A Review on Matrix Factorization Techniques in Recommender Systems

Ms. Rachana Mehta
Department of Computer Engineering
Sarvajanik College of Engineering and Technology
Surat, India - 395001
Email: rachana.mehta19@gmail.com

Dr. Keyur Rana
Department of Computer Engineering
Sarvajanik College of Engineering and Technology
Surat, India - 395001
Email: keyur.rana@scet.ac.in

***Abstract*—Growth of the Internet and web applications has led to vast amount of information over web. Information filtering systems such as Recommenders have become potential tools to deal with such plethora of information, help users select and provide relevant information. Collaborative Filtering is the popular approach to recommendation systems. Collaborative Filtering works on the fact that users with similar behavior will have similar interests in future, and using this notion collaborative filtering recommends items to user. However, the sparseness in data and high dimensionality has become a challenge. To resolve such issues, model based, matrix factorization techniques have well emerged. These techniques have evolved from using simple user-item rating information to auxiliary information such as time and trust. In this paper, we present a comprehensive review on such matrix factorization techniques and their usage in recommenders.**

***Keywords*—Recommendation System, Collaborative filtering, Matrix factorization, Singular Value Decomposition**

I. Introduction

The growth in the usage of Internet and web applications for electronic transactions, business transactions and for seeking information regarding education, shopping and social activities have led to enormous amount of information on web. Electronic retailers provide users with huge selection of products, with unprecedented opportunities that meets variety of needs. [1]. However, such plethora of information causes information overload problem [2], making difficult for users to locate their information of interest. In this context, information filtering systems such as recommenders have shown their prominence by providing users with personalized recommendation of items that suits their taste [3] and providing information they seek.

Recommendation Systems (RS) not only find products or items of user's interest, but are also useful to find people, friends or events of interest. Recommenders have found wide applicability in many e-commerce and electronic companies. Some of the popular ones, include Netflix's Movie Recommendation [1], Amazon's Product Recommendation [4], Google's News Personalization, Google Search for Advertisements [5], YouTube for video [5] and Last.fm for Music [5].

Recommendation systems are mainly used for two main tasks, prediction generation and recommendation generation [5]. The prediction generation involves predicting whether a user will like an item or not and predicting how much a rating item will receive from a particular user. The recommendation generation task involves recommending a set or a bunch of items to a user.

Recommendation systems collect information regarding user's past behavior on a set of items and use them for recommendation. The information can be explicit information (such as user's ratings), implicit information (like browsing history), demographic, social (like friends, tags, trusters or trustees) or contextual information (like time or location) [6]. Recommenders are evaluated on how accurate they are, in providing recommendations or generating predictions. However, achieving the accurate results is a difficult task, as most of the recommenders suffer from data sparsity and cold start issues [7]. Also, they don't work efficiently with high dimensional data [8] and are less scalable [3]. Several recommendation techniques have been evolved over the time to overcome such issues. Amongst which, the techniques based on Collaborative Filtering (CF) approach have gained popularity due to their wide applicability to different items. However, not all CF techniques are efficient in handling high dimensional data. Model based, Matrix Factorization (MF) techniques are an exception, they provides dimensionality reduction and works with sparse data. In this paper, we go through various matrix factorization techniques, especially Singular Value Decomposition and explore how they have grown from using simple user-item rating information to auxiliary time or social information, in order to resolve recommendation issues.

The rest of the paper is organized in following manner: Section 2 gives a brief overview on various recommendation approaches and model learning. Section 3 epitomizes various matrix factorization techniques. Section 4 provides the relative comparison of matrix factorization techniques discussed. At last, Section 5 concludes the survey and give outline for the future research direction.

II. Background

In this section, we review the existing recommendation system approaches, followed by how model learning is done.

A. Recommendation System Approaches

Recommendation system approaches are classified into three main categories, on the basis of how recommendations are carried out [2], [3], [6], [7]. They are Content-Based Filtering, Collaborative Filtering and Hybrid approach. The hierarchy of the same is shown in Fig. 1.

