Stochastic Modeling of Semantic Structures of Online Movie Reviews

Limeng Pu
Louisiana State University and Agricultural and Mechanical College

STOCHASTIC MODELING OF SEMANTIC STRUCTURES OF ONLINE MOVIE REVIEWS

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

in

The School of Electrical Engineering and Computer Science

by
Limeng Pu
B.S., University of Electrical Science and Technology of China, 2013
August 2015

Acknowledgements

This thesis has been carried out at Louisiana State University, from initial ideas to final execution. I would like to thank my supervisor, Dr. Shuangqing Wei, for all the guidance and discussion around the NLP, statistical analysis and modeling. I would also like to express my appreciation to Dr. Yejun Wu for providing help on the semantic data transformation and Dr. Xuebing Liang for being a member of my defense committee.

I would also like to thanks the authors of Stanford CoreNLP and many other community of programmers and engineers who offered help. I can't finish my thesis without their effort.

At last, I want to thank my parents for supporting me throughout all those years, and my lovely wife for taking care of my family and her most generous support.

Table of Contents

Abstract

Facing the enormous volumes of data available nowadays, we try to extract useful information from the data by properly modeling and characterizing the data. In this thesis, we focus on one particular type of semantic data --- online movie reviews, which can be found on all major movie websites. Our objective is mining movie review data to seek quantifiable patterns between reviews on the same movie, or reviews from the same reviewer. A novel approach is presented in this thesis to achieve this goal. The key idea is converting a movie review text into a list of tuples, where each tuple contains four elements: feature word, category of feature word, opinion word and polarity of opinion word. Then we further convert each tuple into an 18-dimension vector. Given a multinomial distribution representing a movie review, we can systematically and consistently quantify the similarity and dependence between reviews made by the same or different reviewers using metrics including KL distance and distance correlation, respectively. Such comparisons allow us to find reviewers sharing similarity in generated multinomial distributions, or demonstrating correlation patterns to certain extent. Among the identified pairs of frequent reviewers, we further investigate the category-wise dependency relationships between two reviewers, which are further captured by our proposed ordinary least square estimation models. The proposed data processing approaches, as well as the corresponding modeling framework, could be further leveraged to develop classification, prediction, and common randomness extraction algorithms for semantic movie review data.

Key words: online movie review, modeling semantic structure, natural language processing (NLP), distance correlation, ordinary least square (OLS) estimation

Chapter 1. Introduction

With the fast development of internet, the information we can access has grown exponentially, especially the emerging of Web 2.0, which emphasizes the participation of users. More and more websites, such as *Amazon* (http://www.amazon.com) and *IMDB* (http://www.imdb.com) encourage people to post their opinions and reviews for the information they are interested in. This thesis proposes a novel approach to interpret the semantic data from online movie reviews and gives the quantifiable results, which we can further use for prediction and random key generation. Natural language processing (NLP) and statistical analysis used are hot topics in the application of data mining and pattern discovery.

Essentially, movies are like a multidimension information source. From cast to story, they contain a lot of information. Human mind or brain is like a filter. We filter out the information given by the movie, and leave the comment with the information we desire. So there must exist some kind of dependency between reviews on the same movie, since those comments share a same information source. The problem is how we are going to compare two reviews consisting of words and sentences to find their dependency. However, the reviews are usually lengthy and only a few sentences of them are really useful information to us. So we need to first summarize the movie. Transform the unstructured movie reviews into structured data, which can be further converted into quantitative results. After the transformation, we use some mathematical approach to model our transformed data and perform further analysis on the modeled data.

Though most of the work in online reviews mining are limited to qualitative results given by various kinds of sentiment analysis, some of the works provide us inspiration on the processing of textual movie data. The most important and inspiring to our work are:

- Sentiment classification. Also called sentiment orientation, opinion orientation or sentiment polarity [14]. It is based on the idea that a document/text expresses an opinion on entity from a holder and tries to measure the sentiment of that holder towards the entity [15]. In [2], Pang and Lee performed sentiment classification on online movie reviews, which tags a sentence with its polarity, using Naïve Bayes, support vector machine and other machine learning techniques. And gives the performance of different techniques. [13] measures the intensity of each sentiment with a score ranging [-1,1], where -1 stands for maximum negativity and 1 stands for maximum positivity, which inspires us to give polarity a score in the following data transformation.
- Opinion summarization: It is especially focused on extracting the main features of an entity shared within one or several documents and the sentiments regarding them [16]. We only consider the single-document perspective in this task, which consists in analyzing internal facts present within the analyzed document. We simply look for the feature and opinion word pairs that satisfies certain

grammatical relationship. In [1], Zhuang, Jing and Zhu used multi-knowledge based approach to summarize a movie review into multiple short sentences using feature and opinion word pairs. [3] used Latent Dirichlet Allocation (LDA) to model the topic of reviews and identify the feature and opinion word pairs without the knowledge of the domain.

We decided to use the approach in [1] to perform the data transformation we needed since they already construct the dictionaries and grammatical relationship template that we can use specifically for movie review domain, while the LDA approach requires large amount of manually labeled data as training set, which we don't have. We also infuse our approach with some techniques in sentiment analysis to make it more refined.

However, the field we are entering, which uses semantic data to get quantifiable results for prediction or clustering, is entirely new to us. And all the work related to this problem is qualitative. We can rarely find any existing work that seeks quantitative results. Hope our work can shed some light on how to further use the information provided by the reviews for more than simple sentiment analysis.

In this thesis, we decompose the problem of review mining and modeling into the following subtasks: 1) mining the feature and opinion word tuples from the original review text; 2) transform the tuples into 18 dimension vectors and further normalize to distributions; 3) use Person correlation coefficient and distance correlation to perform initial model and cluster based on reviewers' own and common commented movie set; 4) use ordinary least square (OLS) estimation to do the inference between a pair of reviewers based on their common commented movie set. We propose a frame work to transform semantic data into numerical data using multiple dictionaries and map them to vector space. After the normalization, they become distributions. This is the main novel idea in this thesis.

The remainder of the thesis is organized as follows: Chapter 2 gives the details about how to transform the semantic data into structured numeral data. Chapter 3 is about the statistical analysis for initial clustering and modeling of the transformed data. Chapter 4 describe how we use the data for inference using OLS to perform inference based on two reviewers' data on the common commented movie set. Finally, the conclusions and future work in presented in Chapter 5.

Chapter 2. Data Transformation

In this chapter, we are going to introduce a novel approach to transform the semantic data from an online movie review into the numerical data, an 18 dimension vector, which will further be normalized into a multinomial distribution. The overview of the frame work is shown in Figure 1. Two dictionaries are used to record information for features and opinions in movie review domain. Feature opinion pairs are generated via some grammatical rules. According to the category of feature word and polarity of opinion word, they are transformed into a vector. Then we normalize the vector by the total number of valid comments made. More details of the proposed approach will be provided in the following.

2.1 Problem formulation

Let $\boldsymbol{X} = X_1, X_2, \dots, X_N$ be a set of reviews on a movie. Each review X_i consists of a set of sentences $< s_1, s_2, \dots, s_M >$. The following can be defined using the similar definition in [1]:

Definition 1. (movie feature): A movie feature is a movie element (e.g., screenplay, music) or a movie-related person (e.g., director, actor) that has been commented on.

Since everyone may use different words or phrases to describe the same movie feature, the authors in [1] manually defined some feature classes (categories) according to IMDB. The categories are divided into two groups: *Element* and *People*. The *Element* categories are: *OA* (overall), *ST* (story), *CH* (character), *VP* (visual effects), *MS* (music and sound effects) and *SE* (special effects). The *People* categories are: *PAC* (actors and actresses), *PDR* (directors), *PPL* (others including editor and screen writer). Each category contains a set of words or phrases, which will be introduced in the next section.

Definition 2. (relevant opinion of a feature): The relevant opinion of a feature is a set of words or phrases that expresses a *positive (POS)* or *negative (NEG)* opinion on the feature.

Definition 3. (feature-opinion pair): A feature-opinion pair consists of a feature and a relevant opinion. If both the feature and the opinion appear in sentence s, it is an explicit pair. If the feature or the opinion doesn't appear in sentence s, we call it an implicit pair.

For example, in sentence "The movie is great!", the feature is "movie" and the relevant opinion is "great". The pair *(movie, great)* is an explicit pair. While in sentence "Master piece!", only relevant opinion "master piece" appears, which certainly describes a movie. We give a feature "movie/film" to it. The pair (movie/film, master piece) is called implicit pair. In our case, we only consider the explicit pairs. One thing to note is, only the appearance of two words is not enough to count them as a valid feature-opinion pair. They have to satisfy certain grammatical

relation to be a valid pair. We will introduce the requirements in the following section.

The task of data transformation is to find all the feature-opinion pairs in a sentence and then identify the category of feature word and polarity of opinion word. Finally turn them into a normalized vector.

2.2 Feature and opinion extraction

2.2.1 Feature words

In [1], the authors introduce an approach to extract feature and opinion word pairs to summarize a movie review. We adopt that approach with some minor changes since we don't have a large quantity of manually labeled data.

According to the results from [4], when people leave comments on product features, the words they used converge. And the same can be said about movies according to the statistic results on the labeled data in [1]. For each feature class, if we remove the feature words with frequency lower than 1% of the total frequency of all feature words, the remaining words can still cover more than 90% feature occurrences. In addition, for most feature classes, the number of remaining words is less than 20. The feature words for different category (non-name) is shown in Table 1. The results indicate that we can use a few words to capture most features. Therefore, we save these remaining words as the main part of our feature word list. Because the feature words don't usually change. That's why we don't need to add the synonymic words to expand.

In movie reviews, movie names and people names can also be feature word, and a name can be expressed in different forms, such as first name only, last name only or full name. To make name recognition easy, we build a cast library as a special part of feature word list by using the movie cast data from *IMDB* (http://www.imdb.com). Since movie fans are only interested in the important movie-related people, such as actor, actresses and directors. We choose only some of the cast data from *IMDB*. For the mining of names of people or movie, some regular expressions are used to check the word sequences beginning with a capital letter. Table 2. shows the regular expressions we used to extract names.

If a sequence is matched with a regular expression, we will search the cast library. If the matched sequence is in the cast library, the corresponding sequence is the recognition result. We take it as a feature word.

The names from our cast library and those we summarize from results in [1] together form our feature word list. We first perform the regular expression detection. Then for those matched ones, we run them through the cast library, if they are in the library, we accept them as feature words. Finally, we match all the words in the non-name feature word list. All the qualified ones are our feature words.

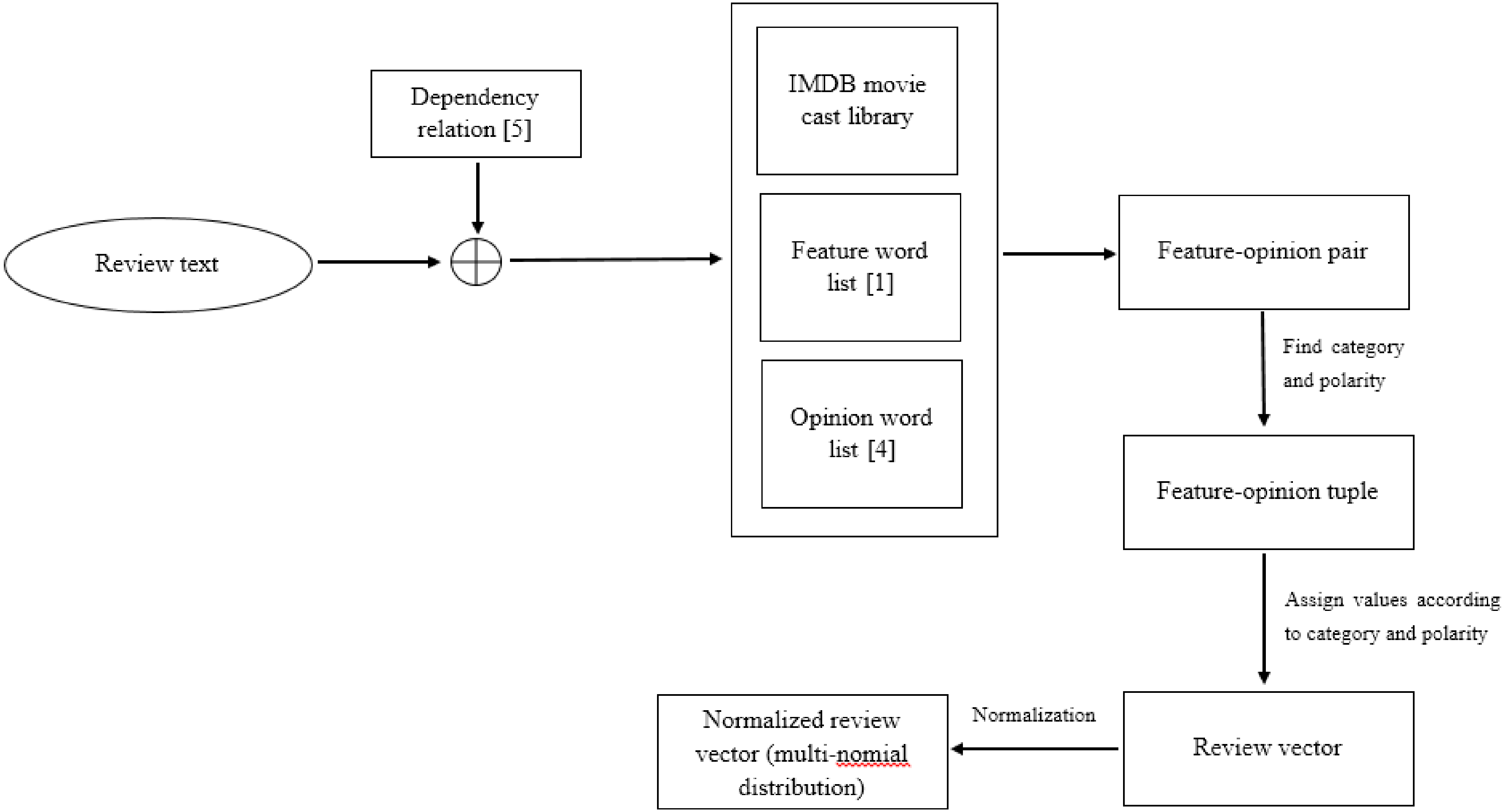


Figure 1. Overview of data transformation framework

Table 1. The feature word for non-name related list.

