Collaborative Filtering with Label Consistent Restricted Boltzmann Machine

Sagar Verma
IIIT Delhi
sagar15056@iiitd.ac.in

Prince Patel
IIIT Delhi
prince15046@iiitd.ac.in

Angshul Majumdar
IIIT Delhi
angshul@iiitd.ac.in

***Abstract*—The possibility of employing restricted Boltzmann machine (RBM) for collaborative filtering has been known for about a decade. However, there has been hardly any work on this topic since 2007. This work revisits the application of RBM in recommender systems. RBM based collaborative filtering only used the rating information; this is an unsupervised architecture. This work adds supervision by exploiting user demographic information and item metadata. A network is learned from the representation layer to the labels (metadata). The proposed label consistent RBM formulation improves significantly on the existing RBM based approach and yield results at par with the state-of-the-art latent factor based models.**

***Index Terms*—Collaborative Filtering, Recommender Systems, Restricted Boltzmann Machine, Supervised Learning**

I. Introduction

With the worldwide boom of E-Commerce (business-to-client) research in recommender systems has become one of the top priorities both for academia and the industry [1], [2]. Recommender systems are beneficial for both, the business, and the client. Unlike a physical marketplace, online portals have virtually an inexhaustible collection of items. It is a huge task for the user to sift through all the options and buy/rent the one he/she is satisfied with. Thus, the recommender systems assist the buyer/client with tailored suggestions. For the E-Commerce portal, without proper recommendations the user does not purchase items; hence the portal looses on business opportunity and thereby loses prospective revenue. The initial days of recommender systems saw the application of content based filtering to recommender systems [3]. Content based filtering was a well-established approach in information retrieval this was in the early 2000s. However, it never became popular; several reasons are briefly mentioned in [4]. Mainly owing to the requirement of user intervention in the definition of content; i.e. one needed to find out the exact attributes for matches between user and item. For example, for books the factors might be the author, genre, publisher; for music, they might be the singer, genre; for movie, they can be anything ranging from the actors to the director to the production house to the genre. One did not know if the expert defined list had already captured all the possible variabilities in the chosen attributes. If the list of attributes is too small, important factors would be missing; if the list is too large one might end up capturing noise in the data. During the same time (the early 2000s) more abstract yet powerful techniques based on latent factor modeling and representation learning started gaining momentum. Instead of explicitly designing the attributes, latent factor model makes an abstract assumption. It assumes that the users choice of items is guided by several latent factors. Instead of designing these factors (as in content based filtering), they were learned from the data. Eventually, matrix factorization [5], [6] became the most popular technique for latent factor model especially after the announcement of the famed Netflix competition [7]. In the late 90s and early 2000s neighborhood based approaches gained popularity in collaborative filtering. They were interpolation based techniques. In the user-based approach [8], similar users were selected to constitute the neighbourhood, and the ratings of these users were used to impute the missing values. The same could be done from an item perspective [9]. These approaches were simple and easy to interpret. But they were heuristic, the interpolation weights were defined rather arbitrarily. The neighborhood based method yielded significantly lower accuracy (at least on the benchmark databases) than the more powerful albeit abstract latent factor models. Even though the matrix factorization technique was the popular choice for latent factor based collaborative filtering, a seminal work [10] showed the possibility of using another representation learning/latent factor approach for collaborative filtering; it was the restricted Boltzmann machine (RBM). There is hardly any work on RBM based collaborative filtering since the publication of [10]. The basic RBM formulation used in [10] is an unsupervised one; it was only based on the users rating on the items. However, in a real system, users and items metadata is always available. User demographic information such as age, occupation, gender etc. is collected as a part of the sign-up process. The item information is also collected during its registration. Prior RBM based formulation could not make use of such auxiliary formation. In this work, we propose to improve upon [10] (in terms of rating prediction accuracy) by exploiting the user demographics and item metadata. In recent times (last few years), researchers have started exploring the possibility of using another powerful representation learning technique for collaborative filtering, stacked autoencoder. Both stacked autoencoders and deep belief network (built from layers of RBM) are used to train deep neural networks. Almost all studies in autoencoder based collaborative filtering are minor variations of each other. The basic autoencoder formulation is directly used in [11]–[13].