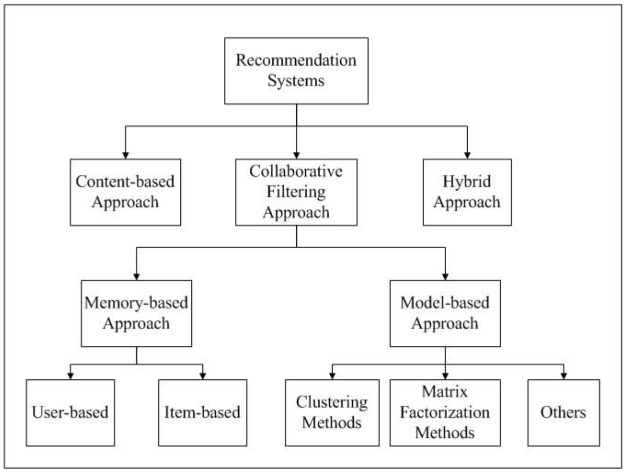

Fig. 1. Recommendation System Approaches

In Content Based Filtering approach, the user is recommended items or products similar to the one user has preferred in the past [3], [9]. The content that is used for making recommendations can either be graphical image, audio stream, video stream, tags or text [6]. However, the content information is not easily machine analyzable [7]. The recommendations in CBF often suffer from Overspecialization [3] and are difficult to evaluate [7]. Due to such issues, CBF approaches are less preferred.

The second approach to recommendation is Collaborative Filtering. Tapestry [10], a mail filtering system, was the foundational base of collaborative filtering. CF works on the fact that users with similar behavior will have similar taste or similar buying habits. As in CF, the ratings of similar users (neighbors) or items are used for making recommendation, there are no issues pertaining to machine analyzable content. Also, CF mitigates problems regarding evaluation [7] and Overspecialization [3]. This makes CF approaches to be widely used in variety of recommenders.

Collaborative filtering is further divided into memory and model based approach, based on how the data of rating matrix are processed [11]. Memory-based approach use similarity measures between users (or items) to find their relevant neighbors [10], [12]. Then use these neighbors for recommending or predicting items (or users). Memory based approach provide easy implementation and interpretation of results. However, the need of whole rating matrix for tasks makes them less preferable for high dimensional and sparse data.

Model based approach, on other hand, learns and fits a parameterized model to the user-item rating matrix. Then use that model for providing recommendation tasks [11]. Model based techniques include Clustering techniques like K-means [13], regression based like Slope-One [14], Matrix Factorization [1], [8], [15], Bayesian classifiers [16] and many others [17]. Amongst them, MF based models have gained wide popularity due to the Netflix Prize Contest [18]. They have emerged well because of their relatively high accuracy, scalability and dimensionality reduction property.

Although CF have achieved wide usage in recommenders they suffer from data sparsity, cold start and high dimensionality issue, which eventually degrades the performance. MF techniques are good at dealing with high dimensional data and mitigating sparsity. To alleviate data sparsity, several techniques have been proposed that leverage use of auxiliary information such as time or social trust. In this paper, we explore and provide theoretical survey on matrix factorization techniques that have emerged on line of using simple user-item rating information to auxiliary information, in order to improvise recommendation performance.

The third approach to recommendation is hybrid approach. It is a combination of Collaborative and Content based filtering. Hybrid systems provide high predictive accuracy then both [11]. However, it inherits additional complexity needed to deal with both approach.

B. Model Learning Phases

In Collaborative Filtering, the process of providing recommendations or predictions start from data collection, followed by data pre-processing and then applying appropriate CF techniques. The model based CF approach fits a parametric model for providing recommendations or predictions [2]. The process of model learning is carried out in 5 phases. The first phase is hypothesis space, it includes generation of a model, i.e. a prediction rule. This rule is generated from the pre-processed input data. The second phase and third phase is to generate an objective function and applying regularization to it. Objective function is used to learn the model parameters. The objective function can either be minimization (loss) or maximization function, depending on how parameter learning is done. In case of recommendation, it is often a minimization function. Several objective functions include logistic loss function, hinge loss function, square loss and absolute loss function [19]. In order to avoid data overfitting problem in model, regularization of parameters is carried out and added up to form overall objective function. The regularization can be simple (L2) or weighted (L2W) regularization [20], [21].

In fourth phase, an optimization strategy is applied to the objective function, so as to estimate best values for model parameters. The optimization strategy can either be heuristic based such as Genetic Algorithm, Simulated Annealing or iterative such as Gradient Descent or Alternating Least Squares Methods [20] or any other. Iterative strategies are more preferred due to their early convergence. After the model parameters are estimated, the last step is to validate and test the model. This includes evaluation metrics, validation methods such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Precision, Recall, Cross Validation and Hold Out Validation techniques [22], [23], [24].

