Collaborative Deep Learning for Recommender Systems

Hao Wang, Naiyan Wang, Dit-Yan Yeung
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
hwangaz@cse.ust.hk, winsty@gmail.com, dyyeung@cse.ust.hk

Abstract

Collaborative filtering (CF) is a successful approach commonly used by many recommender systems. Conventional CF-based methods use the ratings given to items by users as the sole source of information for learning to make recommendation. However, the ratings are often very sparse in many applications, causing CF-based methods to degrade significantly in their recommendation performance. To address this sparsity problem, auxiliary information such as item content information may be utilized. Collaborative topic regression (CTR) is an appealing recent method taking this approach which tightly couples the two components that learn from two different sources of information. Nevertheless, the latent representation learned by CTR may not be very effective when the auxiliary information is very sparse. Inspired by recent success demonstrated by deep learning models, we propose in this paper a hierarchical Bayesian model called collaborative deep learning (CDL), which tightly couples a Bayesian formulation of the stacked denoising autoencoder and probabilistic matrix factorization. Extensive experiments on real-world datasets show that CDL can significantly advance the state of the art.

1 Introduction

Due to the abundance of choice in many online services, recommender systems (RS) now play an increasingly significant role. For individuals, using RS allows us to make more effective use of information. Besides, many companies (e.g., Amazon and Netflix) have been using RS extensively to target their customers by recommending products or services. Existing methods for RS can roughly be categorized into three classes [6]: content-based methods, collaborative filtering (CF) based methods, and hybrid methods. Content-based methods [14] make use of user profiles or product descriptions for recommendation. CF-based methods [19, 23] use the past activities or preferences, such as user ratings on items, without using user or product content information. Hybrid methods [1, 20, 15] seek to get the best of both worlds by combining content-based and CF-based methods.

Due to privacy concerns, it is generally more difficult to collect user profiles than past activities. Nevertheless, CF-based methods do have their limitations. The prediction accuracy often drops significantly when the ratings are very sparse. Moreover, they cannot be used for recommending new products which have yet to receive rating information from users. Consequently, it is inevitable for CF-based methods to exploit auxiliary information and hence hybrid methods have gained popularity in recent years.

According to whether two-way interaction exists between the rating information and auxiliary information, we may further divide hybrid methods into two sub-categories: loosely coupled and tightly coupled methods. Loosely coupled methods like [24] process the auxiliary information once and then use it to provide features for the CF models. Since information flow is one-way, the rating information cannot provide feedback to guide the extraction of useful features. For this sub-category, improvement often has to rely on a manual and tedious feature engineering process. On the contrary, tightly coupled methods like [28] allow two-way interaction. On one hand, the rating information can guide the learning of features. On the other hand, the extracted features can further improve the predictive power of the CF models (e.g., based on matrix factorization of the sparse rating matrix). With two-way interaction, tightly coupled methods can automatically learn features from the auxiliary information and naturally balance the influence of the rating and auxiliary information. This is why tightly coupled methods often outperform loosely coupled ones [29].

Collaborative topic regression (CTR) [28] is a recently proposed tightly coupled method. It is a probabilistic graphical model that seamlessly integrates a topic model, latent Dirichlet allocation (LDA) [5], and a model-based CF method, probabilistic matrix factorization (PMF) [23]. CTR is an appealing method in that it produces promising and interpretable results. Nevertheless, the latent representation learned is often not effective enough especially when the auxiliary information is very sparse. It is this representation learning problem that we will focus on in this paper.

On the other hand, deep learning models recently show great potential for learning effective representations and deliver state-of-the-art performance in computer vision [30] and natural language processing [12, 22] applications. In deep learning models, features are learned in a supervised or unsupervised manner. Although they are more appealing than shallow models in that the features can be learned automatically (e.g., effective feature representation is learned from text content), they are inferior to shallow models such as CF in capturing and learning the similarity and implicit relationship between items. This calls for integrating deep learning with CF by performing deep learning collaboratively. In this paper, we develop a hierarchical Bayesian model called collaborative deep learning (CDL) as a novel tightly coupled method for RS. We first present a Bayesian formulation of a deep learning model called stacked denoising autoencoder (SDAE) [26]. With this, we then present our CDL model which tightly couples the Bayesian SDAE with PMF to allow two-way interaction between learning the rating matrix in PMF and learning the Bayesian SDAE. Experiments show that CDL significantly outperforms the state of the art. Note that although we present CDL as using SDAE for its feature learning component, CDL is actually a more general framework which can also admit other deep learning models such as deep Boltzmann machines [21], recurrent neural networks [8], and convolutional neural networks [13].

The main contribution of this paper is summarized below:

- By performing deep learning collaboratively, CDL can simultaneously extract deep effective feature representation from content and capture the similarity and implicit relationship between items (and users). The learned representation may also be used for tasks other than recommendation.
- Unlike previous deep learning models which use simple target like classification [12] and reconstruction [26], we propose to use CF as a more complex target in a probabilistic framework.
- To the best of our knowledge, CDL is the first tightly coupled method to bridge the gap between state-of-the-art deep learning models and hierarchical Bayesian models for RS. Besides, CDL, as a hierarchical Bayesian model, can be easily extended to incorporate other auxiliary information to further boost the performance.
- Extensive experiments on real-world datasets show that CDL can significantly advance the state of the art.