Category	Feature words
OA	movie, film, DVD, show, shows, series, quality
ST	story, stories, plot, script, script-writing, storyline, dialogue, dialogues, screenplay, dialogue, ending, finale, line, lines, tale, humor, tales
CH	character, characters, characterization, role, roles
VP	scene, scenes, fight-scene, fight-scenes, action-scene, action-scenes, action-sequences, action-sequence, set, sets, battle-scene, battle-scenes, picture, pictures, scenery, sceneries, setting, settings, visual-effect, camerawork, visual-effects, color, colors, background, image, editing
MS	music, score, song, songs, sound, soundtrack, theme, broadway
SE	special-effect, effect, effects, stunt, CGI, SFX
PAC	acting, performance, performers, actor, actors, actress, actresses, performs,
PDR	director
PPL	producer, cast, screenwriter, editor, singer, cameraman, composer

Table 2. The regular expression for movie-related names.

Regular expression	Meaning
[A-Z][a-z]+ [A-Z][a-z]+ [A-Z][a-z]+	First name, middle name, last name
[A-Z][a-z]+ [A-Z][a-z]+	First name, last name
[A-Z][a-z]+ [A-Z][.] [A-Z][a-z]+	Abbreviation for middle name
[A-Z][.] [A-Z][.] [A-Z][a-z]+	Abbreviation for first and middle name
[A-Z][.] [A-Z][a-z]+	Abbreviation for first name, no middle name

2.2.2 Opinion words

For the same reason, the opinion word list is based on an existing opinion lexicon by Hu and Liu [4], which is a list of English words taken from social media. The words have been manually labeled by Dr. Liu's NLP group at UIC. It contains nearly 6800 words labeled positive or negative, including some common typos people make on social media. We simply match the word appeared in the sentence with the words in the lexicon and give the corresponding polarity. One thing to note is same opinion word may have different polarity in movie domain as it has in other domain. For example, “simple” is a neutral word, but in movie domain, “simple” is usually a negative word. As building an opinion lexicon requires huge amount of manually labeled data, which we don’t have. So we didn’t take the domain difference into consideration.

2.2.3 Explicit feature-opinion pair mining

One sentence can have more than one feature and opinion words. Therefore, after locating the feature and opinion words in a sentence, we need to know if they can form a valid feature-opinion pair. For example, "Leonardo Decaprio is amazing but the movie is a disaster", in this sentence we have feature words: *Leonardo Decaprio, movie*, and opinion words: *amazing, disaster*. Now we have four combinations of feature-opinion pair, but apparently only two are available, which are *[Leonardo Decapro, amazing]* and *[movie, disaster]*. To solve this problem we use the dependency grammar graph. Figure 2. is an example of dependency grammar graph generated by Stanford Parser (http://nlp.stanford.edu/software/lex-parser.shtml), without distinguishing governing words and depending words.

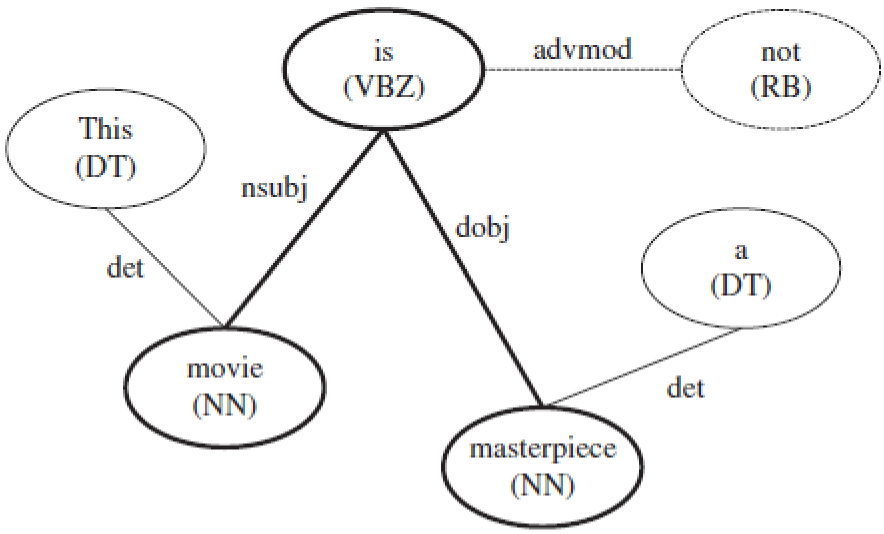


Figure 2. Example dependency grammar graph on sentence "This movie is a masterpiece" [1].

We use the results acquired in [1], where they give a set of frequent dependency relations in movie review domain for feature and opinion word. Table 3. shows the dependency relations.

Table 3. Frequent relations template in movie review domain.

Dependency relation	Feature word's part-of-speech	Opinion word's part-of-speech
NN - amod - JJ	NN	JJ
NN - nsubj - JJ	NN	JJ
NN - dobj VB	NN	VB
NN - nsubj - NN	NN	NN

In order to find the valid feature opinion pair we need to, first, tag the part-of-speech of feature and opinion word. If they match the part-of-speech, then we find the dependency of those two words. If the relation also matches, it is a valid pair. To

achieve all above, we need to parse each sentence and get the dependency relations and POS tag of each word in the sentences using Stanford CoreNLP [5], and match them with the template we have.

Now the explicit pair mining task can be achieved in two steps using feature, opinion word lists and the frequent dependency relations. First, in a sentence, the word lists are used to find all the feature words and opinion words. Then the dependency relations are involved to check if the pairs are valid. For the feature-opinion pair that is matched by the grammatical template, whether there is a negation relation or not is also need to be checked. If there is a negation relation, the polarity is transferred according to the simple rules: not POS → NEG, not NEG → POS.

2.2.4 Tuple generation

After we mine all the feature-opinion pairs, we need more information about those pairs so we can further utilize them.

Definition 4. (feature-opinion tuple): A feature-opinion tuple is a tuple contains a feature-opinion pair, the corresponding category to the feature and the corresponding polarity to the opinion.

The feature-opinion tuple is the final goal for our feature and opinion extraction. We use the tuples to convert to vectors. For example, in sentence "The movie is great", the feature-opinion pair is *(movie, great)*. The feature "movie" is in category *OA (*overall*)*. And the polarity of "great" is obviously *POS (*positive*)*. So the corresponding feature-opinion tuple for this sentence is *(movie, OA, great, POS)*. This tuple is what we use to generate the review vector.

We already have the category and polarity when we build the feature and opinion word lists, so it is easy to match those information and put them into the desired entry in the tuple.

2.3 Vectorization of tuples

Now we have the feature-opinion tuple. But this is still semantic data, we can not use them for any quantitative analysis. The next step is convert them into 18 dimensions vectors.

Definition 5. (review vector): For a reviewer R_i's review X_i^k on movie M_k, we define 9 categories, and each has 2 polarities, which gives us 18 dimensions in total. For all the feature-opinion tuples $T_j^{i,k}$ generated from review X_i^k, if it is about category l with polarity q, we add +1 to corresponding entry in the 18 dimension vector.

For example, below is a review text we download from *Amazon* (http://www.amazon.com):

> Like many who watched the most recent "Oscar's" show, all we kept hearing bout, was this film, "Million Dollar Baby". It kept upstaging its rival, "The Aviator", at every turn. I was skeptical that a film could be that good, and thought, oh it's just because Clint is up

there in age, etc. Let me say that, I'm also like the world's biggest Clint Eastwood fan, but boy was I wrong! This film is brilliant. It begins with a beautiful narrative by Morgan Freeman, introducing us to the main characters in the film, "Frankie Dunn", played by Clint Eastwood, a semi-retired trainer in dusty rat hole of a gym called "The Hit Pit".' This gym is filled with all kinds of likeable and not so likeable fighters and wannabe fighters. Morgan Freeman plays "Scrap" a long since retired boxer who helps "Frankie" run the day to day operations of the gym. One day, out of the blue, in walks "Maggie", brilliantly played by Hillary Swank. She's drawn there by an insatiable desire to be a boxer. She is determined to have 'Frankie' train her and will not take no for an answer. "Frankie" has trained many great boxers but is hesitant to train a girl, as he refers to her. From this premise, one might say okay, sounds familiar, I can guess how this turns out. You would be wrong! This film begins one way and takes a swift turn southward, and never lets up. It explores what motivates people, their background, and their eventual success or failure, and the ramifications of this. The characters are perfectly cast, the script is entertaining, and the acting is exceptional. If you don't cry during this film, you just may not be human. I won't spoil it by revealing what happens but just to say, that I haven't seen a film this brilliant since "Titani", and "Shawshank Redemption". I would dare say you may not see a better film this year.

The tuples we generated from this review are:

[('film', 'OA', 'good', 'POS'), ('film', 'OA', 'brilliant', 'POS'), ('script', 'ST', 'entertaining', 'POS'), ('film', 'OA', 'not brilliant', 'NEG'), ('film', 'OA', 'better', 'POS')]

According to **Definition 5**, the corresponding vector is:

$$V = [3,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$

We are going to normalize the vector using **Definition 6** so that they are distribution-like, and we can use them for more analysis, such as compute the Kullback–Leibler (KL) distance and profile the reviewers. We'll take closer look into them in the following chapter.

Definition 6. (normalized review vector): Given a review vector $V = [v_1, v_2, \dots, v_{18}]$, we have the total number of comments is $n = \sum_{i=1}^{18} v_i$. Now use *n* to normalize *V*. We have normalized vector $V' = [v_1', v_2', \dots, v_{18}']$, such that $\sum_{i=1}^{18} v_i{}' = 1$.

Following the example above, we have the original vecto:

$$V = [3,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$

According to **Definition 6**, the total number of comments *n = 5*, so the normalized vector is

$$V' = [\frac{3}{5}, \frac{1}{5}, \frac{1}{5}, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$

This normalization procedure will give us a multinomial distribution [11] so we can use for further analysis, such as profile the reviewer and compute the KL distance. We'll take closer look into them in the following section.

Chapter 3. Statistical Analysis

In this chapter, we will introduce all the quantitative analysis of semantic reviews we performed and the results about our selected reviewers using the normalized review vectors, including the selection of reviewer, profile of reviewer, Pearson's product-moment correlation coefficient (correlation coefficient) for orthogonality analysis, KL distance , distance correlation and hypothesis test for checking the dependency. We take a hierarchical approach when we analyze our reviewers. We first look at the statistic between all reviewers and we pick several pairs of reviewers to look into the statistic between all 18 categories and polarity of them. In this way, we not only get the style and tendency of reviewers, but also the relationship across all categories between different reviewers. This will help us to do the inference based on one reviewer's data. Details will be given in the following sections.

3.1 Selection of reviewers

Our objective is to find dependency between reviewers and use them for further inference. To make sure the data we select can support our mission, we want to select the reviewers with relatively large number of common reviews (reviews on the same set of movies).

Definition 7. (common interest matrix): For each reviewer *i*, we have $\boldsymbol{X_i} = (X_i^1, X_i^2, \dots, X_i^{N_i})$ reviews, where N_i is the total number of reviews of reviewer *i*. To select suitable reviewers, we define common interest matrix CMI as follow:

$$
\mathrm{CIM}_i = \begin{array}{c} \\ R_1 \\ R_2 \\ \vdots \\ R_i \end{array}
\begin{array}{c}
\begin{array}{cccc} R_1 & R_2 & \cdots & R_i \end{array} \\
\begin{bmatrix}
0 & |S_{12}| & \cdots & |S_{1i}| \\
|S_{21}| & 0 & \cdots & |S_{2i}| \\
\vdots & \vdots & \ddots & \vdots \\
|S_{i1}| & |S_{i2}| & \cdots & 0
\end{bmatrix}
\end{array}
$$

where S_{ij} is the vector stores the product ID of the common movies, and $|S_{ij}|$ is the cardinality of matrix S_{ij}. Since we are looking at different reviewers, we are not interested in the values of diagonal.

With this matrix, we can rank the reviewers based on how many reviews they have on the same movies. In this way, we can better select our object for data acquisition. Also this matrix itself convey a lot of information. For example, if a reviewer has relatively large value entries with all other reviewers, this is also helpful information for our future analysis.

We perform the CMI procedure on some pre-selected reviewers, who have more than 100 reviews posted, in the dataset. According to the result from CMI, we choose 10 reviewers as our subjects. The resulting CMI in shown in Figure 3.

$$
\begin{array}{c c}
 & \begin{array}{cccccccccc} R_1 & R_2 & R_3 & R_4 & R_5 & R_6 & R_7 & R_8 & R_9 & R_{10} \end{array} \\
\begin{array}{c} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \\ R_7 \\ R_8 \\ R_9 \\ R_{10} \end{array} &
\left[\begin{array}{cccccccccc}
0 & 205 & 215 & 129 & 189 & 80 & 78 & 181 & 54 & 176 \\
205 & 0 & 136 & 60 & 97 & 50 & 67 & 92 & 42 & 127 \\
215 & 136 & 0 & 133 & 223 & 82 & 144 & 194 & 92 & 435 \\
129 & 60 & 133 & 0 & 91 & 16 & 40 & 46 & 20 & 80 \\
189 & 97 & 223 & 91 & 0 & 87 & 114 & 161 & 53 & 89 \\
80 & 50 & 82 & 16 & 87 & 0 & 66 & 84 & 24 & 35 \\
78 & 67 & 144 & 40 & 114 & 66 & 0 & 86 & 45 & 64 \\
181 & 92 & 194 & 46 & 161 & 84 & 86 & 0 & 75 & 82 \\
54 & 42 & 92 & 20 & 53 & 24 & 45 & 75 & 0 & 63 \\
176 & 127 & 435 & 80 & 89 & 35 & 64 & 82 & 63 & 0
\end{array}\right]
\end{array}
$$

Figure 3. The CMI of chosen reviewers.

We can see the highest number of common movie set we have is 435, which is not a large number of reviews when it comes to data mining. It would be better to choose reviews based on the types of movie (e.g., drama, romance, action and so on). The resulting CMI will be based on type of movies rather than common movie sets. For each movie type, we have a CMI. This approach has two advantages. The first is we will certainly get more data points for our mining and learning task. The second, people have tendency to remark on different categories for different type of movies when they leave comments. For example, when we comment on an action movie, we usually values more on effects and the performance; while for romance or drama, we tend to care more about the plot and story. This will make our analysis more specific and efficient. Yet, we are not able to do that due to complexity it requires to grab movie type information from *IMDB* automatically. This will be listed as a future work. As a result, the following analysis will be based on the matrix in Figure 3. without considering the movie types.

