them from most important one to the least important one.

1. *Sound Feature*
2. *Reception Function*
3. *Software*
4. *Hardware*
5. *Outlook*
6. *Screen*
7. *Camera*

What are the reasons helping you to decide post No. 141 is helpful? Rank them from most important one to the least important one.

1. *Software*
2. *Reception Function*
3. *Speed*

4. *Storage*
5. *Hardware*
6. *Battery*

What are the reasons helping you to decide post No. 231 is helpful? Rank them from most important one to the least important one.

1. *Reception Function*
2. *Software*
3. *Storage*
4. *Sound Feature*
5. *Weight*
6. *Battery*
7. *Camera*
8. *Mail Feature*
9. *Text Feature*

What are the reasons helping you to decide post No. 215 is NOT helpful? Rank them from most important one to the least important one.

The comment is such a complain comment. The comment is only noted that the Customer does not repair the phone.

What are the reasons helping you to decide post No. 936 is NOT helpful? Rank them from most important one to the least important one.

The customer is only comment the personal view and does not contain any comment on the phone.

QUESTIONNAIRE 2

Student 1

Reviews

No. 145

With any phone, you have problems. Even with these aside, I very strongly reccomend the Droid Cons: The camera is pretty lousy If you are not in daylight, prepare to take several pictures and delete the worst ones. It just can't focus very well at all. Don't let the ""5 Megapixel"" slogan fool you. Megapixels don't make a camera a good camera. If you can't focus, then what's the point? Also, the flash is less than awesome. All of my pictures show a pretty obvious bright circle where the flash was. However, it's usable... and i don't take a lot of pictures anyways It is pretty buggy Sometimes, I'll get a text, and I'll go to it, and there will be no text message there. Then I'll exit and come back to messaging, and i'll see it. Once, i pressed messaging and it went to my Bank of America app. I constantly get email notifications and there'll be no email there. Once, it just started resetting every time i connected it to my computer. However, I just did a factory reset and it worked fine, making me think it was a bad app. The onscreen keyboard is way harder than an Iphone's I got this because it has a real keyboard, which is great. After 2 days or so, I can type just as fast as any other guy, and i use google docs on it a lot. Nothing wrong with the eral keyboard. However, the onscreen keyboard, at least in verticle mode (why would you use it in horizontal mode if the keyboard is right there?) is kinda hard for me. Maybe it is because i don't practice a lot, but i constantly hit the wrong button. This might be because the Droid screen, though longer, is a little thinner than the Iphone's. It is totally usable, don't get me wrong, but i can type twice as fast on an onscreen keyboard You need to do some work to make it be what you want it to be The music player on it is pretty lousy and so is the settings menu. It is tedious if you want to see the date. blah blah blahThat is kinda the fun part, though. There are awesome music players to download (for free). Tune wiki scrolls lyrics from the internet as you listen and mixzing finds album artwork from the internet. There are several different settings apps: a played with one that had different profiles for one-click settings as opposed to changing several individual settings. I downloaded Beautiful Widgets that displays the time and date really well and makes it really easy to switch to silent, wifi, gps, bluetooth, vibrate, etc. You have to (get to) make it your own. However, that's the kind of phone it is. Your phone is SO

DIFFERENT than everyone else's. If you have an hour of free time, you can try new things Not smudge resistant I wipe smudges away after every time i take it out. I don't have anything on the screen though The price for being different You can't get popular apps. I go to UT and can't set up wifi. However, Iphone owners have a app just for the UT wifi. Wow People always wanting to play with your phone Pros:Multitasking The other day, I used Google Docs, turn by turn GPS, a twitter App, AND Pandora at the same time. Awesome Custonizable It is fully customizable. It is not just a black screen with apps on it (like another phone i know about). You can put on your own background, download a different keyboard, work on different browsers, and literally change every aspect of your phone. Your DROID is like no one else's Droid. Maybe you can sleep with it next to you- it is a part of you Real Keyboard Awesome. In the beginning, i kept pressing more than one button at once, but i have had it for a month now and can type long paragraohs on it pretty painlessly. you get used to it after 2 or three days (but i text a lot). Awesome Flash Though it isn't the greatest flash in the world, I can use it as a flashlight with an app i downloaded Light Sensor It has a light sensor so that it goes black when you are on the phone. I got an app that makes it change the brightness according to how bright it is outside Widgets It is not just a screen with apps on it or a menu screen. I can have a music widget that lets me pause, change tracks or whatever without opening anything, I can look at the weather without opening anything, I can look at my calender and to do list without opening apps. It is cool to be able to do anything without haveing to press on an app and wait for it to load. I just have to unlock the phone and look at the screen. Permanent buttons Well, more like touch buttons. It is a good feeling to know that wherever I go, there are four buttons on the botton that can always be used. One is BACK for webpages and apps. It brings you back to the last page you were looking at. It is not an on-screen thing that veries between apps. MENU- this is like an options button. like a right click or control click. If you want to find any options within an app, press that. HOME is... well... the home screen. SEARCH can be used for google on the web, searching for music on the music player, and finding contacts. Loud speakers and Long battery life Real camera button Apps Apple won't give you Verizon network it isn't perfect, despite what you may hear. You have to manually split up your texts into 160 characters if you are not texting someone with verizon. SOOOO DUMB. also the 3G is slower than AT&T Removable memory As soon as they are on the market, I am totally up for a 32GB micro SD card. This is good because I have way more than 16GB of music that I want on my phone. However, I go through phases and just don't put on music i used to listen to but don't listen to anymore That is all i can think of now. I can't talk a whole lot about software, because you make that your own with different Apps. One thing though, don't expect upgrades all the time like Iphone owners. Google goes from one project to the next without focusing a whole lot on older stuff. I really really like my phone Favorite apps in Descending order: These are in order of how much i use them as well as overall awesomeness. That is why Google Goggles, The metal detector, and the barcode scanner are not on it. Also, I am not including anything that is already on the phone or the Advanced Task Killer because those are givens Beautiful Widgets (about $1.50 [only one on this list i payed for] but make my homescreen beautiful with wifi, gps, etc widgets. Also, it displays the weather beautifully. You click on it in different areas to get to Google Calender and Alarm) GDocs (Google Docs. Cloud is goodness) MixZing Lite (Music player. Automatically adds songs that go with your music after a CD or playlist. Like automatic Genius) Bible (different versions of the bible and is my favorite bible app that i have tried) Ringdroid (make ringtones, notification ringtones, and alarms with your music) Abduction (favorite casual game. Trust me, get it) Quick Settings (I use it because it changes the brightness automatically and It makes it easy to get to settings that you actually use) MotoTorch LED (I use it because it comes with a widget that lets me turn on the flash as a flashlight with one press) Robo Defence (Really addicting Tower Defence. You have to buy the real one after a day or so of playing [which is also worth it]. Not a casual game) Seesmic (favorite Twitter app) Astrid (displays a widget to-do list) gStrings (tuner for my guitar) Bebbled (completely free game. No upgrades. I can't believe the quality. It is just a cool casual game)

No. 742

My battery can't hold a charge and my phone freezes up at least once a day and I have to either shut it off or disconnect the battery. It's no good and you should purchase something else.

No. 763

The Nokia E71 is everything that I expected, except for one small detail. I had read several reviews that indicated a problem with the battery cover. I'm having same problem ... cover doesn't lock in place on both sides ... locks on only one side. However, I can live with this imperfection. I love my phone.

No. 852

I bought this phone because ""is a nokia, is a good Cellphone"" when I received this phone, on the top of the front mask have a little open. then I insert a micro SD and the phone no work properly, when I press on the top of front mask phone display ""REMOVE MEMORY CARD AND PRESS OK""I lost maybe ten o more pictures take with the camera, the phone erase it. I NOT RECOMMEND THIS PHONE.

No. 246

It's the best phone I've had... I used to have a Sony Ericsson w800i and traded it for my new Nokia E71 'cause for me, it's the best brand out there. The phone works perfect! Thanks so much!

Questionnaire 2 and responses

What are the reasons helping you to decide post No. 145 is helpful? Rank them from most important one to the least important one.

The user has commented about the good and bad features of the phone briefly. Besides, the user has posted the modification he wished to have for his phone.

What are the reasons helping you to decide post No. 742 is helpful? Rank them from most important one to the least important one.

The user has posted a problem about the battery which is very important to engineer.

What are the reasons helping you to decide post No. 763 is helpful? Rank them from most important one to the least important one.

The user has reported a problem about the phone cover like the other users. It's an useful comment for the engneer.

What are the reasons helping you to decide post No. 852 is helpful? Rank them from most important one to the least important one.

The user reported that the phone didn't work properly since the front mask had a little open. Engineer should be careful about this case.

What are the reasons helping you to decide post No. 246 is NOT helpful? Rank them from most important one to the least important one.

Although the user liked the phone very much, he didn't show the reason.

Student 2

Reviews

No. 81

Let me start with the positive first. This phone has some really good apps and the large screen makes it a true iPhone killer. Here is the bad. Bluetooth voice dialing does not work and Motorola does not seem to care!!! This is a serious safety feature that is a must have in many states. In NYC, if you are caught using a cell phone while driving without using a hands free device, the violation ticket will set you back $135. I can't imagine a cell phone which does not support hands-free voice dialing! Until Motoralo wakes up and puts resources into fixing this product gap, I would suggest people NOT to buy this phone.

No. 145

With any phone, you have problems. Even with these aside, I very strongly reccomend the Droid Cons: The camera is pretty lousy If you are not in daylight, prepare to take several pictures and delete the worst ones. It just can't focus very well at all. Don't let the ""5 Megapixel"" slogan fool you. Megapixels don't make a camera a good camera. If you can't focus, then what's the point? Also, the flash is less than awesome. All of my pictures show a pretty obvious bright circle where the flash was. However, it's usable... and i don't take a lot of pictures anyways It is pretty buggy Sometimes, I'll get a text, and I'll go to it, and there will be no text message there. Then I'll exit and come back to messaging, and i'll see it. Once, i pressed messaging and it went to my Bank of America app. I constantly get email notifications and there'll be no email there. Once, it just started resetting every time i connected it to my computer. However, I just did a factory reset and it worked fine, making me think it was a bad app. The onscreen keyboard is way harder than an Iphone's I got this because it has a real keyboard, which is great. After 2 days or so, I can type just as fast as any other guy, and i use google docs on it a lot. Nothing wrong with the eral keyboard. However, the onscreen keyboard, at least in verticle mode (why would you use it in horizontal mode if the keyboard is right there?) is kinda hard for me. Maybe it is because i don't practice a lot, but i constantly hit the wrong button. This might be because the Droid screen, though longer, is a little thinner than the Iphone's. It is totally usable, don't get me wrong, but i can type twice as fast on an onscreen keyboard You need to do some work to make it be what you want it to be The music player on it is pretty lousy and so is the settings menu. It is tedious if you want to see the date. blah blah blahThat is kinda the fun part, though. There are awesome music players to download (for free). Tune wiki scrolls lyrics from the internet as you listen and mixzing finds album artwork from the internet. There are several different settings apps: a played with one that had different profiles for one-click settings as opposed to changing several individual settings. I downloaded Beautiful Widgets that displays the time and date really well and makes it really easy to switch to silent, wifi, gps, bluetooth, vibrate, etc. You have to (get to) make it your own. However, that's the kind of phone it is. Your phone is SO DIFFERENT than everyone else's. If you have an hour of free time, you can try new things Not smudge resistant I wipe smudges away after every time i take it out. I don't have anything on the screen though The price for being different You can't get popular apps. I go to UT and can't set up wifi. However, Iphone owners have a app just for the UT wifi. Wow People always wanting to play with your phone Pros:Multitasking The other day, I used Google Docs, turn by turn GPS, a twitter App, AND Pandora at the same time. Awesome Custonizable It is fully customizable. It is not just a black screen with apps on it (like another phone i know about). You can put on your own background, download a different keyboard, work on different browsers, and literally change every aspect of your phone. Your DROID is like no one else's Droid. Maybe you can sleep with it next to you- it is a part of you Real Keyboard Awesome. In the beginning, i kept pressing more than one button at once, but i have had it for a month now and can type long paragraohs on it pretty painlessly. you get used to it after 2 or three days (but i text a lot). Awesome Flash Though it isn't the greatest flash in the world, I can use it as a flashlight with an app i downloaded Light Sensor It has a light sensor so that it goes black when you are on the phone. I got an app that makes it change the brightness according to how bright it is outside Widgets It is not just a screen with apps on it or a menu screen. I can have a music widget that lets me pause, change tracks or whatever without opening anything, I can look at the weather without opening anything, I can look at my calender and to do list without opening apps. It is cool to be able to do anything without haveing to press on an app and wait for it to load. I just have to unlock the phone and look at the screen. Permanent buttons Well, more like touch buttons. It is a good feeling to know that wherever I go, there are four buttons on the botton that can always be used. One is BACK for webpages and apps. It brings you back to the last page you were looking at. It is not an on-screen thing that veries between apps. MENU- this is like an options button. like a right click or control click. If you want to find any options within an app, press that. HOME is... well... the home screen. SEARCH can be used for google on the web, searching for music on the music player, and finding contacts. Loud speakers and Long battery life Real camera button Apps Apple won't give you Verizon network it isn't perfect, despite what you may hear. You have to manually split up your texts into 160 characters if you are not texting someone with verizon. SOOOO DUMB. also the 3G is slower than AT&T Removable memory As soon as they are on the market, I am totally up for a 32GB micro SD card. This is good because I have way more than 16GB of music that I want on my phone. However, I go through phases and just don't put on music i used to listen to but don't listen to anymore That is all i can think of now. I can't talk a whole lot about software, because you make that your own with different Apps. One thing though, don't expect upgrades all the time like Iphone owners. Google goes from one project to the next without focusing a whole lot on older stuff. I really really like my phone Favorite apps in Descending order: These are in order of how much i use them as well as overall awesomeness. That is why Google Goggles, The metal detector, and the barcode scanner are not on it. Also, I am not including anything that is already on the phone or the Advanced Task Killer because those are givens Beautiful Widgets (about $1.50 [only one on this list i payed for] but make my homescreen beautiful with wifi, gps, etc widgets. Also, it displays the weather beautifully. You click on it in different areas to get to Google Calender and Alarm) GDocs (Google Docs. Cloud is goodness) MixZing Lite (Music player. Automatically adds songs that go with your music after a CD or playlist. Like automatic Genius) Bible (different versions of the bible and is my favorite bible app that i have tried) Ringdroid (make ringtones, notification ringtones, and alarms with your music) Abduction (favorite casual game. Trust me, get it) Quick Settings (I use it because it changes the brightness automatically and It makes it easy to get to settings that you actually use) MotoTorch LED (I use it because it comes with a widget that lets me turn on the flash as a flashlight with one press) Robo Defence (Really addicting Tower Defence. You have to buy the real one after a day or so of playing [which is also worth it]. Not a casual game) Seesmic (favorite Twitter app) Astrid (displays a widget to-do list) gStrings (tuner for my guitar) Bebbled (completely free game. No upgrades. I can't believe the quality. It is just a cool casual game)

No. 205

I purchased the Nokia n810 as a belated Christmas gift for my son who had rejected the Ipod Touch Gen 2 I originally sent -- I kept the Touch. After a couple weeks he e-mailed:""The Nokia n810 works fairly well. So far, it plays all my movies, music, and images, so it does well in that department and fulfills my wants as a media player. Gamewise, I'm still learning what it can do. My greatest complaint is that Nethack stopped working without explanation. But that's a small thing. In net browsing and e-mail,well, this keyboard may be little, but it beats the heck out of virtual keyboards and handwriting recognition (the n810 actually has the latter). It's a little slow and clumsy on the browsing side, but it does the job adequately. Let's see... What else... I haven't gotten any chat clients running yet, so I'll have to see how that works out. I tnink my greatest worry is that each application I add to this tablet takes up a bit of it's limited (256 MB or so) main memory even when it's not supposed to be running. I'm not sure why, but at least they don't take up too much. Videos require conversion, but I've come to suspect that's the case for all media players. The Sansa and n810 require it, and the iPod made you download pre-converted videos from iTunes. On the plus side, the converted videos are half the size, and I have an 8 GB expansion card to put them on. Music and images are simple drag-and drop. No fuss. Ah, right. My selective memory forgot that getting the n810 to actually work right took a day of screaming and cursing and two calls to tech support before I could get the upgraded OS installed, but since then it's been behaving itself. Overall, I'd have to say I'm satisfied. I don't have everything tweaked quite my way yet, but it -is- very tweakable, and it's much better suited to my needs than an iPod. I mean, seriously. This thing thinks it's a computer. I can even do Telnet on it. But I haven't gotten the Bittorrent ap to work yet.. That would be the coup de grace."" There you have it....

No. 503

Simple put, The Razr is one of worst & best phones I've ever purchased. I purchased one for me and one for my wife. For a point of reference, my first phone was a Motorola Analog Startac in 1995.I'll lay out the good and bad, and let you decide for yourself. Good Voice recognition - works very well. Layout and size - Button layout is good and size if great. Screen - Good size and resolution Camera - Works well if you can get past the Verizon bs to get the pictures off the camera. Audio Quality - Good and good reception. BadSlow - OMG! This phone is so slow. Agonizing slow. Hit menu and wait 3 to 4 seconds for it to come up. Yeah that's right, 4 seconds, I just timed it. Battery - If you plan on talking on this phone, then you need to charge every night. If you talk during the day, charge it when your not on it. Ring - No ring and vib. Only ring or vib. This phone has a lot going for it, but the speed of the menu functions makes all the bells and whistles a distraction.

No. 992

Review is split by Pros, Cons, and Negligible Factors (to me, but maybe not to you) -- I've had the Droid for 2 weeks now, and already can't imagine life without it. Overall, I give it an 8 out of 10. I also have an iPod Touch (how I became familiar with iPhone's platform), and an ancient work BlackBerry -- I count the Droid as my first ""real"" smartphone.THINGS I LIKE- Total gmail, calendar, maps, google-happy-sync world: All you do is enter your gmail address and password, and boom. Your whole life is on there (assuming you live on google apps).- Android 2.0: Super solid. App switching is AWESOME. Much like google itself, I didn't realize how much I needed it until I had it. Today on the Metro I was looking up a Mark Bittman recipe and while that was loading, I sent an email and checked my calendar. To be able to do all that at once is a huge advantage the Android platform has over iPhone. Also background updating is nice.- Service: It works on the DC Metro. It works everywhere. And it's pretty fast. Typically less than 10-15 sec for page loads.- Design: It's only 1mm thicker than iPhone, and about an ounce heavier. I was pleasantly surprised by how portable it is.- Battery life: A single charge powers a day of regular use, easy -- 2 days with light use.- Screen: I'm not a pixel snob by any means, but the screen is GORGEOUS. Resolution and image quality are fantastic. The touchscreen keyboard is a pleasant surprise -- autofill is as good, if not better, than the one on my iPod Touch (same as iPhone's). I use it way more than I expected, partly because it's so easy to use, partly because the keyboard is so bad (more on that later).- Sound quality: I'm not much of a phone person, but my family is -- talked to my mom for 20 minutes today. Everything she said was crystal clear.- Price: $149 with my New Every 2 discount. That's a great deal. Monthly bill is about $100, which is expensive, but standard for smart phones, and worth it to me. THINGS I DON'T LIKE- Keyboard: Surprisingly bad. Flat keys, not very responsive, which makes it hard to type quickly w/o mistakes. And I have small, nimble Asian fingers.

Maybe I just need more practice? - (For Outlook/Exchange users): No ""search"" function in the Droid's internal email client! There's a search in the gmail app, but if you aren't using gmail, you'll want to throw this phone into a wall. Alternately, you can download a $20 app by Touchdown that will fix this. Still, WTF?! That's a crazy oversight.- Camera: It's OK, but hard to focus. I expected better from a 5MP w/flash. I looked it up online, and other users had similar complaints, but expect it to be fixed with a software update.- On/off/sleep switch: Why is this located on a tiny button on the top right and not on the front? This is the only flaw of the design, but it's a pretty annoying one, seeing as you use the on/off/sleep switch so often. THINGS I DON'T CARE ABOUT, BUT YOU MIGHT- Music player: Music menu isn't great (though sound quality is), but I don't use phones as MP3 players anyway. I prefer my hot pink ipod nano.- App selection: There are enough to keep me happy, and plenty of freebies. Recommend Spare Parts to keep tabs of all your apps and their battery drain. Bar code scanner is included too. [Side note: I know Apple has 140K apps for iPhone/iPod Touch, but honestly, I've never cared much for them. I'm generally against useless time-sucks like the bubble wrap game, or the one that turns your iPhone into a flute or other wind instrument. Like, that was fun for the first 5 minutes, um, now what?]- Video camera: All the reviews say the video recorder is excellent, but I never use it.- Storage: 16gb is more than enough for me. You can add up to 32gb total.

Questionnaire 2 and responses

What are the reasons helping you to decide post No. 81 is helpful? Rank them from most important one to the least important one.

The customer has discussed that the good and bad of the product

The customer has compared the product to the others

The customer has mention what criteria are important when he decide to buy the product

What are the reasons helping you to decide post No. 145 is helpful? Rank them from most important one to the least important one.

The customer has mention what features of the phone satisfy him/her

The customer has discussed the features that still have space to improve.

What are the reasons helping you to decide post No. 205 is helpful? Rank them from most important one to the least important one.

The customer has mention the features that need to improve in the future

The customer has compared some feature of the product to the others

What are the reasons helping you to decide post No. 503 is helpful? Rank them from most important one to the least important one.

The customer has mention the good and bad of the product clearly

The customer is a old user of the brand, he know the good and bad of the brand a lot, and can provide useful information

What are the reasons helping you to decide post No. 992 is helpful? Rank them from most important one to the least important one.

The customer has point out what features satisfy him and his need

The customer has mentioned how he uses the phone and what features is useful and what is not.

Student 3

Reviews

No. 145

With any phone, you have problems. Even with these aside, I very strongly reccomend the Droid Cons: The camera is pretty lousy If you are not in daylight, prepare to take several pictures and delete the worst ones. It just can't focus very well at all. Don't let the ""5 Megapixel"" slogan fool you. Megapixels don't make a camera a good camera. If you can't focus, then what's the point? Also, the flash is less than awesome. All of my pictures show a pretty obvious bright circle where the flash was. However, it's usable... and i don't take a lot of pictures anyways It is pretty buggy Sometimes, I'll get a text, and I'll go to it, and there will be no text message there. Then I'll exit and come back to messaging, and i'll see it. Once, i pressed messaging and it went to my Bank of America app. I constantly get email notifications and there'll be no email there. Once, it just started resetting every time i connected it to my computer. However, I just did a factory reset and it worked fine, making me think it was a bad app. The onscreen keyboard is way harder than an Iphone's I got this because it has a real keyboard, which is great. After 2 days or so, I can type just as fast as any other guy, and i use google docs on it a lot. Nothing wrong with the eral keyboard. However, the onscreen keyboard, at least in verticle mode (why would you use it in horizontal mode if the keyboard is right there?) is kinda hard for me. Maybe it is because i don't practice a lot, but i constantly hit the wrong button. This might be because the Droid screen, though longer, is a little thinner than the Iphone's. It is totally usable, don't get me wrong, but i can type twice as fast on an onscreen keyboard You need to do some work to make it be what you want it to be The music player on it is pretty lousy and so is the settings menu. It is tedious if you want to see the date. blah blah blahThat is kinda the fun part, though. There are awesome music players to download (for free). Tune wiki scrolls lyrics from the internet as you listen and mixzing finds album artwork from the internet. There are several different settings apps: a played with one that had different profiles for one-click settings as opposed to changing several individual settings. I downloaded Beautiful Widgets that displays the time and date really well and makes it really easy to switch to silent, wifi, gps, bluetooth, vibrate, etc. You have to (get to) make it your own. However, that's the kind of phone it is. Your phone is SO DIFFERENT than everyone else's. If you have an hour of free time, you can try new things Not smudge resistant I wipe smudges away after every time i take it out. I don't have anything on the screen though The price for being different You can't get popular apps. I go to UT and can't set up wifi. However, Iphone owners have a app just for the UT wifi. Wow People always wanting to play with your phone Pros:Multitasking The other day, I used Google Docs, turn by turn GPS, a twitter App, AND Pandora at the same time. Awesome Custonizable It is fully customizable. It is not just a black screen with apps on it (like another phone i know about). You can put on your own background, download a different keyboard, work on different browsers, and literally change every aspect of your phone. Your DROID is like no one else's Droid. Maybe you can sleep with it next to you- it is a part of you Real Keyboard Awesome. In the beginning, i kept pressing more than one button at once, but i have had it for a month now and can type long paragraohs on it pretty painlessly. you get used to it after 2 or three days (but i text a lot). Awesome Flash Though it isn't the greatest flash in the world, I can use it as a flashlight with an app i downloaded Light Sensor It has a light sensor so that it goes black when you are on the phone. I got an app that makes it change the brightness according to how bright it is outside Widgets It is not just a screen with apps on it or a menu screen. I can have a music widget that lets me pause, change tracks or whatever without opening anything, I can look at the weather without opening anything, I can look at my calender and to do list without opening apps. It is cool to be able to do anything without haveing to press on an app and wait for it to load. I just have to unlock the phone and look at the screen. Permanent buttons Well, more like touch buttons. It is a good feeling to know that wherever I go, there are four buttons on the botton that can always be used. One is BACK for webpages and apps. It brings you back to the last page you were looking at. It is not an on-screen thing that veries between apps. MENU- this is like an options button. like a right click or control click. If you want to find any options within an app, press that. HOME is... well... the home screen. SEARCH can be used for google on the web, searching for music on the music player, and finding contacts. Loud speakers and Long battery life Real camera button Apps Apple won't give you Verizon network it isn't perfect, despite what you may hear. You have to manually split up your texts into 160 characters if you are not texting someone with verizon. SOOOO DUMB. also the 3G is slower than AT&T Removable memory As soon as they are on the market, I am totally up for a 32GB micro SD card. This is good because I have way more than 16GB of music that I want on my phone. However, I go through phases and just don't put on music i used to listen to but don't listen to anymore That is all i can think of now. I can't talk a whole lot about software, because you make that your own with different Apps. One thing though, don't expect upgrades all the time like Iphone owners. Google goes from one project to the next without focusing a whole lot on older stuff. I really really like my phone Favorite apps in Descending order: These are in order of how much i use them as well as overall awesomeness. That is why Google Goggles, The metal detector, and the barcode scanner are not on it. Also, I am not including anything that is already on the phone or the Advanced Task Killer because those are givens Beautiful Widgets (about $1.50 [only one on this list i payed for] but make my homescreen beautiful with wifi, gps, etc widgets. Also, it displays the weather beautifully. You click on it in different areas to get to Google Calender and Alarm) GDocs (Google Docs. Cloud is goodness) MixZing Lite (Music player. Automatically adds songs that go with your music after a CD or playlist. Like automatic Genius) Bible (different versions of the bible and is my favorite bible app that i have tried) Ringdroid (make ringtones, notification ringtones, and alarms with your music) Abduction (favorite casual game. Trust me, get it) Quick Settings (I use it because it changes the brightness automatically and It makes it easy to get to settings that you actually use) MotoTorch LED (I use it because it comes with a widget that lets me turn on the flash as a flashlight with one press) Robo Defence (Really addicting Tower Defence. You have to buy the real one after a day or so of playing [which is also worth it]. Not a casual game) Seesmic (favorite Twitter app) Astrid (displays a widget to-do list) gStrings (tuner for my guitar) Bebbled (completely free game. No upgrades. I can't believe the quality. It is just a cool casual game)

No. 81

Let me start with the positive first. This phone has some really good apps and the large screen makes it a true iPhone killer. Here is the bad. Bluetooth voice dialing does not work and Motorola does not seem to care!!! This is a serious safety feature that is a must have in many states. In NYC, if you are caught using a cell phone while driving without using a hands free device, the violation ticket will set you back $135. I can't imagine a cell phone which does not support hands-free voice dialing! Until Motoralo wakes up and puts resources into fixing this product gap, I would suggest people NOT to buy this phone.

No. 205

I purchased the Nokia n810 as a belated Christmas gift for my son who had rejected the Ipod Touch Gen 2 I originally sent -- I kept the Touch. After a couple weeks he e-mailed: ""The Nokia n810 works fairly well. So far, it plays all my movies, music, and images, so it does well in that department and fulfills my wants as a media player. Gamewise, I'm still learning what it can do. My greatest complaint is that Nethack stopped working without explanation. But that's a small thing. In net browsing and e-mail,well, this keyboard may be little, but it beats the heck out of virtual keyboards and handwriting recognition (the n810 actually has the latter). It's a little slow and clumsy on the browsing side, but it does the job adequately. Let's see... What else... I haven't gotten any chat clients running yet, so I'll have to see how that works out. I tnink my greatest worry is that each application I add to this tablet takes up a bit of it's limited (256 MB or so) main memory even when it's not supposed to be running. I'm not sure why, but at least they don't take up too much. Videos require conversion, but I've come to suspect that's the case for all media players. The Sansa and n810 require it, and the iPod made you download pre-converted videos from iTunes. On the plus side, the converted videos are half the size, and I have an 8 GB expansion card to put them on. Music and images are simple drag-and drop. No fuss. Ah, right. My selective memory forgot that getting the n810 to actually work right took a day of screaming and cursing and two calls to tech support before I could get the upgraded OS installed, but since then it's been behaving itself. Overall, I'd have to say I'm satisfied. I don't have everything tweaked quite my way yet, but it -is- very tweakable, and it's much better suited to my needs than an iPod. I mean, seriously. This thing thinks it's a computer. I can even do Telnet on it. But I haven't gotten the Bittorrent ap to work yet.. That would be the coup de grace."" There you have it....

No. 246

It's the best phone I've had... I used to have a Sony Ericsson w800i and traded it for my new Nokia E71 'cause for me, it's the best brand out there. The phone works perfect! Thanks so much!

No. 503

Simple put, The Razr is one of worst & best phones I've ever purchased. I purchased one for me and one for my wife. For a point of reference, my first phone was a Motorola Analog Startac in 1995.I'll lay out the good and bad, and let you decide for yourself. Good Voice recognition - works very well. Layout and size - Button layout is good and size if great. Screen - Good size and resolution Camera - Works well if you can get past the Verizon bs to get the pictures off the camera. Audio Quality - Good and good reception. Bad Slow - OMG! This phone is so slow. Agonizing slow. Hit menu and wait 3 to 4 seconds for it to come up. Yeah that's right, 4 seconds, I just timed it. Battery - If you plan on talking on this phone, then you need to charge every night. If you talk during the day, charge it when your not on it. Ring - No ring and vib. Only ring or vib. This phone has a lot going for it, but the speed of the menu functions makes all the bells and whistles a distraction.

No. 852

I bought this phone because ""is a nokia, is a good Cellphone"" when I received this phone, on the top of the front mask have a little open. then I insert a micro SD and the phone no work properly, when I press on the top of front mask phone display ""REMOVE MEMORY CARD AND PRESS OK""I lost maybe ten o more pictures take with the camera, the phone erase it. I NOT RECOMMEND THIS PHONE.

