Relational Stacked Denoising Autoencoder for Tag Recommendation

Hao Wang, Xingjian Shi, 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, xshiab@connect.ust.hk, dyyeung@cse.ust.hk

Abstract

Tag recommendation has become one of the most important ways of organizing and indexing online resources like articles, movies, and music. Since tagging information is usually very sparse, effective learning of the content representation for these resources is crucial to accurate tag recommendation. Recently, models proposed for tag recommendation, such as *collaborative topic regression* and its variants, have demonstrated promising accuracy. However, a limitation of these models is that, by using topic models like *latent Dirichlet allocation* as the key component, the learned representation may not be compact and effective enough. Moreover, since relational data exist as an auxiliary data source in many applications, it is desirable to incorporate such data into tag recommendation models. In this paper, we start with a deep learning model called *stacked denoising autoencoder* (SDAE) in an attempt to learn more effective content representation. We propose a probabilistic formulation for SDAE and then extend it to a *relational SDAE* (RSDAE) model. RSDAE jointly performs deep representation learning and relational learning in a principled way under a probabilistic framework. Experiments conducted on three real datasets show that both learning more effective representation and learning from relational data are beneficial steps to take to advance the state of the art.

Introduction

Due to the abundance of online resources like articles, movies, and music, tagging systems (Yu et al. 2014) have become increasingly important for organizing and indexing them. For example, CiteULike[1] uses tags to help categorize millions of articles online and Flickr[2] allows users to use tags to organize their photos. However, it is often not easy to compose a set of words appropriate for the resources. Besides, the large variety in phrasing styles of the users can potentially make the tagging information inconsistent and idiosyncratic. With such technical challenges, research in tag recommendation (TR) (Gupta et al. 2010; Wang et al. 2012) has gained in popularity over the past few years. An accurate tag recommendation system not only can save the pain of users searching for candidate tags on the tip of their tongues, but can also make the tags used more consistent. Consequently, both the user experience and recommendation accuracy can be improved dramatically.

Tag recommendation methods can roughly be categorized into three classes (Wang et al. 2012): content-based methods, co-occurrence based methods, and hybrid methods. Content-based methods (Chen et al. 2008; 2010; Shen and Fan 2010) utilize only the content information (e.g., abstracts of articles, image pixels, and music content) for tag recommendation. Co-occurrence based methods (Garg and Weber 2008; Weinberger, Slaney, and van Zwol 2008; Rendle and Schmidt-Thieme 2010) are similar to *collaborative filtering* (CF) methods (Li and Yeung 2011). The co-occurrence of tags among items, usually represented as an tag-item matrix, is used for tagging. The third class of methods (Wu et al. 2009; Wang and Blei 2011; Yang, Zhang, and Wang 2013; Zhao et al. 2013; Bao, Fang, and Zhang 2014; Chen et al. 2014), also the most popular and effective ones, consists of hybrid methods. They make use of both tagging (co-occurrence) information (the tag-item matrix) and item content information for recommendation.

In hybrid methods, learning of item representations (also called item latent factors in some models) is crucial for the recommendation accuracy especially when the tag-item matrix is extremely sparse. Recently, models like *collaborative topic regression* (CTR) (Wang and Blei 2011) and its variants (Purushotham, Liu, and Kuo 2012; Wang, Chen, and Li 2013) have been proposed and adapted for tag recommendation to achieve promising performance. These models use *latent Dirichlet allocation* (LDA) (Blei, Ng, and Jordan 2003) as the key component for learning item representations and use *probabilistic matrix factorization* (PMF) (Salakhutdinov and Mnih 2007) to process the co-occurrence matrix (tag-item matrix). However, when using LDA, the resulting item representations tend to be quite sparse. Consequently, more dimensions may be needed for the representations to be effective. Unfortunately PMF with the low-rank assumption usually works with quite a small number of latent dimensions, which is not in line with the nature of LDA (or CTR). On the other hand, deep learning models like *stacked denoising autoencoder* (SDAE) (Vincent et

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

[2] http://www.flickr.com

al. 2010) and convolutional neural networks (Krizhevsky, Sutskever, and Hinton 2012) recently show great potential for learning effective and compact representations and deliver state-of-the-art performance in computer vision (Wang and Yeung 2013) and natural language processing (Salakhutdinov and Hinton 2009; Kalchbrenner, Grefenstette, and Blunsom 2014) applications. Intuitively, the effectiveness and compactness of deep learning models like SDAE seem to fit PMF perfectly and can potentially lead to significant boost of recommendation performance. Besides, since relational data exist as an auxiliary data source in many applications (e.g., natural language processing, computational biology), it is desirable to incorporate such data into tag recommendation models. For example, when recommending tags for articles in CiteULike, the citation relations between articles (Vu et al. 2011; Wang and Li 2013) may provide very useful information. However, incorporating relational information into deep neural network (DNN) models like SDAE is non-trivial since with the relational data, the samples are no longer i.i.d., which is the assumption underlying DNN models.

