CSC2515 Final Project

Sijie (Sophie) Tian* **Tina Shen†**

Abstract

Variational autoencoders for collaborative filtering (VAE-CF) has been proposed for recommender systems, and proved to be well suited for modelling implicit feedback for the predictive task of recommendations. While the original work was able to out-perform some state of the arts models, the recommendation process could be viewed from a different angle by making item-based recommendations. In this project, we propose a model that utilizes and combines the results from two VAE-CF frameworks that learn latent representations for users and items respectively. The key idea is to use item information to augment user information from the implicit feedback. Experimental results demonstrated that this methodology can effectively boost more relevant items into the top recommendations, demonstrating a hybrid solution for applying VAE-CF for future recommender systems.

1 Introduction and related work

Recommender systems (RS) have been developed to widely assist and augment the social process of making recommendations to the end users (Resnick and Varian, 1997). They are integral to the web, helping users interact efficiently and effectively with an ever-increasing amount of content. In RS, Collaborative filtering (CF) (Goldberg et al., 1992) is a technique that produce recommendations by leveraging the similarity patterns across users and across items, using information collected through historical data such as ratings and purchase histories.

In collaborative filtering, k-Nearest-Neighbours (kNN) has been an early but arguably the most popular algorithm, which utilizes item-based and used-based filtering techniques (Sarwar et al., 2001; Resnick et al., 1994). As the dimension of user and item profile grows, CF evolves to make use of hybrid techniques (Shinde and Kulkarni, 2012), combining existing algorithms with machine learning techniques to bring scalability, efficiency, and non-linearity into the computational process. Matrix factorization (Luo et al., 2012) and Singular Value Decomposition (SVD) (Vozalis and Margaritis, 2007; Zhang et al., 2005) are other typical latent-factor approaches to reduce sparsity in CF. Sedhain et al. (2015) brings the idea of auto encoding for CF, Liang et al. (2018) later extended variational autoencoders (VAEs) to CF, which introduced a non-linear probabilistic model for RS.

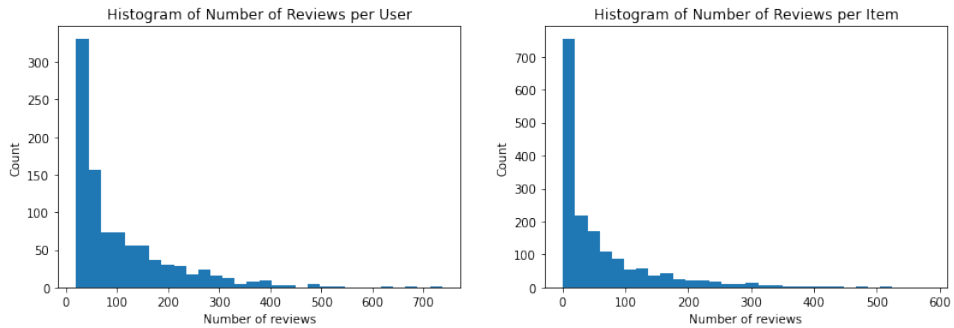
In this paper, we will discuss the implementation of VAE for RS named VAE-CF (Liang et al., 2018), applied to the classic MovieLens 100K dataset (Harper and Konstan, 2015), and the authors' attempt at improving the VAE model. The algorithmic choice was motivated by one of the authors' research direction, and serves to aid the authors' understanding of VAE applied to RS.

2 The dataset

The MovieLens dataset by Harper and Konstan (2015) is a classic dataset used for research in CF algorithms in RS. This dataset describes people's preferences for movies in the form of 1-5 star ratings at a particular point in time, expressed by <userID, itemID, rating, timestamp> tuples, and has

*Department of Mechanical and Industrial Engineering, University of Toronto; sophie.tian@mail.utoronto.ca
†Department of Mechanical and Industrial Engineering, University of Toronto; tina.shen@mail.utoronto.ca

Preprint. Under review.



(a) Frequency of ratings by user

(b) Frequency of ratings by item

Figure 1: Histograms of review frequencies for users and items.

a total of 100,000 reviews collected from 943 users on 1682 movies. For the purposes of this project, no further cleaning of the dataset is required since 1) the authors has kindly filtered the data such that each user has rated at least 20 movies, which alleviates the issue of data sparsity and improves accuracy of CF algorithms, and 2) there is no text data to be processed.