No. 995

The Nokia N800 is a very handy internet tablet. The only downside is that it cannot open attachments for microsoft word, excel or powerpoint.

Questionnaire 2 and responses

What are the reasons helping you to decide post No. 145 is helpful? Rank them from most important one to the least important one.

1. Awesome keyboard

2. Camera can’t focus well

3. Flash makes a bright circle in picture

4. Email connection problem

What are the reasons helping you to decide post No.81 is NOT helpful? Rank them from most important one to the least important one.

Only mention voice dialing problem which had been mentioned by the others

What are the reasons helping you to decide post No.205 is NOT helpful? Rank them from most important one to the least important one.

Overall the user satisfies with the phone.

What are the reasons helping you to decide post No.246 is NOT helpful? Rank them from most important one to the least important one.

In fact, there is no useful comment at all

What are the reasons helping you to decide post No.503 is NOT helpful? Rank them from most important one to the least important one.

I don't know which phone the user is talking about? Analog Startac or Razr

What are the reasons helping you to decide post No.852 is NOT helpful? Rank them from most important one to the least important one.

The problem mentioned should not be related to phone, it should be the microSD card problem.

What are the reasons helping you to decide post No. 995 is NOT helpful? Rank them from most important one to the least important one.

The problem can be solved by installing 3rd party apps. I decided to rate it as not very helpful.

Student 4

Reviews

No. 145

With any phone, you have problems. Even with these aside, I very strongly reccomend the Droid Cons: The camera is pretty lousy If you are not in daylight, prepare to take several pictures and delete the worst ones. It just can't focus very well at all. Don't let the ""5 Megapixel"" slogan fool you. Megapixels don't make a camera a good camera. If you can't focus, then what's the point? Also, the flash is less than awesome. All of my pictures show a pretty obvious bright circle where the flash was. However, it's usable... and i don't take a lot of pictures anyways It is pretty buggy Sometimes, I'll get a text, and I'll go to it, and there will be no text message there. Then I'll exit and come back to messaging, and i'll see it. Once, i pressed messaging and it went to my Bank of America app. I constantly get email notifications and there'll be no email there. Once, it just started resetting every time i connected it to my computer. However, I just did a factory reset and it worked fine, making me think it was a bad app. The onscreen keyboard is way harder than an Iphone's I got this because it has a real keyboard, which is great. After 2 days or so, I can type just as fast as any other guy, and i use google docs on it a lot. Nothing wrong with the eral keyboard. However, the onscreen keyboard, at least in verticle mode (why would you use it in horizontal mode if the keyboard is right there?) is kinda hard for me. Maybe it is because i don't practice a lot, but i constantly hit the wrong button. This might be because the Droid screen, though longer, is a little thinner than the Iphone's. It is totally usable, don't get me wrong, but i can type twice as fast on an onscreen keyboard You need to do some work to make it be what you want it to be The music player on it is pretty lousy and so is the settings menu. It is tedious if you want to see the date. blah blah blahThat is kinda the fun part, though. There are awesome music players to download (for free). Tune wiki scrolls lyrics from the internet as you listen and mixzing finds album artwork from the internet. There are several different settings apps: a played with one that had different profiles for one-click settings as opposed to changing several individual settings. I downloaded Beautiful Widgets that displays the time and date really well and makes it really easy to switch to silent, wifi, gps, bluetooth, vibrate, etc. You have to (get to) make it your own. However, that's the kind of phone it is. Your phone is SO DIFFERENT than everyone else's. If you have an hour of free time, you can try new things Not smudge resistant I wipe smudges away after every time i take it out. I don't have anything on the screen though The price for being different You can't get popular apps. I go to UT and can't set up wifi. However, Iphone owners have a app just for the UT wifi. Wow People always wanting to play with your phone Pros: Multitasking The other day, I used Google Docs, turn by turn GPS, a twitter App, AND Pandora at the same time. Awesome Custonizable It is fully customizable. It is not just a black screen with apps on it (like another phone i know about). You can put on your own background, download a different keyboard, work on different browsers, and literally change every aspect of your phone. Your DROID is like no one else's Droid. Maybe you can sleep with it next to you- it is a part of you Real Keyboard Awesome. In the beginning, i kept pressing more than one button at once, but i have had it for a month now and can type long paragraohs on it pretty painlessly. you get used to it after 2 or three days (but i text a lot). Awesome Flash Though it isn't the greatest flash in the world, I can use it as a flashlight with an app i downloaded Light Sensor It has a light sensor so that it goes black when you are on the phone. I got an app that makes it change the brightness according to how bright it is outside Widgets It is not just a screen with apps on it or a menu screen. I can have a music widget that lets me pause, change tracks or whatever without opening anything, I can look at the weather without opening anything, I can look at my calender and to do list without opening apps. It is cool to be able to do anything without haveing to press on an app and wait for it to load. I just have to unlock the phone and look at the screen. Permanent buttons Well, more like touch buttons. It is a good feeling to know that wherever I go, there are four buttons on the botton that can always be used. One is BACK for webpages and apps. It brings you back to the last page you were looking at. It is not an on-screen thing that veries between apps. MENU- this is like an options button. like a right click or control click. If you want to find any options within an app, press that. HOME is... well... the home screen. SEARCH can be used for google on the web, searching for music on the music player, and finding contacts. Loud speakers and Long battery life Real camera button Apps Apple won't give you Verizon network it isn't perfect, despite what you may hear. You have to manually split up your texts into 160 characters if you are not texting someone with verizon. SOOOO DUMB. also the 3G is slower than AT&T Removable memory As soon as they are on the market, I am totally up for a 32GB micro SD card. This is good because I have way more than 16GB of music that I want on my phone. However, I go through phases and just don't put on music i used to listen to but don't listen to anymore That is all i can think of now. I can't talk a whole lot about software, because you make that your own with different Apps. One thing though, don't expect upgrades all the time like Iphone owners. Google goes from one project to the next without focusing a whole lot on older stuff. I really really like my phone Favorite apps in Descending order: These are in order of how much i use them as well as overall awesomeness. That is why Google Goggles, The metal detector, and the barcode scanner are not on it. Also, I am not including anything that is already on the phone or the Advanced Task Killer because those are givens Beautiful Widgets (about $1.50 [only one on this list i payed for] but make my homescreen beautiful with wifi, gps, etc widgets. Also, it displays the weather beautifully. You click on it in different areas to get to Google Calender and Alarm) GDocs (Google Docs. Cloud is goodness) MixZing Lite (Music player. Automatically adds songs that go with your music after a CD or playlist. Like automatic Genius) Bible (different versions of the bible and is my favorite bible app that i have tried) Ringdroid (make ringtones, notification ringtones, and alarms with your music) Abduction (favorite casual game. Trust me, get it) Quick Settings (I use it because it changes the brightness automatically and It makes it easy to get to settings that you actually use) Moto Torch LED (I use it because it comes with a widget that lets me turn on the flash as a flashlight with one press) Robo Defence (Really addicting Tower Defence. You have to buy the real one after a day or so of playing [which is also worth it]. Not a casual game) Seesmic (favorite Twitter app) Astrid (displays a widget to-do list) gStrings (tuner for my guitar) Bebbled (completely free game. No upgrades. I can't believe the quality. It is just a cool casual game)

No. 205

I purchased the Nokia n810 as a belated Christmas gift for my son who had rejected the Ipod Touch Gen 2 I originally sent -- I kept the Touch. After a couple weeks he e-mailed: ""The Nokia n810 works fairly well. So far, it plays all my movies, music, and images, so it does well in that department and fulfills my wants as a media player. Gamewise, I'm still learning what it can do. My greatest complaint is that Nethack stopped working without explanation. But that's a small thing. In net browsing and e-mail,well, this keyboard may be little, but it beats the heck out of virtual keyboards and handwriting recognition (the n810 actually has the latter). It's a little slow and clumsy on the browsing side, but it does the job adequately. Let's see... What else... I haven't gotten any chat clients running yet, so I'll have to see how that works out. I tnink my greatest worry is that each application I add to this tablet takes up a bit of it's limited (256 MB or so) main memory even when it's not supposed to be running. I'm not sure why, but at least they don't take up too much. Videos require conversion, but I've come to suspect that's the case for all media players. The Sansa and n810 require it, and the iPod made you download pre-converted videos from iTunes. On the plus side, the converted videos are half the size, and I have an 8 GB expansion card to put them on. Music and images are simple drag-and drop. No fuss. Ah, right. My selective memory forgot that getting the n810 to actually work right took a day of screaming and cursing and two calls to tech support before I could get the upgraded OS installed, but since then it's been behaving itself. Overall, I'd have to say I'm satisfied. I don't have everything tweaked quite my way yet, but it -is- very tweakable, and it's much better suited to my needs than an iPod. I mean, seriously. This thing thinks it's a computer. I can even do Telnet on it. But I haven't gotten the Bittorrent ap to work yet.. That would be the coup de grace."" There you have it....

No. 246

It's the best phone I've had... I used to have a Sony Ericsson w800i and traded it for my new Nokia E71 'cause for me, it's the best brand out there. The phone works perfect! Thanks so much!

No. 503

Simple put, The Razr is one of worst & best phones I've ever purchased. I purchased one for me and one for my wife. For a point of reference, my first phone was a Motorola Analog Startac in 1995.I'll lay out the good and bad, and let you decide for yourself. Good Voice recognition - works very well. Layout and size - Button layout is good and size if great.Screen - Good size and resolution Camera - Works well if you can get past the Verizon bs to get the pictures off the camera. Audio Quality - Good and good reception. Bad Slow - OMG! This phone is so slow. Agonizing slow. Hit menu and wait 3 to 4 seconds for it to come up. Yeah that's right, 4 seconds, I just timed it. Battery - If you plan on talking on this phone, then you need to charge every night. If you talk during the day, charge it when your not on it. Ring - No ring and vib. Only ring or vib. This phone has a lot going for it, but the speed of the menu functions makes all the bells and whistles a distraction.

No. 992

Review is split by Pros, Cons, and Negligible Factors (to me, but maybe not to you) -- I've had the Droid for 2 weeks now, and already can't imagine life without it. Overall, I give it an 8 out of 10. I also have an iPod Touch (how I became familiar with iPhone's platform), and an ancient work BlackBerry -- I count the Droid as my first ""real"" smartphone. THINGS I LIKE- Total gmail, calendar, maps, google-happy-sync world: All you do is enter your gmail address and password, and boom. Your whole life is on there (assuming you live on google apps).- Android 2.0: Super solid. App switching is AWESOME. Much like google itself, I didn't realize how much I needed it until I had it. Today on the Metro I was looking up a Mark Bittman recipe and while that was loading, I sent an email and checked my calendar. To be able to do all that at once is a huge advantage the Android platform has over iPhone. Also background updating is nice.- Service: It works on the DC Metro. It works everywhere. And it's pretty fast. Typically less than 10-15 sec for page loads.- Design: It's only 1mm thicker than iPhone, and about an ounce heavier. I was pleasantly surprised by how portable it is.- Battery life: A single charge powers a day of regular use, easy -- 2 days with light use.- Screen: I'm not a pixel snob by any means, but the screen is GORGEOUS. Resolution and image quality are fantastic. The touchscreen keyboard is a pleasant surprise -- autofill is as good, if not better, than the one on my iPod Touch (same as iPhone's). I use it way more than I expected, partly because it's so easy to use, partly because the keyboard is so bad (more on that later).- Sound quality: I'm not much of a phone person, but my family is -- talked to my mom for 20 minutes today. Everything she said was crystal clear.- Price: $149 with my New Every 2 discount. That's a great deal. Monthly bill is about $100, which is expensive, but standard for smart phones, and worth it to me. THINGS I DON'T LIKE- Keyboard: Surprisingly bad. Flat keys, not very responsive, which makes it hard to type quickly w/o mistakes. And I have small, nimble Asian fingers. Maybe I just need more practice? - (For Outlook/Exchange users): No ""search"" function in the Droid's internal email client! There's a search in the gmail app, but if you aren't using gmail, you'll want to throw this phone into a wall. Alternately, you can download a $20 app by Touchdown that will fix this. Still, WTF?! That's a crazy oversight.- Camera: It's OK, but hard to focus. I expected better from a 5MP w/flash. I looked it up online, and other users had similar complaints, but expect it to be fixed with a software update.- On/off/sleep switch: Why is this located on a tiny button on the top right and not on the front? This is the only flaw of the design, but it's a pretty annoying one, seeing as you use the on/off/sleep switch so often. THINGS I DON'T CARE ABOUT, BUT YOU MIGHT- Music player: Music menu isn't great (though sound quality is), but I don't use phones as MP3 players anyway. I prefer my hot pink ipod nano.- App selection: There are enough to keep me happy, and plenty of freebies. Recommend Spare Parts to keep tabs of all your apps and their battery drain. Bar code scanner is included too. [Side note: I know Apple has 140K apps for iPhone/iPod Touch, but honestly, I've never cared much for them. I'm generally against useless time-sucks like the bubble wrap game, or the one that turns your iPhone into a flute or other wind instrument. Like, that was fun for the first 5 minutes, um, now what?]- Video camera: All the reviews say the video recorder is excellent, but I never use it.- Storage: 16gb is more than enough for me. You can add up to 32gb total.

No. 81

Let me start with the positive first. This phone has some really good apps and the large screen makes it a true iPhone killer. Here is the bad. Bluetooth voice dialing does not work and Motorola does not seem to care!!! This is a serious safety feature that is a must have in many states. In NYC, if you are caught using a cell phone while driving without using a hands free device, the violation ticket will set you back $135. I can't imagine a cell phone which does not support hands-free voice dialing! Until Motoralo wakes up and puts resources into fixing this product gap, I would suggest people NOT to buy this phone.

No. 742

My battery can't hold a charge and my phone freezes up at least once a day and I have to either shut it off or disconnect the battery. It's no good and you should purchase something else.

No. 763

The Nokia E71 is everything that I expected, except for one small detail. I had read several reviews that indicated a problem with the battery cover. I'm having same problem ... cover doesn't lock in place on both sides ... locks on only one side. However, I can live with this imperfection. I love my phone.

No. 995

The Nokia N800 is a very handy internet tablet. The only downside is that it cannot open attachments for microsoft word, excel or powerpoint.

Questionnaire 2 and responses

What are the reasons helping you to decide post No. 145 is helpful? Rank them from most important one to the least important one.

Since it give me brief explain on the phone therefore I think the reason are helpful.

What are the reasons helping you to decide post No. 205 is helpful? Rank them from most important one to the least important one.

He uses iPod Touch to compare the function which is helpful.

What are the reasons helping you to decide post No. 246 is helpful? Rank them from most important one to the least important one.

The case should be ranked -2 since it didn't seem to helpful.

What are the reasons helping you to decide post No. 503 is helpful? Rank them from most important one to the least important one.

Since he points out the good thing of the phone like [Voice recognition]…

What are the reasons helping you to decide post No. 992 is helpful? Rank them from most important one to the least important one.

The explanations of the phone are brief enough.

What are the reasons helping you to decide post No. 81 is NOT helpful? Rank them from most important one to the least important one.

Since the comments were not helpful to us.

What are the reasons helping you to decide post No. 742 is NOT helpful? Rank them from most important one to the least important one.

The comment was too short to help us.

What are the reasons helping you to decide post No. 763 is NOT helpful? Rank them from most important one to the least important one.

The comments are talking about the shop which was not related to the phone

What are the reasons helping you to decide post No. 995 is NOT helpful? Rank them from most important one to the least important one.

The comment was too short to help us.

Student 5

Reviews

No. 81

Let me start with the positive first. This phone has some really good apps and the large screen makes it a true iPhone killer. Here is the bad. Bluetooth voice dialing does not work and Motorola does not seem to care!!! This is a serious safety feature that is a must have in many states. In NYC, if you are caught using a cell phone while driving without using a hands free device, the violation ticket will set you back $135. I can't imagine a cell phone which does not support hands-free voice dialing!

Until Motoralo wakes up and puts resources into fixing this product gap, I would suggest people NOT to buy this phone.

No. 742

My battery can't hold a charge and my phone freezes up at least once a day and I have to either shut it off or disconnect the battery. It's no good and you should purchase something else.

No. 763

The Nokia E71 is everything that I expected, except for one small detail. I had read several reviews that indicated a problem with the battery cover. I'm having same problem ... cover doesn't lock in place on both sides ... locks on only one side. However, I can live with this imperfection. I love my phone.

No. 852

I bought this phone because ""is a nokia, is a good Cellphone"" when I received this phone, on the top of the front mask have a little open. then I insert a micro SD and the phone no work properly, when I press on the top of front mask phone display ""REMOVE MEMORY CARD AND PRESS OK""I lost maybe ten o more pictures take with the camera, the phone erase it. I NOT RECOMMEND THIS PHONE.

No. 995

The Nokia N800 is a very handy internet tablet. The only downside is that it cannot open attachments for microsoft word, excel or powerpoint.

No. 145

With any phone, you have problems. Even with these aside, I very strongly reccomend the Droid Cons: The camera is pretty lousy If you are not in daylight, prepare to take several pictures and delete the worst ones. It just can't focus very well at all. Don't let the ""5 Megapixel"" slogan fool you. Megapixels don't make a camera a good camera. If you can't focus, then what's the point? Also, the flash is less than awesome. All of my pictures show a pretty obvious bright circle where the flash was. However, it's usable... and i don't take a lot of pictures anyways It is pretty buggy Sometimes, I'll get a text, and I'll go to it, and there will be no text message there. Then I'll exit and come back to messaging, and i'll see it. Once, i pressed messaging and it went to my Bank of America app. I constantly get email notifications and there'll be no email there. Once, it just started resetting every time i connected it to my computer. However, I just did a factory reset and it worked fine, making me think it was a bad app. The onscreen keyboard is way harder than an Iphone's I got this because it has a real keyboard, which is great. After 2 days or so, I can type just as fast as any other guy, and i use google docs on it a lot. Nothing wrong with the eral keyboard. However, the onscreen keyboard, at least in verticle mode (why would you use it in horizontal mode if the keyboard is right there?) is kinda hard for me. Maybe it is because i don't practice a lot, but i constantly hit the wrong button. This might be because the Droid screen, though longer, is a little thinner than the Iphone's. It is totally usable, don't get me wrong, but i can type twice as fast on an onscreen keyboard You need to do some work to make it be what you want it to be The music player on it is pretty lousy and so is the settings menu. It is tedious if you want to see the date. blah blah blahThat is kinda the fun part, though. There are awesome music players to download (for free). Tune wiki scrolls lyrics from the internet as you listen and mixzing finds album artwork from the internet. There are several different settings apps: a played with one that had different profiles for one-click settings as opposed to changing several individual settings. I downloaded Beautiful Widgets that displays the time and date really well and makes it really easy to switch to silent, wifi, gps, bluetooth, vibrate, etc. You have to (get to) make it your own. However, that's the kind of phone it is. Your phone is SO DIFFERENT than everyone else's. If you have an hour of free time, you can try new things Not smudge resistant I wipe smudges away after every time i take it out. I don't have anything on the screen though The price for being different You can't get popular apps. I go to UT and can't set up wifi. However, Iphone owners have a app just for the UT wifi. Wow People always wanting to play with your phone Pros:Multitasking The other day, I used Google Docs, turn by turn GPS, a twitter App, AND Pandora at the same time. Awesome Custonizable It is fully customizable. It is not just a black screen with apps on it (like another phone i know about). You can put on your own background, download a different keyboard, work on different browsers, and literally change every aspect of your phone. Your DROID is like no one else's Droid. Maybe you can sleep with it next to you- it is a part of you Real Keyboard Awesome. In the beginning, i kept pressing more than one button at once, but i have had it for a month now and can type long paragraohs on it pretty painlessly. you get used to it after 2 or three days (but i text a lot). Awesome Flash Though it isn't the greatest flash in the world, I can use it as a flashlight with an app i downloaded Light Sensor It has a light sensor so that it goes black when you are on the phone. I got an app that makes it change the brightness according to how bright it is outside Widgets It is not just a screen with apps on it or a menu screen. I can have a music widget that lets me pause, change tracks or whatever without opening anything, I can look at the weather without opening anything, I can look at my calender and to do list without opening apps. It is cool to be able to do anything without haveing to press on an app and wait for it to load. I just have to unlock the phone and look at the screen. Permanent buttons Well, more like touch buttons. It is a good feeling to know that wherever I go, there are four buttons on the botton that can always be used. One is BACK for webpages and apps. It brings you back to the last page you were looking at. It is not an on-screen thing that veries between apps.

MENU- this is like an options button. like a right click or control click. If you want to find any options within an app, press that. HOME is... well... the home screen. SEARCH can be used for google on the web, searching for music on the music player, and finding contacts. Loud speakers and Long battery life Real camera button Apps Apple won't give you Verizon network it isn't perfect, despite what you may hear. You have to manually split up your texts into 160 characters if you are not texting someone with verizon. SOOOO DUMB. also the 3G is slower than AT&T Removable memory As soon as they are on the market, I am totally up for a 32GB micro SD card. This is good because I have way more than 16GB of music that I want on my phone. However, I go through phases and just don't put on music i used to listen to but don't listen to anymore That is all i can think of now. I can't talk a whole lot about software, because you make that your own with different Apps. One thing though, don't expect upgrades all the time like Iphone owners. Google goes from one project to the next without focusing a whole lot on older stuff. I really really like my phone Favorite apps in Descending order: These are in order of how much i use them as well as overall awesomeness. That is why Google Goggles, The metal detector, and the barcode scanner are not on it. Also, I am not including anything that is already on the phone or the Advanced Task Killer because those are givens Beautiful Widgets (about $1.50 [only one on this list i payed for] but make my homescreen beautiful with wifi, gps, etc widgets. Also, it displays the weather beautifully. You click on it in different areas to get to Google Calender and Alarm) GDocs (Google Docs. Cloud is goodness) MixZing Lite (Music player. Automatically adds songs that go with your music after a CD or playlist. Like automatic Genius) Bible (different versions of the bible and is my favorite bible app that i have tried) Ringdroid (make ringtones, notification ringtones, and alarms with your music) Abduction (favorite casual game. Trust me, get it) Quick Settings (I use it because it changes the brightness automatically and It makes it easy to get to settings that you actually use) MotoTorch LED (I use it because it comes with a widget that lets me turn on the flash as a flashlight with one press) Robo Defence (Really addicting Tower Defence. You have to buy the real one after a day or so of playing [which is also worth it]. Not a casual game) Seesmic (favorite Twitter app) Astrid (displays a widget to-do list) gStrings (tuner for my guitar) Bebbled (completely free game. No upgrades. I can't believe the quality. It is just a cool casual game)

No. 205

I purchased the Nokia n810 as a belated Christmas gift for my son who had rejected the Ipod Touch Gen 2 I originally sent -- I kept the Touch. After a couple weeks he e-mailed:""The Nokia n810 works fairly well. So far, it plays all my movies, music, and images, so it does well in that department and fulfills my wants as a media player. Gamewise, I'm still learning what it can do. My greatest complaint is that Nethack stopped working without explanation. But that's a small thing. In net browsing and e-mail,well, this keyboard may be little, but it beats the heck out of virtual keyboards and handwriting recognition (the n810 actually has the latter). It's a little slow and clumsy on the browsing side, but it does the job adequately. Let's see... What else... I haven't gotten any chat clients running yet, so I'll have to see how that works out. I tnink my greatest worry is that each application I add to this tablet takes up a bit of it's limited (256 MB or so) main memory even when it's not supposed to be running. I'm not sure why, but at least they don't take up too much. Videos require conversion, but I've come to suspect that's the case for all media players. The Sansa and n810 require it, and the iPod made you download pre-converted videos from iTunes. On the plus side, the converted videos are half the size, and I have an 8 GB expansion card to put them on. Music and images are simple drag-and drop. No fuss. Ah, right. My selective memory forgot that getting the n810 to actually work right took a day of screaming and cursing and two calls to tech support before I could get the upgraded OS installed, but since then it's been behaving itself. Overall, I'd have to say I'm satisfied. I don't have everything tweaked quite my way yet, but it -is- very tweakable, and it's much better suited to my needs than an iPod. I mean, seriously. This thing thinks it's a computer. I can even do Telnet on it. But I haven't gotten the Bittorrent ap to work yet.. That would be the coup de grace."" There you have it....

No. 246

It's the best phone I've had... I used to have a Sony Ericsson w800i and traded it for my new Nokia E71 'cause for me, it's the best brand out there. The phone works perfect! Thanks so much!

No. 992

Review is split by Pros, Cons, and Negligible Factors (to me, but maybe not to you) -- I've had the Droid for 2 weeks now, and already can't imagine life without it. Overall, I give it an 8 out of 10. I also have an iPod Touch (how I became familiar with iPhone's platform), and an ancient work BlackBerry -- I count the Droid as my first ""real"" smart phone. THINGS I LIKE- Total gmail, calendar, maps, google-happy-sync world: All you do is enter your gmail address and password, and boom. Your whole life is on there (assuming you live on google apps).- Android 2.0: Super solid. App switching is AWESOME. Much like google itself, I didn't realize how much I needed it until I had it. Today on the Metro I was looking up a Mark Bittman recipe and while that was loading, I sent an email and checked my calendar. To be able to do all that at once is a huge advantage the Android platform has over iPhone. Also background updating is nice.- Service: It works on the DC Metro. It works everywhere. And it's pretty fast. Typically less than 10-15 sec for page loads.- Design: It's only 1mm thicker than iPhone, and about an ounce heavier. I was pleasantly surprised by how portable it is.- Battery life: A single charge powers a day of regular use, easy -- 2 days with light use.- Screen: I'm not a pixel snob by any means, but the screen is GORGEOUS. Resolution and image quality are fantastic. The touch screen keyboard is a pleasant surprise – auto fill is as good, if not better, than the one on my iPod Touch (same as iPhone's). I use it way more than I expected, partly because it's so easy to use, partly because the keyboard is so bad (more on that later).- Sound quality: I'm not much of a phone person, but my family is -- talked to my mom for 20 minutes today. Everything she said was crystal clear.- Price: $149 with my New Every 2 discount. That's a great deal. Monthly bill is about $100, which is expensive, but standard for smart phones, and worth it to me. THINGS I DON'T LIKE- Keyboard: Surprisingly bad. Flat keys, not very responsive, which makes it hard to type quickly w/o mistakes. And I have small, nimble Asian fingers. Maybe I just need more practice? - (For Outlook/Exchange users): No ""search"" function in the Droid's internal email client! There's a search in the gmail app, but if you aren't using gmail, you'll want to throw this phone into a wall. Alternately, you can download a $20 app by Touchdown that will fix this. Still, WTF?! That's a crazy oversight.- Camera: It's OK, but hard to focus. I expected better from a 5MP w/flash. I looked it up online, and other users had similar complaints, but expect it to be fixed with a software update.- On/off/sleep switch: Why is this located on a tiny button on the top right and not on the front? This is the only flaw of the design, but it's a pretty annoying one, seeing as you use the on/off/sleep switch so often. THINGS I DON'T CARE ABOUT, BUT YOU MIGHT- Music player: Music menu isn't great (though sound quality is), but I don't use phones as MP3 players anyway. I prefer my hot pink ipod nano.- App selection: There are enough to keep me happy, and plenty of freebies. Recommend Spare Parts to keep tabs of all your apps and their battery drain. Bar code scanner is included too. [Side note: I know Apple has 140K apps for iPhone/iPod Touch, but honestly, I've never cared much for them. I'm generally against useless time-sucks like the bubble wrap game, or the one that turns your iPhone into a flute or other wind instrument. Like, that was fun for the first 5 minutes, um, now what?]- Video camera: All the reviews say the video recorder is excellent, but I never use it.- Storage: 16gb is more than enough for me. You can add up to 32gb total.

Questionnaire 2 and responses

What are the reasons helping you to decide post No. 81 is helpful? Rank them from most important one to the least important one.

Bluetooth Voice dialing is not supported

What are the reasons helping you to decide post No. 742is helpful? Rank them from most important one to the least important one.

OS freezes

Charge battery cannot last

What are the reasons helping you to decide post No. 763 is helpful? Rank them from most important one to the least important one.

Cover cannot close properly (Redesign of cover or quality check required)

What are the reasons helping you to decide post No. 852 is helpful? Rank them from most important one to the least important one.

SD card detection problem (requires either OS or Hardware check)

What are the reasons helping you to decide post No. 995 is helpful? Rank them from most important one to the least important one.

Cannot open Word, excel, PowerPoint attachments (its really important since having the ability to open these extensions are the minimum requirements of smart phones these days)

What are the reasons helping you to decide post No. 145 is NOT helpful? Rank them from most important one to the least important one.

Customer mainly talks about the usefulness of the phone, the 3rd party applications and not much about the bad sides of the phone.

What are the reasons helping you to decide post No. 205 is NOT helpful? Rank them from most important one to the least important one.

Customer only talks about the usefulness of the phone, except about video conversations, however it is very difficult for mobile phones to support all types of format nowadays so we can omit that.

What are the reasons helping you to decide post No. 246 is NOT helpful? Rank them from most important one to the least important one.

-Customer just says it's a nice phone, no user comments about it

What are the reasons helping you to decide post No. 992 is NOT helpful? Rank them from most important one to the least important one.

The user mainly talks about the functions of the phone. Since the A855 uses the Google Android Operating system, thus the user interface can only be designed and refined by Google, Motorola can only have the ability to focus on the hardware of the phone, about the hardware side, only small keyboard is a problem which is already very common on other comments already.

