2017 31st International Conference on Advanced Information Networking and Applications Workshops

Stacked Denoising Autoencoder-based Deep Collaborative Filtering Using the Change of Similarity

Yosuke Suzuki
Graduate School of Integrated Basic Sciences
Nihon University
Email: f6115m18@educ.chs.nihon-u.ac.jp

Tomonobu Ozaki
Department of Information Science,
Nihon University
Email: tozaki@chs.nihon-u.ac.jp

***Abstract*—Recommender systems based on deep learning technology pay huge attention recently. In this paper, we propose a collaborative filtering based recommendation algorithm that utilizes the difference of similarities among users derived from different layers in stacked denoising autoencoders. Since different layers in a stacked autoencoder represent the relationships among items with rating at different levels of abstraction, we can expect to make recommendations more novel, various and serendipitous, compared with a normal collaborative filtering using single similarity. The results of experiments using MovieLens dataset show that the proposed recommendation algorithm can improve the diversity of recommendation lists without great loss of accuracy.**

I. Introduction

Nowadays, not only theoretical studies but also application researches on deep learning are conducted actively. It is widely recognized that one of the next promising application fields in deep learning is recommender systems [1]–[12]

Recommender systems are required to predict items preferred by a target user to help the user find useful ones among unmanageable number of items.

A typical deep learning technology applied to recommender systems is autoencoder [1]–[4]. Autoencoders [13] are an unsupervised deep learning algorithm, and they are able to learn a high-level abstract representation of input data. In case of recommender systems, we build an autoencoder by providing absolute values or concrete ratings of items by users as input. Then, a high-level representation of users preference, *e. g.* what kind of genre of items the user likes, can expect to be obtained in the hidden layers of the autoencoder. While the similarity among users play an essential role in recommendation, the obtained high-level representations have an advantage in estimating similarity among users more accurately compared with the similarity estimation by the original input directly.

Collaborative filtering(CF) [14], one of the most fundamental recommendation algorithms, requires to estimate similarity among users to make recommendation lists. In a simple CF, the similarity between two users is estimated as a Pearson correlation coefficient between concrete rating values of items rated by the users in common. Therefore, it is difficult to take the information about genre or domain of items into account to calculate the similarity. In addition, since there exists only small number of items rated by two users in common in general, the obtained similarity is not always accurate, precise and reliable. By contrast, the high-level representations of users preference obtained by autoencoders have the possibility to estimate accurate and reliable similarity among users taking the genre of domain into account even if the number of rated items by the users in common is not large. Hereafter, while we call the similarity or correlation between the rated values as "surface similarity", the similarity derived from the high-level representations is referred to as "latent similarity".

In this paper, we propose a new collaborative filtering algorithm that effectively uses the latent similarity learned by autoencoders to select similar users for making the prediction. In addition to the simple use of the latent similarity, we propose to use the surface similarity and the latent one simultaneously to achieve diverse recommendations. More precisely, the users who are dissimilar in the surface similarity but are similar in the latent similarity are to be selected. In other words, the algorithm uses the latently similar users who rate many different concrete items from the target. Note that, since the users having high surface and latent similarities with the target usually rate the items that the target already rated, they cannot provide novel and serendipitous items. The users having low surface and latent similarities provide no items the target prefers. By employing the users having low surface but high latent similarities for the recommendation, we can expect to provide the novel, divergent and serendipitous items that the target does not know but prefers.

The rest of this paper is organized as follows. In section II, we describe related work. The algorithms of collaborative filtering and autoencoder are introduced in section III. We propose our collaborative filtering algorithms based on stacked denoising autoencoders in section IV. Experimental results are reported in section V. Finally, we conclude the paper and describe future work in section VI.

978-1-5090-6231-7/17 $31.00 © 2017 IEEE
DOI 10.1109/WAINA.2017.72

II. Related Work

In the research area of recommender systems, several algorithms are proposed for predicting the rating values by the target users by using deep learning technology. Some of them employ autoencoders, and the main research focus is to handle unrated items or missing values. To cope with the data sparsity in learning autoencoders for recommendations, Sedhain [2] and Strub [3] propose special algorithms that drop the unit corresponding to the missing values per users in the learning phase and use all units in the prediction phase. In this paper, we adopt the same algorithm for learning and prediction.

As other approach for dealing missing values, the conversion methods of unrated values into statistical representative values such as average are proposed [1], [4]. In these studies, the output values of autoencoders are considered as the predicted values of unrated items, and the values are directly used to make the recommendations.

