Survey of Reccomendation System: Collaborative Filtering

Rajashree S. Sokasane
Department of Electronics and Computer Engineering
Chonnam National University
sokasaners@gmail.com

Sungmin Hwang
Department of Electronics and Computer Engineering
Chonnam National University
hsmvirus@gmail.com

Lingling Zhang
Department of Library and Information Science
Chonnam National University
s06303018@gmail.com

***Abstract*—This paper presents an overview of the field of collaborative filtering which is one of the most successful approaches for building recommender systems. Collaborative filtering (CF) uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users. CF is based on analyzing past interactions between users and items, and hence can be readily applied in a variety of domains, without requiring external information about the traits of the recommended products.In this paper, we first introduce CF tasks and their main challenges, such as data sparsity, scalability, synonymy, etc., and their possible solutions. We then present three main categories of CF techniques: memory-based, model-based, and hybrid CF algorithms.**

I. Introduction

Recommender Systems (RS) are software tools and techniques providing suggestions for items to be of use to a user [1]. The suggestions relate to various decision-making processes, such as what items to buy, what music to listen to, or what online news to read. Recommender systems assist and augment this natural social process to help people sift through available books, articles, webpages, movies, music, restaurants, jokes, grocery products, and so forth to find the most interesting and valuable information for them [2]. Recommender Systems predict a users preference on his unrated items based on the existing rating data and knowledge. Recommender systems play an important role in such highly rated Internet sites as Amazon.com, YouTube, Netflix, Yahoo, Tripadvisor, Last.fm, and IMDb. Moreover many media companies are now developing and deploying RSs as part of the services they provide to their subscribers. For example Netflix, the online movie rental service, awarded a million dollar prize to the team that first succeeded in improving substantially the performance of its recommender system [3].

Recommendation algorithms can be divided into multiple categories like content-based approaches, collaborative ltering, hybrid, etc. First, Content-based Recommender System: The system learns to recommend items that are similar to the ones that the user liked in the past. The similarity of items is calculated based on the features associated with the compared items. For example, if a user has positively rated a movie that belongs to the comedy genre, then the system can learn to recommend other movies from this genre [2]. Second, as one of the most representative categories, collaborative ltering is prevalent for its high performance and simple requirements. Collaborative Filtering (CF) Recommender System: CF relies only on past user behavior, e.g., their previous transactions or product ratings. It analyzes relationships between users and interdependencies among products, in order to identify new user-item associations.

Collaborative filtering: The simplest and original implementation of this approach recommends to the active user the items that other users with similar tastes liked in the past. The similarity in taste of two users is calculated based on the similarity in the rating history of the users. This is the reason why [2] refers to collaborative filtering as people-to-people correlation. Collaborative filtering is considered to be the most popular and widely implemented technique in Recommendation System. The fundamental assumption of CF is that if users X and Y rate n items similarly, or have similar behaviors (e.g., buying, watching, listening), and hence will rate or act on other items similarly [4].

CF techniques use a database of preferences for items by users to predict additional topics or products a new user might like. In a typical CF scenario, there is a list of m users u1, u2, . . . , um and a list of n items i1, i2, . . . , in, and each user, ui, has a list of items, Iui, which the user has rated, or about which their preferences have been inferred through their behaviors. The ratings can either be explicit indications, and so forth, on a 15 scale, or implicit indications, such as purchases or click-throughs [5]. For example, we can convert the list of people and the movies they like or dislike (Table 1(a)) to a user-item ratings matrix (Table 1(b)), in which Tony is the active user that we want to make recommendations for. There are missing values in the matrix where users did not give their preferences for certain items.

There are many challenges for collaborative filtering tasks (Section II). CF algorithms are required to have the ability to deal with highly sparse data, to scale with the increasing numbers of users and items, to make satisfactory recommendations in a short time period, and to deal with other problems like synonymy (the tendency of the same or similar items to have different names), shilling attacks, data noise, and privacy protection problems.

However, there are several limitations for the memorybased CF techniques, such as the fact that the similarity values are based on common items and therefore are unreliable when data are sparse and the common items are therefore few. To achieve better prediction performance and overcome shortcomings of memory-based CF algorithms, model-based CF approaches have been investigated. Model-based CF techniques (Section IV) use the pure rating data to estimate or learn a model

TABLE I. AN EXAMPLE OF A USER-ITEM MATRIX.