In this paper, we propose novel methods to address the above challenges. The main contributions of this paper are summarized as follows:

- We adapt SDAE and use it in conjunction with PMF (or a simplified version of CTR) to significantly boost the recommendation performance.
- To satisfy the need for relational deep learning, we develop a probabilistic formulation for SDAE and, by extending this probabilistic SDAE, we propose a probabilistic relational model called *relational SDAE* (RSDAE) to integrate deep representation learning and relational learning in a principled way. Besides, RSDAE can be naturally extended to handle multi-relational data (with more details provided in the supplementary material).
- Extensive experiments on datasets from different domains show that our models outperform the state of the art.

Problem Statement and Notation

Assume we have a set of items (articles or movies) $\mathbf{X}_c$ to be tagged, with $\mathbf{X}_{c,j*}^T \in \mathbb{R}^B$ denoting the content (attributes) of item j. In the case of tagging articles (papers) in CiteULike, the items are papers, and the content information can be the bag-of-words representation of paper abstracts. Assume we have a set of I tags $\{\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_I\}$ as candidates to be recommended to tag each item. Then a tag-item matrix $\mathbf{R}$ can be used to represent the tagging information for all the items. Each matrix entry $\mathbf{R}_{ij}$ is a binary variable, where $\mathbf{R}_{ij} = 1$ means that tag $\mathbf{t}_i$ is associated with item j and $\mathbf{R}_{ij} = 0$ otherwise. Tag recommendation is to predict the missing values in $\mathbf{R}_{*j} = [\mathbf{R}_{1j}, \mathbf{R}_{2j}, \cdots, \mathbf{R}_{Ij}]^T$ (i.e., recommend tags to items). Besides, we use $\mathbf{I}_K$ to denote a K-dimensional identity matrix and $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_J]$ to denote the *relational latent matrix* with $\mathbf{s}_j$ representing the relational properties of item j. Note that although we focus on tag recommendation for articles and movies in this paper, our proposed models are flexible enough to be used for other applications such as image and video tagging.

From the perspective of SDAE, the J-by-B matrix $\mathbf{X}_c$ represents the clean input to the SDAE and the noise-corrupted matrix of the same size is denoted by $\mathbf{X}_0$. Besides, we denote the output of layer l of the SDAE, a J-by-K_l matrix, by $\mathbf{X}_l$. 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 of layer l, $\mathbf{W}_{l,*n}$ denotes column n of $\mathbf{W}_l$, and L is the number of layers. As a shorthand, we refer to the collection of all layers of weight matrices and biases as $\mathbf{W}^+$. Note that in our models an $L/2$-layer SDAE corresponds to an L-layer network.

Probabilistic Stacked Denoising Autoencoders

In this section we will first have a brief review on SDAE and then give a probabilistic formulation of generalized SDAE, which will be a building block of our relational stacked denoising autoencoder (RSDAE) model.

Stacked Denoising Autoencoders

SDAE (Vincent et al. 2010) is essentially a feedforward neural network for learning representations of the input data by learning to predict the clean input itself in the output. Normally the hidden layer in the middle is constrained to be a narrow bottleneck and the input layer $\mathbf{X}_0$ is a noise-corrupted version of the clean input data. Learning of an SDAE involves solving the following regularized optimization problem:

$$\min_{\{\mathbf{W}_l\},\{\mathbf{b}_l\}} \|\mathbf{X}_c - \mathbf{X}_L\|_F^2 + \lambda \sum_l (\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2),$$

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

Probabilistic Stacked Denoising Autoencoders

If we treat both the clean input (bag-of-words) $\mathbf{X}_c$ and the corrupted input $\mathbf{X}_0$ as observed variables, the generative process for the generalized probabilistic SDAE is as follows:

1. For each layer l of the SDAE network,
   (a) For each column n of the weight matrix $\mathbf{W}_l$, draw $\mathbf{W}_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{K_l})$.
   (b) Draw the bias vector $\mathbf{b}_l \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{K_l})$.
   (c) For each row j of $\mathbf{X}_l$, draw
   $$\mathbf{X}_{l,j*} \sim \mathcal{N}(\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l), \lambda_s^{-1}\mathbf{I}_{K_l}).$$
2. For each item j, draw a clean input[3]
   $$\mathbf{X}_{c,j*} \sim \mathcal{N}(\mathbf{X}_{L,j*}, \lambda_n^{-1}\mathbf{I}_B).$$

[3] Note that while generation of the *clean* input $\mathbf{X}_c$ from $\mathbf{X}_L$ is part of the generative process of the probabilistic 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.