Student 6

Reviews

No. 145

With any phone, you have problems. Even with these aside, I very strongly reccomend the Droid Cons: The camera is pretty lousy If you are not in daylight, prepare to take several pictures and delete the worst ones. It just can't focus very well at all. Don't let the ""5 Megapixel"" slogan fool you. Megapixels don't make a camera a good camera. If you can't focus, then what's the point? Also, the flash is less than awesome. All of my pictures show a pretty obvious bright circle where the flash was. However, it's usable... and i don't take a lot of pictures anyways It is pretty buggy Sometimes, I'll get a text, and I'll go to it, and there will be no text message there. Then I'll exit and come back to messaging, and i'll see it. Once, i pressed messaging and it went to my Bank of America app. I constantly get email notifications and there'll be no email there. Once, it just started resetting every time i connected it to my computer. However, I just did a factory reset and it worked fine, making me think it was a bad app. The onscreen keyboard is way harder than an Iphone's I got this because it has a real keyboard, which is great. After 2 days or so, I can type just as fast as any other guy, and i use google docs on it a lot. Nothing wrong with the eral keyboard. However, the onscreen keyboard, at least in verticle mode (why would you use it in horizontal mode if the keyboard is right there?) is kinda hard for me. Maybe it is because i don't practice a lot, but i constantly hit the wrong button. This might be because the Droid screen, though longer, is a little thinner than the Iphone's. It is totally usable, don't get me wrong, but i can type twice as fast on an onscreen keyboard You need to do some work to make it be what you want it to be The music player on it is pretty lousy and so is the settings menu. It is tedious if you want to see the date. blah blah blahThat is kinda the fun part, though. There are awesome music players to download (for free). Tune wiki scrolls lyrics from the internet as you listen and mixzing finds album artwork from the internet. There are several different settings apps: a played with one that had different profiles for one-click settings as opposed to changing several individual settings. I downloaded Beautiful Widgets that displays the time and date really well and makes it really easy to switch to silent, wifi, gps, bluetooth, vibrate, etc. You have to (get to) make it your own. However, that's the kind of phone it is. Your phone is SO DIFFERENT than everyone else's. If you have an hour of free time, you can try new things Not smudge resistant I wipe smudges away after every time i take it out. I don't have anything on the screen though The price for being different You can't get popular apps. I go to UT and can't set up wifi. However, Iphone owners have a app just for the UT wifi. Wow People always wanting to play with your phone Pros: Multitasking The other day, I used Google Docs, turn by turn GPS, a twitter App, AND Pandora at the same time. Awesome Custonizable It is fully customizable. It is not just a black screen with apps on it (like another phone i know about). You can put on your own background, download a different keyboard, work on different browsers, and literally change every aspect of your phone. Your DROID is like no one else's Droid. Maybe you can sleep with it next to you- it is a part of you Real Keyboard Awesome. In the beginning, i kept pressing more than one button at once, but i have had it for a month now and can type long paragraohs on it pretty painlessly. you get used to it after 2 or three days (but i text a lot). Awesome Flash Though it isn't the greatest flash in the world, I can use it as a flashlight with an app i downloaded Light Sensor It has a light sensor so that it goes black when you are on the phone. I got an app that makes it change the brightness according to how bright it is outside Widgets It is not just a screen with apps on it or a menu screen. I can have a music widget that lets me pause, change tracks or whatever without opening anything, I can look at the weather without opening anything, I can look at my calender and to do list without opening apps. It is cool to be able to do anything without haveing to press on an app and wait for it to load. I just have to unlock the phone and look at the screen. Permanent buttons Well, more like touch buttons. It is a good feeling to know that wherever I go, there are four buttons on the botton that can always be used. One is BACK for webpages and apps. It brings you back to the last page you were looking at. It is not an on-screen thing that veries between apps. MENU- this is like an options button. like a right click or control click. If you want to find any options within an app, press that. HOME is... well... the home screen. SEARCH can be used for google on the web, searching for music on the music player, and finding contacts. Loud speakers and Long battery life Real camera button Apps Apple won't give you Verizon network it isn't perfect, despite what you may hear. You have to manually split up your texts into 160 characters if you are not texting someone with verizon. SOOOO DUMB. also the 3G is slower than AT&T Removable memory As soon as they are on the market, I am totally up for a 32GB micro SD card. This is good because I have way more than 16GB of music that I want on my phone. However, I go through phases and just don't put on music i used to listen to but don't listen to anymore That is all i can think of now. I can't talk a whole lot about software, because you make that your own with different Apps. One thing though, don't expect upgrades all the time like Iphone owners. Google goes from one project to the next without focusing a whole lot on older stuff. I really really like my phone Favorite apps in Descending order: These are in order of how much i use them as well as overall awesomeness. That is why Google Goggles, The metal detector, and the barcode scanner are not on it. Also, I am not including anything that is already on the phone or the Advanced Task Killer because those are givens Beautiful Widgets (about $1.50 [only one on this list i payed for] but make my homescreen beautiful with wifi, gps, etc widgets. Also, it displays the weather beautifully. You click on it in different areas to get to Google Calender and Alarm) GDocs (Google Docs. Cloud is goodness) MixZing Lite (Music player. Automatically adds songs that go with your music after a CD or playlist. Like automatic Genius) Bible (different versions of the bible and is my favorite bible app that i have tried) Ringdroid (make ringtones, notification ringtones, and alarms with your music) Abduction (favorite casual game. Trust me, get it) Quick Settings (I use it because it changes the brightness automatically and It makes it easy to get to settings that you actually use) MotoTorch LED (I use it because it comes with a widget that lets me turn on the flash as a flashlight with one press) Robo Defence (Really addicting Tower Defence. You have to buy the real one after a day or so of playing [which is also worth it]. Not a casual game) Seesmic (favorite Twitter app) Astrid (displays a widget to-do list) gStrings (tuner for my guitar) Bebbled (completely free game. No upgrades. I can't believe the quality. It is just a cool casual game)

No. 246

It's the best phone I've had... I used to have a Sony Ericsson w800i and traded it for my new Nokia E71 'cause for me, it's the best brand out there. The phone works perfect! Thanks so much!

No. 503

Simple put, The Razr is one of worst & best phones I've ever purchased. I purchased one for me and one for my wife. For a point of reference, my first phone was a Motorola Analog Startac in 1995.I'll lay out the good and bad, and let you decide for yourself.GoodVoice recognition - works very well. Layout and size - Button layout is good and size if great. Screen - Good size and resolution Camera - Works well if you can get past the Verizon bs to get the pictures off the camera. Audio Quality - Good and good reception. Bad Slow - OMG! This phone is so slow. Agonizing slow. Hit menu and wait 3 to 4 seconds for it to come up. Yeah that's right, 4 seconds, I just timed it. Battery - If you plan on talking on this phone, then you need to charge every night. If you talk during the day, charge it when your not on it.Ring - No ring and vib. Only ring or vib. This phone has a lot going for it, but the speed of the menu functions makes all the bells and whistles a distraction.

No. 742

My battery can't hold a charge and my phone freezes up at least once a day and I have to either shut it off or disconnect the battery. It's no good and you should purchase something else.

No. 763

The Nokia E71 is everything that I expected, except for one small detail. I had read several reviews that indicated a problem with the battery cover. I'm having same problem ... cover doesn't lock in place on both sides ... locks on only one side. However, I can live with this imperfection. I love my phone.

No. 852

I bought this phone because ""is a nokia, is a good Cellphone"" when I received this phone, on the top of the front mask have a little open. then I insert a micro SD and the phone no work properly, when I press on the top of front mask phone display ""REMOVE MEMORY CARD AND PRESS OK""I lost maybe ten o more pictures take with the camera, the phone erase it. I NOT RECOMMEND THIS PHONE.

No. 992

Review is split by Pros, Cons, and Negligible Factors (to me, but maybe not to you) -- I've had the Droid for 2 weeks now, and already can't imagine life without it. Overall, I give it an 8 out of 10. I also have an iPod Touch (how I became familiar with iPhone's platform), and an ancient work BlackBerry -- I count the Droid as my first ""real"" smartphone. THINGS I LIKE- Total gmail, calendar, maps, google-happy-sync world: All you do is enter your gmail address and password, and boom.

Your whole life is on there (assuming you live on google apps).- Android 2.0: Super solid. App switching is AWESOME. Much like google itself, I didn't realize how much I needed it until I had it. Today on the Metro I was looking up a Mark Bittman recipe and while that was loading, I sent an email and checked my calendar. To be able to do all that at once is a huge advantage the Android platform has over iPhone. Also background updating is nice.- Service: It works on the DC Metro. It works everywhere. And it's pretty fast. Typically less than 10-15 sec for page loads.- Design: It's only 1mm thicker than iPhone, and about an ounce heavier. I was pleasantly surprised by how portable it is.- Battery life: A single charge powers a day of regular use, easy -- 2 days with light use.- Screen: I'm not a pixel snob by any means, but the screen is GORGEOUS. Resolution and image quality are fantastic. The touch screen keyboard is a pleasant surprise -- autofill is as good, if not better, than the one on my iPod Touch (same as iPhone's). I use it way more than I expected, partly because it's so easy to use, partly because the keyboard is so bad (more on that later).- Sound quality: I'm not much of a phone person, but my family is -- talked to my mom for 20 minutes today. Everything she said was crystal clear.- Price: $149 with my New Every 2 discount. That's a great deal. Monthly bill is about $100, which is expensive, but standard for smart phones, and worth it to me. THINGS I DON'T LIKE- Keyboard: Surprisingly bad. Flat keys, not very responsive, which makes it hard to type quickly w/o mistakes. And I have small, nimble Asian fingers. Maybe I just need more practice? - (For Outlook/Exchange users): No ""search"" function in the Droid's internal email client! There's a search in the gmail app, but if you aren't using gmail, you'll want to throw this phone into a wall. Alternately, you can download a $20 app by Touchdown that will fix this. Still, WTF?! That's a crazy oversight.- Camera: It's OK, but hard to focus. I expected better from a 5MP w/flash. I looked it up online, and other users had similar complaints, but expect it to be fixed with a software update.- On/off/sleep switch: Why is this located on a tiny button on the top right and not on the front? This is the only flaw of the design, but it's a pretty annoying one, seeing as you use the on/off/sleep switch so often. THINGS I DON'T CARE ABOUT, BUT YOU MIGHT- Music player: Music menu isn't great (though sound quality is), but I don't use phones as MP3 players anyway. I prefer my hot pink ipod nano.- App selection: There are enough to keep me happy, and plenty of freebies. Recommend Spare Parts to keep tabs of all your apps and their battery drain. Bar code scanner is included too. [Side note: I know Apple has 140K apps for iPhone/iPod Touch, but honestly, I've never cared much for them. I'm generally against useless time-sucks like the bubble wrap game, or the one that turns your iPhone into a flute or other wind instrument. Like, that was fun for the first 5 minutes, um, now what?]- Video camera: All the reviews say the video recorder is excellent, but I never use it.- Storage: 16gb is more than enough for me. You can add up to 32gb total.

No. 995

The Nokia N800 is a very handy internet tablet. The only downside is that it cannot open attachments for microsoft word, excel or powerpoint.

Questionnaire 2 and responses

What are the reasons helping you to decide post No. 145 is helpful? Rank them from most important one to the least important one.

1. *Reception Function*
2. *Software*
3. *Text Feature*
4. *Screen*
5. *Camera*
6. *Storage*

What are the reasons helping you to decide post No. 246 is NOT helpful? Rank them from most important one to the least important one.

The comment only mentions the customer used the brand of phone.

What are the reasons helping you to decide post No. 503 is NOT helpful? Rank them from most important one to the least important one.

The comment is just talk about a few words on the function and it is not clearly know the customer mention.

What are the reasons helping you to decide post No. 742 is NOT helpful? Rank them from most important one to the least important one.

The comment only focuses on the phone battery cell which is not useful for the designer to fully know the problem of the previous phone.

What are the reasons helping you to decide post No. 763 is NOT helpful? Rank them from most important one to the least important one.

The comment only focuses on the phone battery cover which is not useful for the designer to fully know the problem of the previous phone.

What are the reasons helping you to decide post No. 852 is NOT helpful? Rank them from most important one to the least important one.

The comment only focuses on the problem of the memory card which is not useful for the designer to fully know the problem of the previous phone.

What are the reasons helping you to decide post No. 992 is NOT helpful? Rank them from most important one to the least important one.

1. *Reception Function*
2. *Software*
3. *Price on their plan*

What are the reasons helping you to decide post No. 995 is NOT helpful? Rank them from most important one to the least important one.

The comment only focuses on the Microsoft office function which is not useful for the designer to fully know the problem of the previous phone.

APPENDIX II

LINGO MODELS FOR TRANSFORMING THE RESULTS TO THE ORIGINIAL CUSTOMER SATISFACTION RATING

Lingo Model for A810 Dataset

```
model:
sets:
        obj/1..46/:y;
        flag1/1..383/:a;
        flag2/1..253/:b;
        flag3/1..399/:c;
endsets
        min=@sum(flag1:a)+@sum(flag2:b)+@sum(flag3:c);

        @for(obj(p):
                @bnd(1,y(p),5);
                @gin(y(p));
        );
        @for(flag1(q1):
                @bin(a(q1));
        );
        @for(flag2(q2):
                @bin(b(q2));
        );
        @for(flag3(q3):
                @bin(c(q3));
        );

        y(4)-y(3)>=1-M*a(1);            y(3)-y(4)>=1-M*(1-a(1));
        y(5)-y(3)>=1-M*a(2);            y(3)-y(5)>=1-M*(1-a(2));
        y(6)-y(3)>=1-M*a(3);            y(3)-y(6)>=1-M*(1-a(3));
        y(6)-y(4)>=1-M*a(4);            y(4)-y(6)>=1-M*(1-a(4));
        y(6)-y(5)>=1-M*a(5);            y(5)-y(6)>=1-M*(1-a(5));
        y(7)-y(3)>=1-M*a(6);            y(3)-y(7)>=1-M*(1-a(6));
        y(7)-y(4)>=1-M*a(7);            y(4)-y(7)>=1-M*(1-a(7));
        y(7)-y(5)>=1-M*a(8);            y(5)-y(7)>=1-M*(1-a(8));
        y(7)-y(6)>=1-M*a(9);            y(6)-y(7)>=1-M*(1-a(9));
        y(9)-y(2)>=1-M*a(10);           y(2)-y(9)>=1-M*(1-a(10));
        y(9)-y(3)>=1-M*a(11);           y(3)-y(9)>=1-M*(1-a(11));
        y(9)-y(4)>=1-M*a(12);           y(4)-y(9)>=1-M*(1-a(12));
```

```
y(9)-y(5)>=1-M*a(13);
y(9)-y(6)>=1-M*a(14);
y(9)-y(7)>=1-M*a(15);
y(9)-y(8)>=1-M*a(16);
y(10)-y(8)>=1-M*a(17);
y(11)-y(3)>=1-M*a(18);
y(11)-y(8)>=1-M*a(19);
y(11)-y(10)>=1-M*a(20);
y(12)-y(3)>=1-M*a(21);
y(12)-y(8)>=1-M*a(22);
y(12)-y(10)>=1-M*a(23);
y(12)-y(11)>=1-M*a(24);
y(14)-y(13)>=1-M*a(25);
y(15)-y(3)>=1-M*a(26);
y(15)-y(5)>=1-M*a(27);
y(15)-y(8)>=1-M*a(28);
y(15)-y(10)>=1-M*a(29);
y(15)-y(13)>=1-M*a(30);
y(15)-y(14)>=1-M*a(31);
y(16)-y(2)>=1-M*a(32);
y(16)-y(3)>=1-M*a(33);
y(16)-y(4)>=1-M*a(34);
y(16)-y(5)>=1-M*a(35);
y(16)-y(6)>=1-M*a(36);
y(16)-y(7)>=1-M*a(37);
y(16)-y(8)>=1-M*a(38);
y(16)-y(9)>=1-M*a(39);
y(16)-y(10)>=1-M*a(40);
y(16)-y(11)>=1-M*a(41);
y(16)-y(12)>=1-M*a(42);
y(16)-y(13)>=1-M*a(43);
y(16)-y(14)>=1-M*a(44);
y(16)-y(15)>=1-M*a(45);
y(17)-y(1)>=1-M*a(46);
y(17)-y(2)>=1-M*a(47);
y(17)-y(3)>=1-M*a(48);
y(17)-y(4)>=1-M*a(49);
y(17)-y(5)>=1-M*a(50);
y(17)-y(6)>=1-M*a(51);
y(17)-y(7)>=1-M*a(52);
y(17)-y(8)>=1-M*a(53);
y(17)-y(9)>=1-M*a(54);
y(17)-y(10)>=1-M*a(55);
y(17)-y(11)>=1-M*a(56);
y(17)-y(12)>=1-M*a(57);
y(17)-y(13)>=1-M*a(58);
y(5)-y(9)>=1-M*(1-a(13));
y(6)-y(9)>=1-M*(1-a(14));
y(7)-y(9)>=1-M*(1-a(15));
y(8)-y(9)>=1-M*(1-a(16));
y(8)-y(10)>=1-M*(1-a(17));
y(3)-y(11)>=1-M*(1-a(18));
y(8)-y(11)>=1-M*(1-a(19));
y(10)-y(11)>=1-M*(1-a(20));
y(3)-y(12)>=1-M*(1-a(21));
y(8)-y(12)>=1-M*(1-a(22));
y(10)-y(12)>=1-M*(1-a(23));
y(11)-y(12)>=1-M*(1-a(24));
y(13)-y(14)>=1-M*(1-a(25));
y(3)-y(15)>=1-M*(1-a(26));
y(5)-y(15)>=1-M*(1-a(27));
y(8)-y(15)>=1-M*(1-a(28));
y(10)-y(15)>=1-M*(1-a(29));
y(13)-y(15)>=1-M*(1-a(30));
y(14)-y(15)>=1-M*(1-a(31));
y(2)-y(16)>=1-M*(1-a(32));
y(3)-y(16)>=1-M*(1-a(33));
y(4)-y(16)>=1-M*(1-a(34));
y(5)-y(16)>=1-M*(1-a(35));
y(6)-y(16)>=1-M*(1-a(36));
y(7)-y(16)>=1-M*(1-a(37));
y(8)-y(16)>=1-M*(1-a(38));
y(9)-y(16)>=1-M*(1-a(39));
y(10)-y(16)>=1-M*(1-a(40));
y(11)-y(16)>=1-M*(1-a(41));
y(12)-y(16)>=1-M*(1-a(42));
y(13)-y(16)>=1-M*(1-a(43));
y(14)-y(16)>=1-M*(1-a(44));
y(15)-y(16)>=1-M*(1-a(45));
y(1)-y(17)>=1-M*(1-a(46));
y(2)-y(17)>=1-M*(1-a(47));
y(3)-y(17)>=1-M*(1-a(48));
y(4)-y(17)>=1-M*(1-a(49));
y(5)-y(17)>=1-M*(1-a(50));
y(6)-y(17)>=1-M*(1-a(51));
y(7)-y(17)>=1-M*(1-a(52));
y(8)-y(17)>=1-M*(1-a(53));
y(9)-y(17)>=1-M*(1-a(54));
y(10)-y(17)>=1-M*(1-a(55));
y(11)-y(17)>=1-M*(1-a(56));
y(12)-y(17)>=1-M*(1-a(57));
y(13)-y(17)>=1-M*(1-a(58));
```

```
y(17)-y(14)>=1-M*a(59);     y(14)-y(17)>=1-M*(1-a(59));
y(17)-y(15)>=1-M*a(60);     y(15)-y(17)>=1-M*(1-a(60));
y(17)-y(16)>=1-M*a(61);     y(16)-y(17)>=1-M*(1-a(61));
y(18)-y(3)>=1-M*a(62);      y(3)-y(18)>=1-M*(1-a(62));
y(18)-y(4)>=1-M*a(63);      y(4)-y(18)>=1-M*(1-a(63));
y(18)-y(5)>=1-M*a(64);      y(5)-y(18)>=1-M*(1-a(64));
y(18)-y(6)>=1-M*a(65);      y(6)-y(18)>=1-M*(1-a(65));
y(18)-y(7)>=1-M*a(66);      y(7)-y(18)>=1-M*(1-a(66));
y(18)-y(8)>=1-M*a(67);      y(8)-y(18)>=1-M*(1-a(67));
y(18)-y(10)>=1-M*a(68);     y(10)-y(18)>=1-M*(1-a(68));
y(18)-y(11)>=1-M*a(69);     y(11)-y(18)>=1-M*(1-a(69));
y(18)-y(12)>=1-M*a(70);     y(12)-y(18)>=1-M*(1-a(70));
y(18)-y(13)>=1-M*a(71);     y(13)-y(18)>=1-M*(1-a(71));
y(18)-y(14)>=1-M*a(72);     y(14)-y(18)>=1-M*(1-a(72));
y(18)-y(15)>=1-M*a(73);     y(15)-y(18)>=1-M*(1-a(73));
y(19)-y(13)>=1-M*a(74);     y(13)-y(19)>=1-M*(1-a(74));
y(19)-y(14)>=1-M*a(75);     y(14)-y(19)>=1-M*(1-a(75));
y(20)-y(2)>=1-M*a(76);      y(2)-y(20)>=1-M*(1-a(76));
y(20)-y(3)>=1-M*a(77);      y(3)-y(20)>=1-M*(1-a(77));
y(20)-y(4)>=1-M*a(78);      y(4)-y(20)>=1-M*(1-a(78));
y(20)-y(5)>=1-M*a(79);      y(5)-y(20)>=1-M*(1-a(79));
y(20)-y(6)>=1-M*a(80);      y(6)-y(20)>=1-M*(1-a(80));
y(20)-y(7)>=1-M*a(81);      y(7)-y(20)>=1-M*(1-a(81));
y(20)-y(8)>=1-M*a(82);      y(8)-y(20)>=1-M*(1-a(82));
y(20)-y(10)>=1-M*a(83);     y(10)-y(20)>=1-M*(1-a(83));
y(20)-y(11)>=1-M*a(84);     y(11)-y(20)>=1-M*(1-a(84));
y(20)-y(12)>=1-M*a(85);     y(12)-y(20)>=1-M*(1-a(85));
y(20)-y(13)>=1-M*a(86);     y(13)-y(20)>=1-M*(1-a(86));
y(20)-y(14)>=1-M*a(87);     y(14)-y(20)>=1-M*(1-a(87));
y(20)-y(15)>=1-M*a(88);     y(15)-y(20)>=1-M*(1-a(88));
y(20)-y(18)>=1-M*a(89);     y(18)-y(20)>=1-M*(1-a(89));
y(20)-y(19)>=1-M*a(90);     y(19)-y(20)>=1-M*(1-a(90));
y(21)-y(13)>=1-M*a(91);     y(13)-y(21)>=1-M*(1-a(91));
y(21)-y(14)>=1-M*a(92);     y(14)-y(21)>=1-M*(1-a(92));
y(22)-y(8)>=1-M*a(93);      y(8)-y(22)>=1-M*(1-a(93));
y(22)-y(10)>=1-M*a(94);     y(10)-y(22)>=1-M*(1-a(94));
y(22)-y(13)>=1-M*a(95);     y(13)-y(22)>=1-M*(1-a(95));
y(22)-y(14)>=1-M*a(96);     y(14)-y(22)>=1-M*(1-a(96));
y(22)-y(21)>=1-M*a(97);     y(21)-y(22)>=1-M*(1-a(97));
y(23)-y(3)>=1-M*a(98);      y(3)-y(23)>=1-M*(1-a(98));
y(23)-y(5)>=1-M*a(99);      y(5)-y(23)>=1-M*(1-a(99));
y(23)-y(8)>=1-M*a(100);     y(8)-y(23)>=1-M*(1-a(100));
y(23)-y(10)>=1-M*a(101);    y(10)-y(23)>=1-M*(1-a(101));
y(23)-y(13)>=1-M*a(102);    y(13)-y(23)>=1-M*(1-a(102));
y(23)-y(14)>=1-M*a(103);    y(14)-y(23)>=1-M*(1-a(103));
y(23)-y(15)>=1-M*a(104);    y(15)-y(23)>=1-M*(1-a(104));
```

```
y(23)-y(19)>=1-M*a(105);      y(19)-y(23)>=1-M*(1-a(105));
y(23)-y(21)>=1-M*a(106);      y(21)-y(23)>=1-M*(1-a(106));
y(23)-y(22)>=1-M*a(107);      y(22)-y(23)>=1-M*(1-a(107));
y(24)-y(3)>=1-M*a(108);       y(3)-y(24)>=1-M*(1-a(108));
y(24)-y(4)>=1-M*a(109);       y(4)-y(24)>=1-M*(1-a(109));
y(24)-y(5)>=1-M*a(110);       y(5)-y(24)>=1-M*(1-a(110));
y(24)-y(6)>=1-M*a(111);       y(6)-y(24)>=1-M*(1-a(111));
y(24)-y(8)>=1-M*a(112);       y(8)-y(24)>=1-M*(1-a(112));
y(24)-y(10)>=1-M*a(113);      y(10)-y(24)>=1-M*(1-a(113));
y(24)-y(11)>=1-M*a(114);      y(11)-y(24)>=1-M*(1-a(114));
y(24)-y(12)>=1-M*a(115);      y(12)-y(24)>=1-M*(1-a(115));
y(24)-y(13)>=1-M*a(116);      y(13)-y(24)>=1-M*(1-a(116));
y(24)-y(14)>=1-M*a(117);      y(14)-y(24)>=1-M*(1-a(117));
y(24)-y(15)>=1-M*a(118);      y(15)-y(24)>=1-M*(1-a(118));
y(24)-y(19)>=1-M*a(119);      y(19)-y(24)>=1-M*(1-a(119));
y(24)-y(21)>=1-M*a(120);      y(21)-y(24)>=1-M*(1-a(120));
y(24)-y(22)>=1-M*a(121);      y(22)-y(24)>=1-M*(1-a(121));
y(24)-y(23)>=1-M*a(122);      y(23)-y(24)>=1-M*(1-a(122));
y(25)-y(2)>=1-M*a(123);       y(2)-y(25)>=1-M*(1-a(123));
y(25)-y(3)>=1-M*a(124);       y(3)-y(25)>=1-M*(1-a(124));
y(25)-y(4)>=1-M*a(125);       y(4)-y(25)>=1-M*(1-a(125));
y(25)-y(5)>=1-M*a(126);       y(5)-y(25)>=1-M*(1-a(126));
y(25)-y(6)>=1-M*a(127);       y(6)-y(25)>=1-M*(1-a(127));
y(25)-y(7)>=1-M*a(128);       y(7)-y(25)>=1-M*(1-a(128));
y(25)-y(8)>=1-M*a(129);       y(8)-y(25)>=1-M*(1-a(129));
y(25)-y(9)>=1-M*a(130);       y(9)-y(25)>=1-M*(1-a(130));
y(25)-y(10)>=1-M*a(131);      y(10)-y(25)>=1-M*(1-a(131));
y(25)-y(11)>=1-M*a(132);      y(11)-y(25)>=1-M*(1-a(132));
y(25)-y(12)>=1-M*a(133);      y(12)-y(25)>=1-M*(1-a(133));
y(25)-y(13)>=1-M*a(134);      y(13)-y(25)>=1-M*(1-a(134));
y(25)-y(14)>=1-M*a(135);      y(14)-y(25)>=1-M*(1-a(135));
y(25)-y(15)>=1-M*a(136);      y(15)-y(25)>=1-M*(1-a(136));
y(25)-y(16)>=1-M*a(137);      y(16)-y(25)>=1-M*(1-a(137));
y(25)-y(18)>=1-M*a(138);      y(18)-y(25)>=1-M*(1-a(138));
y(25)-y(19)>=1-M*a(139);      y(19)-y(25)>=1-M*(1-a(139));
y(25)-y(20)>=1-M*a(140);      y(20)-y(25)>=1-M*(1-a(140));
y(25)-y(21)>=1-M*a(141);      y(21)-y(25)>=1-M*(1-a(141));
y(25)-y(22)>=1-M*a(142);      y(22)-y(25)>=1-M*(1-a(142));
y(25)-y(23)>=1-M*a(143);      y(23)-y(25)>=1-M*(1-a(143));
y(25)-y(24)>=1-M*a(144);      y(24)-y(25)>=1-M*(1-a(144));
y(26)-y(13)>=1-M*a(145);      y(13)-y(26)>=1-M*(1-a(145));
y(26)-y(14)>=1-M*a(146);      y(14)-y(26)>=1-M*(1-a(146));
y(27)-y(8)>=1-M*a(147);       y(8)-y(27)>=1-M*(1-a(147));
y(27)-y(10)>=1-M*a(148);      y(10)-y(27)>=1-M*(1-a(148));
y(27)-y(13)>=1-M*a(149);      y(13)-y(27)>=1-M*(1-a(149));
y(27)-y(14)>=1-M*a(150);      y(14)-y(27)>=1-M*(1-a(150));
```

```
y(27)-y(21)>=1-M*a(151);    y(21)-y(27)>=1-M*(1-a(151));
y(27)-y(26)>=1-M*a(152);    y(26)-y(27)>=1-M*(1-a(152));
y(28)-y(3)>=1-M*a(153);     y(3)-y(28)>=1-M*(1-a(153));
y(28)-y(4)>=1-M*a(154);     y(4)-y(28)>=1-M*(1-a(154));
y(28)-y(5)>=1-M*a(155);     y(5)-y(28)>=1-M*(1-a(155));
y(28)-y(6)>=1-M*a(156);     y(6)-y(28)>=1-M*(1-a(156));
y(28)-y(8)>=1-M*a(157);     y(8)-y(28)>=1-M*(1-a(157));
y(28)-y(10)>=1-M*a(158);    y(10)-y(28)>=1-M*(1-a(158));
y(28)-y(11)>=1-M*a(159);    y(11)-y(28)>=1-M*(1-a(159));
y(28)-y(12)>=1-M*a(160);    y(12)-y(28)>=1-M*(1-a(160));
y(28)-y(13)>=1-M*a(161);    y(13)-y(28)>=1-M*(1-a(161));
y(28)-y(14)>=1-M*a(162);    y(14)-y(28)>=1-M*(1-a(162));
y(28)-y(15)>=1-M*a(163);    y(15)-y(28)>=1-M*(1-a(163));
y(28)-y(19)>=1-M*a(164);    y(19)-y(28)>=1-M*(1-a(164));
y(28)-y(21)>=1-M*a(165);    y(21)-y(28)>=1-M*(1-a(165));
y(28)-y(22)>=1-M*a(166);    y(22)-y(28)>=1-M*(1-a(166));
y(28)-y(23)>=1-M*a(167);    y(23)-y(28)>=1-M*(1-a(167));
y(28)-y(24)>=1-M*a(168);    y(24)-y(28)>=1-M*(1-a(168));
y(28)-y(26)>=1-M*a(169);    y(26)-y(28)>=1-M*(1-a(169));
y(28)-y(27)>=1-M*a(170);    y(27)-y(28)>=1-M*(1-a(170));
y(29)-y(3)>=1-M*a(171);     y(3)-y(29)>=1-M*(1-a(171));
y(29)-y(4)>=1-M*a(172);     y(4)-y(29)>=1-M*(1-a(172));
y(29)-y(5)>=1-M*a(173);     y(5)-y(29)>=1-M*(1-a(173));
y(29)-y(6)>=1-M*a(174);     y(6)-y(29)>=1-M*(1-a(174));
y(29)-y(8)>=1-M*a(175);     y(8)-y(29)>=1-M*(1-a(175));
y(29)-y(10)>=1-M*a(176);    y(10)-y(29)>=1-M*(1-a(176));
y(29)-y(11)>=1-M*a(177);    y(11)-y(29)>=1-M*(1-a(177));
y(29)-y(12)>=1-M*a(178);    y(12)-y(29)>=1-M*(1-a(178));
y(29)-y(13)>=1-M*a(179);    y(13)-y(29)>=1-M*(1-a(179));
y(29)-y(14)>=1-M*a(180);    y(14)-y(29)>=1-M*(1-a(180));
y(29)-y(15)>=1-M*a(181);    y(15)-y(29)>=1-M*(1-a(181));
y(29)-y(19)>=1-M*a(182);    y(19)-y(29)>=1-M*(1-a(182));
y(29)-y(21)>=1-M*a(183);    y(21)-y(29)>=1-M*(1-a(183));
y(29)-y(22)>=1-M*a(184);    y(22)-y(29)>=1-M*(1-a(184));
y(29)-y(23)>=1-M*a(185);    y(23)-y(29)>=1-M*(1-a(185));
y(29)-y(24)>=1-M*a(186);    y(24)-y(29)>=1-M*(1-a(186));
y(29)-y(26)>=1-M*a(187);    y(26)-y(29)>=1-M*(1-a(187));
y(29)-y(27)>=1-M*a(188);    y(27)-y(29)>=1-M*(1-a(188));
y(30)-y(13)>=1-M*a(189);    y(13)-y(30)>=1-M*(1-a(189));
y(30)-y(14)>=1-M*a(190);    y(14)-y(30)>=1-M*(1-a(190));
y(30)-y(26)>=1-M*a(191);    y(26)-y(30)>=1-M*(1-a(191));
y(31)-y(13)>=1-M*a(192);    y(13)-y(31)>=1-M*(1-a(192));
y(31)-y(14)>=1-M*a(193);    y(14)-y(31)>=1-M*(1-a(193));
y(31)-y(21)>=1-M*a(194);    y(21)-y(31)>=1-M*(1-a(194));
y(32)-y(13)>=1-M*a(195);    y(13)-y(32)>=1-M*(1-a(195));
y(32)-y(14)>=1-M*a(196);    y(14)-y(32)>=1-M*(1-a(196));
```