Alice:	(like) Shrek, SnowWhite, (dislike) Superman
Bob:	(like) SnowWhite, Superman, (dislike) spiderman
Chris:	(like) spiderman, (dislike) Snow white
Tony:	(like) Shrek, (dislike) Spiderman

(a)

	Shrek	SnowWhite	Spider-man	Super-man
Alice	Like			Dislike
Bob		Like	Dislike	Like
Chris		Dislike	Like	
Tony	Like		Dislike	?

(b)

to make predictions [9]. The model can be a data mining or machine learning algorithm. Well-known model-based CF techniques include Bayesian belief nets (BNs) CF models [6], clustering CF models [7], [8], and latent semantic CF models [9]. An MDP (Markov decision process)-based CF system [14] produces a much higher profit than a system that has not deployed the recommender.

Hybrid CF techniques, such as the Personality Diagnosis (PD) [10], combine memory-based and model-based CF techniques, hoping to avoid the limitations of either approach and thereby improve recommendation performance (Section V).

II. CHARACTERISTICS AND CHALLENGES OF COLLABORATIVE FILTERING

Recommendation system is especially important to huge internet market such as Amazon. Attraction from proper recommendation often increases the company's sales output. This makes the prediction rate performance of recommender system- a central goal of the system. Collaborative Filtering has some distinguishable characteristics that are relevant to the performance of the system, and there exist efforts to overcome the characteristics.

A. Data Sparsity

In real internet markets, the number of products and product sets are very large. Large number in products means user-item matrix, which is responsible for the performance, may be sparse. From this characteristic, how to process the sparse data and match is the challenge in Collaborative Filtering.

There are several kinds of sparse problem. The cold start problem occurs when a new user or item has just entered the system, it is difficult to find similar ones because there is not enough information [11], [12] . When new item comes in, it does not have user rating information so it is hard to find the relation. Also, in new user cases, users are unlikely given good recommendations due to the lack of their rating or purchase history.

In the cases that there are too small users ratings compared to the large number of items in the system, the reduced coverage problem occurs. This problem makes it hard to generate recommendation for users.

Neighbor transitivity, which is a problem with sparse databases, in which users with similar tastes may not be identified as such if they have not both rated any of them same products, is also the problem related to data sparsity.

To reduce these problems, many studies have been made. Dimensionality reduction techniques, such as Singular Value Decomposition(SVD) [13] remove unrepresentative or insignificant users or items to reduce the dimensionalities of the user-item matrix directly. These studies reduced the sparsity but caused some issues. When certain users or items are discarded, useful information was discarded too. And that lead to decrease in quality of recommendation. In model based CF, and hybrid CF algorithms, some techniques have been suggested to fix the cold start problems. Content-boosted CF algorithm [14] as hybrid CF, TAN-ELR [6], [15] in Model based CF algorithms are the examples.

B. Scalability

Increase in size brought new problem, the scalability. As data sets grow larger and larger, it became harder to use traditional CF algorithms. In these years, they recommender system should face millions of distinct items with tens of millions of customers. This will obviously take up the limited resources. The dimensionality reduction techniques, that were discussed earlier, can deal with these scalability problems, but they require some procedure such as, matrix factorization steps, which has expensive cost. An incremental SVD algorithm [16] has been suggested incremental system without re-computing to reduce the cost. Memor-based CF algorithms, such as the item-based Pearson correlation CF algorithm reduces the cost by calculating similarities only between the pair of co-rated items by a user. As discussed earlier, when using this technique, the decrease in prediction performance follows.

C. Synonymy

When same kind of products has different names, it is hard to most recommender systems to discover this relation and takes these products differently. Children movie and children film can be the example of this scenario. Memory-based CF systems are vulnerable to this problem. Some attempt to solve this problem led to such as intellectual or automatic term expansion, but they had drawbacks caused by confusion in intended meaning of terms. Latent Semantic Indexing(LSI) in SVD technique are capable of dealing with this synonymy problems but still shows problem in some conditions. No algorithm gives whole solution so far.

