PREFERENCE PREDICTION THROUGH FEATURE-BASED COLLABORATIVE FILTERING OF TEXTUAL REVIEWS

BY

YOGESH KAKODKAR

A thesis submitted to the

GRADUATE SCHOOL-NEW BRUNSWICK

RUTGERS, THE STATE UNIVERSITY OF NEW JERSEY

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

GRADUATE PROGRAM IN ELECTRICAL AND COMPUTER ENGINEERING

written under the direction of

Dr. Ivan Marsic

_____

_____

_____

_____

New Brunswick, New Jersey

May 2011

ABSTRACT OF THE THESIS

Preference Prediction through Feature-based Collaborative Filtering of Textual Reviews

by YOGESH KAKODKAR

Thesis Director: Dr. Ivan Marsic

Text reviews are often used by users to decide whether to buy a product or watch a movie or dine in a restaurant. Most of these reviews are raw text and lack a formal structure. Computers cannot easily understand and interpret these reviews to analyze and aggregate them. Users have to manually read through these reviews to find the useful information about the concerned restaurant. We use the topical and sentimental information compiled from raw textual reviews to understand user preferences. We use these preferences to cluster similar users together and then predict users' topical feelings towards the restaurants for which they may be requesting information and to make suitable recommendations.

Users have similarities in their preferences for particular topics under which the restaurants have been reviewed. Therefore, we can soft-cluster them using these similarities extracted from their reviewing history. These cluster membership probabilities help us make predictions about the user's sentiments in each topic for the target restaurant. Our results show our accuracy for predicting these sentiments and show that we can provide recommendations to users in most topics for the target restaurant.

Acknowledgements

I wish to dedicate this thesis to my mother, Dr. Rita Kakodkar, who has been my sole inspiration and to my grandmother, Kusum Kakodkar, for her complete support. I must also thank my late aunt, Sulekha Kakodkar, without whose blessings, I would never have been able to pursue my Master of Science degree.

My thesis director, Dr. Ivan Marsic, has been very supportive and helpful throughout my research process. I would like to thank Dr. Amélie Marian for her guidance and support for my research.

I would also like to thank Dr. Wade Trappe and Dr. Kristin Dana for being on my thesis defense committee.

I am grateful to Gayatree Ganu for her great help through answering all my doubts and being the great push I needed in this project. I would also like to thank my other lab mates Minji Wu and Jinyun Yan for making this journey fun, yet productive.

I would like to thank Linda Asaro, Marcy Cohen and other staff at the Center for International Faculty and Student Services for taking care of my visa-related issues. I am very grateful to Urmi Otiv and Mohini Mukherjee for letting me become a part of their International Student Orientation Volunteer Program which enriched my stay at Rutgers University beyond bounds. I am extremely grateful to Carissa McCarthy for organizing the International Friendship Program through which I managed to make a lot of great friends in a very short time.

Lastly, I would like to thank all my wonderful friends whom I shall miss when I leave Rutgers University.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

User-generated content is rapidly changing the world we live in. We, as web users, follow peer-authored posts and reviews and incorporate them into out decision making process. Yet, very few web sites have made a conscious effort to understand this textual content in user reviews and harvest user preferences from them. A lot can be learnt from the text in such reviews about their author's behavior and likes and dislikes. This information can help web sites cater better to their clientele.

Most of these reviews are poorly organized and lack structure. Because of this, users have to manually scan through all the reviews to find the relevant ones. A simple keyword search is useless since the same keywords repeat in both good and bad reviews [1]. Identifying structured information from free-form text is a challenging task since most users routinely enter informal text with poor spelling and grammar. Our work uses the classification techniques detailed in [8, 10] as well as in Appendix A. In [8, 10], the authors describe how features and sentiments can be extracted from free-form text. User experience can be enhanced if these underlying features and the user's sentiments towards these features were incorporated into the algorithm which generates the results displayed to the user. Furthermore, this fine-grained information can be used for studying a user's preferences towards individual product features.

Today's web users want relevant information returned to them in the most concise format which the user can understand fast and easily. This is proved by the success of web portals such as Amazon.com and Netflix.com which not only search for the relevant product but also recommend new products. However, recommendation systems on these portals are built on top of structured metadata about these products [17]. On similar grounds, we use only textual data to understand fine-grained user preferences for different features from the written text and employ these preferences to predict the user's sentiments for these features for a target item.Using these text-based predictions we make recommendations to users.

Our work is a part of the User Review Structure Analysis (URSA) project and incorporates the methods and results described by the authors in [8, 10]. Our work in this thesis builds upon the work in [8, 10]. Using the structured information on review text data, we predict the user's preferences in each feature for target restaurants. We employ the iterative Information Bottleneck algorithm [23, 9] to cluster the users depending on the similarity of their categorical preferences. Using the categorical preferences of similarly clustered users who have rated the test restaurant, we estimate the test user's categorical preferences for the test restaurant.

Our contributions to this project are as follows

1. We propose the use of a feature-based soft-clustering of users depending on their reviewing history for each feature as opposed to the review-based clustering of users proposed in [9].

2. We make feature-based qualitative predictions for each test review instead of a quantitative prediction method proposed by the authors in [8, 10].

The remainder of the paper is organized as follows; in chapter 2, we introduce our data set and the preliminary processing which was conducted on the data. In chapter 3, we describe our motivations and methodology for preference prediction. In chapter 4, we explore the experimental setup and the preliminary results. In chapter 5, we study our results and make an in-depth analysis of our findings. In chapter 6, we highlight other related works on prediction and recommendation systems. Finally, in chapter 7 we conclude the paper.

Chapter 2

Data Set and Sentence Classification

In this section, we describe the approach to harvest data from the textual reviews which users have written for various restaurants. We describe our data set in section 2.1. In section 2.2, we highlight the sentence classification methodology which we have used. This preliminary analysis has been described in [8].

2.1 Data Set

Our data corpus contains over 50,000 restaurant reviews from Citysearch New York. All these reviews were extracted from this web site over the course of one week in 2006.

The corpus also contains structured information (location, cuisine, etc.) for 5531 restaurants for whom the reviews have been written. There are a total of 52264 reviews, of which 1359 are editorial reviews and the remaining are user reviews. Reviews contain structured metadata (star rating, date, etc.) along with text. Typically reviews are small; user reviews have 5.28 sentences on an average. The reviews in our corpus have been written by 32284 distinct users. We have only unique username information for each of these users.

The data set is sparse. Restaurants typically have only a few reviews, with only 1388 restaurants having more than 10 reviews. Users typically review few restaurants. Only 299 users (non-editorial) have rated more than 10 restaurants.

2.2 Review Classification

Analysis of the data reveals that all sentences written in the reviews can be classified into six categories, such as Food, Price, Staff, Ambience, Anecdotes and Miscellaneous. These dimensions

focus on particular aspects of restaurants. The first four categories are typical parameters of restaurant ratings (e.g. Zagat). Anecdotal sentences are sentences which describe the reviewer's personal experience or context, but do not usually provide information on the restaurant's quality (e.g. *"I knew upon visiting NYC that I wanted to try an original deli"*). The Miscellaneous category captures sentences that do not belong to the other five categories and include sentences that are general recommendations (e.g. *"Your friends shall thank you for introducing them to this gem!"*). Sentence categories are not mutually exclusive and overlap is allowed.

In addition to sentence categories, sentences have an associated sentiment: Positive, Negative, Neutral, or Conflict. Users often seem to compare and contrast good and bad aspects; this mixed sentiment is captured by the Conflict category (e.g. *"The food here is rather good, but only if you like to wait for it"*).

All sentences in the corpus were classified by annotating them with at least one category and one sentiment. Using these tags, a count of the composition of any particular review was made. For example, a user-written review may have 3 sentences about food, of which 2 are positive and 1 is negative, and 1 negative sentence about price. Using these counts, we make our predictions as described in chapter 3.

Sentence Classification is described in more detail by the authors in [8] and [10]. We have also described their approach in Appendix A.

Chapter 3

Preference Prediction

In this chapter, we first describe our hypothesis in section 3.1. We then move on to explain our methodology and the process we employed to verify our hypothesis in section 3.2.

3.1 Hypothesis

When a user writes a review for a restaurant, the sentence classification gives us a composition for the user's sentiments towards each of the categories for the concerned restaurant. Using similar composition data for user-restaurant pairs from the data set, we can cluster similar users and then predict a test user's sentiments towards a test restaurant for each of the six categories using collaborative filtering. The method proposed in [8] uses all categories and sentiment combinations to develop one clustering system.

Our argument is that users may like the same ambience as some of their friends but may have tastes in food similar to their parents. Therefore, the clustering for each category, such as food, ambience, etc., should be separate.

Using the different clusterings for each category, the category-sentiment values for the test review can be predicted. Using these predicted values, we can tell the overall sentiment (positive, negative or neutral) the test user will have for each of the categories for the test restaurant thus giving a qualitative prediction of the user assessment of the restaurant.

3.2 Method

Our method employs a category-based soft-clustering method unlike the full review based clustering used in [8]. Users are marked as similar based on their tendencies to write similar sentences about food (or other categories) for the restaurants they have reviewed. The iterative Information

Bottleneck method gives a matrix of probabilities with which a certain user can belong to a certain cluster.

Once the users have been clustered, their cluster membership probability is used to predict the test user's sentiment for a particular category for the test restaurant.We predict the values for each of the four sentiments for every category. Then, using a combination function, we compute the overall sentiment the user may have toward the restaurant in the concerned category.

3.2.1 Clustering

Users are clustered using the iterative Information Bottleneck (iIB) method as described in more detail in [23] and in Appendix B.This method takes an input of the joint probability distribution P(X,Y) and the cluster cardinality T. This method clusters the input variable X into T clusters while ensuring that T maintains the maximum information about Y. The use of this method was first proposed by the authors of [9]. However, they cluster the users based on the entire review. We, on the other hand, get a more fine-grained clustering which is distinct for each feature.

Our goal is to cluster our users based on their preferences in order to group like-minded users together. Hence, our 30k+ users shall be represented by the variable X. We plan to cluster the users based on one category at a time. We, therefore, use the (restaurant,sentiment) pairs as features for clustering. The variable Y, thus, represents the sentiments of the users toward restaurants with respect to a particular category (Food, Price, etc.).

Consider the following artificial example with a corpus of five users and three restaurants as shown in table 3.1. The data in table 3.1 is only for food related sentences written by the users for the restaurants and has been normalized for every restaurant. As per our method, we shall cluster these five users based on these four sentiments (Positive, Negative, Neutral and Conflict). Suppose that the iIB algorithm was to cluster these 5 users into 3 clusters. The output of the iIB algorithm would be the *cluster membership probabilities* as shown in 3.2. $User_2$ and $User_3$ are similarly clustered since their rating habits are similar for similar restaurants. The cluster membership probabilities matrix is used for making predictions as described next.

	$Restaurant_1$				$Restaurant_2$				$Restaurant_3$			
	Pos	Neg	Neu	Con	Pos	Neg	Neu	Con	Pos	Neg	Neu	Con
$User_1$	0.6	0.2	0.2	0.0	-	-	-	-	-	-	-	-
$User_2$	0.3	0.6	0.1	0.0	0.9	0.0	0.1	0.0	0.6	0.1	0.2	0.1
$User_3$	0.1	0.7	0.15	0.05	-	-	-	-	0.8	0.2	0.0	0.0
$User_4$	0.9	0.05	0.05	0.0	0.3	0.4	0.2	0.1	-	-	-	-
$User_5$	-	-	-	-	-	-	-	-	0.0	0.7	0.3	0.0

Table 3.1: Input Matrix to the iIB Algorithm

	$Cluster_1$	$Cluster_2$	$Cluster_3$
$User_1$	0.04	0.057	0.903
$User_2$	0.396	0.202	0.402
$User_3$	0.38	0.118	0.502
$User_4$	0.576	0.015	0.409
$User_5$	0.006	0.99	0.004

Table 3.2: Cluster membership probabilities generated by the iIB algorithm

3.2.2 Prediction

After the n^{th} iteration, the iIB algorithm converges and outputs the cluster membership probabilities matrix which clusters the X users into T clusters with probabilities given in $P_n(x|t)$ as shown in table 3.2. These probabilities are used to find the weights to be associated with the users who have reviewed the test restaurant. The predicted value for the sentiment for the test user-restaurant case is the weighted average of the value of the sentiment of all other users who have reviewed the restaurant.

To find the value of the sentiment for a test review category, we first find the contribution of each cluster. We compute the cluster contribution for each cluster by taking the weighted average of all the user sentiment values who have rated the restaurant.

To predict the value of any sentiment of a test user u_t for a test restaurant r_t, let us denote this prediction mathematically as $P(u_t, r_t)$. Let n users have rated this restaurant giving it sentiment ratings such as $Rating(u_1, r_t)$, $Rating(u_2, r_t)$,...,$Rating(u_n, r_t)$. Also let each user u_x belong to a cluster c_w with a cluster probability denoted by $ClusterProb(u_x, c_w)$. Thus the cluster contribution can now be denoted as

$$Contribution(c_i, r_t) = \frac{\sum_{j=1}^{n} ClusterProb(u_j, c_i) * Rating(u_j, r_t)}{\sum_{j=1}^{n} ClusterProb(u_j, c_i)} \tag{3.1}$$

We have M clusters, c_1, c_2,...,c_m, say. To find the final prediction, we need to take the weighted average of all the cluster contributions using the cluster membership probabilities for the test user u_t. Thus the final prediction $P(u_t, r_t)$ is given by the formula:

$$P(u_t, r_t) = \frac{\sum_{i=1}^{m} ClusterProb(u_t, c_i) * Contribution(c_i, r_t)}{\sum_{i=1}^{m} ClusterProb(u_t, c_i)} \tag{3.2}$$

3.2.3 Final sentiment

After we have predicted each of the four sentiments for every category of the six categories such as food, price, etc., for the user-restaurant test case, we now move to determine the sentiment the test user may have for the test restaurant in each of the six categories.

We compute the sentiment level for positive sentiment in each of the categories. Let the predicted sentiments for each category C_i, where $i = 1$ to 6, (for each category and sentiment label, refer Appendix C) for the test user u_t and restaurant r_t be denoted by $S(u_t, r_t, C_i, S_j)$, where $j = 1$ to 4 and so S_j indicates each of the four categories listed in table C.2. Thus, the final Positive level $PosLev(u_t, r_t, C_i)$ is given by

$$PosLev(u_t, r_t, C_i) = \frac{S(u_t, r_t, C_i, S_1) + \frac{S(u_t, r_t, C_i, S_4)}{2}}{\sum_{j=1}^{4} S(u_t, r_t, C_i, S_j)} \tag{3.3}$$

The conflict sentiment, $S(u_t, r_t, C_i, S_4)$, is assigned to those sentences which can be classified as both positive and negative and hence, is added to the concerned sentence to make up for the ambiguous nature of this sentiment. The Neutral sentiment, on the other hand, is assigned to those sentences which can be classified as neither positive nor negative. Therefore, we do not add the neutral sentiment $S(u_t, r_t, C_i, S_3)$ in our calculations.

