Collaborative Deep Ranking: A Hybrid Pair-Wise Recommendation Algorithm with Implicit Feedback

Haochao Ying[1(✉)], Liang Chen[2], Yuwen Xiong[1], and Jian Wu[1]

[1] College of Computer Science and Technology, Zhejiang University, Hangzhou, China
{haochaoying,orpine,wujian2000}@zju.edu.cn
[2] School of Computer Science and Information Technology, RMIT, Melbourne, Australia
liang.chen@rmit.edu.au

Abstract. Collaborative Filtering with Implicit Feedbacks (e.g., browsing or clicking records), named as CF-IF, is demonstrated to be an effective way in recommender systems. Existing works of CF-IF can be mainly classified into two categories, i.e., point-wise regression based and pair-wise ranking based, where the latter one relaxes assumption and usually obtains better performance in empirical studies. In real applications, implicit feedback is often very sparse, causing CF-IF based methods to degrade significantly in recommendation performance. In this case, side information (e.g., item content) is usually introduced and utilized to address the data sparsity problem. Nevertheless, the latent feature representation learned from side information by topic model may not be very effective when the data is too sparse. To address this problem, we propose collaborative deep ranking (CDR), a hybrid pair-wise approach with implicit feedback, which leverages deep feature representation of item content into Bayesian framework of pair-wise ranking model in this paper. The experimental analysis on a real-world dataset shows CDR outperforms three state-of-art methods in terms of recall metric under different sparsity level.

1 Introduction

With the growing community value of personalized services, recommendation techniques have been playing an significant role in online applications [15]. To provide personalized services, users' preference from their past feedback of items is critical. Generally, users' feedback can be classified into two categories: explicit and implicit. Explicit feedback (e.g. the graded ratings 1–5 in Netflix) expresses the users' true preference, which has been well studied in many literatures. However, users may be compelled to convey their rating values in some scenarios. Moreover, users just express their behaviors implicitly in many more situations, such as browsing or not browsing, clicking or not clicking in Web sites. The meaning of unobserved items are ambiguous because users may not like these items

J. Bailey et al. (Eds.): PAKDD 2016, Part II, LNAI 9652, pp. 555–567, 2016.
DOI: 10.1007/978-3-319-31750-2_44

or may be unaware of these items. Therefore, the scenario of recommendation with implicit feedback is more challenging.

Previous works based on implicit feedback include point-wise regression and pair-wise ranking preference algorithm [1,9]. The point-wise regression algorithm supposes that users don't like all unobserved items and optimizes the absolute rating scores, while pair-wise ranking algorithm assumes that users' preference of observed items is stronger than unobserved items and then directly convert the prediction to rank. The latter algorithm actually relaxes assumption and usually obtains better performance in empirical studies. However, a user typically observes limited number of items and doesn't interact with thousands of items. Therefore, the data sparsity is a big problem for pair-wise ranking algorithm. With the increasing availability of auxiliary information about items (e.g., movie plots and item description), referred to as side information, it is wise to explore the possibility of using such information to improve the performance of pair-wise ranking algorithm through alleviating data sparsity [4].

Collaborative topic ranking (CTRank) is a recently proposed hybrid method, which seamlessly combines latent dirchlet allocation and pair-wise ranking model. Although this model learn the feature of side information associated with items, LDA is often not effective enough to learn the latent representation especially when side information is sparse [13]. Alternatively, deep learning, as a set of representation-learning methods, models multiple levels of representation of raw input by composing simple but non-linear modules that each transform the representation at one level into a representation at a higher, slightly more abstract level [3]. It has been proved that deep learning methods are expert in automatically mining and representing intricate structures in high-dimensional data.

In this paper, we propose a hybrid pair-wise recommendation approach with implicit feedback, named collaborative deep ranking (CDR), which integrates abstract representation of side information about items into Bayesian framework of pair-wise ranking model. Specifically, Stacked Denoising Autoencoders (SDAE) is exploited to extract the feature representation of item content. Cooperating with this, pair-wise ranking component can tackle sparsity problem to some extent and improve the recommendation accuracy. Note that, although CDR employs SDAE for feature representation, CDR as a generic framework also can collaborate with other deep learning methods, such as convolutional neural networks and recurrent neural networks.

