Machine Learning in Recommender System

Keywords: Recommender System, “Cold-start”, Collaborative Filtering

1 Motivation

During the last few decades, recommender systems have been an important part of the internet [1]. People are experiencing the change of an era of lack of information to information overload. Recommender relationship between users and items is shown in Fig.2(a). For instance, Netflix is always trying to make proper recommendations to the users aiming at attracting them to stick to the website. It is both a challenge for information users and for information producers. For information users, it is difficult to find the information of interest from a large amount of information; while it is also a challenge for the information producers to make their products stand out. However, information producers are often faced with the dilemma that without users' information, they cannot start making recommendations, but without proper recommendations, the users may not be attracted to the information or the product. It is known as a “cold-start” problem [2, 3]. That's why some websites would ask the users to pick several tags such as “Action Comedies” or “Biographical Documentaries”. Good recommendation systems should be with high relevance, diversity, real-time and even be able to bring serendipity to the users. Therefore, it is necessary to pay attention to and generate new recommender system algorithms.

2 Problem Statement

2.1 Notations

In this section, the notations in recommender systems are proposed. Consider a recommender system of n users' preference on m items, e.g. n users of a music website who show different preferences in m songs, r_{ui} is denoted as the preference score of user u on item i. Notations based on the basic user-item framework are as follows:

(a) θ_{ui}: Expected preference score of a user-item pair (u, i).

(b) K: The number of latent factors for both users and items, which is also the rank in the latent factor model [4].

(c) Ω: Set of observed preference scores. Notice that $|\Omega| << nm$.

(d) $x_{ui} \in X \subset Rd$: A user-item specific covariate vector.

2.2 Terminology

For brevity, main terminologies in Recommender Systems are explained in this section:

(a) Singular Value Decomposition (SVD): The SVD a matrix is a factorization of that matrix that generalizes the eigen-decomposition of a square normal matrix [5]. To be specific, the SVD of a $m \times n$ matrix will decompose the matrix to the form $U \sum V^T$, where U and V are unitary matrix.

(b) L^p-norm: L^p spaces are function spaces defined using a natural generalization of the p-norm for finite-dimensional vector spaces, and the p-norm or L^p-norm of x is defined by:

$$||x||_p = (|x_1|^p + |x_2|^p + +|x_n|^p)^{1/p} \tag{1}$$

(c) MapReduce: It is a programming model for processing and generating big data sets, which is a specialization of the split-apply-combine strategy for data analysis. Inspired by programming functions Map and Reduce, the framework divides huge computation task into small tasks (Map) and integrates them to generate a single result [6]. The scheme of MapReduce is shown in Fig.1(a).

3 Methodology and Pros and Cons

Collaborative filtering (or its modifications) [7] and content-based filtering are two main streams for training recommender systems. Collaborative filtering predicts users' behaviors based on similar users [8], whereas content-based filtering builds user and item profiles recommends items with similar profiles [9, 10]. The methods discussed in this section are based on these two frameworks, and their novel modifications have further improved the accuracy.

3.1 Latent Factor Methods

Latent factor models proposed an approach by transforming both items and users to the same latent factor space [11], thus making them comparable directly [12]. In this section, the framework of Latent Factor Model will be introduced first two novel models based on the framework will then be discussed.

3.1.1 Framework of Latent Factor Model

Matrix Factorization (MF) is one of the most implemented approaches in recommender systems, but traditional MF has some limitations. In this subsection, the framework of a refined Matrix Factorization model, Regularized Latent Factor model, is proposed. A recommender system can be formulated as the following:

$$r_{ui} = \theta_{ui} + \epsilon_{ui} = p_u^T q_i + \epsilon_{ui}, \qquad 1 \leq u \leq n, 1 \leq i \leq m \tag{2}$$

where θ is the expected preference score of a user-item pair (u, i), $\theta_{ui} = E(r_{ui})$, and ϵ_{ui} is the error. The latent factor model assumes that the expected value of preference score can be represented by user factor p_u and item factor q_i as $\theta_{ui} = p_u^T q_i$. In a latent factor model, the objective is to estimate the personalized parameters $P = (p_1, ..., p_n)^T$ and $Q = (q_1, ..., q_m)^T$.