D. Gray Sheep

In some cases, there are users who dont follow the opinions of others, and that makes them to not get the benefit from CF. It just happens with no solution, so it is considered as acceptable failure. There was an approach to reduce this problem by having per-user approach [17].

E. Shilling Attacks

When the item producer wants to make his product look better, he will make many recommendation on his own product, and negative recommendations to the competing product. A study on this attack lead to some researches, and in that research, they found that Item-based CF algorithm attack was

much less affected by the attacks than the user-based CF algorithm and how to detect this kind of attacks has been found too. After the efforts, several ways to solve this problem has been suggested [18], [19].

F. Other Challenges

In some cases, observing each users habit can be the violation of personal privacy. There are some efforts to keep the privacy in recommender systems [5].

Increased noise can cause the performance decrease. As the users get more diversity, the more noise came out. This noise dealing could get help from some techniques, such as ensembles of maximum margin matrix factorizations [20] and instance selection techniques [21].

The last one should be mentioned is explainability. Explaining why the recommendation has been made will help readers.

III. Memory-Based Collaborative Filtering Techniques

Memory-based methods simply memorize the rating matrix and issue recommendations based on the relationship between the queried user and item and the rest of the rating matrix. Memory-based CF algorithms use the entire or a sample of the user-item database to generate a prediction. Every user is part of a group of people with similar interests. By identifying the so-called neighbors of a new user (or active user), a prediction of preferences on new items for him or her can be produced.

The most popular memory-based CF methods are neighborhood-based methods, which predict ratings by referring to users whose ratings are similar to the queried user, or to items that are similar to the queried item. The neighborhood-based CF algorithm, a prevalent memory-based CF algorithm, uses the following steps: calculate the similarity or weight, then aggregate the neighbors to get the top- N most frequent items as the recommendation.

A. Similarity Computation

Similarity computation between items or users is a critical step in memory-based collaborative filtering algorithms. For item-based CF algorithms, the first work is to compute the similarity between item i and item j and for a user-based CF algorithm, we first calculate the similarity wu,v, between the users u and v who have both rated the same items.

1) Correlation-Based Similarity: In this case, similarity $W_{u,v}$ between two users u and v, or similarity $W_{i,j}$ between two items i and j, is measured by computing the Pearson correlation or other correlation-based similarities.The Pearson correlation, which is widely used in research, is a popular algorithm for collaborative filtering. Similarity between two users (and their attributes, such as articles read from a collection of blogs) can be accurately calculated with the Pearson correlation. This algorithm measures the linear dependence between two variables (or users) as a function of their attributes. For the user-based algorithm, the Pearson correlation between users u and v is

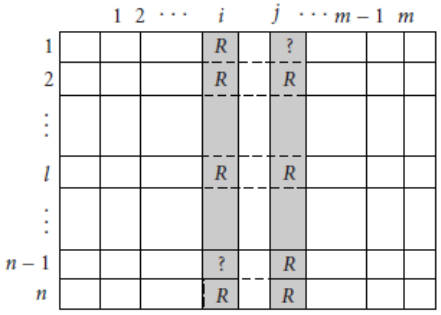

Fig. 1. item-based similarity ($w_{i,j}$) calculation based on the corated items i and j from users 2, l and n.

TABLE II. Simple example of ratings matrix.

	I_1	I_2	I_3	I_4
U_1	4	?	5	5
U_2	4	2	1	
U_3	3		2	4
U_4	4	4		
U_5	2	1	3	5

$$W_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in I}(r_{v,i} - \bar{r}_v)^2}} \quad (1)$$

where the i ∈ I summations are over the items that both the users u and v have rated and is the average rating of the co-rated items of the u-th user. In an example in Table II, we have $W_{1,5}$ = 0.756

For the item-based algorithm, denote the set of users u ∈ U who rated both items i and j, then the Pearson Correlation will be

$$W_{i,j} = \frac{\sum_{u \in U}(r_{u,i} - \bar{r}_i)(r_{u,i} - \bar{r}_j)}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r}_i)^2}\sqrt{\sum_{u \in U}(r_{v,j} - \bar{r}_j)^2}} \quad (2)$$

where$r_{u,i}$ is the rating of user u on item i, $\bar{r}_i$ is the average rating of the ith item by those users, see Figure 1

Usually the number of users in the computation of similarity is regarded as the neighborhood size of the active user, and similarity based CF is deemed as neighborhoodbased CF.