y(33)-y(3)>=1-M*a(197); y(3)-y(33)>=1-M*(1-a(197));
y(33)-y(5)>=1-M*a(198); y(5)-y(33)>=1-M*(1-a(198));
y(33)-y(8)>=1-M*a(199); y(8)-y(33)>=1-M*(1-a(199));
y(33)-y(10)>=1-M*a(200); y(10)-y(33)>=1-M*(1-a(200));
y(33)-y(13)>=1-M*a(201); y(13)-y(33)>=1-M*(1-a(201));
y(33)-y(14)>=1-M*a(202); y(14)-y(33)>=1-M*(1-a(202));
y(33)-y(15)>=1-M*a(203); y(15)-y(33)>=1-M*(1-a(203));
y(33)-y(19)>=1-M*a(204); y(19)-y(33)>=1-M*(1-a(204));
y(33)-y(21)>=1-M*a(205); y(21)-y(33)>=1-M*(1-a(205));
y(33)-y(22)>=1-M*a(206); y(22)-y(33)>=1-M*(1-a(206));
y(33)-y(23)>=1-M*a(207); y(23)-y(33)>=1-M*(1-a(207));
y(33)-y(26)>=1-M*a(208); y(26)-y(33)>=1-M*(1-a(208));
y(33)-y(27)>=1-M*a(209); y(27)-y(33)>=1-M*(1-a(209));
y(33)-y(30)>=1-M*a(210); y(30)-y(33)>=1-M*(1-a(210));
y(33)-y(31)>=1-M*a(211); y(31)-y(33)>=1-M*(1-a(211));
y(33)-y(32)>=1-M*a(212); y(32)-y(33)>=1-M*(1-a(212));
y(34)-y(3)>=1-M*a(213); y(3)-y(34)>=1-M*(1-a(213));
y(34)-y(5)>=1-M*a(214); y(5)-y(34)>=1-M*(1-a(214));
y(34)-y(8)>=1-M*a(215); y(8)-y(34)>=1-M*(1-a(215));
y(34)-y(10)>=1-M*a(216); y(10)-y(34)>=1-M*(1-a(216));
y(34)-y(13)>=1-M*a(217); y(13)-y(34)>=1-M*(1-a(217));
y(34)-y(14)>=1-M*a(218); y(14)-y(34)>=1-M*(1-a(218));
y(34)-y(15)>=1-M*a(219); y(15)-y(34)>=1-M*(1-a(219));
y(34)-y(19)>=1-M*a(220); y(19)-y(34)>=1-M*(1-a(220));
y(34)-y(21)>=1-M*a(221); y(21)-y(34)>=1-M*(1-a(221));
y(34)-y(22)>=1-M*a(222); y(22)-y(34)>=1-M*(1-a(222));
y(34)-y(23)>=1-M*a(223); y(23)-y(34)>=1-M*(1-a(223));
y(34)-y(26)>=1-M*a(224); y(26)-y(34)>=1-M*(1-a(224));
y(34)-y(27)>=1-M*a(225); y(27)-y(34)>=1-M*(1-a(225));
y(34)-y(30)>=1-M*a(226); y(30)-y(34)>=1-M*(1-a(226));
y(34)-y(31)>=1-M*a(227); y(31)-y(34)>=1-M*(1-a(227));
y(34)-y(32)>=1-M*a(228); y(32)-y(34)>=1-M*(1-a(228));
y(34)-y(33)>=1-M*a(229); y(33)-y(34)>=1-M*(1-a(229));
y(35)-y(3)>=1-M*a(230); y(3)-y(35)>=1-M*(1-a(230));
y(35)-y(8)>=1-M*a(231); y(8)-y(35)>=1-M*(1-a(231));
y(35)-y(10)>=1-M*a(232); y(10)-y(35)>=1-M*(1-a(232));
y(35)-y(11)>=1-M*a(233); y(11)-y(35)>=1-M*(1-a(233));
y(35)-y(12)>=1-M*a(234); y(12)-y(35)>=1-M*(1-a(234));
y(35)-y(13)>=1-M*a(235); y(13)-y(35)>=1-M*(1-a(235));
y(35)-y(14)>=1-M*a(236); y(14)-y(35)>=1-M*(1-a(236));
y(35)-y(21)>=1-M*a(237); y(21)-y(35)>=1-M*(1-a(237));
y(35)-y(22)>=1-M*a(238); y(22)-y(35)>=1-M*(1-a(238));
y(35)-y(26)>=1-M*a(239); y(26)-y(35)>=1-M*(1-a(239));
y(35)-y(27)>=1-M*a(240); y(27)-y(35)>=1-M*(1-a(240));
y(35)-y(30)>=1-M*a(241); y(30)-y(35)>=1-M*(1-a(241));
y(35)-y(31)>=1-M*a(242); y(31)-y(35)>=1-M*(1-a(242));

```
y(35)-y(32)>=1-M*a(243);     y(32)-y(35)>=1-M*(1-a(243));
y(36)-y(3)>=1-M*a(244);      y(3)-y(36)>=1-M*(1-a(244));
y(36)-y(4)>=1-M*a(245);      y(4)-y(36)>=1-M*(1-a(245));
y(36)-y(5)>=1-M*a(246);      y(5)-y(36)>=1-M*(1-a(246));
y(36)-y(6)>=1-M*a(247);      y(6)-y(36)>=1-M*(1-a(247));
y(36)-y(8)>=1-M*a(248);      y(8)-y(36)>=1-M*(1-a(248));
y(36)-y(10)>=1-M*a(249);     y(10)-y(36)>=1-M*(1-a(249));
y(36)-y(11)>=1-M*a(250);     y(11)-y(36)>=1-M*(1-a(250));
y(36)-y(12)>=1-M*a(251);     y(12)-y(36)>=1-M*(1-a(251));
y(36)-y(13)>=1-M*a(252);     y(13)-y(36)>=1-M*(1-a(252));
y(36)-y(14)>=1-M*a(253);     y(14)-y(36)>=1-M*(1-a(253));
y(36)-y(15)>=1-M*a(254);     y(15)-y(36)>=1-M*(1-a(254));
y(36)-y(19)>=1-M*a(255);     y(19)-y(36)>=1-M*(1-a(255));
y(36)-y(21)>=1-M*a(256);     y(21)-y(36)>=1-M*(1-a(256));
y(36)-y(22)>=1-M*a(257);     y(22)-y(36)>=1-M*(1-a(257));
y(36)-y(23)>=1-M*a(258);     y(23)-y(36)>=1-M*(1-a(258));
y(36)-y(24)>=1-M*a(259);     y(24)-y(36)>=1-M*(1-a(259));
y(36)-y(26)>=1-M*a(260);     y(26)-y(36)>=1-M*(1-a(260));
y(36)-y(27)>=1-M*a(261);     y(27)-y(36)>=1-M*(1-a(261));
y(36)-y(28)>=1-M*a(262);     y(28)-y(36)>=1-M*(1-a(262));
y(36)-y(29)>=1-M*a(263);     y(29)-y(36)>=1-M*(1-a(263));
y(36)-y(30)>=1-M*a(264);     y(30)-y(36)>=1-M*(1-a(264));
y(36)-y(31)>=1-M*a(265);     y(31)-y(36)>=1-M*(1-a(265));
y(36)-y(32)>=1-M*a(266);     y(32)-y(36)>=1-M*(1-a(266));
y(36)-y(33)>=1-M*a(267);     y(33)-y(36)>=1-M*(1-a(267));
y(36)-y(34)>=1-M*a(268);     y(34)-y(36)>=1-M*(1-a(268));
y(36)-y(35)>=1-M*a(269);     y(35)-y(36)>=1-M*(1-a(269));
y(37)-y(8)>=1-M*a(270);      y(8)-y(37)>=1-M*(1-a(270));
y(37)-y(10)>=1-M*a(271);     y(10)-y(37)>=1-M*(1-a(271));
y(37)-y(13)>=1-M*a(272);     y(13)-y(37)>=1-M*(1-a(272));
y(37)-y(14)>=1-M*a(273);     y(14)-y(37)>=1-M*(1-a(273));
y(37)-y(19)>=1-M*a(274);     y(19)-y(37)>=1-M*(1-a(274));
y(37)-y(21)>=1-M*a(275);     y(21)-y(37)>=1-M*(1-a(275));
y(37)-y(26)>=1-M*a(276);     y(26)-y(37)>=1-M*(1-a(276));
y(37)-y(27)>=1-M*a(277);     y(27)-y(37)>=1-M*(1-a(277));
y(37)-y(31)>=1-M*a(278);     y(31)-y(37)>=1-M*(1-a(278));
y(37)-y(32)>=1-M*a(279);     y(32)-y(37)>=1-M*(1-a(279));
y(38)-y(13)>=1-M*a(280);     y(13)-y(38)>=1-M*(1-a(280));
y(38)-y(14)>=1-M*a(281);     y(14)-y(38)>=1-M*(1-a(281));
y(39)-y(8)>=1-M*a(282);      y(8)-y(39)>=1-M*(1-a(282));
y(39)-y(10)>=1-M*a(283);     y(10)-y(39)>=1-M*(1-a(283));
y(39)-y(13)>=1-M*a(284);     y(13)-y(39)>=1-M*(1-a(284));
y(39)-y(14)>=1-M*a(285);     y(14)-y(39)>=1-M*(1-a(285));
y(39)-y(21)>=1-M*a(286);     y(21)-y(39)>=1-M*(1-a(286));
y(39)-y(22)>=1-M*a(287);     y(22)-y(39)>=1-M*(1-a(287));
y(39)-y(26)>=1-M*a(288);     y(26)-y(39)>=1-M*(1-a(288));
```

```
y(39)-y(27)>=1-M*a(289);     y(27)-y(39)>=1-M*(1-a(289));
y(39)-y(30)>=1-M*a(290);     y(30)-y(39)>=1-M*(1-a(290));
y(39)-y(31)>=1-M*a(291);     y(31)-y(39)>=1-M*(1-a(291));
y(39)-y(32)>=1-M*a(292);     y(32)-y(39)>=1-M*(1-a(292));
y(40)-y(8)>=1-M*a(293);      y(8)-y(40)>=1-M*(1-a(293));
y(40)-y(10)>=1-M*a(294);     y(10)-y(40)>=1-M*(1-a(294));
y(40)-y(13)>=1-M*a(295);     y(13)-y(40)>=1-M*(1-a(295));
y(40)-y(14)>=1-M*a(296);     y(14)-y(40)>=1-M*(1-a(296));
y(40)-y(19)>=1-M*a(297);     y(19)-y(40)>=1-M*(1-a(297));
y(40)-y(21)>=1-M*a(298);     y(21)-y(40)>=1-M*(1-a(298));
y(40)-y(22)>=1-M*a(299);     y(22)-y(40)>=1-M*(1-a(299));
y(40)-y(26)>=1-M*a(300);     y(26)-y(40)>=1-M*(1-a(300));
y(40)-y(27)>=1-M*a(301);     y(27)-y(40)>=1-M*(1-a(301));
y(40)-y(31)>=1-M*a(302);     y(31)-y(40)>=1-M*(1-a(302));
y(40)-y(32)>=1-M*a(303);     y(32)-y(40)>=1-M*(1-a(303));
y(40)-y(37)>=1-M*a(304);     y(37)-y(40)>=1-M*(1-a(304));
y(40)-y(38)>=1-M*a(305);     y(38)-y(40)>=1-M*(1-a(305));
y(41)-y(3)>=1-M*a(306);      y(3)-y(41)>=1-M*(1-a(306));
y(41)-y(8)>=1-M*a(307);      y(8)-y(41)>=1-M*(1-a(307));
y(41)-y(10)>=1-M*a(308);     y(10)-y(41)>=1-M*(1-a(308));
y(41)-y(11)>=1-M*a(309);     y(11)-y(41)>=1-M*(1-a(309));
y(41)-y(12)>=1-M*a(310);     y(12)-y(41)>=1-M*(1-a(310));
y(41)-y(13)>=1-M*a(311);     y(13)-y(41)>=1-M*(1-a(311));
y(41)-y(14)>=1-M*a(312);     y(14)-y(41)>=1-M*(1-a(312));
y(41)-y(21)>=1-M*a(313);     y(21)-y(41)>=1-M*(1-a(313));
y(41)-y(22)>=1-M*a(314);     y(22)-y(41)>=1-M*(1-a(314));
y(41)-y(26)>=1-M*a(315);     y(26)-y(41)>=1-M*(1-a(315));
y(41)-y(27)>=1-M*a(316);     y(27)-y(41)>=1-M*(1-a(316));
y(41)-y(30)>=1-M*a(317);     y(30)-y(41)>=1-M*(1-a(317));
y(41)-y(31)>=1-M*a(318);     y(31)-y(41)>=1-M*(1-a(318));
y(41)-y(32)>=1-M*a(319);     y(32)-y(41)>=1-M*(1-a(319));
y(41)-y(35)>=1-M*a(320);     y(35)-y(41)>=1-M*(1-a(320));
y(41)-y(39)>=1-M*a(321);     y(39)-y(41)>=1-M*(1-a(321));
y(42)-y(8)>=1-M*a(322);      y(8)-y(42)>=1-M*(1-a(322));
y(42)-y(10)>=1-M*a(323);     y(10)-y(42)>=1-M*(1-a(323));
y(42)-y(13)>=1-M*a(324);     y(13)-y(42)>=1-M*(1-a(324));
y(42)-y(14)>=1-M*a(325);     y(14)-y(42)>=1-M*(1-a(325));
y(42)-y(19)>=1-M*a(326);     y(19)-y(42)>=1-M*(1-a(326));
y(42)-y(21)>=1-M*a(327);     y(21)-y(42)>=1-M*(1-a(327));
y(42)-y(26)>=1-M*a(328);     y(26)-y(42)>=1-M*(1-a(328));
y(42)-y(27)>=1-M*a(329);     y(27)-y(42)>=1-M*(1-a(329));
y(42)-y(31)>=1-M*a(330);     y(31)-y(42)>=1-M*(1-a(330));
y(42)-y(32)>=1-M*a(331);     y(32)-y(42)>=1-M*(1-a(331));
y(42)-y(37)>=1-M*a(332);     y(37)-y(42)>=1-M*(1-a(332));
y(42)-y(38)>=1-M*a(333);     y(38)-y(42)>=1-M*(1-a(333));
y(43)-y(13)>=1-M*a(334);     y(13)-y(43)>=1-M*(1-a(334));
```

```
y(43)-y(14)>=1-M*a(335);
y(43)-y(21)>=1-M*a(336);
y(43)-y(31)>=1-M*a(337);
y(43)-y(32)>=1-M*a(338);
y(44)-y(3)>=1-M*a(339);
y(44)-y(5)>=1-M*a(340);
y(44)-y(8)>=1-M*a(341);
y(44)-y(10)>=1-M*a(342);
y(44)-y(13)>=1-M*a(343);
y(44)-y(14)>=1-M*a(344);
y(44)-y(19)>=1-M*a(345);
y(44)-y(21)>=1-M*a(346);
y(44)-y(22)>=1-M*a(347);
y(44)-y(26)>=1-M*a(348);
y(44)-y(27)>=1-M*a(349);
y(44)-y(30)>=1-M*a(350);
y(44)-y(31)>=1-M*a(351);
y(44)-y(32)>=1-M*a(352);
y(44)-y(37)>=1-M*a(353);
y(44)-y(38)>=1-M*a(354);
y(44)-y(39)>=1-M*a(355);
y(44)-y(40)>=1-M*a(356);
y(44)-y(42)>=1-M*a(357);
y(44)-y(43)>=1-M*a(358);
y(45)-y(13)>=1-M*a(359);
y(45)-y(14)>=1-M*a(360);
y(46)-y(3)>=1-M*a(361);
y(46)-y(5)>=1-M*a(362);
y(46)-y(8)>=1-M*a(363);
y(46)-y(10)>=1-M*a(364);
y(46)-y(13)>=1-M*a(365);
y(46)-y(14)>=1-M*a(366);
y(46)-y(15)>=1-M*a(367);
y(46)-y(19)>=1-M*a(368);
y(46)-y(21)>=1-M*a(369);
y(46)-y(22)>=1-M*a(370);
y(46)-y(26)>=1-M*a(371);
y(46)-y(27)>=1-M*a(372);
y(46)-y(30)>=1-M*a(373);
y(46)-y(31)>=1-M*a(374);
y(46)-y(32)>=1-M*a(375);
y(46)-y(37)>=1-M*a(376);
y(46)-y(38)>=1-M*a(377);
y(46)-y(39)>=1-M*a(378);
y(46)-y(40)>=1-M*a(379);
y(46)-y(42)>=1-M*a(380);
y(14)-y(43)>=1-M*(1-a(335));
y(21)-y(43)>=1-M*(1-a(336));
y(31)-y(43)>=1-M*(1-a(337));
y(32)-y(43)>=1-M*(1-a(338));
y(3)-y(44)>=1-M*(1-a(339));
y(5)-y(44)>=1-M*(1-a(340));
y(8)-y(44)>=1-M*(1-a(341));
y(10)-y(44)>=1-M*(1-a(342));
y(13)-y(44)>=1-M*(1-a(343));
y(14)-y(44)>=1-M*(1-a(344));
y(19)-y(44)>=1-M*(1-a(345));
y(21)-y(44)>=1-M*(1-a(346));
y(22)-y(44)>=1-M*(1-a(347));
y(26)-y(44)>=1-M*(1-a(348));
y(27)-y(44)>=1-M*(1-a(349));
y(30)-y(44)>=1-M*(1-a(350));
y(31)-y(44)>=1-M*(1-a(351));
y(32)-y(44)>=1-M*(1-a(352));
y(37)-y(44)>=1-M*(1-a(353));
y(38)-y(44)>=1-M*(1-a(354));
y(39)-y(44)>=1-M*(1-a(355));
y(40)-y(44)>=1-M*(1-a(356));
y(42)-y(44)>=1-M*(1-a(357));
y(43)-y(44)>=1-M*(1-a(358));
y(13)-y(45)>=1-M*(1-a(359));
y(14)-y(45)>=1-M*(1-a(360));
y(3)-y(46)>=1-M*(1-a(361));
y(5)-y(46)>=1-M*(1-a(362));
y(8)-y(46)>=1-M*(1-a(363));
y(10)-y(46)>=1-M*(1-a(364));
y(13)-y(46)>=1-M*(1-a(365));
y(14)-y(46)>=1-M*(1-a(366));
y(15)-y(46)>=1-M*(1-a(367));
y(19)-y(46)>=1-M*(1-a(368));
y(21)-y(46)>=1-M*(1-a(369));
y(22)-y(46)>=1-M*(1-a(370));
y(26)-y(46)>=1-M*(1-a(371));
y(27)-y(46)>=1-M*(1-a(372));
y(30)-y(46)>=1-M*(1-a(373));
y(31)-y(46)>=1-M*(1-a(374));
y(32)-y(46)>=1-M*(1-a(375));
y(37)-y(46)>=1-M*(1-a(376));
y(38)-y(46)>=1-M*(1-a(377));
y(39)-y(46)>=1-M*(1-a(378));
y(40)-y(46)>=1-M*(1-a(379));
y(42)-y(46)>=1-M*(1-a(380));
```

```
y(46)-y(43)>=1-M*a(381);        y(43)-y(46)>=1-M*(1-a(381));
y(46)-y(44)>=1-M*a(382);        y(44)-y(46)>=1-M*(1-a(382));
y(46)-y(45)>=1-M*a(383);        y(45)-y(46)>=1-M*(1-a(383));

y(2)-y(3)>=1-M*b(1);            y(3)-y(2)>=1-M*(1-b(1));
y(2)-y(8)>=1-M*b(2);            y(8)-y(2)>=1-M*(1-b(2));
y(4)-y(8)>=1-M*b(3);            y(8)-y(4)>=1-M*(1-b(3));
y(5)-y(8)>=1-M*b(4);            y(8)-y(5)>=1-M*(1-b(4));
y(6)-y(8)>=1-M*b(5);            y(8)-y(6)>=1-M*(1-b(5));
y(7)-y(8)>=1-M*b(6);            y(8)-y(7)>=1-M*(1-b(6));
y(2)-y(10)>=1-M*b(7);           y(10)-y(2)>=1-M*(1-b(7));
y(4)-y(10)>=1-M*b(8);           y(10)-y(4)>=1-M*(1-b(8));
y(5)-y(10)>=1-M*b(9);           y(10)-y(5)>=1-M*(1-b(9));
y(6)-y(10)>=1-M*b(10);          y(10)-y(6)>=1-M*(1-b(10));
y(7)-y(10)>=1-M*b(11);          y(10)-y(7)>=1-M*(1-b(11));
y(9)-y(10)>=1-M*b(12);          y(10)-y(9)>=1-M*(1-b(12));
y(2)-y(11)>=1-M*b(13);          y(11)-y(2)>=1-M*(1-b(13));
y(4)-y(11)>=1-M*b(14);          y(11)-y(4)>=1-M*(1-b(14));
y(6)-y(11)>=1-M*b(15);          y(11)-y(6)>=1-M*(1-b(15));
y(7)-y(11)>=1-M*b(16);          y(11)-y(7)>=1-M*(1-b(16));
y(9)-y(11)>=1-M*b(17);          y(11)-y(9)>=1-M*(1-b(17));
y(2)-y(12)>=1-M*b(18);          y(12)-y(2)>=1-M*(1-b(18));
y(6)-y(12)>=1-M*b(19);          y(12)-y(6)>=1-M*(1-b(19));
y(7)-y(12)>=1-M*b(20);          y(12)-y(7)>=1-M*(1-b(20));
y(9)-y(12)>=1-M*b(21);          y(12)-y(9)>=1-M*(1-b(21));
y(1)-y(13)>=1-M*b(22);          y(13)-y(1)>=1-M*(1-b(22));
y(2)-y(13)>=1-M*b(23);          y(13)-y(2)>=1-M*(1-b(23));
y(3)-y(13)>=1-M*b(24);          y(13)-y(3)>=1-M*(1-b(24));
y(4)-y(13)>=1-M*b(25);          y(13)-y(4)>=1-M*(1-b(25));
y(5)-y(13)>=1-M*b(26);          y(13)-y(5)>=1-M*(1-b(26));
y(6)-y(13)>=1-M*b(27);          y(13)-y(6)>=1-M*(1-b(27));
y(7)-y(13)>=1-M*b(28);          y(13)-y(7)>=1-M*(1-b(28));
y(8)-y(13)>=1-M*b(29);          y(13)-y(8)>=1-M*(1-b(29));
y(9)-y(13)>=1-M*b(30);          y(13)-y(9)>=1-M*(1-b(30));
y(10)-y(13)>=1-M*b(31);         y(13)-y(10)>=1-M*(1-b(31));
y(11)-y(13)>=1-M*b(32);         y(13)-y(11)>=1-M*(1-b(32));
y(12)-y(13)>=1-M*b(33);         y(13)-y(12)>=1-M*(1-b(33));
y(1)-y(14)>=1-M*b(34);          y(14)-y(1)>=1-M*(1-b(34));
y(2)-y(14)>=1-M*b(35);          y(14)-y(2)>=1-M*(1-b(35));
y(3)-y(14)>=1-M*b(36);          y(14)-y(3)>=1-M*(1-b(36));
y(4)-y(14)>=1-M*b(37);          y(14)-y(4)>=1-M*(1-b(37));
y(5)-y(14)>=1-M*b(38);          y(14)-y(5)>=1-M*(1-b(38));
y(6)-y(14)>=1-M*b(39);          y(14)-y(6)>=1-M*(1-b(39));
y(7)-y(14)>=1-M*b(40);          y(14)-y(7)>=1-M*(1-b(40));
y(8)-y(14)>=1-M*b(41);          y(14)-y(8)>=1-M*(1-b(41));
y(9)-y(14)>=1-M*b(42);          y(14)-y(9)>=1-M*(1-b(42));
```

```
y(10)-y(14)>=1-M*b(43);
y(11)-y(14)>=1-M*b(44);
y(12)-y(14)>=1-M*b(45);
y(2)-y(21)>=1-M*b(46);
y(4)-y(21)>=1-M*b(47);
y(5)-y(21)>=1-M*b(48);
y(6)-y(21)>=1-M*b(49);
y(7)-y(21)>=1-M*b(50);
y(9)-y(21)>=1-M*b(51);
y(15)-y(21)>=1-M*b(52);
y(16)-y(21)>=1-M*b(53);
y(17)-y(21)>=1-M*b(54);
y(18)-y(21)>=1-M*b(55);
y(19)-y(21)>=1-M*b(56);
y(20)-y(21)>=1-M*b(57);
y(2)-y(22)>=1-M*b(58);
y(4)-y(22)>=1-M*b(59);
y(5)-y(22)>=1-M*b(60);
y(6)-y(22)>=1-M*b(61);
y(7)-y(22)>=1-M*b(62);
y(9)-y(22)>=1-M*b(63);
y(15)-y(22)>=1-M*b(64);
y(16)-y(22)>=1-M*b(65);
y(17)-y(22)>=1-M*b(66);
y(18)-y(22)>=1-M*b(67);
y(20)-y(22)>=1-M*b(68);
y(1)-y(26)>=1-M*b(69);
y(2)-y(26)>=1-M*b(70);
y(3)-y(26)>=1-M*b(71);
y(4)-y(26)>=1-M*b(72);
y(5)-y(26)>=1-M*b(73);
y(6)-y(26)>=1-M*b(74);
y(7)-y(26)>=1-M*b(75);
y(8)-y(26)>=1-M*b(76);
y(9)-y(26)>=1-M*b(77);
y(10)-y(26)>=1-M*b(78);
y(11)-y(26)>=1-M*b(79);
y(12)-y(26)>=1-M*b(80);
y(15)-y(26)>=1-M*b(81);
y(16)-y(26)>=1-M*b(82);
y(17)-y(26)>=1-M*b(83);
y(18)-y(26)>=1-M*b(84);
y(20)-y(26)>=1-M*b(85);
y(22)-y(26)>=1-M*b(86);
y(23)-y(26)>=1-M*b(87);
y(24)-y(26)>=1-M*b(88);
y(14)-y(10)>=1-M*(1-b(43));
y(14)-y(11)>=1-M*(1-b(44));
y(14)-y(12)>=1-M*(1-b(45));
y(21)-y(2)>=1-M*(1-b(46));
y(21)-y(4)>=1-M*(1-b(47));
y(21)-y(5)>=1-M*(1-b(48));
y(21)-y(6)>=1-M*(1-b(49));
y(21)-y(7)>=1-M*(1-b(50));
y(21)-y(9)>=1-M*(1-b(51));
y(21)-y(15)>=1-M*(1-b(52));
y(21)-y(16)>=1-M*(1-b(53));
y(21)-y(17)>=1-M*(1-b(54));
y(21)-y(18)>=1-M*(1-b(55));
y(21)-y(19)>=1-M*(1-b(56));
y(21)-y(20)>=1-M*(1-b(57));
y(22)-y(2)>=1-M*(1-b(58));
y(22)-y(4)>=1-M*(1-b(59));
y(22)-y(5)>=1-M*(1-b(60));
y(22)-y(6)>=1-M*(1-b(61));
y(22)-y(7)>=1-M*(1-b(62));
y(22)-y(9)>=1-M*(1-b(63));
y(22)-y(15)>=1-M*(1-b(64));