3.2 Profile vector of reviewers

Before we can look into the details of common reviews, we want to better understand our reviewers. That is only their own information is needed. In order to do that we need to profile them. This is also the first step towards the initial clustering of reviewers. The procedure is described below:

Definition 8. (profile vector): Given the review vectors $\boldsymbol{V}_i$ from reviewer *i* and N_i the total number of reviews by reviewer *i*, for entry T_j (category and polarity) in the profile vector **T**, we have

$$T_j = \frac{\sum_{l=1}^{N_i} v_j^l}{N_i}$$

Each entry in **T** corresponds to a certain category and polarity. Essentially T_j is the empirical mean of a particular category and polarity. With this vector, we can plot the distribution of this reviewer. Below are two distributions from two reviewers (No.3 and No. 5).

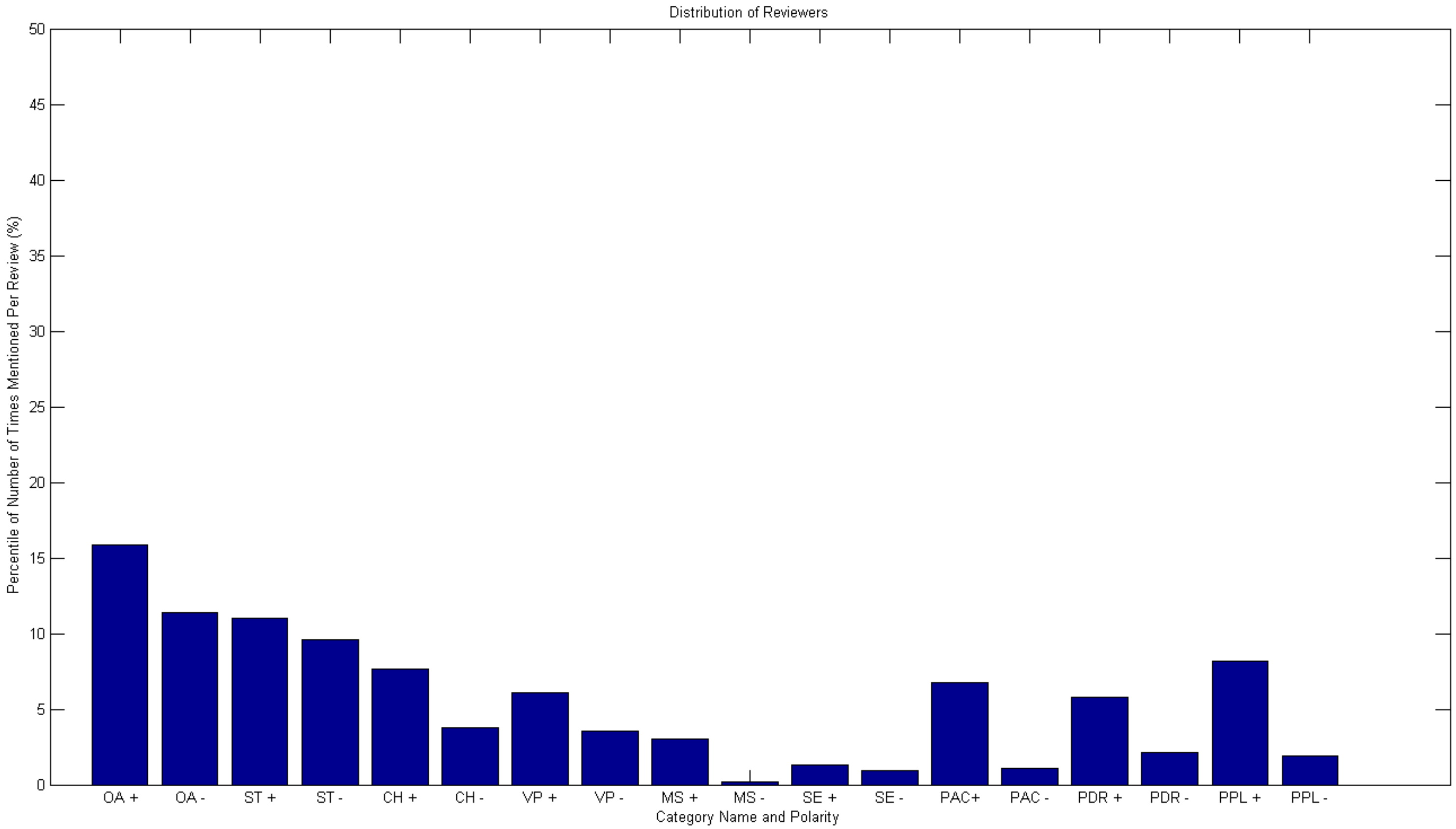


Figure 4. Profile distribution of reviewer No.3. X-axis stands for all the categories and polarities. Y-axis is the percentile of number of comments per review.

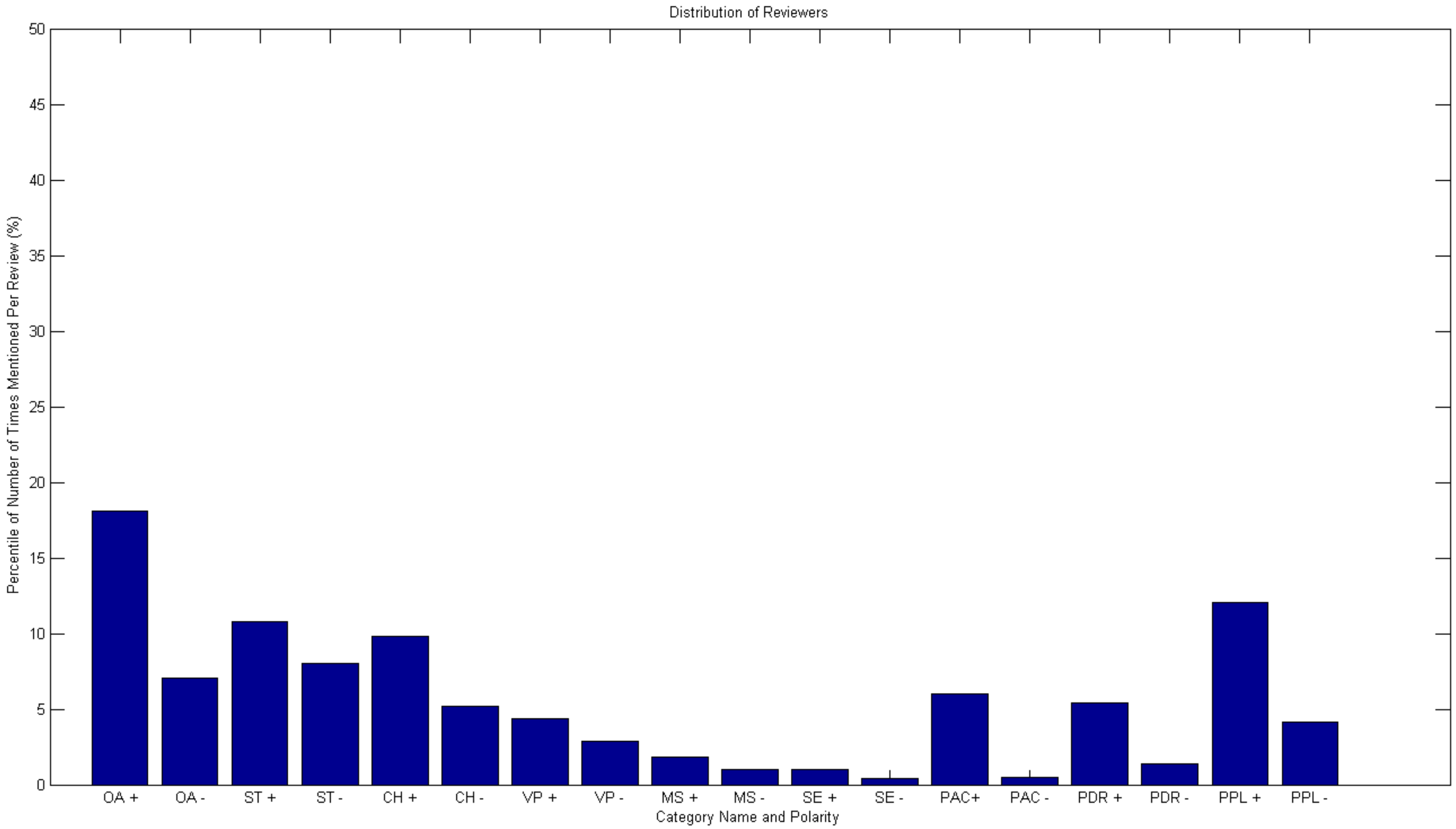


Figure 5. Profile distribution of reviewer No.5. X-axis stands for all the categories and polarities. Y-axis is the percentile of number of comments per review.

The profiling procedure does not involve any common movie sets. The profile is the style and tendency of one review by himself. From the profile distribution, we can see there exists some resemblance from the profiles of these two reviewers. Two other important conclusions can be drawn by observing all the profiles we have acquired:

- Music (MS) and special effect (SE) are rarely mentioned by reviewers, which means they are 0 most of the time in review vectors. This is extremely important if we are going to perform some dimensionality reduction. We can rule out MS and SE if necessary, since basically nobody talks about them.
- Reviewers tend to leave more positive comments than negative comments. This can be explained by positive bias we have as human nature. We tend to be more positive than negative no matter in movie reviews or other aspects of our daily lives [12].

The profile is another means to help us initially cluster the reviewers. It capture the basic style and tendency of our reviewers. But it is essentially a qualitative results by our observations. We need to look deeper to seek quantitative results.

3.3 KL distance analysis

In this section, Kullback–Leibler (KL) distance is used to analysis our reviewers. We compute the symmetrized KL distance between all our chosen reviewers for reviewer clustering.

In information theory, the Kullback–Leibler distance [6][7][8], which is proposed by *Kullback, S.; Leibler, R. A.*, is a non-symmetric measure of the difference between two probability distribution.

Definition 9. (Kullback–Leibler (KL) distance): For discrete probability distributions *P* and *Q*, the KL distance of *Q* from *P* is defined to be:

$$d_{KL}[P||Q] = \sum_{i} P(i)\ln\left(\frac{P(i)}{Q(i)}\right)$$

KL distance is a measure of the information lost when *Q* is used to approximate *P*. We employ a symmetrized version of KL distance for our experiments. The definition of symmetrized KL distance we use is given below.

Definition 10. (symmetrized KL distance): Given two discrete probability distributions P_i and P_j, the symmetrized KL distance D_{ij} can be defined as:

$$d_{ij} = d_{KL}(P_i||P_j) = \sum_{x=1}^{18} P_i(x)\ln\left(\frac{P_i(x)}{P_j(x)}\right)$$

$$d_{ji} = d_{KL}(P_j||P_i) = \sum_{x=1}^{18} P_j(x)\ln\left(\frac{P_j(x)}{P_i(x)}\right)$$

$$D_{ij} = \left(\frac{1}{1/d_{ij} + 1/d_{ji}}\right)$$

where $P_i(x)$ is the probability of reviewer k on category and polarity x. And we define

$0 \times \ln\left(\frac{0}{0}\right) = 0$.

We compute the KL distance between all the reviewers using their profiles. The result is shown below

0.000000	0.152039	0.065450	0.098483	0.075493	0.085478	0.051350	0.138170	0.097649	0.143230
0.152039	0.000000	0.123747	0.128126	0.169252	0.118658	0.101226	0.224578	0.108138	0.238670
0.065450	0.123747	0.000000	0.100502	0.026391	0.104500	0.053334	0.063541	0.169815	0.094326
0.098483	0.128126	0.100502	0.000000	0.091919	0.070628	0.032216	0.139242	0.078240	0.164066
0.075493	0.169252	0.026391	0.091919	0.000000	0.110038	0.078674	0.056262	0.182178	0.102951
0.085478	0.118658	0.104500	0.070628	0.110038	0.000000	0.121759	0.131996	0.054698	0.143492
0.051350	0.101226	0.053334	0.032216	0.078674	0.121759	0.000000	0.141155	0.151661	0.175532
0.138170	0.224578	0.063541	0.139242	0.056262	0.131996	0.141155	0.000000	0.232732	0.097796
0.097649	0.108138	0.169815	0.078240	0.182178	0.054698	0.151661	0.232732	0.000000	0.139768
0.143230	0.238670	0.094326	0.164066	0.102951	0.143492	0.175532	0.097796	0.139768	0.000000

Figure 6. The symmetrized KL distance matrix between 10 chosen reviewers using profiles of reviewers.

From the matrix above, we can see that the symmetrized KL distance is relatively small across all the reviewers. We think this is because in the process of profile the reviewers, we not only normalize the vector, also average them out. So we lose a lot information during the process. But this result can still provide us some insight about our reviewers. We construct two graphs to represent the relationship between all our reviewers using the symmetrized KL distance. First, according to this distance matrix we can rank the distance of each reviewer with other reviewers. And then we have chosen the top *k* nearest neighbor of each reviewer. Construct a weighted graph using the distance and rankings. Two nodes are connected in the graph only when they are both within the *k* nearest neighbor list of each other. Since no common movie set is needed in the computation of KL distance, the graph is more about how the overall style and tendency differs between reviewers. Figure 7 is the graph constructed from 3 nearest neighbor.

The following Figure 8 is the complementary graph of the nearest 3 neighbors. We perform the same procedure for the furthest 3 neighbors. The complementary is shown in Figure 7.

From two graphs, we can see that some reviewers have smaller distance with each other at the same time, for example, No. 3 and No. 5. This makes them the better subjects to use for the coming analysis. This two graphs will help select our reviewer pairs for further analysis. Also we believe this graph is also useful for clustering reviewers. But the profile process causes too much information loss. We want an approach that is able to find patterns between reviewers and preserve the information at the mean time. This is where the distance correlation comes into play.

