Customized Movie Recommendation for Users

Author(s)
Angnuo Li
Xuanyi Li
Mingxin Lu

Abstract

This paper provides a method to predict rating score of a certain user to a movie based on the user's historical reviews and the reviews for the movie from other users. This kind of NLP task is the strength of neural network. Thus, a single network is used to model data, discover the potential connection between users and movies. After prediction of rating score, recommend movie with high score for a certain user.

1 Introduction

There exists tremendous amount of reviews of rating score in an online shopping website or website for product reviews like yelp, IMDB, rotten tomatoes. When browsing this kind of websites, a user could get two types of information: The overall rating score, and detailed chosen reviews. The main limitation for using those informations to find fitted movie is from that due to provided information is based on all viewers experience rather than user himself, directly browsing website can't satisfy individual's taste and cost much time. We are trying to find a model could address the limitation. We will provide a rating score based on user's personality and movie's reviews:

- Two kinds of data for us to collect for the model's accomplishment:
  Users' review vectors $v_1, v_2, v_3..v_m$.
  Movies' review vectors $v_1, v_2, v_3..v_n$.
- Final output: The prediction for a certain user's rating score for a certain movie.

The ideal solution should consider the interruption between received information, training model, and thus give reliable output. The significant challenge is to investigate semantic meaning of review. Deep learning could connect multi-hidden layers to abstract information and complete relative calculation. Thus, it should be an effective tool for us.

2 Related Work

2.1 Comparison to other methods of semantic analysis

Our task is to extract, or in another word, infer sentimental polarity from raw document. Recently, various neural network models are used to capture the sentimental information automatically like convolutional neural networks, recursive neural network and recurrent neural network. The main difficult for semantic analysis is the difference between expression of a certain word for different users. Comparing to other sentimental analysis, which focus on prediction of existing reviews' opinions, our work would give a predicted score for a un-known products. And meanwhile previous researches aim to predict opinions for reviewed products, while our task is to recommend opinion on new products, which the user has not reviewed.

2.2 Reviews Summarization

This is a typical task of natural language prediction(NLP). Most of it work is to extract related words or phrases of users' opinion. Typical method includes using association to find frequent itemset as aspect terms, sequential labeling based method, and topic modeling techniques. Recently, word embedding and recurrent neural networks are becoming a frequently used technique to extract aspect terms.

Extraction of aspect terms is hard to reflect a certain aspect's contribution for the rating result. To solve it, researcher extract and sorted words based on its reliability and readibility. Then create a relative vector which could represent the feature of the text. The graph-based methods and attention-based neural network model is used to absorb information from text units and generation reliable representation vector of the text.

We will use attention model to summarize plain text of review(Srivastava et al. [2014]). We represent the reviews into fixed length vector as presentation.

2.3 Neural Network Model

Dynamic memory network models could apply for multiple types of NLP tasks, especially useful to extract semantic representation of texts, which are consistent with our need. We use Gated Recurrent Unit(GRU) to model the sequential information. (Bahdanau et al. [2014])

3 Model Construction

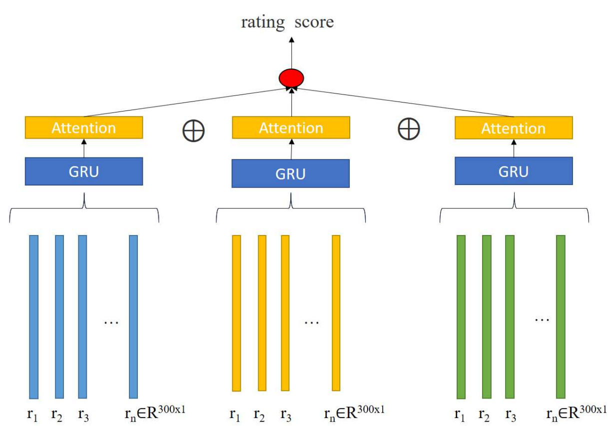

Figure 1: construct multiple-input model

3.1 Review Model Applied to Users and Movies

We construct vectors for each user and each movie by modeling all the reviews of the users. Average word embedding is used to model a review. A review could be represented as $r = \{x_1, x_2, \ldots x_m\}$,$x_i$ represents a word of the review. And the word x_i could be represented as a 300-dimensional embedding e_k^w.To represent the review, we could use the average of the m word's vector.

$$e_r = \sum_k e_k^w / m \tag{1}$$

Then, GRU skill is used to learn the hidden-states of the review. we obtain a sequence of hidden state vectors$\{h_{r_1}, h_{r_2}, h_{r_3} \ldots h_{r_n}\}$, represent the k-th step of the hidden state. We could get the sequence by feeding the word embedding vector:$\{e_{r_1}, e_{r_2}, e_{r_3} \ldots e_{r_n}\}$:

$$h_{r_t} = GRU(e_{r_t}, h_{r\ t-1}) \tag{2}$$

Considering that each reviews have unequal contribution for the user's(movie's) model construction(For example, due to time order), we use an attention mechanism to normalize the reviews' importance. Using $\{h_{r_1}, h_{r_2}, h_{r_3} \dots h_{r_n}\}$ as input, weighted sum of each hidden state is:

$$v = \sum_{k}^{n} a_i h_{r_i} \tag{3}$$

We could use the following scoring function to calculate as a a_i.

$$u_i = tanh(W_r h_{ri} + b_r) \tag{4}$$

$$\alpha_i = \frac{\exp{(u_i)}}{\sum_j \exp{(u_i)}} \tag{5}$$

The W_r and $b_r$are model parameters. Each hidden state has different weight for the vector's generation. Thus, we get the generated user vectors v_{user_i} and movie vectors v_{movie_j} to represent their reviews state.