2 Notation and Problem Formulation

Similar to the work in [28], the recommendation task considered in this paper takes implicit feedback [10] as the training and test data. The entire collection of J items (articles) is represented by a J-by-S matrix $\mathbf{X}_c$, where row j is the bag-of-words vector $\mathbf{X}_{c,j*}$ for item j based on a vocabulary of size S. With I users, we define an I-by-J binary rating matrix $\mathbf{R} = [r_{ij}]_{I\times J}$. Specifically, $r_{ij} = 1$ if user i has article j in his or her personal library and $r_{ij} = 0$ otherwise. Given part of the ratings in $\mathbf{R}$ and the article content information $\mathbf{X}_c$, the problem is to predict the other ratings in $\mathbf{R}$. Note that although we focus on article recommendation like [28] in this paper, our model is general enough to handle other recommendation tasks (e.g., tag recommendation).

The matrix $\mathbf{X}_c$ plays the role of clean input to the SDAE while the noise-corrupted matrix, also a J-by-S matrix, is denoted by $\mathbf{X}_0$. The output of layer l of the SDAE is denoted by $\mathbf{X}_l$ which is a J-by-K_l matrix. Similar to $\mathbf{X}_c$, row j of $\mathbf{X}_l$ is denoted by $\mathbf{X}_{l,j*}$. $\mathbf{W}_l$ and $\mathbf{b}_l$ are the weight matrix and bias vector, respectively, of layer l, $\mathbf{W}_{l,*n}$ denotes column n of $\mathbf{W}_l$, and L is the number of layers. For convenience, we use $\mathbf{W}^+$ to denote the collection of all layers of weight matrices and biases. Note that an $L/2$-layer SDAE corresponds to an L-layer network.

3 Background

We first briefly review the SDAE and CTR models which serve as the background for our proposed model to be presented in the next section.

3.1 Stacked Denoising Autoencoders

SDAE [26] is a feedforward neural network for learning representations (encoding) of the input data by learning to predict the clean input itself in the output, as shown in Figure 1. Usually the hidden layer in the middle, i.e., $\mathbf{X}_2$ in the figure, is constrained to be a bottleneck and the input layer $\mathbf{X}_0$ is a corrupted version of the clean input data. An SDAE solves the following optimization problem:

$$\min_{\{\mathbf{W}_l\},\{\mathbf{b}_l\}} \|\mathbf{X}_c - \mathbf{X}_L\|_F^2 + \lambda \sum_l \|\mathbf{W}_l\|_F^2,$$

where λ is a regularization parameter and $\|\cdot\|_F$ denotes the Frobenius norm.

3.2 Collaborative Topic Regression

CTR is proposed in [28] as a model that combines PMF and LDA for recommending scientific papers to users. With the LDA component, CTR produces much more interpretable results than PMF with higher accuracy.

Suppose there are K topics each of which corresponds to a distribution over the words in a vocabulary. We denote the K topics by $\boldsymbol{\beta} = \boldsymbol{\beta}_{1:K}$. Let $\mathbf{I}_K$ denote a K-by-K identity matrix and the confidence parameter c_{ij} is defined as follows:

$$c_{ij} = \begin{cases} a & \text{if } r_{ij} = 1 \\ b & \text{if } r_{ij} = 0, \end{cases} \tag{1}$$

where a and b are tunable hyperparameters with $a > b > 0$. The generative process of CTR is as follows:

1. Draw a latent user vector for each user i: $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \lambda_u^{-1}\mathbf{I}_K)$.
2. For each item j,
   (a) Draw the topic proportions $\boldsymbol{\theta}_j \sim$ Dirichlet(α).
   (b) Draw a latent item offset vector $\boldsymbol{\epsilon}_j \sim \mathcal{N}(\mathbf{0}, \lambda_v^{-1}\mathbf{I}_K)$, then set the latent item vector as: $\mathbf{v}_j = \boldsymbol{\epsilon}_j + \boldsymbol{\theta}_j$.
   (c) For each word w_{jn} in a document (item) $\mathbf{w}_j$,
      i. Draw a topic assignment $z_{jn} \sim$ Mult$(\boldsymbol{\theta}_j)$.
      ii. Draw a word $w_{jn} \sim$ Mult$(\boldsymbol{\beta}_{z_{jn}})$.
3. Draw a rating r_{ij} for each user-item pair (i, j): $r_{ij} \sim \mathcal{N}(\mathbf{u}_i^T\mathbf{v}_j, c_{ij}^{-1})$.

The key of CTR lies in the latent item offset vector $\boldsymbol{\epsilon}_j$. With this offset, the latent item vector $\mathbf{v}_j$ can be close to the topic proportions $\boldsymbol{\theta}_j$, but it may also deviate from it when necessary according to the rating information. Another nice property of using such Gaussian offset is that, although the LDA and PMF components are seamlessly integrated, sufficient flexibility is still maintained during the learning process for both components to exert their effects separately. The graphical model of CTR is depicted in Figure 1.

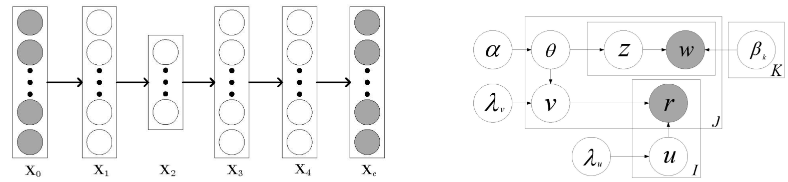


Figure 1: A 2-layer SDAE with $L = 4$ (left) and the graphical model of CTR (right).

4 Collaborative Deep Learning

We are now ready to present details of our CDL model. We first give a Bayesian formulation of SDAE. This is then followed by the presentation of CDL as a hierarchical Bayesian model that tightly integrates the Bayesian SDAE and PMF.

