Representation Learning for Homophilic Preferences

Trong T. Nguyen
School of Information Systems
Singapore Management University
ttnguyen.2014@smu.edu.sg

Hady W. Lauw
School of Information Systems
Singapore Management University
hadywlauw@smu.edu.sg

ABSTRACT

Users express their personal preferences through ratings, adoptions, and other consumption behaviors. We seek to learn latent representations for user preferences from such behavioral data. One representation learning model that has been shown to be effective for large preference datasets is Restricted Boltzmann Machine (RBM). While homophily, or the tendency of friends to share their preferences at some level, is an established notion in sociology, thus far it has not yet been clearly demonstrated on RBM-based preference models. The question lies in how to appropriately incorporate social network into the architecture of RBM-based models for learning representations of preferences. In this paper, we propose two potential architectures: one that models social network among users as additional observations, and another that incorporates social network into the sharing of hidden units among related users. We study the efficacies of these proposed architectures on publicly available, real-life preference datasets with social networks, yielding useful insights.

Keywords

user preferences; homophily; representation learning; social recommendation; restricted Boltzmann machine

1. INTRODUCTION

Representation learning [5] deals with deriving useful latent representations from a large amount of data, so as to enable better learning or prediction. It is an area of active research in diverse fields, including speech recognition [13], computer vision [19], natural language processing, etc., where approaches based on neural networks and deep learning are currently generating a lot of interest.

In this work, we are particularly interested in representation learning for preference data. Users express their preferences in various ways, e.g., when they assign ratings to items, when they tag or bookmark contents they like, when they purchase or re-purchase products, when they watch videos or listen to music. Such behaviors generate a large amount of data that carry a lot of information about what users like and dislike. Our aim is to learn a latent representation for each user's preferences from such behavioral data.

One pioneering work based on undirected graphical model [31] used Restricted Boltzmann Machine (RBM) to model user preferences from ratings. [31] showed that RBM could learn representations for user preferences from large-scale datasets such as Netflix [4], with collaborative filtering performance competitive with matrix factorization's. Moreover, being a basic building block of many deep learning models, RBM could potentially benefit from further development in the currently very active deep learning research.

Many online social media platforms today capture not just users' preferences through their behaviors, but also their social connections with one another through their friendship links. This highlights an important aspect of user preferences that has not yet been factored in by previous RBM-based models, i.e., *homophily*, or the tendency of people with social connections to have shared preferences at some level. Homophily is an established notion in sociology [26], and as we will survey in Section 2, factoring in social network information often helps collaborative filtering algorithms.

Given the preponderance of evidence from the literature on homophily, we hypothesize that social network information could also potentially improve the performance of RBM-based models at learning representations for user preferences. The research questions lie in *how* to do so appropriately. In this paper, we investigate two primary structures.

The first is to model social network links as *observations* that are generated by RBM, in addition to behaviors such as ratings or adoptions. This yields our first proposed model, SocialRBM-*Wing* described in Section 4, which features a dual-wing structure, i.e., two groups of visible units (corresponding to item adoptions and social links respectively) sharing a common layer of hidden units. This has the advantage of simplicity, by learning the social effects from data.

Rather than modeling social links as observations, the second model, SocialRBM-*Deep* described in Section 5, features a deep structure, i.e., a higher layer of *hidden units that are partially shared* among friends, while still maintaining some global sharing across all users. This has the advantage of incorporating the homophily assumption directly into the higher layer of hidden units, so as to allow the original lower layer of hidden units to focus on capturing patterns of item adoptions among all users. Moreover, deeper structures tend to produce more robust features, which are less likely to suffer from noise than shallower structures.

RecSys '16, September 15 - 19, 2016, Boston , MA, USA

ISBN 978-1-4503-4035-9/16/09…$15.00

DOI: http://dx.doi.org/10.1145/2959100.2959157

Contributions. In this work, we make the following contributions:

1. As far as we know, this is the first systematic study of different means for incorporating social network information into RBM-based models for preference data.
2. Towards this objective, we propose two models, namely: SocialRBM-*Wing* in Section 4 and SocialRBM-*Deep* in Section 5, and describe their inferences.
3. To verify the efficacy of the models, we conduct an empirical analysis based on two publicly available real-life datasets: *Delicious* and *Last.fm*, showing promising results in terms of improving the representations learnt by RBM-based models from these datasets.

2. RELATED WORK

A Boltzmann Machine (BM) [1] is a network of stochastic binary units, with one hidden layer and one visible layer. A more popular form with simpler inference is Restricted Boltzmann Machine (RBM) [31], which allows only connections between hidden and visible layers, but not within each layer. Variants of RBM include replacing binary units with Gaussian hidden units [31] or visible units [30] for continuous values, or learning from inequality constraints [35]. RBM can also be extended to have multiple hidden layers, forming a Deep Boltzmann Machine (DBM) [30].