3.2 Finding neighbor users

To construct a reliable model, we will import a neighbor user part. We need to find the neighbor users based on the neighborhood reviews vector by Collaborative filtering algorithm.

First, we need a measurement to compare the distance. Pearson correlation coefficient is a measure of the degree of fit between two sets of data. It revised "exaggerated scores", which means when the data is not totally standard and apparently exists deviation, (For example, the critics' evaluation of movies frequently differed from the general review. Correlation coefficient is used to measure similarity.

$$r(XY) = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{(\sum X^2 - \frac{(\sum X)^2}{N})(\sum Y^2 - \frac{(\sum Y)^2}{N})} \tag{6}$$

The best match reviewers are returned from the dictionary that reflects the preference. The number of returned reviewers is parameter. The list is sorted and the person with the highest evaluation value is ranked first. Thus we choose the neighbor users.

3.3 Generation of the Rating Score

To extract rating score, three elements should be considered: user model, neighbor users model and movie model. We combine all them together and use a neural network to produce a reliable rating score.

$$Y_S = W \times (v_{users} \oplus \sum_{i=1}^{k} v_{neighborusers} \oplus v_{movie}) \tag{7}$$

The W is model parameters which will be generated by neural network.

3.4 Training Process

To review scoring result, we need to get the loss function first. The loss function is defined as:

$$L(\theta) = \sum_{i=1}^{N} (Y^*_{S_i} - Y_{S_i})^2 + \frac{\lambda}{2} ||\theta||^2 \tag{8}$$

Standard back propagation is performed to optimize parameters using a variant gradient descent algorithm called Adam(Kingma and Ba [2014]). For all GRU models, we empirically set the size of the hidden layers to 128. We utilize the pre-trained word embedding of 300 dim provided by Google. In order to avoid over-fitting, dropout (Chung et al. [2014]) is used with a ratio of 0.2. The neighbor similarity threshold is set to 0.25.

4 Experiment

4.1 Experiment Setting

Table 1: dataset information

movies	772
users	906
reviews	20,000

Table 2: Basic Experiment information

Word Embedding Dim	300
Dropout	0.2
Similarity Threshold	0.25
Hidden State	128

4.2 Experiment Result

The training data from users is combined with two parts:1. Users vector. 2 Neighbor users vector. To understand the influence of neighbor users vector to the experiment result, we trained data twice. First time, we don't combine the neighbor users vector to the group. And combine it in the second time. Here is the sample of an individual user's result in differant model.

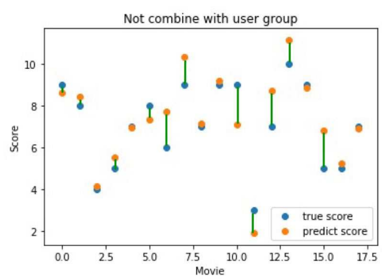

Figure 2: sample without neighbor users vector

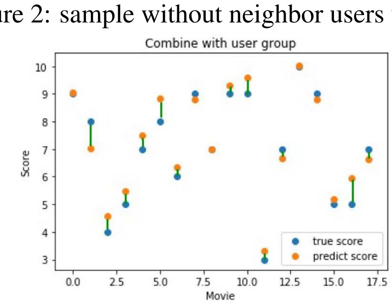

Figure 3: sample with neighbor users vector

Figure 4 and 5 could reflect overall train loss (blue line) and test loss(orange line) in training process. The full marks of rating score is 10. And by comparison with prediction without neighbor users, the neighbor users model could work.

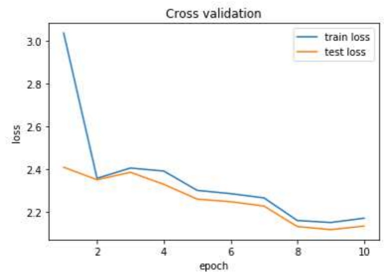

Figure 4: test loss without neighbor users

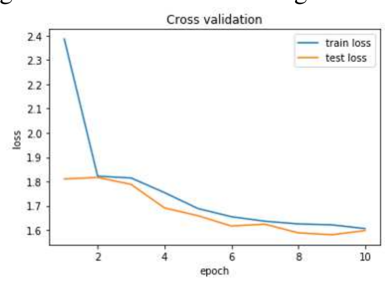

Figure 5: test loss with neighbor users vector

Table 3: test loss

test loss without neighbor users	1.8
test loss with neighbor users	1.5

Conclusion

A dynamic memory model is used for us to accomplish movie recommendation, by generation of rating score for a certain user. Deep memory network model used to find the connection between website users and movies. Results also shows that the model of neighbor users have positive effect on the preciser rating scores' generation.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1): 1929–1958, 2014.