4.1 Bayesian SDAE

If we assume that both the clean input $\mathbf{X}_c$ and the corrupted input $\mathbf{X}_0$ are observed, similar to [4, 16, 3, 7], we can define the following generative process:

1. For each layer l of the SDAE network,
   (a) For each column n of the weight matrix $\mathbf{W}_l$, draw $\mathbf{W}_{l,*n} \sim \mathcal{N}(\mathbf{0}, \lambda_w^{-1}\mathbf{I}_{K_l})$.
   (b) Draw the bias vector $\mathbf{b}_l \sim \mathcal{N}(\mathbf{0}, \lambda_w^{-1}\mathbf{I}_{K_l})$.
   (c) For each row j of $\mathbf{X}_l$, draw $\mathbf{X}_{l,j*} \sim \text{Delta}(\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l))$.
2. For each item j, draw a clean input $\mathbf{X}_{c,j*} \sim \mathcal{N}(\mathbf{X}_{L,j*}, \lambda_n^{-1}\mathbf{I}_J)$. [1]

$\text{Delta}(\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l))$ denotes a Dirac delta distribution [25] centered at $\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l)$, where $\sigma(\cdot)$ is the sigmoid function. We note that it is equivalent to a Gaussian distribution $\mathcal{N}(\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l), \lambda^{-1}\mathbf{I}_{K_l})$ with λ approaching infinity.

Note that the first $L/2$ layers of the network act as an encoder and the last $L/2$ layers act as a decoder. Maximization of the posterior probability is equivalent to minimization of the reconstruction error with weight decay taken into consideration.

4.2 Collaborative Deep Learning

Using the Bayesian SDAE as a component, the generative process of CDL is defined as follows:

1. For each layer l of the SDAE network,
   (a) For each column n of the weight matrix $\mathbf{W}_l$, draw $\mathbf{W}_{l,*n} \sim \mathcal{N}(\mathbf{0}, \lambda_w^{-1}\mathbf{I}_{K_l})$.
   (b) Draw the bias vector $\mathbf{b}_l \sim \mathcal{N}(\mathbf{0}, \lambda_w^{-1}\mathbf{I}_{K_l})$.
   (c) For each row j of $\mathbf{X}_l$, draw $\mathbf{X}_{l,j*} \sim \text{Delta}(\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l))$.
2. For each item j,
   (a) Draw a clean input $\mathbf{X}_{c,j*} \sim \mathcal{N}(\mathbf{X}_{L,j*}, \lambda_n^{-1}\mathbf{I}_J)$.
   (b) Draw a latent item offset vector $\boldsymbol{\epsilon}_j \sim \mathcal{N}(\mathbf{0}, \lambda_v^{-1}\mathbf{I}_K)$ and then set the latent item vector to be: $\mathbf{v}_j = \boldsymbol{\epsilon}_j + \mathbf{X}^T_{\frac{L}{2},j*}$.
3. Draw a latent user vector for each user i: $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \lambda_u^{-1}\mathbf{I}_K)$.
4. Draw a rating r_{ij} for each user-item pair (i,j): $r_{ij} \sim \mathcal{N}(\mathbf{u}_i^T\mathbf{v}_j, c_{ij}^{-1})$.

Here λ_w, λ_n, λ_u, and λ_v are hyperparameters and c_{ij} is a confidence parameter similar to that for CTR. Note that the middle layer $\mathbf{X}_{L/2}$ serves as a bridge between the Bayesian SDAE and PMF. This

[1]Note that while generation of the *clean* input $\mathbf{X}_c$ from $\mathbf{X}_L$ is part of the generative process of the Bayesian SDAE, generation of the *noise-corrupted* input $\mathbf{X}_0$ from $\mathbf{X}_c$ is an artificial noise injection process to help the SDAE learn a more robust feature representation.

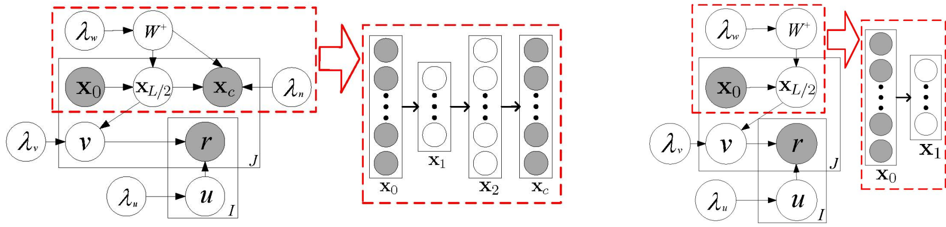


Figure 2: On the left is the graphical model of CDL. The part inside the dashed rectangle represents an SDAE. An example SDAE with $L = 2$ is shown. On the right is the graphical model of the degenerated CDL. The part inside the dashed rectangle represents the encoder of an SDAE. An example SDAE with $L = 2$ is shown on the right of it. Note that although L is still 2, the decoder of the SDAE vanishes. To prevent clutter, we omit all variables $\mathbf{x}_l$ except $\mathbf{x}_0$ and $\mathbf{x}_{L/2}$ in graphical models.

middle layer, along with the latent offset $\boldsymbol{\epsilon}_j$, is the key that enables CDL to simultaneously learn an effective feature representation and capture the similarity and (implicit) relationship between items (and users).

The graphical model of CDL is shown in Figure 2, where, for notational simplicity, we use $\mathbf{x}_0$, $\mathbf{x}_{L/2}$, and $\mathbf{x}_L$ in place of $\mathbf{X}_{0,j*}^T$, $\mathbf{X}_{\frac{L}{2},j*}^T$, and $\mathbf{X}_{L,j*}^T$, respectively.