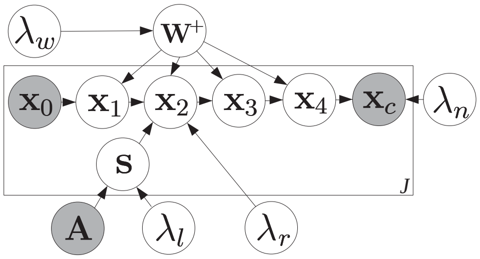

Figure 1: Graphical model of RSDAE for $L = 4$. λ_s is not shown here to prevent clutter.

Here, λ_w, λ_s, and λ_n are hyperparameters and $\sigma(\cdot)$ is the sigmoid function.

Following the generative process above, maximizing the posterior probability is equivalent to maximizing the joint log-likelihood of $\{\mathbf{X}_l\}$, $\mathbf{X}_c$, $\{\mathbf{W}_l\}$, and $\{\mathbf{b}_l\}$ given λ_s, λ_w, and λ_n:

$$\begin{aligned}\mathscr{L} = &-\frac{\lambda_w}{2}\sum_l(\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2)\\ &-\frac{\lambda_n}{2}\sum_j \|\mathbf{X}_{L,j*} - \mathbf{X}_{c,j*}\|_2^2\\ &-\frac{\lambda_s}{2}\sum_l\sum_j \|\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l) - \mathbf{X}_{l,j*}\|_2^2.\end{aligned}$$

Note that as the hyperparameter λ_s approaches infinity, the last term will disappear and the model will degenerate to the original SDAE where $\mathbf{X}_{l,j*} = \sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l)$. That is why we call it generalized SDAE.

Relational Stacked Denoising Autoencoders

We will now use the probabilistic SDAE described above as a building block to formulate the RSDAE model.

Model Formulation

We formulate RSDAE as a novel probabilistic model which can seamlessly integrate layered representation learning and the relational information available. This way our model can learn simultaneously the feature representation from the content information and the relation between items. The graphical model of RSDAE is shown in Figure 1 and the generative process is listed as follows:

1. Draw the relational latent matrix $\mathbf{S}$ from a *matrix variate normal distribution* (Gupta and Nagar 2000):

$$\mathbf{S} \sim \mathcal{N}_{K,J}(0, \mathbf{I}_K \otimes (\lambda_l \mathscr{L}_a)^{-1}). \tag{1}$$

2. For layer l of the SDAE where $l = 1, 2, \ldots, \frac{L}{2} - 1$,
   (a) For each column n of the weight matrix $\mathbf{W}_l$, draw $\mathbf{W}_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{K_l})$.
   (b) Draw the bias vector $\mathbf{b}_l \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{K_l})$.
   (c) For each row j of $\mathbf{X}_l$, draw

$$\mathbf{X}_{l,j*} \sim \mathcal{N}(\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l), \lambda_s^{-1}\mathbf{I}_{K_l}).$$

3. For layer $\frac{L}{2}$ of the SDAE network, draw the representation vector for item j from the product of two Gaussians (PoG) (Gales and Airey 2006):

$$\mathbf{X}_{\frac{L}{2},j*} \sim \mathrm{PoG}(\sigma(\mathbf{X}_{\frac{L}{2}-1,j*}\mathbf{W}_l + \mathbf{b}_l), \mathbf{s}_j^T, \lambda_s^{-1}\mathbf{I}_K, \lambda_r^{-1}\mathbf{I}_K).$$

4. For layer l of the SDAE network where $l = \frac{L}{2}+1, \frac{L}{2}+2, \ldots, L$,
   (a) For each column n of the weight matrix $\mathbf{W}_l$, draw $\mathbf{W}_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{K_l})$.
   (b) Draw the bias vector $\mathbf{b}_l \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{K_l})$.
   (c) For each row j of $\mathbf{X}_l$, draw

$$\mathbf{X}_{l,j*} \sim \mathcal{N}(\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l), \lambda_s^{-1}\mathbf{I}_{K_l}).$$

5. For each item j, draw a clean input

$$\mathbf{X}_{c,j*} \sim \mathcal{N}(\mathbf{X}_{L,j*}, \lambda_n^{-1}\mathbf{I}_B).$$

Here $K = K_{\frac{L}{2}}$ is the dimensionality of the learned representation vector for each item, $\mathbf{S}$ denotes the $K \times J$ relational latent matrix in which column j is the *relational latent vector* $\mathbf{s}_j$ for item j. Note that $\mathcal{N}_{K,J}(0, \mathbf{I}_K \otimes (\lambda_l \mathscr{L}_a)^{-1})$ in (1) is a matrix variate normal distribution defined as (Gupta and Nagar 2000):

$$\begin{aligned}p(\mathbf{S}) &= \mathcal{N}_{K,J}(0, \mathbf{I}_K \otimes (\lambda_l \mathscr{L}_a)^{-1})\\ &= \frac{\exp\{\mathrm{tr}[-\frac{\lambda_l}{2}\mathbf{S}\mathscr{L}_a\mathbf{S}^T]\}}{(2\pi)^{JK/2}|\mathbf{I}_K|^{J/2}|\lambda_l\mathscr{L}_a|^{-K/2}},\end{aligned} \tag{2}$$