III. Collaborative Filtering and Stacked Denoising Autoencoders

Before entering upon a discussion of the Collaborative Filtering (CF) and the Stacked Denoising Autoencoders (SDAE), we give a formal definition of the input data in this paper.

Let $U = \{u_1, \cdots, u_{|U|}\}$ be a set of users and $I = \{i_1, \cdots, i_{|I|}\}$ be a set of items. We define the range of ratings as $\mathcal{R} \in \mathbb{R} \cup \{NA\}$ where NA represent missing value. A rating vector of a user u is denoted as $\boldsymbol{r}_u = (r_{u1}, \cdots, r_{u|I|}) \in \mathcal{R}^{|I|}$. If r_{ui} is NA, it means that the user u has not rated to the item i yet.

A. Collaborative Filtering

Collaborative filtering is one of the most basic recommendation algorithms. In this paper, we show the user-based collaborative filtering proposed by GroupLens [14].The basic idea of collaborative filtering is to recommend items which are received a high evaluation by the top-N users similar to the target user. Concretely speaking, a collaborative filtering has two procedures: calculation of similarities among users and prediction of rating values. At first, the similarity of preferences between a target user and the others is estimated by Pearson correlation coefficient. The Pearson correlation between two users x and y is defined as

$$sim_{xy}^{pearson} = \frac{\sum_{j \in \mathcal{J}_{xy}} (r_{xj} - \overline{r}_x)(r_{yj} - \overline{r}_y)}{\sqrt{\sum_{j \in \mathcal{J}_{xy}} (r_{xj} - \overline{r}_x)^2}\sqrt{\sum_{j \in \mathcal{J}_{xy}} (r_{yj} - \overline{r}_y)^2}}$$

where $\mathcal{J}_{xy}$ is a set of items evaluated by both users, and $\overline{r}_u$ is the average of ratings by user u. In the second phase, the top-N users similar to the target are selected, and the rating values of unrated items by the target are to be predicted. A predicted value $\tilde{r}_{ui}$ of an item i by a user u is calculated by the following formula.

$$\tilde{r}_{ui} = \overline{r}_u + \frac{\sum_{x \in \mathscr{U}_u \cap \mathcal{J}_i} sim(u,x) \cdot (r_{xi} - \overline{r}_x)}{\sum_{x \in \mathscr{U}_u \cap \mathcal{J}_i} sim(u,x)}$$

where $\mathscr{U}_u$ is a set of similar users, $\mathcal{J}_i$ is a set of users who rate the item i and $\mathrm{sim}(u,x)$ denotes the similarity between u and x, *i.e.* $sim_{xy}^{pearson}$ in the basic algorithm. The recommendation list will be generated by collecting items having high predicted values.

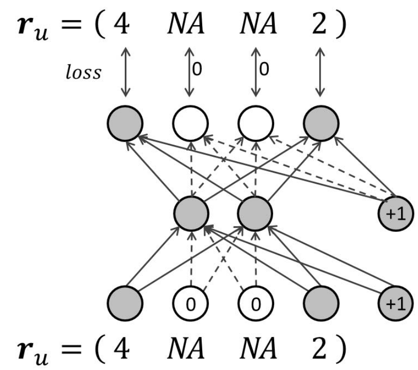

Figure 1. The structure of single layer autoencoders for recommendation

B. Stacked Denoising Autoencoders

Autoencoders are one of unsupervised Deep Learning models. As shown in Figure 1, a basic autoencoder has 3 layer networks called input layer, hidden layer and output layer. For simplicity, we expound a autoencoder model whose input data is a dense rating vector $\hat{\boldsymbol{r}}_u$. The output is represented

$$h(\hat{\boldsymbol{r}}_u; \theta) = f\left(\boldsymbol{W} \cdot g(\boldsymbol{V}\hat{\boldsymbol{r}}_u + \boldsymbol{\mu}) + \boldsymbol{b}\right),$$

where f, g is arbitrary non-linear function which is calld activation function, $\boldsymbol{V}, \boldsymbol{W}$ mean weight and $\boldsymbol{b}, \boldsymbol{\mu}$ mean bias. Here, we designate the model parameter set of $\{\boldsymbol{W}, \boldsymbol{V}, \boldsymbol{b}, \boldsymbol{\mu}\}$ by θ. An autoencoder trains parameter θ using SGD so that the loss function $E(\theta)$ will be decreased.

$$E(\theta) = \sum_{u \in U} \|\hat{\boldsymbol{r}}_u - h(\hat{\boldsymbol{r}}_u; \theta)\|^2$$

In addition, we employ a Denoising Autoencoder (DAE) to enhance the robustness. The difference between simple autoencoder and DAE is that whether we add noise to input value. The loss function of DAE is defined as

$$E(\theta) = \sum_{u \in U} \|\hat{\boldsymbol{r}}_u - h(\tilde{\boldsymbol{r}}_u; \theta)\|^2$$

where $\tilde{\boldsymbol{r}}_{\boldsymbol{u}}$ is the corrupted input value of $\boldsymbol{r}_u$.