4.3 Learning the Parameters

Based on the CDL model above, all parameters could be treated as random variables so that fully Bayesian methods such as Markov chain Monte Carlo (MCMC) or variational approximation methods [11] may be applied. However, such treatment typically incurs high computational cost. Besides, since CTR is our primary baseline for comparison, it would be fair and reasonable to take an approach analogous to that used in CTR. Consequently, we devise below an EM-style algorithm for obtaining the maximum a posteriori (MAP) estimates, as in [28].

Like in CTR, maximizing the posterior probability is equivalent to maximizing the joint log-likelihood of $\mathbf{U}$, $\mathbf{V}$, $\{\mathbf{X}_l\}$, $\mathbf{X}_c$, $\{\mathbf{W}_l\}$, $\{\mathbf{b}_l\}$, and $\mathbf{R}$ given λ_u, λ_v, λ_w, and λ_n:

$$\begin{aligned}\mathscr{L} = & -\frac{\lambda_u}{2}\sum_i \|\mathbf{u}_i\|_2^2 - \frac{\lambda_w}{2}\sum_l (\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2) - \frac{\lambda_v}{2}\sum_j \|\mathbf{v}_j - f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T\|_2^2 \\ & -\frac{\lambda_n}{2}\sum_j \|f_r(\mathbf{X}_{0,j*}, \mathbf{W}^+) - \mathbf{X}_{c,j*}\|_2^2 - \sum_{i,j}\frac{c_{ij}}{2}(r_{ij} - \mathbf{u}_i^T\mathbf{v}_j)^2,\end{aligned}$$

where the encoder function $f_e(\cdot, \mathbf{W}^+)$ takes the corrupted content vector $\mathbf{X}_{0,j*}$ of item j as input and computes the encoding of the item, and the function $f_r(\cdot, \mathbf{W}^+)$ also takes $\mathbf{X}_{0,j*}$ as input, computes the encoding and then the reconstructed content vector of item j. For example, if the number of layers $L = 6$, $f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)$ is the output of the third layer while $f_r(\mathbf{X}_{0,j*}, \mathbf{W}^+)$ is the output of the sixth layer.

From the perspective of optimization, the third term in the objective function above is equivalent to a multi-layer perceptron using the latent item vectors $\mathbf{v}_j$ as target while the fourth term is equivalent to an SDAE minimizing the reconstruction error.

Note that when the ratio λ_n/λ_v approaches positive infinity, it will degenerate to a two-step model in which the latent representation learned using SDAE is put directly into the CTR. Another extreme happens when λ_n/λ_v goes to zero where the decoder of the SDAE essentially vanishes. On the right of Figure 2 is the graphical model of the degenerated CDL when λ_n/λ_v goes to zero. As demonstrated in the experiments, the predictive performance will suffer greatly for both extreme cases.

For $\mathbf{u}_i$ and $\mathbf{v}_j$, coordinate ascent similar to [28, 10] is used. Given the current $\mathbf{W}^+$, we compute the gradients of $\mathscr{L}$ with respect to $\mathbf{u}_i$ and $\mathbf{v}_j$ and set them to zero, leading to the following update rules:

$$\mathbf{u}_i \leftarrow (\mathbf{V}\mathbf{C}_i\mathbf{V}^T + \lambda_u \mathbf{I}_K)^{-1}\mathbf{V}\mathbf{C}_i\mathbf{R}_i$$
$$\mathbf{v}_j \leftarrow (\mathbf{U}\mathbf{C}_i\mathbf{U}^T + \lambda_v \mathbf{I}_K)^{-1}(\mathbf{U}\mathbf{C}_j\mathbf{R}_j + \lambda_v f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T),$$

where $\mathbf{U} = (\mathbf{u}_i)_{i=1}^I$, $\mathbf{V} = (\mathbf{v}_j)_{j=1}^J$, $\mathbf{C}_i = \text{diag}(c_{i1}, \ldots, c_{iJ})$ is a diagonal matrix, $\mathbf{R}_i = (r_{i1}, \ldots, r_{iJ})^T$ is a column vector containing all the ratings of user i, and c_{ij}, as defined in Equation (1), reflects the confidence controlled by a and b as discussed in [10].

Given $\mathbf{U}$ and $\mathbf{V}$, we can learn the weights $\mathbf{W}_l$ and biases $\mathbf{b}_l$ for each layer using the back-propagation learning algorithm. By alternating the update of $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}_l$, and $\mathbf{b}_l$, we can find a local optimum for $\mathscr{L}$. Several commonly used techniques such as using a momentum term may be used to alleviate the local optimum problem.

4.4 Prediction

Let D be the observed test data. Similar to [28], we use the point estimates of $\mathbf{u}_i$, $\mathbf{W}^+$ and $\boldsymbol{\epsilon}_j$ to calculate the predicted rating:

$$E[r_{ij}|D] \approx E[\mathbf{u}_i|D]^T(E[f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T|D] + E[\boldsymbol{\epsilon}_j|D]),$$

where $E[\cdot]$ denotes the expectation operation. In other words, we approximate the predicted rating as:

$$r_{ij}^* \approx (\mathbf{u}_j^*)^T(f_e(\mathbf{X}_{0,j*}, \mathbf{W^+}^*)^T + \boldsymbol{\epsilon}_j^*) = (\mathbf{u}_i^*)^T\mathbf{v}_j^*.$$

Note that for any new item j with no rating in the training data, its offset $\boldsymbol{\epsilon}_j^*$ will be $\mathbf{0}$.