Exploratory data analysis Due to the simplistic forms of data in the MovieLens 100K dataset, we focus our data analysis on understanding the distribution of the number of reviews available per user and per item, as shown in Figure 1, which provides motivation for the type of CF method we should employ. Figure 1(a) suggests that the majority of users rated less than 100 movies, and Figure 1(b) shows that the majority of movies are given less than 20 ratings. Additionally, there are 141 movies with only a single review, and 384 movies with less than 5 reviews within this dataset. Since each user has rated at least 20 movies, we see that item-based data is much more sparse than user-based data. As a result, the performance of item-based CF methods alone may suffer compared to user-based methods. However, if we can primarily utilize user-based CF methods, then augment it using some item-based information, the performance of CF algorithms could potentially improve.

Data processing Since the VAE-CF model by Liang et al. (2018) performs CF for implicit feedback, which bases predictions on whether users enjoyed the movie, we binarize the explicit ratings by keeping ratings of four and five. To process the dataset into train, validation and test sets, we sort each user's ratings by timestamps, and insert the oldest 50% of ratings into the train set, the newer 20% into the validation set, and the newest 30% into the test set. This ensures that every user appears in all datasets, so we do not encounter the cold-start problem, and we always train on the oldest data, and validate and test on "future" data. We use this set of preprocessed data, which contain the fields "userID", “itemID", and "rating", for all baseline models as well as the VAE-CF model.

3 Predictive task, baseline and metrics

Using the MovieLens 100k dataset, our goal is to infer users' preferences for movies in the form of a ranked list of recommendations for each user. From the recommendations produced, we evaluate the model by computing two ranking metrics that evaluate against the top k recommendations: Recall@k and Normalized Discounted Cumulative Gain@k (NDCG@k). Recall@k computes the proportion of the top k recommended items that are preferred by users in the test set, and NDCG@k uses a graded relevance scale to measure the "gain" of a recommendation based on its position in the top-k list, emphasizing the importance of higher ranks. For this project, we selected Recall@50, Recall@100 and NDCG@100, corresponding to the values selected by Liang et al. (2018).

Since neighborhood-based CF has been the most popular at the beginning of the RS, we chose both user-based and item-based CF using kNN as baseline methods. We used a modified version of the kNN implementation from the CaseRecommender library developed by da Costa et al. (2018). In addition, we also selected SVD as a baseline since it demonstrates the application of dimensionality reduction in recommendations, which we developed ourselves.