where the operator $\otimes$ denotes the Kronecker product of two matrices (Gupta and Nagar 2000), $\mathrm{tr}(\cdot)$ denotes the trace of a matrix, and $\mathscr{L}_a$ is the Laplacian matrix incorporating the relational information. $\mathscr{L}_a = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is a diagonal matrix whose diagonal elements $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$ and $\mathbf{A}$ is the adjacency matrix representing the relational information with binary entries indicating the links (or relations) between items. $\mathbf{A}_{jj'} = 1$ indicates that there is a link between item j and item j' and $\mathbf{A}_{jj'} = 0$ otherwise. $\mathrm{PoG}(\sigma(\mathbf{X}_{\frac{L}{2}-1,j*}\mathbf{W}_l + \mathbf{b}_l), \mathbf{s}_j^T, \lambda_s^{-1}\mathbf{I}_K, \lambda_r^{-1}\mathbf{I}_K)$ denotes the product of the Gaussian $\mathcal{N}(\sigma(\mathbf{X}_{\frac{L}{2}-1,j*}\mathbf{W}_l + \mathbf{b}_l), \lambda_s^{-1}\mathbf{I}_K)$ and the Gaussian $\mathcal{N}(\mathbf{s}_j^T, \lambda_r^{-1}\mathbf{I}_K)$, which is also a Gaussian (Gales and Airey 2006).

According to the generative process above, maximizing the posterior probability is equivalent to maximizing the joint log-likelihood of $\{\mathbf{X}_l\}$, $\mathbf{X}_c$, $\mathbf{S}$, $\{\mathbf{W}_l\}$, and $\{\mathbf{b}_l\}$ given λ_s, λ_w, λ_l, λ_r, and λ_n:

$$\begin{aligned}\mathscr{L} = &-\frac{\lambda_l}{2}\mathrm{tr}(\mathbf{S}\mathscr{L}_a\mathbf{S}^T) - \frac{\lambda_r}{2}\sum_j \|(\mathbf{s}_j^T - \mathbf{X}_{\frac{L}{2},j*})\|_2^2\\ &-\frac{\lambda_w}{2}\sum_l(\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2)\\ &-\frac{\lambda_n}{2}\sum_j \|\mathbf{X}_{L,j*} - \mathbf{X}_{c,j*}\|_2^2\\ &-\frac{\lambda_s}{2}\sum_l\sum_j \|\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l) - \mathbf{X}_{l,j*}\|_2^2,\end{aligned} \tag{3}$$

where $\mathbf{X}_{l,j*} = \sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l)$. Similar to the generalized SDAE, taking λ_s to infinity, the last term of the joint log-likelihood will vanish. Note that the first term $-\frac{\lambda_l}{2}\operatorname{tr}(\mathbf{S}\mathscr{L}_a\mathbf{S}^T)$ corresponds to $\log p(\mathbf{S})$ in the matrix variate distribution in Equation (2). Besides, by simple manipulation (details can be found in the supplementary material), we have $\operatorname{tr}(\mathbf{S}\mathscr{L}_a\mathbf{S}^T) = \sum_{k=1}^{K}\mathbf{S}_{k*}^T\mathscr{L}_a\mathbf{S}_{k*}$ where $\mathbf{S}_{k*}$ denotes the kth row of $\mathbf{S}$. As we can see, maximizing $-\frac{\lambda_l}{2}\operatorname{tr}(\mathbf{S}^T\mathscr{L}_a\mathbf{S})$ is equivalent to making $\mathbf{s}_j$ closer to $\mathbf{s}_{j'}$ if item j and item j' are linked (namely $\mathbf{A}_{jj'} = 1$).

Learning Relational Representation

We now derive an EM-style algorithm for maximum a posteriori (MAP) estimation.

In terms of the relational latent matrix $\mathbf{S}$, we first fix all rows of $\mathbf{S}$ except the kth one $\mathbf{S}_{k*}$ and then update $\mathbf{S}_{k*}$. Specifically, we take the gradient of $\mathscr{L}$ with respect to $\mathbf{S}_{k*}$, set it to 0, and get the following linear system:

$$(\lambda_l\mathscr{L}_a + \lambda_r\mathbf{I})\mathbf{S}_{k*} = \lambda_r\mathbf{X}^T_{\frac{L}{2},*k}. \tag{4}$$