978-1-5386-2241-4/17/$31.00 © 2017 IEEE

In [14] baseline prediction is used along with the ratings in the autoencoder framework; the baseline values are simply appended with the available ratings so that the autoencoder learns to reconstruct both the ratings and the baseline values. A combination of marginalized denoising autoencoder and probabilistic matrix factorization is used in [15] for rating prediction. All representation learning approaches are inherently nonconvex. Therefore, the associated theoretical problems are ever present. There is an elegant convex solution to the matrix factorization approach for collaborative filtering; this is called matrix completion [16], [17]. It is a convex variant which directly solves for the missing ratings instead of going through the intermediate steps of determining the latent factors for the users and the items. However, this is unrelated to the representation learning/latent factor model based approaches we will not discuss it in detail. The rest of the paper will be organized into several sections. Background on collaborative filtering will be discussed in the next section. The proposed formulation will be detailed in section 3. The experimental results will be shown in section 4. Conclusions of this work and future directions will be discussed in section 5.

II. Background

A. Neighborhood Models

Even though neighborhood/memory based models are not the focus of this work, we discuss it nevertheless for the sake of completion. Collaborative filtering can be thought of as a matrix completion problem. We assume that the users are the rows and the items along the columns. Each user has rated a few items. Based on such parsimonious ratings, recommender systems need to predict the missing ratings. Once it has predicted the ratings, it recommends items to users with high valued ratings. Neighborhood based approach follows a simple interpolation based formulation. For an active user, it predicts the missing ratings. First, it finds users similar to the active user (neighborhood) by computing some kind of similarity (cosine, inverse distance etc.). Next, it interpolates the active users missing ratings as a linear combination of the ratings from the neighborhood. The linear interpolation weights are heuristically fixed; usually, they are the normalized similarity weights computed in the first step. Such a technique is called the user-based approach [8]. One can perform exactly the same steps from an items perspective, leading to the item based approach [9]. There are combined user and item based techniques [18] as well. Such neighborhood based models are simple to understand and implement. The results are easy to analyze. However, they are heuristic and do not yield very good results.

B. Matrix Factorization

The latent factor model assumes that the users choice in items is determined by a handful of factors. For example, in content based filtering, these factors are hand-picked; for books, they might be author, publisher, and genre and for TV shows they might be star cast and genre. However latent factor models do not try to select these factors, rather they learn the abstract hidden factors from the data. Say a users latent factor be represented as a vector u_i and the items latent factor be represented as v_j. The user will like the item and provide a high rating if the corresponding latent factors match. This is captured as an inner product between the two vectors. Therefore, the rating of the i^{th} user on the j^{th} item is modeled as:

$$r_{i,j} = u_i v_j, \forall i, j \tag{1}$$

For the entire rating matrix, i.e. for M users and N items, the latent factor model is expressed as:

$$R = UV \text{ where } U = [u+1|\dots|u_M]\ V^T = [v_1|\dots|v_N] \tag{2}$$

In collaborative filtering, the full rating matrix is not available. A partially sampled version (the items for which users have provided ratings) are only available. The objective is to predict the missing values so that recommendations can be made based on those. The rating acquisition can be mathematically expressed as:

$$Y = M \odot R = M \odot (UV) \tag{3}$$

Here M is the binary mask; it has one where the ratings are available and zeroes where they are missing. The symbol $\odot$ shows binary multiplication.

This (3) is a highly-under-determined problem. Usually, less than 5% of the matrix is filled. Therefore, correctly predicting the missing ratings is a challenging problem. However, the number of latent factors are usually small, for books it may be just two or three, for movies it can be as large as 40; but still, the number of factors are much smaller than the rows or columns of the matrix. Hence the rating matrix will be a low-rank matrix. Therefore, the number of parameters to be learned (elements in U and V) are usually much smaller than the dimensionality of the matrix. Intuitively speaking, this gives some hope for solving the under-determined system (3); as long as the pieces of information (ratings in this case) is larger than the number of parameters to be learned, we can hope to be able to estimate them. The user and item latent factor matrices are recovered by solving the standard matrix factorization problem

$$\min_{U,V} \left\| Y - M \odot (UV) \right\|_F^2 + \lambda(\|U\|_F^2 + \|V\|_F^2) \tag{4}$$

The Tikhonov type penalties prevent overfitting. There can be various ways to solve the factorization problem (4). It can be as simple as stochastic gradient descent and alternating least squares to as complex as probabilistic matrix factorization. Authors in [19] argued that even though users latent factors can be dense since it expected that human beings have a certain degree of affinity towards all factors; an item cannot possess all the latent factors simultaneously. Hence, it is likely that they would be sparse. Following this argument, a sparsity promoting l1-norm penalty on V has been proposed in the aforesaid study.