III. Matrix Factorization Techniques

Matrix factorization have become potent solution to reduce the dimension of data, extract latent features of the data and mitigate sparsity. They are used widely in recommenders owing to their characteristics. The matrix factorization techniques used in recommenders include Singular Value Decomposition (SVD) [8], Principal Component Analysis (PCA) [15], Probabilistic Matrix Factorization (PMF) [25] and Non-negative Matrix Factorization (NMF) [28]. This factorization models can be formulated as optimization problem with objective function and constraints.

An approach to use PCA in recommenders was first proposed for an online joke recommendation system [15]. PCA generates co-variance matrix of user-item rating matrix and then performs dimension reduction. However, the inherent property of PCA to use symmetric matrix, have led to its limited usage.

PMF [25] treats ratings as a probabilistic graphical model. It provides a probabilistic approach using Gaussian distribution noise on the known data and the factor matrices. Such model fits the rating data in an approximate manner. Bayesian PMF (BPMF) was proposed [26] to overcome the model selection drawback of PMF. BPMF approximate posterior distribution using Markov Chain Monte Carlo based Gibbs sampling inference method. However, the use of multivariate Gaussian distribution prior dominated it's cubic time complexity, leading it to be less preferable for large datasets. To overcome this, a scalable approach Scalable Bayesian Matrix Factorization was proposed [27]. It considered independent univariate Gaussian prior and had linear time and space complexity. However, these probabilistic model are limited in their usage of auxiliary side information.

NMF [28] is a matrix factorization technique that imposes constraints of non-negativity on factor matrices. NMF works on fact, that all real world data like images or videos are non-negative. SVD is a realization of NMF with no constraints on sign of data. In this section, we further discuss SVD and its variants.

A. Singular Value Decomposition [8]

Singular Value Decomposition [8] technique was first used in 2000 for Recommendation systems. SVD reduces the dimensionality of a user-item rating matrix and generate low rank matrix approximations, which represents the latent features (hidden features) of users and items inherent in rating matrix. For example, latent features can be price or brand for product, genres for movies and so on. The low rank approximations are further used for recommendation tasks for a new and unknown user-item rating. The main challenge in SVD lies in finding a lower dimensional rank matrix.

Given a a x b user-item rating matrix R and matrix rank parameter r, SVD factorizes R into three low-dimension matrix, U $(a$ x $r)$, S $(r$ x $r)$ and V $(r$ x $b)$.

$$R = USV^T \tag{1}$$

Here, U and V are orthogonal matrix and S is a singular diagonal matrix. The entries of matrix S are positive and stored in descending order of their magnitude. Since, some of the values are extremely small, they can be ignored and by ignoring such values, the dimensions of matrix are reduced. The dimension reduction is done by reducing the matrix of rank-r to rank-n, where $n<r$. The reconstructed matrix which is closest rank-n approximation of original rating matrix is given in equation 2.

$$R_n = U_n S_n V_n^T \tag{2}$$

Thus, applying SVD in this manner performs dimensionality reduction. The prediction and recommendation task is carried out by first imputing missing values in rating matrix and then performing matrix factorization. The SVD [8] provided better performance than memory based approach, on e-commerce and movie data [8]. However, the imputation causes distortions in data and makes the recommendation model less accurate. Also, this imputation incurs data overfitting problem and are significantly expensive, as they increase the volume of data.

B. Regularized SVD [29]

Regularized SVD (RSVD) [29], proposed by Simon Funk for Netflix prize contest [18], was popularized for its ability to overcome data overfitting problem and incorporation of gradient descent optimization strategy. RSVD is one such extension of SVD, which works on observed user-item rating values, so that no data distortions are caused by missing data imputations, as was in the case of SVD [8].

