

# Social Popularity based SVD++ Recommender System

Rajeev Kumar  
Student, M. Tech  
IET, Alwar, Rajasthan, India

B. K. Verma, Ph.D  
Associate Prof., CS Deptt.  
IET, Alwar, Rajasthan, India

Shyam Sunder Rastogi  
Student, M. Tech  
IET, Alwar, Rajasthan, India

## ABSTRACT

Recommender systems have shown a lot of awareness in the past decade. Due to their great business value, recommender systems have also been successfully deployed in business, such as product recommendation at flipkart, HomeShop18, and music recommendation at Last.fm, Pandora, and movie recommendation at Flixstreet, MovieLens, and Jinni. In the past few years, the incredible growth of Web 2.0 web sites and applications constitute new challenges for Traditional recommender systems. Traditional recommender systems always ignore social interaction among users. But in our real life, when we are asking our friends or looking opinions, reviews for recommendations of Mobile or heart touching music, movies, electronic gadgets, restaurant, book, games, software Apps, we are actually using social information for recommendations. In this paper social popularity factor are incorporated in SVD++ factorization method as implicit feedback to improve accuracy and scalability of recommendations.

## Keywords

CF Based Recommendation; SVD; Social Popularity

## 1. INTRODUCTION

Recommender systems have been used by various ecommerce websites for recommending product, item, movies, music etc. Recommender systems are essential tool in e-commerce on the Web [1]. Nowadays, they are being used by the lot of customer data in existing commercial databases, and more they are available at social networking websites that are most successful system for recommending is collaborative filtering based approach. In order to generate recommendations, Collaborative filtering systems need to compare basically different objects like items against users. There are two main approaches to help such a comparison that make the two main parts of Collaborative filtering approach: the neighborhood approach and latent factor models. Neighborhood methods are based on computing the relationships between users or items. Singular Value Decomposition (SVD) approach, latent factor models, which converting both users and items to the same latent factor that are comparable to each other. The incredible growth of customers and products due to social web and e-commerce websites creates two key challenges for recommender systems.

The first challenge is that how to improve the quality of the recommendations for the customers. If quality of the recommender system is good, then customers can trust a recommender; purchase a product, like to book a movies show and finds out he does not like the product, the customer will be unlikely to use the recommender system again. Second challenge is that how to improve scalability of the collaborative filtering algorithms. In somehow there are conflicts in these two challenges. If algorithm spends less time for searching a neighbors, it will be more scalable and worse

its quality. We need to consider these two challenges simultaneously so the solutions discovered are practical. We need a new technology that can be useful and dramatically enhance the scalability of recommender systems. Many researchers have suggested that Singular Value Decomposition (SVD) [2] may be such a technology in most of the cases. SVD-based approach can generate results that were much better than a traditional collaborative filtering algorithm most of the time when tested on a MovieLens dataset. But there are some serious limitations when we apply SVD-based approach for recommending which make its less suitable for large scale deployment in e-commerce system. The matrix factorization step is computationally very expensive because it takes a lot of memory and time to factorize a matrix. This is a major problem towards achieving a high scalability while producing good predictive accuracy. In real world e-commerce application, a large number of customers only buy or rate a very small percentage of products, which is real problem. Dimensionality reduction in recommender system is used due to these two problems. That will help to improve the precision of recommendations and reduce the complexity of real time computations.

The paper is organized as follows. The next section discusses dimensionality reduction algorithm. Section 3 explains about social popularity concept with integrated model. Section 4 presents experimental evaluation procedure and in Section 5 results and discussion. The conclusion and future research work are given in the last sections.