The use of RBM for collaborative filtering was pioneered by [31]. This has been extended in several directions. [8] expanded into two sets of hidden layers: for modeling correlations among users and items respectively. [32] used autoencoder in place of RBM. [7] used autoencoder on ratings to learn representation for initializing an existing matrix factorization [22]. In this work, we build upon the well-established RBM model [31] to further incorporate social networks.

By incorporating social networks into modeling ratings or adoptions, our work is related to multi-modal representation learning. For instance, [17, 39] modeled text and images, [33] modeled audio and video, while [38] modeled text and ratings. We are distinct in two ways. For one, we are modeling a different set of modalities. For another, most of these works treat modalities as observations, whereas we also explore another means for incorporating one modality (social network) into the configuration of shared hidden units.

Previous works in collaborative filtering were mostly based on matrix factorization [2, 16, 18]. It thus follows that past works on incorporating social networks were modifications of matrix factorization [10, 40], e.g., generating social network links [23], regularizing friends' latent vectors [24], or expressing one's ratings [21] or latent vector [15] as a function of those of friends. These and our work essentially follow two different forks of collaborative filtering paradigms: matrix factorization and RBM respectively. The two paradigms are effectively complementary and co-existent. Previous studies showed that ensembles of collaborative filtering paradigms could yield better performance than any one paradigm on its own [3, 4, 14]. In this work, we focus on investigating the effects of homophily on RBM-based preference models.

There are also efforts to factor in social network into other types of models, such as topic modeling [28, 37]. Such works rely heavily on the availability of rich features, such as words. Modeling features is an orthogonal direction to our focus here on the effects of social networks on adoptions.

3. PRELIMINARIES

Restricted Boltzmann Machine (RBM). RBM is a form of Markov Random Field, with the structure of a bipartite graph, connecting two types of binary stochastic units: *visible* units $\mathbf{x} \in \{0,1\}^{\mathcal{N}}$ and *hidden* units $\mathbf{h} \in \{0,1\}^{K}$.

As a member of the family of energy-based models, its energy function is defined as follows:

$$E(\mathbf{x}, \mathbf{h}) = -\mathbf{a}^T\mathbf{x} - \mathbf{b}^T\mathbf{h} - \mathbf{x}^T\mathbf{W}\mathbf{h}, \tag{1}$$

where $\mathbf{a}$ and $\mathbf{b}$ are bias vectors for visible and hidden units respectively, and $\mathbf{W}$ is the $\mathcal{N} \times K$ matrix of weights associated with the connections between visible and hidden units.

Based on the energy function, the likelihood $P(\mathbf{x})$ of an observed boolean vector $\mathbf{x}$ is as follows:

$$P(\mathbf{x}; \Theta) = \sum_{\mathbf{h}} \frac{\exp(-E(\mathbf{x}, \mathbf{h}))}{Z}, \tag{2}$$

where $Z = \sum_{\mathbf{x}', \mathbf{h}'} \exp(-E(\mathbf{x}', \mathbf{h}'))$ is the *partition function* for normalization, while Θ is the set of model parameters.

The two conditional probabilities $P(\mathbf{h}|\mathbf{x})$ and $P(\mathbf{x}|\mathbf{h})$ can be expressed as follows:

$$P(h_k = 1|\mathbf{x}) = \sigma\left(\sum_{i=1}^{\mathcal{N}} W_{ik}x_i + b_k\right), \tag{3}$$

$$P(x_i = 1|\mathbf{h}) = \sigma\left(\sum_{k=1}^{K} W_{ik}h_k + a_i\right), \tag{4}$$

where $\sigma(t) = 1/1 + exp(-t)$ is the logistic function. The conditional distributions indicate that units in one layer are activated independently given activations from the other.

The model can be trained by maximizing the log-likelihood $\log P(\mathbf{x}; \Theta)$ using an approximation gradient ascent "contrastive divergence" (CD) introduced by [12].

RBM for Preference Data. Let us denote $\mathcal{U}$ to be the number of users, and $\mathcal{N}$ to be the number of items. For each user, we observe a visible vector $\mathbf{x} \in \{0,1\}^{\mathcal{N}}$, where '1' indicates the user's adoption of an item, and '0' otherwise. Here, for simplicity, we model binary adoptions. For multi-scale ratings, we can use softmax units instead, as in [31].

These observations can be modeled as one RBM instance for each user, with the same number of K hidden units, and all the instances share the same $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{W}$ weights. This way, we can derive a $K-$dimensional latent representation vector $\mathbf{h}$ for each user, within a huge space of combinations resulting from the activations of various binary hidden units.