We define two thresholds $Threshold_{Positive}(C_i)$ and $Threshold_{Negative}(C_i)$. Both of these thresholds, $Threshold_{Positive}(C_i)$ and $Threshold_{Negative}(C_i)$, can be chosen according to the distribution of the actual and predicted values of $PosLev(u_t, r_t, C_i)$. We explain this in depth in section 4.3. Using these thresholds, the final sentiment is computed as per algorithm 1.

Algorithm 1 Determining the Final Sentiment

if $PosLev(u_t, r_t, C_i) > Threshold_{Positive}(C_i)$ **then**
  Final Sentiment is Positive
else if $PosLev(u_t, r_t, C_i) < Threshold_{Negative}(C_i)$ **then**
  Final Sentiment is Negative
else
  Final Sentiment is Neutral
end if

Chapter 4

Experimental Analysis

The Sentence Analysis module gives us an output with the proportion of the total number of sentences in the review, that is of a particular category-sentiment (such as Food-Positive, Price-Neutral and so on) combination. Users may be similar to some users depending on their food tastes, but similar to a completely different set of users for their choice of dining ambience. We, therefore, cluster the users separately depending on their ratings for a particular category like Food or Staff.

4.1 Preliminary Setup

The first observation we made, from the data corpus, was that most users do not write a review containing sentences belonging to each of the six categories. This is a major problem since we cannot test if we predicted the correct sentiment if there is no record of the user ever writing about that category in the actual review. Also, there are fewer users who write negative sentences in any category for most restaurants. As a result, there is very little evidence to help us make predictions about the negative sentiment leading us to a cold start during prediction of the negative sentiment. Hence, we developed six different test sets, each for one of the six categories, such that we have enough evidence to analyze our algorithm and avoid the cold start problem as much as possible.

For our algorithm to work well, we needed to reduce the sparsity of the training data available for making the prediction. Therefore, we enforced the following criteria while choosing the test cases.

For a review, from the corpus, to be considered as a test case,

1. its user should have rated at least four other restaurants in the same category and
2. the restaurant should have been rated by at least two other users in the same category.

We thus generated six different test sets for each category of sentences. Each test set has 215 reviews written by 215 different users.

4.2 Experiments

Each category has been experimented with separately. Each category has its own test set and training data. We constructed training data for each test set by blanking out the test cases from the training data for that category.

The training data contains a matrix of values for all four sentiments for that category written by all users for all the restaurants, except the test cases, of course. Each matrix was first row-normalized and then matrix-normalized to generate the a-priori data for the iterative Information Bottleneck clustering algorithm.

The iterative Information Bottleneck algorithm gives us the matrix of cluster membership probabilities for each user. We use each of the six such matrices generated separately for each category to predict the values for the test cases. The prediction mechanism, described in detail in section 3.2.2, gives us the predictions for what the composition would be for that category, in each of the test reviews.

4.3 Distribution of the Predicted and Actual Positive Levels

After we predicted the values for each component sentiment of the concerned category for each review in the test set, we computed the value of $PosLev(u_t, r_t, C_i)$ from these predictions. We also computed these values using the actual values of the sentiments for that category for all the reviews in the test set. We then plotted a histogram of the distribution of these values to see the similarity between the actual and predicted sentiments.

The histograms in fig 4.1 have been plotted by dividing the values into 20 bins from 0.0 to 1.0. The blue plots are for the actual values and the red plots are for the predicted values of $PosLev(u_t, r_t, C_i)$.

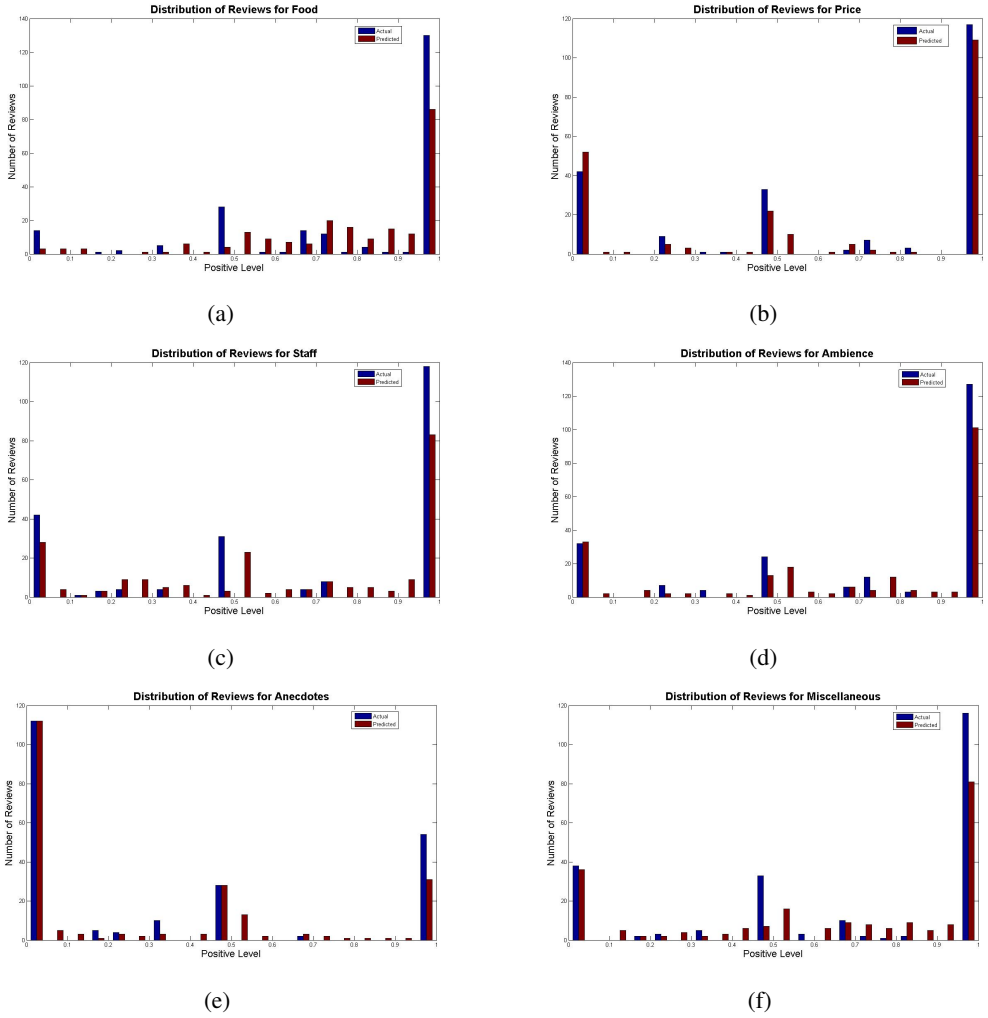


Figure 4.1: Distribution of Reviews

We observe that the distributions of the actual and predicted values of $PosLev(u_t, r_t, C_i)$ for the concerned category for reviews in the test sets are almost similar to each other. There are three distinct peaks in the actual distributions for *Positive* (at to 1.0), *Negative* (at 0.0) and for *Neutral* (at 0.5). The predicted distributions also show peaks for *Positive* (around 0.95), *Negative* (around 0.1) and for *Neutral* (around 0.5).

The existence of three distinct peaks in the actual and predicted distributions, leads us to hypothesize that if we set two thresholds on $PosLev(u_t, r_t, C_i)$, such as $Threshold_{Positive}(C_i)$ and $Threshold_{Negative}(C_i)$, then the reviews can be classified effectively into their final sentiments, such as *Positive, Negative* and *Neutral*, using algorithm 1.

Chapter 5

Results

We now study the performance of our two thresholds, $Threshold_{Positive}(C_i)$ and $Threshold_{Negative}(C_i)$, using the precision and recall graphs for *Positive* and *Negative* sentiment prediction.

5.1 Performance Evaluation

As mentioned in section 4.1, we have distinct test sets for each of the six categories. Hence, we evaluate the performance of our method by computing the precision and recall for each of the categories separately. In order to explain our method for computing the precision and recall values, we introduce the following notations; let $N_{C_i}(S_{actual}, S_{predicted})$ be the number of reviews with the actual sentiment S_{actual} and the predicted sentiment $S_{predicted}$ for the category C_i. A sentiment S_j can be either *Positive, Negative* or *Neutral*. Thus $\neg S_j$ can be either *Not Positive, Not Negative* or *Not Neutral* respectively. We can therefore compute precision using equation 5.1 and recall using equation 5.2.

$$Precision_{C_i}(S_j) = \frac{N_{C_i}(S_j, S_j)}{N_{C_i}(S_j, S_j) + N_{C_i}(\neg S_j, S_j)} \tag{5.1}$$

$$Recall_{C_i}(S_j) = \frac{N_{C_i}(S_j, S_j)}{N_{C_i}(S_j, S_j) + N_{C_i}(S_j, \neg S_j)} \tag{5.2}$$

Precision is, thus, the ratio of the number of reviews correctly predicted with a certain sentiment to the total number which have been predicted with the same sentiment. Recall is the ratio of the number of reviews correctly predicted to the total number of reviews which have the same actual sentiment.

We also evaluate the accuracy and F1 measures for our predictions. Accuracy is ratio of the

number correctly predicted to the total number of reviews in the test set. Accuracy is evaluated using equation 5.3.

$$Accuracy_{C_i}(S_j) = \frac{N_{C_i}(S_j, S_j) + N_{C_i}(\neg S_j, \neg S_j)}{N_{C_i}(S_j, S_j) + N_{C_i}(\neg S_j, S_j) + N_{C_i}(S_j, \neg S_j) + N_{C_i}(\neg S_j, \neg S_j)} \tag{5.3}$$

The $Threshold_{Positive}(C_i)$ in algorithm 1, was varied from 0.0 to 1.0 with an increment of 0.1 for both the actual and predicted review sentiments. We thus computed the precision and recall for predicting the *Positive* final sentiment for all the reviews in the test sets for each of the categories. A review is given a *Positive* sentiment if its positive level is greater than the positive threshold, $PosLev(u_t, r_t, C_i) > Threshold_{Positive}(C_i)$ and *not positive* otherwise. We also plot the Accuracy and F1 measure.

Similarly, $Threshold_{Negative}(C_i)$ in algorithm 1, was varied from 0.0 to 1.0 with an increment of 0.1 for both the actual and predicted review sentiments. We computed the precision and recall for predicting the *Negative* final sentiment for all the reviews in the test sets for each of the categories. A review is given a *Negative* sentiment if its positive level is lesser than the negative threshold, $PosLev(u_t, r_t, C_i) < Threshold_{Negative}(C_i)$ and *not negative* otherwise. We also plot the Accuracy and F1 measure.

5.2 Predicting Positive Reviews

Analysis of the actual distribution of the reviews, in fig 4.1, shows us that most positive reviews have their positive level, $PosLev(u_t, r_t, C_i)$ close to 1.0. The predicted distribution is similar but is more uniformly distributed than the actual distribution. This can be explained by the fact that users write short reviews with either positive or negative sentences or sometimes almost equal number of positive and negative sentences (as shown by the peaks at 0, 0.5 and 1.0). In contrast, the predicted values are weighted averages calculated using the iIB cluster membership probabilities, as detailed in equations 3.1 and 3.2.

We, therefore, vary the threshold, $Threshold_{Positive}(C_i)$, from 0.0 to 1.0 for both the actual and predicted values of $PosLev(u_t, r_t, C_i)$, to study the change in Positive Precision and Recall.

5.2.1 Precision

The distribution histograms, in fig 4.1, show that most positive reviews in all categories tend to have their positive level, $PosLev(u_t, r_t, C_i)$, equal to or near 1.0. This is true for both the predicted and actual distributions for all categories. Even *Anecdotes* show a local maximum at 1.0 even though most of them have a positive level, $PosLev(u_t, r_t, C_i)$, near 0.0.

We conducted an in-depth analysis of the positive precisions for each of the six categories as shown in fig 5.1. The goal of this analysis was to find the variation in precision as the $Threshold_{Positive}(C_i)$, for both actual and predicted reviews, is varied from 0.0 to 1.0.

However, for both actual and predicted $Threshold_{Positive}(C_i) = 1.0$, the precision is 0 since no reviews can be predicted as positive due to the fact that all reviews have a $PosLev(u_t, r_t, C_i) \leq 1.0$. Therefore, the graphs in fig 5.1 are drawn up to a $Threshold_{Positive}(C_i) = 0.9$ for actual reviews and $Threshold_{Positive}(C_i) = 0.99$ for predicted reviews.

Food

As seen in figure 5.1(a),the positive precision for Food varies from a maximum of 0.98, at an actual threshold of 0.9 and a predicted threshold of 0.99, to a minimum of 0.56 at an actual threshold of 0.0 and a predicted threshold of 0.99. The positive precision for food is 0.943, which is $\approx 5\%$ below the maximum, for an actual threshold of 0.8 and a predicted threshold of 0.6.

Price

The positive precision for Food varies from a maximum of 0.843, at an actual threshold of 0.9 and a predicted threshold of 0.7, to a minimum of 0.542 at an actual threshold of 0.0 and a predicted threshold of 0.0, as seen in fig 5.1(b). The positive precision for price is 0.784, which is $\approx 5\%$ below the maximum, for an actual threshold of 0.7 and a predicted threshold of 0.6.

Staff

As seen in figure 5.1(c), the positive precision for Staff varies from a maximum of 0.817, at an actual threshold of 0.8 and a predicted threshold of 0.8, to a minimum of 0.518 at an actual threshold of 0.1

and a predicted threshold of 0.1. The positive precision for Staff is 0.7785, which is $\approx 5\%$ below the maximum, for an actual threshold of 0.8 and a predicted threshold of 0.6.