5 Experiments

As reported previously by other researchers [18, 29], with only item content information and the rating matrix, CTR [28] is currently the best tightly coupled model in terms of prediction accuracy. Due to space constraints, we mainly focus on comparing CDL with CTR in this paper.

5.1 Datasets

We use two real-world datasets from CiteULike[2] for our experiments. The two datasets, from [29], were collected in different ways, specifically, with different scales and different degrees of sparsity to mimic different practical situations. The first dataset, *citeulike-a*, is from [28]. The second dataset, *citeulike-t*, was collected independently of the first one. They manually selected 273 seed tags and collected all the articles with at least one of those tags. Similar to [28], users with fewer than 3 articles are not included. As a result, *citeulike-a* contains 5551 users and 16980 items. For *citeulike-t*, the numbers are 7947 and 25975. We can see that *citeulike-t* contains more users and items than *citeulike-a*. Also, *citeulike-t* is much sparser as only 0.07% of its user-item matrix entries contain ratings but *citeulike-a* has ratings in 0.22% of its user-item matrix entries.

We follow the same procedure as that in [28] to preprocess the text information (item content) extracted from the titles and abstracts of the articles. After removing stop words, the top S discriminative words according to the tf-idf values are chosen to form the vocabulary (S is 8000 and 20000 for the two datasets respectively).

5.2 Evaluation Scheme

For each dataset, similar to [29], we randomly select P items associated with each user to form the training set and use all the rest of the dataset as the test set. To evaluate and compare the models under both sparse and dense settings, we set P to 1 and 10, respectively, in our experiments. For

[2]CiteULike allows users to create their own collections of articles. There are abstract, title, and tags for each article. More details about the CiteULike data can be found at `http://www.citeulike.org`.

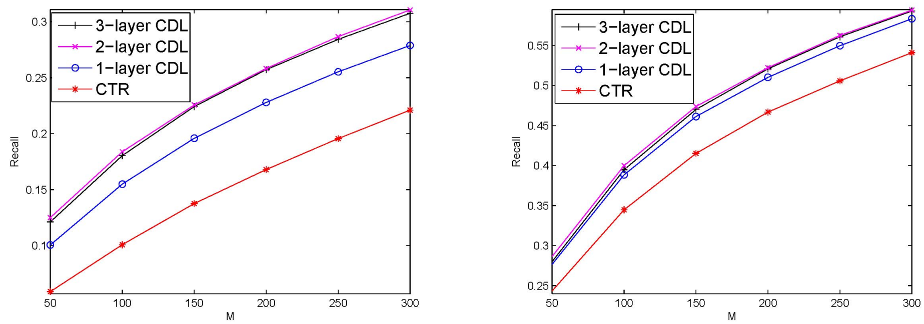


Figure 3: Performance comparison of CDL and CTR based on recall@M for *citeulike-a*. For the left figure $P = 1$ (sparse setting) and for the right one $P = 10$ (dense setting).

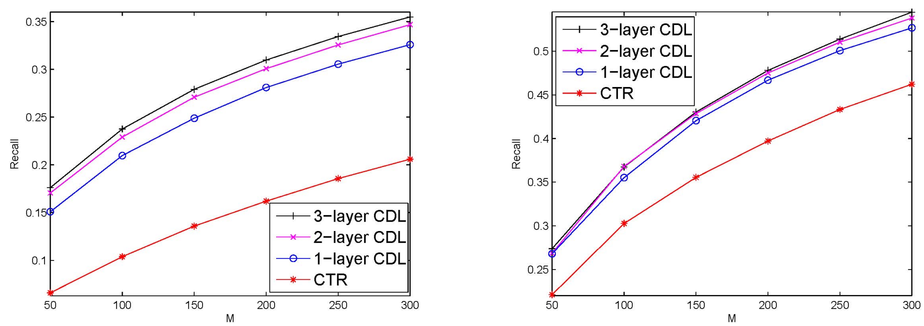


Figure 4: Performance comparison of CDL and CTR based on recall@M for *citeulike-t*. For the left figure $P = 1$ (sparse setting) and for the right one $P = 10$ (dense setting).

each value of P, we repeat the evaluation five times with different randomly selected training sets and the average performance is reported.

As in [28, 18, 29], we use recall as the performance measure because the rating information is in the form of implicit feedback [10, 19]. Specifically, a zero entry may be due to the fact that the user is not interested in the item, or that the user is not aware of its existence. As such, precision is not a suitable performance measure. Like most recommender systems, we sort the predicted ratings of the candidate items and recommend the top M items to the target user. The recall@M for each user is then defined as:

$$\text{recall@}M = \frac{\text{number of items that the user likes among the top } M}{\text{total number of items that the user likes}}.$$

The final result reported is the average recall over all users.

5.3 Experimental Settings

In the experiments, we first use a validation set to find the optimal hyperparameters for CTR. We find that CTR can achieve good prediction performance when $\lambda_u = 0.1$, $\lambda_v = 10$, $a = 1$, $b = 0.01$, and $K = 50$ (note that a and b determine the confidence parameters c_{ij}). For CDL, we directly set $a = 1$, $b = 0.01$, $K = 50$ and perform grid search on the hyperparameters λ_u, λ_v, λ_n, and λ_w. For the grid search, we split the training data and use 5-fold cross validation.