The main contribution is summarized as follows:

1. We propose a hierarchical Bayesian framework, named as CDR, which combines deep feature presentation of item content and user implicit preference for sparsity reduction.
2. We conduct experiments on a real-world dataset to evaluate the effectiveness of CDR. Experimental result shows CDR outperforms three state-of-art methods in terms of recall metric under different sparsity level.

The remainder of this paper is organized as follows. Section 2 gives an overview of the related work. Section 3 demonstrates details of our proposed model. Section 4 shows the experimental results and Sect. 5 concludes the paper.

2 Related Work

In many practical recommendation scenarios, users rarely express their explicit behaviors, while implicit ones are more common (e.g. clicking and browsing history). This class of collaborative filtering with only positive examples is also called One-Class Collaborative Filtering (OCCF) [5]. There are mainly two types of existing approaches for OCCF: point-wise and pair-wise [14].

Point-wise algorithm directly optimizes the absolute value of binary rating. Hu et al. [1] estimate user-item pair preference whether the user would like or dislike the item and then assign a confidence level for this. After defining preference and confidence level, they join them into traditional probabilistic matrix factorization. However, the performance of CF-based models degrades significantly when facing data sparsity problem. Many models explore side information about items and users to alleviate this problem. Wang et al. [11] propose collaborative topic regression (CTR) for recommending scientific articles. In CTR, item content based on probabilistic topic model incorporates into traditional collaborative filtering. Based on this work, Purushotham [7] further study the influence of users' social network and propose CTR with SMF model. Recently, Wang et al. [13] employs deep learning model to automatically learn effective representation of content of items. From their experiments, we can observe that deep learning models are more appealing than traditional topic model in feature representation.

Different from point-wise model with intermediate step of predicting rating for recommendation, pair-wise algorithm generates a preference ranking of items for each user. Rendle et al. [8,9,15] assume that user prefers observed items than unobserved items and propose a generic Bayesian Personalized Ranking (BPR) framework optimization criterion. They also demonstrate matrix factorization and adaptive kNN learned by BPR are superior to the same model with respect to other criteria under AUC evaluation. Pan et al. [6] relax individual and independence assumption in BPR by adding the interaction of users and propose group Bayesian personalized ranking (GBPR). For sparsity reduction, several models extend pair-wise ranking techniques via taking extra side information into consideration. Grimberghe et al. [2] combine social graph and BPR matrix factorization for social network data. Yao et al. [14] propose a hierarchical Bayesian framework, which integrates latent dirchelet allocation into BPR matrix factorization. However, the topic model is not effective enough when side information is sparse. Therefore, in this paper, we integrate deep representation learning of content of items into pair-wise ranking model and propose a generalized hierarchical Bayesian model, called CDR.

3 Collabrative Deep Ranking

In this section, we present details of our proposed algorithm, CDR, which integrates pair-wise ranking models and side information about the items.

Notation and Problem Definition. Let $\mathcal{U}$ denote the set of users and $\mathcal{I}$ denote the set of items. The size of $\mathcal{U}$ and $\mathcal{I}$ are n and m, respectively. This

paper focuses on implicit feedback recommendation scenarios, which means the implicit interaction matrix $R \in \mathcal{U} \times \mathcal{I}$ is available. Specifically, the elements $r_{ij} = 1$ in matrix R denotes user i prefers item j, while $r_{ij} = 0$ implies that user i is not interesting in item j or might not observe item j yet. For a given user i, pair-wise algorithms [9] suppose that user i prefers item j over item k if and only if $j \in \mathcal{I}^+$ and $k \in \mathcal{I} \backslash \mathcal{I}^+$, where $\mathcal{I}^+ = \{j : r_{ij} = 1\}$.

Except the observed binary matrix R, side information about items could be collected in many scenarios (e.g. item profile in Amazon). Given an $m \times s$ matrix X_c presents the side information about all items, the j-th row denotes the bag-of-words vector of item j based on vocabulary of size S (i.e. $X_{c,j*}$). Let u_i, v_j denote the latent factor with low dimension K of user i and item j, respectively. Our objective is to learn the latent factor $U = (u_i)_{i=1}^n$ and $V = (v_j)_{j=1}^m$ from implicit interaction and item information matrix for recommending an personalized ranking list for users.