3.1.2 Regularized SVD Method

Singular value decomposition (SVD), as explained in Section 2.2, is a factorization of matrix [13]. A regularized SVD method in recommender systems was first proposed [14] as an essential part of recommender systems. A more recent regularized SVD method [15] decompose θ_{ui} in the latent factor model. With the motivation to make the matrix θ_{ui} sparser [16], $\theta = P_u^T Q_i$ will be decomposed [17] into $\theta = \bar{P}_u^T \bar{Q}_i$ where $\bar{P}^T \bar{P}$ and $\bar{Q}^T \bar{Q}$ are diagonal.

Compared to factorization imposed by orthogonality, this method is seeking for a sparser factorization. The reason why a more sparse matrix is preferred is due to the fact that the set of observed preference scores is far smaller than the possible user-item possibility space mn, $\Omega << mn$. The redundant p_0, q_0 and, accordingly, $\theta_0 = p_0^T q_0$ will be removed. The method also decomposes P and Q to global information and local information:

$$\theta_{ij} = x_j^T \alpha + \beta^T y_i + \alpha_j^T b_j \tag{3}$$

where x_j and y_i represent global information of users and items, whereas a_j and b_i reflect specific or local information for the j-th user over the i-th item. The global information includes the overall situation of an user or an item, whereas the local information is only the specific preference for a user over an item. Finally, the objective function estimating $P = (p_1, ..., p_n)T$ and $Q = (q_1, ..., q_m)T$ is as follows:

$$\min_{P,Q} \frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} (r_{ui} - p_u^T q_i)^2 + \lambda\{\sum_{u=1}^{n} J(p_u) + \sum_{i=1}^{m} J(q_i)\} \tag{4}$$

where λ is a tuning parameter and J is a penalty function.

Overall, the regularized SVD model made great progress in this field. It can solve the problem of high missing value in recommender systems by seeking a sparsest factorization [15]. The above method also applies personalized prediction, which includes users' demographic profiles together with content information such as item-related web-browsing history. One of the most important developments of this method is its "decomposition and combination" strategy. Recommender systems usually have to process huge data sets. The method applies MapReduce to break large-scale optimization into many small sub-problems and solve them in a parallel manner.

However, a number of challenges in this method should also be considered. First, the method makes an assumption that the missing of values occurs completely due to randomness, which ignores that the missing itself is informative. By sub-grouping [18, 19], missing information can be utilized in making recommendations. Second, linearity restricts the performance of the model. Third, the user-item network, which is available, is ignored in the model. For example, the social network of a user consists of information of preference, but the model does not include the social network information. The importance of network is shown in Fig.1(b).

3.1.3 Smooth Neighborhood Recommender Systems

More recently, a smooth recommender system includes user-item specific network information and user and/or item specific covariates, which highly improves the accuracy [20]. One of the most important strategies of the method is to pool information across user-item pairs to improve the accuracy. The method defines $S_{u'i'}^{ui} = S_{u'}^{u} S_{i'}^{i} = 1$ as an indicator of existence of edges connecting u and u' in a user-specific network and i and i' in an item-specific network, whereas $S = 0$ indicates no existence. Driven from the definition of root mean square error (RMSE) and intuition from SVD model, the objective function for a smooth neighborhood recommender system is shown by $L(P, Q)$:

$$RMSE(\hat{\theta}, \theta^0) = (\frac{1}{nm} \sum_{u=1}^{n} \sum_{i=1}^{m} (\hat{\theta_{ui}}, \theta_{ui}^0)^2)^{1/2} \tag{5}$$

$$L(P, Q) = \frac{1}{nm} \sum_{u=1}^{n} \sum_{i=1}^{m} (\sum_{(u',i')\in\Omega} w_{ui,u',i'} (r_{u'i'} - p_u^T q_i)^2) + \lambda_1 \sum_{u=1}^{n} J(p_u) + \lambda_2 \sum_{i=1}^{m} J(q_i) \tag{6}$$