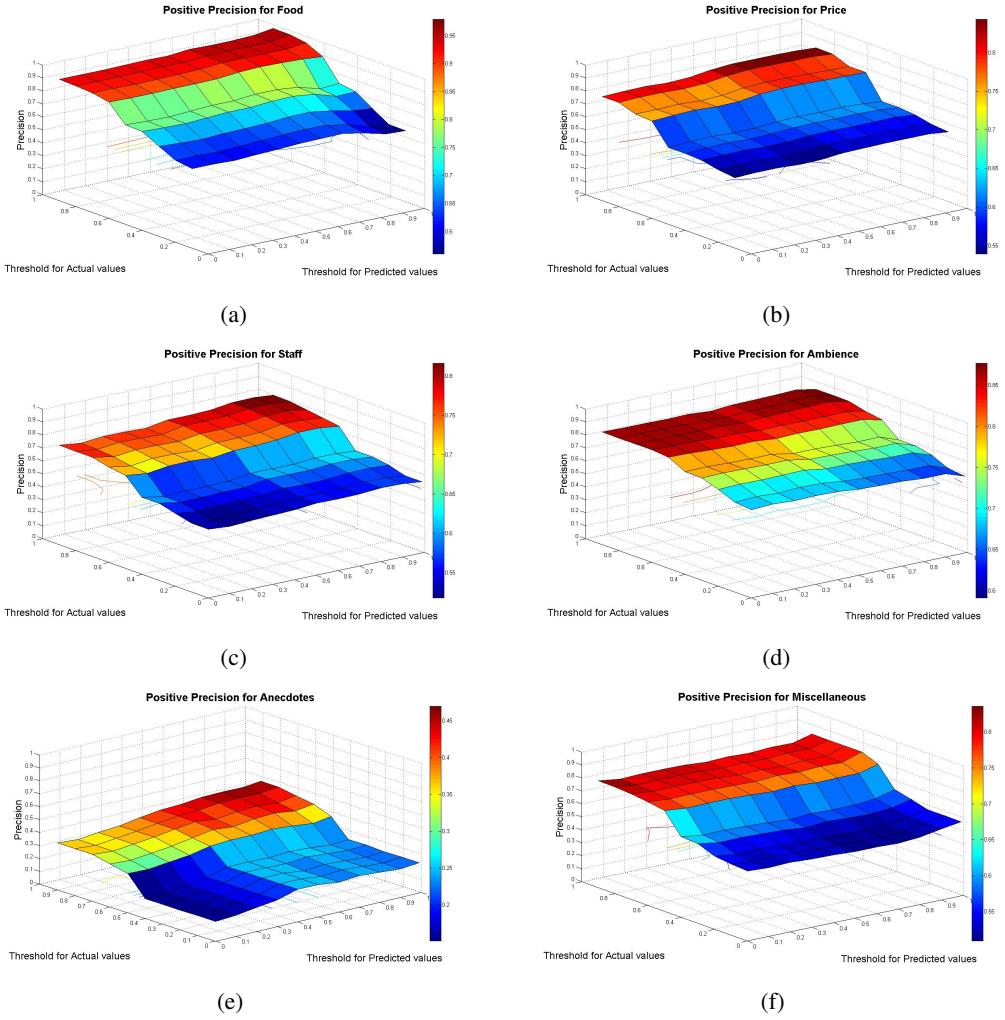


Figure 5.1: Positive Precision

Ambience

The positive precision for Ambience varies from a maximum of 0.877, at an actual threshold of 0.9 and a predicted threshold of 0.8, to a minimum of 0.592 at an actual threshold of 0.0 and a predicted threshold of 0.0, as seen in fig 5.1(d). The positive precision for Ambience is 0.845, which is $\approx 5\%$ below the maximum, for an actual threshold of 0.7 and a predicted threshold of 0.6.

Anecdotes

As seen in figure 5.1(e), the positive precision for Anecdotes varies from a maximum of 0.47, at an actual threshold of 0.9 and a predicted threshold of 0.9, to a minimum of 0.152 at an actual threshold of 0.0 and a predicted threshold of 0.0. The positive precision for Anecdotes is 0.453, which is $\approx 5\%$ below the maximum, for an actual threshold of 0.8 and a predicted threshold of 0.6.

Overall, the precision for predicting the positive sentiment for Anecdotes is low. This is due to the fact that most anecdotal sentences have an actual neutral sentiment. As a result the predicted review also contains a high composition of neutral sentences in the anecdotes category. This causes the anecdotal reviews to be clustered mostly around $PosLev(u_t, r_t, C_i) = 0$ as seen in the distribution histogram for anecdotes in fig 4.1(e).

Miscellaneous

As seen in figure 5.1(f), the positive precision for Miscellaneous varies from a maximum of 0.836, at an actual threshold of 0.9 and a predicted threshold of 0.9, to a minimum of 0.54 at an actual threshold of 0.0 and a predicted threshold of 0.0. The positive precision for Miscellaneous is 0.792, which is $\approx 5\%$ below the maximum, for an actual threshold of 0.8 and a predicted threshold of 0.7.

The positive precision for all categories shows a maximum at actual and predicted values close to 1.0 and almost the minimum at actual and predicted values equal to 0.0. This positive precision trend serves to strengthen our belief that we can have similar high thresholds for predicting the positive sentiment since most positive reviews have $PosLev(u_t, r_t, C_i) \approx 1.0$. Thus, the precision trend shows that most of our predictions for the positive sentiment are correct.

The Precisions, as shown in fig 5.1, tend to peak, i.e. varies 5% from the maximum, between 0.7 to 1.0 for actual $Threshold_{Positive}(C_i)$ and between 0.6 to 1.0 for predicted $Threshold_{Positive}(C_i)$. This phenomenon occurs due to the concentration of positive reviews, in both the actual and predicted scenarios, near 1.0. A high threshold of 0.7 on the actual reviews restricts the returned reviews to those which are polar towards positive. A threshold of 0.6 on the predicted values allows us to filter out all reviews except those which have a predicted value greater than 0.6. Out of these reviews which are predicted positive, most are positive and quite polar as evident from the higher precision, as seen in fig 5.1, as the thresholds increase towards 1.0.

Higher threshold values, thus, increase our precision and allow us to correctly recognize positive reviews.

5.2.2 Recall

As stated earlier, a higher threshold implies greater selectivity to correctly identify positive reviews. But the main drawback of high thresholds is that fewer reviews are rated as positive. Recall measures how many positive reviews are correctly predicted as opposed to the total number of positive reviews as shown in equation 5.2. Thus, Recall reduces as fewer reviews are predicted as positive towards higher values of the predicted threshold.

The Recall, in fig 5.2, reduces from 1.0 to 0.0 as the predicted threshold, $Threshold_{Positive}(C_i)$, is increased from 0.0 to 1.0. This happens because the number of predicted reviews classified as positive reduces as we increase the threshold from 0.0 to 1.0. However, as the actual threshold is increased, the recall is not affected greatly. This phenomenon occurs due to the fact that actual reviews are quite polar towards three distinct positive levels, $PosLev(u_t, r_t, C_i)$, of 1.0, 0.5 or 0.0.

We computed the recall for predicted threshold, $Threshold_{Positive}(C_i)$, which was varied from 0.0 to 1.0. However, the actual threshold was varied from 0.0 to 0.9. At actual $Threshold_{Positive}(C_i) = 1.0$, there are no actually positive reviews which leave us with a meaningless recall. The recalls for each of the six categories are plotted in fig 5.2.

Thus, for a recall to be higher, the threshold must be lower, such that more reviews are correctly predicted as positive. Since precision requires a higher threshold to be high and recall decreases with a higher threshold, a trade-off needs to be made between the precision and recall to find the optimum threshold.

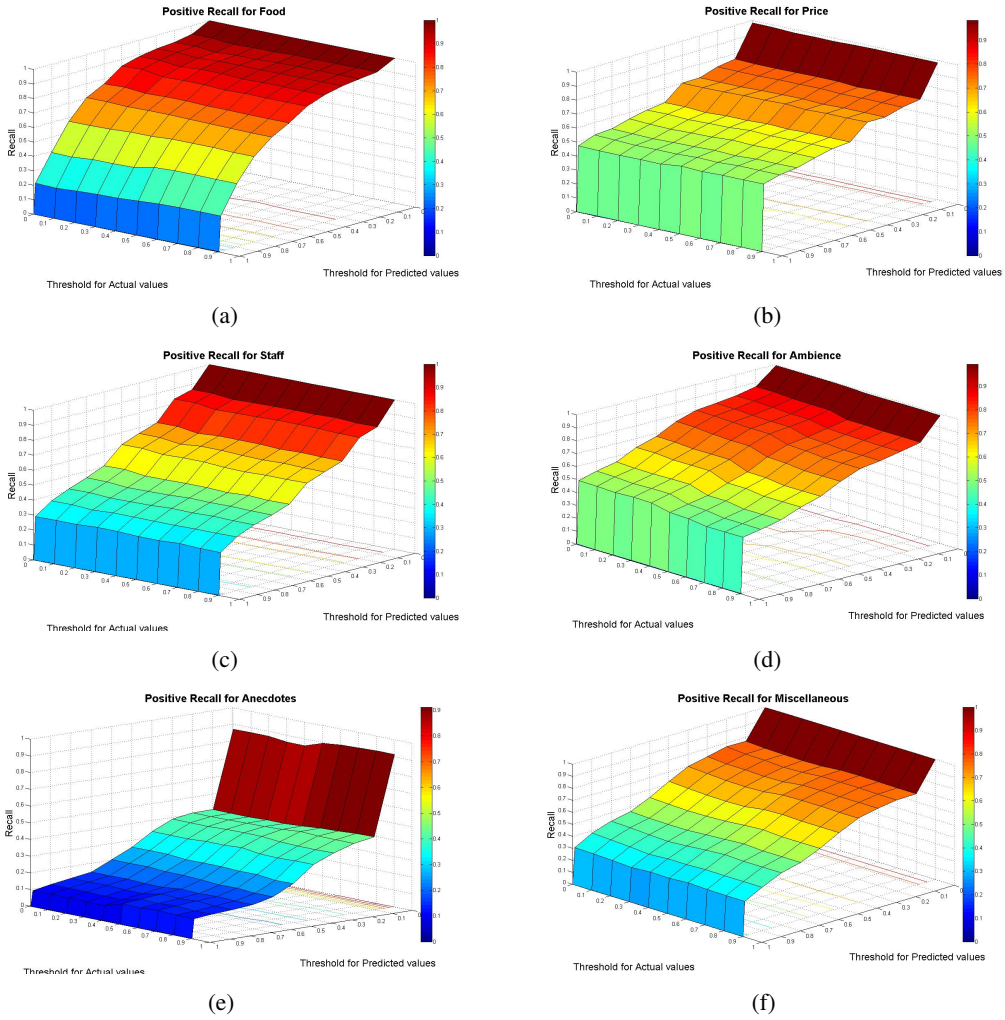


Figure 5.2: Positive Recall

The recall for *Anecdotes* is not 1.0 for a threshold of 0.0. This behavior occurs due to the fact that there is very limited evidence for *Positive* sentiment in *Anecdotes* since most sentences have a *Neutral* sentiment attached to them. This is the same reason why the precision for predicting *Positive* sentiment for *Anecdotes* is much lower than other categories. Also, the recall for anecdotes falls sharply as the threshold increases from 0.0 to 0.1. This happens because most reviews have $PosLev(u_t, r_t, Anecdotes) = 0.0$.

5.2.3 Accuracy

Accuracy measures the correctness of prediction. That is, accuracy is the proportion of correctly predicted reviews to the total number of reviews in the data set.

The trend in accuracy for positive prediction is shown in fig 5.3. Accuracy increases as we increase the actual threshold for positive sentiment prediction. This happens because a high actual threshold reduces the number of reviews which have to be predicted as positive making it easier for the predictor.It reduces as we increase the predicted threshold. Increasing the predicted threshold, restricts the prediction of reviews as positive since the predictions are more uniformly distributed than the actual reviews which have a polar distribution as seen in fig 4.1.

Food

The positive accuracy for Food, as seen in fig 5.3(a), shows a maximum of 0.935 for an actual threshold of 0.9 and a predicted threshold of 0.0. At an actual threshold of 0.0, the accuracy shows a maximum value of 0.617 for a predicted threshold of 0.0 and a minimum of 0.414 for a predicted threshold of 0.99. Whereas, at an actual threshold of 0.9, the accuracy shows a maximum value of 0.935 for a predicted threshold of 0.0 and a minimum of 0.288 for a predicted threshold of 0.99.

Price

The positive accuracy for Price, as seen in fig 5.3(b), shows a maximum of 0.791 for an actual threshold of 0.9 and a predicted threshold of 0.0. At an actual threshold of 0.0, the accuracy shows a maximum value of 0.54 for a predicted threshold of 0.0 and a minimum of 0.5 for a predicted threshold of 0.99. Whereas, at an actual threshold of 0.9, the accuracy shows a maximum value of 0.791 for a predicted threshold of 0.0 and a minimum of 0.5 for a predicted threshold of 0.99.

Staff

The positive accuracy for Staff, as seen in fig 5.3(c), shows a maximum of 0.8 for an actual threshold of 0.9 and a predicted threshold of 0.0. At an actual threshold of 0.0, the accuracy shows a maximum value of 0.55 for a predicted threshold of 0.0 and a minimum of 0.47 for a predicted threshold of 0.99. Whereas, at an actual threshold of 0.9, the accuracy shows a maximum value of 0.8 for a predicted threshold of 0.0 and a minimum of 0.35 for a predicted threshold of 0.99.

