ECEN765 Course Project Midterm Report

Name: Ziwei Zhu UIN:325000431

I. Progress Schedule

1. Week 1-2 (10/01-10/14):

   Plan: Literature search and study.

   Done: I read and studied several papers [1, 2, 3, 4] related to the recommender system using neural networks and deep learning techniques. I also read the paper [5] introducing the MovieLens dataset which I will use in the experiment. After I got familiar with the basic idea of recommender system, I decided to further research on the method of Restricted Boltzmann Machines for Collaborative Filtering for movie recommendation.

2. Week 3-5 (10/15-11/4):

   Plan: Implement one or two algorithms based on the papers read.

   Done: From my literature research, I learned that the method proposed by Salakhutdinov [1] is a very famous one which uses Restricted Boltzmann Machines (RBM) to do Collaborative Filtering. This model has a very good performance, and more important, it can handle very large datasets, which is a big problem for traditional Collaborative Filtering. However, the problem of this algorithm is that it can only model the correlation between items or between users, which ignores the correlation crossing users and items. Based on this idea, Georgiev [2] developed a user-item-based RBM to further improve the performance. I decided to implement these algorithms and analyze the results or further improve them.

   During these three weeks, I played with the MovieLens 1M dataset, and did some data preprocessing jobs using Python, including droping the useless attributes in the original file, generating the Matlab readable csv file for the ratings data, encoding the categorical attributes into numbers, extracting years of the movies from their name strings, and creating movie genres attributes.

   Then based on the papers [1, 2], I implemented the single-object-based RBM in Matlab. I developed two versions of the model, the first one uses the structure of networks shown in Figure 1, the second one uses the structure of networks shown in Figure 2. I have uploaded the code to my GitHub.

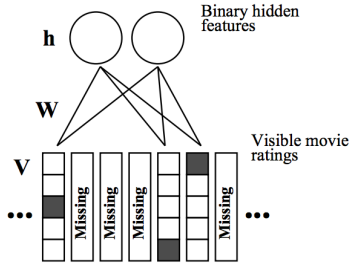

Figure 1 Use softmax visible units and binary hidden units

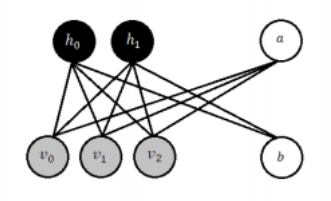

Figure 2 Use Gaussian visible units and binary hidden units

3. Week 6 (11/05-11/10):

Plan: Analyze and summarize the preliminary results. And write the midterm report.

Done: The preliminary results of the experiments of run the implemented models on the MovieLens 1M dataset are shown in Figure 3, where I set the number of latent units as 100 for both models. The mean absolute error for model 1 is 1.08, and the error for model 2 is 0.78. The Figure 4 and 5 show the training error over all iterations. As we can see, the model of using Gaussian visible units and binary hidden units can provide best performance. However, model 1 need very few iterations to converge.

During this week, I also wrote the midterm report.