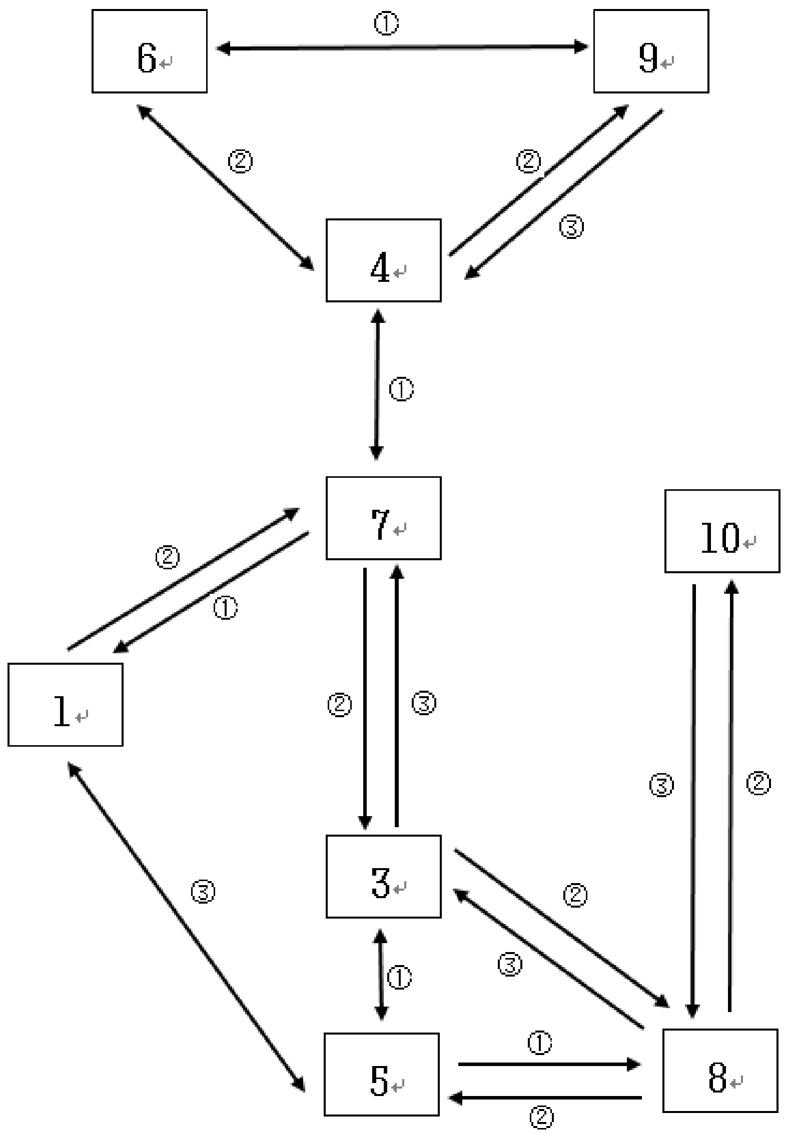


Figure 7. The nearest 3 neighbors graph based on the symmetrized KL distance using profiles of reviewers. No common movie set is needed. The number on the edge indicates the rank of one reviewer in another reviewer's list. If it is double arrow, it means they both have the same rank in each other's list.

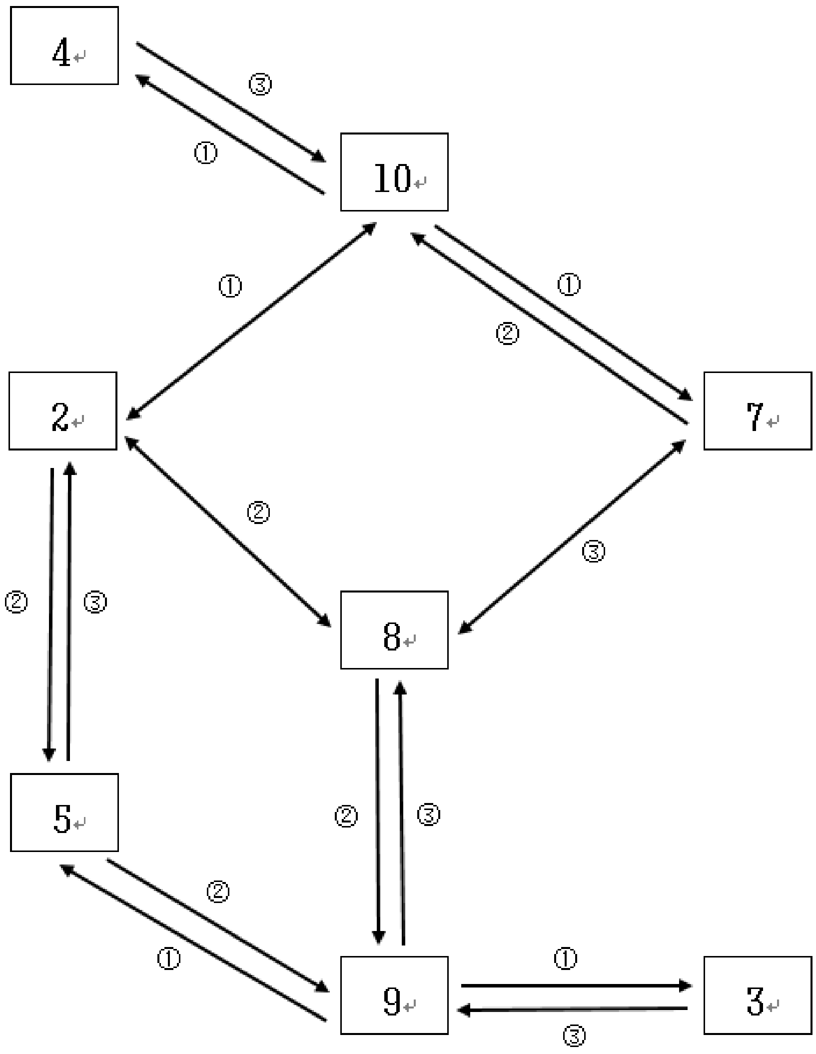


Figure 8. Complementary graph of Figure 7. using furthest 3 neighbor. Also no common movie set is needed. The number on the edge indicates the rank of one reviewer in another reviewer's list. If it is double arrow, it means they both have the same rank in each other's list.

3.4 Distance correlation and dependency test

3.4.1 Distance correlation analysis

After our initial analysis on the reviewers, we want to look deeper into their dependency. The classic dependency measure, Pearson's correlation coefficient, is mainly sensitive to a linear relationship between two variables. Also the correlation coefficient is 0 doesn't imply true independency. In our case, the relationships of semantic data from different people are most likely not linear. So we want to seek another approach that can truly capture the dependency relationship between two

reviewers or between two categories from two reviewers. This is the reason we choose distance correlation.

Distance correlation [9], is introduced 2007 by *Székely, Rizzo and Bakirov* to overcome the defects of correlation coefficient (only accounts for linear relationship). This measure is derived from a number of other quantities that are used in its specification. Specifically: distance variance, distance standard deviation and distance covariance. Distance correlation provides a new approach to the problem of testing the joint independence of random vectors. For all distributions with finite first moments, distance correlation *R* generalizes the idea of correlation in two fundamental ways:

- *R(X,Y)* is defined for *X* and *Y* in arbitrary dimensions;
- *R(X, Y) = 0* characterizes independence of *X* and *Y*.

Distance correlation has properties of a true dependence measure, analogous to product-moment correlation coefficient.

To define distance correlation, we have to define distance covariance first.

Definition 11. (distance covariance): The distance covariance (dCov) between random vectors X and Y with finite first moments is the nonnegative number *V(X, Y)* defined by:

$$V^2(X,Y) = ||f_{X,Y}(t,s) - f_X(t)f_Y(s)||^2$$

where f_X and f_Y is the characteristic function of *X* and *Y*.

Similarly, we have the distance variance (dVar) can be defined as:

$$V^2(X,X) = ||f_{X,X}(t,s) - f_X(t)f_X(s)||^2$$

Definition 12. (distance correlation): The distance correlation (dCor) between random vectors *X* and *Y* with finite first moments is the non-negative number *R(X, Y)* defined by:

$$R^2(X,Y) = \begin{cases} \dfrac{V^2(X,Y)}{\sqrt{V^2(X,Y)V^2(X,Y)}}, V^2(X,Y)V^2(X,Y) > 0 \\ 0 \qquad\qquad\qquad\quad , V^2(X,Y)V^2(X,Y) = 0 \end{cases}$$

Clearly the definition of *R* suggests an analogy with the product moment correlation coefficient.

The distance dependence statistics can be computed as follows. For an observed random sample $(\boldsymbol{X},\boldsymbol{Y}) = \{(X_k, Y_k): k = 1,2,\dots,n\}$ from the joint distribution of random vectors *X* and *Y*, We first compute all pairwise distances:

$$a_{j,k} = ||X_j - X_k||; j,k = 1,2,\dots,n$$

$$b_{j,k} = ||Y_j - Y_k||; j,k = 1,2,\dots,n$$

where || || is the Euclidean norm. Then we have the $\mathrm{n} \times \mathrm{n}$ distance matrix $a_{j,k}$ and $b_{j,k}$ Take all doubly centered distances,

$$A_{j,k} = a_{j,k} - \bar{a}_{j.} - \bar{a}_{.k} - \bar{a}_{..}$$

$$B_{j,k} = b_{j,k} - \bar{b}_{j.} - \bar{b}_{.k} - \bar{b}_{..}$$

where $\bar{a}_{j.}$ is the mean of j-th row, $\bar{a}_{.k}$ is the mean of k-th column and $\bar{a}_{..}$ is the grand mean. Then we have the distance covariance as,

$$\mathrm{dCov}^2(X,Y) = \frac{1}{n^2}\sum_{j,k=1}^{n} A_{j,k}B_{j,k}$$

And we have the distance variance,

$$\mathrm{dVar}^2(X) = \mathrm{dCov}^2(X,X)$$

Finally the distance correlation can be computed as,

$$\mathrm{dCor}^2(X,Y) = \frac{\mathrm{dCov}^2(X,Y)}{\sqrt{\mathrm{dVar}^2(X)\mathrm{dVar}^2(Y)}}$$

Next we compute the distance correlation between two reviewers based on their common movie set. The result is shown below:

1	0.204835	0.24488	0.298265	0.255132	0.303316	0.249674	0.279286	0.334524	0.246555
0.204835	1	0.297855	0.3163	0.275428	0.413163	0.239499	0.336248	0.250717	0.258312
0.24488	0.297855	1	0.292444	0.284889	0.435202	0.222844	0.319077	0.311367	0.207424
0.298265	0.3163	0.292444	1	0.344528	0.627462	0.262438	0.534632	0.587407	0.328882
0.255132	0.275428	0.284889	0.344528	1	0.354906	0.301038	0.324327	0.4122	0.440249
0.303316	0.413163	0.435202	0.627462	0.354906	1	0.306787	0.409633	0.535258	0.566082
0.249674	0.239499	0.222844	0.262438	0.301038	0.306787	1	0.335594	0.289137	0.326792
0.279286	0.336248	0.319077	0.534632	0.324327	0.409633	0.335594	1	0.323214	0.384624
0.334524	0.250717	0.311367	0.587407	0.4122	0.535258	0.289137	0.323214	1	0.33776
0.246555	0.258312	0.207424	0.328882	0.440249	0.566082	0.326792	0.384624	0.33776	1

Figure 9. The distance correlation matrix between all reviewers using review vectors based on common movie sets.

In this way, we successfully preserve all the information by using the review vectors themselves. Again we construct the 3 nearest neighbor graph and its complementary graph using the same technique mentioned in the previous section with distance correlation matrix. One thing to note is that different from the graph constructed using KL distance, the computation of distance correlation is based on the common movie set between two reviewers. This means the distance correlation graph emphasizes more about the dependency relationship between reviewers given the same set of movies they commented on.

We can see from the following Figure 10 and 11. that the results are different from the previous 3 nearest neighbor graph using KL distance. The main reason is that the distance correlation is about the dependency while KL distance is more about similarity between two reviewers. The distance correlation graph utilizes the common movie set between reviewers while the KL distance is just the distance between profiles of reviewers, of which some information is lost during the transformation. With the graphs, we are able to choose desired pair of reviewers for further analysis.

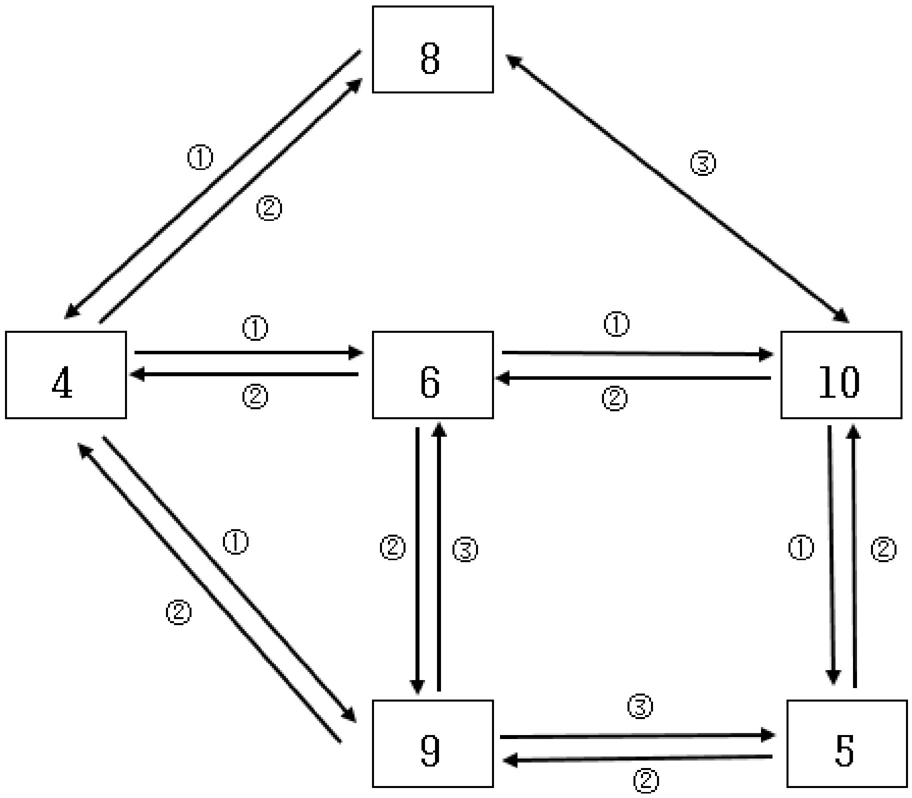


Figure 10. The 3 nearest neighbor graph constructed using distance correlation based on the common movie set between a pair of reviewers. The graph emphasizes more on the dependency relationship between reviewers. The number on the edge indicates the rank of one reviewer in another reviewer's list. If it is double arrow, it means they both have the same rank in each other's list.