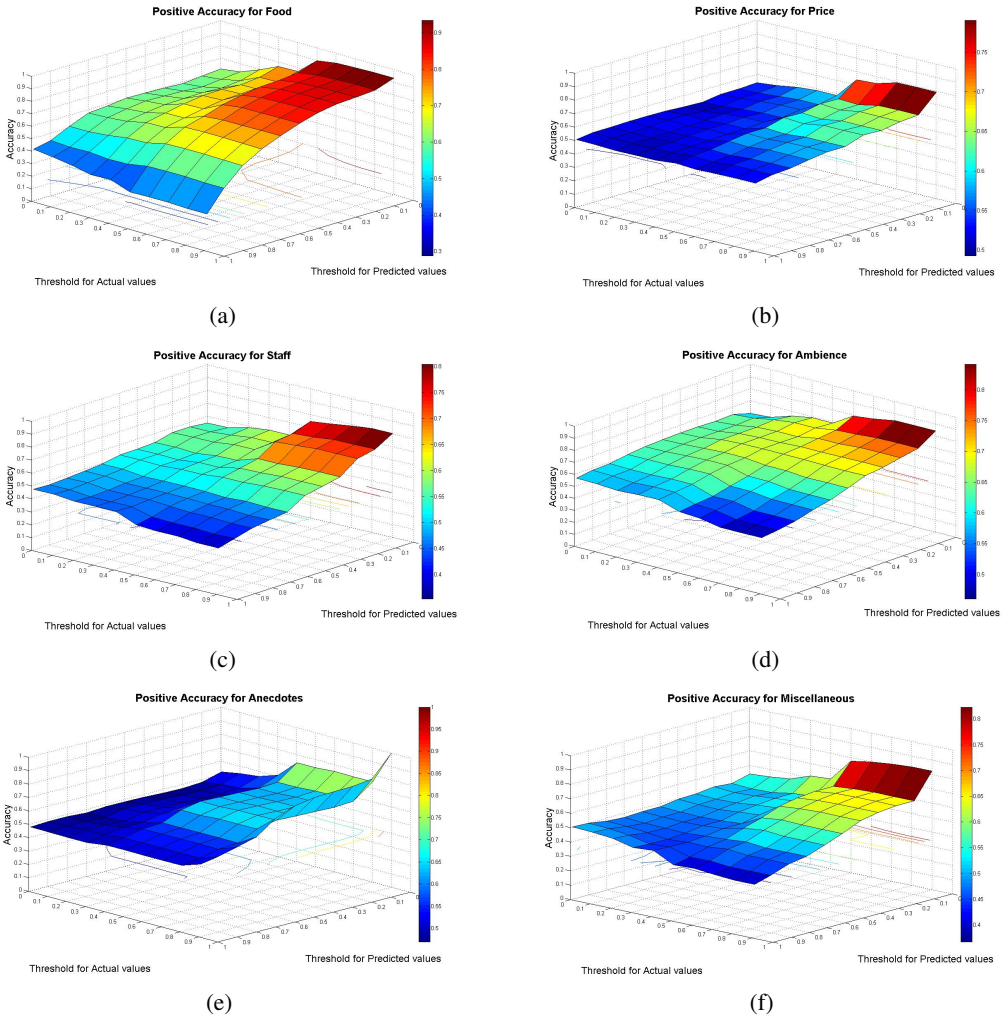


Figure 5.3: Positive Accuracy

Ambience

The positive accuracy for Ambience, as seen in fig 5.3(d), shows a maximum of 0.84 for an actual threshold of 0.9 and a predicted threshold of 0.0. At an actual threshold of 0.0, the accuracy shows a maximum value of 0.59 for a predicted threshold of 0.0 and a minimum of 0.56 for a predicted threshold of 0.99. Whereas, at an actual threshold of 0.9, the accuracy shows a maximum value of 0.84 for a predicted threshold of 0.0 and a minimum of 0.46 for a predicted threshold of 0.99.

Anecdotes

The positive accuracy for Anecdotes, as seen in fig 5.3(e), shows a maximum of 0.65 for an actual threshold of 0.9 and a predicted threshold of 0.0. At an actual threshold of 0.0, the accuracy shows a maximum value of 0.47 for a predicted threshold of 0.0 and a minimum of 0.46 for a predicted threshold of 0.99. Whereas, at an actual threshold of 0.9, the accuracy shows a maximum value of 0.65 for a predicted threshold of 0.0 and a minimum of 0.26 for a predicted threshold of 0.99.

Miscellaneous

The positive accuracy for Miscellaneous, as seen in fig 5.3(f), shows a maximum of 0.82 for an actual threshold of 0.9 and a predicted threshold of 0.0. At an actual threshold of 0.0, the accuracy shows a maximum value of 0.54 for a predicted threshold of 0.0 and a minimum of 0.5 for a predicted threshold of 0.99. Whereas, at an actual threshold of 0.9, the accuracy shows a maximum value of 0.82 for a predicted threshold of 0.0 and a minimum of 0.37 for a predicted threshold of 0.99.

These trends in positive accuracy for the six categories inform us that most polar positive reviews are being predicted with $PosLev(u_t, r_t, C_i)$ varying from 0.6 to 1.0 because of which we have a high accuracy level at 0.6 predicted threshold which reduces as the predicted threshold increases to 0.99. The Accuracy for Price, Staff, Ambience and Miscellaneous shows a sudden drop from 0.0 to 0.1 since there are a few reviews which are predicted as polar negative instead of positive due to lack of sufficient positive evidence for those restaurants in the training data.

5.3 Predicting Negative Reviews

The actual distribution, as shown in fig 4.1, shows us most negative reviews have their positive level, $PosLev(u_t, r_t, C_i)$, close to 0.0. The predicted distribution is almost similar to the actual distribution, in all categories except *Food*, since users mostly write very few sentences in these categories and if reviews are negative, they mostly contain only a few negative sentences. For the category *Food*, users tend to write reviews with sentences spanning over all four sentiments and so negative reviews do not have only negative sentences. Therefore, there is a need to have a separate $Threshold_{Negative}(C_i)$, in order to capture the negative reviews which may get predicted with a

low value of $PosLev(u_t, r_t, C_i)$ instead of a pure zero.

We varied the threshold, $Threshold_{Negative}(C_i)$, from 0.0 to 1.0 for both the actual and predicted values of $PosLev(u_t, r_t, C_i)$, to study the change in Negative Precision and Recall for each of the six categories.

5.3.1 Precision

Revisiting the distribution histograms in fig 4.1, we observe that most actually negative reviews are concentrated near the value 0.0 for the positive level, $PosLev(u_t, r_t, C_i)$. This tendency is also observed in the predicted reviews. So, most reviews are concentrated in the region near $PosLev(u_t, r_t, C_i) = 0.0$.

We conducted an in-depth analysis of the negative precisions for each of the six categories as shown in fig 5.4. The goal of this analysis was to find the variation in precision as the $Threshold_{Negative}(C_i)$, for both actual and predicted reviews, is varied from 0.0 to 1.0.

However, for both actual and predicted $Threshold_{Negative}(C_i) = 0.0$, the precision is 0 since no reviews can be predicted as negative due to the fact that all reviews have a $PosLev(u_t, r_t, C_i) \geq 0.0$. Therefore, the graphs in fig 5.4 are drawn from a $Threshold_{Negative}(C_i) = 0.1$ to $Threshold_{Negative}(C_i) = 1.0$ for actual and predicted reviews.

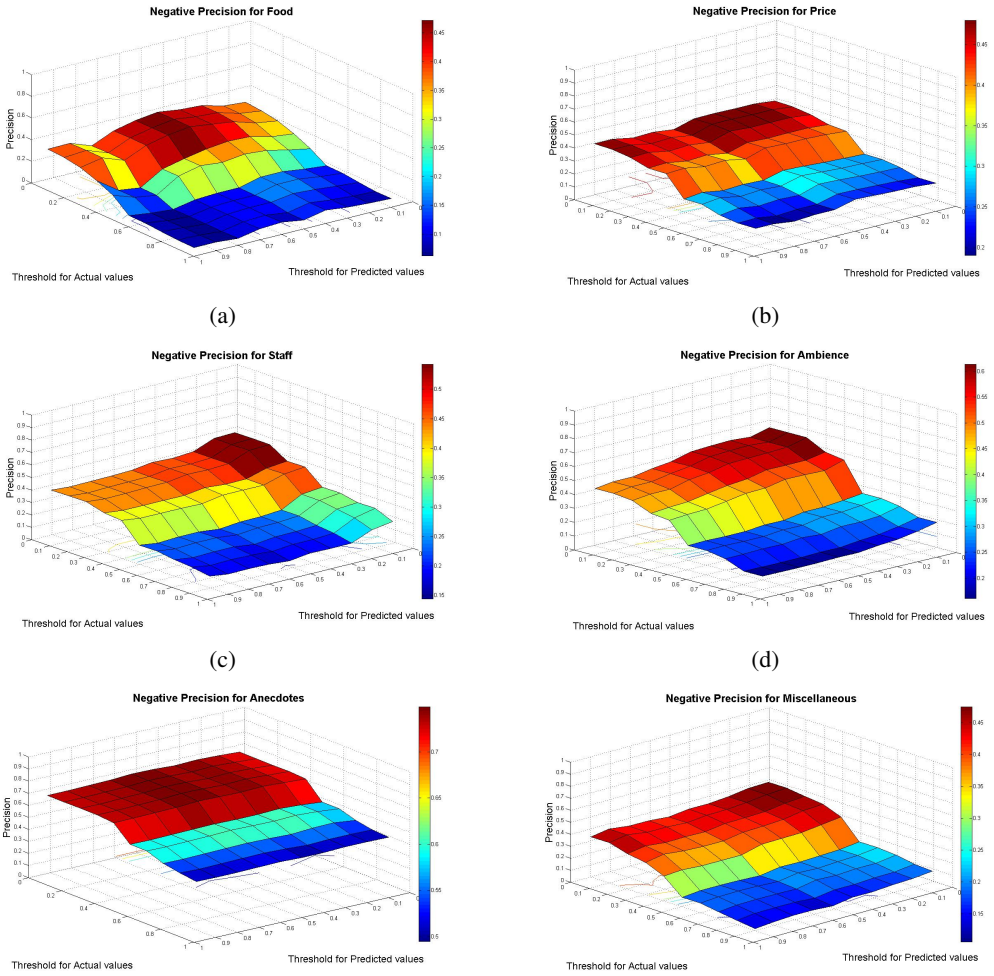


Figure 5.4: Negative Precision

Food

As seen in figure 5.4(a), the negative precision for Food varies from a maximum of 0.446, at an actual threshold of 0.1 and a predicted threshold of 0.5, to a minimum of 0.077 at an actual threshold of 1.0 and a predicted threshold of 1.0. The negative precision for Food is 0.42, which is $\approx 5\%$ below the maximum, for an actual threshold of 0.3 and a predicted threshold of 0.5.

Price

As seen in figure 5.4(b), the negative precision for Price varies from a maximum of 0.48, at an actual threshold of 0.1 and a predicted threshold of 0.3, to a minimum of 0.19 at an actual threshold of 1.0

and a predicted threshold of 0.8. The negative precision for Price is 0.468, which is $\approx 5\%$ below the maximum, for an actual threshold of 0.3 and a predicted threshold of 0.4.

Staff

As seen in figure 5.4(c), the negative precision for Staff varies from a maximum of 0.543, at an actual threshold of 0.1 and a predicted threshold of 0.1, to a minimum of 0.144 at an actual threshold of 1.0 and a predicted threshold of 0.6. The negative precision for Staff is 0.53, which is $\approx 5\%$ below the maximum, for an actual threshold of 0.3 and a predicted threshold of 0.2.

Ambience

The negative precision for Ambience varies from a maximum of 0.614, at an actual threshold of 0.1 and a predicted threshold of 0.1, to a minimum of 0.161 at an actual threshold of 1.0 and a predicted threshold of 0.9, as seen in figure 5.4(d). The negative precision for Ambience is 0.58, which is $\approx 5\%$ below the maximum, for an actual threshold of 0.3 and a predicted threshold of 0.2.

Anecdotes

As seen in figure 5.4(e), the negative precision for Anecdotes varies from a maximum of 0.745, at an actual threshold of 0.1 and a predicted threshold of 0.5, to a minimum of 0.497 at an actual threshold of 1.0 and a predicted threshold of 0.6. The negative precision for Anecdotes is 0.72, which is $\approx 5\%$ below the maximum, for an actual threshold of 0.5 and a predicted threshold of 0.5.

Miscellaneous

The negative precision for Miscellaneous varies from a maximum of 0.474, at an actual threshold of 0.1 and a predicted threshold of 0.1, to a minimum of 0.106 at an actual threshold of 1.0 and a predicted threshold of 1.0, as seen in fig 5.4(f). The negative precision for Miscellaneous is 0.467, which is $\approx 5\%$ below the maximum, for an actual threshold of 0.2 and a predicted threshold of 0.2.

The negative precision shows a maximum at actual and predicted values close to 0.0 and almost the minimum at actual and predicted values equal to 1.0. This negative precision trend serves to strengthen our belief that we can have similar low thresholds for predicting the negative sentiment

since most negative reviews have $PosLev(u_t, r_t, C_i) \approx 0.0$. Thus, the trend in precision shows that most of our predictions for the negative sentiment are correct.

The precisions, as shown in fig 5.4, tend to peak, i.e. varies 5% from the maximum, between 0.1 to 0.3 for actual $Threshold_{Negative}(C_i)$ and between 0.1 to 0.4 for predicted $Threshold_{Negative}(C_i)$. Reducing $Threshold_{Negative}(C_i)$ gives us only those reviews which are polar towards the negative having a $PosLev(u_t, r_t, C_i) \approx 0.0$. Precision increases, as we reduce $Threshold_{Negative}(C_i)$, since fewer reviews are wrongly predicted as negative. Hence, the precision for predicting negative reviews increases as $Threshold_{Negative}(C_i)$ reduces towards 0. Thus, we can get a higher precision for predicting negative reviews if we reduce $Threshold_{Negative}(C_i)$.

However, *Food* reviews have a different behavior for precision. This is, again, due to the fact that they tend to have substantially high *positive, neutral* and *conflict* components to their predicted values. Other categories tend to have a more crisp negative prediction due to very low *positive, neutral* and *conflict* components in their predicted values.

One more fact about the precision that needs to be discussed is that the precision for predicting negative reviews is much lower than the precision for predicting positive reviews. Our prediction system does not predict polar negative due to the imbalance in the amount of evidence available for predicting positive and negative values for the reviews. Most of the sentences in the training data are positive which increases the precision for predicting positives and prevents our algorithm from achieving the same high precision for predicting negative reviews.

5.3.2 Recall

As we reduce $Threshold_{Negative}(C_i)$, only the polar negative reviews get selected which are fewer in number. Since recall is the measure of number of reviews correctly predicted as negative in comparison to the total number of actually negative reviews, it reduces as $Threshold_{Negative}(C_i)$ reduces, since there are lesser reviews correctly predicted as negative.