2) Vector Cosine-Based Similarity: The similarity between two documents can be measured by treating each document as a vector of word frequencies and computing the cosine of the angle formed by the frequency vectors [61]. This formalism can be adopted in collaborative filtering, which uses users or items instead of documents and ratings instead of word frequencies.

Formally, if R is the m n user-item matrix, then the similarity between two items, i and j, is defined as the cosine of the n dimensional vectors corresponding to the ith and jth column of matrix R.

Vector cosine similarity between items i and j is given by

$$W_{i,j} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \bullet \vec{j}}{\| \vec{i} \| * \| \vec{j} \|} \quad (3)$$

where • denotes the dot-product of the two vectors. To get the desired similarity computation, for n items, an n n similarity matrix is computed [22]. For example, if the vector $\overrightarrow{A}$ = {x1, y1}, vector $\overrightarrow{B}$ = {x2, y2}, the vector cosine similarity between $\overrightarrow{A}$ and $\overrightarrow{B}$ is

$$W_{A,B} = \cos(\overrightarrow{A}, \overrightarrow{B}) = \frac{\overrightarrow{A} \bullet \overrightarrow{B}}{\| \overrightarrow{A} \| * \| \overrightarrow{B} \|} = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2}\sqrt{x_2^2 + y_2^2}} \quad (4)$$

B. Prediction and Recommendation Computation

To obtain predictions or recommendations is the most important step in a collaborative filtering system. In the neighborhoodbased CF algorithm, a subset of nearest neighbors of the active user are chosen based on their similarity with him or her, and a weighted aggregate of their ratings is used to generate predictions for the active user.

1) Weighted Sum of Others Ratings: To make a prediction for the active user, a, on a certain item, i, we can take a weighted average of all the ratings on that item by using the following formula

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in U}(r_{u,i} - \bar{r}_i) \cdot W_{a,u}}{\sum_{u \in U} |W_{a,u}|} \quad (5)$$

where r_a and r_u are the average ratings for the user a and user u on all other rated items, and $w_{a,u}$ is the weight between the user a and user u. The summations are over all the users u U who have rated the item i. For the simple example in Table II, using the user-based CF algorithm, to predict the rating for U1 on I2, we have

$$\begin{aligned} P_{a,i} &= \bar{r}_1 + \frac{\sum_{u \in U}(r_{u,2} - \bar{r}_i) \cdot W_{1,u}}{\sum_u |W_{1,u}|} \\ &= \bar{r}_1 + \frac{(r_{2,2} - \bar{r}_2)W_{1,2} + (r_{4,2} - \bar{r}_4)W_{1,4} + (r_{5,2} - \bar{r}_5)W_{1,5}}{|W_{1,2}| + |W_{1,4}| + |W_{1,5}|} \\ &= 4.67 + \frac{(2 - 2.5)(-1) + (4 - 4)0 + (1 - 3.33)0.756}{1 + 0 + 0.756} \\ &= 3.95 \end{aligned} \quad (6)$$

Note the above prediction is based on the neighborhood of the active users.

2) Simple Weighted Average: For item-based prediction, we can use the simple weighted average to predict the rating, $P_{u,i}$, for user u on item i

$$P_{u,i} = \frac{\sum_{n \in N}(r_{u,n} W_{i,u}}{\sum_{n \in N} |W_{i,n}|} \quad (7)$$

where the summations are over all other rated items n N for user u,$w_{i,n}$ is the weight between items i and n, $r_{u,n}$ is the rating for user u on item n.