Incorporating Social Networks. We assume that we are also given a social network graph $\mathcal{G}$, where each edge is a symmetric connection between two users. Our objective is to integrate $\mathcal{G}$ into an RBM model for user preferences, so as to arrive at a better latent representation $\mathbf{h}$ for each user.

In the RBM above, there is no user-specific parameter. Therefore, unlike some non-RBM approaches outlined in Section 2, we cannot simply tie user-specific parameters of friends. Directly employing regularization among hidden layers of RBM would not be applicable either, because the shared weights would lead to optimization on the whole network globally at the same time, rather than just locally among friends. This motivates our approach of placing social constraints to express homophily through the model structure so as to learn personalized representation using the hidden layers. In the next two sections, we describe two RBM-based structures that we find effective for this problem.

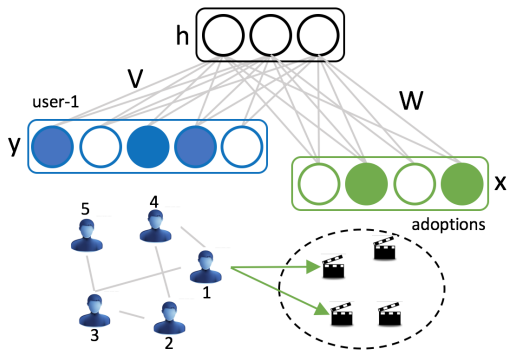

Figure 1: SocialRBM-*Wing*: both social connections and ratings/adoptions play a role as observations encoded jointly through a shared hidden layer.

4. SOCIAL NETWORK AS OBSERVATION

Our first model, SocialRBM-*Wing*, models item adoptions and social connections as two sets of observations, which are encoded through a shared hidden layer. As shown in Figure 1, the two sets of visible units resemble two "wings" attached to a "body" (the shared hidden layer), thus the name of the model. Our intuition is that if the model can learn co-occurrence patterns across both users' item adoptions and social connections, we would then be able to predict item adoptions given one's social connections.

4.1 Model

Each user is associated with two sets of visible units. The first set, for item adoptions, is $\mathbf{x} \in \{0,1\}^{\mathcal{N}}$, as described in Section 3. The second, for social connections, is $\mathbf{y} \in \{0,1\}^{\mathcal{U}}$, i.e., a binary vector of $\mathcal{U}$ dimensions, where each element y_j in the vector is activated if the user has a connection to user u_j in the social network $\mathcal{G}$. In the following, for simplicity, we describe the model and the learning for an individual user. For the collection of all users, the gradients with respect to the shared weight parameters are averaged over all users.

Figure 1 shows an illustration for one example user *user-1* or u_1, with a social connection to herself as well as to her friends (u_3 and u_4), thus the visible units y_1, y_4 and y_5 are activated (shaded in blue). The user is also observed to have two item adoptions (the second and fourth items), thus the visible units x_2 and x_4 are activated (shaded in green).

The conditional distribution of $\mathbf{x}$ is as shown in Eq. (4). In turn, the conditional distribution of $\mathbf{y}$ is shown in Eq. (5) below, where V_{jk} is the weight parameter (shared among all users) associated with the connection between visible unit y_j and hidden unit h_k.

$$P(y_j = 1|\mathbf{h}) = \sigma\left(\sum_{k=1}^{K} V_{jk} h_k + a_j\right) \tag{5}$$

The conditional distribution over hidden units for encoding both visible layers is given in Eq. (6), while the energy function is given in Eq. (7).

$$P(h_k = 1|\mathbf{x}, \mathbf{y}) = \sigma\left(\sum_{j=1}^{\mathcal{U}} V_{jk} y_j + \sum_{i=1}^{\mathcal{N}} W_{ik} x_i + b_k\right) \tag{6}$$

$$\begin{aligned} E(\mathbf{x}, \mathbf{y}, \mathbf{h}) = &- \sum_{j=1}^{\mathcal{U}} \sum_{k=1}^{K} y_j V_{jk} h_k - \sum_{i=1}^{\mathcal{N}} \sum_{k=1}^{K} x_i W_{ik} h_j \\ &- \sum_{k=1}^{K} h_k b_k - \sum_{j=1}^{\mathcal{U}} y_j a_j - \sum_{i=1}^{\mathcal{N}} x_i a_i \end{aligned} \tag{7}$$