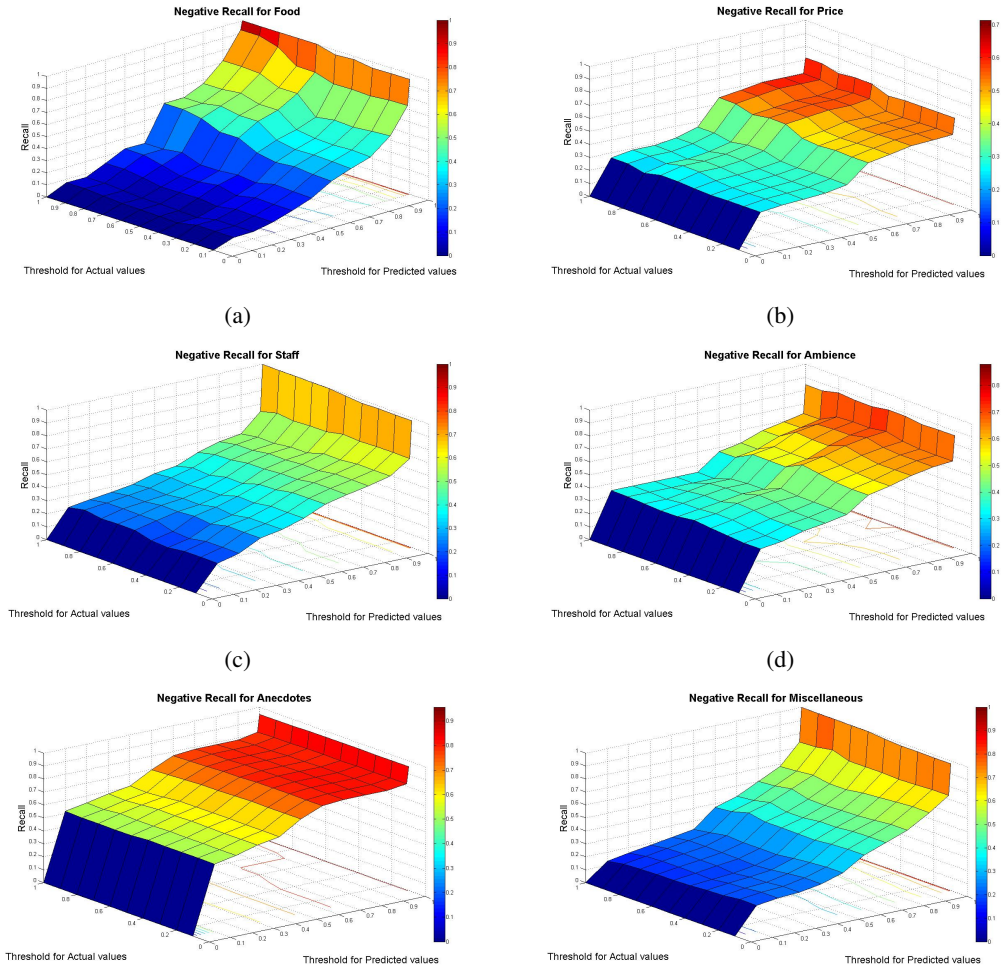


Figure 5.5: Negative Recall

The Recall, in fig 5.5, decreases from 1.0 to 0.0 as the predicted threshold, $Threshold_{Negative}(C_i)$, is decreased from 1.0 to 0.0. However, there is no major change in the recall as the actual threshold, $Threshold_{Negative}(C_i)$, varies. This is due to the fact that most actual reviews have a distinct values of 0.0, 0.5 and 1.0 for the $PosLev(u_t, r_t, C_i)$.

We computed the recall for predicted threshold, $Threshold_{Negative}(C_i)$, varied from 0.0 to 1.0. However, the actual threshold was varied from 0.1 to 1.0. At actual $Threshold_{Negative}(C_i) = 0.0$, there are no actually negative reviews which leaves us with a meaningless recall. The recalls for each of the six categories are plotted in fig 5.5.

Since recall increases and precision decreases as we increase $Threshold_{Negative}(C_i)$, we need to find a tradeoff between them to get the best $Threshold_{Negative}(C_i)$.

5.3.3 Accuracy

The trend in accuracy for negative prediction is shown in fig 5.6. Accuracy increases as we decrease the actual threshold for positive sentiment prediction. This happens because a low actual threshold reduces the number of reviews which have to be predicted as negative making it easier for the predictor.It reduces as we decrease the predicted threshold. Decreasing the predicted threshold, restricts the prediction of reviews as positive since the predictions are more uniformly distributed than the actual reviews which have a polar distribution as seen in fig 4.1.

Food

The negative accuracy for Food, as seen in fig 5.6(a), shows a maximum of 0.88 for an actual threshold of 0.1 and a predicted threshold of 1.0. At an actual threshold of 1.0, the accuracy shows a maximum value of 0.59 for a predicted threshold of 1.0 and a minimum of 0.414 for a predicted threshold of 0.1. Whereas, at an actual threshold of 0.1, the accuracy shows a maximum value of 0.88 for a predicted threshold of 1.0 and a minimum of 0.12 for a predicted threshold of 0.1.

Price

The negative accuracy for Price, as seen in fig 5.6(b), shows a maximum of 0.66 for an actual threshold of 0.1 and a predicted threshold of 1.0. At an actual threshold of 1.0, the accuracy shows a maximum value of 0.53 for a predicted threshold of 1.0 and a minimum of 0.51 for a predicted threshold of 0.1. Whereas, at an actual threshold of 0.1, the accuracy shows a maximum value of 0.66 for a predicted threshold of 1.0 and a minimum of 0.45 for a predicted threshold of 0.1.

Staff

The negative accuracy for Staff, as seen in fig 5.6(c), shows a maximum of 0.73 for an actual threshold of 0.1 and a predicted threshold of 1.0. At an actual threshold of 1.0, the accuracy shows a maximum value of 0.56 for a predicted threshold of 1.0 and a minimum of 0.46 for a predicted threshold of 0.1. Whereas, at an actual threshold of 0.1, the accuracy shows a maximum value of 0.73 for a predicted threshold of 1.0 and a minimum of 0.22 for a predicted threshold of 0.1.

Ambience

The negative accuracy for Ambience, as seen in fig 5.6(d), shows a maximum of 0.75 for an actual threshold of 0.1 and a predicted threshold of 1.0. At an actual threshold of 1.0, the accuracy shows a maximum value of 0.64 for a predicted threshold of 1.0 and a minimum of 0.45 for a predicted threshold of 0.1. Whereas, at an actual threshold of 0.1, the accuracy shows a maximum value of 0.75 for a predicted threshold of 1.0 and a minimum of 0.26 for a predicted threshold of 0.1.

Anecdotes

The negative accuracy for Anecdotes, as seen in fig 5.6(e), shows a maximum of 0.64 for an actual threshold of 0.1 and a predicted threshold of 1.0. At an actual threshold of 1.0, the accuracy shows a maximum value of 0.48 for a predicted threshold of 1.0 and a minimum of 0.47 for a predicted threshold of 0.1. Whereas, at an actual threshold of 0.1, the accuracy shows a maximum value of 0.64 for a predicted threshold of 1.0 and a minimum of 0.25 for a predicted threshold of 0.1.

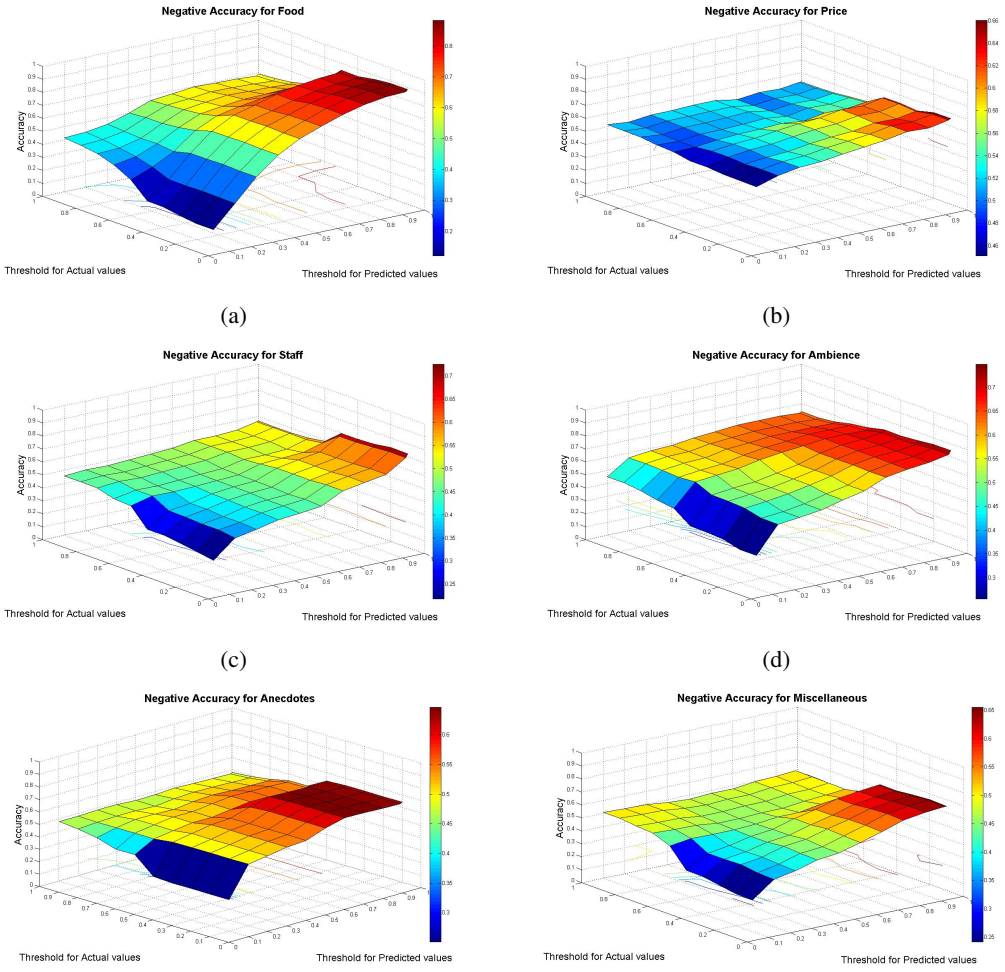


Figure 5.6: Negative Accuracy

Miscellaneous

The negative accuracy for Miscellaneous, as seen in fig 5.6(f), shows a maximum of 0.65 for an actual threshold of 0.1 and a predicted threshold of 1.0. At an actual threshold of 1.0, the accuracy shows a maximum value of 0.51 for a predicted threshold of 1.0 and a minimum of 0.48 for a predicted threshold of 0.4. Whereas, at an actual threshold of 0.1, the accuracy shows a maximum value of 0.65 for a predicted threshold of 1.0 and a minimum of 0.24 for a predicted threshold of 0.1.

These trends in negative accuracy for the six categories inform us that most polar negative reviews are being predicted with $PosLev(u_t, r_t, C_i)$ varying from 0.0 to 0.6 because of which

we have a high accuracy level at 0.6 predicted threshold which reduces as the predicted threshold decreases to 0.1. The Accuracy for Price, Staff, Ambience and Miscellaneous shows a sudden drop from 1.0 to 0.99 since there are a few reviews which are predicted as polar positive instead of negative due to lack of sufficient negative evidence for those restaurants in the training data.

5.4 Comparison with a Baseline Predictor

We compared the outcome of our method with a baseline method having a simple User-Average prediction mechanism instead of our iIB Clustering mechanism. We computed the positive and negative precision and recall for each category of sentences as described in sections 5.2 and 5.3.

The averaging predictor predicts the composition of the test reviews by averaging the composition of other reviews written by other users for the same restaurant. The main observations which can be made from this baseline approach are that the recalls are very similar to those shown in figures 5.2 and 5.5 for the iIB approach. The main difference is in the precision for predicting the positive and negative reviews.

5.4.1 Positive Precision for Averaging Predictor

The positive precision for user average based predictor are shown in fig 5.7. Comparing figures 5.7 and 5.1, we observe that there is a similarity in the trend of positive precision for both prediction methods.

We conducted an analysis of the positive precisions for each of the six categories as shown in fig 5.7 . The goal of this analysis was to find the variation in precision as the $Threshold_{Positive}(C_i)$, for both actual and predicted reviews, is varied from 0.0 to 1.0.

However, for both actual and predicted $Threshold_{Positive}(C_i) = 1.0$, the precision is 0 since no reviews can be predicted as positive due to the fact that all reviews have a $PosLev(u_t, r_t, C_i) \leq 1.0$. Therefore, the graphs in fig 5.7 are drawn up to a $Threshold_{Positive}(C_i) = 0.9$ for actual reviews and $Threshold_{Positive}(C_i) = 0.99$ for predicted reviews.

Food

The positive precision for Food varies from a maximum of 1.0, at an actual threshold of 0.9 and a predicted threshold of 0.99, to a minimum of 0.614 at an actual threshold of 0.0 and a predicted threshold of 0.0, as seen in fig 5.7(a).

Price

The positive precision for Price varies from a maximum of 0.9, at an actual threshold of 0.9 and a predicted threshold of 0.99, to a minimum of 0.542 at an actual threshold of 0.0 and a predicted threshold of 0.0, as seen in fig 5.7(b).

Staff

The positive precision for Staff varies from a maximum of 1.0, at an actual threshold of 0.9 and a predicted threshold of 0.99, to a minimum of 0.549 at an actual threshold of 0.0 and a predicted threshold of 0.0, as seen in fig 5.7(c).

Ambience

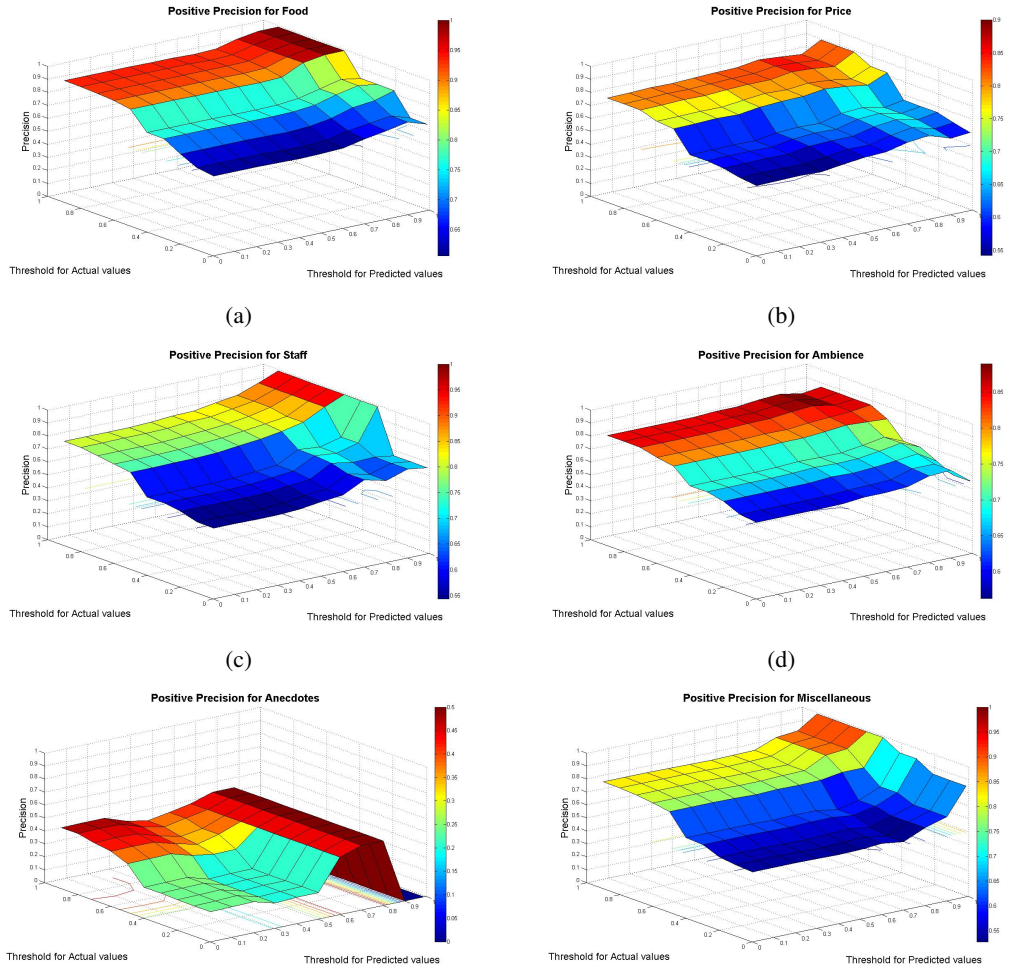
The positive precision for Ambience varies from a maximum of 0.875, at an actual threshold of 0.9 and a predicted threshold of 0.99, to a minimum of 0.592 at an actual threshold of 0.0 and a predicted threshold of 0.0, as seen in fig 5.7(d).