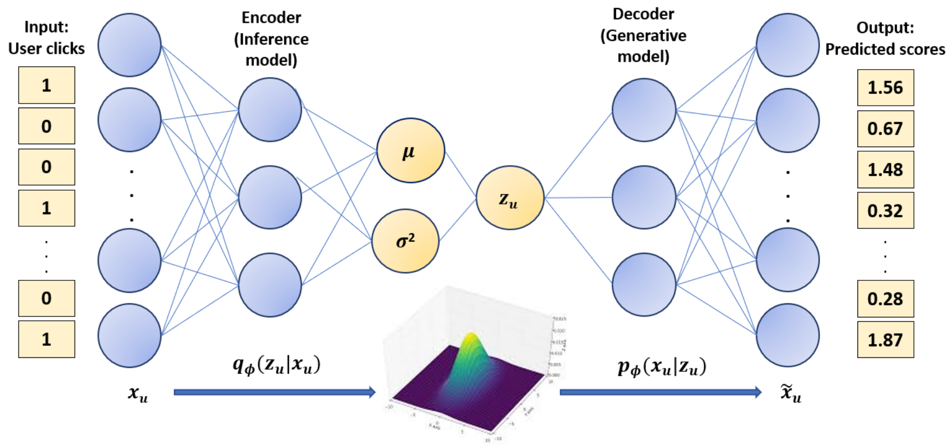


Figure 2: Model architecture

4 Model

The model architecture of VAE-CF, as proposed by Liang et al. (2018), is presented in Figure 2. The encoder portion, which is an inference model, learns a latent representation z_u for each user assumed to be sampled from a standard Gaussian prior. It uses variational inference to learn the variational parameters to the distribution $q_\phi(z_u|x_u) = \mathcal{N}(\mu_u, diag\{\sigma_u^2\})$ to approximate the intractable posterior distribution $p(z_u|x_u)$ for the latent embeddings given the user click histories. The decoder portion, which is a generative model, assumes the user click history $x_u \sim Mult(N_u, \pi(z_u))$ are drawn from a multinomial likelihood and perform generation, where $\pi(z_u)$ is a probability distribution over all the items produced by the non-linear transformation of z_u.

The original VAE-CF model performs CF in a user-based approach. However, instead of learning latent embeddings for users, the learning process can also be applied to items. Motivated by insights drawn from Section 2, the team attempted to enhance the VAE-CF model by incorporating item-based information into the recommendations produced. Specifically, we combined the user-rating predictions produced from both user-based and item-based VAE-CF so that the item information could be used to augment user-based results, and introduced hyperparameters as the mixture coefficients for the two models. Then, to optimize model performance, all hyperparameters were tuned using grid search against the defined metrics. A k-fold cross validation should be performed during hyperparameter tuning to produce a more reliable performance. Since the dataset is time-series data with high sparsity level, the data should be split such that the training data are older than the test data.

5 Results and discussion

After the hyperprameter tuning process, a best set of hyperparameters were found to form the final model. As seen in Figure 3, We landed at assigning weights of 0.1 and 0.9 respectively to item-based and user-based predictions, which corresponds to our initial expectations that we could use primarily user-based information and some item-based information to make predictions. We found that the performance decreases with larger weight assignments to the item-based approach, which is consistent with the fact that item-based data is much sparser than user-based data for the dataset selected.

The comparison between the performance of the proposed model with that of the baseline models are presented in Table 1. While the modified VAE-CF model outperformed other models in the recall@K metrics, it was not able to out-compete other models for NDCG@100. This suggests that while the model was able to retrieve more relevant items in the top 50 and 100 recommendations, it was not able to place the correctly predicted items at higher rankings. The performances of VAE-CF and the modified VAE-CF correspond to the claim by Liang et al. (2018) that such models are well suited for modelling implicit feedback data and therefore handles the sparsity issues created by large data sets well, compared to the selected baseline models which performs linear computations. Comparing to other nonlinear model utilized for Collaborative Filtering such as AutoRec (Sedhain

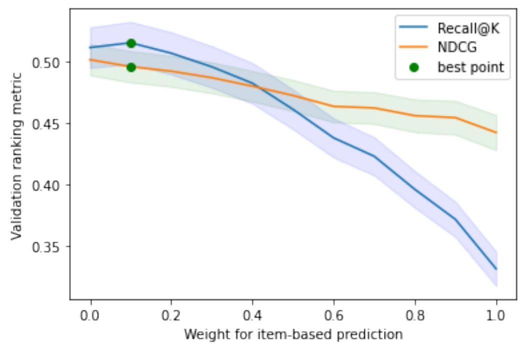


Figure 3: Metric results with incremental item-based coefficients.

Table 1: Comparison of final results.

Method	Recall@50	Recall@100	NDCG@100
VAE-CF	0.3484	0.5117	0.5018
Modified VAE-CF	**0.3492**	**0.5156**	0.4961
Used-based kNN	0.3373	0.5000	0.5044
Item-based kNN	0.3465	0.5103	**0.5076**
SVD	0.3175	0.4561	0.4970

et al., 2015), VAE-CF adds in the aspect of variational inference to learn a distribution of the user or item embeddings. The performance of CF using kNN heavily depends on the computed similarity matrix. Although it is easy to implement, it is not capable of handling the cold-start issues. However, for this project, kNN performed relatively well since the cold-start issue was alleviated from the preprocessing techniques used and a good similarity matrix has been computed. While SVD is simple to implement and able to handle sparsity issues by modeling the latent representations for each user-item pair, it had the worst performance among the models being compared. A reason for this result is that SVD learns user and item representations based on the training dataset which has a different embedded feature representations with the test dataset (e.g. users' movie preference changed from watching popular movies to certain genres).