A naive approach is to solve the linear system by setting $\mathbf{S}_{k*} = \lambda_r(\lambda_l\mathscr{L}_a + \lambda_r\mathbf{I}_J)^{-1}\mathbf{X}^T_{\frac{L}{2},*k}$. Unfortunately, the complexity is $O(J^3)$ for one single update. Similar to (Li and Yeung 2009), the steepest descent method (Shewchuk 1994) is used to iteratively update $\mathbf{S}_{k*}$:

$$\begin{aligned}
\mathbf{S}_{k*}(t+1) &\leftarrow \mathbf{S}_{k*}(t) + \delta(t)r(t)\\
r(t) &\leftarrow \lambda_r\mathbf{X}^T_{\frac{L}{2},*k} - (\lambda_l\mathscr{L}_a + \lambda_r\mathbf{I}_J)\mathbf{S}_{k*}(t)\\
\delta(t) &\leftarrow \frac{r(t)^Tr(t)}{r(t)^T(\lambda_l\mathscr{L}_a + \lambda_r\mathbf{I}_J)r(t)}.
\end{aligned}$$

As discussed in (Li and Yeung 2009), the use of steepest descent method dramatically reduces the computation cost in each iteration from $O(J^3)$ to $O(J)$.

Given $\mathbf{S}$, we can learn $\mathbf{W}_l$ and $\mathbf{b}_l$ for each layer using the back-propagation algorithm. By alternating the update of $\mathbf{S}$, $\mathbf{W}_l$, and $\mathbf{b}_l$, a local optimum for $\mathscr{L}$ can be found. Also, techniques such as including a momentum term may help to avoid being trapped in a local optimum.

Tag Recommendation

After the representation for each item is learned, we can use a simplified version of CTR (Wang and Blei 2011) to learn the latent vectors $\mathbf{u}_i$ for tag i and $\mathbf{v}_j$ for item j. Similar to (Wang and Blei 2011), predicted ratings $\mathbf{R}_{ij}$ can be computed as the inner product of $\mathbf{u}_i$ and $\mathbf{v}_j$. Essentially we will be maximizing the following objective function:

$$\begin{aligned}
\mathscr{L} = &-\frac{\lambda_u}{2}\sum_i\|\mathbf{u}_i\|_2^2 - \frac{\lambda_v}{2}\sum_j\|\mathbf{v}_j - \mathbf{X}^T_{\frac{L}{2},j*}\|_2^2\\
&-\sum_{i,j}\frac{c_{ij}}{2}(\mathbf{R}_{ij} - \mathbf{u}_i^T\mathbf{v}_j)^2,
\end{aligned}$$

where λ_u and λ_v are hyperparameters. c_{ij} is set to 1 for the existing ratings and 0.01 for the missing entries.

Experiments

Datasets

For our experiments, we use three real-world datasets with two (Wang and Blei 2011; Wang, Chen, and Li 2013) from CiteULike[4] and one from MovieLens[5]. There are 7386 tags, 16980 articles (items), and 204987 tag-item pairs in the first dataset, *citeulike-a*. For the second one, *citeulike-t*, the numbers are 8311, 25975, and 134860. The third dataset, *movielens-plot*, originally from *MovieLens-10M* and enriched by us, contains 2988 tags, 7261 movies (items), and 51301 tag-item pairs.

The text information (item content) extracted from the titles and abstracts of the articles and from the plots of movies has been preprocessed using the same procedure as that in (Wang and Blei 2011). The sizes of the vocabulary are 8000, 20000, and 20000 for the three datasets respectively.

Regarding the relational information, we use the citation networks for *citeulike-a* and *citeulike-t*. For *movielens-plot* we have two types of relational information (two graphs): co-staff graph and co-genre graph. Existence of an edge in the co-staff graph means that the two connected movies share more than one staff member and an edge in the co-genre graph means that the two movies have identical genre combination. The numbers of edges in the citation networks are 44709 and 32665 for *citeulike-a* and *citeulike-t*, respectively. For the co-staff graph in *movielens-plot* there are 118126 edges in total and that number is 425495 for the co-genre graph. Note that our RSDAE model can support multi-relational data (like *movielens-plot*), though we present the uni-relational setting in the previous section for simplicity. Details of the full multi-relational SDAE can be found in the supplementary material.

Evaluation Scheme

In each dataset, similar to (Wang, Chen, and Li 2013), P items associated with each tag are randomly selected to form the training set and all the rest of the dataset is used as the test set. P is set to 1 and 10, respectively, to evaluate and compare the models under both sparse and dense settings in the experiments. For each value of P, the evaluation is repeated five times with different randomly selected training sets and the average performance is reported.

Following (Wang and Blei 2011; Purushotham, Liu, and Kuo 2012; Wang, Chen, and Li 2013), we use recall as the performance measure since the rating information appears in the form of implicit feedback (Hu, Koren, and Volinsky 2008; Rendle et al. 2009), which means a zero entry may be due to irrelevance between the tag and the item or the user's ignorance of the tags when tagging items. As such, precision is not suitable as a performance measure. Like most recommender systems, we sort the predicted ratings of the candidate tags and recommend the top M tags to the target item. The recall@M for each item is defined as:

$$\text{recall@}M = \frac{\text{number of tags the item is associated with in top } M}{\text{total number of tags the item is associated with}}.$$

[4] CiteULike allows users to create their own collections of articles. There are abstract, title, and tags for each article.