Anecdotes

The positive precision for Anecdotes varies from a maximum of 0.5, at a predicted threshold of 0.8, to a minimum of 0.235 at an actual threshold of 0.0 and a predicted threshold of 0.0, as seen in fig 5.7(e).

Miscellaneous

The positive precision for Miscellaneous varies from a maximum of 1.0, at an actual threshold of 0.9 and a predicted threshold of 0.99, to a minimum of 0.54 at an actual threshold of 0.0 and a predicted threshold of 0.0, as seen in fig 5.7(f).



(a) (b)

(c) (d)

(e) (f)

Figure 5.7: Positive Precision for Averaging Predictor

5.4.2 Negative Precision for Averaging Predictor

The negative precision for iIB peaks at a threshold value of 0.1 for both actual and predicted reviews (as observed in fig 5.4). However, in fig 5.8 for the averaging predictor, the peak is observed for a predicted threshold value close to 1.0 and an actual threshold value of 0.1.

We conducted an analysis of the negative precisions for each of the six categories as shown in fig 5.8. The goal of this analysis was to find the variation in precision as the $Threshold_{Negative}(C_i)$, for both actual and predicted reviews, is varied from 0.0 to 1.0.

However, for both actual and predicted $Threshold_{Negative}(C_i) = 0.0$, the precision is 0 since no reviews can be predicted as positive due to the fact that all reviews have a $PosLev(u_t, r_t, C_i) \geq$

0.0. Therefore, the graphs in fig 5.8 are drawn from a $Threshold_{Negative}(C_i) = 0.1$ to a $Threshold_{Negative}(C_i) = 1.0$ for actual and predicted reviews.

Food

The negative precision for Food varies from a maximum of 0.583, at an actual threshold of 0.1 and a predicted threshold of 0.7, to a minimum of 0.0 at an actual threshold of 1.0 and a predicted threshold of 0.6, as seen in fig 5.8(a).

Price

The negative precision for Price varies from a maximum of 0.592, at an actual threshold of 0.1 and a predicted threshold of 0.5, to a minimum of 0.2 at an actual threshold of 1.0 and a predicted threshold of 0.9, as seen in fig 5.8(b).

Staff

The negative precision for Staff varies from a maximum of 0.53, at an actual threshold of 0.1 and a predicted threshold of 0.7, to a minimum of 0.0 at an actual threshold of 1.0 and a predicted threshold of 0.1, as seen in fig 5.8(c).

Ambience

The negative precision for Ambience varies from a maximum of 0.571, at an actual threshold of 0.1 and a predicted threshold of 0.4, to a minimum of 0.0 at an actual threshold of 1.0 and a predicted threshold of 0.3, as seen in fig 5.8(d).

Anecdotes

The negative precision for Anecdotes varies from a maximum of 0.757, at an actual threshold of 0.1 and a predicted threshold of 0.6, to a minimum of 0.42 at an actual threshold of 1.0 and a predicted threshold of 0.3, as seen in fig 5.8(e).

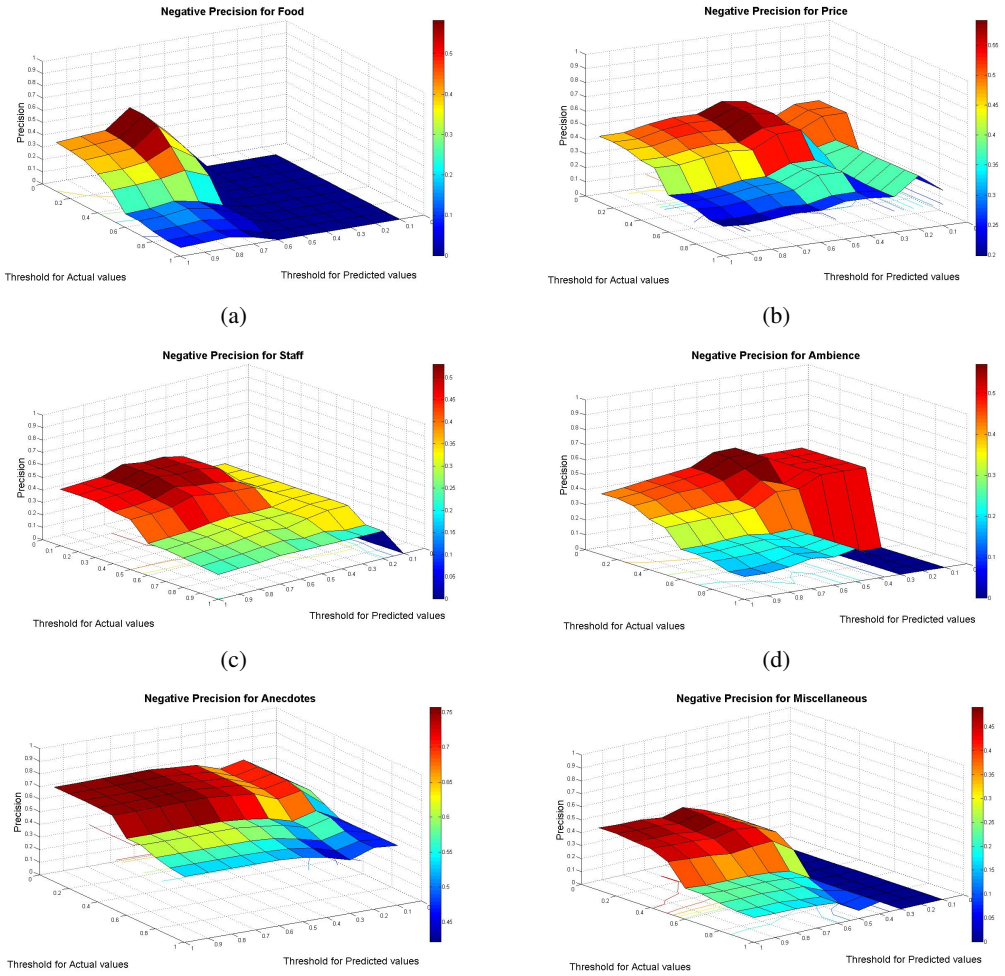


Figure 5.8: Negative Precision for Averaging Predictor

Miscellaneous

The negative precision for Miscellaneous varies from a maximum of 0.491, at an actual threshold of 0.1 and a predicted threshold of 0.6, to a minimum of 0.0 at an actual threshold of 1.0 and a predicted threshold of 0.3, as seen in fig 5.8(f).

5.4.3 Positive Accuracy for Averaging Predictor

The trend in accuracy for positive prediction is shown in fig 5.9. Accuracy increases as we increase the actual threshold for positive sentiment prediction. This happens because a high actual threshold

reduces the number of reviews which have to be predicted as positive making it easier for the predictor.It reduces as we increase increase the predicted threshold. Increasing the predicted threshold, restricts the prediction of reviews as positive.

Food

The positive accuracy for Food, as seen in fig 5.9(a), shows a maximum of 0.935 for an actual threshold of 0.9 and a predicted threshold of 0.0. At an actual threshold of 0.0, the accuracy shows a maximum value of 0.614 for a predicted threshold of 0.0 and a minimum of 0.03 for a predicted threshold of 0.99. Whereas, at an actual threshold of 0.9, the accuracy shows a maximum value of 0.935 for a predicted threshold of 0.0 and a minimum of 0.05 for a predicted threshold of 0.99.

Price

The positive accuracy for Price, as seen in fig 5.9(b), shows a maximum of 0.79 for an actual threshold of 0.9 and a predicted threshold of 0.0. At an actual threshold of 0.0, the accuracy shows a maximum value of 0.53 for a predicted threshold of 0.0 and a minimum of 0.08 for a predicted threshold of 0.99. Whereas, at an actual threshold of 0.9, the accuracy shows a maximum value of 0.79 for a predicted threshold of 0.0 and a minimum of 0.13 for a predicted threshold of 0.99.

Staff

The positive accuracy for Staff, as seen in fig 5.9(c), shows a maximum of 0.81 for an actual threshold of 0.9 and a predicted threshold of 0.0. At an actual threshold of 0.0, the accuracy shows a maximum value of 0.54 for a predicted threshold of 0.0 and a minimum of 0.009 for a predicted threshold of 0.99. Whereas, at an actual threshold of 0.9, the accuracy shows a maximum value of 0.81 for a predicted threshold of 0.0 and a minimum of 0.014 for a predicted threshold of 0.99.

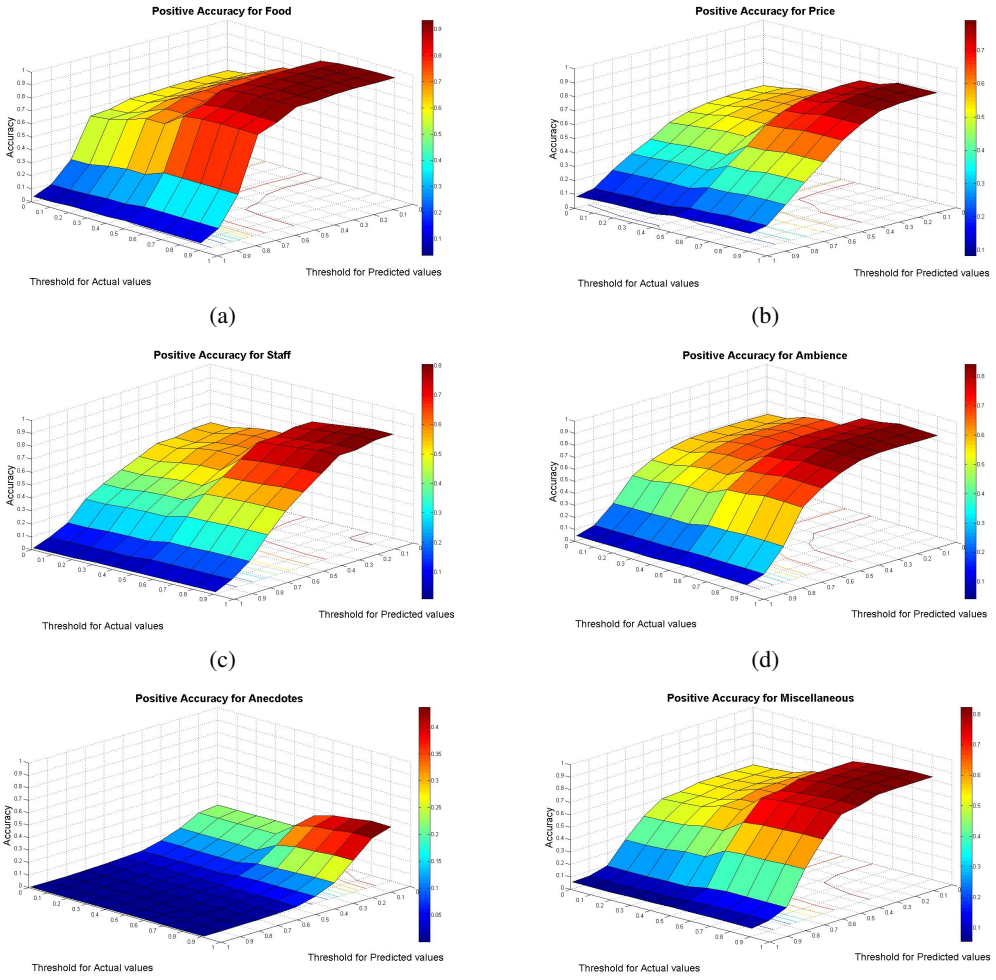


Figure 5.9: Positive Accuracy for Averaging Predictor

Ambience

The positive accuracy for Ambience, as seen in fig 5.9(d), shows a maximum of 0.842 for an actual threshold of 0.9 and a predicted threshold of 0.0. At an actual threshold of 0.0, the accuracy shows a maximum value of 0.586 for a predicted threshold of 0.0 and a minimum of 0.0 for a predicted threshold of 0.99. Whereas, at an actual threshold of 0.9, the accuracy shows a maximum value of 0.842 for a predicted threshold of 0.0 and a minimum of 0.0 for a predicted threshold of 0.99.

Anecdotes

The positive accuracy for Anecdotes, as seen in fig 5.9(e), shows a maximum of 0.44 for an actual threshold of 0.9 and a predicted threshold of 0.0. At an actual threshold of 0.0, the accuracy shows a maximum value of 0.22 for a predicted threshold of 0.0 and a minimum of 0.0 for a predicted threshold of 0.99. Whereas, at an actual threshold of 0.9, the accuracy shows a maximum value of 0.44 for a predicted threshold of 0.0 and a minimum of 0.0 for a predicted threshold of 0.99.

Miscellaneous

The positive accuracy for Miscellaneous, as seen in fig 5.9(f), shows a maximum of 0.82 for an actual threshold of 0.9 and a predicted threshold of 0.0. At an actual threshold of 0.0, the accuracy shows a maximum value of 0.53 for a predicted threshold of 0.0 and a minimum of 0.05 for a predicted threshold of 0.99. Whereas, at an actual threshold of 0.9, the accuracy shows a maximum value of 0.82 for a predicted threshold of 0.0 and a minimum of 0.06 for a predicted threshold of 0.99.