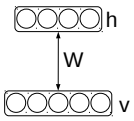

Fig. 1. Restricted Boltzman machine

$$\min_{U,V} \left\| Y - M \odot (UV) \right\|_F^2 + \lambda (\|U\|_F^2 + \|V\|_1) \tag{5}$$

The problems (4 and 5) are bi-linear, hence non-convex. There is no guarantee that they will converge to the optimum. Matrix completion is an elegant formulation that directly solves for the ratings and not the user and the item latent factors. It recovers the rating matrix directly by searching for a low-rank solution. However, minimizing the rank (number of singular values) is a NP-hard problem hence it's closest convex proxy the nuclear norm (sum of singular values) is used instead [20], [21]. The formulation for matrix completion is,

$$\min_{R} \|Y - M \odot R\|_F^2 + \lambda (\|R\|_N N) \tag{6}$$

This is a convex formulation which can be solved by semidefinite programming. Many faster algorithms also exist. As mentioned earlier, collaborative filtering is a highly under determined problem. So far, we only described basic techniques which required ratings only. In such a scenario, it is likely that using secondary information may improve the results. In practice metadata regarding the user is easily available; demographic information of the users is also available to the portal from the registration / sign-up process. Several studies [22], [23] have used this auxiliary information to boost the accuracy of a recommender system. Others have combined information from neighborhood based models with matrix factorization. In [24] the similarity between individuals was used as a graph regularization in the matrix factorization framework.

C. Restricted Boltzmann Machine

The restricted Boltzmann machine has been proposed in [10] to address the collaborative filtering problem. It is a two-layered (input and representation) undirected graphical model (as shown in Figure 1).

A simple RBM has a visible layer V and hidden layer h. Edges connecting these two layers are undirected and the whole network is a bipartite graph. The visible layer V is $K \times m$ matrix with $V_i^k = 1$, if a user rating for movie i is k. Hidden layer h is a $1 \times F$ vector. Joint distribution between hidden layer h and visible layer V takes the form:

$$p(V, h) \propto e^{-E(V,h)} \tag{7}$$

where $E(V, h) = -h^T WV - a^T V - b^T hW$ with parameters $\theta = (W, a, b,)$. Equation 8 captures predictive information about the input vector. $P(h|V)$ has a similar form.

$$p(V_i^k = 1|h) = \frac{exp(V_i^k + \sum_{j=1}^{F} h_j W_{ij}^k)}{\sum_{l=1}^{K} exp(V_i^l + \sum_{j=1}^{F} h_j W_{ij}^l)} \tag{8}$$

$$p(h_j = 1|V) = \sigma(b_j + \sum_{i=1}^{m} \sum_{k=1}^{K} V_i^k W_{ij}^k) \tag{9}$$

where $\sigma(x)$ is a logistic function, W_{ij}^K is weight parameter between hidden unit(feature) j and rating k of movie i, b_i^k is the bias of rating k for movie i, b_j is the bias of feature j.

The marginal distribution over the visible ratings V is:

$$p(V) = \sum \frac{exp(-E(V, h))}{\sum_{V'h'} exp(-E(V', h'))} \tag{10}$$

Here $E(V, h)$ is energy function given by Equation 7.