Stacked Denoising Autoencoders. Generally, a good representation of side information about items can improve performance of Recommender System. Denosing Autoencoders (DAE) [10] learns an compressed representation from corrupted input to recover the clean input through a feedforward neural network. SDAE stacks DAE to form a deep network by feeding the output code of DAE found on the layer below as input to the current layer and the highest level output representation is used as item feature. An SDAE network is to minimize the regularized optimization problem as below:

$$\min_{\{W_l\},\{b_l\}} \|X_c - X_L\|_F^2 + \lambda_w \sum_l (\|W\|_F^2 + \|b\|_F^2), \tag{1}$$

where W_l and b_l is the weight matrix and bias vector of layer l, L is the number of layers, and λ_w is the regularization hyperparameter.

Supposing that the corrupted input X_0 and the clean input X_c are observed variables, SDAE can be generalized as a probabilistic model [12]. The generative process is as follows:

1. For each layer l of the SDAE network,
   (a) For each column n of the weight matrix W_l, draw $W_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l})$.
   (b) Draw the bias vector $b_l \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l})$
   (c) For each row j of X_l, draw $X_{l,j*} \sim \mathcal{N}(\sigma(X_{l-1,j*} W_l + b_l), \lambda_s^{-1} I_{K_l})$
2. For each item j, draw a clean input,

$$X_{c,j*} \sim \mathcal{N}(X_{L,j*}, \lambda_n^{-1} I_m)$$

where I_{K_l} is a K-dimensional identity matrix of layer l, λ_w, λ_s, λ_n is the hyperparameters and $\sigma(.)$ is the sigmoid function Through maximizing a posteriori estimation, the model will degenerate to be the original SDAE if λ_s goes to infinity (i.e. $X_{l-1,j*} = \sigma(X_{l-1,j*} * W_l + b_l)$). After this process, $X_{\frac{L}{2}}$ could effectively present the latent feature representation of side information about all items.

Collabrative Deep Ranking. CDR exploits pair-wise preferences and content-based items feature together for collaborative filtering. Figure 1 shows the

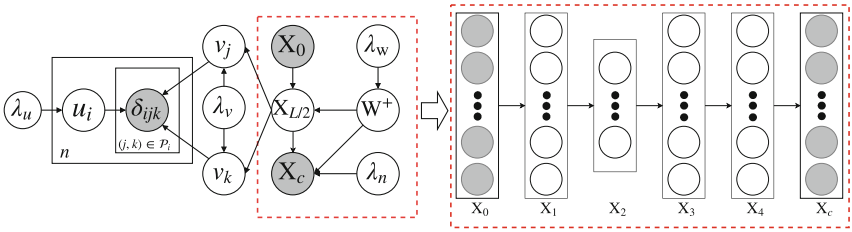


Fig. 1. The graphic model of CDR. SDAE with $L = 4$ is presented inside the dashed rectangle. Note that W^+ denotes the set of weight matrices and bias vectors of all layers (Color figure online).

graphic model of CDR. Obviously, there are two generative processes in our model. First, the original SDAE process (in the red dashed rectangle) extracts feature representation from side information about items and then integrates them into latent factor of items in pair-wise ranking model. Second, the pair-wise ranking model captures special relationship $\delta_{ijk} = r_{ij} - r_{ik}$, which delegates the preference of user i on item j and k. Unlike the point-wise approach directly predicting the value r_{ij}, pair-wise approach instead classifies the difference of $r_{ij} - r_{ik}$ [9]. The generative process of CDR is as follows:

1. For each layer l of the SDAE network,
   (a) For each column q, draw the weight matrix and bias vector W_l^+, draw $W_{l,*q}^+ \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l})$.
   (b) For each row j of X_l, draw $X_{l,j*} \sim \mathcal{N}(\sigma(X_{l-1,j*} W_l + b_l), \lambda_s^{-1} I_{K_l})$
2. For each item j,
   (a) Draw a clean input $X_{c,j*} \sim \mathcal{N}(X_{L,j*}, \lambda_n^{-1} I_m)$
   (b) Draw a latent item offset vector $\epsilon_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K)$and then set the latent item vector to be:
$$v_j = \epsilon_j + X_{\frac{L}{2},j*}^T$$
3. For each user i,
   (a) Draw user factor vector $u_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K)$
   (b) For each pair-wise preference $(j,k) \in \mathcal{P}_i$, where $\mathcal{P}_i = \{(j,k) : r_{ij} - r_{ik} > 0\}$, draw the estimator,
$$\delta_{ijk} \sim \mathcal{N}(u_i^T v_j - u_i^T v_k, c_{ijk}^{-1})$$