RSVD working with explicit user-item rating data, first factorizes the original rating matrix to obtain latent feature matrices of users and items. The prediction rule of RSVD is given in equation 3. Here, p_u and q_i are latent feature vectors of user u and item i. P'_{ui} is the predicted value for rating of item i by user u

$$P'_{ui} = {p_u}^T q_i \tag{3}$$

The latent feature matrices of model are learned using square loss objective function [19]. The objective function formed is incorporated with the regularization terms of latent features in order to avoid the data overfitting problem. The regularization used in RSVD is simple L2 regularization [21], it penalizes all the parameters with same amount (value). The parameters of the model are estimated using gradient descent optimization strategy with early stopping criteria and one feature learning at a time. RSVD has shown good accuracy improvement for Netflix data [18]. However, it doesn't handle inherent systematic tendencies.

C. Improved Regularized SVD [23]

In collaborative filtering, the user-item rating data exhibits systematic tendencies for some items to receive lower rating values than other items, and some users to give lower ratings than other users or vice-versa [20]. These tendencies are known as biases or intercepts, they are independent of any interactions, but causes variations in rating values [23]. Improved regularized SVD (RSVD2) [23] was proposed to inculcate such biases and handle data variations, which RSVD didn't [29].

In RSVD2, the system is formed in such a manner that it identifies the part that biases explain. While, only true user-item interactions needs modeling. The parametric form of prediction rule of RSVD2 is:

$$P'_{ui} = b_u + b_i + (p_u^T q_i) \tag{4}$$

Here, b_u is observed deviation of user *u* and b_i is observed deviation of item *i*. The model learning and parameter estimation is carried out using squared loss function [19] with weighted regularization (L2W) [21] and gradient descent optimization strategy. RSVD2 has shown better predictive accuracy than RSVD on Netflix dataset [18]. Thus, RSVD2 handles variations due to biases and overcomes data overfitting problem when working on explicit rating information. RSVD and RSVD2 are pioneering work for most of SVD based recommenders. However, their usage is limited as they are confined by explicit rating data, even in sparse environment.

D. SVD++ [1]

Recommendation systems often suffer from data sparsity and cold start issues. Limited explicit information about user-item rating interactions, as in SVD, RSVD and RSVD2, don't give much efficient recommendation performance. An approach to mitigate these problems is to consider additional information. Implicit feedback's like browsing or purchasing history are among such additional source of information. SVD++ [1] is one such model that incorporates both explicit and implicit feedback information to improvise recommendation performance.

The SVD++ [1] has considered Boolean implicit feedback, which is whether a item has been rated by a user or not. The parametric form of SVD++ is shown in equation 4.

$$P'_{ui} = b_{ui} + q_i^T (p_u + \mid N(u) \mid^{-\frac{1}{2}} \sum_{j \in N(u)} y_j) \tag{5}$$

Here, $b_{ui} = b_u + b_i + \mu$, is baseline estimate of biases. $N(u)$ is set of items for which user *u* has implicit preference and y_j is item factors.

The model learning and parameter estimation is carried out similar to RSVD model. SVD++ provides good accuracy improvement over RSVD2 on Netflix dataset [18] by incorporating implicit feedback. SVD++ model is suitable when ratings are the only source of information available.

E. TimeSVD++ [30]

The matrix factorization models so far discussed are static, as they don't consider temporal information about user-item rating data. The perception of product and popularity changes over the time, as new products emerges. In a similar manner, user's taste evolve over the time. In such scenarios, there is a need of model that incorporates temporal information, reflects the dynamic and time drifting nature of user-item rating interactions. TimeSVD++ [30], proposed in 2010, is such an improvement over SVD++ [1].

The parameters that vary over the time are user biases, item biases and user's latent feature vectors. Whereas, items feature vector remains static over the time. The parametric form of TimeSVD++ is similar to that of SVD++, except that the parameters are time biased. TimeSVD++ model has provided better accuracy than SVD++ model on Netflix dataset [1]. However, the time required to train the model and learn parameters is more as compared to SVD++ [30].

F. TrustMF [31]

TrustMF [31], a variant of RSVD was proposed in the direction of using additional information to enhance recommendation performance. TrustMF uses social trust relationship between users as auxiliary information along with explicit rating information. Trust is an important social information as users are more likely to accept viewpoints of users they trust [31]. Trust networks are seen in online recommenders like Epinions [32], CIAO [33] and movie recommenders like Flixster [34].

The users in TrustMF are mapped to two low-dimensional space, truster and trustee, by factorizing trust network based on the directional property of trust [31]. The truster and trustee space are then individually modeled to obtain latent feature of users and items. The truster and trustee specific models are shown in Fig. 2 and 3.

