Rerating the Movie Scores in Douban through Word Embedding

Mingyu Cui\textsuperscript{1,*}
\textsuperscript{1}School of Computer Science and Engineering, SouthEast University, Nanjing 211189, China

*mingyuc36@163.com

Abstract. The movie scores in the social networking service website such as IMDb, Totten Tomatoes and Douban are important references to evaluate the movies. Always, it will influence the box office directly. However, the public rating has strong bias depended on the types of movies, release time, and ages and background of the audiences. To fix the bias and give a movie a fair judgement is an important problem. In the paper, we focus on the movie scores on Douban, which is one of the most famous Chinese movie network community. We decompose the movie scores into two parts. One is the basis scores based on the basic properties of movies. The other is the extra scores which represent the excess value of the movies. We use the word-embedding technique to reduce the movies in a small dense subspace. Then, in the reduced subspace, we use the k-means method to offer the similar movies a basis scores.

1. Introduction
The social networking service websites such as IMDb, Totten Tomatoes and Douban are an important approach for common people to express the point of view of the movies. The final movie scores are the average of all personal ratings. It is difficult to cheat through bribing a small group of movie experts or critics. Thus, it is always deemed to be a fair way to evaluate the values of movies. However, most of the people know little professional standard to judge a movie. Since they provide the scores just followed their hearts, the public rating has strong bias depended on the types of movies, release time, and ages and background of the audiences. It is usually unfair to compare a commercial film to a literary film directly. Sometimes, it is also difficult to compare an action movie to a documentary. On the other hand, people are unlikely not see a small minority movie which they are not interested in. It means that if one person has watched a small minority movie, then with high probability, they would give it a high score. On the contrary, most people have the experience to see popular movies even do not know the movie title clearly.

Thus, it is important to provide a fair evaluation. Professional accreditation, such as Oscar is one approach. But usually, it is unrelated to the public rating. In the paper, we provide an alternative way to give a fair evaluation totally dependent on the public ratings themselves. Briefly speaking, we decompose the movie scores into two parts. One is the basis scores based on the basic properties of movies. The other is the extra scores which represent the excess value of the movies.

We use the word embedding technique to reduce the movies in a small dense subspace. The movies with similar basis score are located in the close locations in the subspace. Thus, we can use the (scaled) average scores among its neighbors as its basis scores. Then, according to the real scores of the movie,
we can obtain is extra scores. After many attempts, we conclude a formula to calculate the real score of a movie. Formally speaking,

\[ \text{Real Score} = \alpha \times \text{Basis Score} + (1 - \alpha) \times \text{Extra Score} \ldots \]

Word embedding is the collective techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers. Representation of words as embedding vectors has a long history. Bengio et al. [1] provided a feedforward neural network with a linear projection layer and a non-linear hidden layer to learn jointly the word vectors representation. Rowes and Saul [2] published in Science how to use "locally linear embedding" (LLE) to discover representations of high dimensional data structure. In 2013, Tomas Mikolov et al.[3] [4] introduced the Continuous Bag-of-Words model (CBOW) and the Skip-Gram model which achieved large improvements in accuracy at much lower computational cost compared to the previous methods.

Based on the methods, they created word2vec, a toolkit which is particularly computationally-efficient predictive model for learning word embedding. Although considerable research has been devoted to the sentiment analysis of the linguistic contexts, rather less attention has been paid to use word embeddings to analyze the potential laws in movies. In the paper, we focus on the movie scores on Douban, which is one of the most famous Chinese movie network community. We randomly choose 1000 movies from Douban. Collect the scores, basic informations such as directors, editors, actors and types, and film reviews of these movies. Based on the data, we use word2vec to construct the embedding vectors. Then, according to the labels such as actors and movie types, we embedding the movies in a small subspace. Finally, we applied K-means to cluster movies in the space and calculate the basis scores for each one.

2. Methodology

2.1. Embedding the Movies in Small Spaces.

As we all know, how to represent the precise features of the data is a key problem in machine learning. In audio, we can use the matrix which contains the audio spectrums to solve the problem and we can use some distance and rate to judge whether there exist similarities between two audios. While in Natural Language Processing (NLP), all we can use are words which only consist of characters. Therefore, how to express a word and judge the similarities between two words in program are important.

One-hot representation is one of the easiest way to present words which are to use a long vector to stands for a word and the length of the vector is the number of all words. In that vector, we use 1 to represent the word and the rest of elements in the vector are 0. In fact, this method can solve many tasks but the dimension of the vector will increase if the number of total words increases. The representation is extreme sparse which will waste many spaces. Besides, any two vectors are independent which means that we can’t obtain semantic information from any two vectors.

Therefore, reduce the one-hot vectors in a small, dense subspace is a key problem. We call the new representation distributed representation and call the process word embedding. In order to avoid the disadvantages brought by one-hot representation, the word embedding changes data types from integer to float and pushes the big dimension vectors into a smaller dimension vectors. There are three main ways to achieve distributed representation which are respectively based on matrix [5], clustering [6] and neutral network [7].

The neutral network is the most efficient way for word embedding. The basic network is illustrated in Figure 1. See [3] [4] for more details.

Word2vec, a computationally-efficient toolkit for learning word embeddings, involves statistical language model, CBOW model and Skip-gram model. The statistical language model is to use probability to judge whether a sentence is reasonable which means that calculating the probability that a term appears in the premise of some given words. And the CBOW model and Skip-gram model both contains three layers which are input layer, projection layer and output layer. CBOW model is to
predict a possible word in the premise of some context of this word. And Skip-gram model is a reverse process of CBOW.

When we are training this model, we can obtain the word vectors which means that word vectors are the by-product of training language neutral network model.