Learning. From Eq. (7), we can derive the log-likelihood $\mathcal{L}(\mathbf{x}, \mathbf{y}; \Theta)$ of visible inputs $\mathbf{x}$ and $\mathbf{y}$ for each user:

$$P(\mathbf{x}, \mathbf{y}; \Theta) = \sum_{\mathbf{h}} \frac{\exp\left(-E(\mathbf{x}, \mathbf{y}, \mathbf{h})\right)}{Z(\Theta)}, \tag{8}$$

$$\mathcal{L}(\mathbf{x}, \mathbf{y}; \Theta) = \log \sum_{\mathbf{h}} \exp\left(-E(\mathbf{x}, \mathbf{y}, \mathbf{h})\right) - \log Z(\Theta), \tag{9}$$

where $Z(\Theta) = \sum_{\mathbf{x}', \mathbf{y}', \mathbf{h}'} \exp\left(-E(\mathbf{x}', \mathbf{y}', \mathbf{h}')\right)$ is the partition function to normalize the probability sum to 1.

Model parameters are learned by contrastive divergence [12] (CD) with n-step sampling (CD-n), as shown in Eq. (10) where P_0, P_n are respectively data distributions at step 0 and n after sampling from conditional distributions. In practice, we use one-step (CD-1) to approximate the gradients, which is commonly used for training RBM [34].

$$\frac{\partial \mathcal{L}(\mathbf{x}, \mathbf{y})}{\partial \Theta} = \langle \frac{-\partial E(\mathbf{x}, \mathbf{y}, \mathbf{h})}{\partial \Theta} \rangle_{P_0} - \langle \frac{-\partial E(\mathbf{x}', \mathbf{y}', \mathbf{h}')}{\partial \Theta} \rangle_{P_n} \tag{10}$$

Regularization. Due to the strong imbalance in data between seen and unseen (missing) items, we find that CD tends to lead to hidden biases increasing over time. If most of the hidden units were activated during the training process, then the model would not differentiate the representations among users. Hence, we incorporate regularization of the expected hidden activations around a desired level of activation τ into the objective function [20]. As shown in Eq. (11), the aggregated gradients are updated at the same time with the direction found in Eq. (10), where $\bar{P}$ is the expected activation of hidden units. λ is a tunable coefficient.

$$\tilde{\mathcal{L}}(\mathbf{x}, \mathbf{y}; \Theta) = \mathcal{L}(\mathbf{x}, \mathbf{y}; \Theta) - \lambda \sum_{k=1}^{K} | \tau - \bar{P}(h_k|\mathbf{x}, \mathbf{y}) |^2 \tag{11}$$

4.2 Inference

After learning the model parameters, we estimate the latent representation, and predict unseen adoptions by reconstructing the visible layer from the hidden layer, by performing one step of sampling to reconstruct the data as below:

$$\begin{aligned} \hat{h}_k &\leftarrow \sigma\left(\sum_{j=1}^{\mathcal{U}} V_{jk} y_j + \sum_{i=1}^{\mathcal{N}} W_{ik} x_i + b_k\right) \\ \hat{r}_i &\leftarrow \sigma\left(\sum_{k=1}^{K} W_{ki} \hat{h}_k + a_i\right) \end{aligned} \tag{12}$$

Compared to the RBM model for adoptions only, incorporating social network puts more constraints on deciding the user preferences, which will be affected by both global (all users) and local (their friends) patterns. In terms of learning, one more benefit is the potential to reduce overfitting due to cross-modality patterns produced from two sides of observations. In addition, to deal with cold-start users with few or no observed adoptions, the model can make use of the observations from their social connections to infer item adoptions. This is related to the notion of cross-modality in [39] with similar structure, but with different types of stochastic units, targeted for modeling text and images.

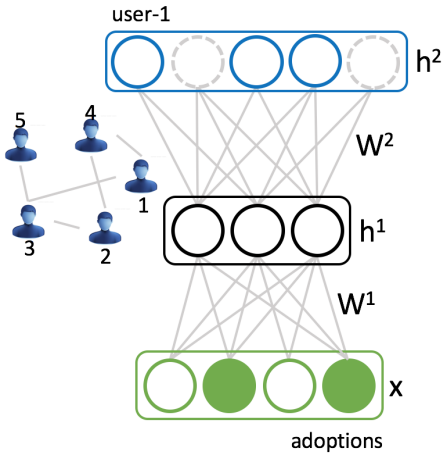

Figure 2: SocialRBM-*Deep*: The top layer $\mathbf{h^2}$ has $\mathcal{U}$ hidden units, corresponding to $\mathcal{U}$ users. Each user is represented by a single hidden unit on the top layer with weights shared with their friends. For example, user u_1 has connections to u_3 and u_4, thus the hidden units h_1^2, h_3^2 and h_4^2 are available for encoding u_1's adoptions. The other hidden units h_2^2 and h_5^2 will be unavailable (dashed). All K units in the middle layer $\mathbf{h^1}$ will always be available to all users.