Where confidence parameter c_{ijk} denotes how much user i prefers item j than item k. For simplicity, we set $c_{ijk} = 1$ in the experiments. Note that the linkage between SDAE and pair-wise ranking model is the middle layer $X_{\frac{L}{2},j*}$. In the extreme case, if $\varepsilon_j = 0$, the latent factor of items completely generates from content information, which will ignore the information contained in user preference matrix.

For learning model parameters, maximum a posterior probability (MAP) estimator can be utilized. Through Bayesian inference, we have

$$P(U, V, X_l, W^+|\delta, X_0, X_c, \lambda_u, \lambda_v, \lambda_s, \lambda_w, \lambda_n) \propto$$
$$P(U|\lambda_u)P(V|\lambda_v, X_{\frac{L}{2}})P(W^+|\lambda_w)P(\delta|U, V)P(X_c|X_L, \lambda_n)P(X_l|X_{l-1}, W_l^+, \lambda_s) \tag{2}$$

Because we place Gaussian priors on user, item and wight matrix, corresponding conditional probability is

$$\begin{aligned} P(U|\lambda_u) &= \prod_{i=1}^{n} \mathcal{N}(u_i|0, \lambda_u^{-1} I_K) \\ P(V|\lambda_v, X_{\frac{L}{2}}) &= \prod_{j=1}^{m} \mathcal{N}(v_j|X_{\frac{L}{2}}, \lambda_v^{-1} I_K) \\ P(W^+|\lambda_w) &= \prod_{l=1}^{L} \mathcal{N}(0, \lambda_w^{-1} I_{K_l}) \end{aligned} \tag{3}$$

Similarly, we have below corresponding conditional probability based on assumption of Gaussian distribution:

$$\begin{aligned} P(X_c|X_L, \lambda_n) &= \prod_{j=1}^{m} \mathcal{N}(X_{c,j*}|X_{L,j*}, \lambda_n^{-1} I_m) \\ P(X_l|X_{l-1}, W_l^+, \lambda_s) &= \prod_{j=1}^{m} \mathcal{N}(X_{l,j*}|\sigma(X_{l-1,j*} W_l + b_l), \lambda_s^{-1} I_{K_l}) \\ P(\delta|U, V) &= \prod_{ijk} \mathcal{N}(\delta_{ijk}|u_i^T v_j - u_i^T v_k, c_{ijk}^{-1}) \end{aligned} \tag{4}$$

Similar to the generalized SDAE, λ_s goes to infinity and maximization of posterior probability is equivalent to maximizing the joint log-likelihood of U, V, X_l, X_c, W^+, and δ given λ_u, λ_v, λ_n,

$$\begin{aligned} \mathcal{L} = &- \sum_{ijk} \frac{c_{ijk}}{2} (\delta_{ijk} - (u_i^T v_j - u_i^T v_k))^2 - \frac{\lambda_w}{2} \sum_j (W_l^2 + b_l^2) \\ &- \frac{\lambda_u}{2} \sum_i u_i^T u_i - \frac{\lambda_v}{2} \sum_j (v_j - X_{\frac{L}{2},j*}^T)^2 - \frac{\lambda_n}{2} \sum_j (X_{L,j*} - X_{c,j*})^2 \end{aligned} \tag{5}$$

In the generative process, we assume that the preference δ_{ijk} follows Gaussian Distribution. However, similar loss function can be obtained with different assumption on δ_{ijk} (e.g. Bernoulli distribution in [14]). Note that the model CTRank, proposed in [14], is analogous to our model, which also combines pair-wise ranking and side information about items. The big difference is that CTRank extracts topic proportions from content of items to conduct the learning of latent factors for ranking, while CDR exploits deep network to mine effective feature representation of items. Note that prior distribution of LDA-based models is difficult to define. What's worse, topic proportions can not effectively represent the latent feature of items when side information is very sparse. As showed in the experiments, CDR gets better performance.

