Education4You – The Deep Learning Based Faculty Recommendation System

Abhinay Kumar Jha [cu.16bcs1147@gmail.com](mailto:cu.16bcs1147@gmail.com)

*Student, Dept. of Computer Science and Engineering, Chandigarh University, Gharuan, Mohali, Punjab 140413, India*

**Abstract**

In now a day the online services are the revolutionary step in order to communicate between the real worlds from a single place. It is not limited within any organization but in fact it is spreaded around each corner of our society. It is typically used in order to provide the virtual communication between us and real world and can be compared to other form of services like going to shops for purchasing goods, going to theatre to watch movie, similarly going to electronics shop to purchase music cassettes, etc. Recently, it has also become popular in order to use by government for its various bodies. Some online service provider algorithm provides only the related recommendation for the particular person based on the search perform by them on websites. In order to modify the little breach or shortcomings in these algorithms I design my product to provide more efficient and useful scenario of recommendation of related result, people search for in relation with the popular products world are seeking for with better interface so that the user got some ease of use and connect efficiently without any fluctuation with real world.

1. **Introduction**

A recommendation system is a type of information filtering system which attempts to predict the preferences of a user, and

make suggests based on these preferences. There are a wide variety of applications for

recommendation systems. These have become increasingly popular over the last few years and are now utilized in most online platforms that we use. The content of such platforms varies from movies, music, books and videos, to friends and stories on social media platforms, to products on e-commerce websites, to people on professional and dating websites, to search results returned on Google.

Often, these systems are able to collect information about a user’s choices, and can use this information to improve their suggestions in the future. For example, Facebook can monitor your interaction with various stories on your feed in order to learn what types of stories appeal to you. Sometimes, the recommender systems can make improvements based on the activities of a large number of people. For example, if Amazon observes that a large number of customers who buy the latest Apple Macbook also buy a USB-C-toUSB Adapter, they can recommend the Adapter to a new user who has just added a Macbook to his cart. Due to the advances in recommender systems, students constantly expect good recommendations. They have a low threshold for services that are not able to make appropriate suggestions. If a music streaming app is not able to predict and play music that the user likes, then the user will simply stop using it. This has led to a high emphasis by tech companies on improving their recommendation systems. However, the problem is more complex than it seems.

Every user has different preferences and likes. In addition, even the taste of a single user can vary depending on a large number of factors, such as mood, season, or type of activity the user is doing. For example, the type of music one would like to hear while exercising differs greatly from the type of music he’d listen to when cooking dinner. Another issue that recommendation systems have to solve is the exploration vs exploitation problem. They must explore new domains to discover more about the user, while still making the most of what is already known about of the user. Two main approaches are widely used for recommender systems. One is content-based filtering, where we try to profile the user’s interests using information collected, and recommend items based on that profile. The other is collaborative filtering, where we try to group similar students together and use information about the group to make recommendations to the user.

* 1. **Problem Statement**

With the increase in number of technologies we are getting more advanced day by day. But we are also lacking somewhere like in the education field. The students are not be able to get the solution of their problem in mean time when they caught any problem whose solution they need immediately.

The online query portal like “Quora”, “StackOverflow”, etc. gives efficient solution without any doubt but to get solution from this we need to post the problem and we get the solution after 24hours. And again if we not able to understand the problem then the query resolution process takes a several recursive process which causes many difficulties to students.

1. **Related Work**

I uses the Restricted Boltzmann Machine(RBM) for collaborative filtering in order to build my recommendation system.

* 1. **Explanation of CF**

**Collaborative Filtering,** doesn’t need anything else except users’ historical preference on a set of items. Because it’s based on historical data, the core assumption here is that the users who have agreed in the past tend to also agree in the future. In terms of user preference, it usually expressed by two categories. **Explicit Rating**, is a rate given by a user to an item on a sliding scale, like 5 stars for Titanic. This is the most direct feedback from users to show how much they like an item. **Implicit Rating,** suggests users preference indirectly, such as page views, clicks, purchase records, whether or not listen to a music track, and so on. In this article, I will take a close look at collaborative filtering that is a traditional and powerful tool for recommender systems.