[5] http://www.grouplens.org/datasets

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

Experimental Settings

Experiments in (Wang, Chen, and Li 2013) have demonstrated that CTR and CTR-SR clearly outperform state-of-the-art content-based methods, co-occurrence based methods, and other hybrid methods. Due to space constraints, in the experiments we use only CTR (Wang and Blei 2011) and CTR-SR (Wang, Chen, and Li 2013) as baselines. CTR is a model combining LDA and PMF for recommendation. CTR-SR is a powerful extension of CTR in a sense that it seamlessly incorporates relational data into the model. We fix $K = 50$ and use a validation set to find the optimal hyperparameters for CTR and CTR-SR. For SDAE and RSDAE, tag recommendation can be divided into two steps: learning relational representation and PMF. We set λ_s to infinity for efficient computation and fair comparison with SDAE. Furthermore, since there are only four terms left in Equation (3) after the last term vanishes, we can directly fix $\lambda_r = 1$. The remaining hyperparameters of the first step (λ_l, λ_w, and λ_n) are found by grid search (λ_w is the hyperparameter for weight decay and can be ignored if we choose not to use it) and hyperparameters of the second step are fixed to values the same as those of CTR. For the grid search, we split the training data and 5-fold cross validation is used.

On the SDAE side, a masking noise with a noise level of 0.3 is added to the clean input $\mathbf{X}_c$ to obtain the corrupted input $\mathbf{X}_0$. We use a fixed dropout rate of 0.1 (Hinton et al. 2012; Wager, Wang, and Liang 2013) to achieve adaptive regularization. For the network architecture, we set the number of non-bottleneck hidden units K_l to 200. K_0 and K_L are set to B, the number of words in the dictionary. $K_{L/2}$ is equal to K, the number of latent factors in PMF. For example, a 2-layer SDAE has an architecture of '20000-200-50-200-20000' for the dataset *movielens-plot*.

Performance Evaluation

Figures 2, 3, and 4 show the recall@M for all three datasets in the sparse and dense settings, with M ranging from 50 to 300. As we can see, CTR-SR outperforms CTR by incorporating relational data into the model. What is surprising is that even without using any relational information, SDAE in conjunction with PMF still outperforms CTR-SR which utilizes abundant relational information, especially for *citeulike-t* as shown in Figure 3. Furthermore, RSDAE can achieve even higher recall by jointly performing representation learning and relational learning in a principled way.

Figure 5(left) shows the recall@M of RSDAE for *citeulike-t* in the sparse setting when $L = 2, 4, 6$ (corresponding to 1-layer, 2-layer, and 3-layer RSDAE, respectively). As we can see, the recall increases with the number of layers. Similar phenomena can be observed for other datasets which are omitted here due to space constraints. Note that the standard deviations are negligible in all experiments (from 4.56×10^{-5} to 3.57×10^{-3}). To prevent clutter, the standard deviations are not separately reported for all figures in this paper.

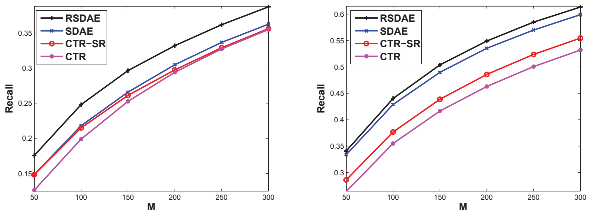

Figure 2: Performance comparison of all methods based on recall@M for *citeulike-a* when $P = 1$ (left) and $P = 10$ (right).

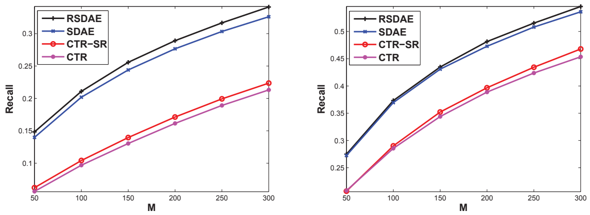

Figure 3: Performance comparison of all methods based on recall@M for *citeulike-t* when $P = 1$ (left) and $P = 10$ (right).

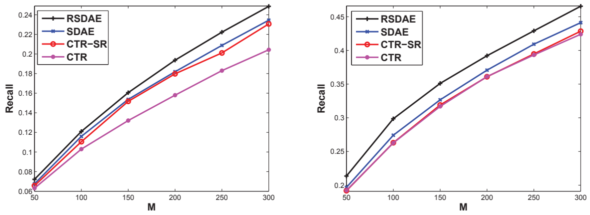

Figure 4: Performance comparison of all methods based on recall@M for *movielens-plot* when $P = 1$ (left) and $P = 10$ (right).