The basic autoencoders assume that the input vector $\hat{r}_u$ has no missing value. Actually, however, rating data is very sparse and contains lots of missing values. In this paper,

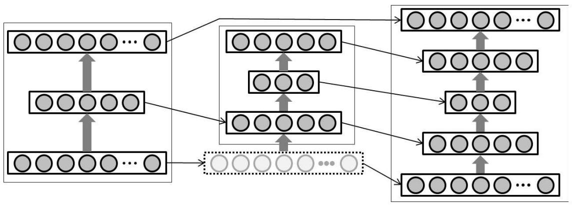

Figure 2. The structure of stacked autoencoders

we consult an autoencoder model proposed by Sedhain [2] and Strub [3] to treat the sparse input. In the learning phase of autoencoders, we perform the forward propagation by replacing the missing value with 0, and do the backward propagation by setting the error of the unit corresponds to missing value is 0 regardless of the actual values. In the prediction phase, as the same as the learning phase, we perform the forward propagation by replacing the missing value with 0 again.

The stacked autoencoders, depicted in Figure 2, can be easily learned by applying the above procedure repeatedly. In addition, stacked denoising autoencoders can be obtained by using corrupted inputs. We denote the input layer as 0th layer and subsequent layers as first-, second-, $\cdots$, layers. The output of l-th layer is denoted as $\boldsymbol{z}_u^l$. The outputs can be formally defined as

$$\boldsymbol{z}_u^l = \begin{cases} \boldsymbol{r}_u & (l = 0) \\ f^l(\boldsymbol{W^{l-1}} \cdot \boldsymbol{z_u^{l-1}} + \boldsymbol{b^{l-1}}) & (l > 0) \end{cases}$$

where f^l is a nonlinear activation function of l-th layer, $\boldsymbol{W}^l$ and $\boldsymbol{b}^l$ are weight and bias between $l-1$ and l layer, respectively.

IV. Collaborative Filtering with Stacked Denoising Autoencoders

In this section, we propose two advanced collaborative filtering algorithms that use the Stacked Denoising Autoencoders (SDAE).

A. CF using Hidden Layer

Our first proposal is a collaborative filtering using the output of hidden layers for enhancing the accuracy of users similarity calculation and recommendation. The idea is just to change the similarity calculation method as follows:

- We use the output of hidden layer instead of rating data.
- We adopt not Pearsons correlation but cosine similarity for the similarity among users.
- Each unit value is standardized.

We call the obtained similarity as the latent similarity. The latent similarity enables us to calculate the similarity among users in higher accuracy than the simple CF using the surface similarity, because it can capture the semantics of the items and it is robust with respect to the difference of rated items. The proposed collaborative filtering method is denoted as CF_l where l represents the number of hidden layer. For the comparison purpose, we prepare a simple variant of CF_l, denoted as $\tilde{CF}_l$, obtained by using the non-standardized output of hidden layer. The concept of the proposed method CF_l is shown in ③ of Figure 3.

B. CF using Change of Similarity

We propose a collaborative filtering using the difference between the surface similarity and the latent one for enhancing the diversity. The basic idea of the proposal is to utilize the users who are not similar with a target user in the surface similarity but are similar in the latent similarity. Because of the high latent similarity, we believe that such users rate a lot of items the target user do not usually observe but the target prefers. Thus, we can expect to make the recommendation more diverse.

Given a user u and user specified parameter N and θ, the proposed algorithm requires the following conditions to a user v to be used in the recommendation:

1) the rank of surface similarity between u and v is greater than N,
2) the value of latent similarity between u and v is greater than θ, and
3) the rank of "the change of similarity" defined below is in top-N within the users satisfying the first and second conditions.

The change of similarity is defined as

$$change_{uv} = \frac{(Rank_{uv}^{surface} - Rank_{uv}^{latent})}{Rank_{uv}^{surface}}$$

where $Rank_{uv}^{surface}$ and $Rank_{uv}^{latent}$ denotes the ranks of surface and latent similarity between u and v, respectively.