As mentioned above, the proposed modified model has the advantage that it accounted for more information by combining the prediction output results. Since the item-based and the user-based approach captures different information and identifies relevant items differently, in this way, more relevant items (hits) could be pushed into the top K recommendations list. However, the variational inference and generation process for these two approaches were operated independently, which could hurt the ranking performance due to the fact that the probability assignment for the identified relevant items were performed independently, and therefore it failed to push more relevant items to the top of the ranking list. Future work therefore includes investigating the interactions between used-based and item-based embeddings in the modified VAE-CF model and leveraging any patterns found to enhance its performance. Moreover, making recommendations based on only the historical rating score creates limitations. Side information from the items and users' profiles could be leveraged to learn better items and users' latent representations such as the work presented by Deng et al. (2019).

6 Conclusion

The sparse and large dataset used by the recommender systems consist of a great amount of implicit feedback and as the dimension of such datasets grow, non-linear probabilistic latent-variable model like VAE-CF is more desirable. The work from this project sheds a light on making use of the a combined approach for making recommendations, proving that enrolling more information from the user and item profile is beneficial for the predictive task. Future work on the VAE-CF models for both user-based and item-based approaches includes operating the variational inference and generation processes in parallel to obtain more consistent probabilistic allocations for the end predictions.

7 Attributions

For this project, both members of the team contributed equally. In particular, Tina focused on implementing the main VAE framework and conducting grid search for hyperparameter tuning, and Sophie focused on the exploratory data analysis, data preprocessing and baseline models. We also checked each other's work upon completion of each component to ensure its correctness. All other components of the project such as the compilation of the final report and the presentation received equal contributions from each member of the team.

References

da Costa, Arthur, Eduardo Fressato, Fernando Neto, Marcelo Manzato, Ricardo Campello. 2018. *Proceedings of the 12th ACM Conference on Recommender Systems*. RecSys '18, ACM, New York, NY, USA, 494–495. doi:10.1145/3240323.3241611. URL `http://doi.acm.org/10.1145/3240323.3241611`.

Deng, Xiaoyi, Fuzhen Zhuang, Zhiguo Zhu. 2019. Neural variational collaborative filtering with side information for top-k recommendation. *International Journal of Machine Learning and Cybernetics* **10** 3273–3284.

Goldberg, David, David Nichols, Brian M Oki, Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM* **35** 61–70.

Harper, F. Maxwell, Joseph A. Konstan. 2015. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems* **5**. doi:10.1145/2827872.

Liang, Dawen, Rahul G Krishnan, Matthew D Hoffman, Tony Jebara. 2018. Variational autoencoders for collaborative filtering. *Proceedings of the 2018 World Wide Web Conference*. 689–698.

Luo, Xin, Yunni Xia, Qingsheng Zhu. 2012. Incremental collaborative filtering recommender based on regularized matrix factorization. *Knowledge-Based Systems* **27** 271–280.

Resnick, Paul, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, John Riedl. 1994. Grouplens: an open architecture for collaborative filtering of netnews. *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. 175–186.

Resnick, Paul, Hal R Varian. 1997. Recommender systems. *Communications of the ACM* **40** 56–58.

Sarwar, Badrul, George Karypis, Joseph Konstan, John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th international conference on World Wide Web*. 285–295.

Sedhain, Suvash, Aditya Krishna Menon, Scott Sanner, Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. *Proceedings of the 24th international conference on World Wide Web*. 111–112.

Shinde, Subhash K, Uday Kulkarni. 2012. Hybrid personalized recommender system using centering-bunching based clustering algorithm. *Expert Systems with Applications* **39** 1381–1387.

Vozalis, Manolis G, Konstantinos G Margaritis. 2007. Using svd and demographic data for the enhancement of generalized collaborative filtering. *Information Sciences* **177** 3017–3037.

Zhang, Sheng, Weihong Wang, James Ford, Fillia Makedon, Justin Pearlman. 2005. Using singular value decomposition approximation for collaborative filtering. *Seventh IEEE International Conference on E-Commerce Technology (CEC'05)*. IEEE, 257–264.