## 2. RELATED WORK

The aim of Collaborative filtering based approach is to recommend new products or to predict the value of a certain product for a particular user, based on the user's previous liking and the opinions of same type of users. Several successful systems have been implemented in various organizations like Amazon's, Netflix, and Last.fm. The weakness of Collaborative filtering based approach for large, sparse databases motivated us to investigate alternative recommender system algorithms. Latent Semantic Indexing (LSI) [3] is used to reduce the dimensionality of user-item ratings matrix. In information retrieval system, LSI is used to solve the problem of synonymy and polysemy. LSI uses singular value decomposition (SVD) as its fundamental dimensionality reduction algorithm, maps well into the collaborative filtering based recommender system challenges. However, SVD cannot be applied to explicit rating in the collaborative filtering based approach because user does not rate most of product so user-item rating matrix have lot of missing values. Furthermore, only few known entries may causes of overfitting [4]. Recent work says that we can fills in missing ratings values and make user-item rating matrix dense. But it is more expensive as compared to other method. Therefore, more recent works recommended that modeling with only the observed ratings, while avoiding over fitting through sufficient regularized model.

## 2.1 Singular Value Decomposition (SVD)

SVD is a **matrix factorization** method used to generate low rank approximations [5]. Given a  $a \times b$  matrix  $A$ , with  $a \geq b$  and rank  $k$ , the singular value decomposition, SVD ( $A$ ), is defined as

$$SVD(A) = U \times S \times V^T \quad (1)$$

Where  $U$ ,  $S$  and  $V$  are of dimensions  $a \times a$ ,  $a \times b$  and  $b \times b$  respectively.  $U$  and  $V$  are two orthogonal matrices. The first  $k$  columns of  $U$  and  $V$  represent the orthogonal eigenvectors associated with the  $k$  nonzero eigenvalues of  $AA^T$  and  $AA^T$ , respectively. In other words, the  $k$  columns of  $U$  corresponding to the nonzero singular values cover the column vector, and the  $k$  columns of  $V$  cover the row vector of the matrix  $A$ .  $U$  and  $V$  are known as the left and the right singular vectors, respectively. Matrix  $S$  is called a diagonal matrix (also known as singular matrix), which have only  $k$  nonzero entries. The diagonal entries ( $s_1, s_2, s_3, \dots, s_k$ ) of  $S$  have the property that  $s_k > 0$  and  $s_1 > s_2 > s_3 \dots > s_k$ . SVD gives the best low-rank linear approximation of the original matrix  $A$  which is an essential property that makes it particularly attractive for our application. It is likely to retain only  $m \ll k$  singular values by removal other entries. We call this reduced matrix  $S_m$  since the entries in  $S$  are sorted i.e.,  $s_1 > s_2 > s_3 \dots > s_m$  the reduction process is performed by retaining the first  $m$  singular values. The  $U$  and  $V$  matrices are reduced to produce matrices  $U_m$  and  $V_m$ , respectively. The matrix  $U_m$  is produced by removing  $(k - m)$  columns from the matrix  $U$  and matrix  $V_m$  is produced by removing  $(k - m)$  rows from the matrix  $V$ . When we multiply these three reduced matrices, we obtain a matrix  $A_m$ . The reconstructed matrix  $A_m = U_m \times S_m \times V_m^T$  is the closest approximation to the original matrix  $A$  of rank  $m$ . So,  $A_m$  minimizes the *Frobenius norm*  $\|A - A_m\|_F$  over all rank  $m$  matrices [6]. Researchers find out that the low-rank approximation of the original vector is better than the original vector itself due to the filtering out of the small singular values that introduce “noise” in the user-item relationship. The dimensionality reduction approach in SVD can be very valuable for the collaborative filtering process. SVD generates a set of uncorrelated eigenvectors [7]. Each user and item is represented by its corresponding eigenvector. The process of dimensionality reduction may help users who rated similar products (but not exactly the same items) to be mapped into the space spanned by the same eigenvectors.