To update parameters gradient ascent is required in the log-likelihood, this can be obtained from equation 10.

$$\Delta W_{ij}^k = \mu \frac{\delta log p(V)}{\delta W_{ij}^k} = \mu(< V_i^k h_j >_{data} - < V_i^k h_j >_{model}) \tag{11}$$

where μ is the learning rate. The expectation $< V_i^k h_j >_{data}$ defines the frequency with which movie rating k and feature j are on together when the features are being driven by the observed user-rating data from the training set using Equation 10, and $< . >_{model}$ is an expectation with respect to the distribution defined by model. The expectation $< . >_{model}$ cannot be computed analytically in less than exponential time. Contrastive Divergence (CD) [24] is used to approximate computation of gradient of objective function.

$$\Delta W_{ij}^k = \mu(< V_i^k h_k >_{data} - < V_i^k h_k >_T) \tag{12}$$

The expectation $< . >_{data}$ represents a distribution of samples from running the Gibbs sampler, initialized at the data, for T full steps.

Given the observed ratings R, we can predict a rating for a new query movie m in time linear in the number of hidden units:

$$\begin{aligned} p(V_m^k = 1|R) &\propto \sum_{h_1 \ldots h_p} exp(-E(V_m^k, R, h)) \\ &\propto \Gamma_m^k \Pi_{j=1}^F \sum_{h_j \in \{0,1\}} exp(\sum_{il} V_i^l h_j W_{ij}^l + V_m^k h_j W_{mj}^k + b_j h_j) \\ &= \Gamma_m^k \Gamma_{j=1}^F (1 + exp(\sum_{il} V_i^l W_{ij}^l + V_m^k W_{mj}^k + b_j)) \end{aligned} \tag{13}$$

where $\Gamma_m^k = exp(V_m^k b_m^k)$.

Once the unnormalized scores are obtained, pick the rating with the maximum score as the prediction, or perform normalization over K values to get probabilities $p(V_m = k|R)$ and take the expectation $E[V_m]$ as the prediction.

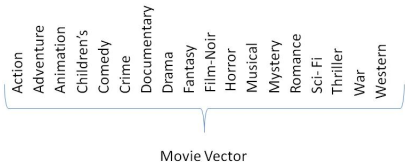

Fig. 2. Movie(genre) feature vector

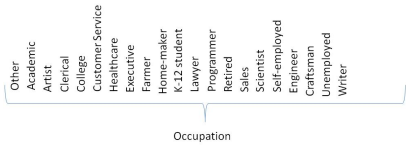

Fig. 3. Encoding Occupation Information

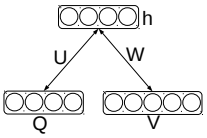

Fig. 4. RBM variant with item metadata

III. Proposed Approach

We will first discuss prior studies in label consistency penalties in latent factor model. This was introduced by one of the authors in prior studies [25]. In the next sub-section, we will describe our proposed formulation.

A. Label Consistent Latent Factor Model

The basic formulation for the latent factor model is as follows,

$$Y = M \odot R = M \odot (UV) \tag{14}$$

The symbols have their usual meaning. In the label consistent formulation [25]. The user latent factors are mapped to demographic information encoded as binarized labels and the item metadata is linearly mapped to their respective binarized metadata information. Formation of the label information from metadata is explained with examples.

Let us take the example of movie recommendation. There are many features for the movies available movie id, movie title, release date, video release date, IMDb URL, Genre (Action, Adventure, Animation, Childrens, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci- Fi, Thriller, War, Western). All except the genre information is irrelevant for choosing a movie. We have considered only the genre information here.

First, we will describe how the movie features have been-generated. We generate a binary vector from the genre, the vector contains a 1 if the movie belongs to that genre or 0 otherwise. The vector is shown in Figure 2.

Say a movie like Shawshank Redemption is tagged as crime and drama in IMDB. The corresponding feature vector will be [0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0]T; It has 1s corresponding to crime and drama and 0 everywhere else. Corresponding to each movie/item one has such a label vector. For all the movies let it be denoted as Q.

We now discuss encoding the user demographic information. Encoding the gender information is the simplest. It is a tuple encoded as [1,0]T for male and [0,1] T for female.

To encode the occupation information, we have an ordered representation of the different occupations as in Figure 3. For a particular user, one of the occupations is 1 the rest are 0s.

To encode the age information, we divide the users into several ranges; more specifically into set into 8 groups (7-14,14-21, 22-28, 29-36, 37-48, 49-55, 56-65 and 66-73). The groups are divided keeping in mind the relevance of mentality of users according to age. The particular age group, where the individual belongs to is 1 and the rest of them are 0s.