The first term in Eq. 5 extracts user preference from implicit matrix $\mathcal{R}$ to construct pair-wise ranking loss, while the fourth term integrates content of items. Therefore, if two item j and k have similar side information (i.e. similar $X_{\frac{L}{2}}$), the distance between v_j and v_k will be reduced. As we have mentioned,

$X_{\frac{L}{2}}$ serves as a bridge between pair-wise ranking and SDAE model. When λ_v/λ_n goes to positive infinity, the disappeared reconstruction error will lead to invalid feature representation $X_{\frac{L}{2}}$, meanwhile $X_{\frac{L}{2}}$ dominate the learning process of V. On the other hand, when λ_v/λ_n approaches to zeros, CDL will decouple into two models and the learned V is not influenced by side information about items. Both extreme cases demonstrate bad performance in the experiments.

Parameter Learning. Similar to [1,13,14], we optimize this function using coordinate ascent by alternatively optimizing latent factors u_i, v_j and weight matrix & bias vector W^+. Given a current estimate of W^+, we update u_i, v_j and v_k based on the following stochastic Newton-Raphson rules:

$$\begin{aligned} u_i &= u_i - \alpha \frac{\lambda_u u_i - c_{ijk}\mathcal{E}_{ijk}(v_j - v_k)}{\lambda_u + c_{ijk}(v_j - v_k)^T(v_j - v_k)} \\ v_j &= v_j - \alpha \frac{\lambda_v(v_j - X^T_{\frac{L}{2},j*}) - c_{ijk}\mathcal{E}_{ijk}u_i}{\lambda_v + c_{ijk}u_i^T u_i} \\ v_k &= v_k - \alpha \frac{\lambda_v(v_k - X^T_{\frac{L}{2},k*}) + c_{ijk}\mathcal{E}_{ijk}u_i}{\lambda_v + c_{ijk}u_i^T u_i} \end{aligned} \tag{6}$$

where α is learning rate and $\mathcal{E}_{ijk} = \delta_{ijk} - (u_i^T v_j - u_i^T v_k)$. Note that when updating, bootstrap sampling is applied to sample observed item j and unobserved item k of user i [9].

Given U and V, wight matrix W_l and bias vector b_l for each layer update by back-propagation learning algorithm. The gradient of $\mathcal{L}$ with respect to W_l and b_l is as follows:

$$\begin{aligned} \nabla_{W_l}\mathcal{L} &= -\lambda_w W_l - \lambda_v \sum_j \nabla_{W_l} X^T_{\frac{L}{2},j*}(X_{\frac{L}{2},j*} - v_j) - \lambda_n \sum_j \nabla_{W_l} X_{L,j*}(X_{L,j*} - X_{c,j*}) \\ \nabla_{b_l}\mathcal{L} &= -\lambda_w b_l - \lambda_v \sum_j \nabla_{b_l} X^T_{\frac{L}{2},j*}(X_{\frac{L}{2},j*} - v_j) - \lambda_n \sum_j \nabla_{b_l} X_{L,j*}(X_{L,j*} - X_{c,j*}) \end{aligned} \tag{7}$$

Prediction. After learning the optimal parameters U,V,W^+, we predict R_{ij} from its expectation:

$$E[R_{ij}|U, V, W^+, ...] \approx u_i^T(X_{\frac{L}{2}} + \epsilon_j) = u_i v_j,$$

and then a ranked list of items is generated for each user based on these prediction values.

Complexity Analysis. According to updating rules, the complexity of computing U is approximately $O(nrK)$ where r is the average number of items a user interacts. The complexity of computing the output of encoder is controlled by the computation of first layer. Therefore, the complexity of updating V is $O(nrK + sK_1)$, where K_1 is the dimension of first layer. The complexity of updating all the wights and bias is $O(msK_1)$. Hence, the total complexity is $O(2nrK + sK_1 + msK_1)$.

4 Experiments

In this section, we compare performance of our approach with some state-of-art algorithms. All experiments are conducted on a server with 2 Intel E5-2620 CPUs and 1 GTX Titan GPU.