![Figure 1: The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.](image)

2.2. Compute the Basis Scores

The distributed representation can retain the semantic information of a word. We assume that the information of movie can be executed by distributed representation and the information of movie is saved to some extent and by this way, we can revise the point of movie according to the labels it has. That is to say if a director shot a movie and received a high evaluation, by clustering all the embeddings of director, we can obtain some groups that only contain the excellent director who shot a good movie once and some groups that only contain the director who had a bad performance once. Then, we can obtain a point of movie according to the director label. Therefore, we want to find out whether a good movie still has a good performance in our model and revise the point of movie using the word embeddings.

Following the idea, we maintain several hash-tables to map the movies to the word vector spaces. In each space, we group the movies in several clusters. For each movie in any cluster, we use the average scores of the movies in the cluster as one components of the basis score of the movie. Denote the number of spaces as m. Then, following the process, each movie will have m components of basis scores. We use the average scores of the components as the final basis scores of the movies. Then, we discuss the details of the hash tables and the cluster algorithm.

| Name               | Director | Editor    | Actor       | Type   | Point |
|--------------------|----------|-----------|-------------|--------|-------|
| *Farewell My Concubine* | Kaige Chen | Pik Wah Lee | Leslie Cheung | Drama  | 9.50  |

In the paper, we construct four hash mappings, i.e., m = 4. Concretely speaking, they are (movie - the word vector of director), (movie - the word vector of editor), (movie - the word vector of actor), (movie - the word vector of type). See Table1 for an example. Note that, here the word vector is the word-embedding vector of the corresponding word obtained through word2vec.

The cluster algorithm used in the paper is K-means algorithms. The K-means algorithm is to regard the average data of every subgroup as the representative point and obtains some groups which they are intra-class compactness and inter-class separability. Formally speaking, the algorithm clusters the data set into different groups by iterating and minimize the within-cluster sum of squared criterion function as follows.
argmin_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 

... (2)

Where $\mu_i$ is the mean of points in $S_i$.

2.3. Experiments
The problem was solved in two main steps. At first, we used word2vec to train the information of movies and obtained word embeddings of each label. The contexts of movies contain the name, director, editor, type and 5 short comments of each movie. What’s more, we applied a word segmentation tool to do segmentation. After word embedding, we obtained 28000 word vectors approximately which are partly shown in Figure 2.

After that, we applied the K-means algorithm to cluster the word embeddings to ten groups according to the label of word embeddings. As to the director label, the point of one group was calculated by calculating the average point of the total movies in this group. And the rest of the groups were also calculated by this method. After the whole process, the four labels of each movie have a new point. Then, we applied Alpha Coefficient to calculate the comprehensive point of each movie and after many attempts we set Alpha=0.7 to calculate real score. There is an example about the executed information of a movie shown in Table 2. Finally, all we obtained that the information and the real scores of movies are shown in Table 3.

![Figure 2. Example of embedding words.](image)

**Table 2.** Example of the executed information of a movie.

| Name           | Director Of Point | Editor Of Point | Actor Of Point | Type Of Point | Point(Basis score) | Point(Extra score) | Real Score |
|----------------|-------------------|-----------------|----------------|--------------|--------------------|--------------------|------------|
| Farewell Concubine | 7.38              | 7.30            | 7.56           | 7.57         | 7.45               | 9.50               | 8.88       |

**Table 3.** Example of executed movies which contains the new point of four labels and final point.

| Name           | Director of Actor of Type of Point(Real) |
|----------------|------------------------------------------|
|                |                                           |
Point | Point | Point | Point | Basis score) | score) | Score
---|---|---|---|---|---|---
Life is beautiful | 7.58 | 7.45 | 7.32 | 7.57 | 7.48 | 9.5 | 8.89
Farewell Concubine My | 7.38 | 7.3 | 7.56 | 7.57 | 7.45 | 9.5 | 8.88
Laputa: Castle in the sky | 7.42 | 7.39 | 7.46 | 8.41 | 7.67 | 9.1 | 8.60
The Matrix | 7.42 | 7.39 | 7.59 | 6.71 | 7.27 | 8.8 | 8.34
X-men | 7.58 | 7.45 | 7.32 | 6.71 | 7.26 | 7.6 | 7.49
The void | 5.80 | 5.80 | 5.80 | 6.83 | 6.05 | 5.8 | 5.87

3. Conclusion
In the paper, it is now possible to state that in our experiment a movie that possesses a high point in DOUBAN after word embedding still possesses a high point and a movie that does not have a good performance in DOUBAN still has a terrible performance in our study. In our process of revising the point of movie, our experiment results are consistent with the movie point form DOUBAN. Therefore we can assure that the word embeddings about the information of movies still contains semantic information of movies, which proves that analyzing movies by distributed representation method is feasible.

Our study shows that the mechanism of DOUBAN is convincing and word embeddings can be applied to correct the semantic information. In the future, it would be interesting to analyze which director is a good director or he always shoots good movies. Similarly, we can analyze which actor is a good actor or he always presents to us good movies.

References
[1] Bengio, Yoshua, et al. "A neural probabilistic language model." Journal of machine learning research 3.Feb (2003): 1137-1155.
[2] Roweis, Sam T., and Lawrence K. Saul. "Nonlinear dimensionality reduction by locally linear embedding." science 290.5500 (2000): 2323-2326.
[3] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv: 1301.3781 (2013).
[4] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.
[5] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation[C]/Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.
[6] Turian J, Ratinov L, Bengio Y. Word representations: a simple and general method for semi-supervised learning[C]/Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 2010: 384-394.
[7] Brown P F, Desouza P V, Mercer R L, et al. Class-based n-gram models of natural language [J]. Computational linguistics, 1992, 18(4): 467-479.