y(22)-y(16)>=1-M*(1-b(65));
y(22)-y(17)>=1-M*(1-b(66));
y(22)-y(18)>=1-M*(1-b(67));
y(22)-y(20)>=1-M*(1-b(68));
y(26)-y(1)>=1-M*(1-b(69));
y(26)-y(2)>=1-M*(1-b(70));
y(26)-y(3)>=1-M*(1-b(71));
y(26)-y(4)>=1-M*(1-b(72));
y(26)-y(5)>=1-M*(1-b(73));
y(26)-y(6)>=1-M*(1-b(74));
y(26)-y(7)>=1-M*(1-b(75));
y(26)-y(8)>=1-M*(1-b(76));
y(26)-y(9)>=1-M*(1-b(77));
y(26)-y(10)>=1-M*(1-b(78));
y(26)-y(11)>=1-M*(1-b(79));
y(26)-y(12)>=1-M*(1-b(80));
y(26)-y(15)>=1-M*(1-b(81));
y(26)-y(16)>=1-M*(1-b(82));
y(26)-y(17)>=1-M*(1-b(83));
y(26)-y(18)>=1-M*(1-b(84));
y(26)-y(20)>=1-M*(1-b(85));
y(26)-y(22)>=1-M*(1-b(86));
y(26)-y(23)>=1-M*(1-b(87));
y(26)-y(24)>=1-M*(1-b(88));
```

```
y(25)-y(26)>=1-M*b(89);     y(26)-y(25)>=1-M*(1-b(89));
y(2)-y(27)>=1-M*b(90);      y(27)-y(2)>=1-M*(1-b(90));
y(4)-y(27)>=1-M*b(91);      y(27)-y(4)>=1-M*(1-b(91));
y(5)-y(27)>=1-M*b(92);      y(27)-y(5)>=1-M*(1-b(92));
y(6)-y(27)>=1-M*b(93);      y(27)-y(6)>=1-M*(1-b(93));
y(7)-y(27)>=1-M*b(94);      y(27)-y(7)>=1-M*(1-b(94));
y(9)-y(27)>=1-M*b(95);      y(27)-y(9)>=1-M*(1-b(95));
y(15)-y(27)>=1-M*b(96);     y(27)-y(15)>=1-M*(1-b(96));
y(16)-y(27)>=1-M*b(97);     y(27)-y(16)>=1-M*(1-b(97));
y(17)-y(27)>=1-M*b(98);     y(27)-y(17)>=1-M*(1-b(98));
y(18)-y(27)>=1-M*b(99);     y(27)-y(18)>=1-M*(1-b(99));
y(20)-y(27)>=1-M*b(100);    y(27)-y(20)>=1-M*(1-b(100));
y(23)-y(27)>=1-M*b(101);    y(27)-y(23)>=1-M*(1-b(101));
y(24)-y(27)>=1-M*b(102);    y(27)-y(24)>=1-M*(1-b(102));
y(25)-y(27)>=1-M*b(103);    y(27)-y(25)>=1-M*(1-b(103));
y(1)-y(30)>=1-M*b(104);     y(30)-y(1)>=1-M*(1-b(104));
y(2)-y(30)>=1-M*b(105);     y(30)-y(2)>=1-M*(1-b(105));
y(3)-y(30)>=1-M*b(106);     y(30)-y(3)>=1-M*(1-b(106));
y(4)-y(30)>=1-M*b(107);     y(30)-y(4)>=1-M*(1-b(107));
y(5)-y(30)>=1-M*b(108);     y(30)-y(5)>=1-M*(1-b(108));
y(6)-y(30)>=1-M*b(109);     y(30)-y(6)>=1-M*(1-b(109));
y(7)-y(30)>=1-M*b(110);     y(30)-y(7)>=1-M*(1-b(110));
y(9)-y(30)>=1-M*b(111);     y(30)-y(9)>=1-M*(1-b(111));
y(11)-y(30)>=1-M*b(112);    y(30)-y(11)>=1-M*(1-b(112));
y(12)-y(30)>=1-M*b(113);    y(30)-y(12)>=1-M*(1-b(113));
y(15)-y(30)>=1-M*b(114);    y(30)-y(15)>=1-M*(1-b(114));
y(16)-y(30)>=1-M*b(115);    y(30)-y(16)>=1-M*(1-b(115));
y(17)-y(30)>=1-M*b(116);    y(30)-y(17)>=1-M*(1-b(116));
y(18)-y(30)>=1-M*b(117);    y(30)-y(18)>=1-M*(1-b(117));
y(20)-y(30)>=1-M*b(118);    y(30)-y(20)>=1-M*(1-b(118));
y(23)-y(30)>=1-M*b(119);    y(30)-y(23)>=1-M*(1-b(119));
y(24)-y(30)>=1-M*b(120);    y(30)-y(24)>=1-M*(1-b(120));
y(25)-y(30)>=1-M*b(121);    y(30)-y(25)>=1-M*(1-b(121));
y(28)-y(30)>=1-M*b(122);    y(30)-y(28)>=1-M*(1-b(122));
y(29)-y(30)>=1-M*b(123);    y(30)-y(29)>=1-M*(1-b(123));
y(2)-y(31)>=1-M*b(124);     y(31)-y(2)>=1-M*(1-b(124));
y(4)-y(31)>=1-M*b(125);     y(31)-y(4)>=1-M*(1-b(125));
y(5)-y(31)>=1-M*b(126);     y(31)-y(5)>=1-M*(1-b(126));
y(6)-y(31)>=1-M*b(127);     y(31)-y(6)>=1-M*(1-b(127));
y(7)-y(31)>=1-M*b(128);     y(31)-y(7)>=1-M*(1-b(128));
y(9)-y(31)>=1-M*b(129);     y(31)-y(9)>=1-M*(1-b(129));
y(15)-y(31)>=1-M*b(130);    y(31)-y(15)>=1-M*(1-b(130));
y(16)-y(31)>=1-M*b(131);    y(31)-y(16)>=1-M*(1-b(131));
y(17)-y(31)>=1-M*b(132);    y(31)-y(17)>=1-M*(1-b(132));
y(18)-y(31)>=1-M*b(133);    y(31)-y(18)>=1-M*(1-b(133));
y(19)-y(31)>=1-M*b(134);    y(31)-y(19)>=1-M*(1-b(134));
```

```
y(20)-y(31)>=1-M*b(135);        y(31)-y(20)>=1-M*(1-b(135));
y(23)-y(31)>=1-M*b(136);        y(31)-y(23)>=1-M*(1-b(136));
y(24)-y(31)>=1-M*b(137);        y(31)-y(24)>=1-M*(1-b(137));
y(25)-y(31)>=1-M*b(138);        y(31)-y(25)>=1-M*(1-b(138));
y(28)-y(31)>=1-M*b(139);        y(31)-y(28)>=1-M*(1-b(139));
y(29)-y(31)>=1-M*b(140);        y(31)-y(29)>=1-M*(1-b(140));
y(1)-y(32)>=1-M*b(141);         y(32)-y(1)>=1-M*(1-b(141));
y(2)-y(32)>=1-M*b(142);         y(32)-y(2)>=1-M*(1-b(142));
y(3)-y(32)>=1-M*b(143);         y(32)-y(3)>=1-M*(1-b(143));
y(4)-y(32)>=1-M*b(144);         y(32)-y(4)>=1-M*(1-b(144));
y(5)-y(32)>=1-M*b(145);         y(32)-y(5)>=1-M*(1-b(145));
y(6)-y(32)>=1-M*b(146);         y(32)-y(6)>=1-M*(1-b(146));
y(7)-y(32)>=1-M*b(147);         y(32)-y(7)>=1-M*(1-b(147));
y(8)-y(32)>=1-M*b(148);         y(32)-y(8)>=1-M*(1-b(148));
y(9)-y(32)>=1-M*b(149);         y(32)-y(9)>=1-M*(1-b(149));
y(10)-y(32)>=1-M*b(150);        y(32)-y(10)>=1-M*(1-b(150));
y(11)-y(32)>=1-M*b(151);        y(32)-y(11)>=1-M*(1-b(151));
y(12)-y(32)>=1-M*b(152);        y(32)-y(12)>=1-M*(1-b(152));
y(15)-y(32)>=1-M*b(153);        y(32)-y(15)>=1-M*(1-b(153));
y(16)-y(32)>=1-M*b(154);        y(32)-y(16)>=1-M*(1-b(154));
y(17)-y(32)>=1-M*b(155);        y(32)-y(17)>=1-M*(1-b(155));
y(18)-y(32)>=1-M*b(156);        y(32)-y(18)>=1-M*(1-b(156));
y(19)-y(32)>=1-M*b(157);        y(32)-y(19)>=1-M*(1-b(157));
y(20)-y(32)>=1-M*b(158);        y(32)-y(20)>=1-M*(1-b(158));
y(21)-y(32)>=1-M*b(159);        y(32)-y(21)>=1-M*(1-b(159));
y(22)-y(32)>=1-M*b(160);        y(32)-y(22)>=1-M*(1-b(160));
y(23)-y(32)>=1-M*b(161);        y(32)-y(23)>=1-M*(1-b(161));
y(24)-y(32)>=1-M*b(162);        y(32)-y(24)>=1-M*(1-b(162));
y(25)-y(32)>=1-M*b(163);        y(32)-y(25)>=1-M*(1-b(163));
y(27)-y(32)>=1-M*b(164);        y(32)-y(27)>=1-M*(1-b(164));
y(28)-y(32)>=1-M*b(165);        y(32)-y(28)>=1-M*(1-b(165));
y(29)-y(32)>=1-M*b(166);        y(32)-y(29)>=1-M*(1-b(166));
y(31)-y(32)>=1-M*b(167);        y(32)-y(31)>=1-M*(1-b(167));
y(2)-y(35)>=1-M*b(168);         y(35)-y(2)>=1-M*(1-b(168));
y(7)-y(35)>=1-M*b(169);         y(35)-y(7)>=1-M*(1-b(169));
y(9)-y(35)>=1-M*b(170);         y(35)-y(9)>=1-M*(1-b(170));
y(16)-y(35)>=1-M*b(171);        y(35)-y(16)>=1-M*(1-b(171));
y(17)-y(35)>=1-M*b(172);        y(35)-y(17)>=1-M*(1-b(172));
y(18)-y(35)>=1-M*b(173);        y(35)-y(18)>=1-M*(1-b(173));
y(20)-y(35)>=1-M*b(174);        y(35)-y(20)>=1-M*(1-b(174));
y(25)-y(35)>=1-M*b(175);        y(35)-y(25)>=1-M*(1-b(175));
y(28)-y(35)>=1-M*b(176);        y(35)-y(28)>=1-M*(1-b(176));
y(2)-y(39)>=1-M*b(177);         y(39)-y(2)>=1-M*(1-b(177));
y(4)-y(39)>=1-M*b(178);         y(39)-y(4)>=1-M*(1-b(178));
y(5)-y(39)>=1-M*b(179);         y(39)-y(5)>=1-M*(1-b(179));
y(6)-y(39)>=1-M*b(180);         y(39)-y(6)>=1-M*(1-b(180));
```

```
y(7)-y(39)>=1-M*b(181);     y(39)-y(7)>=1-M*(1-b(181));
y(9)-y(39)>=1-M*b(182);     y(39)-y(9)>=1-M*(1-b(182));
y(15)-y(39)>=1-M*b(183);    y(39)-y(15)>=1-M*(1-b(183));
y(16)-y(39)>=1-M*b(184);    y(39)-y(16)>=1-M*(1-b(184));
y(17)-y(39)>=1-M*b(185);    y(39)-y(17)>=1-M*(1-b(185));
y(18)-y(39)>=1-M*b(186);    y(39)-y(18)>=1-M*(1-b(186));
y(20)-y(39)>=1-M*b(187);    y(39)-y(20)>=1-M*(1-b(187));
y(23)-y(39)>=1-M*b(188);    y(39)-y(23)>=1-M*(1-b(188));
y(24)-y(39)>=1-M*b(189);    y(39)-y(24)>=1-M*(1-b(189));
y(25)-y(39)>=1-M*b(190);    y(39)-y(25)>=1-M*(1-b(190));
y(28)-y(39)>=1-M*b(191);    y(39)-y(28)>=1-M*(1-b(191));
y(29)-y(39)>=1-M*b(192);    y(39)-y(29)>=1-M*(1-b(192));
y(33)-y(39)>=1-M*b(193);    y(39)-y(33)>=1-M*(1-b(193));
y(34)-y(39)>=1-M*b(194);    y(39)-y(34)>=1-M*(1-b(194));
y(36)-y(39)>=1-M*b(195);    y(39)-y(36)>=1-M*(1-b(195));
y(2)-y(41)>=1-M*b(196);     y(41)-y(2)>=1-M*(1-b(196));
y(7)-y(41)>=1-M*b(197);     y(41)-y(7)>=1-M*(1-b(197));
y(9)-y(41)>=1-M*b(198);     y(41)-y(9)>=1-M*(1-b(198));
y(16)-y(41)>=1-M*b(199);    y(41)-y(16)>=1-M*(1-b(199));
y(17)-y(41)>=1-M*b(200);    y(41)-y(17)>=1-M*(1-b(200));
y(18)-y(41)>=1-M*b(201);    y(41)-y(18)>=1-M*(1-b(201));
y(20)-y(41)>=1-M*b(202);    y(41)-y(20)>=1-M*(1-b(202));
y(25)-y(41)>=1-M*b(203);    y(41)-y(25)>=1-M*(1-b(203));
y(28)-y(41)>=1-M*b(204);    y(41)-y(28)>=1-M*(1-b(204));
y(36)-y(41)>=1-M*b(205);    y(41)-y(36)>=1-M*(1-b(205));
y(2)-y(43)>=1-M*b(206);     y(43)-y(2)>=1-M*(1-b(206));
y(4)-y(43)>=1-M*b(207);     y(43)-y(4)>=1-M*(1-b(207));
y(5)-y(43)>=1-M*b(208);     y(43)-y(5)>=1-M*(1-b(208));
y(6)-y(43)>=1-M*b(209);     y(43)-y(6)>=1-M*(1-b(209));
y(7)-y(43)>=1-M*b(210);     y(43)-y(7)>=1-M*(1-b(210));
y(9)-y(43)>=1-M*b(211);     y(43)-y(9)>=1-M*(1-b(211));
y(15)-y(43)>=1-M*b(212);    y(43)-y(15)>=1-M*(1-b(212));
y(16)-y(43)>=1-M*b(213);    y(43)-y(16)>=1-M*(1-b(213));
y(17)-y(43)>=1-M*b(214);    y(43)-y(17)>=1-M*(1-b(214));
y(18)-y(43)>=1-M*b(215);    y(43)-y(18)>=1-M*(1-b(215));
y(19)-y(43)>=1-M*b(216);    y(43)-y(19)>=1-M*(1-b(216));
y(20)-y(43)>=1-M*b(217);    y(43)-y(20)>=1-M*(1-b(217));
y(23)-y(43)>=1-M*b(218);    y(43)-y(23)>=1-M*(1-b(218));
y(24)-y(43)>=1-M*b(219);    y(43)-y(24)>=1-M*(1-b(219));
y(25)-y(43)>=1-M*b(220);    y(43)-y(25)>=1-M*(1-b(220));
y(28)-y(43)>=1-M*b(221);    y(43)-y(28)>=1-M*(1-b(221));
y(29)-y(43)>=1-M*b(222);    y(43)-y(29)>=1-M*(1-b(222));
y(33)-y(43)>=1-M*b(223);    y(43)-y(33)>=1-M*(1-b(223));
y(34)-y(43)>=1-M*b(224);    y(43)-y(34)>=1-M*(1-b(224));
y(36)-y(43)>=1-M*b(225);    y(43)-y(36)>=1-M*(1-b(225));
y(37)-y(43)>=1-M*b(226);    y(43)-y(37)>=1-M*(1-b(226));
```

```
y(40)-y(43)>=1-M*b(227);    y(43)-y(40)>=1-M*(1-b(227));
y(42)-y(43)>=1-M*b(228);    y(43)-y(42)>=1-M*(1-b(228));
y(2)-y(45)>=1-M*b(229);     y(45)-y(2)>=1-M*(1-b(229));
y(4)-y(45)>=1-M*b(230);     y(45)-y(4)>=1-M*(1-b(230));
y(5)-y(45)>=1-M*b(231);     y(45)-y(5)>=1-M*(1-b(231));
y(6)-y(45)>=1-M*b(232);     y(45)-y(6)>=1-M*(1-b(232));
y(7)-y(45)>=1-M*b(233);     y(45)-y(7)>=1-M*(1-b(233));
y(9)-y(45)>=1-M*b(234);     y(45)-y(9)>=1-M*(1-b(234));
y(15)-y(45)>=1-M*b(235);    y(45)-y(15)>=1-M*(1-b(235));
y(16)-y(45)>=1-M*b(236);    y(45)-y(16)>=1-M*(1-b(236));
y(17)-y(45)>=1-M*b(237);    y(45)-y(17)>=1-M*(1-b(237));
y(18)-y(45)>=1-M*b(238);    y(45)-y(18)>=1-M*(1-b(238));
y(19)-y(45)>=1-M*b(239);    y(45)-y(19)>=1-M*(1-b(239));
y(20)-y(45)>=1-M*b(240);    y(45)-y(20)>=1-M*(1-b(240));
y(23)-y(45)>=1-M*b(241);    y(45)-y(23)>=1-M*(1-b(241));
y(24)-y(45)>=1-M*b(242);    y(45)-y(24)>=1-M*(1-b(242));
y(25)-y(45)>=1-M*b(243);    y(45)-y(25)>=1-M*(1-b(243));
y(28)-y(45)>=1-M*b(244);    y(45)-y(28)>=1-M*(1-b(244));
y(29)-y(45)>=1-M*b(245);    y(45)-y(29)>=1-M*(1-b(245));
y(33)-y(45)>=1-M*b(246);    y(45)-y(33)>=1-M*(1-b(246));
y(34)-y(45)>=1-M*b(247);    y(45)-y(34)>=1-M*(1-b(247));
y(36)-y(45)>=1-M*b(248);    y(45)-y(36)>=1-M*(1-b(248));
y(37)-y(45)>=1-M*b(249);    y(45)-y(37)>=1-M*(1-b(249));
y(38)-y(45)>=1-M*b(250);    y(45)-y(38)>=1-M*(1-b(250));
y(40)-y(45)>=1-M*b(251);    y(45)-y(40)>=1-M*(1-b(251));
y(42)-y(45)>=1-M*b(252);    y(45)-y(42)>=1-M*(1-b(252));
y(44)-y(45)>=1-M*b(253);    y(45)-y(44)>=1-M*(1-b(253));

M*c(1)>=y(1)-y(2);          M*c(1)>=y(2)-y(1);
M*c(2)>=y(1)-y(3);          M*c(2)>=y(3)-y(1);
M*c(3)>=y(1)-y(4);          M*c(3)>=y(4)-y(1);
M*c(4)>=y(2)-y(4);          M*c(4)>=y(4)-y(2);
M*c(5)>=y(1)-y(5);          M*c(5)>=y(5)-y(1);
M*c(6)>=y(2)-y(5);          M*c(6)>=y(5)-y(2);
M*c(7)>=y(4)-y(5);          M*c(7)>=y(5)-y(4);
M*c(8)>=y(1)-y(6);          M*c(8)>=y(6)-y(1);
M*c(9)>=y(2)-y(6);          M*c(9)>=y(6)-y(2);
M*c(10)>=y(1)-y(7);         M*c(10)>=y(7)-y(1);
M*c(11)>=y(2)-y(7);         M*c(11)>=y(7)-y(2);
M*c(12)>=y(1)-y(8);         M*c(12)>=y(8)-y(1);
M*c(13)>=y(3)-y(8);         M*c(13)>=y(8)-y(3);
M*c(14)>=y(1)-y(9);         M*c(14)>=y(9)-y(1);
M*c(15)>=y(1)-y(10);        M*c(15)>=y(10)-y(1);
M*c(16)>=y(3)-y(10);        M*c(16)>=y(10)-y(3);
M*c(17)>=y(1)-y(11);        M*c(17)>=y(11)-y(1);
M*c(18)>=y(5)-y(11);        M*c(18)>=y(11)-y(5);
```

```
M*c(19)>=y(1)-y(12);
M*c(20)>=y(4)-y(12);
M*c(21)>=y(5)-y(12);
M*c(22)>=y(1)-y(15);
M*c(23)>=y(2)-y(15);
M*c(24)>=y(4)-y(15);
M*c(25)>=y(6)-y(15);
M*c(26)>=y(7)-y(15);
M*c(27)>=y(9)-y(15);
M*c(28)>=y(11)-y(15);
M*c(29)>=y(12)-y(15);
M*c(30)>=y(1)-y(16);
M*c(31)>=y(1)-y(18);
M*c(32)>=y(2)-y(18);
M*c(33)>=y(9)-y(18);
M*c(34)>=y(16)-y(18);
M*c(35)>=y(17)-y(18);
M*c(36)>=y(1)-y(19);
M*c(37)>=y(2)-y(19);
M*c(38)>=y(3)-y(19);
M*c(39)>=y(4)-y(19);
M*c(40)>=y(5)-y(19);
M*c(41)>=y(6)-y(19);
M*c(42)>=y(7)-y(19);
M*c(43)>=y(8)-y(19);
M*c(44)>=y(9)-y(19);
M*c(45)>=y(10)-y(19);
M*c(46)>=y(11)-y(19);
M*c(47)>=y(12)-y(19);
M*c(48)>=y(15)-y(19);
M*c(49)>=y(16)-y(19);
M*c(50)>=y(17)-y(19);
M*c(51)>=y(18)-y(19);
M*c(52)>=y(1)-y(20);
M*c(53)>=y(9)-y(20);
M*c(54)>=y(16)-y(20);
M*c(55)>=y(17)-y(20);
M*c(56)>=y(1)-y(21);
M*c(57)>=y(3)-y(21);
M*c(58)>=y(8)-y(21);
M*c(59)>=y(10)-y(21);
M*c(60)>=y(11)-y(21);
M*c(61)>=y(12)-y(21);
M*c(62)>=y(1)-y(22);
M*c(63)>=y(3)-y(22);
M*c(64)>=y(11)-y(22);
M*c(19)>=y(12)-y(1);
M*c(20)>=y(12)-y(4);
M*c(21)>=y(12)-y(5);
M*c(22)>=y(15)-y(1);
M*c(23)>=y(15)-y(2);
M*c(24)>=y(15)-y(4);
M*c(25)>=y(15)-y(6);
M*c(26)>=y(15)-y(7);
M*c(27)>=y(15)-y(9);
M*c(28)>=y(15)-y(11);
M*c(29)>=y(15)-y(12);
M*c(30)>=y(16)-y(1);
M*c(31)>=y(18)-y(1);
M*c(32)>=y(18)-y(2);
M*c(33)>=y(18)-y(9);
M*c(34)>=y(18)-y(16);
M*c(35)>=y(18)-y(17);
M*c(36)>=y(19)-y(1);
M*c(37)>=y(19)-y(2);
M*c(38)>=y(19)-y(3);
M*c(39)>=y(19)-y(4);
M*c(40)>=y(19)-y(5);
M*c(41)>=y(19)-y(6);
M*c(42)>=y(19)-y(7);
M*c(43)>=y(19)-y(8);
M*c(44)>=y(19)-y(9);
M*c(45)>=y(19)-y(10);
M*c(46)>=y(19)-y(11);
M*c(47)>=y(19)-y(12);
M*c(48)>=y(19)-y(15);
M*c(49)>=y(19)-y(16);
M*c(50)>=y(19)-y(17);
M*c(51)>=y(19)-y(18);
M*c(52)>=y(20)-y(1);
M*c(53)>=y(20)-y(9);
M*c(54)>=y(20)-y(16);
M*c(55)>=y(20)-y(17);
M*c(56)>=y(21)-y(1);
M*c(57)>=y(21)-y(3);
M*c(58)>=y(21)-y(8);
M*c(59)>=y(21)-y(10);
M*c(60)>=y(21)-y(11);
M*c(61)>=y(21)-y(12);
M*c(62)>=y(22)-y(1);
M*c(63)>=y(22)-y(3);
M*c(64)>=y(22)-y(11);
```

```
M*c(65)>=y(12)-y(22);       M*c(65)>=y(22)-y(12);
M*c(66)>=y(19)-y(22);       M*c(66)>=y(22)-y(19);
M*c(67)>=y(1)-y(23);        M*c(67)>=y(23)-y(1);
M*c(68)>=y(2)-y(23);        M*c(68)>=y(23)-y(2);
M*c(69)>=y(4)-y(23);        M*c(69)>=y(23)-y(4);
M*c(70)>=y(6)-y(23);        M*c(70)>=y(23)-y(6);
M*c(71)>=y(7)-y(23);        M*c(71)>=y(23)-y(7);
M*c(72)>=y(9)-y(23);        M*c(72)>=y(23)-y(9);
M*c(73)>=y(11)-y(23);       M*c(73)>=y(23)-y(11);
M*c(74)>=y(12)-y(23);       M*c(74)>=y(23)-y(12);
M*c(75)>=y(16)-y(23);       M*c(75)>=y(23)-y(16);
M*c(76)>=y(17)-y(23);       M*c(76)>=y(23)-y(17);
M*c(77)>=y(18)-y(23);       M*c(77)>=y(23)-y(18);
M*c(78)>=y(20)-y(23);       M*c(78)>=y(23)-y(20);
M*c(79)>=y(1)-y(24);        M*c(79)>=y(24)-y(1);
M*c(80)>=y(2)-y(24);        M*c(80)>=y(24)-y(2);
M*c(81)>=y(7)-y(24);        M*c(81)>=y(24)-y(7);
M*c(82)>=y(9)-y(24);        M*c(82)>=y(24)-y(9);
M*c(83)>=y(16)-y(24);       M*c(83)>=y(24)-y(16);
M*c(84)>=y(17)-y(24);       M*c(84)>=y(24)-y(17);
M*c(85)>=y(18)-y(24);       M*c(85)>=y(24)-y(18);
M*c(86)>=y(20)-y(24);       M*c(86)>=y(24)-y(20);
M*c(87)>=y(1)-y(25);        M*c(87)>=y(25)-y(1);
M*c(88)>=y(17)-y(25);       M*c(88)>=y(25)-y(17);
M*c(89)>=y(19)-y(26);       M*c(89)>=y(26)-y(19);
M*c(90)>=y(21)-y(26);       M*c(90)>=y(26)-y(21);
M*c(91)>=y(1)-y(27);        M*c(91)>=y(27)-y(1);
M*c(92)>=y(3)-y(27);        M*c(92)>=y(27)-y(3);
M*c(93)>=y(11)-y(27);       M*c(93)>=y(27)-y(11);
M*c(94)>=y(12)-y(27);       M*c(94)>=y(27)-y(12);
M*c(95)>=y(19)-y(27);       M*c(95)>=y(27)-y(19);