Datasets. To effectively illustrate the performance of CDR, we use the same dataset in [11,13,14]. The dataset is collected from CiteULike[1], which provides service for managing and discovering articles for users. In this dataset, if a user has collected an article in his library, we consider that the user implicitly prefers the article, rating as '1' otherwise '0'. The preliminary statistics shows that the dataset contains 5,551 users and 16,980 articles with 204,986 user-item preference pairs. Note that the sparsity is 99.78 % and each user has at least 10 articles in their preference library. To obtain the side information about articles, the title and abstract of articles are exploited. After removing stop words, we extract 8000 distinct words through sorting their TF-IDF values. As a result, the size of X_c is 16980×8000 as clean input of SDAE.

Evaluation. Similar to [7,11], we employ the metric recall to quantize the performance of recommendation, since the metric precision is not suited to implicit feedback datasets. Because the meaning of zero entry in the user-item matrix is ambiguous, which represent either user don't like item or is unaware of item. Instead, the positive rating (e.g. $r_{ij} = 1$) only hints the user i likes the item j, we focus on recall metric. Specifically, after predicting the ratings in the test dataset, we sort them and recommend top M items for each user. The recall@M is defined as follows:

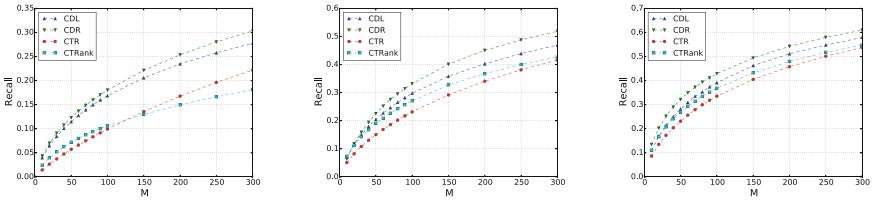
$$recall@M = \frac{\text{number of items the user likes in Top M}}{\text{total number of items the user likes}}$$

Average recall from all users points out the performance of method.

Baselines and Experiments Setting. In order to evaluate effectiveness of our approach, CDR, we compare it with three state-of-art hybrid recommendation algorithm for implicit feedback as follows:

- **CTR.** Collaborative Topic Regression is a point-wise algorithm, which combines probability matrix factorization and latent dirchelet allocation [11].
- **CTRank.** Collaborative Topic Ranking, a pair-wise algorithm, which integrates side information of items into Bayesian personalized ranking [14]. With different assumption of preference, Two algorithms, CTRank-log and CTRank-squared, have been proposed. We choose CTRank-squared as our compared approach, because it has higher performance than CTRank-log.
- **CDL.** Collaborative deep learning is a point-wise hierarchical Bayesian model, which first tightly couples deep representation feature of content information and collaborative filtering [13].
- **CDR.** Collaborative Deep Ranking is our proposed model described in Sect. 3.

[1] http://www.citeulike.org/.



(a) $P = 1$ (b) $P = 5$ (c) $P = 10$

Fig. 2. Performance comparison of CDR, CDL, CTRank, CTR under different P.

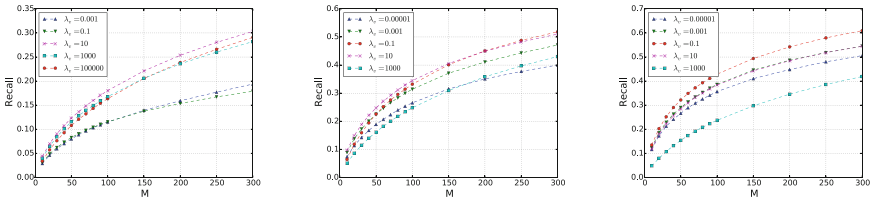
Similar to [13], we randomly choose P items from each user to consist of train set and take all the rest as test set in the experiments. In particular, we vary train set sparsity is 0.006 %, 0.03 %, 0.06 % (i.e. $P = 1, 5, 10$). Each approach is performed 5 times with different random seeds for each sparsity and the average performance is reported. The grid search is applied to find optimal hyperparameters for each approach. For CTR, $\lambda_u = 0.1$, $\lambda_v = 10$, $a = 1$, $b = 0.01$, $K = 50$, and $\alpha = 1$ can reach good performance (note that α is the dirchelet prior). For CTRank, we find $\lambda_u = 0.025$, positive $\lambda_v = 0.25$, negative $\lambda_v = 0.025$, $K = 200$, and $c = 1$ can achieve best results. For CDL, we set the same parameters of $a = 1$, $b = 0.01$, $K = 50$, $\lambda_w = 0.0001$, and a 2-layer SDAE architecture '8000-200-50-200-8000' for different P. Otherwise, $\lambda_u = 1$, $\lambda_v = 10$, and $\lambda_n = 1000$ when $P = 1$ and 5, while $\lambda_u = 0.1$, $\lambda_v = 1$, and $\lambda_n = 100$ when $P = 10$.