5. SOCIAL NETWORK AS SHARING OF HIDDEN UNITS

In this section, we propose an alternative approach that adapts the model architecture to the social network structure. The intuition of SocialRBM-*Deep* is that instead of letting social connections bring users and their friends closer through shared observations, we allow friends to affect each other's representations through sharing hidden units.

5.1 Model

In terms of its structure, this model SocialRBM-*Deep* is a "flipped" version of SocialRBM-*Wing*, with the social network layer now stacked up on top of the item adoption layer. Its role also changes from visible units for observation to hidden units, as illustrated in Figure 2.

The structure of SocialRBM-*Deep* is reminiscent of a two-layer Deep Boltzmann Machine (DBM), thus its name. There are two layers of hidden units $\mathbf{h^1}$ (middle layer) and $\mathbf{h^2}$ (top layer). The middle layer is shared across all training instances (users). However, one critical difference from DBM is that our top layer is not shared across *all* instances. Our structure is such that each user is represented at the top layer by a group of hidden units. When learning the representation of a user, only the groups of hidden units corresponding to her own, as well as those of her friends' could be activated. This induces sharing particularly among users with social connections. Without losing generality, in this work we use a group size of one, to keep the number of parameters of SocialRBM-*Deep* the same with SocialRBM-*Wing*, thus establishing parity for comparison later. In other words, to encode the item adoptions of a user, SocialRBM-*Deep* makes use of $(F_u + 1)$ hidden units, where F_u is the number of friends of u. In Figure 2, for user u_1 with two friends, three hidden units are available at the top layer.

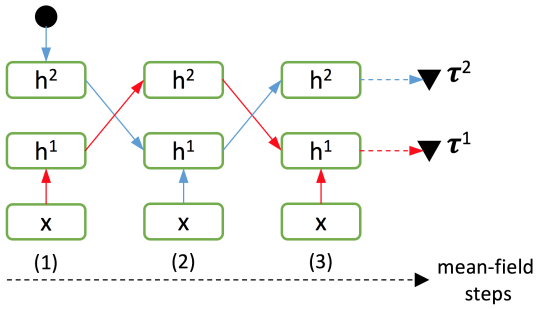

Figure 3: A visual for the sequence of forward steps in the mean-field inference, in which at the first step, hidden units $\mathbf{h^2}$ in the top layer will be randomized for initialization. The errors from both τ^1 vs. τ^2 in the regularization are backpropagated through this sequence for the gradient updates.

The energy function with all connections in the model is shown below:

$$E(\mathbf{x}, \mathbf{h^1}, \mathbf{h^2}) = -\sum_{i=1}^{\mathcal{N}}\sum_{k=1}^{K} x_i W_{ik}^1 h_k^1 - \sum_{k=1}^{K}\sum_{j=1}^{\mathcal{U}} h_k^1 W_{kj}^2 h_j^2 - \sum_{i=1}^{\mathcal{N}} x_i a_i - \sum_{k=1}^{K} h_k^1 b_k - \sum_{j=1}^{\mathcal{U}} h_j^2 a_j, \quad (13)$$

where $\mathbf{h^1}$ and $\mathbf{h^2}$ are the vectors of hidden units at the middle and top layers respectively.

Learning. The parameter updates are similar to Eq. (10), except that $P_0 = P(\mathbf{h^1}, \mathbf{h^2}|\mathbf{x})$ is now approximated by a variational distribution $Q(\mathbf{h^1}, \mathbf{h^2})$ [30] as described in Section 5.2. For the full distribution, we run n-step Gibbs sampler through following conditional distributions:

$$p(h_k^1 = 1|\mathbf{x}, \mathbf{h^2}) = \sigma\left(\sum_{i=1}^{\mathcal{N}} W_{ik}^1 x_i + \sum_{j=1}^{\mathcal{U}} W_{kj}^2 h_j^2 + b_k\right),$$

$$p(h_j^2 = 1|\mathbf{h^1}) = \sigma\left(\sum_{k=1}^{K} W_{kj}^2 h_k^1 + a_j\right),$$

$$p(x_i = 1|\mathbf{h^1}) = \sigma\left(\sum_{k=1}^{K} W_{ik}^1 h_k^1 + a_i\right).$$

Our model is based on social network to allow sharing in the training process of the top layer. Due to the connectivity across layers, the effect of sharing permeates through all levels of the deep structure. The model is a combination of two kinds of sharing: one is "global" across all users (the first layer); the other is "local" based on the graph structure, whereby users are co-trained only with direct friends or through mutual friends (via overlapping hidden groups). In contrast, the original RBM only explores global patterns.