5.4.4 Negative Accuracy for Averaging Predictor

The trend in accuracy for negative prediction is shown in fig 5.10. Accuracy increases as we decrease the actual threshold for positive sentiment prediction. This happens because a low actual threshold reduces the number of reviews which have to be predicted as negative making it easier for the predictor.It reduces as we decrease the predicted threshold. Decreasing the predicted threshold, restricts the prediction of reviews as positive since the predictions are more uniformly distributed than the actual reviews which have a polar distribution as seen in fig 4.1.

Food

The negative accuracy for Food, as seen in fig 5.10(a), shows a maximum of 0.37 for an actual threshold of 0.1 and a predicted threshold of 1.0. At an actual threshold of 1.0, the accuracy shows a maximum value of 0.07 for a predicted threshold of 1.0 and a minimum of 0.0 for a predicted threshold of 0.1. Whereas, at an actual threshold of 0.1, the accuracy shows a maximum value of 0.37 for a predicted threshold of 1.0 and a minimum of 0.0 for a predicted threshold of 0.1.

Price

The negative accuracy for Price, as seen in fig 5.10(b), shows a maximum of 0.4 for an actual threshold of 0.1 and a predicted threshold of 1.0. At an actual threshold of 1.0, the accuracy shows a maximum value of 0.18 for a predicted threshold of 1.0 and a minimum of 0.004 for a predicted threshold of 0.1. Whereas, at an actual threshold of 0.1, the accuracy shows a maximum value of 0.4 for a predicted threshold of 1.0 and a minimum of 0.01 for a predicted threshold of 0.1.

Staff

The negative accuracy for Staff, as seen in fig 5.10(c), shows a maximum of 0.45 for an actual threshold of 0.1 and a predicted threshold of 1.0. At an actual threshold of 1.0, the accuracy shows a maximum value of 0.2 for a predicted threshold of 1.0 and a minimum of 0.0 for a predicted threshold of 0.1. Whereas, at an actual threshold of 0.1, the accuracy shows a maximum value of 0.45 for a predicted threshold of 1.0 and a minimum of 0.0 for a predicted threshold of 0.1.

Ambience

The negative accuracy for Ambience, as seen in fig 5.10(d), shows a maximum of 0.38 for an actual threshold of 0.1 and a predicted threshold of 1.0. At an actual threshold of 1.0, the accuracy shows a maximum value of 0.14 for a predicted threshold of 1.0 and a minimum of 0.0 for a predicted threshold of 0.1. Whereas, at an actual threshold of 0.1, the accuracy shows a maximum value of 0.38 for a predicted threshold of 1.0 and a minimum of 0.004 for a predicted threshold of 0.1.

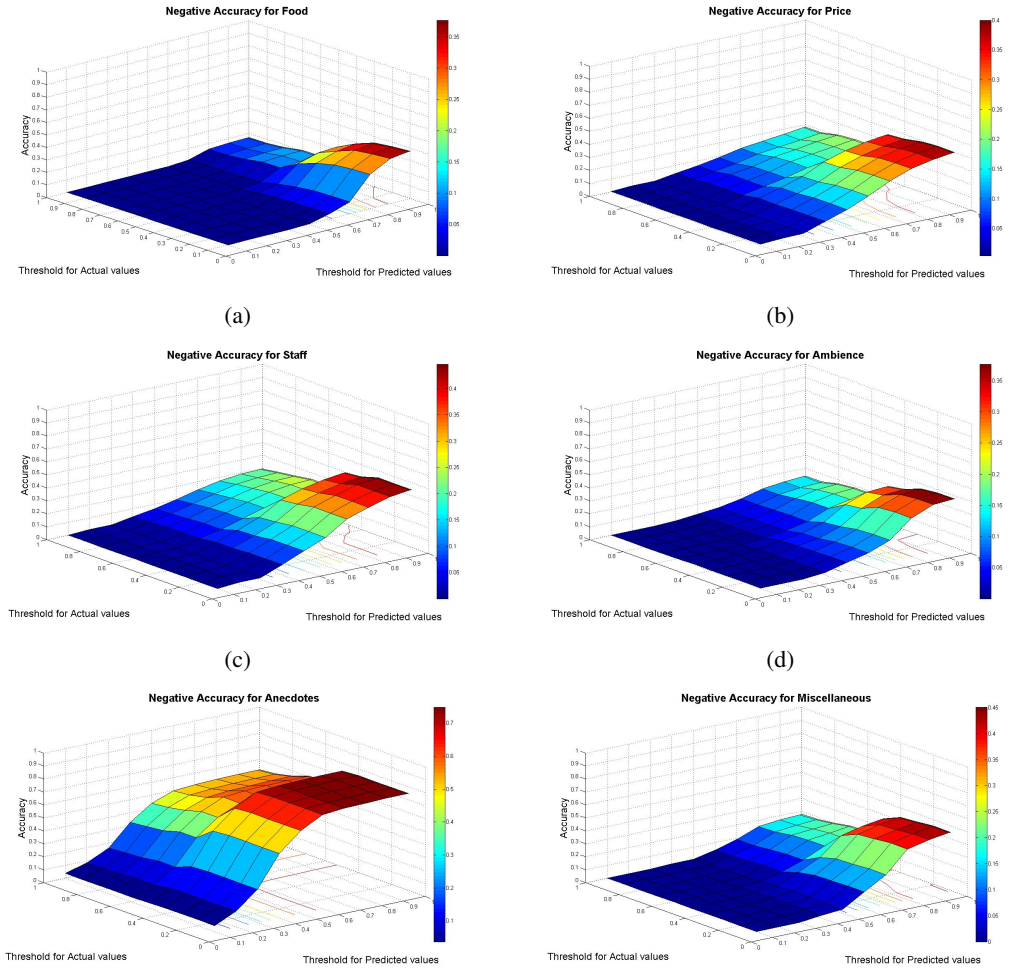


Figure 5.10: Negative Accuracy for Averaging Predictor

Anecdotes

The negative accuracy for Anecdotes, as seen in fig 5.10(e), shows a maximum of 0.75 for an actual threshold of 0.1 and a predicted threshold of 1.0. At an actual threshold of 1.0, the accuracy shows a maximum value of 0.52 for a predicted threshold of 1.0 and a minimum of 0.03 for a predicted threshold of 0.1. Whereas, at an actual threshold of 0.1, the accuracy shows a maximum value of 0.75 for a predicted threshold of 1.0 and a minimum of 0.04 for a predicted threshold of 0.1.

Miscellaneous

The negative accuracy for Miscellaneous, as seen in fig 5.10(f), shows a maximum of 0.45 for an actual threshold of 0.1 and a predicted threshold of 1.0. At an actual threshold of 1.0, the accuracy shows a maximum value of 0.18 for a predicted threshold of 1.0 and a minimum of 0.48 for a predicted threshold of 0.4. Whereas, at an actual threshold of 0.1, the accuracy shows a maximum value of 0.45 for a predicted threshold of 1.0 and a minimum of 0.0 for a predicted threshold of 0.1.

5.4.5 Inference

Our data corpus is far richer in sentences with a positive sentiment than negative sentences. This fact is greatly reflected by the averaging predictor since it has a high precision for predicting positive reviews in all of the six categories. We now compare the performance of the iIB-based prediction method with the performance of the average-based predictor.

Precision and Recall

The positive precision for our iIB-based predictor is detailed in section 5.2.1 and in fig 5.1. The positive precision for the average-based predictor is detailed in section 5.4.1 and in fig 5.7.

The positive precision for the averaging predictor is high because it averages the positive sentiment heavy evidence to predict that the review will also be positive. This fact causes the peak in precision to occur at a threshold value of 1.0 for both predicted and actual reviews just like iIB-based prediction. The maximum precision for Food, Price, Staff and Miscellaneous is greater for average-based than for iIB-based prediction. The trend for positive precision for Food and Staff is overall greater for the average-based prediction than for iIB-based prediction.

The exact opposite phenomenon occurs while predicting the negative. The average-based predictor still has a positive sentiment heavy evidence to predict the sentiment for the test review. As a result, the $PosLev(u_t, r_t, C_i)$ is high for a negative review also, requiring a high $Threshold_{Negative}(C_i)$ to predict it as negative.

Average-based predictions require higher $Threshold_{Negative}(C_i)$ for all categories, as compared to iIB-based predictions since most of their reviews have a high predicted $PosLev(u_t, r_t, C_i)$. The high predicted $PosLev(u_t, r_t, C_i)$ itself shows that the average-based predictor fails to predict

an overall negative sentiment. This fact shows the failure of the average based predictor to make successful prediction of any sentiment other than the positive sentiment due to the dearth of evidence for all other sentiments.

Accuracy

The trend in accuracy for predicting positives using iIB-based prediction (fig 5.3) and average-based prediction (fig 5.9) shows the same maximum (at an actual threshold of 0.9 and predicted threshold of 0.0) for five out of six categories (0.935 for *Food*, 0.791 for *Price*, 0.81 for *Staff*, 0.842 for *Ambience* and 0.82 for *Miscellaneous*). The maximum accuracy for *Anecdotes* is 0.65 using iIB-based prediction and 0.44 using average-based prediction. This is primarily due to the lack of positive evidence to correctly predict positive anecdotal review composition. Most of the anecdotal sentences are classified as neutral by the sentence classification system.

For iIB-based prediction, the positive accuracy shows a slight drop as predicted threshold increases to 0.1 from 0.0. It then remains almost constant, for *Food, Price, Anecdotes and* , followed by a gradual decrease (for predicted thresholds > 0.5).

For average-based prediction, the positive accuracy tends to remain at its maximum for predicted thresholds from 0.0 to 0.3. It then tends to show a sharper drop than iIB-based prediction to reach a minimum much lower than the minimum observed for iIB based prediction.

The above trend is mainly due to the fact that average-based prediction tends to have a more normal distribution than iIB-based prediction. iIB-based prediction has demonstrated that it can predict a majority of reviews in the polar regions close to 0.0 and 1.0 for $PosLev(u_t, r_t, C_i)$ as seen in the distribution histograms plotted in fig 4.1. Hence, iIB-based prediction maintains higher accuracy towards high values of predicted threshold.

For lower values of predicted threshold, the iIB-based prediction is not performing any better than the average-based prediction. But at higher values of predicted threshold, the iIB-based prediction has performs much better due to its capability of predicting polar reviews as opposed to the average-based prediction's tendency to predict reviews with a $PosLev(u_t, r_t, C_i)$ near the mean predicted threshold value of 0.5.

The accuracy of iIB-based prediction for predicting negative reviews is much higher than the accuracy of average-based prediction to predict negative reviews.

Thus, iIB-based prediction demonstrates a better trend in accuracy for effectively predicting the $PosLev(u_t, r_t, C_i)$ of test reviews than the baseline method of average-based prediction.

5.5 Combining Positive and Negative Thresholds

In sections 5.2 and 5.3, we studied how precisions decrease as recalls increase. We also observed that we need to find a tradeoff between the precisions and recalls and find the best possible values for $Threshold_{Positive}(C_i)$ and $Threshold_{Negative}(C_i)$. We also need to use both the thresholds to find both the positive and negative reviews. All those reviews which are between these two thresholds are considered neutral.

Observing the positive precision in fig 5.1, we need to have $Threshold_{Positive}(C_i)$ close to 1.0 if not exactly 1.0. Similarly, from 5.4, $Threshold_{Negative}(C_i)$ should be assigned a value close to 0.0 if not at 0.0.

We experimented by varying $Threshold_{Positive}(C_i)$ from 0.5 to 1.0 and $Threshold_{Negative}(C_i)$ from 0.5 to 0.0 for both actual and predicted reviews. Using these values in the algorithm 1, we classified all reviews into *Positive, Negative* or *Neutral* sentiments.

We computed the accuracy for each set of actual and predicted, positive and negative thresholds. We define accuracy as the ratio of the number of reviews predicted with the correct sentiment (*Positive, Negative* or *Neutral*) to the total number of reviews in the test set as shown in equation 5.4

$$Accuracy_{C_i} = \frac{\sum_{k=1}^{3} N_{C_i}(S_k, S_k)}{\sum_{k=1}^{3} \sum_{j=1}^{3} N_{C_i}(S_k, S_j)} \tag{5.4}$$

We, thus, computed the accuracy of our algorithm for each set of these threshold values. Setting $Threshold_{Positive}(C_i)$ close to 1.0 and $Threshold_{Negative}(C_i)$ close to 0.0 should allow us to correctly predict most of the positive and negative reviews. To reduce the complexity, we report the highest observed values of accuracy for each of the six categories. The highest attained accuracies are reported in table 5.1.

The accuracy of predicting *Food* preference is the highest (amongst all categories) at 0.86. The accuracy for *Staff*, *Ambience* and *Price* preference prediction is also high at 0.8, 0.79 and 0.71 respectively. The accuracy for *Anecdotes* and *Miscellaneous* is moderate at 0.61 and 0.63 respectively. The accuracy for *Anecdotes* is low due to the fact that most anecdotal sentences are either neutral

Category	Thresholds				Accuracy
	Actual		Predicted		
	Positive	Negative	Positive	Negative	
Food	0.8	0.3	0.6	0.5	0.8651
Price	0.7	0.3	0.6	0.4	0.7093
Staff	0.8	0.3	0.6	0.2	0.8047
Ambience	0.7	0.3	0.6	0.2	0.7907
Anecdotes	0.8	0.5	0.6	0.5	0.6086
Miscellaneous	0.8	0.2	0.7	0.2	0.6279

Table 5.1: Accuracy

or conflict in sentiment which reduces our accuracy for predicting whether the sentiment will be positive or negative. The *Miscellaneous* category is, after all, a category or outliers. The sentiment is difficult to predict because of a somewhat uniform distribution of the predicted values, as seen in fig 4.1(f). It is therefore difficult to predict the overall sentiment of the review using thresholds.

With these values of $Threshold_{Positive}(C_i)$ and $Threshold_{Negative}(C_i)$, we can classify a predicted review as *Positive, Negative* or *Neutral* in at least four of the categories namely *Food, Staff, Price and Ambience* using algorithm 1 since our prediction accuracy is high for these categories. We can predict, with limited accuracy, whether a certain user would write *positive, negative* or *neutral* anecdotal sentences for the restaurant for which the user wants us to predict his or her sentiments. Using our method, we can generate a report for the user for a target restaurant which contains what we believe to be his or her sentiments for the various aspects of the restaurant. This approach should give a more qualitative understanding to the user than a simple star or numerical rating, on a scale of 1 to 5, as done in the prior work on the URSA project in [8, 10].