In the pretrain of CDR, SDAE employs a mixture of edge detectors and grating filters (i.e. masking noise) with a noise level of 30 % to obtain the corrupted input X_0 from the clean input X_c. Meanwhile, dropout rate is set to 0.1 for achieving adaptive regularization when the number of layers is more than 2. The number of hidden units K_l is set to 1000 ($l \neq \frac{L}{2}$), while the number of middle layer is 200. That is, the dimension of feature representation and latent factor u_i, v_j is 200. Note that K_0 and K_L are equal to the size of vocabulary. After grid searching, we find that the hyperparameters $\lambda_u = 0.01$, $\lambda_v = 0.1$, $\lambda_n = 5$, and $\lambda_w = 0.0001$ can achieve good performance when $P = 5$ and $P = 10$, while we set larger hyperparameters (i.e. $\lambda_u = 1$, $\lambda_v = 10$, $\lambda_n = 1000$, and $\lambda_w = 0.0001$) to prevent overfitting when $P = 1$.

Comparison. Figure 2 provides comparison results of CDR, CDL, CTRank and CTR under different sparsity. A 3-layers CDR '8000-1000-200-1000-8000' is used. It can be observed that our proposed approach outperforms other three methods at all sparsity levels. As a whole, baseline CDL performs better than CTR and CTRank. That is, deep learning approach (e.g. SDAE) can admire better feature quality of side information about items than topic model (e.g. LDA). The reason may be that deep learning approach captures distributed features, while the features (i.e., topics) in LDA is independent. Otherwise, CTRank outperforms CTR in most case (both models use LDA model) and CDR outperforms CDL (both model employ deep learning architecture). Therefore, pair-wise algorithm

Table 1. Impact of #layers at recall @300 under different P(%)

#layers	1	2	3	4
$P = 1$	9.74	13.43	30.32	30.89
$P = 5$	49.62	49.26	51.84	47.07
$P = 10$	61.09	59.03	60.96	59.41



(a) $P = 1$ (b) $P = 5$ (c) $P = 10$

Fig. 3. The impact of λ_v/λ_n under different P.

with directly optimizing ranking has advantage over point-wise method which optimizes rating. Concretely, when recommending 300 articles CDR relatively improves 36.58 %, 66.96 %, 9.22 % than CTR, CTRank, CDL respectively at $P = 1$ while the value is 13.44 %, 11.34 %, 5.25 % at $P = 10$. Thus, when data is sparse, the relative improvement is more significant. That is, CDR can alleviate data sparsity to some extent.

Impact of #layers. Table 1 presents the recall@300 under different P with various #layers. As we can observe, when the number of layers is 1 and 2, the recall is quite low at $P = 1$, while the performance significantly enhance with the number of layers growing. That is, the performance of recommendation depends on the quality of feature representation of side information about items when the data is extremely sparsity. As reducing the sparsity degree, the effective of pair-wise ranking component begins to present and CDR starts to overfit when $\#layers = 4$ and $P = 5$. We also can find that the recall value is similar when $P = 10$ in different number of layers. That means with increasing of train set size, pair-wise ranking model can guide the further learning of features in CDR model.

Impact of λ_v/λ_n. Figure 3 shows the impact of λ_v/λ_n under different P, via changing λ_v and fixing other hyperparameters. We can observe that as increasing or reducing the λ_v from the optimal λ_v, the performance degrades gradually. This result is consistent to the explanation in Sect. 3. When λ_v/λ_n is large, the side information about items dominates the learning process of V and the performance purely depend on $X_{\frac{L}{2}}$. When λ_v/λ_n is small, the performance purely generates by the pair-wise ranking component. The experimental result indicates that appropriately combining pair-wise ranking and content of items can achieve better performance than in above two extreme case.

Table 2. An example of validity of CDR.