The proposed algorithm is denoted as CF_l^{Diff} where l means the number of hidden layer. The algorithm uses the latent similarities of selected users to predict the rating values. The procedure of CF_l^{Diff} is depicted in ① and ② of Figure 3.

V. Experiments

A. Datasets and Overview

Experiments using MovieLens 1M datasets[1] are conducted for evaluating the proposed methods. The dataset contains 1 million of ratings on 3900 movies by 6000 users. The range of rating is from 1 to 5. In the experiments, we randomly selected 70% of users for training, 10% for validation and 20% for test. We trained the stacked denoising autoencoder with the parameters shown in table I. In addition, if we confirm the overfitting in the training, then we apply early stopping.

In the test phase, 20% of ratings are randomly masked per each user. We set $N = 100$ and $\theta = 0$, respectively. For each

[1] http://grouplens.org/datasets/movielens/

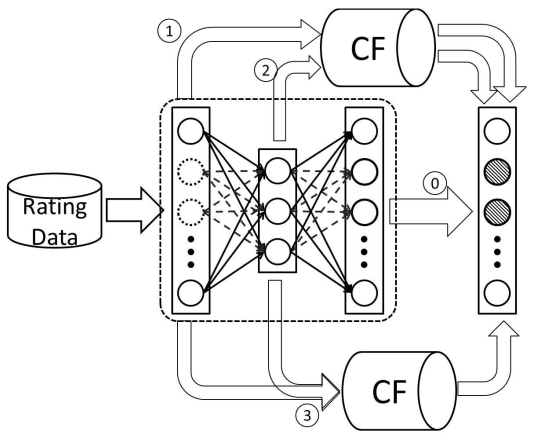

Figure 3. Collaborative Filtering with AE

Table I
PARAMETERS OF SDAE

Hidden layer number	3
Number of unit	50 unit，20 unit
Number of epochs	5000
Batch size	30
Activation function in the hidden layers	Sigmoid
Activation funciton in the output layer	Identity Mapping
Loss Function	MSE(Mean Squared Error)
Gaussian Noise	$\mu = 0$, $\sigma^2 = 0.001$
learning method	Adam

target user, we exclude the users who rate less than 6 items in common with the target from the similarity computation.

B. prediction accuracy

We first evaluate the recommendation accuracy with Root Mean Squared Error(RMSE). RMSE is a fundamental evaluation measure in recommender systems, and is defined formally as follows:

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{N}(y_i - \hat{y_i})^2}$$

where y_i is an actual rating value, $\hat{y_i}$ is a predicted rating value and M is number of predicted items.

The result is shown in table II. While the third column in table II shows the RMSE for all predicted items, the fourth column stores the RMSE for the recommendation lists containing Top-50 items having high predicted values. The third model CF_1, which utilizes the latent similarity, performed the best on the RMSE for all items. This result shows that the use of latent similarity derived from a hidden layers in autoencoders is effective on the gain of the recommendation accuracy. Since the model CF_1 and CF_2 performed better than $\tilde{CF_l}$ and $\tilde{CF_2}$, respectively, we can confirm that the standardization plays essential roles for the gain of the accuracy.

Table II
RMSE

ID	Model	RMSE(all item)	RMSE(rec list)
1	CF	1.072	1.300
2	$\tilde{CF_1}$	0.938	1.033
3	CF_1	0.914	0.965
4	$\tilde{CF_2}$	0.948	1.169
5	CF_2	0.935	1.001
6	CF_1^{Diff}	1.251	1.207
7	CF_2^{Diff}	1.252	1.185

The results by CF_l^{Diff} and CF_2^{Diff} are worse than that by CF in RMSE for all items. However, the actual recommendation quality by CF_l^{Diff} and CF_2^{Diff} must be better than that by CF since they achieve better RMSE for recommendation lists.

C. differences among recommended items

As the next experiement, we evaluate the similarity of the recommended items by different models. In the experiments, the recommendation lists are build by selecting top 50 items with respect to the predicted values. We measure the similarity between two recommendation models using average value of Jaccard similarity for users formally defined as

$$sim^l(A, B) = \frac{1}{|U^T|}\sum_{u \in U^T}\frac{|list_A(u) \cap list_B(u)|}{|list_A(u) \cup list_B(u)|}$$

where $list_x(u)$ denotes the user u's recommendation list created by a model x. The results is shown in tableIII.

It can be observed from the result that all the similarities between CF and others are low. Thus, we can confirm that, regardless of the application of standardization, all the proposed algorithms using latent similarities can provide different items from those by a traditional algorithm. Since both of the similarity between CF_1 and CF_2 and that between $\tilde{CF_1}$ and $\tilde{CF_2}$ are relatively high, we can judge that the effect by the difference of layers in autoencoders is small and less than the that by the difference of models for selecting similar users.