We use a masking noise with a noise level of 0.3 to get the corrupted input $\mathbf{X}_0$ from the clean input $\mathbf{X}_c$. For CDL with more than one layer of SDAE ($L > 2$), we use a dropout rate [2, 27, 9] of 0.1 to achieve adaptive regularization. In terms of network architecture, for the Bayesian SDAE of the CDL, the number of hidden units K_l is set to 200 for l such that $l \neq L/2$ and $0 < l < L$. While both K_0 and K_L are equal to the number of words S in the dictionary, $K_{L/2}$ is set to K which is the number of latent factors in PMF. For example, the 2-layer CDL model ($L = 4$) has a Bayesian SDAE of architecture '8000-200-50-200-8000' for the *citeulike-a* dataset.

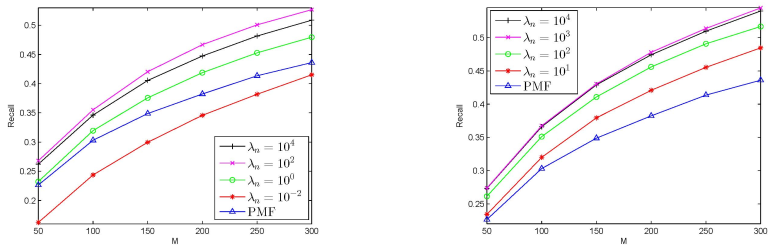


Figure 5: Performance of CDL based on recall@M for different values of λ_n on *citeulike-t*. The left plot is for $L = 2$ and the right one is for $L = 6$.

5.4 Quantitative Comparison

Figures 3 and 4 show the results that compare CDL and CTR using the two datasets under both the sparse ($P = 1$) and dense ($P = 10$) settings. We can see that even a 1-layer ($L = 2$) CDL consistently outperforms CTR by a large margin. For the *citeulike-a* dataset, 1-layer CDL outperforms CTR by a margin of 4.2%$\sim$6.0% in the sparse setting and 3.3%$\sim$4.6% in the dense setting. If we increase the number of layers to 3 ($L = 6$), the margin will go up to 5.8%$\sim$8.0% and 4.3%$\sim$5.8%, respectively. For the *citeulike-t* dataset, 1-layer CDL outperforms CTR by a margin of 8.5%$\sim$12.0% in the sparse setting and 4.7%$\sim$6.4% in the dense setting. When the number of layers is increased to 3, the margin will even go up to 11.0%$\sim$14.9% and 5.2%$\sim$8.2%, respectively. Since the standard deviation is always very small ($4.31 \times 10^{-5} \sim 9.31 \times 10^{-3}$), we do not include it in the figures as it is not noticeable anyway.

We note that the results are somewhat different for the two datasets. This is due to the different ways in which the datasets were collected, as discussed above. Specifically, the text information and rating matrix in the *citeulike-t* dataset are both much sparser.[3] By seamlessly integrating the Bayesian SDAE and CTR, CDL can handle both the sparse rating matrix and the sparse text information much better and learn a much more effective latent representation for each item and hence each user.

Figure 5 shows the results for different values of λ_n using *citeulike-t* under the dense setting. We set $\lambda_u = 0.01$, $\lambda_v = 100$, and L to 2 and 6. Similar phenomena are observed when the number of layers and the value of P are varied but they are omitted here due to space constraints. As mentioned in the previous section, when λ_n is extremely large, λ_n/λ_v will approach positive infinity so that CDL degenerates to two separate models. In this case the latent item representation will be learned by the SDAE in an unsupervised manner and then it will be put directly into (a simplified version of) the CTR. Consequently, there is no interaction between the Bayesian SDAE and the collaborative filtering component based on matrix factorization and hence the prediction performance will suffer greatly. For the other extreme when λ_n is extremely small, λ_n/λ_v will approach zero so that CDL degenerates to that in Figure 2 in which the decoder of the Bayesian SDAE component essentially vanishes. This way the encoder of the Bayesian SDAE component will easily overfit the latent item vectors learned by simple matrix factorization. As we can see in Figure 5, the prediction performance degrades significantly as λ_n gets very large or very small. When $\lambda_n < 0.1$, the recall@M is already very close to (or even worse than) the result of PMF.

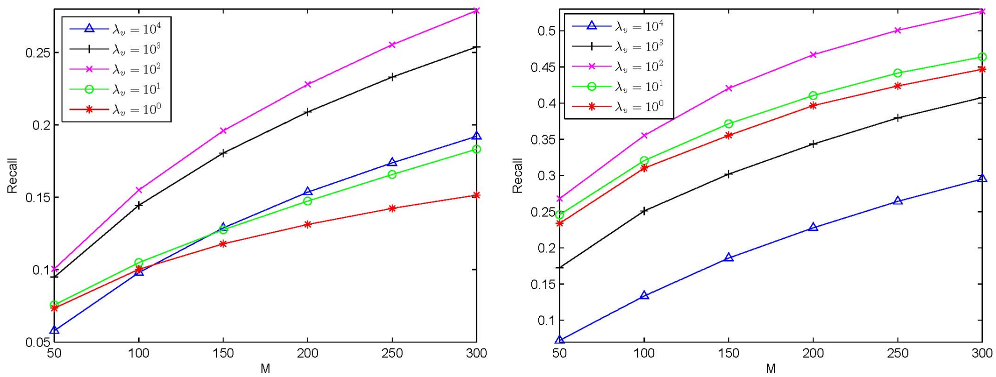
6 Parameter Sensitivity

In this section we investigate how the hyperparameters λ_u and λ_v affect the prediction performance. We first fix λ_u to 0.01 and vary λ_v from 1 to 10000 to see how the recall changes for the two datasets. We then fix λ_v to 100 and vary λ_u from 0.0001 to 1. As shown in Figures 6 and 7, the recall is relatively sensitive to λ_v but does not change very much as λ_u varies.