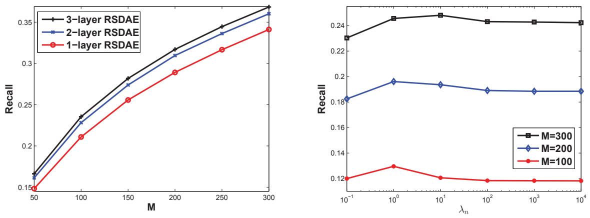

Figure 5: The effect of the number of layers in RSDAE (left) and the effect of λ_n in RSDAE (right).

Sensitivity to Hyperparameters

Figure 5(right) shows how recall@M is affected by the choice of hyperparameter λ_n for *movielens-plot* in the sparse setting when $\lambda_r = 1$ and $\lambda_l = 100$. As shown in the figure, recall@M increases with λ_n initially and gradually decreases at some point after $\lambda_n = 1$. It is not very sensitive within a wide range of values, especially after the optimal point. Similar phenomena are observed for other hypeparameters like λ_l. More details can be found in the supplementary material.

Case Study

To gain a deeper insight into the difference between SDAE and RSDAE, we choose one example article from *citeulike-a* and one example movie from *movielens-plot* to conduct a case study. The experiments are conducted in the sparse setting for *citeulike-a* and in the dense setting

Table 1: Example items (one movie and one article) with recommended tags

Example Article	Title: Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews			
	Top topic 1: language, text, mining, representation, semantic, concepts, words, relations, processing, categories			
Top 10 recommended tags	SDAE	True tag?	RSDAE	True tag?
	1. instance_within_labeled_concepts	no	1. sentiment_analysis	no
	2. consumer	**yes**	2. instance_within_labeled_concepts	no
	3. sentiment_analysis	no	**3. consumer**	**yes**
	4. summary	no	4. summary	no
	5. 31july09	no	**5. sentiment**	**yes**
	6. medline	no	**6. product_review_mining**	**yes**
	7. eit2	no	**7. sentiment_classification**	**yes**
	8. l2r	no	8. 31july09	no
	9. exploration	no	**9. opinion_mining**	**yes**
	10. biomedical	no	**10. product**	**yes**
Example Movie	Title: E.T. the Extra-Terrestrial			
	Top topic 1: crew, must, on, earth, human, save, ship, rescue, by, find, scientist, planet			
Top 10 recommended tags	SDAE	True tag?	RSDAE	True tag?
	1. Saturn Award (Best Special Effects)	**yes**	**1. Steven Spielberg**	**yes**
	2. Want	no	**2. Saturn Award (Best Special Effects)**	**yes**
	3. Saturn Award (Best Fantasy Film)	no	**3. Saturn Award (Best Writing)**	**yes**
	4. Saturn Award (Best Writing)	**yes**	4. Oscar (Best Editing)	no
	5. Cool but freaky	no	5. Want	no
	6. Saturn Award (Best Director)	no	6. Liam Neeson	no
	7. Oscar (Best Editing)	no	**7. AFI 100 (Cheers)**	**yes**
	8. almost favorite	no	**8. Oscar (Best Sound)**	**yes**
	9. Steven Spielberg	**yes**	9. Saturn Award (Best Director)	no
	10. sequel better than original	no	**10. Oscar (Best Music - Original Score)**	**yes**

for *movielens-plot*. We list the top 10 recommended tags provided by SDAE and RSDAE for the target items. Note that in the sparse setting recommendation is very challenging due to extreme sparsity of tagging information. As we can see in Table 1, the precisions for the target article are 10% and 60%, respectively. For the target movie the numbers are 30% and 60%. The huge gap shows that relational information plays a significant role in boosting the recommendation accuracy for the target items.

Looking into the recommended tag lists and the data more closely, we find that the example article 'Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews' is a WWW paper about sentiment classification. As shown in the table, most of the recommended tags provided by SDAE are trivial or irrelevant while RSDAE can understand the focus of the article a lot better and achieve a precision up to 60%. Among the six tags correctly predicted by RSDAE, two of them are related to articles linked to the target article directly. This means RSDAE is not simply recommending tags associated to linked articles in the citation network. By jointly performing relational learning and deep representation learning, these two parts actually benefit from each other and yield additional performance gain.

A similar phenomenon is observed in the example movie 'E.T. the Extra-Terrestrial' directed by Steven Spielberg. RSDAE correctly recommends three more tags for the award-winning movie. Among the three, two tags are related to movies directly linked to the target one. Interestingly, although the remaining tag 'Oscar (Best Music - Original Score)' does not show up in the tag lists of the linked movies, we find that 'E.T. the Extra-Terrestrial' is directly linked to the movie 'Raiders of the Lost Ark' (also directed by Steven Spielberg), which was once nominated for Oscar's academy award for best music. These results show that RSDAE as a relational representation learning model seems to do quite a good job in predicting award winners as well.