**Prediction computation using SVD** Let the  $a \times b$  ratings matrix  $A$  is decomposed and reduced into three SVD component matrices with  $m$  features  $U_m$ ,  $S_m$  and  $V_m$ . The generation of prediction can be done by computing the **log likelihood similarities** [8] between  $a$  users  $U_m \cdot (S_m^{1/2})^T$  and  $b$  items  $(S_m^{1/2}) \cdot V_m^T$ . The prediction score  $P_{x,y}$  for the  $x$ -th user on the  $y$ -th item by adding the row average,  $\bar{r}_x$  to the similarity. **Formally**

$$\bar{r}_{x,y} = r_x + U_m \cdot (S_m^{1/2})^T(x) \cdot (S_m^{1/2}) \cdot V_m^T(y) \quad (2)$$

## 2.2 SVD with Implicit Feedback (SVD++)

SVD++, matrix factorization model, makes use of implicit feedback information. However, implicit feedback [9] is used to refer to any kinds of users' rental, purchase history information that can assist users' preference. Each user  $u$  is related with a user-factors vector  $p_u \in R^f$  each item  $i$  with an

item-factors vector  $q_i \in R^f$ . Prediction can be calculated by the rule:

$$\bar{r}_{ui} = b_{ui} + q_i^T p_u \quad (3)$$

Here,  $b_{ui}$  is baseline predictor for unknown rating  $r_{ui}$ . Baseline predictor [10] can be defined as  $b_{ui} = \mu + b_u + b_i + q_i$  and regularized squared error can be minimized in following way that can be performed by either stochastic gradient descent or alternating least squares.

$$\min \sum_{(u,i) \in m} (r_{ui} - \mu - b_u - b_i - q_i^T p_u)^2 + \lambda_4 (b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2) \quad (4)$$

Here  $\lambda_4$  is a constant that are usually determined by cross validation [10], which is used to control the extent of regularization. The MovieLens dataset does not only notify us the rating values, but also which movies users rate, despite the consequences of how they rated these movies. A user implicitly tells us about her preferences by choosing to say his or her opinion and vote a (high or low) rating. So rating matrix are reduced into a binary matrix, where “1” stands for “rated”, and “0” for “not rated”. Admittedly, this binary data [11] is not as huge and independent as other sources of implicit feedback could be. Nonetheless, including this kind of implicit data – which essentially exist in every rating based recommender system, improves prediction accuracy. This model is not limited to a certain kind of implicit data. To maintain simplification, each user  $u$  is related with two sets of items, one is denoted by  $r(u)$ , and contains all the items for which ratings by  $u$  are available. The other one, which denoted by  $N(u)$ , contains all items for that  $u$  provided an implicit preference. More accurate results can be found by integrating implicit feedback in SVD [12], leading to the following model:

$$\bar{r}_{ui} = \mu + b_u + b_i + q_i^T \left( p_u + \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} y_j \right) \quad (5)$$

Here  $N(u)$  is the set of implicit feedback (the set of items user  $u$  rated).

## 3. SOCIAL POPULARITY

### 3.1 Information Flow

Popularity is a social occurrence [13] but it can also demonstrate to things that people act as a team. To make something popular, group interest play an important role and apart from it something's popularity is raised very rapidly by information flow. Rankings for things, like movies and music, often do not tell the public's taste, rather than the taste of a small number of buyers or users because social influence plays a big role in determining what is popular and what is not through an information flow [14]. Independent of personal interest, the information flow acts as a strong influence, which causes individual to like the actions of others. Whether or not they are agree with that. When downloading music, people don't essentially decide for themselves what exact song to buy. Instead, they find out the list of most downloaded movies, songs, apps and come to a decision to get those same top things. Since people depend on another actions because they get to know by looking website's download rankings that what is top rated or used, later it become popular among public. Though, popularity is mainly constructed [15] as a general consensus of a group's attitude about things, but word-of-mouth is a new valuable technique to attract someone attention. Websites and blogs initiate by recommendations from one companion to another, as they go through social

web. This distribution by word-of-mouth [16] is the social information flow that allows something to grow in practice and attention throughout a social group until one person tells to another person about it, so it's become more popular.

### 3.2 Zipf's Law

Zipf's power law [17] states that high quantities have a low frequency and low quantities have high frequency. This can be used to measure popularity of products, person or things. This can also be seen in social web, such as Facebook that most number of users has about 130 friends, while very few users have larger social networks [18]. However, some persons do have more than 6,000 friends. This tells us that very few persons can be really well-connected, but many users are not well connected. The number of friends one has been a way to decide how popular a person is, so the few person who have an very high number of friends is a way of using social web, like Facebook, to show how only a few people are more popular.