To better initialize the parameters for mean-field steps, we apply pretraining stage for each layer in the deep network. As discussed in [30], mid-layer hidden units can be activated from both higher and lower layers; thus, the aggregation of posterior over these units could be contributed by halving the weights after learning, or duplicating them in training and keeping its value in the testing process.

Regularization. As in Section 4, we apply regularization to hidden activations. However, due to the use of vari-

ational inference for posterior approximation (as discussed in Section 5.2), the gradients could be computed via back-propagation algorithm (illustrated in Figure 3) through n steps of mean-field update as discussed in [29]. The aggregate objective function is as follows:

$$\begin{aligned}\tilde{\mathcal{L}}(\mathbf{x};\Theta) = &- \lambda_1 \sum_{k=1}^{K} \mid \tau_1 - \bar{P}(\mathbf{h_k^1}|\mathbf{x}) \mid^2 \\ &- \lambda_2 \sum_{j=1}^{\mathcal{U}} \mid \tau_2 - \bar{P}(\mathbf{h_j^2}|\mathbf{x}) \mid^2 + \mathcal{L}(\mathbf{x};\Theta),\end{aligned} \tag{14}$$

where τ_1, τ_2 are the desired levels of hidden unit activations in each of the two respective layers.

5.2 Inference

We apply the mean-field inference [30] to approximate the true posterior by a fully factorized distribution Q. Learning is conducted via minimizing the $KL(Q(\mathbf{h^1}, \mathbf{h^2})||P(\mathbf{h^1}, \mathbf{h^2}|\mathbf{x}))$ or maximizing the lower-bound of likelihood with respect to variational parameters $\big(\mu_k = Q(h_k^1 = 1), \mu_j = Q(h_j^2 = 1)\big)$ as shown below:

$$\begin{aligned}\ln P(\mathbf{x};\Theta) \geq &\sum_{i=1}^{\mathcal{N}} \sum_{k=1}^{K} x_i W_{ik}^1 \mu_k + \sum_{k=1}^{K} \sum_{j=1}^{\mathcal{U}} \mu_k W_{kj}^2 \mu_j - \ln Z(\Theta) \\ &+ \sum_{k=1}^{K} (\mu_k \ln \mu_k + (1-\mu_k) \ln(1-\mu_k)) \\ &+ \sum_{j=1}^{\mathcal{U}} (\mu_j \ln \mu_j + (1-\mu_j) \ln(1-\mu_j))\end{aligned} \tag{15}$$

The fixed-point equations are produced below :

$$\mu_k \leftarrow \sigma\left(\sum_{i=1}^{\mathcal{N}} W_{ik}^1 x_i + \sum_{j=1}^{\mathcal{U}} W_{kj}^2 \mu_j + b_k\right) \tag{16}$$

$$\mu_j \leftarrow \sigma\left(\sum_{j=1}^{\mathcal{U}} W_{kj}^2 \mu_k + a_j\right) \tag{17}$$

$$\hat{r}_i \leftarrow \sigma\left(\sum_{k=1}^{K} W_{ki}^1 \mu_k + a_i\right) \tag{18}$$

For experiments, we set 15 iterations for updating alternatingly from Eq. (16) and Eq. (17) until convergence. Finally, the probabilities for unseen items are computed via Eq. (18), and ranked in descending order for prediction.

6. EXPERIMENTS

The experimental objective is primarily to investigate the effects of homophily on RBM-based models for learning the representation of user preferences. We pursue this objective by evaluating the performance of comparable RBM-based models on two publicly-available[1] real-life datasets [6].

Datasets. The first dataset, *Delicious*, originally came from an online bookmarking site, whereby a set of users bookmark a set of URL's. These are essentially binary observations of item adoptions. In addition, it is also an online social network, which allows users to indicate friendship links. The second dataset, *LastFM*, originated from an online radio site, where users can tag their favourite artists (items). These are also modeled as binary observations of adoptions. Similarly, it has a social network among users.

[1]http://grouplens.org/datasets/hetrec-2011

	Delicious	LastFM
No. of users	1,867	1,892
No. of items	69,226	17,632
No. of adoptions <user, item>	104,220	92,834
No. of social links <user, user>	15,328	25,434
Adoption density	0.08%	0.27%
Social network density	0.44%	0.71%

Table 1: Dataset Sizes

The statistics for these two datasets are shown in Table 1. While the number of users are similar, *Delicious* is the larger and sparser dataset, with many more items and significantly lower adoption density. This sparsity also implies that it is the more difficult dataset for prediction. Other than their sizes, the two datasets are characteristically quite different. *Delicious*, driven by bookmarks, tends to have a greater level of personalization and expected homophily, whereas *LastFM*, driven by music artists, tends to have greater uniformity due to the presence of popular artists with broad appeal. Experimentation with these two contrasting natures would allow us to derive greater insights.