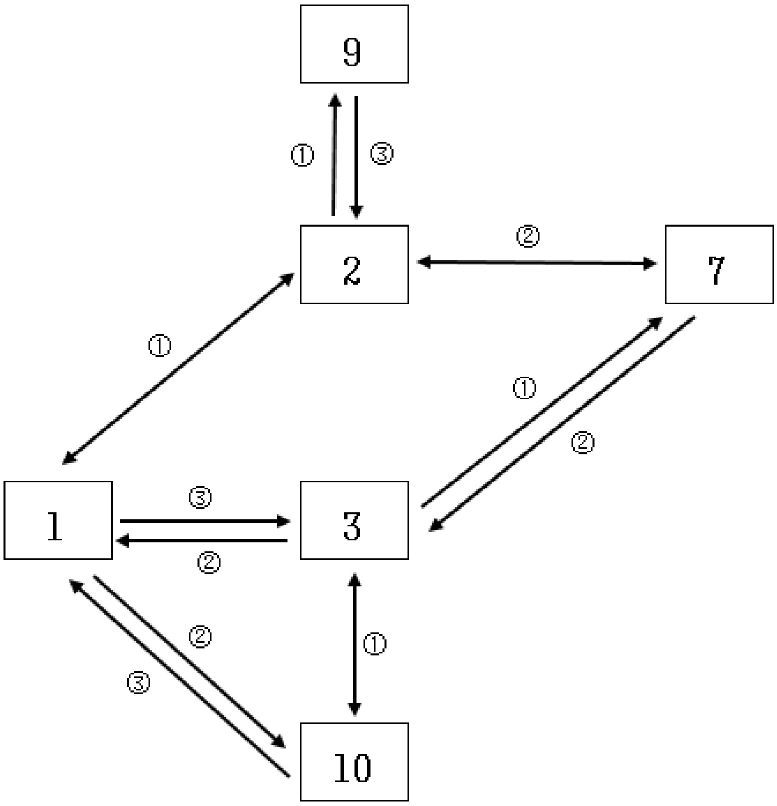


Figure 11. The 3 furthest graph constructed using distance correlation based on the common movie set. The number on the edge indicates the rank of one reviewer in another reviewer's list. If it is double arrow, it means they both have the same rank in each other's list.

3.4.2 2-D histogram

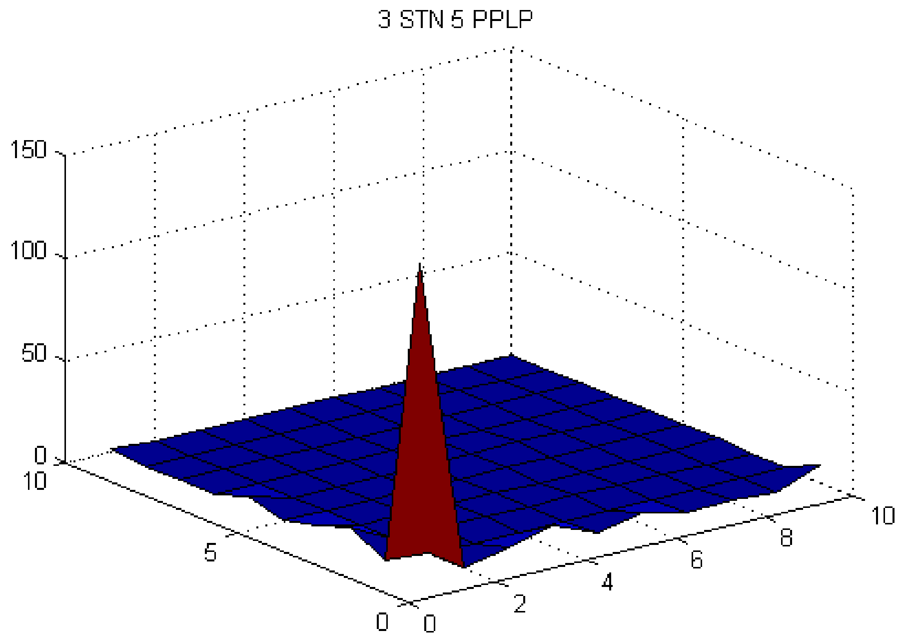
From the results above, we choose reviewer No. 1 and 2 as the example for weak dependency and reviewer No. 3 and 5 as the example for strong dependency (not in the graph but they have high distance correlation and relatively large common set of common movies). To gain a better understanding of the dependency between categories of two reviewers, we use the same approach when we compute the distance correlation between reviewers. We compute the distance correlation between different categories from two reviewers. An example of output matrix is given below:

3 \ 5	OA-P	OA-N	ST-P	ST-N	CH-P	CH-N	VP-P	VP-N	MS-P	MS-N	SE-P	SE-N	PAC-P	PAC-N	PDR-P	PDR-N	PPL-P	PPL-N
OA-P	0.053842	0.160458	0.068135	0.063562	0.088749	0.05533	0.090588	0.068336	0.05951	0.130359	0.115268	nan	0.045172	0.055013	0.082621	0.076785	0.076815	0.074089
OA-N	0.117041	0.173527	0.081532	0.082993	0.097846	0.066439	0.043576	0.050818	0.093133	0.107518	0.071971	nan	0.103641	0.037942	0.057294	0.04085	0.042682	0.072704
ST-P	0.054001	0.045067	0.061002	0.047704	0.055623	0.052168	0.063292	0.164989	0.084203	0.049325	0.144411	nan	0.046782	0.084126	0.076767	0.075044	0.072989	0.042735
ST-N	0.083968	0.077959	0.060535	0.080088	0.034205	0.057231	0.068053	0.145741	0.041043	0.039123	0.039612	nan	0.062774	0.039715	0.103009	0.109278	0.081168	0.046762
CH-P	0.046824	0.049599	0.069692	0.136556	0.067335	0.068007	0.054427	0.136447	0.208415	0.034675	0.035109	nan	0.064069	0.0352	0.071374	0.055406	0.060208	0.089195
CH-N	0.115573	0.050693	0.05322	0.15598	0.067656	0.068607	0.072712	0.055507	0.04651	0.164376	0.027585	nan	0.052039	0.027657	0.076335	0.110594	0.060151	0.043898
VP-P	0.062446	0.056957	0.070316	0.073033	0.107112	0.062112	0.075586	0.057702	0.048349	0.028322	0.133942	nan	0.093108	0.028751	0.045241	0.210128	0.051054	0.168781
VP-N	0.091399	0.057001	0.061663	0.066814	0.078115	0.12223	0.063735	0.047466	0.039772	0.023298	0.023589	nan	0.10092	0.023651	0.068824	0.069995	0.081879	0.044004
MS-P	0.062556	0.111296	0.050103	0.083053	0.059924	0.062635	0.047702	0.057465	0.058504	0.028205	0.028558	nan	0.08647	0.348272	0.063884	0.045068	0.052811	0.05009
MS-N	0.064273	0.031467	0.037042	0.033019	0.03107	0.027968	0.023362	0.017834	0.014943	0.008754	0.008863	nan	0.028777	0.446474	0.030612	0.013987	0.182578	0.022517
SE-P	0.048693	0.030456	0.16805	0.064964	0.030071	0.02707	0.190073	0.286072	0.014463	0.008472	0.008578	nan	0.027853	0.008601	0.078656	0.013537	0.051989	0.021793
SE-N	0.067733	0.032046	0.086001	0.149769	0.031642	0.028483	0.023792	0.018163	0.015218	0.008915	0.009026	nan	0.180567	0.00905	0.031176	0.014244	0.042428	0.022931
PAC-P	0.200472	0.06001	0.048186	0.069622	0.076547	0.063649	0.084343	0.043923	0.119311	0.041983	0.063899	nan	0.063815	0.042619	0.073948	0.078447	0.073101	0.069641
PAC-N	0.050097	0.136498	0.064293	0.039411	0.089028	0.033383	0.027885	0.255299	0.017837	0.010448	0.010579	nan	0.034349	0.010607	0.191361	0.23545	0.049727	0.026876
PDR-P	0.045039	0.117041	0.098641	0.051984	0.070929	0.119683	0.050959	0.126986	0.041092	0.032559	0.032966	nan	0.053858	0.033052	0.079069	0.084641	0.065905	0.045192
PDR-N	0.075329	0.043965	0.088364	0.057662	0.072229	0.046324	0.170773	0.042543	0.140727	0.22084	0.021142	nan	0.14394	0.021198	0.044934	0.033365	0.212676	0.053713
PPL-P	0.070028	0.110719	0.064155	0.126487	0.172399	0.058163	0.047972	0.046012	0.070939	0.041133	0.041647	nan	0.05921	0.059408	0.062478	0.092299	0.077184	0.063982
PPL-N	0.043273	0.254999	0.066833	0.070439	0.066281	0.059665	0.049838	0.038046	0.235907	0.018674	0.018907	nan	0.096895	0.018957	0.065305	0.029838	0.069269	0.062461

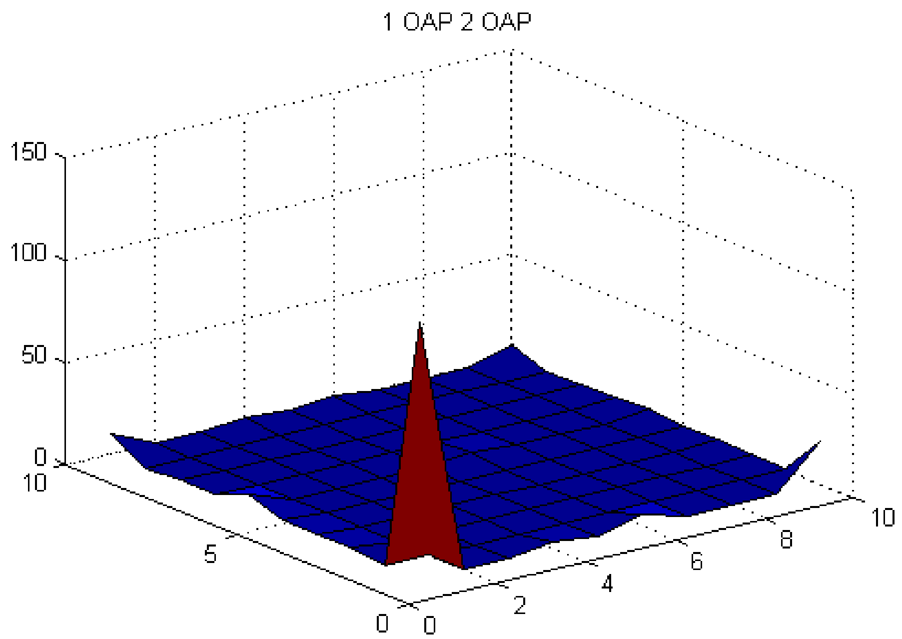
1 \ 2	OA-P	OA-N	ST-P	ST-N	CH-P	CH-N	VP-P	VP-N	MS-P	MS-N	SE-P	SE-N	PAC-P	PAC-N	PDR-P	PDR-N	PPL-P	PPL-N
OA-P	0.188821	0.071426	0.126917	0.074742	0.060032	0.068513	0.11104	0.054971	0.034841	nan	0.048944	0.034841	0.056879	nan	0.202872	0.069396	0.063293	0.183725
OA-N	0.086901	0.070471	0.052954	0.034114	0.13181	0.031271	0.097493	0.220532	0.015902	nan	0.022339	0.015902	0.03445	nan	0.021908	0.022546	0.026017	0.015902
ST-P	0.05227	0.066892	0.043393	0.064358	0.097005	0.057395	0.092484	0.057247	0.029018	nan	0.040765	0.029018	0.043824	nan	0.039979	0.041142	0.089723	0.029018
ST-N	0.058793	0.032943	0.046698	0.036025	0.199369	0.259939	0.05352	0.033129	0.016793	nan	0.023591	0.016793	0.036379	nan	0.023135	0.023809	0.027474	0.016793
CH-P	0.048369	0.055555	0.056538	0.223488	0.029254	0.121555	0.054111	0.033495	0.016978	nan	0.023851	0.016978	0.036781	nan	0.181988	0.024072	0.027778	0.016978
CH-N	0.08291	0.037657	0.023845	0.015361	0.012338	0.014081	0.022821	0.014126	0.007161	nan	0.010059	0.007161	0.015512	nan	0.009865	0.010152	0.011715	0.007161
VP-P	0.256117	0.051945	0.032892	0.021189	0.017019	0.019424	0.03148	0.019486	0.009877	nan	0.013876	0.009877	0.021398	nan	0.013608	0.350462	0.01616	0.009877
VP-N	0.052699	0.037311	0.023626	0.01522	0.012225	0.357203	0.022612	0.013997	0.007095	nan	0.009967	0.007095	0.01537	nan	0.009775	0.010059	0.011608	0.007095
MS-P	0.10603	0.02656	0.229206	0.010834	0.008702	0.009932	0.016096	0.009964	0.005051	nan	0.007095	0.005051	0.010941	nan	0.006958	0.007161	0.008263	0.005051
MS-N	nan	nan	nan	nan	nan	nan	nan	nan	nan	1	nan	nan	nan	1	nan	nan	nan	nan
SE-P	0.037513	0.02656	0.016818	0.010834	0.008702	0.009932	0.016096	0.009964	0.005051	nan	0.007095	0.005051	0.010941	nan	0.006958	0.007161	0.008263	0.005051
SE-N	nan	nan	nan	nan	nan	nan	nan	nan	nan	1	nan	nan	nan	1	nan	nan	nan	nan
PAC-P	0.065326	0.074861	0.06062	0.05342	0.042906	0.048968	0.079363	0.049126	0.024902	nan	0.034982	0.024902	0.11224	nan	0.034307	0.035306	0.175978	0.024902
PAC-N	0.037513	0.02656	0.016818	0.010834	0.008702	0.009932	0.016096	0.009964	0.005051	nan	0.007095	0.005051	0.010941	nan	0.006958	0.007161	0.008263	0.005051
PDR-P	0.080494	0.192036	0.048991	0.031561	0.025349	0.028931	0.046888	0.029024	0.014712	nan	0.020667	0.014712	0.194265	nan	0.020269	0.020859	0.02407	0.014712
PDR-N	0.037513	0.02656	0.016818	0.010834	0.008702	0.009932	0.016096	0.009964	0.005051	nan	0.007095	0.005051	0.010941	nan	0.006958	0.007161	0.008263	0.005051
PPL-P	0.028688	0.066853	0.052853	0.043599	0.035018	0.039966	0.05582	0.040094	0.020324	nan	0.028551	0.020324	0.044028	nan	0.028	0.028815	0.109578	0.020324
PPL-N	nan	nan	nan	nan	nan	nan	nan	nan	nan	1	nan	nan	nan	1	nan	nan	nan	nan

Figure 12. The distance correlation between all categories of reviewer pair No. 3 & No. 5 and No. 1 & No. 2("nan" stands for "not as number" due to 0 division).