Conclusion

In this paper we first adapt SDAE to learn deep item representations for tag recommendation. Furthermore, we develop a probabilistic formulation for SDAE and, by extending this probabilistic SDAE, we propose RSDAE as a novel relational extension for integrating deep representation learning and relational learning in a principled way. Our model can also be naturally extended to handle multi-relational data due to its probabilistic nature. Experiments on real-world datasets from different domains show that our models are effective and outperform the state of the art. Besides, our framework is general enough to be adapted for other deep learning models like CNN as well.

Acknowledgments

This research has been supported by General Research Fund 621310 from the Research Grants Council of Hong Kong.

References

Bao, Y.; Fang, H.; and Zhang, J. 2014. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In *AAAI*, 2–8.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *JMLR* 3:993–1022.

Chen, H.-M.; Chang, M.-H.; Chang, P.-C.; Tien, M.-C.; Hsu, W. H.; and Wu, J.-L. 2008. Sheepdog: group and tag recommendation for flickr photos by automatic search-based learning. In *ACM Multimedia*, 737–740.

Chen, L.; Xu, D.; Tsang, I. W.-H.; and Luo, J. 2010. Tag-based web photo retrieval improved by batch mode re-tagging. In *CVPR*, 3440–3446.

Chen, C.; Zheng, X.; Wang, Y.; Hong, F.; and Lin, Z. 2014. Context-aware collaborative topic regression with social matrix factorization for recommender systems. In *AAAI*, 9–15.

Gales, M. J. F., and Airey, S. S. 2006. Product of gaussians for speech recognition. *CSL* 20(1):22–40.

Garg, N., and Weber, I. 2008. Personalized, interactive tag recommendation for flickr. In *RecSys*, 67–74.

Gupta, A., and Nagar, D. 2000. *Matrix Variate Distributions*. Chapman & Hall/CRC Monographs and Surveys in Pure and Applied Mathematics. Chapman & Hall.

Gupta, M.; Li, R.; Yin, Z.; and Han, J. 2010. Survey on social tagging techniques. *SIGKDD Explorations* 12(1):58–72.

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

Hu, Y.; Koren, Y.; and Volinsky, C. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM*, 263–272.

Kalchbrenner, N.; Grefenstette, E.; and Blunsom, P. 2014. A convolutional neural network for modelling sentences. *ACL* 655–665.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1106–1114.

Li, W.-J., and Yeung, D.-Y. 2009. Relation regularized matrix factorization. In *IJCAI*, 1126–1131.

Li, W.-J., and Yeung, D.-Y. 2011. Social relations model for collaborative filtering. In *AAAI*.

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

Rendle, S., and Schmidt-Thieme, L. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM*, 81–90.

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

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

Salakhutdinov, R., and Mnih, A. 2007. Probabilistic matrix factorization. In *NIPS*, 1257–1264.

Shen, Y., and Fan, J. 2010. Leveraging loosely-tagged images and inter-object correlations for tag recommendation. In *ACM Multimedia*, 5–14.

Shewchuk, J. R. 1994. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Carnegie Mellon University, Pittsburgh, PA, USA.

Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; and Manzagol, P.-A. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR* 11:3371–3408.

Vu, D. Q.; Asuncion, A. U.; Hunter, D. R.; and Smyth, P. 2011. Dynamic egocentric models for citation networks. In *ICML*, 857–864.

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

Wang, C., and Blei, D. M. 2011. Collaborative topic modeling for recommending scientific articles. In *KDD*, 448–456.

Wang, H., and Li, W.-J. 2013. Online egocentric models for citation networks. In *IJCAI*, 2726–2732.

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

Wang, M.; Ni, B.; Hua, X.-S.; and Chua, T.-S. 2012. Assistive tagging: A survey of multimedia tagging with human-computer joint exploration. *ACM Comput. Surv.* 44(4):25.

Wang, H.; Chen, B.; and Li, W.-J. 2013. Collaborative topic regression with social regularization for tag recommendation. In *IJCAI*, 2719–2725.

Weinberger, K. Q.; Slaney, M.; and van Zwol, R. 2008. Resolving tag ambiguity. In *ACM Multimedia*, 111–120.

Wu, L.; Yang, L.; Yu, N.; and Hua, X.-S. 2009. Learning to tag. In *WWW*, 361–370.

Yang, X.; Zhang, Z.; and Wang, Q. 2013. Personalized recommendation based on co-ranking and query-based collaborative diffusion. In *AAAI*.

Yu, H.; Deng, Z.; Yang, Y.; and Xiong, T. 2014. A joint optimization model for image summarization based on image content and tags. In *AAAI*, 215–221.

Zhao, L.; Pan, S. J.; Xiang, E. W.; Zhong, E.; Lu, Z.; and Yang, Q. 2013. Active transfer learning for cross-system recommendation. In *AAAI*.