Chapter 6

Related Work

Online reviews are the best resource to learn the views of your peers [5]. Many users have a vague idea of the product or services they wish to have and therefore would like to get recommendations. Due to the dearth of recommendation systems, users have to go through the frustrating task of accessing and searching text reviews manually. Many previous works have focused on designing a good recommendation system. These various techniques have been compared in [12] and [3]. The recent Netflix challenge [2] brought a lot of attention to collaborative filtering and recommendation systems. The data used by the Netflix project and other typical recommendation systems like the pioneer GroupLens project [20], is comprised of highly structured metadata such as the rating given by a user to a product. Our work, on the other hand, utilizes only the unstructured textual content of reviews to make predictions.

Identification of topical and sentiment data from a textual review has been an open question. Review processing has focused on identifying sentiments or product features [6, 18, 7, 24] or a combination of both [13, 15, 1, 26]. Textual features and expressed sentiments can be identified using unsupervised classification methods also. Unsupervised methods do not require manually annotated set for training the classifiers. An unsupervised text classification technique for Citysearch restaurant reviews data set is presented in [4].

Identification of individual product features and sentiments is the focus of many studies such as [13] and [19]. However, these studies have not used the extracted features for collaborative filtering. Most of the work in sentimental analysis concentrates at the review level. Our work focuses on a sentence as a unit for the sentiment analysis. Thus, a review was modeled as a fine-grained combination of topics and sentiments.

In [8], the authors focus on using sentiment analysis to predict a single rating that the user might give to a particular restaurant. We have used a similar collaborative filtering technique to predict

the composition of the review the user might write for a particular restaurant and then we predict the sentiments the user may have for each category of rating topics, such as food, staff, etc., for that restaurant. The main difference between these two studies is that we have used a topic based soft clustering technique which gives us different sets of cluster membership probabilities for each category of rating topics such as food, staff, etc. In contrast, the clustering technique used in [8] employs all categories and sentiments of the reviews to give a single set of cluster membership probabilities.

Our clustering mechanism is based on the iterative Information Bottleneck algorithm. The Information Bottleneck (IB) method was first introduced in [25]. The IB algorithm involves squeezing the information that the input variable X contains about the relevant variable Y through a compact bottleneck formed by a limited set of clusters. It has been successfully applied for clustering and unsupervised classification in document classification [22], unsupervised image clustering [11], word clustering, biological and many other applications.

Chapter 7

Conclusions

Collaborative filtering largely employs clustering algorithms to cluster users based on different features. We proposed that users should be clustered differently based on different categories of features. Our argument was that if a user has a taste for food similar to his family and a choice of restaurant ambience similar to his or her colleagues, then that user should be clustered similarly to his or her family while predicting for food preferences and to his or her friends while predicting ambience preferences.

We employed the iterative Information Bottleneck algorithm [23] to help us with the categorical clustering. We predicted the composition of a review that the user may write for the target restaurant using the cluster membership probabilities, which we obtain from iIB, and other reviews written for that restaurant. Using the predicted composition and algorithm 1, we predicted the sentiment the user would have for each of the six categories.

We found that the precisions (equation 5.1) are high if prediction thresholds are polar, i.e. $Threshold_{Positive}(C_i) \approx 1.0$ and $Threshold_{Negative}(C_i) \approx 0.0$. However, the recalls (equation 5.2) are low for these threshold values. Therefore, we found a tradeoff between precision and recall by varying $Threshold_{Positive}(C_i)$ and $Threshold_{Negative}(C_i)$ from 0.5 to 1.0 and 0.5 to 0.0 respectively and calculating the accuracy (equation 5.3). We found that the accuracy is the highest for the threshold values listed in table 5.1. These accuracy values prove that our predicted reviews can indeed be classified to give them sentiments such as *Positive, Negative* or *Neutral*. The high accuracy values which we obtained in table 5.1 validate our hypothesis.

The previous work done on the URSA project helped us understand that collaborative filtering can be performed on unstructured textual data if relevant features can be extracted from this data [8, 10]. In addition to the findings reported by the authors of [8, 10], our results help us to understand that collaborative filtering can use categorical clustering and predictions can be made with a good

accuracy.This accuracy depends on the available features and concentration of evidence for each feature in the corpus. Moreover, our work can be used to make fine-grained predictions and suitable recommendations in distinct categories. Our work can provide the user with predictions for his or her would-be sentiment for six different categories for a restaurant which he or she may visit instead of a single score-based prediction provided by the method proposed by the authors of [8, 10].

In this paper, we showed that,

1. Feature-based soft-clustering can be performed on unstructured textual content.

2. Fine-grained predictions for each feature can be provided to the user.

We would have liked to perform a user survey to determine whether fine-grained feature-based prediction provided by our method is easier to understand than a score-based prediction. But the time constraint of the Master's thesis prevented us from pursuing this direction. Our method can be used in multiple scenarios to guide users towards making an educated decision using predictions in broken down categories rather than a single score-based or binary prediction which leaves no choice for the user other than following it blindly or disregarding it completely.

References

[1] N. Archak, A. Ghose, and P. G. Ipeirotis. Show me the money!: Deriving the pricing power of product features by mining consumer reviews. In *SIGKDD*, 2007.

[2] J. Bennet and S. Lanning. The Netflix Prize. In *KDD Cup and Workshop*, 2007.

[3] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52, 1998.

[4] S. Brody and N. Elhadad. An unsupervised aspect-sentiment model for online reviews. In *NAACL HLT*, 2010.

[5] J. A. Chevalier and D. Mazylin. The effect of word of wouth on sales: Online book reviews. *Journal of Marketing Research*, 43(3):345–354, August 2006.

[6] K. Dave. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *International Conference on World Wide Web*, pages 345–354, 2003.

[7] M. Gamon. Sentiment Classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *COLING*, pages 841–847, 2005.

[8] G. Ganu, N. Elhadad, and A. Marian. Beyond the Stars: Improving Rating Predictions using Review Text Content. In *Proc. of Twelfth International Workshop on the Web and Databases (WebDB 2009)*, Providence, Rhode Island, USA, June 2009.

[9] G. Ganu and A. Marian. Improving rating prediction using textual information in online user reviews. Technical report, Rutgers University DCS Technical Report No. 685, 2011.

[10] G. Ganu, A. Marian, and N. Elhadad. Ursa - user review structure analysis: Understanding online reviewing trends. Technical report, Rutgers University DCS Technical Report No. 668, 2010.

[11] J. Goldberger, H. Greenspan, and S. Gordon. Unsupervised image clustering using the information bottleneck method. In *Proceedings of the* 24^{th} *DAGM Symposium on Pattern Recognition*, pages 158–165, 2002.

[12] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

[13] M. Hu and B. Liu. Mining and summarizing customer reviews. In *SIGKDD*, pages 168–177, 2004.

[14] T. Joachims. A Support Vector method for Multivariate performance measures. In *JCML*, 2005.

[15] S. M. Kim and E. Hovy. Identifying and analysing judgement opinions. In *HLT-NAACL*, 2006.

[16] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. of the International Joint Conference on AI*, 1995.

[17] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. In *IEEE Internet Computing*, pages 7:76–80, 2003.

[18] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up: Sentiment classification using machine learning techniques. In *ACL-EMNLP*, pages 79–86, 2002.

[19] A. M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *HLT-EMNLP*, pages 339–346, 2005.

[20] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Reidl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proc. of the ACM conference on Computer Supported Cooperative Work*, pages 175–186, 1994.

[21] S. Siegel and J.N. John Castellan. *Nonparametric Statistics for the Behavioral Sciences, Second Edition*. McGraw-Hill, 1988.

[22] N. Slonim, N. Friedman, and N. Tishby. Unsupervised document classification using sequential information maximization. In *Proc. of the 25th Annual ACM SIGIR conf. on Research and Development of Information Retrieval*, 2002.

[23] N. Slonim and N. Tishby. *Information Bottleneck: Theory and Applications*. Phd thesis, Hebrew University, Jerusalem, Israel, 2002.

[24] B. Snyder and R. Barzilay. Multiple aspect ranking using the good grief algorithm. In *NAACL*, 2007.

[25] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[26] I. Titov and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL*, 2008.

Appendix A

Sentence Classification

The authors of [8] and [10] performed sentence classification on the URSA data corpus. The sentences in the reviews were classified into six categories : *Food, Price, Staff, Ambience, Anecdotes* and *Miscellaneous*. The supervised classification used manually annotated text as described in A.1. The sentences in the remaining reviews were then classified into categories and sentiments as detailed in A.2.

A.1 Manual Sentence Annotation

Classification of text along different topics and sentiments is a challenge [10]. To classify sentences into the above mentioned categories and sentiment classes, a training set of approximately 3400 sentences was manually annotated with both category and sentiment information. To check for agreement, 450 of these sentences were annotated by three different annotators. The kappa coefficient (K) measures pairwise agreement among a set of annotators making category judgements, correcting for expected chance agreement [21]. A Kappa value of 1 implies perfect agreement, the lower the value, the lower the agreement. The inter-annotator agreements for the annotations were very good (Kappa above 0.8) for the Food, Price and Staff categories and Positive sentiment. The negative sentiment (0.78), Neutral and Conflict sentiments, Miscellaneous and Ambience categories all had good agreements (Kappa above 0.6). The ambiguous Anecdotes category is the only one for which the Kappa value was moderate (0.51).

A.2 Automatic sentence Classification

Support Vector Machine classifiers [14] were trained and tested on the manually annotated data (one classifier for each category and one for each sentiment type). Features for all classifiers were

Sentence Category	Accuracy	Precision	Recall
Food	0.8432	0.8143	0.7672
Staff	0.9192	0.8100	0.7294
Price	0.9552	0.7911	0.7355
Ambience	0.9099	0.7010	0.5464
Anecdotes	0.8720	0.4915	0.4426
Miscellaneous	0.7940	0.6128	0.6420

Sentiment	Accuracy	Precision	Recall
Positive	0.7332	0.7494	0.7660
Negative	0.7942	0.5323	0.4568
Neutral	0.8086	0.3234	0.2354
Conflict	0.9206	0.4396	0.3568

Table A.1: 7-Fold cross validation of classifier results

stemmed words. Svm light[1], with default parameters, was used for this purpose.

7-fold cross validation [16] was performed and used accuracy precision and recall to evaluate the quality of the classification (see Table A.1). Precision and recall for the main categories of Food, Staff and Price and the positive sentiment were high (> 0.7), while these values are lower for the Anecdotes, Miscellaneous, Neutral and Conflict categories. These low results could be due to the ambiguous nature of these categories but also due to the small amount of training instances in the corpus for these categories in particular.

[1] http://svmlight.joachims.org

Appendix B

iIB Implementation

We implemented the iIB algorithm for probabilistic clustering of similar users. The pseudo code for the Iterative Information Bottleneck Algorithm, adapted from [23], is given in algorithm 2.

As shown in the pseudo code, the iIB algorithm takes as input the joint probability distribution $p(x, y)$ containing the information X contains about the relevant variable Y. This probability distribution is represented as a matrix. The algorithm also begins with a fixed value for the trade-off parameter β, the number of clusters and a convergence parameter ϵ.

The iIB algorithm begins with a random initialization of the matrix representing the conditional probability $p(t|x)$. The probability distributions $p(t)$ and $p(y|t)$ are computed using the conditional probability $p(t|x)$ and the IB Markovian relation $T \leftrightarrow X \leftrightarrow Y$, as shown in equations B.1 and B.2.

During each iteration, the three probability matrices $p(t|x)$, $p(t)$ and $p(y|t)$ are updated using the equations B.3, B.4 and B.5 respectively. Note that, the computations in equation B.4 and B.5 are similar to the those in the initialization equations B.1 and B.2, but they use the updated probability matrices. The iterations are stopped when the algorithm converges, i.e., the JS divergence between the $p(y|t)$ matrices of two consecutive iterations is below the threshold parameter ϵ. Please refer to [23] for the proof of convergence of the iIB algorithm. On completion at the n^{th} iteration,the iIB algorithm returns the probabilistic clustering of the data points as represented in the updated matrix $p^{(n)}(t|x)$.

Algorithm 2 Iterative Information Bottleneck (iIB) Clustering

Input:
Joint distribution $p(x, y)$
Trade-off parameter β
Cluster cardinality parameter M, convergence parameter ϵ

Output:
A soft partition T of X into M clusters.

Initialization:
Randomly initialize $p(t|x)$ and find the corresponding $p(t)$, $p(y|t)$ using the following equations:

$$p(t) = \sum_x p(x).p(t|x) \tag{B.1}$$

$$p(y|t) = \frac{1}{p(t)} \sum_x p(t|x).p(x, y) \tag{B.2}$$

Iteration:
repeat

$$p^{(m+1)}(t|x) = \frac{p^{(m)}(t)}{Z^{(m+1)}(x, \beta)} e^{-\beta D_{KL}[p(y|x)||p(y|t)]} \qquad \dots \forall t \in T, \forall x \in X \tag{B.3}$$

$$p^{(m+1)}(t) = \sum_x p(x).p^{(m+1)}(t|x) \qquad \dots \forall t \in T \tag{B.4}$$

$$p^{(m+1)}(y|t) = \frac{1}{p^{(m+1)}(t)} \sum_x p^{(m+1)}(t|x).p(x, y) \qquad \dots \forall t \in T, \forall y \in Y \tag{B.5}$$

until $\forall x \in X, JS_{(1/2,1/2)}[p^{(m+1)}(t|x), p^{(m)}(t|x)] \leq \epsilon$

Definitions:
D_{KL} is the Kullback-Leibler divergence given by:

$$D_{KL}[p_1||p_2] = \sum_x p_1(x).log\frac{p_1 x}{p_2 x} \tag{B.6}$$

The Jensen-Shannon (JS) divergence is given by:

$$JS_{(\pi_1,\pi_2)}[p_1, p_2] = \pi_1 D_{KL}[p_1||\bar{p}] + \pi_2 D_{KL}[p_2||\bar{p}] \tag{B.7}$$

where $0 < \pi_1, \pi_2 < 1, \pi_1 + \pi_2 = 1$ and $\bar{p} = \pi_1 p_1 + \pi_2 p_2$

Appendix C

Labels

The following labels have been used throughout the thesis to denote the six categories and four sentiments.

	Category
C_1	Food
C_2	Price
C_3	Staff
C_4	Ambience
C_5	Anecdotes
C_5	Miscellaneous

Table C.1: Category Labels

	Sentiment
S_1	Positive
S_2	Negative
S_3	Neutral
S_4	Conflict

Table C.2: Sentiment Labels

Vita

Yogesh Kakodkar

2011 M. S. in Electrical and Computer Engineering, Rutgers University

2008 B. E. in Computer Engineering from Mumbai University

2004 Graduated from Ramnivas Ruia Junior College.