Table III
JACCARD SIMILARITY AMONG RECOMMENDED ITEMS

ID \ ID	CF_1	CF_2	CF_1^{Diff}	CF_2^{Diff}
CF	0.015	0.016	0.012	0.012
CF_1		0.098	0.011	0.013
CF_2			0.01	0.011
CF_1^{Diff}				0.101

D. Diversity of recommended items

It is expected that a small number of items are rated by a target user and latently similar users in common in the proposed algorithms since CF_l ignores the surface

Table IV
DIVERSITY OF RECOMMENDATION MODELS

Model	diversity	avg. number of co-rated items
CF	0.958	8.49
CF_1	0.910	30.20
CF_2	0.920	25.92
CF_1^{Diff}	0.975	3.36
CF_2^{Diff}	0.976	3.61

similarity and $\tilde{CF}_l$ employs the users having small surface similarity. In order to assess that the similar users used in recommendation models can provide different items to a target, we calculate the diversity of the recommendation through the jaccard similarity of rated items among the target and top-N similar users by following expression.

$$sim^M = 1 - \frac{1}{|U^T|} \sum_{u \in U^T} \frac{1}{N} \sum_{v \in U_u^{sim}} \frac{|list_M(u) \cap list_M(v)|}{|list_M(u) \cup list_M(v)|}$$

where U^T is a set of test users and U_u^{sim} is a set of top-N similar users with the target u in the model. In addition, we confirm the average number of rated items by a target and similar users in common. The results are summarized in table IV.

It can be observed that the proposed models CF_1^{Diff} and CF_2^{Diff} have higher values of diversity than the traditional model CF. This result confirms that the proposed models can provide diverse recommendations. On the other hand, the number of rated items in common are very large in the models CF_1 and CF_2. Thus, while the diversity becomes low in these two models, we can expect to select similar users accurately and it lead the accurate recommendations.

VI. CONCLUSION

In this paper, we proposed a stacked denoising autoencoder based deep collaborative filtering using the change of similarity, and conduct several experimental evaluations using MovieLens1M datasets. As the result, we confirmed that similarity calculation using hidden layer as well as the standardization of the values contribute to the recommendation accuracy. In addition, we confirm that the proposed algorithm using the change of the surface and latent similarity has great potential to realize diverse recommendations without loss of recommendation accuracy.

For future work, we plan an online evaluation for the further assessment of the contribution of proposed algorithms on the diversity. In addition, we are investigating more effective measures for capturing the difference between surface and latent similarity.

REFERENCES

[1] K. Kazuya and M. Yutaka, "The application of deep learning in recommender system," *Proceedings of the 28th Annual Conference of the Japanese Society for Artificial Intelligence*, vol. 28, pp. 1–4, 2014. (in Japanese)

[2] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders Meet Collaborative Filtering," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 111–112, 2015.

[3] F. Strub and J. Mary, "Collaborative Filtering with Stacked Denoising AutoEncoders and Sparse Inputs," in *NIPS Workshop on Machine Learning for eCommerce*, 2015.

[4] S. Yosuke and O. Tomonobu, "Experimental Evaluation of Input Vector Complemention for Autoencoder-based Recommendation Systems", SIG-KBS-B504-02 in the Japanese Society for Artificial Intelligence, pp.6–11, 2016. (in Japanese)

[5] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th International Conference on Machine Learning*, pp.791–798, 2007.

[6] Y. Zheng, C. Liu, B. Tang, and H. Zhou, "Neural autoregressive collaborative filtering for implicit feedback," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp.2–6, 2016.

[7] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp.7–10, 2016.

[8] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp.17-22, 2016.

[9] J. B. P. Vuurens, M. Larson, and A. P. de Vries, "Exploring deep space: Learning personalized ranking in a semantic space," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp.23–28, 2016.

[10] G. Zanotti, M. Horvath, L. N. Barbosa, V. T. K. G. Immedisetty, and J. Gemmell, "Infusing collaborative recommenders with distributed representations," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp.35–42, 2016.

[11] H. Dai, Y. Wang, R. Trivedi, and L. Song, "Recurrent coevolutionary latent feature processes for continuous-time recommendation," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp. 29–34, 2016.

[12] F. Strub, R. Gaudel, and J. Mary, "Hybrid recommender system based on autoencoders," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp.11-16, 2016.

[13] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, pp.1096–1103, 2008.

[14] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pp.175–186, 1994.