where J is a penalty, λ_1 and λ_2 are two tuning parameters, and $w_{ui,u',i'}$ is the weight introduced to define smooth neighborhood. The kernel function of weight is defined as the following:

$$K_h(x_{ui}, x_{u'i'}) = K(h^{-1}||x_{ui} - x_{u'i'}||_2) \tag{7}$$

The kernel function 7 only calculates the observed data with $S_{u'i'}^{ui} = 1$. The degree of smoothness 2.2 of the kernel reflects how the preference varies in terms of $||x_{ui} - x_{u'i'}||_2$ told from the specific user and item network. The smooth neighborhood is constructed based on continuous and discrete covariates as well as user-item specific networks.

The technique has the following advantages. Compared to the regularized SVD method, which ignores the network information, the smooth neighborhood method integrates user-item specific covariates and network information to reduce the missing value. Second, the "cold-start" problem is solved by the user and item specific network. The information of network provides initial value to the preference, which would yield more accurate estimators of p_u and q_i.

One major limitation of the method is the difficulty of guaranteeing the user and item specific network to be unbiased. For instance, the paper uses the music data from Last.fm (http://www.last.fm), which only links to part of users' network and the tags that the users choose at the time of registration may not have good reliability after a long time. Aside from the reliability of the network, the accessibility of social networks information should not be assumed for more general situations.

3.2 Neural Classification-based Collaborative Filtering (NCCF)

Aside from the Matrix Factorization (MF) model, neural networks also show good performance in recommender systems and some neutral networks architectures are based on MF process. Both the MF model and the Neural Collaborative Filtering (NCF) model [21, 22] are based on regression. Neural Classification-based Collaborative Filtering (NCCF) [23], which is a rather novel technique, instead, is based on classification neural networks, which is not based on MF process. Another difference is that the result of NCCF is represented as possibility rather than true-or-false prediction. The difference among NCCF, NCF, and other related techniques is shown in Fig.2(b).

The Neural Classification-based Collaborative Filtering (NCCF) trains the most possible recommendations. The process to predict users' preferences is shown in Fig.3. NCCF input data is made up of user vectors, with each vector containing item information, but it does not combine the vectors of the user and those of the item. The input data will be marked as binary relevant/non-relevant vote and voted/ non-voted item information. The model will then learn by these categorical labels. A sample will be created when the label relevance is marked as relevant. The model will then be trained with the relevant ratings of each user and repeat the process for the user until every relevant rating of the user has been traversed. The NCCF model will select N recommendations based on the N highest probabilities.

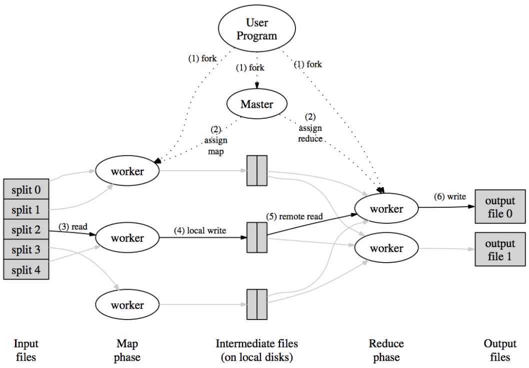
The NCCF model shows good performance in recommender systems. To be specific, the advantages are as follows: First, the model maintains a high accuracy when it keeps the size and complexity smaller than those of other frameworks, which is especially important for recommender systems due to the huge data. Second, the model can help users discover new interests, which also grants the model with another advantage of solving the "cold-start" problem pretty well since it has better starting points.

The model also has its limitations. A major limitation is that the model cannot handle fresh data [24]. Consider the recommendation systems of YouTube, in which video concludes most information of users' preferences, the model can only deal with information from words and images, but not the video. More generity can be introduced if the interaction of the model and the user can be improved.