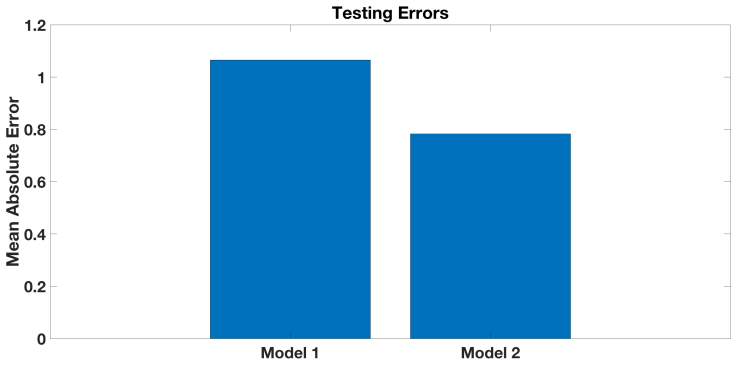

Figure 3 Testing for the two models

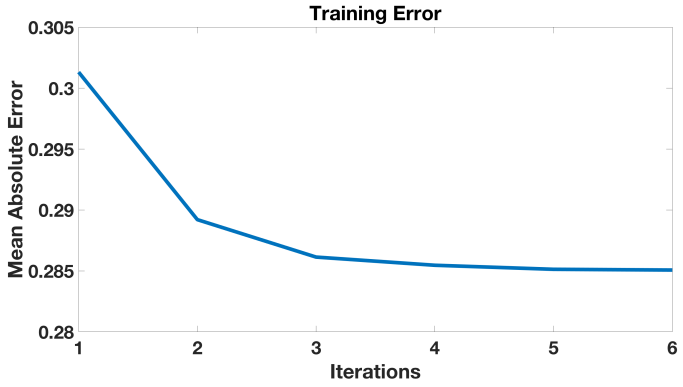

Figure 4 Training error for model 1

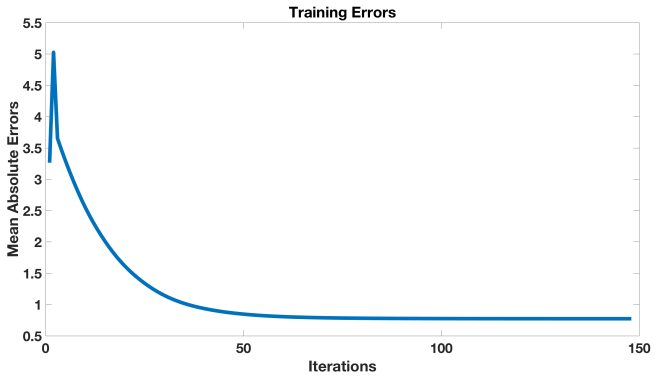

Figure 5 Training error for model 2

4. Week 7-8 (11/11-11/25):

   Original Plan: Test the implemented algorithms on real-world datasets, and analyze the empirical results.

   Current Plan: Based on the paper [2], implement the user-item-based RBM and do the experiment to test the performance.

5. Week 9-11 (11/26-12/13):

   Original Plan: Prepare the presentation and write the final report.

   Current Plan: Try to find out a way to apply the auxiliary information about the users (age and occupation) and the movies (year and genres) into the user-item-based model to improve the performance. Do experiment and analyze the results. Prepare the presentation and write the final report.

II. Existing Challenges

1. **Challenge**: The model with the structure of Figure 1 has a very slow runtime speed. It is because for each movie, the model uses five units. If there are 6000 movies, and we use 100 hidden units, then there will be 6000 * 100 * 5 = 3000000 parameters needing to be trained. It takes a long time for each iteration to update the large size weights vector.
   Plan: Implement a mini-batch version of algorithm which just uses a batch of data to update the weights in each iteration. This method can drop the time used for each iteration. However, I cannot make sure whether it can save a lot of time overall, and I am not clear about whether the performance of the trained model will be influenced.
2. **Challenge**: Although, I implemented the algorithm and can achieve a good empirical result, I do not really understand the theory of RBM, especially for how to derive the updating rule for the weights in the networks.
   Plan: I plan to search and read more papers about the basic RBM to study and understand the theory.
3. **Challenge**: I want to introduce the movies and users' information into the recommender system to improve the performance. The motivation is that the sparsity of the user-movie rating matrix is very high, and some users only have few rated movies, which makes it very hard for the model to learn their latent features. With the movies and users' individual information, the system can treat these situations with little history data better. The challenge now is how to integrate the auxiliary information into the existing model.

Plan: I will first try to treat the auxiliary the same as the rating scores use the same way to train the model to analyze the experimental results. Then I plan to try several modified structures to see the performance. Furthermore, I plan to deeper research into this problem after I systematically study the RBM theory as I said in II 2.

III. Changes from the Proposal

1. In the proposal, I plan to implement some deep-learning based recommender algorithms. However, after my literature research and study, I found that RBM based recommender system is the first case which uses neural networks for recommending. And it can astonishingly provide very good performance. Therefore, I switch my focus onto the RBM collaborative filtering models.
2. In the proposal, I listed several available datasets for the project. Now, I just selected the MovieLens 1M dataset. However, the implemented models can also be applied to the other datasets with very little modification.
3. In the proposal, I just introduced my intention at a very high level. After the six-week work, I specified the project on implementing user-based, item-based and user-item-based RBMs for collaborative filtering. Moreover, I plan to integrate the movies and users' auxiliary information, including movie years and genres, user occupations and ages, into the existing models to further improve the performance.

IV. Reference

[1] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM, 2007.

[2] Kostadin Georgiev and Preslav Nakov. A non-iid framework for collaborative filtering with restricted boltzmann machines. In *International Conference on Machine Learning*, pages 1148–1156, 2013.

[3] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

[4] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. A neural autoregressive approach to collaborative filtering. *arXiv preprint arXiv:1605.09477*, 2016.

[5] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2016.