Thus, for ever user one will have a binary label vector. For all users, this is a matrix T.

When incorporated in the latent factor model [25] using the matrix factorization framework, the learning is expressed as,

$$\min_{U,V,M_1,M_2} \left\|Y - M \odot (UV)\right\|_F^2 |\delta(\|T - M_1 U\|_F^2 + \|Q - M_2 V\|_F^2) \tag{15}$$

Instead of learning the maps from the latent factors, one can also learn it directly from the ratings [25], i.e. the users ratings are linearly mapped to the users binary labels encoding the demographic information and the item ratings are mapped to their corresponding binarized metadata. This is expressed as,

$$\min_{U,V,M_1,M_2} \left\|Y - M \odot (R)\right\|_F^2 |\delta(\|T - M_1 R\|_F^2 + \|Q - R M_2\|_F^2) \tag{16}$$

Collaborative filtering is a highly-under-determined problem. We have mentioned before, that only 5% of the ratings are available and one needs to predict the remaining 95%. In such a situation, any extra information helps. It is not surprising that the metadata information in [25] indeed improve the results.

B. Label Consistent RBM

Motivated by the success of label consistent formulations in latent factor models, we propose a novel label consistent RBM to improve upon the existing CF formulation [10]. However, unlike [25], it is not possible to capture both user and item metadata in a single framework for the RBM based formulations. To capture item metadata we add genre information of movies to the simple RBM. In this variation, there are two different layers of visible units, one layer consists of ratings of N users and another layer is of binary label information (genre vector Q) for that movie.

The joint probability distribution of this model is given by,

$$p(Q, V, h) \propto e^{-E(Q,V,h)} \tag{17}$$

where we define the new energy function as follows:

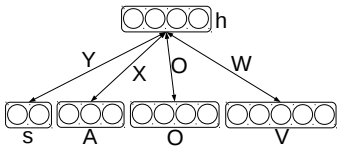

Fig. 5. RBM variant with user metadata

$$E(Q,V,h) = -h^T WV - a^T V - b^T h - c^T Q - h^T UQ \quad (18)$$

with parameters $\Theta = (W, a, b, c, U)$. The model is illustrated in Figure 4. We find the values of visible and hidden units using Equations 9, 19 and 20 respectively.

$$p(h_j = 1|V,Q) = \sigma(b_j + U_{jq} + \sum_q W_{jq} V_q) \quad (19)$$

$$p(Q_q = 1|h) = \frac{exp(c_q + \sum_j U_{jq} h_j)}{\sum_{q=1}^{Q} exp(c_q + \sum_j U_{jq} h_j)} \quad (20)$$

where σ is the logistic sigmoid. These equations are meant to capture the predictive information about the input vector as well as the target class.

To perform the learning, we use the same Equation 12, representation is as follows:

$$\begin{aligned}\Delta W_{ij}^k &= \mu \frac{\delta logp(V,Q)}{\delta W_{ij}^k} \\ &= \mu(< V_i^k h_j Q_i >_{data} - < V_i^k h_j Q_i >_{model}) \end{aligned} \quad (21)$$

where μ is learning rate.

To capture user meta data we add occupation, age, and gender to the simple RBM. In this variation, we have four different visible layers. Layer 1 consists of ratings given to all the movies by user i, and has size $M \times K$. Layer 2 captures occupation information for user i, and has size O. Layer 3 captures age information for user i, and has size A. Layer 4 captures gender information for user i, and has size S. The joint probability distribution of this model is given by:

$$p(O,S,A,V,h) \propto e^{-E(O,S,A,V,h)} \quad (22)$$

The modified energy function now becomes:

$$\begin{aligned}E(O,S,A,V,h) = &-h^t WV - a^T V - b^T h - d^T O \\ &- d^T Z - e^T A - e^T XA - f^T S - f^T YS \end{aligned} \quad (23)$$

With parameters $\Theta = (W, a, b, d, e, f, X, Z, Y)$. This model is illustrated in Figure 5. To compute the parameters, we use the following equations:

$$p(h_j = 1|V,O,S,A) = \sigma(b_j + Z_{jo} + X_{ja} + Y_{js} + \sum_i W_{ji} V_i) \quad (24)$$