Suppose that we have the number of elements is  $N$ ; its rank is  $k$ ; exponent [19] characterizing the distribution is  $s$ . Popularity of product of  $N$  elements, the occurrence of elements of rank  $k$  can be predicted as

$$f(k; s, N) = \frac{1/k^s}{\sum_{n=1}^N (1/n^s)}. \quad (6)$$

Power law distributions are given by  $p(f) = (\alpha f^{-1-1/s})$ .

### 3.3 An Integrated Model

Social popularity [25] model are integrated into SVD++, which provide better accuracy as compared to SVD++. Novel modified model are given below.

$$\bar{r}_{ui} = \mu + b_u + b_i + q_i^T \left( p_u + \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} y_j + \frac{1}{\sqrt{|P(i)|}} \sum_{i \in P(i)} X_i \right) \quad (7)$$

$P(i)$  is the set of implicit feedback (the set of item that are rated by most users). This, popularity information exists in user-item matrix as those items that are used most of the user. Implicit feedbacks are inserted as who have rated to which item. So  $N(u)$  is the set of item user  $u$  rated [21] as given in equation (5). Similarly there can be a set of items that are rated by most of user. So the list of most rated item can be made and include it as an implicit feedback in decreasing order and recommendations are provided from highest popularity to lowest popularity. This is significantly improving accuracy of recommendation.

## 4. EXPERIMENTAL EVALUATIONS

This section investigates our experimental proof of the social popularity based SVD++ algorithm. Experimental platform such as the dataset and evaluation metric are described firstly, then experimental procedure followed by the results, discussion and future work.

### 4.1 Dataset

MovieTweatings [22] dataset consists of ratings on movies that were contained in well-organized tweets on Twitter. This dataset is the result of research done by [Simon Doods] public rating datasets, like MovieLens or Netflix, have been admired and widely used in the recommender systems domain for carrying out tests and evaluation. However they are becoming outdated and fail to integrate new and relevant

items. In this work, due to huge availability of public information on social media, new movie rating dataset MovieTweatings are created, that are based on public tweets exist on twitter which add around 500 or more new ratings per day. This dataset will be valuable because it is always up-to-date. This dataset will be updated as much as possible to include rating data from the newest tweets available. The earliest rating enclosed in this dataset is from 28 Feb 2013. This dataset is automatically gathered [23] through Twitter API and IMDb apps. Users are selected randomly to obtain 150, 000 ratings from the database. The data set was converted into a user-movie matrix  $A$  that had 786 rows (users) and 882 columns (movies). For experiments, data set are divided into training and a test portion [24]. Training and test data ratio are varied by using a parameter  $x$ , where  $x = 0.9$  means that 90% data was used for training the algorithm and 10% was used as test.

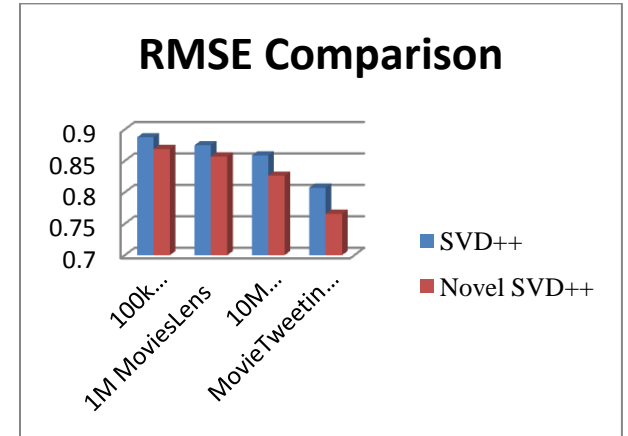
### 4.2 Evaluation Metrics

A recommender system has used different types of measures for evaluating [27] the success of a recommender system. The quality of prediction or recommendation are most important because user's only interested in the output of a recommender system that why Root Mean Squared Error (RMSE) are used for evaluation purpose.

$$RMSE = \sqrt{\sum \frac{(R_{u,i} - r_{u,i})^2}{N}} \quad (8)$$

The lower RMSE denote that recommender system are predicting user rating more accurately, so finally recommendation will be more accurate.