M*c(96)>=y(22)-y(27);       M*c(96)>=y(27)-y(22);
M*c(97)>=y(1)-y(28);        M*c(97)>=y(28)-y(1);
M*c(98)>=y(2)-y(28);        M*c(98)>=y(28)-y(2);
M*c(99)>=y(7)-y(28);        M*c(99)>=y(28)-y(7);
M*c(100)>=y(9)-y(28);       M*c(100)>=y(28)-y(9);
M*c(101)>=y(16)-y(28);      M*c(101)>=y(28)-y(16);
M*c(102)>=y(17)-y(28);      M*c(102)>=y(28)-y(17);
M*c(103)>=y(18)-y(28);      M*c(103)>=y(28)-y(18);
M*c(104)>=y(20)-y(28);      M*c(104)>=y(28)-y(20);
M*c(105)>=y(25)-y(28);      M*c(105)>=y(28)-y(25);
M*c(106)>=y(1)-y(29);       M*c(106)>=y(29)-y(1);
M*c(107)>=y(2)-y(29);       M*c(107)>=y(29)-y(2);
M*c(108)>=y(7)-y(29);       M*c(108)>=y(29)-y(7);
M*c(109)>=y(9)-y(29);       M*c(109)>=y(29)-y(9);
M*c(110)>=y(16)-y(29);      M*c(110)>=y(29)-y(16);
```

```
M*c(111)>=y(17)-y(29);          M*c(111)>=y(29)-y(17);
M*c(112)>=y(18)-y(29);          M*c(112)>=y(29)-y(18);
M*c(113)>=y(20)-y(29);          M*c(113)>=y(29)-y(20);
M*c(114)>=y(25)-y(29);          M*c(114)>=y(29)-y(25);
M*c(115)>=y(28)-y(29);          M*c(115)>=y(29)-y(28);
M*c(116)>=y(8)-y(30);           M*c(116)>=y(30)-y(8);
M*c(117)>=y(10)-y(30);          M*c(117)>=y(30)-y(10);
M*c(118)>=y(19)-y(30);          M*c(118)>=y(30)-y(19);
M*c(119)>=y(21)-y(30);          M*c(119)>=y(30)-y(21);
M*c(120)>=y(22)-y(30);          M*c(120)>=y(30)-y(22);
M*c(121)>=y(27)-y(30);          M*c(121)>=y(30)-y(27);
M*c(122)>=y(1)-y(31);           M*c(122)>=y(31)-y(1);
M*c(123)>=y(3)-y(31);           M*c(123)>=y(31)-y(3);
M*c(124)>=y(8)-y(31);           M*c(124)>=y(31)-y(8);
M*c(125)>=y(10)-y(31);          M*c(125)>=y(31)-y(10);
M*c(126)>=y(11)-y(31);          M*c(126)>=y(31)-y(11);
M*c(127)>=y(12)-y(31);          M*c(127)>=y(31)-y(12);
M*c(128)>=y(22)-y(31);          M*c(128)>=y(31)-y(22);
M*c(129)>=y(26)-y(31);          M*c(129)>=y(31)-y(26);
M*c(130)>=y(27)-y(31);          M*c(130)>=y(31)-y(27);
M*c(131)>=y(30)-y(31);          M*c(131)>=y(31)-y(30);
M*c(132)>=y(26)-y(32);          M*c(132)>=y(32)-y(26);
M*c(133)>=y(30)-y(32);          M*c(133)>=y(32)-y(30);
M*c(134)>=y(1)-y(33);           M*c(134)>=y(33)-y(1);
M*c(135)>=y(2)-y(33);           M*c(135)>=y(33)-y(2);
M*c(136)>=y(4)-y(33);           M*c(136)>=y(33)-y(4);
M*c(137)>=y(6)-y(33);           M*c(137)>=y(33)-y(6);
M*c(138)>=y(7)-y(33);           M*c(138)>=y(33)-y(7);
M*c(139)>=y(9)-y(33);           M*c(139)>=y(33)-y(9);
M*c(140)>=y(11)-y(33);          M*c(140)>=y(33)-y(11);
M*c(141)>=y(12)-y(33);          M*c(141)>=y(33)-y(12);
M*c(142)>=y(16)-y(33);          M*c(142)>=y(33)-y(16);
M*c(143)>=y(17)-y(33);          M*c(143)>=y(33)-y(17);
M*c(144)>=y(18)-y(33);          M*c(144)>=y(33)-y(18);
M*c(145)>=y(20)-y(33);          M*c(145)>=y(33)-y(20);
M*c(146)>=y(24)-y(33);          M*c(146)>=y(33)-y(24);
M*c(147)>=y(25)-y(33);          M*c(147)>=y(33)-y(25);
M*c(148)>=y(28)-y(33);          M*c(148)>=y(33)-y(28);
M*c(149)>=y(29)-y(33);          M*c(149)>=y(33)-y(29);
M*c(150)>=y(1)-y(34);           M*c(150)>=y(34)-y(1);
M*c(151)>=y(2)-y(34);           M*c(151)>=y(34)-y(2);
M*c(152)>=y(4)-y(34);           M*c(152)>=y(34)-y(4);
M*c(153)>=y(6)-y(34);           M*c(153)>=y(34)-y(6);
M*c(154)>=y(7)-y(34);           M*c(154)>=y(34)-y(7);
M*c(155)>=y(9)-y(34);           M*c(155)>=y(34)-y(9);
M*c(156)>=y(11)-y(34);          M*c(156)>=y(34)-y(11);
```

```
M*c(157)>=y(12)-y(34);
M*c(158)>=y(16)-y(34);
M*c(159)>=y(17)-y(34);
M*c(160)>=y(18)-y(34);
M*c(161)>=y(20)-y(34);
M*c(162)>=y(24)-y(34);
M*c(163)>=y(25)-y(34);
M*c(164)>=y(28)-y(34);
M*c(165)>=y(29)-y(34);
M*c(166)>=y(1)-y(35);
M*c(167)>=y(4)-y(35);
M*c(168)>=y(5)-y(35);
M*c(169)>=y(6)-y(35);
M*c(170)>=y(15)-y(35);
M*c(171)>=y(19)-y(35);
M*c(172)>=y(23)-y(35);
M*c(173)>=y(24)-y(35);
M*c(174)>=y(29)-y(35);
M*c(175)>=y(33)-y(35);
M*c(176)>=y(34)-y(35);
M*c(177)>=y(1)-y(36);
M*c(178)>=y(2)-y(36);
M*c(179)>=y(7)-y(36);
M*c(180)>=y(9)-y(36);
M*c(181)>=y(16)-y(36);
M*c(182)>=y(17)-y(36);
M*c(183)>=y(18)-y(36);
M*c(184)>=y(20)-y(36);
M*c(185)>=y(25)-y(36);
M*c(186)>=y(1)-y(37);
M*c(187)>=y(2)-y(37);
M*c(188)>=y(3)-y(37);
M*c(189)>=y(4)-y(37);
M*c(190)>=y(5)-y(37);
M*c(191)>=y(6)-y(37);
M*c(192)>=y(7)-y(37);
M*c(193)>=y(9)-y(37);
M*c(194)>=y(11)-y(37);
M*c(195)>=y(12)-y(37);
M*c(196)>=y(15)-y(37);
M*c(197)>=y(16)-y(37);
M*c(198)>=y(17)-y(37);
M*c(199)>=y(18)-y(37);
M*c(200)>=y(20)-y(37);
M*c(201)>=y(22)-y(37);
M*c(202)>=y(23)-y(37);
M*c(157)>=y(34)-y(12);
M*c(158)>=y(34)-y(16);
M*c(159)>=y(34)-y(17);
M*c(160)>=y(34)-y(18);
M*c(161)>=y(34)-y(20);
M*c(162)>=y(34)-y(24);
M*c(163)>=y(34)-y(25);
M*c(164)>=y(34)-y(28);
M*c(165)>=y(34)-y(29);
M*c(166)>=y(35)-y(1);
M*c(167)>=y(35)-y(4);
M*c(168)>=y(35)-y(5);
M*c(169)>=y(35)-y(6);
M*c(170)>=y(35)-y(15);
M*c(171)>=y(35)-y(19);
M*c(172)>=y(35)-y(23);
M*c(173)>=y(35)-y(24);
M*c(174)>=y(35)-y(29);
M*c(175)>=y(35)-y(33);
M*c(176)>=y(35)-y(34);
M*c(177)>=y(36)-y(1);
M*c(178)>=y(36)-y(2);
M*c(179)>=y(36)-y(7);
M*c(180)>=y(36)-y(9);
M*c(181)>=y(36)-y(16);
M*c(182)>=y(36)-y(17);
M*c(183)>=y(36)-y(18);
M*c(184)>=y(36)-y(20);
M*c(185)>=y(36)-y(25);
M*c(186)>=y(37)-y(1);
M*c(187)>=y(37)-y(2);
M*c(188)>=y(37)-y(3);
M*c(189)>=y(37)-y(4);
M*c(190)>=y(37)-y(5);
M*c(191)>=y(37)-y(6);
M*c(192)>=y(37)-y(7);
M*c(193)>=y(37)-y(9);
M*c(194)>=y(37)-y(11);
M*c(195)>=y(37)-y(12);
M*c(196)>=y(37)-y(15);
M*c(197)>=y(37)-y(16);
M*c(198)>=y(37)-y(17);
M*c(199)>=y(37)-y(18);
M*c(200)>=y(37)-y(20);
M*c(201)>=y(37)-y(22);
M*c(202)>=y(37)-y(23);
```

```
M*c(203)>=y(24)-y(37);
M*c(204)>=y(25)-y(37);
M*c(205)>=y(28)-y(37);
M*c(206)>=y(29)-y(37);
M*c(207)>=y(30)-y(37);
M*c(208)>=y(33)-y(37);
M*c(209)>=y(34)-y(37);
M*c(210)>=y(35)-y(37);
M*c(211)>=y(36)-y(37);
M*c(212)>=y(1)-y(38);
M*c(213)>=y(2)-y(38);
M*c(214)>=y(3)-y(38);
M*c(215)>=y(4)-y(38);
M*c(216)>=y(5)-y(38);
M*c(217)>=y(6)-y(38);
M*c(218)>=y(7)-y(38);
M*c(219)>=y(8)-y(38);
M*c(220)>=y(9)-y(38);
M*c(221)>=y(10)-y(38);
M*c(222)>=y(11)-y(38);
M*c(223)>=y(12)-y(38);
M*c(224)>=y(15)-y(38);
M*c(225)>=y(16)-y(38);
M*c(226)>=y(17)-y(38);
M*c(227)>=y(18)-y(38);
M*c(228)>=y(19)-y(38);
M*c(229)>=y(20)-y(38);
M*c(230)>=y(21)-y(38);
M*c(231)>=y(22)-y(38);
M*c(232)>=y(23)-y(38);
M*c(233)>=y(24)-y(38);
M*c(234)>=y(25)-y(38);
M*c(235)>=y(26)-y(38);
M*c(236)>=y(27)-y(38);
M*c(237)>=y(28)-y(38);
M*c(238)>=y(29)-y(38);
M*c(239)>=y(30)-y(38);
M*c(240)>=y(31)-y(38);
M*c(241)>=y(32)-y(38);
M*c(242)>=y(33)-y(38);
M*c(243)>=y(34)-y(38);
M*c(244)>=y(35)-y(38);
M*c(245)>=y(36)-y(38);
M*c(246)>=y(37)-y(38);
M*c(247)>=y(1)-y(39);
M*c(248)>=y(3)-y(39);
```

```
M*c(203)>=y(37)-y(24);
M*c(204)>=y(37)-y(25);
M*c(205)>=y(37)-y(28);
M*c(206)>=y(37)-y(29);
M*c(207)>=y(37)-y(30);
M*c(208)>=y(37)-y(33);
M*c(209)>=y(37)-y(34);
M*c(210)>=y(37)-y(35);
M*c(211)>=y(37)-y(36);
M*c(212)>=y(38)-y(1);
M*c(213)>=y(38)-y(2);
M*c(214)>=y(38)-y(3);
M*c(215)>=y(38)-y(4);
M*c(216)>=y(38)-y(5);
M*c(217)>=y(38)-y(6);
M*c(218)>=y(38)-y(7);
M*c(219)>=y(38)-y(8);
M*c(220)>=y(38)-y(9);
M*c(221)>=y(38)-y(10);
M*c(222)>=y(38)-y(11);
M*c(223)>=y(38)-y(12);
M*c(224)>=y(38)-y(15);
M*c(225)>=y(38)-y(16);
M*c(226)>=y(38)-y(17);
M*c(227)>=y(38)-y(18);
M*c(228)>=y(38)-y(19);
M*c(229)>=y(38)-y(20);
M*c(230)>=y(38)-y(21);
M*c(231)>=y(38)-y(22);
M*c(232)>=y(38)-y(23);
M*c(233)>=y(38)-y(24);
M*c(234)>=y(38)-y(25);
M*c(235)>=y(38)-y(26);
M*c(236)>=y(38)-y(27);
M*c(237)>=y(38)-y(28);
M*c(238)>=y(38)-y(29);
M*c(239)>=y(38)-y(30);
M*c(240)>=y(38)-y(31);
M*c(241)>=y(38)-y(32);
M*c(242)>=y(38)-y(33);
M*c(243)>=y(38)-y(34);
M*c(244)>=y(38)-y(35);
M*c(245)>=y(38)-y(36);
M*c(246)>=y(38)-y(37);
M*c(247)>=y(39)-y(1);
M*c(248)>=y(39)-y(3);
```

```
M*c(249)>=y(11)-y(39);
M*c(250)>=y(12)-y(39);
M*c(251)>=y(19)-y(39);
M*c(252)>=y(35)-y(39);
M*c(253)>=y(37)-y(39);
M*c(254)>=y(38)-y(39);
M*c(255)>=y(1)-y(40);
M*c(256)>=y(2)-y(40);
M*c(257)>=y(3)-y(40);
M*c(258)>=y(4)-y(40);
M*c(259)>=y(5)-y(40);
M*c(260)>=y(6)-y(40);
M*c(261)>=y(7)-y(40);
M*c(262)>=y(9)-y(40);
M*c(263)>=y(11)-y(40);
M*c(264)>=y(12)-y(40);
M*c(265)>=y(15)-y(40);
M*c(266)>=y(16)-y(40);
M*c(267)>=y(17)-y(40);
M*c(268)>=y(18)-y(40);
M*c(269)>=y(20)-y(40);
M*c(270)>=y(23)-y(40);
M*c(271)>=y(24)-y(40);
M*c(272)>=y(25)-y(40);
M*c(273)>=y(28)-y(40);
M*c(274)>=y(29)-y(40);
M*c(275)>=y(30)-y(40);
M*c(276)>=y(33)-y(40);
M*c(277)>=y(34)-y(40);
M*c(278)>=y(35)-y(40);
M*c(279)>=y(36)-y(40);
M*c(280)>=y(39)-y(40);
M*c(281)>=y(1)-y(41);
M*c(282)>=y(4)-y(41);
M*c(283)>=y(5)-y(41);
M*c(284)>=y(6)-y(41);
M*c(285)>=y(15)-y(41);
M*c(286)>=y(19)-y(41);
M*c(287)>=y(23)-y(41);
M*c(288)>=y(24)-y(41);
M*c(289)>=y(29)-y(41);
M*c(290)>=y(33)-y(41);
M*c(291)>=y(34)-y(41);
M*c(292)>=y(37)-y(41);
M*c(293)>=y(38)-y(41);
M*c(294)>=y(40)-y(41);
M*c(249)>=y(39)-y(11);
M*c(250)>=y(39)-y(12);
M*c(251)>=y(39)-y(19);
M*c(252)>=y(39)-y(35);
M*c(253)>=y(39)-y(37);
M*c(254)>=y(39)-y(38);
M*c(255)>=y(40)-y(1);
M*c(256)>=y(40)-y(2);
M*c(257)>=y(40)-y(3);
M*c(258)>=y(40)-y(4);
M*c(259)>=y(40)-y(5);
M*c(260)>=y(40)-y(6);
M*c(261)>=y(40)-y(7);
M*c(262)>=y(40)-y(9);
M*c(263)>=y(40)-y(11);
M*c(264)>=y(40)-y(12);
M*c(265)>=y(40)-y(15);
M*c(266)>=y(40)-y(16);
M*c(267)>=y(40)-y(17);
M*c(268)>=y(40)-y(18);
M*c(269)>=y(40)-y(20);
M*c(270)>=y(40)-y(23);
M*c(271)>=y(40)-y(24);
M*c(272)>=y(40)-y(25);
M*c(273)>=y(40)-y(28);
M*c(274)>=y(40)-y(29);
M*c(275)>=y(40)-y(30);
M*c(276)>=y(40)-y(33);
M*c(277)>=y(40)-y(34);
M*c(278)>=y(40)-y(35);
M*c(279)>=y(40)-y(36);
M*c(280)>=y(40)-y(39);
M*c(281)>=y(41)-y(1);
M*c(282)>=y(41)-y(4);
M*c(283)>=y(41)-y(5);
M*c(284)>=y(41)-y(6);
M*c(285)>=y(41)-y(15);
M*c(286)>=y(41)-y(19);
M*c(287)>=y(41)-y(23);
M*c(288)>=y(41)-y(24);
M*c(289)>=y(41)-y(29);
M*c(290)>=y(41)-y(33);
M*c(291)>=y(41)-y(34);
M*c(292)>=y(41)-y(37);
M*c(293)>=y(41)-y(38);
M*c(294)>=y(41)-y(40);
```

```
M*c(295)>=y(1)-y(42);
M*c(296)>=y(2)-y(42);
M*c(297)>=y(3)-y(42);
M*c(298)>=y(4)-y(42);
M*c(299)>=y(5)-y(42);
M*c(300)>=y(6)-y(42);
M*c(301)>=y(7)-y(42);
M*c(302)>=y(9)-y(42);
M*c(303)>=y(11)-y(42);
M*c(304)>=y(12)-y(42);
M*c(305)>=y(15)-y(42);
M*c(306)>=y(16)-y(42);
M*c(307)>=y(17)-y(42);
M*c(308)>=y(18)-y(42);
M*c(309)>=y(20)-y(42);
M*c(310)>=y(22)-y(42);
M*c(311)>=y(23)-y(42);
M*c(312)>=y(24)-y(42);
M*c(313)>=y(25)-y(42);
M*c(314)>=y(28)-y(42);
M*c(315)>=y(29)-y(42);
M*c(316)>=y(30)-y(42);
M*c(317)>=y(33)-y(42);
M*c(318)>=y(34)-y(42);
M*c(319)>=y(35)-y(42);
M*c(320)>=y(36)-y(42);
M*c(321)>=y(39)-y(42);
M*c(322)>=y(40)-y(42);
M*c(323)>=y(41)-y(42);
M*c(324)>=y(1)-y(43);
M*c(325)>=y(3)-y(43);
M*c(326)>=y(8)-y(43);
M*c(327)>=y(10)-y(43);
M*c(328)>=y(11)-y(43);
M*c(329)>=y(12)-y(43);
M*c(330)>=y(22)-y(43);
M*c(331)>=y(26)-y(43);
M*c(332)>=y(27)-y(43);
M*c(333)>=y(30)-y(43);
M*c(334)>=y(35)-y(43);
M*c(335)>=y(38)-y(43);
M*c(336)>=y(39)-y(43);
M*c(337)>=y(41)-y(43);
M*c(338)>=y(1)-y(44);
M*c(339)>=y(2)-y(44);
M*c(340)>=y(4)-y(44);
M*c(295)>=y(42)-y(1);
M*c(296)>=y(42)-y(2);
M*c(297)>=y(42)-y(3);
M*c(298)>=y(42)-y(4);
M*c(299)>=y(42)-y(5);
M*c(300)>=y(42)-y(6);
M*c(301)>=y(42)-y(7);
M*c(302)>=y(42)-y(9);
M*c(303)>=y(42)-y(11);
M*c(304)>=y(42)-y(12);
M*c(305)>=y(42)-y(15);
M*c(306)>=y(42)-y(16);
M*c(307)>=y(42)-y(17);
M*c(308)>=y(42)-y(18);
M*c(309)>=y(42)-y(20);
M*c(310)>=y(42)-y(22);
M*c(311)>=y(42)-y(23);
M*c(312)>=y(42)-y(24);
M*c(313)>=y(42)-y(25);
M*c(314)>=y(42)-y(28);
M*c(315)>=y(42)-y(29);
M*c(316)>=y(42)-y(30);
M*c(317)>=y(42)-y(33);
M*c(318)>=y(42)-y(34);
M*c(319)>=y(42)-y(35);
M*c(320)>=y(42)-y(36);
M*c(321)>=y(42)-y(39);
M*c(322)>=y(42)-y(40);
M*c(323)>=y(42)-y(41);
M*c(324)>=y(43)-y(1);
M*c(325)>=y(43)-y(3);
M*c(326)>=y(43)-y(8);
M*c(327)>=y(43)-y(10);
M*c(328)>=y(43)-y(11);
M*c(329)>=y(43)-y(12);
M*c(330)>=y(43)-y(22);
M*c(331)>=y(43)-y(26);
M*c(332)>=y(43)-y(27);
M*c(333)>=y(43)-y(30);
M*c(334)>=y(43)-y(35);
M*c(335)>=y(43)-y(38);
M*c(336)>=y(43)-y(39);
M*c(337)>=y(43)-y(41);
M*c(338)>=y(44)-y(1);
M*c(339)>=y(44)-y(2);
M*c(340)>=y(44)-y(4);
```

```
M*c(341)>=y(6)-y(44);
M*c(342)>=y(7)-y(44);
M*c(343)>=y(9)-y(44);
M*c(344)>=y(11)-y(44);
M*c(345)>=y(12)-y(44);
M*c(346)>=y(15)-y(44);
M*c(347)>=y(16)-y(44);
M*c(348)>=y(17)-y(44);
M*c(349)>=y(18)-y(44);
M*c(350)>=y(20)-y(44);
M*c(351)>=y(23)-y(44);
M*c(352)>=y(24)-y(44);
M*c(353)>=y(25)-y(44);
M*c(354)>=y(28)-y(44);
M*c(355)>=y(29)-y(44);
M*c(356)>=y(33)-y(44);
M*c(357)>=y(34)-y(44);
M*c(358)>=y(35)-y(44);
M*c(359)>=y(36)-y(44);
M*c(360)>=y(41)-y(44);
M*c(361)>=y(1)-y(45);
M*c(362)>=y(3)-y(45);
M*c(363)>=y(8)-y(45);
M*c(364)>=y(10)-y(45);
M*c(365)>=y(11)-y(45);
M*c(366)>=y(12)-y(45);
M*c(367)>=y(21)-y(45);
M*c(368)>=y(22)-y(45);
M*c(369)>=y(26)-y(45);
M*c(370)>=y(27)-y(45);
M*c(371)>=y(30)-y(45);
M*c(372)>=y(31)-y(45);
M*c(373)>=y(32)-y(45);
M*c(374)>=y(35)-y(45);
M*c(375)>=y(39)-y(45);
M*c(376)>=y(41)-y(45);
M*c(377)>=y(43)-y(45);
M*c(378)>=y(1)-y(46);
M*c(379)>=y(2)-y(46);
M*c(380)>=y(4)-y(46);
M*c(381)>=y(6)-y(46);
M*c(382)>=y(7)-y(46);
M*c(383)>=y(9)-y(46);
M*c(384)>=y(11)-y(46);
M*c(385)>=y(12)-y(46);
M*c(386)>=y(16)-y(46);
M*c(341)>=y(44)-y(6);
M*c(342)>=y(44)-y(7);
M*c(343)>=y(44)-y(9);
M*c(344)>=y(44)-y(11);
M*c(345)>=y(44)-y(12);
M*c(346)>=y(44)-y(15);
M*c(347)>=y(44)-y(16);
M*c(348)>=y(44)-y(17);
M*c(349)>=y(44)-y(18);
M*c(350)>=y(44)-y(20);
M*c(351)>=y(44)-y(23);
M*c(352)>=y(44)-y(24);
M*c(353)>=y(44)-y(25);
M*c(354)>=y(44)-y(28);
M*c(355)>=y(44)-y(29);
M*c(356)>=y(44)-y(33);
M*c(357)>=y(44)-y(34);
M*c(358)>=y(44)-y(35);
M*c(359)>=y(44)-y(36);
M*c(360)>=y(44)-y(41);
M*c(361)>=y(45)-y(1);
M*c(362)>=y(45)-y(3);
M*c(363)>=y(45)-y(8);
M*c(364)>=y(45)-y(10);
M*c(365)>=y(45)-y(11);
M*c(366)>=y(45)-y(12);
M*c(367)>=y(45)-y(21);
M*c(368)>=y(45)-y(22);
M*c(369)>=y(45)-y(26);
M*c(370)>=y(45)-y(27);
M*c(371)>=y(45)-y(30);
M*c(372)>=y(45)-y(31);
M*c(373)>=y(45)-y(32);
M*c(374)>=y(45)-y(35);
M*c(375)>=y(45)-y(39);
M*c(376)>=y(45)-y(41);
M*c(377)>=y(45)-y(43);
M*c(378)>=y(46)-y(1);
M*c(379)>=y(46)-y(2);
M*c(380)>=y(46)-y(4);
M*c(381)>=y(46)-y(6);
M*c(382)>=y(46)-y(7);
M*c(383)>=y(46)-y(9);
M*c(384)>=y(46)-y(11);
M*c(385)>=y(46)-y(12);
M*c(386)>=y(46)-y(16);
```

```
        M*c(387)>=y(17)-y(46);          M*c(387)>=y(46)-y(17);
        M*c(388)>=y(18)-y(46);          M*c(388)>=y(46)-y(18);
        M*c(389)>=y(20)-y(46);          M*c(389)>=y(46)-y(20);
        M*c(390)>=y(23)-y(46);          M*c(390)>=y(46)-y(23);
        M*c(391)>=y(24)-y(46);          M*c(391)>=y(46)-y(24);
        M*c(392)>=y(25)-y(46);          M*c(392)>=y(46)-y(25);
        M*c(393)>=y(28)-y(46);          M*c(393)>=y(46)-y(28);
        M*c(394)>=y(29)-y(46);          M*c(394)>=y(46)-y(29);
        M*c(395)>=y(33)-y(46);          M*c(395)>=y(46)-y(33);
        M*c(396)>=y(34)-y(46);          M*c(396)>=y(46)-y(34);
        M*c(397)>=y(35)-y(46);          M*c(397)>=y(46)-y(35);
        M*c(398)>=y(36)-y(46);          M*c(398)>=y(46)-y(36);
        M*c(399)>=y(41)-y(46);          M*c(399)>=y(46)-y(41);

data:
        M=10000;
enddata
end
```