* 1. **RBM for CF**

Restricted Boltzmann Machine is a two-layer stochastic neural network consisting of visible and hidden units. Each visible unit is connected to all the hidden units in an undirected form. No visible/hidden unit is connected to any other visible/hidden unit. The stochastic, binary visible units encode user preferences on the items from the training data, therefore the state of every visible unit is known. Hidden units are also stochastic, binary variables that capture the latent features. A probability p(v; h) is assigned to each pair of a hidden and a visible vector:

(1)

where E is the energy of the system and Z is a normalizing factor, as defined in (Hinton, 2002).To train for the weights, a Contrastive Divergence method was proposed by Hinton (Hinton, 2002). Salakhutdinov et al. (Salakhutdinov et al., 2007), proposed an RBM framework for CF. Their model assumes one RBM for each user and takes only rated items into consideration when learning the weights. They presented the results of their approach on the Netflix data and showed that their technique was more accurate than Netflix’s own system. The focus of this RBM approach was only on the accuracy and predicting error and not explanation generation.



Figure 1. Conditional RBM for explanation

1. **Proposed Method**

This is the proposed method for Restricted Boltzmann Machine (RBM) algorithm which has been used by me where first I present my approach for measuring explainability of each teacher for each student using explicit rating data. Next, I propose the explainable RBM framework.

1. **Explanability**

Explainability can be formulated based on the rating distribution within the active user’s neighborhood. The main idea is that if many neighbors have rated the recommended item, then this could provide a basis upon which to explain the recommendations, using neighborhood style explanation mechanisms. For user-based neighbor-style explanations, such as the ones shown in Figure 1, we can therefore define the Explainability Score of item i for user u as follows:

(2)

where is the set of student u’s k neighbors, is the rating of x on teacher i and is the maximum rating value of on i. Neighbors are determined based on the cosine similarity. Without loss of information, is considered as 0 for missing ratings, indicating that student x does not contribute to the student based neighbor-style explanation of teacher i for student u. Given this definition, it is obvious that Explainability Score is between zero and one. Teacher i is explainable for student u only if its explainability score is larger than zero. When no explanation can be made, the explainability ratio would be zero.

1. **Conditional RBM**

The conditional RBM model takes explainability into account with an additional visible layer, m, with n nodes, where n is the number of items. Each node has a value between 0 and 1, indicating the explainability score of the relative item to the current user in the iteration, calculated as explained in Section 4. The idea is to define a joint distribution over (v, h), conditional on the explainability scores, m. Figure 1 presents the conditional RBM model with explainability. Based on (Hinton, 2010), the

are defined as:

(3)

) (4)

) (5)

where a, b, and c are biases, f and n are the numbers of hidden and visible units, respectively, and W and D are the weights of the entire network. is the logistic function

To avoid computing a model, we follow an approximation to the gradient of a different objective function called “Contrastive Divergence” (CD) (Hinton, 2002):

(6)

where is an element of a learned matrix that models the effect of ratings on h. Learning D, which is the effect of explainability on h, using CD, is similar

and takes the form:

(7)

1. **Project Result**

I tested my approach on the data set gather by myself from different sources including my university on teachers ratings data which consists of 10M ratings, on a scale of 1 to 5, for around 10681 teachers and 9508 students. The data is split into training and test sets such that 10% of the latest ratings from each students are selected for the test set and the remaining are used in the training set. Ratings are normalized between 0 and 1, to be used as RBM input. Then I compare my results with RBM, Explainable Matrix Factorization (Abdollahi & Nasraoui, 2016), user-based top-n recommendations (Herlocker et al., 1999), and non-personalized most popular items.