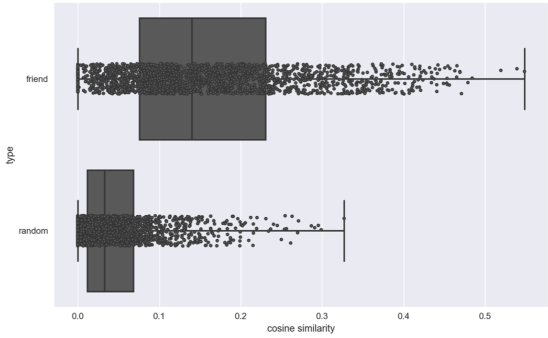
4 Conclusion

In conclusion, based on the frameworks of Collaborative filtering content-based filtering, the "cold-start" problem of training recommender systems reaches higher accuracy. Open-source programming framework [25] also enables us to apply multimodal recommender systems based on these frameworks. This review discussed modifications of the two frameworks and novel models in this field, which improved performance over previous approaches, but there is more to be explored, more to be created in this field.

Figures

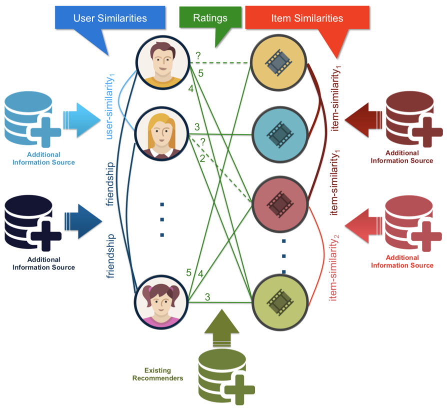


(a) MapReduce Execution overview [6]

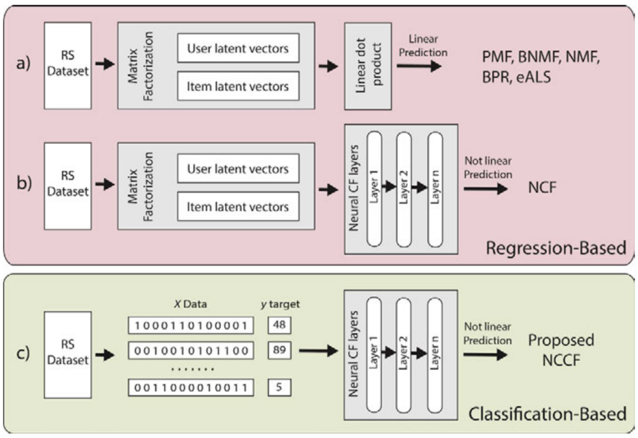


(b) Comparison of the cosine similarity of logarithm of users vector between friends and randomly selected [20].

Figure 1: Recommendation systems relationship and network influence.



(a) Relationship between users and items in recommender Systems [26].



(b) Comparison among NCCF model, NCF model and Matrix Factorization methods [23].