We can see the distance correlation is not very high, but as long as they are not zero, we assume they are dependent. Our objective is to infer one reviewer's review from others. In order to do that, we want to have an intuitive understanding of the dependence we are looking for. Next, we empirical construct the joint probability mass function (2-d histogram) for the selected reviewers for certain categories. The result from reviewer pair No.3 & No. 5 and No. 1 & No. 2 is shown below:



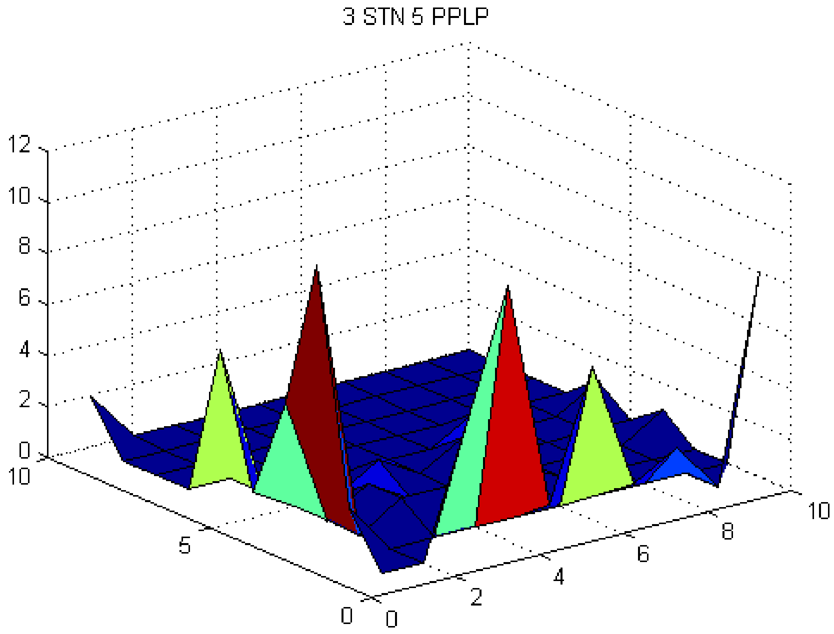
(a)



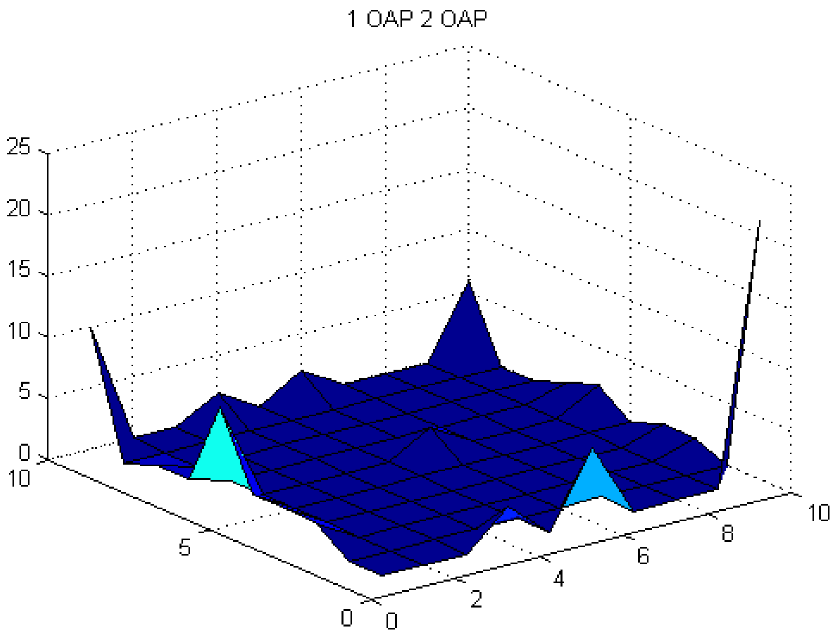
(b)

Figure 13. 2-d histogram (a) No.3 's *ST NEG* and No. 5's *PPL POS* (b) No.1 's *OA POS* and No. 2's *OA POS* (before elimination of double-zero component).

From the figure above, we can see the histogram is dominated by the double-zero component (both reviewers don't mention this category and polarity) at (0, 0). We can't pick out any patterns from the 2-d histogram. So we decide to eliminate all the double-zero component and run the experiment again. The result become better after the elimination of double-zero component.



(a)



(b)

Figure 14. 2-d histogram of reviewer No.3 's *ST NEG* and No. 5's *PPL POS* (after elimination of double-zero component).

The result is clearly improved a lot. We can use the 2-d histograms to find some patterns. But the size of the data we use is too small (average 50 data points for each histogram). We might be able to locate some patterns from the 2-d histogram. Now the 2-d histogram is just a demonstration. We fail to extract any patterns from the histograms.

Another reason we can't get anything out of 2-d histograms is, during the course of analyzing the 2-d histograms, some of the data for one category to another are orthogonal with each other, which means though they have high distance correlation, their correlation coefficient is very low (close to 0). In this case, most of values focus on the axis. For example, below is reviewer No. 3's *ST POS* and No. 8 *OA NEG*.

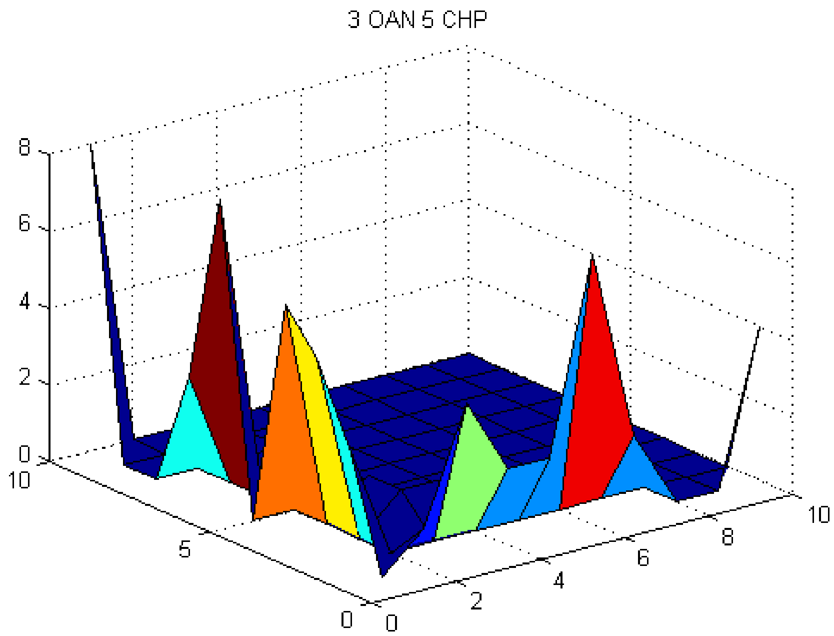


Figure 15. 2-d histogram of reviewer No.3 's *OA NEG* and No. 5's *CH POS* (after elimination of double-zero component), most of the non-zeros are on the axis.

We can see that most of the non-zero values are focused are the axis, which means either you don't mention it or I don't mention it. Though they have high distance correlation (0.63), the result is not useful at all. So only look at the distance correlation between two categories could be misleading. Therefore, we want to combine correlation coefficient with distance correlation together. We look for those with high distance correlation and high correlation coefficient. Again, due to the data size, we can't acquire very ideal results. Yet, we can see it is an approach worth our attention. An optimistic example is shown below.

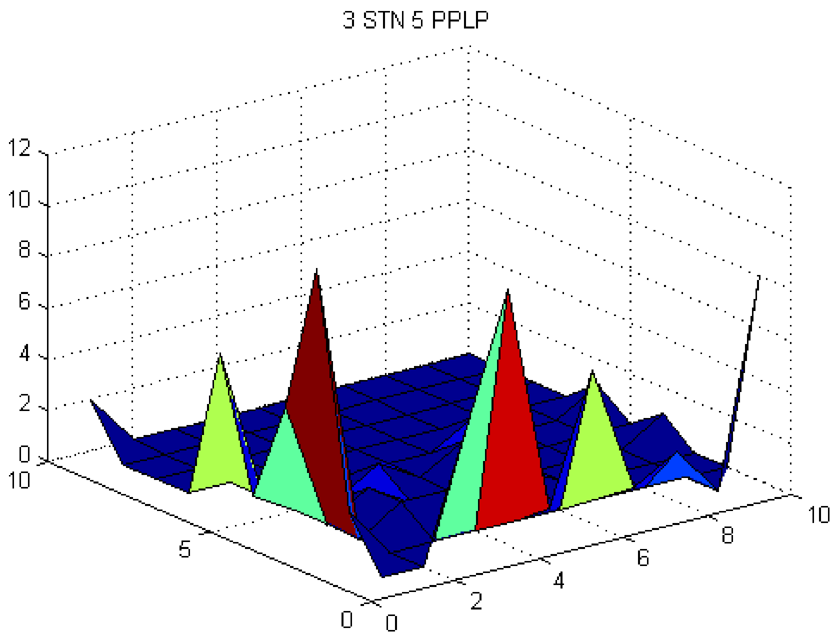


Figure 16. 2-d histogram of reviewer No.3 's *ST NEG* and No. 5's *PPL POS* (after elimination of double-zero component), not all the points are focused on the axis..

This time we have some points on the inner plain compare with everything focus on the axis. If we can get a better data size, we believe this approach can help us pick out some patterns between categories of different reviewers.

3.4.3 Dependency test

Although zero distance correlation implies independence, we want to know if the dependency between reviewers and categories are statistically significant enough. To achieve that, we use the distance correlation to perform hypothesis test on all our reviewers and categories to decide if they are statistically dependent. According to [9], **Theorem 1.** gives us the hypothesis test to reject independence:

Theorem 1. Suppose $T(X, Y, \alpha, n)$ is the test that rejects independence if

$$\frac{nV^2(X,Y)}{S} > (\phi^{-1}(1-\alpha/2))^2$$

where $= \frac{1}{n^2}\sum_{k,l=1}^{n}|X_k - X_l|\frac{1}{n^2}\sum_{k,l=1}^{n}|Y_k - Y_l|$, each X_k is the k-th 18 dimensional vector from reviewer X and $\phi()$ denotes the cumulative distribution function of standard normal distribution. Let $\alpha(X, Y, n)$ denotes the achieved significance level of $T(X, Y, \alpha, n)$.

We select two reviewers and their common set of movies. Then according to the theorem we compute the corresponding distance covariance and *S*. Finally, the hypothesis test is performed according to the equation above. We set two thresholds, α = 0.1 and α = 0.01. Unfortunately, the results come out to be independent between all reviewers across 18 dimensions for both thresholds.

	1	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0
8	0	0	0	0	0	0	0	1	0	0
9	0	0	0	0	0	0	0	0	1	0
10	0	0	0	0	0	0	0	0	0	1

Figure 17. The dependency matrix between all the reviewers for $\alpha = 0.1$. Not a single reviewer is dependent with other reviewers.

However, this is to be expected. After all, it is a big world. Two people could come from totally different background with totally different taste in movies. Also, we expect the movie type information may help us find some dependency relationship.

As always, we pick two reviewers and look at their categories for dependence. This time, we are lucky. We find some of the categories are dependent on some categories between two different reviewers, which means we can use the dependent categories to run the inference.

As we can see from the figure below. For each category from reviewer No. 3, we can find at least one category that is dependent with it. So if we want to use reviewer No. 3 to infer reviewer No. 5, we have at least one category from reviewer No. 3 that we can use for most of the categories of reviewer No. 5. And we can also find that since reviewer pair No.3 & No. 5 has higher distance correlation than reviewer pair No.1 & No.2, the number of dependent categories they have is also one times more than No.1 & No.2. This result also matches the previous 3 nn graph we constructed.

Though we didn't use just one category from one reviewer to infer one category from another reviewer, certain dependence is definitely valuable for inference. Next, we consider taking all the 18-dimensional data from one reviewer to infer another category from another reviewer. In this way, we include all the dependent categories. The results of inference will be introduce in the next chapter.

3 \ 5	OA-P	OA-N	ST-P	ST-N	CH-P	CH-N	VP-P	VP-N	MS-P	MS-N	SE-P	SE-N	PAC-P	PAC-N	PDR-P	PDR-N	PPL-P	PPL-N
OA-P	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OA-N	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ST-P	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
ST-N	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
CH-P	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
CH-N	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
VP-P	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1
VP-N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MS-P	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
MS-N	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
SE-P	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
SE-N	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0
PAC-P	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PAC-N	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0
PDR-P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PDR-N	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	0	1	0
PPL-P	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
PPL-N	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

1 \ 2	OA-P	OA-N	ST-P	ST-N	CH-P	CH-N	VP-P	VP-N	MS-P	MS-N	SE-P	SE-N	PAC-P	PAC-N	PDR-P	PDR-N	PPL-P	PPL-N
OA-P	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1
OA-N	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
ST-P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ST-N	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
CH-P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
CH-N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
VP-P	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
VP-N	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
MS-P	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MS-N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SE-P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SE-N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PAC-P	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
PAC-N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PDR-P	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
PDR-N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PPL-P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PPL-N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 18. The dependency matrix between reviewer pair No. 3 & No. 5 and No.1 & No. 2 across all 18 categories for $\alpha = 0.1$ (The red cells are dependent ones, the grey cells are two categories rarely mentioned by reviewers).

Chapter 4. Inference and Conditional Dependence

Our objective is find quantitative results from reviews of reviewers. The inference is certainly an important part of it. We use ordinary least square (linear) estimation to do the inference. From the results of last chapter, we know that only some of the categories between reviewers are dependent. In order to do the inference, we need to use the dependent categories. As a result, we choose the approach that leaves no information out by using all 18 dimensional data from one reviewer to infer a particular category of another reviewer. In this way, we include all the dependent pairs.

4.1 Inference using OLS

The method of least squares is about estimating parameters by minimizing the squared discrepancies between observed data, on the one hand, and their expected values on the other.