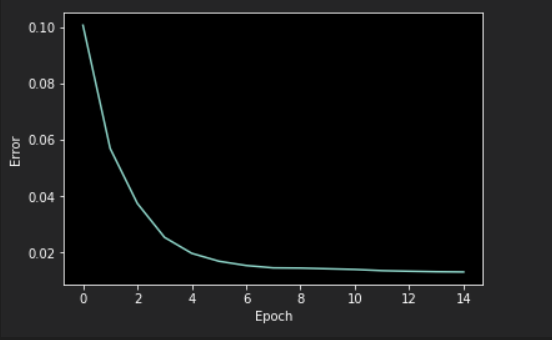


Figure 2. Error while training of RBM

Experiments is run 10 times and the average results are reported. To assess the rating prediction accuracy, we used the Root Mean Squared Error (RMSE) metric:

.

Note that RMSE can only be calculated for the prediction-based methods and not the top-n recommendation techniques.

To generate the top-n recommendation or prediction for the particular students I combine the output generate by each hidden layer before going to output layer and predict the recommended teacher to that particular students by mentioning most rated teacher at top and further it suggest by descending order of the recommendation score.

1. **Conclusion**

I presented an explainable Restricted Boltzmann Machine approach for Collaborative Filtering recommendations technique to recommend the best teacher to students based on their preference that achieves both accuracy and interpretability by learning an RBM network that tries to estimate accurate students ratings while also taking into account the explainability of a teacher to a students. Both rating prediction and explainability are integrated within one learning goal allowing to learn a network

**Reference**

[1] Abdollahi, Behnoush and Nasraoui, Olfa. Explainable matrix factorization for collaborative filtering. In Proceedings of the 25th International Conference Companion on World Wide Web, pp. 5–6. InternationalWorld WideWeb Conferences Steering Committee, 2016.

[2] Bilgic, Mustafa and Mooney, Raymond J. Explaining recommendations: Satisfaction vs. promotion. In Beyond Personalization Workshop, IUI, volume 5, 2005.

[3] Billsus, Daniel and Pazzani, Michael J. A personal news agent that talks, learns and explains. In Proceedings of the third annual conference on Autonomous Agents, pp. 268–275. ACM, 1999.

[4] Herlocker, Jonathan L, Konstan, Joseph A, Borchers, Al, and Riedl, John. An algorithmic framework for performing collaborative filtering. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 230–237. ACM, 1999.

[5] Herlocker, Jonathan L, Konstan, Joseph A, and Riedl, John. Explaining collaborative filtering recommendations. In Proceedings of the 2000 ACM conference on Computer supported cooperative work,

pp. 241–250. ACM, 2000.

[6] Herlocker, Jonathan L, Konstan, Joseph A, Terveen, Loren G, and Riedl, John T. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), 22 (1):5–53, 2004.

[7] Hinton, Geoffrey. A practical guide to training restricted boltzmann machines. Momentum, 9(1): 926, 2010.

[8] Hinton, Geoffrey E. Training products of experts by minimizing contrastive divergence. Neural Comput., 14(8):1771–1800, August 2002. ISSN 0899-7667. doi: 10.1162/089976602760128018. URL <http://dx.doi.org/10.1162/089976602760128018>.

[9] Hinton, Geoffrey E and Salakhutdinov, Ruslan R. Reducing the dimensionality of data with neural networks. Science, 313(5786):504–507, 2006.

[10] Mooney, Raymond J and Roy, Loriene. Contentbased book recommending using learning for text categorization. In Proceedings of the fifth ACM conference on Digital libraries, pp. 195–204. ACM, 2000.

[11] [https://towardsdatascience.com/intro to-recommender-system-collaborative-filtering-64a238194a26](https://towardsdatascience.com/intro%20to-recommender-system-collaborative-filtering-64a238194a26)

[12] <https://medium.com/@tanaykarmarkar/explainable-restricted-boltzmann-machine-for-collaborative-filtering-6f011035352d>