Top 3 topics	1. users-user-semantic-similarity-collaborative-filtering-items-recommendations-recommender-implicit	
	2. social-individuals-tags-tagging-tag-navigation-networking-ties-emergent-popularity	
	3. web-search-pages searching-page-engine-engines-google-searches-pagerank	
Top 10 articles	1. getting our head in the clouds toward evaluation studies of tagclouds	False
	2. usage patterns of collaborative tagging systems	True
	3.tagbased social interest discovery	False
	4. recommending scientific articles using citeulike	True
	5. collaborative filtering recommender systems	True
	6. open user profiles for adaptive news systems help or harm	True
	7. can all tags be used for search	True
	8. optimizing web search using social annotations	True
	9. evaluating collaborative filtering recommender systems	True
	10. can social bookmarking improve web search	True

Latent Factor Interpretability. To demonstrate the validity of CDR deeply, Table 2 show one example users of top 3 topic of his all articles and the top 10 recommended articles under the setting $P = 10$. From the top 2 topics, we can speculate the user focus on tag recommendation research, while the user also study web search based on the third topic. CDR successfully captures all three topics and reach 80 % recall when recommending top 10 articles. It is worth mentioning, the rank of recommended articles of the first topic is higher than the third topic.

5 Conclusion

In this paper, we propose a hybrid recommendation approach (CDR) with implicit feedback. Specifically, CDR employs SDAE to extract deep feature representation from side information and then integrates into pair-wise ranking model for alleviating sparsity reduction. Our study presents that CDR outperforms other three state-of-art algorithms at all sparsity level. In the future, we plan to use other deep learning methods to replace SDAE for boosting further performance in our hierarchical Bayesian framework. For example, the convolutional neural network which considers the context and order of words may improve the performance. Beyond that, we also consider how to incorporate other side information into our framework, such as users social network and items relationship.

Acknowledgement. This research was partially supported by the Natural Science Foundation of China under grant of No. 61379119, Science and Technology Program of Zhejiang Province under grant of No. 2013C01073, the Open Project of Qihoo360 under grant of No. 15-124002-002.

References

1. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Eighth IEEE International Conference on Data Mining, pp. 263–272. IEEE (2008)
2. Krohn-Grimberghe, A., Drumond, L., Freudenthaler, C., Schmidt-Thieme, L.: Multi-relational matrix factorization using Bayesian personalized ranking for social network data. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, pp. 173–182. ACM (2012)
3. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
4. Ning, X., Karypis, G.: Sparse linear methods with side information for top-n recommendations. In: Proceedings of the Sixth ACM Conference on Recommender Systems, pp. 155–162. ACM (2012)
5. Pan, R., Zhou, Y., Cao, B., Liu, N.N., Lukose, R., Scholz, M., Yang, Q.: One-class collaborative filtering. In: Eighth IEEE International Conference on Data Mining, pp. 502–511. IEEE (2008)
6. Pan, W., Chen, L.: GBPR: group preference based bayesian personalized ranking for one-class collaborative filtering. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp. 2691–2697. AAAI Press (2013)
7. Purushotham, S., Liu, Y., Kuo, C.C.J.: Collaborative topic regression with social matrix factorization for recommendation systems. In: Proceedings of the 29th International Conference on Machine Learning (2012)
8. Rendle, S., Freudenthaler, C.: Improving pairwise learning for item recommendation from implicit feedback. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pp. 273–282. ACM (2014)
9. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461. AUAI Press (2009)
10. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**, 3371–3408 (2010)
11. Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 448–456. ACM (2011)
12. Wang, H., Shi, X., Yeung, D.Y.: Relational stacked denoising autoencoder for tag recommendation. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
13. Wang, H., Wang, N., Yeung, D.Y.: Collaborative deep learning for recommender systems. In: Twenty-First ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (2015)

14. Yao, W., He, J., Wang, H., Zhang, Y., Cao, J.: Collaborative topic ranking: leveraging item meta-data for sparsity reduction. In: Twenty-Ninth AAAI Conference on Artificial Intelligence(2015)
15. Zhong, H., Pan, W., Xu, C., Yin, Z., Ming, Z.: Adaptive pairwise preference learning for collaborative recommendation with implicit feedbacks. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 1999–2002. ACM(2014)