The objective of OLS consists of adjusting the parameters of a model function to best fit a data set. A simple data set consists of n points (data pairs). Consider an over-determined (more equations than unknowns) system,

$$\sum_{j=1}^{n} X_{ij}\beta_j = y_i \, , i = 1,2, \dots , m$$

of m linear equations in n unknown coefficients, $\beta_1, \beta_2, \dots, \beta_n$ with $m > n$. This can be written in matrix form as

$$\boldsymbol{X\beta} = \boldsymbol{Y}$$

where

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{m1} & X_{m2} & \cdots & X_{mn} \end{bmatrix}, \qquad \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

Such a system usually has no solution, so the goal is instead to find the coefficients β which fit the equations "best", in the sense of solving the quadratic minimization problem.

$$\hat{\beta} = \arg\min_{\beta} S(\beta)$$

where $S(\beta)$ is the objective function

$$S(\beta) = \sum_{i=1}^{m} |y_i \; - \sum_{j=1}^{n} X_{ij}\beta_j \,|^2 = ||\boldsymbol{Y} - \boldsymbol{X\beta}||^2$$

Using the definitions above we can use OLS to estimate one reviewer's comment based on another reviewer's historical data across all categories. Our implementation of OLS can be summarized into following steps:

Step 1: Convert each category value under a particular movie from a reviewer to the log-likelihood scale, i.e. $\ln(C_i)$, and if C_i = 0, we replace it by a big negative number, say, -50.

Step 2: We want to linearly estimate the log-likelihood value of a particular category, for example, $C_j(8)$ for the reviewer No.8 under the j-th category, we need to put say *N* movies reviewed by reviewer No. 3 into a $N \times 18$ matrix, where each row is the 18-dimensional vector from reviewer No. 3.

Step 3: Denote the estimation of $C_j(8)$ as $\hat{C}_j$ (8), for each estimation, we have:

$$\widehat{C}_j = \begin{pmatrix} \widehat{C}_j(8,1) \\ \widehat{C}_j(8,2) \\ \cdots \\ \widehat{C}_j(8,N) \end{pmatrix} = \begin{pmatrix} C_1(3,1), C_2(3,1), \ldots, C_{18}(3,1) \\ C_1(3,2), C_2(3,2), \ldots, C_{18}(3,2) \\ \cdots \\ C_1(3,N), C_2(3,N), \ldots, C_{18}(3,N) \end{pmatrix} \vec{\beta}_j$$

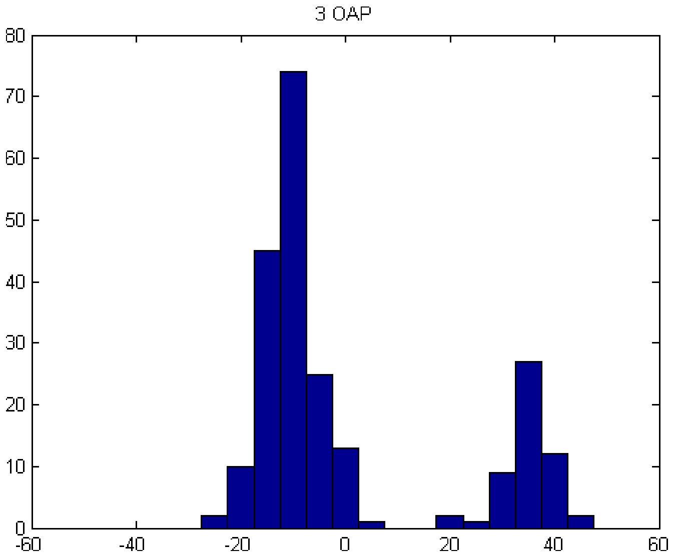
where *N* is the total number of common movies they commented on and $\vec{\beta}_j$ is a 18×1 vector. Our goal is to find such $\vec{\beta}_j$ minimizes the norm square between all estimations and real values,

$$\vec{\beta}_j = \arg\min_{\beta} \sum_{k=1}^{N} |C_j \ (8,k) - \hat{C}_j(8,k)|^2$$

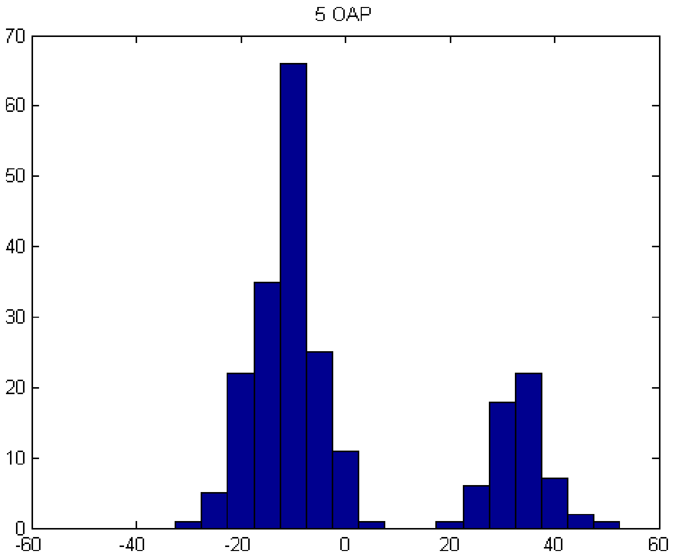
We perform the OLS on each category of one reviewer using their common movie set. And we can get the error matrix of each category using the following equation:

$$\boldsymbol{\varepsilon}_j = \boldsymbol{C}_j - \widehat{\boldsymbol{C}}_j$$

We perform OLS estimation using reviewer No. 3 and No. 5, from both directions (using No. 3 to infer No. 5 and the other way around). Then we compute the statistics of error vector $\boldsymbol{\varepsilon}_j$ for each category. It turns out the mean of $\boldsymbol{\varepsilon}_j$ is very small but the standard deviation is huge compare to the mean. Next we draw the histogram of the error vector to demonstrate the distribution. Below is the resulting histogram between reviewer pair No. 3 & No. 5. We pick two categories: *OA POS* and *PPL POS*, each has two directions: from No.3 to No. 5 and the other way around.

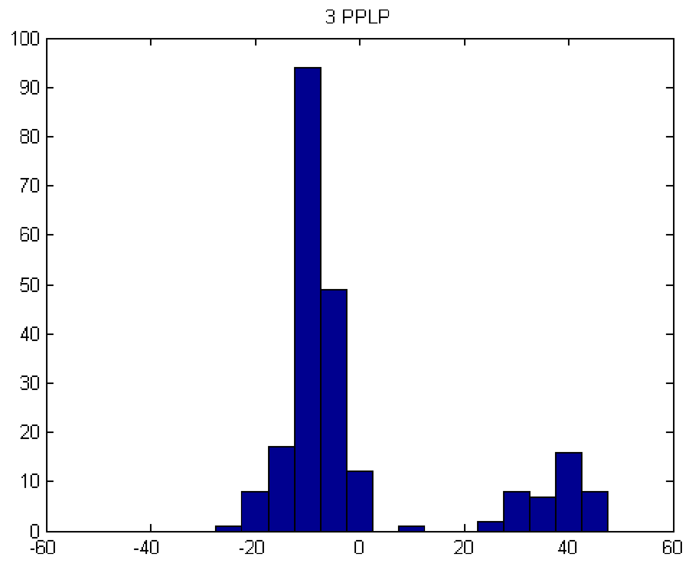


(a)

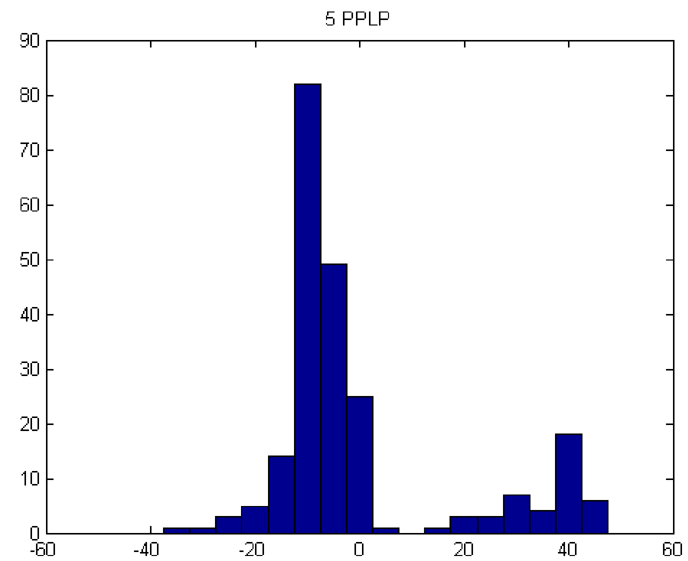


(b)

Figure 19. The error distribution (histogram) of reviewer pair No. 3 & No. 5. (a) using reviewer No. 5 to infer reviewer No. 3's *OA POS* (b) using reviewer No.3 to infer reviewer No. 5's *OA POS*



(c)



(d)

Figure 19 (continued). The error distribution (histogram) of reviewer pair No. 3 & No. 5. (c) using reviewer No. 5 to infer reviewer No. 3's *PPL POS* (d) using reviewer No. 3 to infer reviewer No. 5's *PPL POS*

We can see from the figure above the distribution is a multimodal distribution. It looks like a mixture of two Gaussian distribution. This is basically the same for other categories. Further analysis need to be done about the distribution of error vectors.

4.2 Conditional dependency test

After we get the error component using OLS, we can further perform conditional dependency test [10] between different categories for one reviewer. We want to use the results from conditional dependency test to construct a pair-wise conditional dependency graph. For example, when we estimate i-th and j-th category of reviewer No.3 using 18-dimensional data of reviewer No. 5. If category *i* and *j* are connected, we say they are conditionally dependent. Following are the steps towards that goal:

- Step 1: Compute the error component, $\boldsymbol{\varepsilon}_X^j$ and $\boldsymbol{\varepsilon}_Y^j$, where $\boldsymbol{\varepsilon}_X^j$ is the error component on category *j* of reviewer *X* using reviewer *Y*'s data equation:

$$\boldsymbol{\varepsilon}_X^j = \boldsymbol{C}_X^j - \widehat{\boldsymbol{C}}_X^j$$

- Step 2: Compute the empirical distance covariance between $\boldsymbol{\varepsilon}_X^j$ and $\boldsymbol{\varepsilon}_Y^j$.

$$a_{i,l} = ||\varepsilon_X^j(i) - \varepsilon_X^j(l)||;\ i,l = 1,2,\dots,n$$

$$b_{i,l} = ||\varepsilon_Y^j(i) - \varepsilon_Y^j(l)||;\ i,l = 1,2,\dots,n$$

where || || is the Euclidean norm. Then we have the $\mathrm{n} \times \mathrm{n}$ distance matrix $a_{j,k}$ and $b_{j,k}$ Take all doubly centered distances,

$$A_{i,l} = a_{i,l} - \bar{a}_{i.} - \bar{a}_{.l} - \bar{a}_{..}$$

$$B_{i,l} = b_{i,l} - \bar{b}_{i.} - \bar{b}_{.l} - \bar{b}_{..}$$

where $\bar{a}_{i.}$ is the mean of i-th row, $\bar{a}_{.l}$ is the mean of l-th column and $\bar{a}_{..}$ is the grand mean. Then we have the distance covariance as,

$$\mathrm{dCov}^2(\boldsymbol{\varepsilon}_X^j, \boldsymbol{\varepsilon}_Y^j) = \frac{1}{n^2}\sum_{i,l=1}^{n} A_{i,l}B_{i,l}$$

- Step 3: Compute the distance covariance $V^2(X,Y)$ and *S* using $\boldsymbol{\varepsilon}_X^j$ and $\boldsymbol{\varepsilon}_Y^j$,

where

$$S = \frac{1}{n^2}\sum_{i,l=1}^{n} |\varepsilon_X^j(i) - \varepsilon_X^j(l)| \frac{1}{n^2}\sum_{i,l=1}^{n} |\varepsilon_Y^j(i) - \varepsilon_Y^j(l)|$$

- Step 3: Rejects independence if we have,

$$\frac{nV^2(X,Y)}{S} > (\phi^{-1}(1-\alpha/2))^2$$

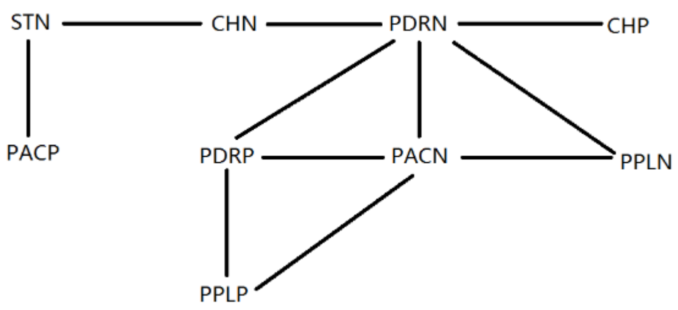
where $\boldsymbol{\varepsilon}_X^j$ is the error vector on category *j* from reviewer *X* and $\phi()$ denotes the cumulative distribution function of standard normal distribution. Let $\alpha(X,Y,n)$ denotes the achieved significance level of $T(X,Y,\alpha,n)$.