We focus on binary adoptions, because our main concern here is on the effects of social networks. With some modification, the models could apply to multi-scale ratings, which we will consider for future work. Moreover, we do not use tag information from the datasets, for parity with the original RBM model [31] that does not use such features either.

Comparisons. Because of our objective of studying homophily on RBM-based models, we can demonstrate this most clearly and directly by comparing the two proposed models SocialRBM-*Wing* and SocialRBM-*Deep* (with social network) to the original RBM model [31] (without social network). To establish parity among the models, we use the same dimensionality for the latent representations, i.e., $K = 100$ hidden units. This setting was also used in in [14, 31]. We will conduct this comparison in Section 6.1.

Although the dimensionality of the representation is the same, the models do not all have the same number of parameters. The two models with social networks SocialRBM-*Wing* and SocialRBM-*Deep* have an identical number of parameters. Compared to RBM, both have an additional number of $K \times \mathcal{U}$ weight parameters, which are required to connect the social layer and the hidden units. To ensure that the observed effects are not due to these additional parameters alone, in Section 6.2, we conduct another set of experiments comparing the same models (SocialRBM-*Wing* or SocialRBM-*Deep* respectively), but replacing the social networks with random networks of the same structure.

Finally, in Section 6.3, we briefly explore whether the learnt representations of friends tend to exhibit greater similarity after incorporating social networks during learning.

6.1 Comparison of Various Models

In this section, we conduct a comparison between the two proposed models and the baseline RBM model.

Task. Since one of the main applications of learning representations from preference data is for recommendation, here we evaluate the models on the task of predicting users' item adoptions. The adoption data for each user is ran-

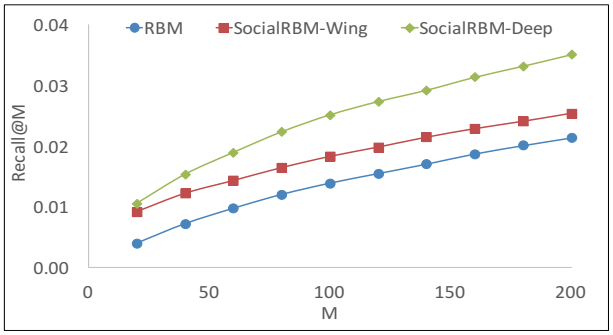

Figure 4: Recall@M for Delicious at various M's.

domly split into 80% for training vs. 20% for testing. For each dataset, we create ten such training/testing splits. For each user, we seek to predict the held-out item adoptions in the testing set. This is done by getting each model to learn from the training set, and output a ranked list of the top M items whose adoptions were not seen in the training set.

In the training stage, each sample data is divided into small batches of 100 instances (users) for each iteration. All models are trained with a learning rate of 0.003 in 1000 iterations util convergence. We also apply a momentum of 0.8 to speed up and a weight-decay of 0.001. For the pretraining stage in SocialRBM-*Deep*, we train each layer in 500 iterations with the similar process discussed in [30].

Metric. There are various ways to evaluate recommendations [11, 27]. For sparse preference datasets such as ours, unseen item adoptions may not necessarily mean that a user dislikes the items; the user might simply have been unaware of them. This makes it difficult to compute precision accurately. However, since the observed item adoptions in the testing sets are known to be true positives, following [28, 36], we focus on measuring recall. The recall of a model's prediction of top M items for a user is defined as follows:

$$recall@M \leftarrow \frac{\text{number of correctly predicted items in top } M}{\text{total number of held-out adopted items}}$$

For comparison across models, we average the recall@M across all the users. We vary the value of M in the range of $[20 \ldots 200]$. Higher recall at lower M indicates a better result, implying that the correct items tend to be among the top-ranked predictions. We average the results across the ten training/testing splits described above. Where appropriate, we present not only the mean, but also the standard deviation, as well as statistical significance test results.

Analysis. First, we look into the *Delicious* dataset. Figure 4 shows the cumulative recall up to top 200. As expected, as M increases, recall generally increases, because the numerator of the recall equation above would increase, while the denominator is stable. Importantly, based on the trend lines in Figure 4, it is evident that both SocialRBM-*Wing* and SocialRBM-*Deep* have higher recall results than the baseline RBM. Between the two models with social networks, SocialRBM-*Deep* has a higher performance than SocialRBM-*Wing*. Interestingly, the former tends to increase faster in recall than the latter as M increases, indicating SocialRBM-*Deep*'s greater effectiveness at placing the ground truth items higher in the prediction ranked lists.