Lingo Model for H6500 Dataset

```
model:
sets:
        obj/1..31/:y;
        flag1/1..196/:a;
        flag2/1..141/:b;
        flag3/1..128/:c;
endsets
        min=@sum(flag1:a)+@sum(flag2:b)+@sum(flag3:c);

        @for(obj(p):
                @bnd(1,y(p),5);
                @gin(y(p));
        );
        @for(flag1(q1):
                @bin(a(q1));
        );
        @for(flag2(q2):
                @bin(b(q2));
        );
        @for(flag3(q3):
                @bin(c(q3));
        );

        y(3)-y(1)>=1-M*a(1);            y(1)-y(3)>=1-M*(1-a(1));
        y(3)-y(2)>=1-M*a(2);            y(2)-y(3)>=1-M*(1-a(2));
```

```
y(5)-y(2)>=1-M*a(3);        y(2)-y(5)>=1-M*(1-a(3));
y(5)-y(4)>=1-M*a(4);        y(4)-y(5)>=1-M*(1-a(4));
y(6)-y(4)>=1-M*a(5);        y(4)-y(6)>=1-M*(1-a(5));
y(7)-y(4)>=1-M*a(6);        y(4)-y(7)>=1-M*(1-a(6));
y(7)-y(6)>=1-M*a(7);        y(6)-y(7)>=1-M*(1-a(7));
y(9)-y(1)>=1-M*a(8);        y(1)-y(9)>=1-M*(1-a(8));
y(9)-y(2)>=1-M*a(9);        y(2)-y(9)>=1-M*(1-a(9));
y(9)-y(4)>=1-M*a(10);       y(4)-y(9)>=1-M*(1-a(10));
y(9)-y(5)>=1-M*a(11);       y(5)-y(9)>=1-M*(1-a(11));
y(9)-y(6)>=1-M*a(12);       y(6)-y(9)>=1-M*(1-a(12));
y(9)-y(7)>=1-M*a(13);       y(7)-y(9)>=1-M*(1-a(13));
y(9)-y(8)>=1-M*a(14);       y(8)-y(9)>=1-M*(1-a(14));
y(10)-y(1)>=1-M*a(15);      y(1)-y(10)>=1-M*(1-a(15));
y(10)-y(2)>=1-M*a(16);      y(2)-y(10)>=1-M*(1-a(16));
y(10)-y(4)>=1-M*a(17);      y(4)-y(10)>=1-M*(1-a(17));
y(10)-y(5)>=1-M*a(18);      y(5)-y(10)>=1-M*(1-a(18));
y(10)-y(6)>=1-M*a(19);      y(6)-y(10)>=1-M*(1-a(19));
y(10)-y(7)>=1-M*a(20);      y(7)-y(10)>=1-M*(1-a(20));
y(10)-y(8)>=1-M*a(21);      y(8)-y(10)>=1-M*(1-a(21));
y(10)-y(9)>=1-M*a(22);      y(9)-y(10)>=1-M*(1-a(22));
y(11)-y(4)>=1-M*a(23);      y(4)-y(11)>=1-M*(1-a(23));
y(11)-y(8)>=1-M*a(24);      y(8)-y(11)>=1-M*(1-a(24));
y(12)-y(4)>=1-M*a(25);      y(4)-y(12)>=1-M*(1-a(25));
y(12)-y(6)>=1-M*a(26);      y(6)-y(12)>=1-M*(1-a(26));
y(12)-y(8)>=1-M*a(27);      y(8)-y(12)>=1-M*(1-a(27));
y(12)-y(11)>=1-M*a(28);     y(11)-y(12)>=1-M*(1-a(28));
y(14)-y(2)>=1-M*a(29);      y(2)-y(14)>=1-M*(1-a(29));
y(14)-y(4)>=1-M*a(30);      y(4)-y(14)>=1-M*(1-a(30));
y(14)-y(5)>=1-M*a(31);      y(5)-y(14)>=1-M*(1-a(31));
y(14)-y(6)>=1-M*a(32);      y(6)-y(14)>=1-M*(1-a(32));
y(14)-y(7)>=1-M*a(33);      y(7)-y(14)>=1-M*(1-a(33));
y(14)-y(8)>=1-M*a(34);      y(8)-y(14)>=1-M*(1-a(34));
y(14)-y(11)>=1-M*a(35);     y(11)-y(14)>=1-M*(1-a(35));
y(14)-y(12)>=1-M*a(36);     y(12)-y(14)>=1-M*(1-a(36));
y(14)-y(13)>=1-M*a(37);     y(13)-y(14)>=1-M*(1-a(37));
y(15)-y(2)>=1-M*a(38);      y(2)-y(15)>=1-M*(1-a(38));
y(15)-y(4)>=1-M*a(39);      y(4)-y(15)>=1-M*(1-a(39));
y(15)-y(5)>=1-M*a(40);      y(5)-y(15)>=1-M*(1-a(40));
y(15)-y(6)>=1-M*a(41);      y(6)-y(15)>=1-M*(1-a(41));
y(15)-y(7)>=1-M*a(42);      y(7)-y(15)>=1-M*(1-a(42));
y(15)-y(8)>=1-M*a(43);      y(8)-y(15)>=1-M*(1-a(43));
y(15)-y(11)>=1-M*a(44);     y(11)-y(15)>=1-M*(1-a(44));
y(15)-y(12)>=1-M*a(45);     y(12)-y(15)>=1-M*(1-a(45));
y(15)-y(13)>=1-M*a(46);     y(13)-y(15)>=1-M*(1-a(46));
y(15)-y(14)>=1-M*a(47);     y(14)-y(15)>=1-M*(1-a(47));
y(16)-y(2)>=1-M*a(48);      y(2)-y(16)>=1-M*(1-a(48));
```

```
y(16)-y(4)>=1-M*a(49);
y(16)-y(6)>=1-M*a(50);
y(16)-y(8)>=1-M*a(51);
y(16)-y(11)>=1-M*a(52);
y(16)-y(12)>=1-M*a(53);
y(16)-y(13)>=1-M*a(54);
y(18)-y(8)>=1-M*a(55);
y(18)-y(13)>=1-M*a(56);
y(18)-y(17)>=1-M*a(57);
y(19)-y(2)>=1-M*a(58);
y(19)-y(4)>=1-M*a(59);
y(19)-y(6)>=1-M*a(60);
y(19)-y(7)>=1-M*a(61);
y(19)-y(8)>=1-M*a(62);
y(19)-y(11)>=1-M*a(63);
y(19)-y(12)>=1-M*a(64);
y(19)-y(13)>=1-M*a(65);
y(19)-y(17)>=1-M*a(66);
y(19)-y(18)>=1-M*a(67);
y(20)-y(2)>=1-M*a(68);
y(20)-y(4)>=1-M*a(69);
y(20)-y(5)>=1-M*a(70);
y(20)-y(6)>=1-M*a(71);
y(20)-y(7)>=1-M*a(72);
y(20)-y(8)>=1-M*a(73);
y(20)-y(11)>=1-M*a(74);
y(20)-y(12)>=1-M*a(75);
y(20)-y(13)>=1-M*a(76);
y(20)-y(14)>=1-M*a(77);
y(20)-y(17)>=1-M*a(78);
y(20)-y(18)>=1-M*a(79);
y(20)-y(19)>=1-M*a(80);
y(21)-y(2)>=1-M*a(81);
y(21)-y(4)>=1-M*a(82);
y(21)-y(6)>=1-M*a(83);
y(21)-y(7)>=1-M*a(84);
y(21)-y(8)>=1-M*a(85);
y(21)-y(11)>=1-M*a(86);
y(21)-y(12)>=1-M*a(87);
y(21)-y(13)>=1-M*a(88);
y(21)-y(17)>=1-M*a(89);
y(21)-y(18)>=1-M*a(90);
y(22)-y(2)>=1-M*a(91);
y(22)-y(4)>=1-M*a(92);
y(22)-y(6)>=1-M*a(93);
y(22)-y(8)>=1-M*a(94);
y(4)-y(16)>=1-M*(1-a(49));
y(6)-y(16)>=1-M*(1-a(50));
y(8)-y(16)>=1-M*(1-a(51));
y(11)-y(16)>=1-M*(1-a(52));
y(12)-y(16)>=1-M*(1-a(53));
y(13)-y(16)>=1-M*(1-a(54));
y(8)-y(18)>=1-M*(1-a(55));
y(13)-y(18)>=1-M*(1-a(56));
y(17)-y(18)>=1-M*(1-a(57));
y(2)-y(19)>=1-M*(1-a(58));
y(4)-y(19)>=1-M*(1-a(59));
y(6)-y(19)>=1-M*(1-a(60));
y(7)-y(19)>=1-M*(1-a(61));
y(8)-y(19)>=1-M*(1-a(62));
y(11)-y(19)>=1-M*(1-a(63));
y(12)-y(19)>=1-M*(1-a(64));
y(13)-y(19)>=1-M*(1-a(65));
y(17)-y(19)>=1-M*(1-a(66));
y(18)-y(19)>=1-M*(1-a(67));
y(2)-y(20)>=1-M*(1-a(68));
y(4)-y(20)>=1-M*(1-a(69));
y(5)-y(20)>=1-M*(1-a(70));
y(6)-y(20)>=1-M*(1-a(71));
y(7)-y(20)>=1-M*(1-a(72));
y(8)-y(20)>=1-M*(1-a(73));
y(11)-y(20)>=1-M*(1-a(74));
y(12)-y(20)>=1-M*(1-a(75));
y(13)-y(20)>=1-M*(1-a(76));
y(14)-y(20)>=1-M*(1-a(77));
y(17)-y(20)>=1-M*(1-a(78));
y(18)-y(20)>=1-M*(1-a(79));
y(19)-y(20)>=1-M*(1-a(80));
y(2)-y(21)>=1-M*(1-a(81));
y(4)-y(21)>=1-M*(1-a(82));
y(6)-y(21)>=1-M*(1-a(83));
y(7)-y(21)>=1-M*(1-a(84));
y(8)-y(21)>=1-M*(1-a(85));
y(11)-y(21)>=1-M*(1-a(86));
y(12)-y(21)>=1-M*(1-a(87));
y(13)-y(21)>=1-M*(1-a(88));
y(17)-y(21)>=1-M*(1-a(89));
y(18)-y(21)>=1-M*(1-a(90));
y(2)-y(22)>=1-M*(1-a(91));
y(4)-y(22)>=1-M*(1-a(92));
y(6)-y(22)>=1-M*(1-a(93));
y(8)-y(22)>=1-M*(1-a(94));
```

```
y(22)-y(11)>=1-M*a(95);    y(11)-y(22)>=1-M*(1-a(95));
y(22)-y(12)>=1-M*a(96);    y(12)-y(22)>=1-M*(1-a(96));
y(22)-y(13)>=1-M*a(97);    y(13)-y(22)>=1-M*(1-a(97));
y(22)-y(17)>=1-M*a(98);    y(17)-y(22)>=1-M*(1-a(98));
y(22)-y(18)>=1-M*a(99);    y(18)-y(22)>=1-M*(1-a(99));
y(24)-y(8)>=1-M*a(100);    y(8)-y(24)>=1-M*(1-a(100));
y(24)-y(13)>=1-M*a(101);    y(13)-y(24)>=1-M*(1-a(101));
y(24)-y(17)>=1-M*a(102);    y(17)-y(24)>=1-M*(1-a(102));
y(24)-y(18)>=1-M*a(103);    y(18)-y(24)>=1-M*(1-a(103));
y(24)-y(23)>=1-M*a(104);    y(23)-y(24)>=1-M*(1-a(104));
y(25)-y(17)>=1-M*a(105);    y(17)-y(25)>=1-M*(1-a(105));
y(25)-y(23)>=1-M*a(106);    y(23)-y(25)>=1-M*(1-a(106));
y(26)-y(8)>=1-M*a(107);    y(8)-y(26)>=1-M*(1-a(107));
y(26)-y(13)>=1-M*a(108);    y(13)-y(26)>=1-M*(1-a(108));
y(26)-y(17)>=1-M*a(109);    y(17)-y(26)>=1-M*(1-a(109));
y(26)-y(18)>=1-M*a(110);    y(18)-y(26)>=1-M*(1-a(110));
y(26)-y(23)>=1-M*a(111);    y(23)-y(26)>=1-M*(1-a(111));
y(26)-y(25)>=1-M*a(112);    y(25)-y(26)>=1-M*(1-a(112));
y(27)-y(8)>=1-M*a(113);    y(8)-y(27)>=1-M*(1-a(113));
y(27)-y(13)>=1-M*a(114);    y(13)-y(27)>=1-M*(1-a(114));
y(27)-y(17)>=1-M*a(115);    y(17)-y(27)>=1-M*(1-a(115));
y(27)-y(18)>=1-M*a(116);    y(18)-y(27)>=1-M*(1-a(116));
y(27)-y(23)>=1-M*a(117);    y(23)-y(27)>=1-M*(1-a(117));
y(27)-y(25)>=1-M*a(118);    y(25)-y(27)>=1-M*(1-a(118));
y(27)-y(26)>=1-M*a(119);    y(26)-y(27)>=1-M*(1-a(119));
y(28)-y(1)>=1-M*a(120);    y(1)-y(28)>=1-M*(1-a(120));
y(28)-y(2)>=1-M*a(121);    y(2)-y(28)>=1-M*(1-a(121));
y(28)-y(4)>=1-M*a(122);    y(4)-y(28)>=1-M*(1-a(122));
y(28)-y(5)>=1-M*a(123);    y(5)-y(28)>=1-M*(1-a(123));
y(28)-y(6)>=1-M*a(124);    y(6)-y(28)>=1-M*(1-a(124));
y(28)-y(7)>=1-M*a(125);    y(7)-y(28)>=1-M*(1-a(125));
y(28)-y(8)>=1-M*a(126);    y(8)-y(28)>=1-M*(1-a(126));
y(28)-y(11)>=1-M*a(127);    y(11)-y(28)>=1-M*(1-a(127));
y(28)-y(12)>=1-M*a(128);    y(12)-y(28)>=1-M*(1-a(128));
y(28)-y(13)>=1-M*a(129);    y(13)-y(28)>=1-M*(1-a(129));
y(28)-y(14)>=1-M*a(130);    y(14)-y(28)>=1-M*(1-a(130));
y(28)-y(15)>=1-M*a(131);    y(15)-y(28)>=1-M*(1-a(131));
y(28)-y(16)>=1-M*a(132);    y(16)-y(28)>=1-M*(1-a(132));
y(28)-y(17)>=1-M*a(133);    y(17)-y(28)>=1-M*(1-a(133));
y(28)-y(18)>=1-M*a(134);    y(18)-y(28)>=1-M*(1-a(134));
y(28)-y(19)>=1-M*a(135);    y(19)-y(28)>=1-M*(1-a(135));
y(28)-y(20)>=1-M*a(136);    y(20)-y(28)>=1-M*(1-a(136));
y(28)-y(21)>=1-M*a(137);    y(21)-y(28)>=1-M*(1-a(137));
y(28)-y(22)>=1-M*a(138);    y(22)-y(28)>=1-M*(1-a(138));
y(28)-y(23)>=1-M*a(139);    y(23)-y(28)>=1-M*(1-a(139));
y(28)-y(24)>=1-M*a(140);    y(24)-y(28)>=1-M*(1-a(140));
```

```
y(28)-y(25)>=1-M*a(141);     y(25)-y(28)>=1-M*(1-a(141));
y(28)-y(26)>=1-M*a(142);     y(26)-y(28)>=1-M*(1-a(142));
y(28)-y(27)>=1-M*a(143);     y(27)-y(28)>=1-M*(1-a(143));
y(29)-y(2)>=1-M*a(144);      y(2)-y(29)>=1-M*(1-a(144));
y(29)-y(4)>=1-M*a(145);      y(4)-y(29)>=1-M*(1-a(145));
y(29)-y(6)>=1-M*a(146);      y(6)-y(29)>=1-M*(1-a(146));
y(29)-y(7)>=1-M*a(147);      y(7)-y(29)>=1-M*(1-a(147));
y(29)-y(8)>=1-M*a(148);      y(8)-y(29)>=1-M*(1-a(148));
y(29)-y(11)>=1-M*a(149);     y(11)-y(29)>=1-M*(1-a(149));
y(29)-y(12)>=1-M*a(150);     y(12)-y(29)>=1-M*(1-a(150));
y(29)-y(13)>=1-M*a(151);     y(13)-y(29)>=1-M*(1-a(151));
y(29)-y(17)>=1-M*a(152);     y(17)-y(29)>=1-M*(1-a(152));
y(29)-y(18)>=1-M*a(153);     y(18)-y(29)>=1-M*(1-a(153));
y(29)-y(19)>=1-M*a(154);     y(19)-y(29)>=1-M*(1-a(154));
y(29)-y(21)>=1-M*a(155);     y(21)-y(29)>=1-M*(1-a(155));
y(29)-y(23)>=1-M*a(156);     y(23)-y(29)>=1-M*(1-a(156));
y(29)-y(24)>=1-M*a(157);     y(24)-y(29)>=1-M*(1-a(157));
y(29)-y(25)>=1-M*a(158);     y(25)-y(29)>=1-M*(1-a(158));
y(29)-y(26)>=1-M*a(159);     y(26)-y(29)>=1-M*(1-a(159));
y(29)-y(27)>=1-M*a(160);     y(27)-y(29)>=1-M*(1-a(160));
y(30)-y(4)>=1-M*a(161);      y(4)-y(30)>=1-M*(1-a(161));
y(30)-y(6)>=1-M*a(162);      y(6)-y(30)>=1-M*(1-a(162));
y(30)-y(8)>=1-M*a(163);      y(8)-y(30)>=1-M*(1-a(163));
y(30)-y(11)>=1-M*a(164);     y(11)-y(30)>=1-M*(1-a(164));
y(30)-y(12)>=1-M*a(165);     y(12)-y(30)>=1-M*(1-a(165));
y(30)-y(13)>=1-M*a(166);     y(13)-y(30)>=1-M*(1-a(166));
y(30)-y(17)>=1-M*a(167);     y(17)-y(30)>=1-M*(1-a(167));
y(30)-y(18)>=1-M*a(168);     y(18)-y(30)>=1-M*(1-a(168));
y(30)-y(23)>=1-M*a(169);     y(23)-y(30)>=1-M*(1-a(169));
y(30)-y(24)>=1-M*a(170);     y(24)-y(30)>=1-M*(1-a(170));
y(30)-y(25)>=1-M*a(171);     y(25)-y(30)>=1-M*(1-a(171));
y(30)-y(26)>=1-M*a(172);     y(26)-y(30)>=1-M*(1-a(172));
y(30)-y(27)>=1-M*a(173);     y(27)-y(30)>=1-M*(1-a(173));
y(31)-y(2)>=1-M*a(174);      y(2)-y(31)>=1-M*(1-a(174));
y(31)-y(4)>=1-M*a(175);      y(4)-y(31)>=1-M*(1-a(175));
y(31)-y(5)>=1-M*a(176);      y(5)-y(31)>=1-M*(1-a(176));
y(31)-y(6)>=1-M*a(177);      y(6)-y(31)>=1-M*(1-a(177));
y(31)-y(7)>=1-M*a(178);      y(7)-y(31)>=1-M*(1-a(178));
y(31)-y(8)>=1-M*a(179);      y(8)-y(31)>=1-M*(1-a(179));
y(31)-y(11)>=1-M*a(180);     y(11)-y(31)>=1-M*(1-a(180));
y(31)-y(12)>=1-M*a(181);     y(12)-y(31)>=1-M*(1-a(181));
y(31)-y(13)>=1-M*a(182);     y(13)-y(31)>=1-M*(1-a(182));
y(31)-y(14)>=1-M*a(183);     y(14)-y(31)>=1-M*(1-a(183));
y(31)-y(17)>=1-M*a(184);     y(17)-y(31)>=1-M*(1-a(184));
y(31)-y(18)>=1-M*a(185);     y(18)-y(31)>=1-M*(1-a(185));
y(31)-y(19)>=1-M*a(186);     y(19)-y(31)>=1-M*(1-a(186));
```

```
y(31)-y(20)>=1-M*a(187);     y(20)-y(31)>=1-M*(1-a(187));
y(31)-y(21)>=1-M*a(188);     y(21)-y(31)>=1-M*(1-a(188));
y(31)-y(22)>=1-M*a(189);     y(22)-y(31)>=1-M*(1-a(189));
y(31)-y(23)>=1-M*a(190);     y(23)-y(31)>=1-M*(1-a(190));
y(31)-y(24)>=1-M*a(191);     y(24)-y(31)>=1-M*(1-a(191));
y(31)-y(25)>=1-M*a(192);     y(25)-y(31)>=1-M*(1-a(192));
y(31)-y(26)>=1-M*a(193);     y(26)-y(31)>=1-M*(1-a(193));
y(31)-y(27)>=1-M*a(194);     y(27)-y(31)>=1-M*(1-a(194));
y(31)-y(29)>=1-M*a(195);     y(29)-y(31)>=1-M*(1-a(195));
y(31)-y(30)>=1-M*a(196);     y(30)-y(31)>=1-M*(1-a(196));

y(1)-y(2)>=1-M*b(1);         y(2)-y(1)>=1-M*(1-b(1));
y(1)-y(4)>=1-M*b(2);         y(4)-y(1)>=1-M*(1-b(2));
y(3)-y(4)>=1-M*b(3);         y(4)-y(3)>=1-M*(1-b(3));
y(1)-y(6)>=1-M*b(4);         y(6)-y(1)>=1-M*(1-b(4));
y(3)-y(6)>=1-M*b(5);         y(6)-y(3)>=1-M*(1-b(5));
y(5)-y(6)>=1-M*b(6);         y(6)-y(5)>=1-M*(1-b(6));
y(1)-y(8)>=1-M*b(7);         y(8)-y(1)>=1-M*(1-b(7));
y(3)-y(8)>=1-M*b(8);         y(8)-y(3)>=1-M*(1-b(8));
y(5)-y(8)>=1-M*b(9);         y(8)-y(5)>=1-M*(1-b(9));
y(7)-y(8)>=1-M*b(10);        y(8)-y(7)>=1-M*(1-b(10));
y(1)-y(11)>=1-M*b(11);       y(11)-y(1)>=1-M*(1-b(11));
y(3)-y(11)>=1-M*b(12);       y(11)-y(3)>=1-M*(1-b(12));
y(5)-y(11)>=1-M*b(13);       y(11)-y(5)>=1-M*(1-b(13));
y(7)-y(11)>=1-M*b(14);       y(11)-y(7)>=1-M*(1-b(14));
y(9)-y(11)>=1-M*b(15);       y(11)-y(9)>=1-M*(1-b(15));
y(10)-y(11)>=1-M*b(16);      y(11)-y(10)>=1-M*(1-b(16));
y(1)-y(12)>=1-M*b(17);       y(12)-y(1)>=1-M*(1-b(17));
y(3)-y(12)>=1-M*b(18);       y(12)-y(3)>=1-M*(1-b(18));
y(5)-y(12)>=1-M*b(19);       y(12)-y(5)>=1-M*(1-b(19));
y(7)-y(12)>=1-M*b(20);       y(12)-y(7)>=1-M*(1-b(20));
y(9)-y(12)>=1-M*b(21);       y(12)-y(9)>=1-M*(1-b(21));
y(10)-y(12)>=1-M*b(22);      y(12)-y(10)>=1-M*(1-b(22));
y(1)-y(13)>=1-M*b(23);       y(13)-y(1)>=1-M*(1-b(23));
y(3)-y(13)>=1-M*b(24);       y(13)-y(3)>=1-M*(1-b(24));
y(5)-y(13)>=1-M*b(25);       y(13)-y(5)>=1-M*(1-b(25));
y(7)-y(13)>=1-M*b(26);       y(13)-y(7)>=1-M*(1-b(26));
y(9)-y(13)>=1-M*b(27);       y(13)-y(9)>=1-M*(1-b(27));
y(10)-y(13)>=1-M*b(28);      y(13)-y(10)>=1-M*(1-b(28));
y(1)-y(16)>=1-M*b(29);       y(16)-y(1)>=1-M*(1-b(29));
y(3)-y(16)>=1-M*b(30);       y(16)-y(3)>=1-M*(1-b(30));
y(9)-y(16)>=1-M*b(31);       y(16)-y(9)>=1-M*(1-b(31));
y(10)-y(16)>=1-M*b(32);      y(16)-y(10)>=1-M*(1-b(32));
y(15)-y(16)>=1-M*b(33);      y(16)-y(15)>=1-M*(1-b(33));
y(1)-y(17)>=1-M*b(34);       y(17)-y(1)>=1-M*(1-b(34));
y(2)-y(17)>=1-M*b(35);       y(17)-y(2)>=1-M*(1-b(35));
```

```
y(3)-y(17)>=1-M*b(36);
y(4)-y(17)>=1-M*b(37);
y(5)-y(17)>=1-M*b(38);
y(6)-y(17)>=1-M*b(39);
y(7)-y(17)>=1-M*b(40);
y(8)-y(17)>=1-M*b(41);
y(9)-y(17)>=1-M*b(42);
y(10)-y(17)>=1-M*b(43);
y(11)-y(17)>=1-M*b(44);
y(12)-y(17)>=1-M*b(45);
y(13)-y(17)>=1-M*b(46);
y(14)-y(17)>=1-M*b(47);
y(15)-y(17)>=1-M*b(48);
y(16)-y(17)>=1-M*b(49);
y(1)-y(18)>=1-M*b(50);
y(3)-y(18)>=1-M*b(51);
y(5)-y(18)>=1-M*b(52);
y(7)-y(18)>=1-M*b(53);
y(9)-y(18)>=1-M*b(54);
y(10)-y(18)>=1-M*b(55);
y(14)-y(18)>=1-M*b(56);
y(15)-y(18)>=1-M*b(57);
y(1)-y(22)>=1-M*b(58);
y(3)-y(22)>=1-M*b(59);
y(9)-y(22)>=1-M*b(60);
y(10)-y(22)>=1-M*b(61);
y(14)-y(22)>=1-M*b(62);
y(15)-y(22)>=1-M*b(63);
y(20)-y(22)>=1-M*b(64);
y(1)-y(23)>=1-M*b(65);
y(2)-y(23)>=1-M*b(66);
y(3)-y(23)>=1-M*b(67);
y(4)-y(23)>=1-M*b(68);
y(5)-y(23)>=1-M*b(69);
y(6)-y(23)>=1-M*b(70);
y(7)-y(23)>=1-M*b(71);
y(8)-y(23)>=1-M*b(72);
y(9)-y(23)>=1-M*b(73);
y(10)-y(23)>=1-M*b(74);
y(11)-y(23)>=1-M*b(75);
y(12)-y(23)>=1-M*b(76);
y(13)-y(23)>=1-M*b(77);
y(14)-y(23)>=1-M*b(78);
y(15)-y(23)>=1-M*b(79);
y(16)-y(23)>=1-M*b(80);
y(17)-y(23)>=1-M*b(81);
y(17)-y(3)>=1-M*(1-b(36));
y(17)-y(4)>=1-M*(1-b(37));
y(17)-y(5)>=1-M*(1-b(38));
y(17)-y(6)>=1-M*(1-b(39));
y(17)-y(7)>=1-M*(1-b(40));
y(17)-y(8)>=1-M*(1-b(41));
y(17)-y(9)>=1-M*(1-b(42));
y(17)-y(10)>=1-M*(1-b(43));
y(17)-y(11)>=1-M*(1-b(44));
y(17)-y(12)>=1-M*(1-b(45));
y(17)-y(13)>=1-M*(1-b(46));
y(17)-y(14)>=1-M*(1-b(47));
y(17)-y(15)>=1-M*(1-b(48));
y(17)-y(16)>=1-M*(1-b(49));
y(18)-y(1)>=1-M*(1-b(50));
y(18)-y(3)>=1-M*(1-b(51));
y(18)-y(5)>=1-M*(1-b(52));
y(18)-y(7)>=1-M*(1-b(53));
y(18)-y(9)>=1-M*(1-b(54));
y(18)-y(10)>=1-M*(1-b(55));
y(18)-y(14)>=1-M*(1-b(56));
y(18)-y(15)>=1-M*(1-b(57));
y(22)-y(1)>=1-M*(1-b(58));
y(22)-y(3)>=1-M*(1-b(59));
y(22)-y(9)>=1-M*(1-b(60));
y(22)-y(10)>=1-M*(1-b(61));
y(22)-y(14)>=1-M*(1-b(62));
y(22)-y(15)>=1-M*(1-b(63));
y(22)-y(20)>=1-M*(1-b(64));
y(23)-y(1)>=1-M*(1-b(65));
y(23)-y(2)>=1-M*(1-b(66));
y(23)-y(3)>=1-M*(1-b(67));
y(23)-y(4)>=1-M*(1-b(68));
y(23)-y(5)>=1-M*(1-b(69));
y(23)-y(6)>=1-M*(1-b(70));
y(23)-y(7)>=1-M*(1-b(71));
y(23)-y(8)>=1-M*(1-b(72));
y(23)-y(9)>=1-M*(1-b(73));
y(23)-y(10)>=1-M*(1-b(74));
y(23)-y(11)>=1-M*(1-b(75));
y(23)-y(12)>=1-M*(1-b(76));
y(23)-y(13)>=1-M*(1-b(77));
y(23)-y(14)>=1-M*(1-b(78));
y(23)-y(15)>=1-M*(1-b(79));
y(23)-y(16)>=1-M*(1-b(80));
y(23)-y(17)>=1-M*(1-b(81));
```

```
y(18)-y(23)>=1-M*b(82);        y(23)-y(18)>=1-M*(1-b(82));
y(19)-y(23)>=1-M*b(83);        y(23)-y(19)>=1-M*(1-b(83));
y(20)-y(23)>=1-M*b(84);        y(23)-y(20)>=1-M*(1-b(84));
y(21)-y(23)>=1-M*b(85);        y(23)-y(21)>=1-M*(1-b(85));
y(22)-y(23)>=1-M*b(86);        y(23)-y(22)>=1-M*(1-b(86));
y(1)-y(24)>=1-M*b(87);         y(24)-y(1)>=1-M*(1-b(87));
y(3)-y(24)>=1-M*b(88);         y(24)-y(3)>=1-M*(1-b(88));
y(5)-y(24)>=1-M*b(89);         y(24)-y(5)>=1-M*(1-b(89));
y(7)-y(24)>=1-M*b(90);         y(24)-y(7)>=1-M*(1-b(90));
y(9)-y(24)>=1-M*b(91);         y(24)-y(9)>=1-M*(1-b(91));
y(10)-y(24)>=1-M*b(92);        y(24)-y(10)>=1-M*(1-b(92));
y(14)-y(24)>=1-M*b(93);        y(24)-y(14)>=1-M*(1-b(93));
y(15)-y(24)>=1-M*b(94);        y(24)-y(15)>=1-M*(1-b(94));
y(19)-y(24)>=1-M*b(95);        y(24)-y(19)>=1-M*(1-b(95));
y(20)-y(24)>=1-M*b(96);        y(24)-y(20)>=1-M*(1-b(96));
y(21)-y(24)>=1-M*b(97);        y(24)-y(21)>=1-M*(1-b(97));
y(1)-y(25)>=1-M*b(98);         y(25)-y(1)>=1-M*(1-b(98));
y(2)-y(25)>=1-M*b(99);         y(25)-y(2)>=1-M*(1-b(99));
y(3)-y(25)>=1-M*b(100);        y(25)-y(3)>=1-M*(1-b(100));
y(4)-y(25)>=1-M*b(101);        y(25)-y(4)>=1-M*(1-b(101));
y(5)-y(25)>=1-M*b(102);        y(25)-y(5)>=1-M*(1-b(102));
y(6)-y(25)>=1-M*b(103);        y(25)-y(6)>=1-M*(1-b(103));
y(7)-y(25)>=1-M*b(104);        y(25)-y(7)>=1-M*(1-b(104));
y(8)-y(25)>=1-M*b(105);        y(25)-y(8)>=1-M*(1-b(105));
y(9)-y(25)>=1-M*b(106);        y(25)-y(9)>=1-M*(1-b(106));
y(10)-y(25)>=1-M*b(107);       y(25)-y(10)>=1-M*(1-b(107));
y(11)-y(25)>=1-M*b(108);       y(25)-y(11)>=1-M*(1-b(108));
y(12)-y(25)>=1-M*b(109);       y(25)-y(12)>=1-M*(1-b(109));
y(13)-y(25)>=1-M*b(110);       y(25)-y(13)>=1-M*(1-b(110));
y(14)-y(25)>=1-M*b(111);       y(25)-y(14)>=1-M*(1-b(111));
y(15)-y(25)>=1-M*b(112);       y(25)-y(15)>=1-M*(1-b(112));
y(16)-y(25)>=1-M*b(113);       y(25)-y(16)>=1-M*(1-b(113));
y(18)-y(25)>=1-M*b(114);       y(25)-y(18)>=1-M*(1-b(114));
y(19)-y(25)>=1-M*b(115);       y(25)-y(19)>=1-M*(1-b(115));
y(20)-y(25)>=1-M*b(116);       y(25)-y(20)>=1-M*(1-b(116));
y(21)-y(25)>=1-M*b(117);       y(25)-y(21)>=1-M*(1-b(117));
y(22)-y(25)>=1-M*b(118);       y(25)-y(22)>=1-M*(1-b(118));
y(24)-y(25)>=1-M*b(119);       y(25)-y(24)>=1-M*(1-b(119));
y(1)-y(26)>=1-M*b(120);        y(26)-y(1)>=1-M*(1-b(120));
y(3)-y(26)>=1-M*b(121);        y(26)-y(3)>=1-M*(1-b(121));
y(5)-y(26)>=1-M*b(122);        y(26)-y(5)>=1-M*(1-b(122));
y(7)-y(26)>=1-M*b(123);        y(26)-y(7)>=1-M*(1-b(123));
y(9)-y(26)>=1-M*b(124);        y(26)-y(9)>=1-M*(1-b(124));
y(10)-y(26)>=1-M*b(125);       y(26)-y(10)>=1-M*(1-b(125));
y(14)-y(26)>=1-M*b(126);       y(26)-y(14)>=1-M*(1-b(126));
y(15)-y(26)>=1-M*b(127);       y(26)-y(15)>=1-M*(1-b(127));
```

```
y(19)-y(26)>=1-M*b(128);        y(26)-y(19)>=1-M*(1-b(128));
y(20)-y(26)>=1-M*b(129);        y(26)-y(20)>=1-M*(1-b(129));
y(21)-y(26)>=1-M*b(130);        y(26)-y(21)>=1-M*(1-b(130));
y(1)-y(27)>=1-M*b(131);         y(27)-y(1)>=1-M*(1-b(131));
y(3)-y(27)>=1-M*b(132);         y(27)-y(3)>=1-M*(1-b(132));
y(5)-y(27)>=1-M*b(133);         y(27)-y(5)>=1-M*(1-b(133));
y(7)-y(27)>=1-M*b(134);         y(27)-y(7)>=1-M*(1-b(134));
y(9)-y(27)>=1-M*b(135);         y(27)-y(9)>=1-M*(1-b(135));
y(10)-y(27)>=1-M*b(136);        y(27)-y(10)>=1-M*(1-b(136));
y(14)-y(27)>=1-M*b(137);        y(27)-y(14)>=1-M*(1-b(137));
y(15)-y(27)>=1-M*b(138);        y(27)-y(15)>=1-M*(1-b(138));
y(19)-y(27)>=1-M*b(139);        y(27)-y(19)>=1-M*(1-b(139));
y(20)-y(27)>=1-M*b(140);        y(27)-y(20)>=1-M*(1-b(140));
y(21)-y(27)>=1-M*b(141);        y(27)-y(21)>=1-M*(1-b(141));

M*c(1)>=y(2)-y(4);              M*c(1)>=y(4)-y(2);
M*c(2)>=y(1)-y(5);              M*c(2)>=y(5)-y(1);
M*c(3)>=y(3)-y(5);              M*c(3)>=y(5)-y(3);
M*c(4)>=y(2)-y(6);              M*c(4)>=y(6)-y(2);
M*c(5)>=y(1)-y(7);              M*c(5)>=y(7)-y(1);
M*c(6)>=y(2)-y(7);              M*c(6)>=y(7)-y(2);
M*c(7)>=y(3)-y(7);              M*c(7)>=y(7)-y(3);
M*c(8)>=y(5)-y(7);              M*c(8)>=y(7)-y(5);
M*c(9)>=y(2)-y(8);              M*c(9)>=y(8)-y(2);
M*c(10)>=y(4)-y(8);             M*c(10)>=y(8)-y(4);
M*c(11)>=y(6)-y(8);             M*c(11)>=y(8)-y(6);
M*c(12)>=y(3)-y(9);             M*c(12)>=y(9)-y(3);
M*c(13)>=y(3)-y(10);            M*c(13)>=y(10)-y(3);
M*c(14)>=y(2)-y(11);            M*c(14)>=y(11)-y(2);
M*c(15)>=y(6)-y(11);            M*c(15)>=y(11)-y(6);
M*c(16)>=y(2)-y(12);            M*c(16)>=y(12)-y(2);
M*c(17)>=y(2)-y(13);            M*c(17)>=y(13)-y(2);
M*c(18)>=y(4)-y(13);            M*c(18)>=y(13)-y(4);
M*c(19)>=y(6)-y(13);            M*c(19)>=y(13)-y(6);
M*c(20)>=y(8)-y(13);            M*c(20)>=y(13)-y(8);
M*c(21)>=y(11)-y(13);           M*c(21)>=y(13)-y(11);
M*c(22)>=y(12)-y(13);           M*c(22)>=y(13)-y(12);
M*c(23)>=y(1)-y(14);            M*c(23)>=y(14)-y(1);
M*c(24)>=y(3)-y(14);            M*c(24)>=y(14)-y(3);
M*c(25)>=y(9)-y(14);            M*c(25)>=y(14)-y(9);
M*c(26)>=y(10)-y(14);           M*c(26)>=y(14)-y(10);
M*c(27)>=y(1)-y(15);            M*c(27)>=y(15)-y(1);
M*c(28)>=y(3)-y(15);            M*c(28)>=y(15)-y(3);
M*c(29)>=y(9)-y(15);            M*c(29)>=y(15)-y(9);
M*c(30)>=y(10)-y(15);           M*c(30)>=y(15)-y(10);
M*c(31)>=y(5)-y(16);            M*c(31)>=y(16)-y(5);
```

```
M*c(32)>=y(7)-y(16);
M*c(33)>=y(14)-y(16);
M*c(34)>=y(2)-y(18);
M*c(35)>=y(4)-y(18);
M*c(36)>=y(6)-y(18);
M*c(37)>=y(11)-y(18);
M*c(38)>=y(12)-y(18);
M*c(39)>=y(16)-y(18);
M*c(40)>=y(1)-y(19);
M*c(41)>=y(3)-y(19);
M*c(42)>=y(5)-y(19);
M*c(43)>=y(9)-y(19);
M*c(44)>=y(10)-y(19);
M*c(45)>=y(14)-y(19);
M*c(46)>=y(15)-y(19);
M*c(47)>=y(16)-y(19);
M*c(48)>=y(1)-y(20);
M*c(49)>=y(3)-y(20);
M*c(50)>=y(9)-y(20);
M*c(51)>=y(10)-y(20);
M*c(52)>=y(15)-y(20);
M*c(53)>=y(16)-y(20);
M*c(54)>=y(1)-y(21);
M*c(55)>=y(3)-y(21);
M*c(56)>=y(5)-y(21);
M*c(57)>=y(9)-y(21);
M*c(58)>=y(10)-y(21);
M*c(59)>=y(14)-y(21);
M*c(60)>=y(15)-y(21);
M*c(61)>=y(16)-y(21);
M*c(62)>=y(19)-y(21);
M*c(63)>=y(20)-y(21);
M*c(64)>=y(5)-y(22);
M*c(65)>=y(7)-y(22);
M*c(66)>=y(16)-y(22);
M*c(67)>=y(19)-y(22);
M*c(68)>=y(21)-y(22);
M*c(69)>=y(2)-y(24);
M*c(70)>=y(4)-y(24);
M*c(71)>=y(6)-y(24);
M*c(72)>=y(11)-y(24);
M*c(73)>=y(12)-y(24);
M*c(74)>=y(16)-y(24);
M*c(75)>=y(22)-y(24);
M*c(76)>=y(2)-y(26);
M*c(77)>=y(4)-y(26);
M*c(32)>=y(16)-y(7);
M*c(33)>=y(16)-y(14);
M*c(34)>=y(18)-y(2);
M*c(35)>=y(18)-y(4);
M*c(36)>=y(18)-y(6);
M*c(37)>=y(18)-y(11);
M*c(38)>=y(18)-y(12);
M*c(39)>=y(18)-y(16);
M*c(40)>=y(19)-y(1);
M*c(41)>=y(19)-y(3);
M*c(42)>=y(19)-y(5);
M*c(43)>=y(19)-y(9);
M*c(44)>=y(19)-y(10);
M*c(45)>=y(19)-y(14);
M*c(46)>=y(19)-y(15);
M*c(47)>=y(19)-y(16);
M*c(48)>=y(20)-y(1);
M*c(49)>=y(20)-y(3);
M*c(50)>=y(20)-y(9);
M*c(51)>=y(20)-y(10);
M*c(52)>=y(20)-y(15);
M*c(53)>=y(20)-y(16);
M*c(54)>=y(21)-y(1);
M*c(55)>=y(21)-y(3);
M*c(56)>=y(21)-y(5);
M*c(57)>=y(21)-y(9);
M*c(58)>=y(21)-y(10);
M*c(59)>=y(21)-y(14);
M*c(60)>=y(21)-y(15);
M*c(61)>=y(21)-y(16);
M*c(62)>=y(21)-y(19);
M*c(63)>=y(21)-y(20);
M*c(64)>=y(22)-y(5);
M*c(65)>=y(22)-y(7);
M*c(66)>=y(22)-y(16);
M*c(67)>=y(22)-y(19);
M*c(68)>=y(22)-y(21);
M*c(69)>=y(24)-y(2);
M*c(70)>=y(24)-y(4);
M*c(71)>=y(24)-y(6);
M*c(72)>=y(24)-y(11);
M*c(73)>=y(24)-y(12);
M*c(74)>=y(24)-y(16);
M*c(75)>=y(24)-y(22);
M*c(76)>=y(26)-y(2);
M*c(77)>=y(26)-y(4);
```

```
M*c(78)>=y(6)-y(26);
M*c(79)>=y(11)-y(26);
M*c(80)>=y(12)-y(26);
M*c(81)>=y(16)-y(26);
M*c(82)>=y(22)-y(26);
M*c(83)>=y(24)-y(26);
M*c(84)>=y(2)-y(27);
M*c(85)>=y(4)-y(27);
M*c(86)>=y(6)-y(27);
M*c(87)>=y(11)-y(27);
M*c(88)>=y(12)-y(27);
M*c(89)>=y(16)-y(27);
M*c(90)>=y(22)-y(27);
M*c(91)>=y(24)-y(27);
M*c(92)>=y(3)-y(28);
M*c(93)>=y(9)-y(28);
M*c(94)>=y(10)-y(28);
M*c(95)>=y(1)-y(29);
M*c(96)>=y(3)-y(29);
M*c(97)>=y(5)-y(29);
M*c(98)>=y(9)-y(29);
M*c(99)>=y(10)-y(29);
M*c(100)>=y(14)-y(29);
M*c(101)>=y(15)-y(29);
M*c(102)>=y(16)-y(29);
M*c(103)>=y(20)-y(29);
M*c(104)>=y(22)-y(29);
M*c(105)>=y(28)-y(29);
M*c(106)>=y(1)-y(30);
M*c(107)>=y(2)-y(30);
M*c(108)>=y(3)-y(30);
M*c(109)>=y(5)-y(30);
M*c(110)>=y(7)-y(30);
M*c(111)>=y(9)-y(30);
M*c(112)>=y(10)-y(30);
M*c(113)>=y(14)-y(30);
M*c(114)>=y(15)-y(30);
M*c(115)>=y(16)-y(30);
M*c(116)>=y(19)-y(30);
M*c(117)>=y(20)-y(30);
M*c(118)>=y(21)-y(30);
M*c(119)>=y(22)-y(30);
M*c(120)>=y(28)-y(30);
M*c(121)>=y(29)-y(30);
M*c(122)>=y(1)-y(31);
M*c(123)>=y(3)-y(31);
M*c(78)>=y(26)-y(6);
M*c(79)>=y(26)-y(11);
M*c(80)>=y(26)-y(12);
M*c(81)>=y(26)-y(16);
M*c(82)>=y(26)-y(22);
M*c(83)>=y(26)-y(24);
M*c(84)>=y(27)-y(2);
M*c(85)>=y(27)-y(4);
M*c(86)>=y(27)-y(6);
M*c(87)>=y(27)-y(11);
M*c(88)>=y(27)-y(12);
M*c(89)>=y(27)-y(16);
M*c(90)>=y(27)-y(22);
M*c(91)>=y(27)-y(24);
M*c(92)>=y(28)-y(3);
M*c(93)>=y(28)-y(9);
M*c(94)>=y(28)-y(10);
M*c(95)>=y(29)-y(1);
M*c(96)>=y(29)-y(3);
M*c(97)>=y(29)-y(5);
M*c(98)>=y(29)-y(9);
M*c(99)>=y(29)-y(10);
M*c(100)>=y(29)-y(14);
M*c(101)>=y(29)-y(15);
M*c(102)>=y(29)-y(16);
M*c(103)>=y(29)-y(20);
M*c(104)>=y(29)-y(22);
M*c(105)>=y(29)-y(28);
M*c(106)>=y(30)-y(1);
M*c(107)>=y(30)-y(2);
M*c(108)>=y(30)-y(3);
M*c(109)>=y(30)-y(5);
M*c(110)>=y(30)-y(7);
M*c(111)>=y(30)-y(9);
M*c(112)>=y(30)-y(10);
M*c(113)>=y(30)-y(14);
M*c(114)>=y(30)-y(15);
M*c(115)>=y(30)-y(16);
M*c(116)>=y(30)-y(19);
M*c(117)>=y(30)-y(20);
M*c(118)>=y(30)-y(21);
M*c(119)>=y(30)-y(22);
M*c(120)>=y(30)-y(28);
M*c(121)>=y(30)-y(29);
M*c(122)>=y(31)-y(1);
M*c(123)>=y(31)-y(3);
```

```
    M*c(124)>=y(9)-y(31);       M*c(124)>=y(31)-y(9);
    M*c(125)>=y(10)-y(31);      M*c(125)>=y(31)-y(10);
    M*c(126)>=y(15)-y(31);      M*c(126)>=y(31)-y(15);
    M*c(127)>=y(16)-y(31);      M*c(127)>=y(31)-y(16);
    M*c(128)>=y(28)-y(31);      M*c(128)>=y(31)-y(28);

data:
    M=10000;
enddata
end
```