3) Top-N Recommendations: Top-N recommendation is to recommend a set of N top-ranked items that will be of interest to a certain user. For example, if you are a returning customer, when you log into your http://amazon.com/ account, you may be recommended a list of books (or other products) that may be of your interest. Top-N recommendation techniques analyze the user-item matrix to discover relations between different users or items and use them to compute the recommendations. Some models, such as association rule mining based models, can be used to make top-N recommendations

IV. Model-Based Collaborative Filtering Techniques

The design and development of models can allow the system to learn to recognize complex patterns based on the training data, and then make intelligent predictions for the collaborative filtering tasks for test data or real-world data, based on the learned models. Model-based CF algorithms, such as Bayesian models, clustering models, and dependency networks, have been investigated to solve the shortcomings of memory-based CF algorithms [7]. Usually, classification algorithms can be used as CF models if the user ratings are categorical, and regression models and SVD methods and be used for numerical ratings.

A. Bayesian Belief Net CF Algorithms

A Bayesian belief net (BN) is a directed, acyclic graph (DAG) with a triplet (N,A, Θ) , where each node n ∈ N represents a random variable, each directed arc a ∈ A between nodes is a probabilistic association between variables, and Θ is a conditional probability table quantifying how much a node depends on its parents. Bayesian belief nets (BNs) are often used for classification tasks.

1) Simple Bayesian CF Algorithm: The simple Bayesian CF algorithm uses a nave Bayes (NB) strategy to make predictions for CF tasks. Assuming the features are independent given the class, the probability of a certain class given all of the features can be computed, and then the class with the highest probability will be classified as the predicted class [23]. For incomplete data, the probability calculation and classification production are computed over observed data (the subscript o in the following equation indicates observed values):

$$class = \arg \max_{j \in classSet} p(class_j) \Pi_o P(X_o = x_o | class_j) \quad (8)$$

The Laplace Estimator is used to smooth the probability calculation and avoid a conditional probability of 0:

$$P(X_i = x_i | Y = y) = \frac{\#(X_i = x_i, Y = y) + 1}{\#(Y = y) + |X_i|} \quad (9)$$

where —Xi— is the size of the class set Xi. For an example of binary class, P(Xi = 0 — Y = 1) = 0/2will be (0+1)/(2+2) = 1/4, P(Xi = 1 — Y = 1) = 2/2 will be (2 + 1)/(2 + 2) = 3/4 using the Laplace Estimator. Using the same example in Table II, the class set is 1, 2, . . . , 5, to produce the rating

for U1 on I2 using the simple Bayesian CF algorithm and the Laplace Estimator, we have

$$\begin{aligned} class &= \arg \max_{c_j \in \{1,2,3,4,5\}} p(c_j | U_2 = 2, U_4 = 4, U_5 = 1) \\ &= \arg \max_{c_j \in \{1,2,3,4,5\}} p(c_j) P(U_2 = 2 | c_j) P(U_4 = 4 | c_j) \\ &\times P(U_5 = 1 | c_j) \\ &= \arg \max_{c_j \in \{1,2,3,4,5\}} \{0, 0, 0, 0.0031, 0.0019\} \end{aligned} \tag{10}$$

in which p(5) P(U_2 = 2 | 5) P(U_4 = 4 | 5) P(U_5 = 1 | 5) = (2/5) $*$ (1/7) $*$ (1/7) $*$ (1/7) = 0.0019

B. Clustering CF Algorithms

A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. The measurement of the similarity between objects is determined using metrics such as Minkowski distance and Pearson correlation.

In most situations, clustering is an intermediate step and the resulting clusters are used for further analysis or processing to conduct classification or other tasks. Clustering CF models can be applied in different ways. Using the k-means method with k = 2, the RecTree method, proposed by Chee et al.[8], recursively splits the originally large rating data into two sub-clusters as it constructs the RecTree from the root to its leaves. The resulting RecTree resembles an unbalanced binary tree, of which leaf nodes have a similarity matrix and internal nodes maintain rating centroids of their subtrees. The prediction is made within the leaf node that the active user belongs to. RecTree scales by O(n log2(n)) for off-line recommendation and O(b) for on-line recommendation, where n is the dataset size and b is the partition size, a constant, and it has an improved accuracy over the Pearson correlation-based CF when selecting an appropriate size of advisors (cluster of users).