5 \ 5	OA-P	OA-N	ST-P	ST-N	CH-P	CH-N	VP-P	VP-N	MS-P	MS-N	SE-P	SE-N	PAC-P	PAC-N	PDR-P	PDR-N	PPL-P	PPL-N
OA-P		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OA-N	0		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
ST-P	0	0		0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
ST-N	0	0	0		0	1	0	0	1	0	1	1	1	0	0	0	0	0
CH-P	0	0	0	0		0	0	0	1	0	1	1	0	0	0	1	0	0
CH-N	0	0	0	1	0		0	0	1	1	1	1	0	0	1	1	0	0
VP-P	0	0	0	0	0	0		0	1	1	1	1	0	0	0	0	0	0
VP-N	0	0	0	0	0	0	0		1	0	1	1	0	0	0	0	0	0
MS-P	0	0	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1
MS-N	0	1	1	0	0	1	1	0	1	1	1	1	0	1	0	1	0	1
SE-P	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1
SE-N	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
PAC-P	0	0	0	1	0	0	0	0	0	0	0	1		0	0	0	0	0
PAC-N	0	0	0	0	0	0	0	0	1	1	1	1	0		1	1	1	1
PDR-P	0	0	0	0	0	1	0	0	0	0	1	1	0	1		1	1	0
PDR-N	0	0	0	0	1	1	0	0	1	1	1	1	0	1	1		0	1
PPL-P	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0		0
PPL-N	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	1	0	

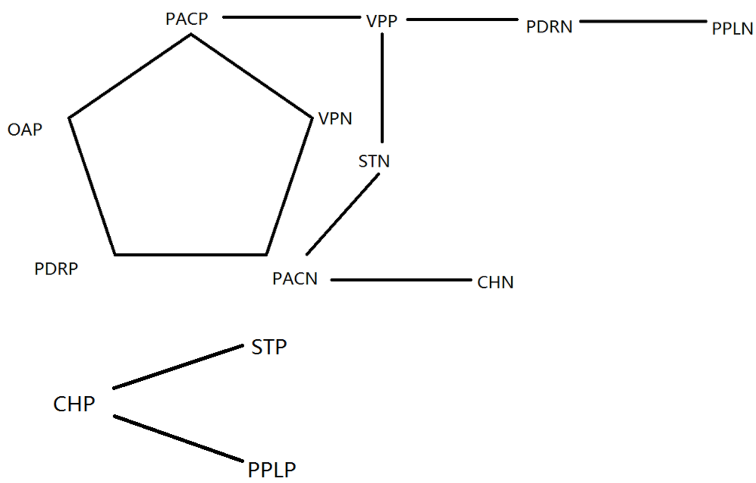
3 \ 3	OA-P	OA-N	ST-P	ST-N	CH-P	CH-N	VP-P	VP-N	MS-P	MS-N	SE-P	SE-N	PAC-P	PAC-N	PDR-P	PDR-N	PPL-P	PPL-N
OA-P		0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0
OA-N	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ST-P	0	0		0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
ST-N	0	0	0		0	0	1	0	0	0	1	0	0	1	0	0	0	0
CH-P	0	0	1	0		0	0	0	0	0	0	0	0	0	0	0	1	0
CH-N	0	0	0	0	0		0	0	0	0	0	0	0	1	0	0	0	0
VP-P	0	0	0	1	0	0		0	0	0	1	1	1	0	0	1	0	0
VP-N	0	0	0	0	0	0	0		0	0	0	1	1	1	0	0	0	0
MS-P	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
MS-N	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0
SE-P	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	0
SE-N	1	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0
PAC-P	1	0	0	0	0	0	1	1	0	0	0	0		0	0	0	0	0
PAC-N	0	0	0	1	0	1	0	1	0	1	1	0	0		1	0	0	0
PDR-P	1	0	0	0	0	0	0	0	0	0	0	0	0	1		0	0	0
PDR-N	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0		0	1
PPL-P	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0		0
PPL-N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	

Figure 20. Conditional dependency matrix of reviewer No. 5's categories using reviewer No. 3 to inference (above) and reviewer No. 3's categories using reviewer No. 5 to inference (below) α = 0.1.

The result shows dependency relationship between some categories of reviewer 5's categories (*PAC POS* and *ST NEG*). According to the matrix, we again construct a graph to describe the dependency relationship between categories. If there is an edge connect two nodes, it means they are dependent conditioned on the reviewer we used to infer.



(a)



(b)

Figure 21. The conditional dependency relationship graph of reviewer No. 5's categories conditioned on the data of reviewer No.3.

We can see from the graph not a lot categories are dependent with another conditioned on the data of another reviewer. But when it comes to people-related category, we can see some strong dependence conditioned on data from another reviewer.

4.3 Applications of discovered patterns

There are many ways we can utilize our discovered patterns, such as the multinomial distributions of one reviewer, the dependency graph and error distribution. In this section, we provide some possible applications using the patterns we've discovered.

- Identification of reviewer

  For each reviewer, we've already converted all its review texts into a matrix, which is a set of multinomial distributions. Essentially, we have a distribution of distributions for every reviewer. Using this, we can carry out the identification of reviewer. So each reviewer becomes a point in an 18-dimensional distribution space. Given a newly generated multinomial distribution, we can use SVM [17] to identify if the reviewer belongs to a previously seen reviewer.

- Prediction using error distribution

  For a pair of reviewer *X* and *Y*, we have

  $$\boldsymbol{X\beta} + \boldsymbol{\varepsilon} = \boldsymbol{Y}$$

  If we can model the error component's distribution, which is a two-mode distribution mentioned in previous section, we can use it to predict our reviewers' future review distribution. Given the historical data from two reviewers *X* and *Y* along with the review from reviewer *X* on movie *M*, we can predict the review of reviewer *Y* on the same movie by using the error component's distribution as well as the proposed OLS method.

- Common randomness extraction

  Given a pair of reviews X_i and X_j from reviewer i and j, respectively, their relationship under our proposed OLS model can be characterized by the following:

  $$\beta_{i,j}X_i + \varepsilon_{i,j} = X_j$$

  where $\beta_{i,j}$ is an 18 by 18 dimensional matrix, whose entries are determined using labelled data by using the reviews by the i-th reviewer to estimate those by the j-th reviewer. And X_i and X_j are two log-likelihood values of the two 18-dimensional multinomial distributions, and $\varepsilon_{i,j}$ is the resulting estimation errors. Given this directional linear inference model, we intend to extract common randomness from a set of such review pairs on reviewer i and j, using the methods proposed in [18][19]. We need a model characterizing the joint distribution of X_i and X_j, each of which is the log-likelihood function of a multinomial distribution of dimension 18, based on the empirical distribution of the error components learned from training data sets. Such joint distribution should satisfy two conditions: 1) The resulting marginal distributions of X_i and X_j derived from the joint PMF should agree with the empirical ones attained using data; 2) If we switch the between two reviewers, namely, using the review X_j to infer X_i, the resulting model should be compatible with the other direction. Such requirement is not going to be satisfied easily. We need to consider more advanced estimation models than OLS to satisfy the above two conditions in order to extract common randomness using the approaches in [18][19].

Chapter 5. Conclusions and Future Work

In this thesis, we present a novel approach to transform unstructured semantic data, online movie reviews to structured semantic data, review tuples. Then we further convert them into numeral data, reviewer vectors and multinomial distributions.

After the transformation, we run initial clustering of our chosen reviewers using the KL distance between profiles of each reviewer based on their own set of movies. Next, we run a similar clustering on all chosen reviewers using distance correlation based on the common reviewed movie set between a pair of reviewers. The dependency relation result comes out to be very different with the KL distance result. We also use 2-d histogram try to visualize the dependence we are seeking, but due to the size of data set, we are not able to pick out any patterns from the 2-d histogram. Dependency test is performed on all reviewers using statistics in distance correlation. Though none of the reviewers are statistically significantly dependent, some categories are dependent between two reviewers. Based on the result that there is dependence between categories of two reviewers. We perform the inference using OLS. With the distribution of error component, we give a few examples of possible applications with our system. The results prove this transformation of data can help us find certain patterns, and quantifiable results.

Some future work need to be done to further utilize the transformed data including, 1) Involve movie type information to discover patterns and to expand the data size. The movie type information is a key information. It stands for the taste and style of one reviewer. If we can add another element to represent the movie type information into our tuples, we will certainly be able to use it to locate more patterns; 2) Further analysis on the 2-d histogram. Expand the data size by searching through all the reviewers' CMI. Choose proper reviewers with large data size to run the 2-d histogram to overcome the problem of small data size; 3) Model the error component in the OLS estimation. With the modeling of error component in OLS, we can run prediction and identification of our reviewers; 4) Utilize the conditional dependency relationship for common randomness extraction.

References

[1] Zhuang, Li, Feng Jing, and Xiao-Yan Zhu. “Movie Review Mining And Summarization.” Conference On Information & Knowledge Management (2006): 43.

[2] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. “Thumbs Up? Sentiment Classification Using Machine Learning Techniques.” (2002): *arXiv*.

[3] Zheng, Xiaolin, et al. “Incorporating Appraisal Expression Patterns Into Topic Modeling For Aspect And Sentiment Word Identification.” *Knowledge-Based Systems* 61.(2014): 29-47.

[4] Hu, Minqing, and Bing Liu. “Mining And Summarizing Customer Reviews.” *Proceedings Of The Tenth ACM SIGKDD International Conference: Knowledge Discovery & Data Mining* (2004): 168.

[5] Manning, Christopher D., Surdeanu, Mihai, Bauer, John, Finkel, Jenny, Bethard, Steven J., and McClosky, David. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60

[6] Kullback, S.; Leibler, R. A. “On Information and Sufficiency.” *Ann. Math. Statist.* 22 (1951), no. 1, 79--86.

[7] Kullback, Solomon. *Information Theory And Statistics*. n.p.: New York, Wiley [1959], 1959.

[8] Kullback, S. “Letter to the Editor: The Kullback–Leibler distance.” *The American Statistician* 41 (4): 340–341.JSTOR 2684769.

[9] Székely, Gábor J., Maria L. Rizzo, and Nail K. Bakirov. “Measuring And Testing Dependence By Correlation Of Distances.” (2008): *arXiv*.

[10] Fan, Jianqing, Yang Feng, and Lucy Xia. “A Conditional Dependence Measure With Applications To Undirected Graphical Models.” (2015): arXiv.

[11] Leon-Garcia, Alberto, “Probability, Statistics, and Random Processes For Electrical Engineering”. (2008)

[12] Dodds, Peter Sheridan, et al. “Human Language Reveals A Universal Positivity Bias.” Proceedings Of The National Academy Of Sciences Of The United States Of America 112.8 (n.d.): 2389-2394. Biological Abstracts 1969 - Present.

[13] L. Zhou, P. Chaovalit, “Ontology-supported polarity mining”, J. Am. Soc. Inform. Sci. Technol. 59 (1) (2008) 98–110.

[14] L.-C. Yu, J.-L. Wu, P.-C. Chang, H.-S. Chu, “Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news”, Knowl.-Based Syst. 41 (2013) 89–97.

[15] Serrano-Guerrero, Jesus, et al. “Sentiment Analysis: A Review And Comparative Analysis Of Web Services.” Information Sciences 311.(2015): 18-38. ScienceDirect. Web. 15 May 2015.

[16] D. Wang, S. Zhu, T. Li, “SumView: a Web-based engine for summarizing product reviews and customer opinions”, Expert Syst. Appl. 40 (1) (2013) 27–33.

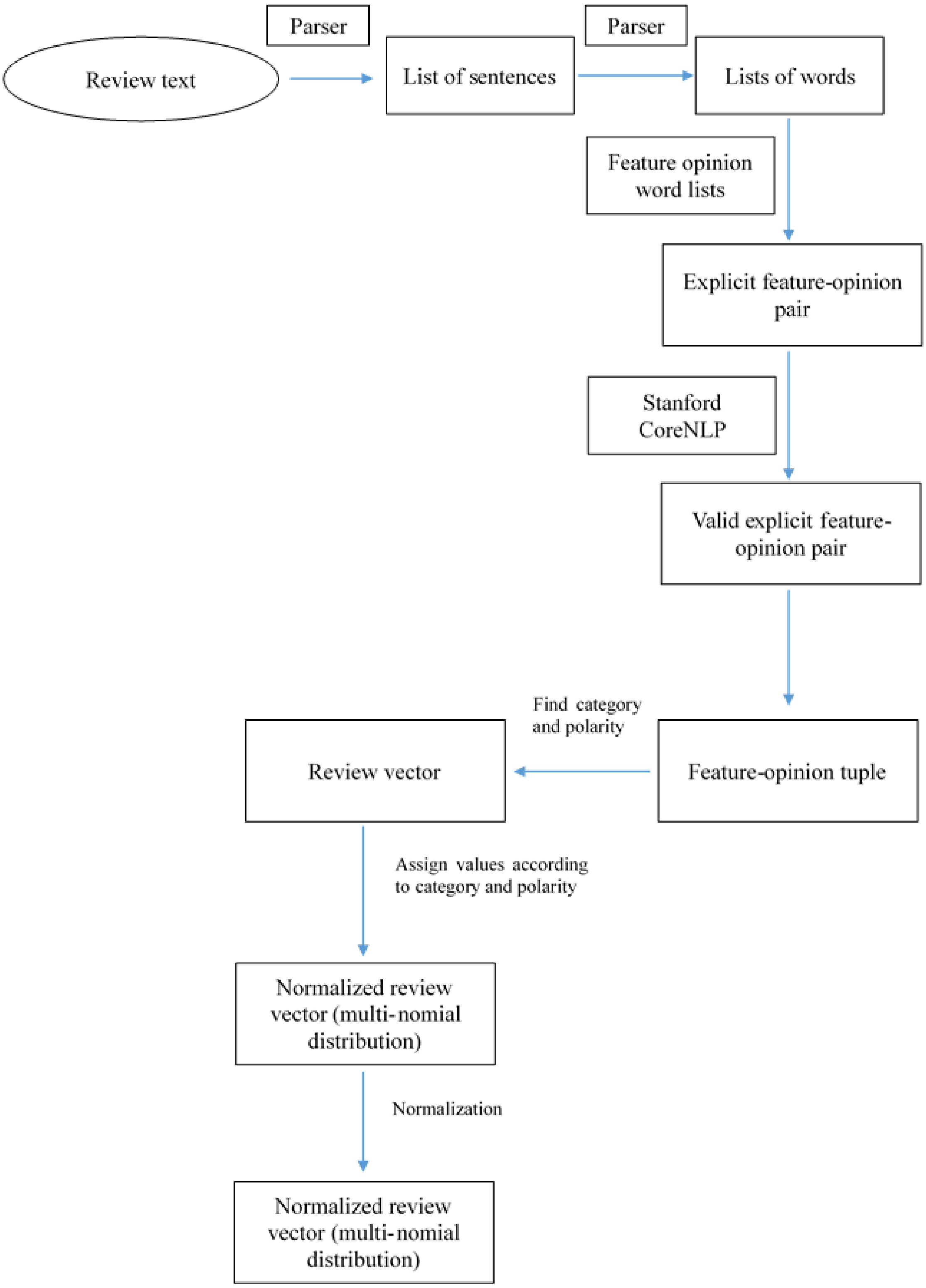
[17] C. Cortes, V. Vapnik, “Support-Vector Networks”, Machine Learning, 20, 273-297 (1995).

[18] U. E. Maurer, “Secret key agreement by public discussion from common information,” Information Theory, IEEE Transactions on, vol. 39, pp. 733–742, May. 1993.

[19] R. Ahlswede and I. Csiszar, “Common randomness in information theory and cryptography – Part I: secret sharing,” Information Theory, IEEE Transactions on, vol. 39, pp. 1121–1132, July 1993.

Appendix

The program flowchart of data transformation



Vita

Limeng Pu, an international student from China, received his Bachelor's degree at University of Electrical Science and Technology of China in Chengdu,China in 2013. Thereafter, he made a decision to pursue more advance knowledge in The School of Electrical Engineering and Computer Science at Louisiana State University. He will receive his master's degree in August 2015 and plans to begin work on his doctorate upon graduation.