7 Qualitative Comparison

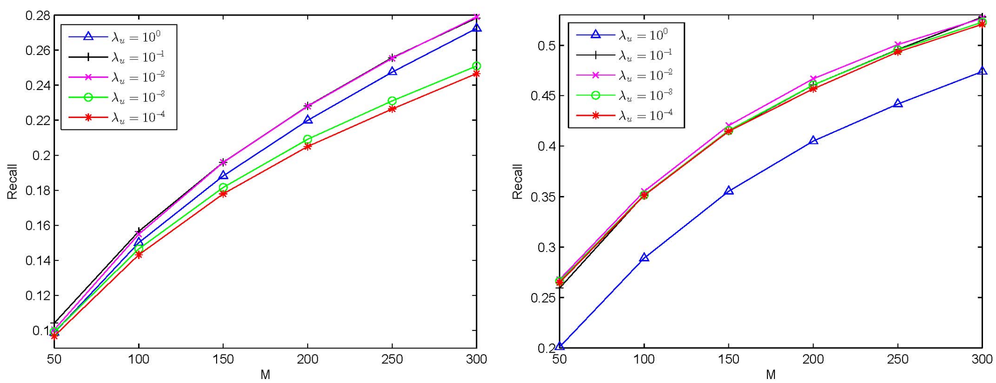
To gain a better insight into CDL, we take a look at two example users and represent the profile of each of them using the top three matched topics. We examine the top 10 recommended articles

[3] Each article in *citeulike-a* has 66.6 words on average and that for *citeulike-t* is 18.8.



(a) $L = 2$, $P = 1$, *citeulike-a* (b) $L = 2$, $P = 10$, *citeulike-t*

Figure 6: Performance of CDL when λ_v varies from 1 to 10000 with λ_u set to 0.01.



(a) $L = 2$, $P = 1$, *citeulike-a* (b) $L = 2$, $P = 10$, *citeulike-t*

Figure 7: Performance of CDL when λ_u varies from 0.0001 to 1 with λ_v set to 100.

returned by a 3-layer ($L = 6$) CDL and CTR. The *citeulike-t* dataset is used under the sparse setting ($P = 1$). From Table 1, we can speculate that user I might be a computer scientist with focus on tag recommendation, as clearly indicated by the first topic in CDL and the second one in CTR. CDL correctly recommends many articles on tagging systems while CTR focuses on social networks instead. When digging into the data, we find that the only rated article in the training data is 'What drives content tagging: the case of photos on Flickr', which is an article that talks about the impact of social networks on tagging behaviors. This may explain why CTR focuses its recommendation on social networks. On the other hand, CDL can better understand the key points of the article (i.e., tagging and CF) to make appropriate recommendation accordingly. Consequently, the precision of CDL and CTR is 70% and 10%, respectively.

From the matched topics returned by both CDL and CTR, user II might be a researcher on blood flow dynamic theory particularly in the field of medical science. CDL correctly captures the user profile and achieves a precision of 100%. However, CTR recommends quite a few articles on astronomy instead. When examining the data, we find that the only rated article returned by CTR is 'Simulating deformable particle suspensions using a coupled lattice-Boltzmann and finite-element method'. As expected, this article is on deformable particle suspension and the flow of blood cells. CTR might have misinterpreted this article, focusing its recommendation on words like 'flows' and 'formation' separately. This explains why CTR recommends articles like 'Formation versus destruction: the evolution of the star cluster population in galaxy mergers' (formation) and 'Macroscopic effects of the spectral structure in turbulent flows' (flows). As a result, its precision is only 30%.