A flexible mixture model (FMM) extends existing clustering algorithms for CF by clustering both users and items at the same time, allowing each user and item to be in multiple clusters and modeling the clusters of users and items separately [24]. Experimental results show that the FMM algorithm has better accuracy than the Pearson correlationbased CF algorithm and aspect model.

Clustering models have better scalability than typical collaborative filtering methods because they make predictions within much smaller clusters rather than the entire customer base [8], [13]. The complex and expensive clustering computation is run offline. However, its recommendation quality is generally low.

C. Latent Semantic CF Models

A Latent semantic CF technique relies on a statistical modeling technique that introduces latent class variables in a mixture model setting to discover user communities and prototypical interest profiles. Conceptionally, it decomposes user preferences using overlapping user communities. The main advantages of this technique over standard memory-based methods are its higher accuracy and scalability [9].

V. Hybrid Collaborative Filtering Techniques

The two major classes of CF approaches, memory-based and model-based CF approaches, can be combined to form hybrid CF approaches. The recommendation performances of these algorithms are generally better than some pure memory-based CF algorithms and model-based CF algorithms [12].

A. Probabilistic memory-based collaborative filtering

Probabilistic memory-based collaborative filtering (PMCF) combines memory-based and model-based techniques [12]. They use a mixture model built on the basis of a set of stored user profiles and use the posterior distribution of user ratings to make predictions. To address the new user problem, an active learning extension to the PMCF system can be used to actively query a user for additional information when insufficient information is available. To reduce the computation time, PMCF selects a small subset called profile space from the entire database of user ratings and gives predictions from the small profile space instead of the whole database. PMCF has better accuracy than the Pearson correlation-based CF and the model-based CF using nave Bayes.

B. Personality diagnosis

Personality diagnosis (PD) is a representative hybrid CF approach that combinesmemory-based and model-based CF algorithms and retains some advantages of both algorithms. In PD, the active user is assumingly generated by choosing one of the other users uniformly at random and adding Gaussian noise to his or her ratings. Given the active users known ratings, we can calculate the probability that he or she is the same personality type as other users, and the probability he or she will like the new items. PD can also be regarded as a clustering method with exactly one user per cluster. Working on EachMovie and CiteSeer, PD makes better predictions than Pearson correlation-based and vector similarity-based CF algorithms and the two modelbased algorithms, Bayesian clustering and Bayesian network, investigated by Breese et al. [7]. As an ensemble classifier is able to give more accurate prediction than a member classifier, a hybrid CF system that combines different CF algorithms using an ensemble scheme will also be helpful to improve predictive performance of CF tasks [25].

VI. Conclusion

Collaborative filtering (CF) is one of the most successful recommender techniques. Mainly, there are memory-based CF techniques such as the neighborhood-based CF algorithm; model-based CF techniques such as Bayesian belief nets CF algorithms, clustering CF algorithms; and hybrid CF techniques such as the Personality diagnosis.

As a representative memory-based CF technique, neighborhood-based CF computes similarity between users or items, and then use the weighted sum of ratings or simple weighted average to make predictions based on the similarity values. Pearson correlation and vector cosine similarity are commonly used similarity calculations, which are usually conducted between co-rated items by a certain user or both users that have co-rated a certain item. To make top-N recommendations, neighborhood-based methods

can be used according to the similarity values. Memory-based CF algorithms are easy to implement and have good performances for dense datasets. Shortcomings of memory-based CF algorithms include their dependence on user ratings, decreased performance when data are sparse, new users and items problems, and limited scalability for large datasets, and so forth [6].

Model-based CF techniques need to train algorithmic models, such as Bayesian belief nets, clustering techniques to make predictions for CF tasks. Advanced Bayesian belief nets CF algorithms with the ability to deal with missing data are found to have better performance than simple Bayesian CF models and Pearson correlation-based algorithms [6]. Clustering CF algorithms make recommendations within small clusters rather than the whole dataset, and achieve better scalability. There are downsides of model-based CF techniques, for example, they may not be practical when the data are extremely sparse, the solutions using dimensionality reduction or transformation of multiclass data into binary ones may decrease their recommendation performance, the model-building expense may be high, and there is a tradeoff between prediction performance and scalability for many algorithms.