REFERENCES

[ACMS94] Robert L. Armacost, Paul J. Componation, Michael A. Mullens, and William W. Swart. An AHP framework for prioritizing customer requirements in QFD: an industrialized housing application. *IIE Transactions*, 26(4):72--79, 1994.

[AGI07] Nikolay Archak, Anindya Ghose, and Panagiotis G. Ipeirotis. Show me the money!: Deriving the pricing power of product features by mining consumer reviews. In *KDD'07*, pages 56--65. ACM, 2007.

[Aka04] Yoji Akao. *QFD*: *Quality Function Deployment - Integrating Customer Requirements into Product Design*. Productivity Press, 2004.

[Aka90] Yoji Akao. *Quality Function Deployment: Integrating Customer Requirements into Product Design*. Productivity Press, Cambridge, MA, 1990.

[BDP07] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL'07*, Prague, Czech Republic, 2007.

[BHK98] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th conference on Uncertainty in artificial intelligence*, pages 43--52, 1998.

[BP02] Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'02, pages 136--145, London, UK, 2002. Springer-Verlag.

[Bre96] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123--140, Aug 1996.

[BS97] Marko Balabanovic and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66--72, 1997.

[BSR+05] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *ICML'05*. ACM, 2005.

[Bur02] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331--370, 2002.

[Can02] John Canny. Collaborative filtering with privacy via factor analysis. In *SIGIR'02*, pages 238--245, New York, NY, USA, 2002. ACM.

[CFT06] Yizeng Chen, Richard Y. K. Fung, and Jiafu Tang. Rating technical attributes in fuzzy QFD by integrating fuzzy weighted average method and fuzzy expected value operator. *European Journal of Operational Research*, 174(3):1553--1566, 2006.

[CGM+99] Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. In *SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*, Berkeley, California, 1999. ACM.

[Chu01] P.-T. Chuang. Combining the analytic hierarchy process and Quality Function Deployment for a location decision from a requirement perspective. *The International Journal of Advanced Manufacturing Technology*, 18:842--849, 2001.

[CI11] S. L. Chan and W. H. Ip. A dynamic decision support system to predict the value of customer for new product development. *Decision Support Systems*, 52(1):178 -- 188, 2011.

[CJSM11] Ankur Chaudha, Rajeev Jain, A. Singh, and P. Mishra. Integration of Kano's model into Quality Function Deployment (QFD). *The International Journal of Advanced Manufacturing Technology*, 53:689--698, 2011.

[Coh95] Lou Cohen. *Quality Function Deployment*. Prentice Hall PTR, 1995.

[CQL+07] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *ICML'07*. ACM, 2007.

[CT10] Chien Chin Chen and You-De Tseng. Quality evaluation of product reviews using an information quality framework. *Decision Support Systems*, 50(4):755--768, 2010.

[CT12] Wingyan Chung and Tzu-Liang (Bill) Tseng. Discovering business intelligence from online product reviews: A rule-induction framework. *Expert Systems with Applications*, 1(1):1--10, 2012.

[CW02] Lai-Kow Chan and Ming-Lu Wu. Quality Function Deployment: A literature review. *European Journal of Operational Research*, 143(3):463--497, 2002.

[CZ06] David Cossock and Tong Zhang. Subset ranking using regression. In *COLT'06*, pages 605--619. Springer-Verlag, 2006.

[Das01] Sanmay Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *ICML'01*, pages 74--81, 2001.

[DK04] Mukund Deshpande and George Karypis. Item-based top-N recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):143--177, 2004.

[DK11] Christian Desrosiers and George Karypis. *A Comprehensive Survey of Neighborhood-based Recommendation Methods*, pages 107--144. Springer US, 2011.

[DKK+09] Cristian Danescu-Niculescu-Mizil, Gueorgi Kossinets, Jon Kleinberg, and Lillian Lee. How opinions are received by online communities: A case study on Amazon.com helpfulness votes. In *WWW'09*, pages 141--150. ACM, 2009.

[DL07] Xiaowen Ding and Bing Liu. The utility of linguistic rules in opinion mining. In *SIGIR'07*. ACM, 2007.

[DLY08] Xiaowen Ding, Bing Liu, and Philip S. Yu. A holistic lexicon-based approach to opinion mining. In *WSDM'08*. ACM, 2008.

[DLZ09] Xiaowen Ding, Bing Liu, and Lei Zhang. Entity discovery and assignment for opinion mining applications. In *KDD'09*. ACM, 2009.

[DTCY10] Weifu Du, Songbo Tan, Xueqi Cheng, and Xiaochun Yun. Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon. In *WSDM'10*, pages 111--120, 2010.

[FCT06] Richard Y. K. Fung, Yizeng Chen, and Jiafu Tang. Estimating the functional relationships for Quality Function Deployment under uncertainties. *Fuzzy Sets and Systems*, 157(1):98--120, 2006.

[FFGL00] F. Fabbrini, M. Fusani, S. Gnesi, and G. Lami. Quality evaluation of software requirements specifications. In *Proceeding of Software and Internet Quality Week 2000 Conference*, 2000.

[FLSS03] Yoav Freund, Raj Lyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *The*

Journal of Machine Learning Research, 4:933--969, 2003.

[FPX98] Richard Y. K. Fung, K. Popplewell, and J. Xie. An intelligent hybrid system for customer requirements analysis and product attribute targets determination. *International Journal of Production Research*, 36(1):13--34, 1998.

[FTTC03] Richard Y. K. Fung, Jiafu Tang, Paul Yiliu Tu, and Yizeng Chen. Modelling of Quality Function Deployment planning with resource allocation. *Research in Engineering Design*, 14:247--255, 2003.

[GAO+05] Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. Pulse: Mining customer opinions from free text. In *Proceedings of Advances in Intelligent Data Analysis VI, 6th International Symposium on Intelligent Data Analysis*, pages 121--132, 2005.

[GCXZ10] Xiuli Geng, Xuening Chu, Deyi Xue, and Zaifang Zhang. An integrated approach for rating engineering characteristics' final importance in product-service system development. *Computers and Industrial Engineering*, 59(4):585 -- 594, 2010.

[GE03] Isabelle Guyon and Andre Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157--1182, 2003.

[Gei93] Seymour Geisser. *Predictive Inference: An Introduction*. Chapman and Hall, New York, NY, 1993.

[GG94] Anders Gustafsson and N. Gustafsson. Exceeding customer expectations. In *Proceedings of the 6th Symposium on Quality Function Deployment*, pages 52--57, 1994.

[GH93] Abbie Griffin and John R. Hauser. *The voice of the customer. Marketing Science*, 12:1--27, 1993.

[GM03] Michel Galley and Kathleen McKeown. Improving word sense disambiguation in lexical chaining. In *IJCAI'03*, pages 1486--1488, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[GNOT92] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM - Special issue on information filtering*, 35(12):61--70, 1992.

[GSSZ05] Russell Greiner, Xiaoyuan Su, Bin Shen, and Wei Zhou. Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. *Machine Learning*, 59:297--322, 2005.

[HC09] Xuanjing Huang and W. Bruce Croft. A unified relevance model for opinion retrieval. In *CIKM'09*, pages 947--956. ACM, 2009.

[HGO00] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. *Large margin rank boundaries for ordinal regression*, page 115–132. MIT Press, Cambridge, MA, 2000.

[HKC04] Chang Hee Han, Jae Kyeong Kim, and Sang Hyun Choi. Prioritizing engineering characteristics in Quality Function Deployment with incomplete information: A linear partial ordering approach. *International Journal of Production Economics*, 91(3):235--249, 2004.

[HL04a] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *KDD'04*, pages 168--177, 2004.

[HL04b] Minqing Hu and Bing Liu. Mining opinion features in customer reviews. In *AAAI'04*, pages 755--760, 2004.

[HMP+06] Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. Ontonotes: the 90% solution. In *NAACL'06*, pages 57--60, Stroudsburg, PA, USA, 2006. Association for

Computational Linguistics.

[HOK07] Ishrar Hussain, Olga Ormandjieva, and Leila Kosseim. Automatic quality assessment of SRS text by means of a decision-tree-based text classifier. In *7th International Conference on Quality Software*, pages 209--218, 2007.

[HPFO01] J. A. Harding, K. Popplewell, Richard Y. K. Fung, and A. R. Omar. An intelligent information framework for market driven product design. *Computers in Industry*, 44(1):49--63, 2001.

[HR10] Ahmed Hassan and Dragomir Radev. Identifying text polarity using random walks. In *ACL'10*, pages 395--403, 2010.

[IMK08] Ryu Iida, Diana McCarthy, and Rob Koeling. Gloss-based semantic similarity metrics for predominant sense acquisition. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, pages 561--568, 2008.

[JGDE08] Andreas Janecek, Wilfried Gansterer, Michael Demel, and Gerhard Ecker. On the relationship between feature selection and classification accuracy. In *JMLR Workshop and Conference Proceedings*, pages 90--105, 2008.

[JH09] Wei Jin and Hung Hay Ho. A novel lexicalized HMM-based learning framework for web opinion mining. In *ICML'09*. ACM, 2009.

[JL08] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *WSDM'08*. ACM, 2008.

[JLLZ07] Thorsten Joachims, Hang Li, Tie-Yan Liu, and Chengxiang Zhai. Learning to rank for information retrieval (LR4IR 2007). *SIGIR Forum*, 41:58--62, December 2007.

[Joa02] Thorsten Joachims. Optimizing search engines using clickthrough data. In *KDD'02*, pages 133--142, 2002.

[Joa06] Thorsten Joachims. Training linear SVMs in linear time. In *KDD'06*, pages 217--226. ACM, 2006.

[JZ07] Jing Jiang and Chengxiang Zhai. Instance weighting for domain adaptation in NLP. In *ACL'07*, pages 264--271. Association for Computational Linguistics, 2007.

[KB11] Yehuda Koren and Robert Bell. *Advances in Collaborative Filtering*, pages 145--186. Springer US, 2011.

[KCBC07] C. K. Kwong, Y. Chen, H. Bai, and D. S. K. Chan. A methodology of determining aggregated importance of engineering characteristics in QFD. *Computers and Industrial Engineering*, 53(4):667--679, 2007.

[Ken96] Ron S. Kenett. Software specifications metrics: a quantitative approach to assess the quality of documents. In *19th Convention of Electrical and Electronics Engineers*, pages 166--169, 1996.

[KH04] Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In *COLING'04*. Association for Computational Linguistics, 2004.

[KH05] Soo-Min Kim and Eduard Hovy. Identifying opinion holders for question answering in opinion texts. In *AAAI'05 Workshop: Question Answering in Restricted Domains*, pages 20--26, 2005.

[KJ97] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273--324, 1997.

[KMC05] Rob Koeling, Diana McCarthy, and John Carroll. Domain-specific sense distributions and predominant sense acquisition. In *HLT'05*, pages 419--426, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[KMDE00] Kwang-Jae Kim, Herbert Moskowitz, Anoop Dhingra, and Gerald Evans. Fuzzy multicriteria models for Quality Function Deployment. *European Journal of Operational Research*, 121(3):504--518, 2000.

[KMM+04] Jaap Kamps, Maarten Marx, Robert J. Mokken, and Maarten de Rijke. Using wordnet to measure semantic orientation of adjectives. In *LREC'04*, pages 1115--1118, 2004.

[KWC09] C. K. Kwong, T. C. Wong, and K. Y. Chan. A methodology of generating customer satisfaction models for new product development using a neuro-fuzzy approach. *Expert Systems With Applications*, 36(8):11262--11270, 2009.

[KYCC11] C. K. Kwong, Y. Ye, Y. Chen, and K. L. Choy. A novel fuzzy group decision-making approach to prioritizing engineering characteristics in QFD under uncertainties. *International Journal of Production Research*, 49(19):5801--5820, 2011.

[Lan95] Ken Lang. Newsweeder: learning to filter netnews. In *ICML'95*, pages 331--339. ACM, 1995.

[LCL+07] Jingjing Liu, Yunbo Cao, Chin-Yew Lin, Yalou Huang, and Ming Zhou. Low-quality product review detection in opinion summarization. In *EMNLP-CoNLL'07*, pages 334--342, 2007.

[Lee07a] Thomas Lee. Constraint-based ontology induction from online customer reviews. *Group Decision and Negotiation*, 16(3):255--281, 2007.

[Lee07b] Thomas Lee. Needs-based analysis of online customer reviews. In *ICEC'07*, volume 258, pages 311--318. ACM, 2007.

[LH09] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *CIKM'09*, pages 375--384, 2009.

[LHAY08] Yang Liu, Xiangji Huang, Aijun An, and Xiaohui Yu. Modeling and predicting the helpfulness of online reviews. In *ICDM'08*, pages 443--452. IEEE Computer Society, 2008.

[LHC05] Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: Analyzing and comparing opinions on the web. In *WWW'05*. ACM, 2005.

[LHH+10] Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. Structure-aware review mining and summarization. In *COLING'10*, pages 653--661. Association for Computational Linguistics, 2010.

[Liu09] Tie-Yan Liu. *Learning to Rank for Information Retrieval*, volume 3 of *Foundations and Trends in Information Retrieval*. now Publishers inc, Hanover, MA, 2009.

[LL07] Ying Liu and Han Tong Loh. Corpus building for corporate knowledge discovery and management: a case study of manufacturing. In *Proceedings of the 11th international conference, KES 2007 and XVII Italian workshop on neural networks conference on Knowledge-based intelligent information and engineering systems: Part I*, KES'07/WIRN'07, pages 542--550, 2007.

[LLL07] Ying Liu, Wen Feng Lu, and Han Tong Loh. Knowledge discovery and management for product design through text mining - a case study of online information integration for designers. In *Proceedings of the 16th international Conference on Engineering Design*, pages 329/1--12, 2007.

[LLL09] Soon Chong Johnson Lim, Ying Liu, and Wing Bun Lee. Faceted search and retrieval based on semantically annotated product family ontology. In *WSDM'09 Workshop on Exploiting Semantic Annotations in Information Retrieval*. ACM, 2009.

[LPH08] Jumin Lee, Do-Hyung Park, and Ingoo Han. The effect of negative online consumer reviews on product attitude: An information processing view. *Electronic Commerce Research and Applications*, 7(3):341 -- 352, 2008.

[LTX07] Xin Lai, Kay-Chuan Tan, and Min Xie. Optimizing product design using quantitative Quality Function Deployment: a case study. *Quality and Reliability Engineering International*, 23(1):45--57, 2007.

[LY05] Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering.*IEEE Transactions on Knowledge and Data Engineering*, 17(4):491--502, 2005.

[LYZW11] Yijun Li, Qiang Ye, Ziqiong Zhang, and Tienan Wang. Snippet-based unsupervised approach for sentiment classification of Chinese online reviews. *International Journal of Information Technology and Decision Making*, 10(6):1097--1110, 2011.

[LZ09] Kang Liu and Jun Zhao. Cross-domain sentiment classification using a two-stage method. In *CIKM'09*, pages 1717--1720, 2009.

[LZFW10] Binyang Li, Lanjun Zhou, Shi Feng, and Kam-Fai Wong. A unified graph model for sentence-based opinion retrieval. In *ACL'10*, pages 1367--1375, 2010.

[LZG09] Zhaoling Li, Dongling Zhang, and Qisheng Gao. A grey method of prioritizing engineering characteristics in QFD. In *Control and Decision Conference*, pages 3443 --3447, June 2009.

[LZS09] Tao Li, Yi Zhang, and Vikas Sindhwani. A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In *ACL'09*, pages 244--252, 2009.

[MBR98] Raymond J. Mooney, Paul N. Bennett, and Loriene Roy. Book recommending using text categorization with extracted information. In *AAAI'98 Workshop: Learning for Text Categorization*, pages 70--74, 1998.

[MC03] Rada Mihalcea and Timothy Chklovski. Open mind word expert: Creating large annotated data collections with web users' help. In *Proceedings of the EACL Workshop on Linguistically Annotated Corpora*, page 53–60, 2003.

[MG99] Dunja Mladenic and Marko Grobelnik. Feature selection for unbalanced class distribution and naïve Bayes. In *ICML'99*, pages 258--267, 1999.

[MGL09] Prem Melville, Wojciech Gryc, and Richard D. Lawrence. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *KDD'09*, pages 1275--1284. ACM, 2009.

[MH98] Kurt Matzler and Hans H. Hinterhuber. How to make product development projects more successful by integrating Kano's model of customer satisfaction into Quality Function Deployment. *Technovation*, 18(1):25--38, 1998.

[MHN+07] Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. Structured models for fine-to-coarse sentiment analysis. In *ACL'07*, 2007.

[Mih05] Rada Mihalcea. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *HLT'05*, pages 411–418, 2005.

[MKWC07] Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. Unsupervised acquisition of predominant word senses.

Computational Linguistics, 33(4):553--590, 2007.

[MLD09] Qingliang Miao, Qiudan Li, and Ruwei Dai. Amazing: A sentiment mining and retrieval system. *Expert Systems With Applications*, 36(3):7192--7198, 2009.

[MLW+07] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and Chengxiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW'07*, pages 171--180, 2007.

[MMN02] Prem Melville, Raymod J. Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI'02*, pages 187--192, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[MTCK08] Li-Feng Mu, Jia-Fu Tang, Yi-Zeng Chen, and C. K. Kwong. A fuzzy multi-objective model of QFD product planning integrating Kano model. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 16(6):793 813, 2008.

[MWG09] Yunhe Mu, Yinglin Wang, and Jianmei Guo. Extracting software functional requirements from free text documents. In *International Conference on Information and Multimedia Technology*, pages 194--198, 2009.

[Nal04] Ramesh Nallapati. Discriminative models for information retrieval. In *SIGIR'04*, pages 64--71. ACM, 2004.

[Nav09] Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):1--69, 2009.

[OS09] Michael P. O'Mahony and Barry Smyth. Learning to recommend helpful hotel reviews. In *RecSys'09*, pages 305--308. ACM, 2009.

[Ott09] Jahna Otterbacher. “helpfulness” in online communities: A measure of message quality. In *CHI'09*, pages 955--964. ACM, 2009.

[Paz99] Michael J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review - Special issue on data mining on the Internet*, 13(5-6):393--408, December 1999.

[PB97] Michael Pazzani and Daniel Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning - Special issue on multistrategy learning*, 27(3):313--331, 1997.

[PE05] Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, HLT/EMNLP*, pages 339--346, 2005.

[Ped06] Ted Pedersen. Unsupervised corpus-based methods for WSD. In *Word Sense Disambiguation: Algorithms and Applications*, pages 133--166. Springer, 2006.

[PG08] Jantima Polpinij and Aditya Ghose. An automatic elaborate requirement specification by using hierarchical text classification. In *2008 International Conference on Computer Science and Software Engineering*, volume 1, pages 706--709, 2008.

[PHLG00] David M. Pennock, Eric Horvitz, Steve Lawrence, and C. Lee Giles. Collaborative filtering by personality diagnosis: A hybrid memory and model-based approach. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 473--480, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[PL02] Patrick Pantel and Dekang Lin. Discovering word senses from text. In *KDD'02*, pages 613--619, New York, NY, USA, 2002. ACM.

[PL04] Bo Pang and Lillian Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL'04*, 2004.

[PLV02] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP'02*, volume 10, pages 79--86. Association for Computational Linguistics, 2002.

[PNS+10] Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Cross-domain sentiment classification via spectral feature alignment. In *WWW'10*, pages 751--760. ACM, 2010.

[QLBC09] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Expanding domain sentiment lexicon through double propagation. In *IJCAI'09*, pages 1199--1204, 2009.

[QZHZ09] Likun Qiu, Weishi Zhang, Changjian Hu, and Kai Zhao. SELC:a self-supervised model for sentiment classification. In *CIKM'09*. ACM, 2009.

[QZT+08] Tao Qin, Xu-Dong Zhang, Ming-Feng Tsai, De-Sheng Wang, Tie-Yan Liu, and Hang Li. Query-level loss functions for information retrieval. *Information Processing and Management*, 44(2):838--855, 2008.

[QZW+07] Tao Qin, Xu-Dong Zhang, De-Sheng Wang, Tie-Yan Liu, Wei Lai, and Hang Li. Ranking with multiple hyperplanes. In *SIGIR'07*, pages 279--286, 2007.

[RAR05] Joao Ramires, Pedro Antunes, and Ana Respicio. Software requirements negotiation using the software Quality Function Deployment. In *Proceedings of the 11th international conference on Groupware: design, Implementation, and Use*, pages 308--324, 2005.

[RPMS11] Leonardo Rigutini, Tiziano Papini, Marco Maggini, and Franco Scarselli. Sortnet: Learning to rank by a neural preference function. *IEEE Transactions on Neural Networks*, 22:1368--1380, 2011.

[RRS11] Francesco Ricci, Lior Rokach, Bracha Shapira. Introduction to Recommender Systems Handbook, *Recommender Systems Handbook*, Springer, pages 1-35, 2011.

[Saa80] Thomas L. Saaty. *The Analytic Hierarchy Process*. McGraw-Hill, New York, 1980.

[Sch98] Hinrich Schutze. Automatic word sense discrimination. *Computational Linguistics - Special issue on word sense disambiguation*, 24(1):97--123, 1998.

[SK06] Xiaoyuan Su and Taghi M. Khoshgoftaar. Collaborative filtering for multi-class data using belief nets algorithms. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, pages 497--504, 2006.

[SKKR01] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW'01*, pages 285--295, 2001.

[SL03] Amnon Shashua and Anat Levin. Ranking with large margin principle: Two approaches. In *NIPS'03*, page 937–944, 2003.

[SSG+03] Bin Shen, Xiaoyuan Su, Russell Greiner, Petr Musilek, and Corrine Cheng. Discriminative parameter learning of general Bayesian network classifiers. In *Proceedings of 15th IEEE International Conference on Tools with Artificial Intelligence*, pages 296 -- 305, 2003.

[STX00] X. X. Shen, K. C. Tan, and M. Xie. An integrated approach to innovative product development using Kano's model and QFD. *European Journal of Innovation Management*, 3(2):91--99, 2000.

[Tur02] Peter D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL'02*, pages 417--424, 2002.

[VO05] Slobodan Vucetic and Zoran Obradovic. Collaborative filtering using a regression-based approach. *Knowledge and Information Systems*, 7(1):1--22, 2005.

[VS06] Carlos Videira and David Ferreira Alberto Rodrigues da Silva. A linguistic patterns approach for requirements specification. In *32nd EUROMICRO Conference on Software Engineering and Advanced Applications*, pages 302--309, 2006.

[Wan08] Xiaojun Wan. Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis. In *EMNLP'08*, pages 553--561. Association for Computational Linguistics, 2008.

[Wan09] Xiaojun Wan. Co-training for cross-lingual sentiment classification. In *ACL'09*, pages 235--243. Association for Computational Linguistics, 2009.

[WC11] Ying-Ming Wang and Kwai-Sang Chin. Technical importance ratings in fuzzy QFD by integrating fuzzy normalization and fuzzy weighted average. *Computers and Mathematics with Applications*, 62(11):4207 -- 4221, 2011.

[WLW05] H. H. Wu, A. Y. H. Liao, and P. C. Wang. Using grey theory in Quality Function Deployment to analyse dynamic customer requirements. *The International Journal of Advanced Manufacturing Technology*, 25:1241--1247, 2005.

[WS06] Hsin-Hung Wu and Jiunn-I Shieh. Using a markov chain model in

Quality Function Deployment to analyse customer requirements. *The International Journal of Advanced Manufacturing Technology*, 30:141--146, 2006.

[WXG98] H. Wang, M. Xie, and T. N. Goh. A comparative study of the prioritization matrix method and the analytic hierarchy process technique in Quality Function Deployment. *Total Quality Management*, 9(6):421--430, 1998.

[XLLS11] Kaiquan Xu, Stephen Shaoyi Liao, Jiexun Li, and Yuxia Song. Mining comparative opinions from customer reviews for competitive intelligence. *Decision Support Systems*, 50:743--754, 2011.

[XLW+08] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank:theory and algorithm. In *ICML'08*. ACM, 2008.

[YF02] David Yarowsky and Radu Florian. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, 8(4):293--310, 2002.

[YLHA10] Xiaohui Yu, Yang Liu, Xiangji Huang, and Aijun An. A quality-aware model for sales prediction using reviews. In *WWW'10*, pages 1217--1218. ACM, 2010.

[YP97] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *ICML'97*, pages 412--420, 1997.

[YST+04] Kai Yu, Anton Schwaighofer, Volker Tresp, Xiaowei Xu, and Hans-Peter Kriegel. Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):56--69, 2004.

[ZC08] Taras Zagibalov and John Carroll. Automatic seed word selection for unsupervised sentiment classification of Chinese text. In *COLING'08*, pages 1073--1080. Association for Computational Linguistics, 2008.

[ZJYM08] Wei Zhang, Lifeng Jia, Clement Yu, and Weiyi Meng. Improve the effectiveness of the opinion retrieval and opinion polarity classification. In *CIKM'08*, pages 1415--1416, 2008.

[ZLL09] Jiaming Zhan, Han Tong Loh, and Ying Liu. Gather customer concerns from online product reviews - a text summarization approach. *Expert Systems With Applications*, 36(2 Part 1):2107--2115, 2009.

[ZLW08] Jun Zhao, Kang Liu, and Gen Wang. Adding redundant features for CRFs-based sentence sentiment classification. In *EMNLP'08*, volume 10, pages 117--126. Association for Computational Linguistics, 2008.

[ZT10a] Richong Zhang and Thomas Tran. An information gain-based approach for recommending useful product reviews. *Knowledge and Information Systems*, 26:1--16, 2010.

[ZT10b] Richong Zhang and Thomas Tran. A novel approach for recommending ranked user-generated reviews. *Advances in Artificial Intelligence*, 6085:324--327, 2010.

[ZWTZ09] Jingbo Zhu, Huizhen Wang, Benjamin K. Tsou, and Muhua Zhu. Multi-aspect opinion polling from textual reviews. In *CIKM'09*, pages 1799--1802. ACM, 2009.

[ZY08] Min Zhang and Xingyao Ye. A generation model to unify topic relevance and lexicon-based sentiment for opinion retrieval. In *SIGIR'08*, pages 411--418. ACM, 2008.