For occupation of the user

$$p(O_o = 1|h) = \frac{exp(d_o + \sum_j Z_{jo} h_j)}{\sum_{o=1}^{O} exp(d_o + \sum_j Z_{jo} h_j)} \quad (25)$$

For gender of the user, following equation is used

$$p(A_a = 1|h) = \frac{exp(e_a + \sum_j X_{ja} h_j)}{\sum_{a=1}^{A} exp(f_s + \sum_j Y_{js} h_j)} \quad (26)$$

For age of the user, following equation is used

$$p(S_s = 1|h) = \frac{exp(f_s + \sum_j Y_{js} h_j)}{\sum_{s=1}^{S} exp(f_s + \sum_j Y_{js} h_j)} \quad (27)$$

To perform the learning, we use the same Equation 12, representation is as follows:

$$\begin{aligned}\Delta W_{ij}^k &= \mu \frac{\delta logp(V,O,S,A)}{\delta W_{ij}^k} \\ &= \mu(< V_i^k O_i S_i A_i h_j >_{data} - < V_i^k O_i S_i A_i h_j >_{model}) \end{aligned} \quad (28)$$

where μ is learning rate.

IV. Expertimental Evaluation

A. Description of dataset

Experiments were carried out on the popular Movielens dataset [26]. We used the 100K and 1M dataset. These are the only datasets consisting of the user and item metadata. The larger 10M dataset does not contain this (user and item metadata) information.

The 100K consists of 100,000 ratings (1-5) from 943 users and 1682 movies. Simple demographic information of the user including age, occupation, gender, and zip code is available. The zip code information is irrevelant [22] and is not used here. For items genre information is present; other information can be obtained from IMDB, however, we do not use it in our experiments. In the 1M dataset, there are 1,000,209 ratings of approximately 3900 movies by 6040 users.

B. Experimental setup and evaluation criteria

The standard experimental protocol is followed. 5 fold cross validation is performed on the standard splits.

In [19] it was argued that the factors should be sparse. This is based on the observation that it is not possible for one item to possess all types of factors. Similarly, we argue that as users, we have an affinity towards a few factors only. Therefore, user latent factors should be sparse versions of our LC-RBM formulation.

The learning rate is 0.0005 and number of epochs were 100. The number of hidden units is 100.

The quality of prediction is compared in terms of the standard metrics of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE)

Method	100K Dataset		1M Dataset	
	MAE	RMSE	MAE	RMSE
LC-RBM (Item)	0.7511	0.9451	0.7056	0.8625
Sparse LC-RBM (Item)	0.7352	0.9207	0.6895	0.8521
LC-RBM (User)	0.7709	0.9686	0.7269	0.8853
Sparse LC-RBM (User)	0.7418	0.9317	0.6995	0.8721
RBM [12]	0.8264	1.453	0.7821	1.023
PMF	0.7564	0.9639	0.7241	0.9127
LCMC	0.7193	0.9145	0.6731	0.8559

TABLE I
COMPARITIVE RESULTS

C. Results

We have compared our technique with the baseline RBM [10]. For benchmarking, the de facto standard of probabilistic matrix factorization (PMF) [27]–[29] is used. The results are shown in the following table.

Results show that our proposed formulation of supervised RBM indeed improves upon the unsupervised one [10]. Even without sparsity, we (LC-RBM item) produce better results than the benchmark RMF; the formulation using user metadata (without sparsity) is however slightly worse than PMF. With sparsity, our results improve even further. We always beat PMF. We do not beat the results from LCMC. To the best of our knowledge, it is the best-known algorithm for collaborative filtering today.

V. CONCLUSION

Today RBM is popular as a building block for deep belief network (DBN). However, an early work on the topic showed, how it can be used in collaborative filtering. However the results from RBM could not compete with matrix factorization based latent factor models, and hence there is no significant work on this topic (RBM on recommender systems) since the publication of [10] in 2007. However, RBM based pre-processing has been done for improving results of matrix factorization; in fact, it has been used by the winners of the famous Netflix competition.

This work revisits RBM for collaborative filtering. We propose a new supervised model for RBM - label consistent RBM. This has been proposed to incorporate user and item metadata information commonly found in recommender systems. This significantly improves the performance, improving upon the state-of-the-art unsupervised techniques like probabilistic matrix factorization. In future, we would like to add graph-based similarity measure [30] and item popularity information [31] to our proposed method.