Most hybrid CF techniques combine CF methods with content-based techniques or other recommender systems to alleviate shortcomings of either system and to improve prediction and recommendation performance. Besides improved performance, hybrid CF techniques rely on external content information that is usually not available, and they generally have increased complexity.

ACKNOWLEDGMENT

The author would like to thank Prof. Kyungbaek Kim for adding Latex knowledge into our knowledge base.

REFERENCES

[1] T. Mahmood and F. Ricci, "Improving recommender systems with adaptive conversational strategies," in *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*, ser. HT '09. New York, NY, USA: ACM, 2009, pp. 73–82.

[2] P. B. Kantor, *Recommender systems handbook*. New York; London: Springer, 2009.

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

[4] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Inf. Retr.*, vol. 4, no. 2, pp. 133–151, Jul. 2001.

[5] B. N. Miller, J. A. Konstan, and J. Riedl, "Pocketlens: Toward a personal recommender system," *ACM Trans. Inf. Syst.*, vol. 22, no. 3, pp. 437–476, Jul. 2004.

[6] X. Su and T. M. Khoshgoftaar, "Collaborative filtering for multi-class data using belief nets algorithms." in *ICTAI*. IEEE Computer Society, 2006, pp. 497–504.

[7] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 43–52. [Online]. Available: http://dl.acm.org/citation.cfm?id=2074094.2074100

[8] S. H. S. Chee, J. Han, and K. Wang, "Rectree: An efficient collaborative filtering method," in *Proceedings of the Third International Conference on Data Warehousing and Knowledge Discovery*, ser. DaWaK '01. London, UK, UK: Springer-Verlag, 2001, pp. 141–151.

[9] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 89–115, Jan. 2004.

[10] D. Y. Pavlov and D. M. Pennock, "A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains," in *In Proceedings of Neural Information Processing Systems*. MIT Press, 2002, pp. 1441–1448.

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

[12] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel, "Probabilistic memory-based collaborative filtering." *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 1, pp. 56–69, 2004.

[13] D. Billsus and M. Pazzani, "Learning collaborative information filters," in *15th International Conference on Machine Learning*, 1998, pp. 46–54.

[14] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," pp. 187–192, 2002.

[15] R. Greiner, X. Su, B. Shen, and W. Zhou, "Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers." *Machine Learning*, vol. 59, no. 3, pp. 297–322, 2005.

[16] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Incremental singular value decomposition algorithms for highly scalable recommender systems," in *Fifth International Conference on Computer and Information Science*, 2002, pp. 27–28.

[17] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin, "Combining content-based and collaborative filters in an online newspaper," in *ACM SIGIR Workshop on Recommender Systems*, 19August 1999.

[18] M. O'Mahony, N. Hurley, N. Kushmerick, and G. Silvestre, "Collaborative recommendation: A robustness analysis," *ACM Trans. Inter. Tech.*, vol. 4, no. 4, pp. 344–377, November 2004.

[19] R. M. Bell and Y. Koren, "Improved neighborhood-based collaborative filtering," in *1st KDDCup'07*, San Jose, California, 2007.

[20] D. DeCoste, "Collaborative prediction using ensembles of maximum margin matrix factorizations," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 249–256.

[21] K. Yu, X. Xu, J. Tao, M. Ester, and H.-P. Kriegel, "Instance selection techniques for memory-based collaborative filtering," in *Proceedings of Second SIAM International Conference on Data Mining (SDM?02)*, 2002.

[22] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proceedings of the 2Nd ACM Conference on Electronic Commerce*, ser. EC '00. New York, NY, USA: ACM, 2000, pp. 158–167.

[23] K. Miyahara and M. J. Pazzani, "Collaborative filtering with the simple bayesian classifier," in *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence*, ser. PRICAI'00. Berlin, Heidelberg: Springer-Verlag, 2000, pp. 679–689.

[24] L. Si and R. Jin, "Flexible mixture model for collaborative filtering," in *In Proc. of ICML*. AAAI Press, 2003, pp. 704–711.

[25] X. Su, R. Greiner, T. M. Khoshgoftaar, and X. Zhu, "Hybrid collaborative filtering algorithms using a mixture of experts," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, ser. WI '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 645–649.