Table 1: Interpretability of the latent structures learned

	user I (CDL)	in user's lib?
top 3 topics	1. search, image, query, images, queries, tagging, index, tags, searching, tag	
	2. social, online, internet, communities, sharing, networking, facebook, friends, ties, participation	
	3. collaborative, optimization, filtering, recommendation, contextual, planning, items, preferences	
top 10 articles	1. The structure of collaborative tagging Systems	yes
	2. Usage patterns of collaborative tagging systems	yes
	3. Folksonomy as a complex network	no
	4. HT06, tagging paper, taxonomy, Flickr, academic article, to read	yes
	5. Why do tagging systems work	yes
	6. Information retrieval in folksonomies: search and ranking	no
	7. tagging, communities, vocabulary, evolution	yes
	8. The complex dynamics of collaborative tagging	yes
	9. Improved annotation of the blogosphere via autotagging and hierarchical clustering	no
	10. Collaborative tagging as a tripartite network	yes
	user I (CTR)	in user's lib?
top 3 topics	1. social, online, internet, communities, sharing, networking, facebook, friends, ties, participation	
	2. search, image, query, images, queries, tagging, index, tags, searching, tag	
	3. feedback, event, transformation, wikipedia, indicators, vitamin, log, indirect, taxonomy	
top 10 articles	1. HT06, tagging paper, taxonomy, Flickr, academic article, to read	yes
	2. Structure and evolution of online social networks	no
	3. Group formation in large social networks: membership, growth, and evolution	no
	4. Measurement and analysis of online social networks	no
	5. A face(book) in the crowd: social searching vs. social browsing	no
	6. The strength of weak ties	no
	7. Flickr tag recommendation based on collective knowledge	no
	8. The computer-mediated communication network	no
	9. Social capital, self-esteem, and use of online social network sites: A longitudinal analysis	no
	10. Increasing participation in online communities: A framework for humanCcomputer interaction	no
	user II (CDL)	in user's lib?
top 3 topics	1. flow, cloud, codes, matter, boundary, lattice, particles, galaxies, fluid, galaxy	
	2. mobile, membrane, wireless, sensor, mobility, lipid, traffic, infrastructure, monitoring, ad	
	3. hybrid, orientation, stress, fluctuations, load, temperature, centrality, mechanical, two-dimensional, heat	
top 10 articles	1. Modeling the flow of dense suspensions of deformable particles in three dimensions	yes
	2. Simplified particulate model for coarse-grained hemodynamics simulations	yes
	3. Lattice Boltzmann simulations of blood flow: non-newtonian rheology and clotting processes	yes
	4. A genome-wide association study for celiac disease identifies risk variants	yes
	5. Efficient and accurate simulations of deformable particles	yes
	6. A multiscale model of thrombus development	yes
	7. Multiphase hemodynamic simulation of pulsatile flow in a coronary artery	yes
	8. Lattice Boltzmann modeling of thrombosis in giant aneurysms	yes
	9. A lattice Boltzmann simulation of clotting in stented aneursysms	yes
	10. Predicting dynamics and rheology of blood flow	yes
	user II (CTR)	in user's lib?
top 3 topics	1. flow, cloud, codes, matter, boundary, lattice, particles, galaxies, fluid, galaxy	
	2. transition, equations, dynamical, discrete, equation, dimensions, chaos, transitions, living, trust	
	3. mobile, membrane, wireless, sensor, mobility, lipid, traffic, infrastructure, monitoring, ad	
top 10 articles	1. Multiphase hemodynamic simulation of pulsatile flow in a coronary artery	yes
	2. The metallicity evolution of star-forming galaxies from redshift 0 to 3	no
	3. Formation versus destruction: the evolution of the star cluster population in galaxy mergers	no
	4. Clearing the gas from globular clusters	no
	5. Macroscopic effects of the spectral structure in turbulent flows	no
	6. The WiggleZ dark energy survey	no
	7. Lattice-Boltzmann simulation of blood flow in digitized vessel networks	no
	8. Global properties of 'ordinary' early-type galaxies	no
	9. Proteus : a direct forcing method in the simulations of particulate flows	yes
	10. Analysis of mechanisms for platelet near-wall excess under arterial blood flow conditions	yes

From these two users, we can see that with a more effective representation, CDL can capture the key points of articles and the user preferences more accurately (e.g., user I). Besides, it can model the co-occurrence and relations of words better (e.g., user II).

8 Conclusion and Future Work

We have demonstrated in this paper that state-of-the-art performance can be achieved by tightly integrating the Bayesian SDAE and PMF. As far as we know, CDL is the first tightly coupled method that bridges the gap between deep learning models and hierarchical Bayesian models for recommendation applications.

Among the possible extensions that could be made to CDL, the bag-of-words representation may be replaced by more powerful alternatives, such as [17]. Besides, as remarked above, CDL actually provides a framework that can also admit deep learning models other than SDAE. One promising choice is the convolutional neural network model which, among other things, can explicitly take the context and order of words into account. Further performance boost may be possible.

References

[1] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *KDD*, pages 19–28, 2009.

[2] P. Baldi and P. J. Sadowski. Understanding dropout. In *NIPS*, pages 2814–2822, 2013.

[3] Y. Bengio, L. Yao, G. Alain, and P. Vincent. Generalized denoising auto-encoders as generative models. In *NIPS*, pages 899–907, 2013.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[6] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowledge Based Systems*, 46:109–132, 2013.

[7] M. Chen, Z. E. Xu, K. Q. Weinberger, and F. Sha. Marginalized denoising autoencoders for domain adaptation. In *ICML*, 2012.

[8] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pages 369–376, 2006.

[9] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[10] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, 2008.

[11] M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[12] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. *ACL*, 2014.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

[14] K. Lang. Newsweeder: Learning to filter netnews. In *ICML*, 1995.

[15] W.-J. Li, D.-Y. Yeung, and Z. Zhang. Generalized latent factor models for social network analysis. In *IJCAI*, pages 1705–1710, 2011.

[16] D. J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[18] S. Purushotham, Y. Liu, and C.-C. J. Kuo. Collaborative topic regression with social matrix factorization for recommendation systems. In *ICML*, 2012.

[19] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[20] A. Said. Identifying and utilizing contextual data in hybrid recommender systems. In *RecSys*, 2010.

[21] R. Salakhutdinov and G. E. Hinton. Deep Boltzmann machines. In *AISTATS*, pages 448–455, 2009.

[22] R. Salakhutdinov and G. E. Hinton. Semantic hashing. *Int. J. Approx. Reasoning*, 50(7):969–978, 2009.

[23] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, 2007.

[24] S. G. Sevil, O. Kucuktunc, P. Duygulu, and F. Can. Automatic tag expansion using visual similarity for photo sharing websites. *Multimedia Tools Appl.*, 49(1):81–99, 2010.

[25] R. S. Strichartz. *A Guide to Distribution Theory and Fourier Transforms*. World Scientific, 2003.

[26] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 11, 2010.

[27] S. Wager, S. Wang, and P. Liang. Dropout training as adaptive regularization. In *NIPS*, pages 351–359, 2013.

[28] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *KDD*, 2011.

[29] H. Wang, B. Chen, and W.-J. Li. Collaborative topic regression with social regularization for tag recommendation. In *IJCAI*, 2013.

[30] N. Wang and D.-Y. Yeung. Learning a deep compact image representation for visual tracking. In *NIPS*, pages 809–817, 2013.