REFERENCES

[1] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in *Proceedings of the 1st ACM conference on Electronic commerce*, 1999, pp. 158–166.

[2] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," in *Data mining and knowledge discovery*, vol. 5, no. 1-2, 2001, pp. 115–153.

[3] R. Van Meteren and M. Van Someren, "Using content-based filtering for recommendation," in *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, 2000, pp. 47–56.

[4] T. Hofmann, "Latent semantic models for collaborative filtering," in *TOIS*, vol. 22, no. 1, 2004, pp. 89–115.

[5] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," in *Computer*, vol. 42, no. 8, 2009.

[6] J. D. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *ICML*, 2005, pp. 713–719.

[7] J. Bennett *et al.*, "Kdd cup and workshop 2007," in *ACM SIGKDD Explorations Newsletter*, vol. 9, no. 2, 2007, pp. 51–52.

[8] J. L. Herlocker *et al.*, "An algorithmic framework for performing collaborative filtering," in *ACM SIGIR*, 1999, pp. 230–237.

[9] B. Sarwar *et al.*, "Item-based collaborative filtering recommendation algorithms," in *ICWWW*, 2001, pp. 285–295.

[10] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *ICML*, 2007, pp. 791–798.

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

[12] Y. Ouyang, W. Liu, W. Rong, and Z. Xiong, "Autoencoder-based collaborative filtering," in *NIPS*, 2014, pp. 284–291.

[13] Y. Wu *et al.*, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 2016, pp. 153–162.

[14] F. Strub and J. Mary, "Collaborative filtering with stacked denoising autoencoders and sparse inputs," in *NIPS workshop on machine learning for eCommerce*, 2015.

[15] S. Li, J. Kawale, and Y. Fu, "Deep collaborative filtering via marginalized denoising auto-encoder," in *ICIKM*, 2015, pp. 811–820.

[16] J. Abernethy *et al.*, "A new approach to collaborative filtering: Operator estimation with spectral regularization," in *Machine Learning Research*, vol. 10, no. Mar, 2009, pp. 803–826.

[17] O. Shamir and S. Shalev-Shwartz, "Collaborative filtering with the trace norm: Learning, bounding, and transducing," in *Proceedings of the 24th Annual Conference on Learning Theory*, 2011, pp. 661–678.

[18] J. Wang, A. P. De Vries, and M. J. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *ACM SIGIR*, 2006, pp. 501–508.

[19] A. Gogna and A. Majumdar, "Distributed elastic net regularized blind compressive sensing for recommender system design," in *SIMOD*, 2014, pp. 29–37.

[20] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," in *IEEE Transactions on Information Theory*, vol. 56, no. 5, 2010, pp. 2053–2080.

[21] B. Recht, "A simpler approach to matrix completion," in *Machine Learning Research*, vol. 12, no. Dec, 2011, pp. 3413–3430.

[22] A. Gogna and A. Majumdar, "A comprehensive recommender system model: Improving accuracy for both warm and cold start users," in *IEEE Access*, vol. 3, 2015, pp. 2803–2813.

[23] Q. Gu, J. Zhou, and C. Ding, "Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs," in *ICDM*, 2010, pp. 199–210.

[24] I. Sutskever and T. Tieleman, "On the convergence properties of contrastive divergence," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 789–795.

[25] A. Gogna and A. Majumdar, "Supervised learning in matrix completion framework for recommender system design." in *COMAD*, 2016, pp. 35–46.

[26] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," in *TiiS*, vol. 5, no. 4, 2016, p. 19.

[27] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *NIPS*, 2008, pp. 1257–1264.

[28] H. Shan and A. Banerjee, "Generalized probabilistic matrix factorizations for collaborative filtering," in *ICDM*, 2010, pp. 1025–1030.

[29] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *CIKM*, 2008, pp. 931–940.

[30] N. D. Phuong *et al.*, "Collaborative filtering with a graph-based similarity measure," in *ComManTel*, 2014, pp. 251–256.

[31] H. Steck, "Item popularity and recommendation accuracy," in *RecSys*, 2011, pp. 125–132.