Figure 2: Comparison among methods and neural network architecture

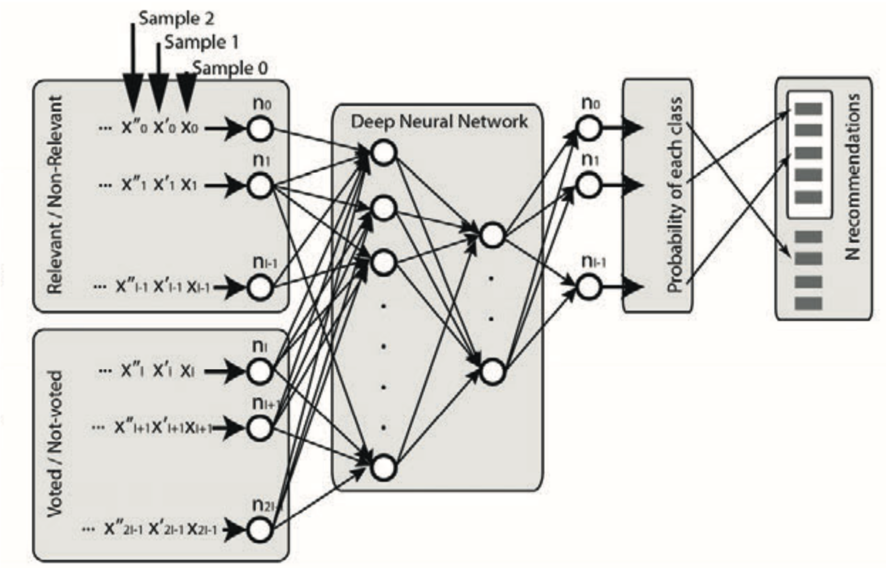


Figure 3: Neural network architecture of Neural Classification-based Collaborative Filtering method [23]

References

[1] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[2] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Systems with Applications*, vol. 41, no. 4, pp. 2065–2073, 2014.

[3] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme, "Learning attribute-to-feature mappings for cold-start recommendations," in *2010 IEEE International Conference on Data Mining*, pp. 176–185, IEEE, 2010.

[4] A. Mukherjee, K. Chen, N. Wang, and J. Zhu, "On the degrees of freedom of reduced-rank estimators in multivariate regression," *Biometrika*, vol. 102, no. 2, pp. 457–477, 2015.

[5] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1273–1284, 2014.

[6] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[7] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, 2009.

[8] R. M. Bell and Y. Koren, "Scalable collaborative filtering with jointly derived neighborhood interpolation weights," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pp. 43–52, IEEE, 2007.

[9] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," *Aaai/iaai*, vol. 23, pp. 187–192, 2002.

[10] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial intelligence review*, vol. 13, no. 5-6, pp. 393–408, 1999.

[11] L. Qiu, S. Gao, W. Cheng, and J. Guo, "Aspect-based latent factor model by integrating ratings and reviews for recommender system," *Knowledge-Based Systems*, vol. 110, pp. 233–243, 2016.

[12] A. Feuerverger, Y. He, and S. Khatri, "Statistical significance of the netflix challenge," *Statistical Science*, pp. 202–231, 2012.

[13] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[14] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," 2007.

[15] Y. Zhu, X. Shen, and C. Ye, "Personalized prediction and sparsity pursuit in latent factor models," *Journal of the American Statistical Association*, vol. 111, no. 513, pp. 241–252, 2016.

[16] S. Bag, S. Kumar, A. Awasthi, and M. K. Tiwari, "A noise correction-based approach to support a recommender system in a highly sparse rating environment," *Decision Support Systems*, vol. 118, pp. 46–57, 2019.

[17] T. Hastie, R. Mazumder, J. D. Lee, and R. Zadeh, "Matrix completion and low-rank svd via fast alternating least squares," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 3367–3402, 2015.

[18] X. Bi, A. Qu, J. Wang, and X. Shen, "A group-specific recommender system," *Journal of the American Statistical Association*, vol. 112, no. 519, pp. 1344–1353, 2017.

[19] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "Grouplens: applying collaborative filtering to usenet news," *Communications of the ACM*, vol. 40, no. 3, pp. 77–87, 1997.

[20] B. Dai, J. Wang, X. Shen, and A. Qu, "Smooth neighborhood recommender systems," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 589–612, 2019.

[21] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, pp. 173–182, 2017.

[22] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434, 2008.

[23] J. Bobadilla, F. Ortega, A. Gutiérrez, and S. Alonso, "Classification-based deep neural network architecture for collaborative filtering recommender systems.," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 6, no. 1, 2020.

[24] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, pp. 191–198, 2016.

[25] A. Salah, Q.-T. Truong, and H. W. Lauw, "Cornac: A comparative framework for multimodal recommender systems," *Journal of Machine Learning Research*, vol. 21, no. 95, pp. 1–5, 2020.

[26] P. Kouki, S. Fakhraei, J. Foulds, M. Eirinaki, and L. Getoor, "Hyper: A flexible and extensible probabilistic framework for hybrid recommender systems," in *Proceedings of 9th ACM Conference on Recommender Systems*, pp. 99–106, 2015.