## 5. RESULTS AND DISCUSSION



**Fig 1: RMSE Comparison**

Firstly tests are performed for SVD++ approach on MovieLens Dataset and then MovieTweatings dataset. RMSE has been noted down RMSE on 100k, 1M and 10M MovieLens dataset and finally using MovieTweatings dataset [26]. SVD++ based factorizer work well with RMSE 0.88746 on 100k size dataset, RMSE 0.87486 with 1M dataset and RMSE 0.85873 with 10M dataset and it take less time to recommendation for small size dataset with less prediction accuracy [27] but as data size increases then it take more time to recommendation but better prediction accuracy. When prediction generations are good then recommendation must be better with lower RMSE. MovieTweatings dataset also work well with RMSE 0.80743 which is proving much better result compared to MovieLens dataset. Finally Social Popularity based SVD++ recommender system are tested on both dataset



and RMSE 0.86879 on 100k, RMSE 0.85685 on 1M, RMSE 0.82643 on 10M MovieLens dataset are noted down. This shows improvement in RMSE [28] as well as prediction accuracy. When experiments have been performed with MovieTweets dataset then it gives RMSE 0.76583. There is much improvement as compared with previous dataset that can be seen from result given here.

## 6. CONCLUSION AND FUTURE WORK

Recommender systems are a leading technology which is used to enhance business value by extracting useful products from customer databases. These systems assist customers get products they would like to buy from a business. It helps customers by making them to get products they like. On the other hand, they help the business by generating more sales. It is a crucial tool in e-commerce on the Web. It is being used by the massive amount of customer data in private corporate databases, and will be used even more by the growing amount of customer data available on the public Web. New technologies are required that can significantly improve the scalability of recommender systems. SVD++ based method generates a very fast online performance, needed only a very easy arithmetic operations for each recommendation. But SVD++ matrix factorization process is very expensive in terms of time and memory. SVD++ based approach could be applied [29] for recommendation in different ways; by reducing rating matrix in low dimensions or applying SVD++ to find out useful items that would help to improve recommendation. Study shows that social popularity based SVD++ Recommender System may be such a technology in most of the cases. An extension to SVD++ has been proposed that based on latent factor models; permit improved accuracy by integrating popularity factor into the previous model. Other aspects of the data that may be incorporated to improve prediction quality [30] are content information like attributes of users or products, or data related with the ratings, which may help to explain move in user preferences.

Future study is to identify with exactly why SVD works fine for some recommender applications, and not as much fine for others. Also, SVD can be implemented in different ways to solve recommender systems problems. SVD can be used to create low dimensional visualizations of user-item rating matrix or SVD can help to identify important items or products that will help to recommend right items or products. The use of the implicit sentiment analysis within the CF in social web will be main concern in future