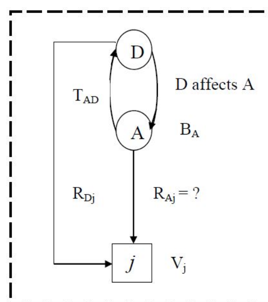

Fig. 2. Truster Model, how others affect user A's rating on item *j* [31]

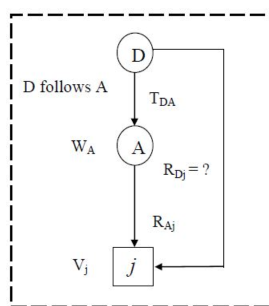

Fig. 3. Trustee Model, how others follow user A's rating on item *j* [31]

This truster and trustee model defines how much other users (*D*) follow and affect the rating (*R*) of a target user (*A*). B_A

TABLE I
COMPARISON OF MATRIX FACTORIZATION TECHNIQUES

Technique-Year	Explicit rating information	Implicit rating information	Temporal information	Explicit Trust information	Implicit Trust information	Data imputation	Biases used	Regularization used	Dynamic
SVD [8], 2000	Yes	No	No	No	No	Yes	No	No	No
RSVD [29], 2006	Yes	No	No	No	No	No	No	Yes	No
RSVD2 [23], 2007	Yes	No	No	No	No	No	Yes	Yes	No
SVD++ [1], 2008	Yes	Yes	No	No	No	No	Yes	Yes	No
TimeSVD++ [30], 2010	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes
TrustMF [31], 2013	Yes	No	No	Yes	No	No	No	Yes	No
TSVDTU [36], 2015	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No
TrustSVD [37], 2016	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No

and W_A are truster and trustee feature vector and V_j is item specific vector. The user-user trust is represented as a binary matrix *T*, where '1' represents user 1 trusts user 2 and '0' shows distrust.

Truster model provides truster specific vector and item vector specific to truster. Similarly, trustee model provides trustee specific vector and item vector specific to trustee. The features of both the model are fused to form synthetic model of TrustMF. The model learning and parameter estimation is done using squared loss function [19] with weighted regularization [21] and gradient descent optimization [20] strategy.

TrustMF, though incorporating auxiliary information didn't provided accuracy improvement over SVD++ [1] when tested on Epinions data [35]. Also, it can be noticed that TrustMF used only explicit information of ratings, while TimeSVD++ [30] used both explicit and implicit ratings information.

G. TrustSVD with Trusted Users [36]

TrustSVD with Trusted Users (TSVDTU) [36] works on the same notion of incorporating trust information along with user-item rating information to generate predictions. The difference in TSVDTU, compared to TrustMF, lies in the fact that it considers implicit rating information and implicit trust. The implicit trust considered here is binary, which is whether a user A trusts user B or not. TSVDTU also considers in the biases terms to handle data variations. TSVDTU, however, leverage the use of trusted users only and not trustee users.

The TSVDTU model is similar to SVD++ [1], with just the addition of truster based parameters while factoring users. The objective function in TSVDTU is carried out using squared loss function [19] and Adaptive regularization, which is an extension of weighted regularization (L2W). In Adaptive regularization popular users and items are less penalized while cold start users and niche items are more penalized [36]. The parameter estimation is done using gradient descent optimization strategy.

TSVDTU model provided better accuracy than SVD++ [1] and TrustMF [31] on trust dataset of Epinions and CIAO [35], [33], just by incorporating implicit information and truster knowledge. Though, the model is still static.

H. TrustSVD [37]

Owing to the improvement seen in trust based models like TSVDTU, Guo et. al. proposed TrustSVD [37], in 2016. TrustSVD extends TSVDTU by incorporating both truster and trustee specific features. TrustSVD infuses truster and trustee specific feature vectors while factoring users along with implicit and explicit user-item rating information. TrustSVD also incorporates trust parameter in model formation so as to control the amount of influence truster and trustee have while predicting ratings. The truster and trustee model are similar to the one in TrustMF [31].