## 7. REFERENCES

- [1] B G. Adomavicius and A. Tuzhilin, "Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", IEEE Transactions on Knowledge and Data Engineering 17 (2005), 634–749.
- [2] Deurwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science. 41(6).
- [3] R. Harshman, "Indexing by Latent Semantic Analysis", Journal of the Society for Information Science 41 (1990), 391–407
- [4] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. (2000). Application of Dimensionality Reduction in Recommender System—A Case Study. In ACM WebKDD'00 (Web-mining for Ecommerce Workshop).
- [5] Sarwar, B.M., Konstan, J.A., Borchers, A., and Riedl, J. 1999. "Applying Knowledge from KDD to Recommender Systems." Technical Report TR99-013, Dept. of Computer Science, University of Minnesota.
- [6] R. Bell and Y. Koren, "Lessons from the Netflix Prize Challenge", SIGKDD Explorations 9 (2007), 75–79.
- [7] Yehuda Koren, 2008, Factorization meets the Neighborhood: a Multifaceted Collaborative Filtering Model In Proc. Of ACM KDD'08, August 24–27, Las Vegas, Nevada, USA
- [8] Bennet and S. Lanning, "The Netflix Prize", KDD Cup and Workshop, 2007. www.netflixprize.com.
- [9] J. Canny, "Collaborative Filtering with Privacy via Factor Analysis", Proc. 25th ACM SIGIR Conf.on Research and Development in Information Retrieval (SIGIR'02), pp. 238–245, 2002
- [10] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation", Journal of Machine Learning Research 3 (2003), 993–1022
- [11] Scott, B. A., & Judge, T. A. (2009). The popularity contest at work: Who wins, why, and what do they receive? Journal of Applied Psychology, 94(1), 20-33.
- [12] Anderson, L. and Holt, C. (1997). Information cascades in the laboratory The American Economic Review, 87, 847-863.
- [13] Lansu, T. M., & Cillessen, A. N. (2012). Peer status in emerging adulthood: Associations of popularity and preference with social roles and behavior. Journal of Adolescent Research, 27(1), 132-150.
- [14] Shardanand, U., and Maes, P. (1995). Social Information Filtering: Algorithms for Automating 'Word of Mouth'. In Proc. of CHI '95.
- [15] Adamic, L. (2002). Zipf, power-laws, and pareto-a ranking tutorial. Glottometrics, 3, 143-150.
- [16] S. Funk, "Netflix Update: Try This At Home", <http://sifter.org/~simon/journal/20061211.html>, 2006.
- [17] D. Goldberg, D. Nichols, B. M. Oki and D. Terry, "Using Collaborative Filtering to Weave an Information Berry, M. W., Dumais, S. T., and O'Brian, G. W. (1995). Using Linear Algebra for Intelligent Information Retrieval. SIAM Review, 37(4).
- [18] Simon Dooms, Toon De Pessemer, Dieter Verslype, Jelle Nelis, Jonas De Meulenaere, Wendy Van den Broeck, Luc Martens, and Chris Develder. Omus: an optimized multimedia service for the home environment. Multimedia Tools and Applications , 2013
- [19] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pages 230{237. ACM, 1999.
- [20] James Bennett and Stan Lanning. The Netflix prize. In Proceedings of KDD cup and workshop , volume 2007, page 35, 2007
- [21] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In Recommender systems handbook, pages 257, Springer, 2011.

- [22] Gupta, D., and Goldberg, K. (1999). Jester 2.0: A Linear Time Collaborative Filtering Algorithm Applied to Jokes. In *Proc. of the ACM SIGIR '99*.
- [23] Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An Algorithmic Framework for Performing Collaborative Filtering. In *Proc. Of ACM SIGIR'99*.
- [24] Hill, W., Stead, L., Rosenstein, M., and Furnas, G. (1995). Recommending and Evaluating Choices in a Virtual Community of Use. In *Proc. of CHI'95*.
- [25] Simon D., Toon De P. and Luc M. MovieTweatings: a Movie Rating Dataset Collected From Twitter In *Proc. of Crowdrec 2013*.
- [26] Simon D., Toon De P. and Luc M. MovieTweatings: a Movie Rating Dataset Collected From Twitter In *Proc. of Crowdrec 2013*
- [27] Hill, W., Stead, L., Rosenstein, M., and Furnas, G. (1995). Recommending and Evaluating Choices in a Virtual Community of Use. In *Proc. of CHI'95*.
- [28] Sharda nand, U., and Maes, P. (1995). Social Information Filtering: Algorithms for Automating' Word of Mouth'. In *Proc. of CHI '95*.
- [29] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. (2000). Analysis of Recommendation Algorithms for E-Commerce. In *Proc. of the ACM EC'00 Conference*. Minneapolis, MN, pp. 158-167.
- [30] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). GroupLens: A Open Architecture for Collaborative Filtering of Netnews. In *Proc. of CSCW '94*, Chapel Hill, NC.