The objective function formulation here, is same as that of TSVDTU, squared loss function [19] with adaptive regularization [36]. The parameter estimation is carried out using gradient descent optimization [20]. The incorporation of social trust information has been beneficial to TrustSVD, as it resulted in achieving higher accuracy than almost all models, so far discussed, on trust dataset [33], [35]. Thus, it can be seen that addition of auxiliary information can significantly enhance recommendation performance. However, trust models are static in nature, as they don't incorporate time drifting behavior, as was leveraged by TimeSVD++ [30].

IV. COMPARISON

Table I shows the comparative analysis of the matrix factorization techniques discussed so far. The first few parameters represent the type of information used by MF model to carry out prediction. This information includes explicit ratings, implicit ratings, temporal information, implicit and explicit social trust information. In order to alleviate data sparsity issues of recommendation systems, the natural approach is inclusion of additional auxiliary information. Relative to that, it can be seen from the table I how the SVD based techniques have emerged in usage of auxiliary information. The other possible solution to handle data sparsity issues is missing data imputations. It can be seen from the table I that only SVD [8] technique carry out data imputations. While the rest of the techniques use observed rating information only, they work on the fact that data imputation causes distortions which makes model less accurate.

The other comparison parameters are based on whether the technique handles biases, uses regularization and is dynamic or not. The user-item rating data in collaborative filtering often exhibit systematic tendencies for some items and users. It is necessary to handle such bias variations so that only true ratings interactions are modeled. We can infer from table I that most of the recent techniques handles biases and their variations.

The main task of model based approaches is to make the model fit for training, so that it provides accurate predictions on untrained data. In order to achieve this, recommendation systems usually adds new parameters. However it leads to data overffitting problem. Regularization is a solution to avoid such problem. As seen in table I, most of the factorization techniques uses regularization. The last parameter, dynamic represents whether a particular MF model handles time variations or not. It is visualized that only one model does so.

V. Conclusion

Recommender systems are one of the potent solution, to deal with information overload problem. As a result, recommenders are used in variety of applications. Several recommendation approaches exist, amongst which, collaborative filtering have emerged due to their wide item applicability. However, they often suffer from issues pertaining to data sparsity, cold start and high dimensional data. In this paper, we explored various matrix factorization techniques that leverages use of simple user-item rating information to various auxiliary information, to mitigate such problems. These matrix factorization models have shown a good improvement in recommendation performance. In future, we will look forward to how multiple auxiliary information can be fused together to improve recommendation performance.

References

[1] Y. Koren, "Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model", Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 426-434, 2008.

[2] Z. Yang, B. Wu, K. Zheng, X. Wang and L. Lei, "A Survey of Collaborative Filtering-Based Recommender Systems for Mobile Internet Applications", in IEEE Access, vol. 4, no. 1, pp. 3273-3287, 2016.

[3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions", in IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734-749, 2005.

[4] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering", IEEE Internet computing vol. 7, no. 1 pp. 76-80,2003.

[5] C.C. Aggarwal, "An Introduction to Recommender Systems, In Recommender Systems, Springer International Publishing, pp. 1-28, 2016.

[6] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutirrez, "Recommender systems survey, Knowledge-Based Systems, Elsevier Journal, vol. 46 no.1, pp.109-132, 2013.

[7] F. Cacheda, V. Carneiro, D. Fernndez, and V. Formoso, "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems, ACM Transactions on the Web (TWEB), vol.5, no. 1, pp. 1-33, 2011

[8] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Application of Dimensionality Reduction in Recommender Systems-A Case Study", Proc. ACM WebKDD Workshop, pp. 1-12, 2000.

[9] C.C. Aggarwal, "Content-Based Recommender Systems, In Recommender Systems, Springer International Publishing, pp. 139-166, 2016.

[10] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry, "Using Collaborative Filtering to Weave an Information Tapestry, Comm. ACM, vol. 35, no. 12, pp. 61-70, 1992.

[11] J. Lee, M. Sun and G. Lebanon, "A comparative study of collaborative filtering algorithms", Arvix, pp. 1-27, 2012.

[12] B. Sarwar, G. Karypis, J. Konstan, J. and Riedl, "Item-based collaborative filtering recommendation algorithms, In Proceedings of the 10th international conference on World Wide Web, ACM, pp. 285-295, 2001.

[13] G.R. Xue, C. Lin, Q. Yang, W. Xi, H.J. Zeng, Y. Yu, and Z. Chen, "Scalable collaborative filtering using cluster-based smoothing, In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval , pp. 114-121, 2005.

[14] D. Lemire,and A. Maclachlan, "Slope One Predictors for Online Rating-Based Collaborative Filtering, In SDM , Vol. 5, no. 1, pp. 1-5, 2005.

[15] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A Constant Time Collaborative Filtering Algorithm, Information Retrieval J., vol. 4, no. 2, pp. 133-151, 2001.

[16] Y.-H. Chien and E.I. George, "A Bayesian Model for Collaborative Filtering, Proc. Seventh Intl Workshop Artificial Intelligence and Statistics, 1999.

[17] C.C. Aggarwal, "Model-based Collaborative Filtering, In Recommender Systems, Springer International Publishing, pp. 71-138, 2016.

[18] J. Bennett and S. Lanning, "The Netflix Prize", In Proc. of KDD Cup Workshop at SIGKDD'07, 13th ACM Int. Conf. on Knowledge Discovery and Data Mining, pp. 3-6, 2007.

[19] L. Rosasco, E. D. Vito, A. Caponnetto, M. Piana, and A. Verri. "Are loss functions all the same?.", Neural Computation, MIT Press vol.16, no. 5 pp. 1063-1076, 2004.

[20] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems", IEEE Computing, vol. 42, no. 8, pp. 30-37, 2009.

[21] C. Teflioudi, F. Makari and R. Gemulla, "Distributed Matrix Completion", IEEE 12th International Conference on Data Mining, pp. 655-664, 2012.

[22] D. Parra and S. Sahebi, "Recommender systems: Sources of knowledge and evaluation metrics", in Advanced Techniques in Web Intelligence-2, Springer, pp.149-175, 2013.

[23] A. Paterek, "Improving Regularized Singular Value Decomposition for Collaborative Filtering", Proc. ACM KDD Cup and Workshop, pp. 39-42, 2007.

[24] Y. Bengio and Y. Grandvalet, "No umbiased estimator of the variance of k-fold cross-validation", Journal of Machine Learning Research pp. 1089-1105, 2004.

[25] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization, in Advances in Neural Information Processing Systems (NIPS), vol. 20, pp. 1257-1264, 2008.

[26] R. Salakhutdinov and A. Mnih, "Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo", Proceedings of the 25th International Conference on Machine Learning, pp. 1-8, 2008.

[27] A. Saha, "Scalable Bayesian Factorization Models for Recommender Systems", [Online]. Available: http://www.cse.iitm.ac.in/ravi/papers/Avijit_thesis.pdf. [Accessed: 21- Feb- 2017].

[28] D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization", Nature, 401, pp.788-791, 1999.

[29] S. Funk, "Netflix Update: Try This at Home", Sifter.org, 2006. [Online]. Available: http://sifter.org/ simon/journal/20061211.html. [Accessed: 09-Oct- 2016]

[30] Y. Koren, "Collaborative Filtering with Temporal Dynamics", Proc. 15th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 447-455, 2010.

[31] B.Yang, Y. Lei and D. Liu, "Social Collaborative Filtering by Trust, in AAAI Conference on Artificial Intelligence, pp.2747-2753, 2013.

[32] "Epinions.com: Read expert reviews on Electronics, Cars, Books, Movies, Music and More.", Epinions.com, 2017. [Online]. Available: http://www.epinions.com/. [Accessed: 05- Jan- 2017].

[33] "Ciao! - Best reviews and opinions from real consumers - Join the community", Ciao.com, 2017. [Online]. Available: http://www.ciao.com. [Accessed: 05- Jan- 2017].

[34] "Flixster - Buy, Rent, and Watch Movies & TV on Flixster", Flixster.com, 2017. [Online]. Available: https://www.flixster.com. [Accessed: 05- Jan- 2017].

[35] "Downloaded Epinions Trust Network Datasets", Trustlet.org, 2016. [Online]. Available: http://www.trustlet.org/downloaded_epinions.html [Accessed: 02- Dec-2016].

[36] Guo, Guibing, Jie Zhang, and Neil Yorke-Smith, "TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings", In AAAI Conference of Artificial Intelligence, pp. 123-129. 2015

[37] G. Guo, J. Zhang and N. Yorke-Smith, "A Novel Recommendation Model Regularized with User Trust and Item Ratings", in IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 7, pp. 1607-1620, 2016.