To look into the differences between various models in greater detail, Table 2 shows not just the mean recall values,

M	RBM	SocialRBM-*Wing*	SocialRBM-*Deep*
20	0.0040 ± 0.0007	$0.0092 \pm 0.0007^{\dagger\dagger}$	$\mathbf{0.0105} \pm 0.0008^{\dagger\dagger\S\S}$
40	0.0072 ± 0.0007	$0.0122 \pm 0.0012^{\dagger\dagger}$	$\mathbf{0.0153} \pm 0.0013^{\dagger\dagger\S\S}$
60	0.0097 ± 0.0008	$0.0143 \pm 0.0013^{\dagger\dagger}$	$\mathbf{0.0190} \pm 0.0013^{\dagger\dagger\S\S}$
80	0.0120 ± 0.0004	$0.0164 \pm 0.0014^{\dagger\dagger}$	$\mathbf{0.0224} \pm 0.0014^{\dagger\dagger\S\S}$
100	0.0139 ± 0.0005	$0.0183 \pm 0.0011^{\dagger\dagger}$	$\mathbf{0.0252} \pm 0.0014^{\dagger\dagger\S\S}$
200	0.0214 ± 0.0011	$0.0254 \pm 0.0015^{\dagger\dagger}$	$\mathbf{0.0351} \pm 0.0022^{\dagger\dagger\S\S}$

Table 2: Comparison of Recall@M (mean $\pm$ standard deviation) for Delicious. Best results are in bold. $\dagger$ (0.05 level) and $\dagger\dagger$ (0.01 level) indicate statistically significant improvement over RBM. §§ (0.01 level) indicates statistically significant improvement over SocialRBM-*Wing*.

M	RBM	SocialRBM-*Wing*	SocialRBM-*Deep*
20	0.2298 ± 0.0020	$0.2319 \pm 0.0031^{\dagger}$	$\mathbf{0.2380} \pm 0.0037^{\dagger\dagger\S\S}$
40	0.3109 ± 0.0023	$0.3133 \pm 0.0030^{\dagger\dagger}$	$\mathbf{0.3252} \pm 0.0039^{\dagger\dagger\S\S}$
60	0.3651 ± 0.0027	$0.3677 \pm 0.0031^{\dagger\dagger}$	$\mathbf{0.3809} \pm 0.0033^{\dagger\dagger\S\S}$
80	0.4057 ± 0.0026	$0.4080 \pm 0.0033^{\dagger\dagger}$	$\mathbf{0.4224} \pm 0.0039^{\dagger\dagger\S\S}$
100	0.4369 ± 0.0023	$0.4397 \pm 0.0033^{\dagger}$	$\mathbf{0.4542} \pm 0.0042^{\dagger\dagger\S\S}$
200	0.5317 ± 0.0029	$0.5353 \pm 0.0033^{\dagger\dagger}$	$\mathbf{0.5500} \pm 0.0038^{\dagger\dagger\S\S}$

Table 3: Comparison of Recall@M for LastFM.

but also the standard deviations. We see that SocialRBM-*Deep* is the best (in bold). The standard deviations are also relatively small, implying that the mean values are quite reflective of the relative performances across models. We also conduct paired samples Student's t-test for statistically significant differences, indicating that the outperformance by SocialRBM-*Wing* over RBM, as well as that by SocialRBM-*Deep* over the other two models, are indeed statistically significant.

We reiterate that the key result here is the relative outperformance by the models with social network over the baseline RBM, of which the results here are strongly indicative. Compared to RBM, SocialRBM-*Deep* shows an increase by a factor of 1.5X to 2.5X. The increase by SocialRBM-*Wing* is smaller, but still significant. The absolute values of recall in *Delicious* are low, because it represents a very challenging dataset, especially without using any tag feature. A random predictor would attain a recall@200 of merely 0.0029. Thus, the performance of our models represents an increase by an order of magnitude over a random baseline.

Turning to *LastFM*, we show the corresponding table of results in Table 3. Similar observations as made above for *Delicious* on the relative outperformance by the models incorporating social networks can also be made for *LastFM*.

Comparing the two datasets, in terms of the absolute values, the results on *Delicious* are lower than those on *LastFM*. This can be explained by the disparity in the adoption densities shown in Table 1, i.e., 0.08% for *Delicious* vs. 0.27% for *LastFM*. The lower density indicates greater uncertainty in prediction, thus higher error rate. However, the relative improvements among models on *Delicious* are more significant than on *LastFM*. We hypothesize that this comes from the different characteristics of two datasets. For popular items, such as music artists, even unrelated people may still prefer similar music artists. Conversely, bookmarks are less frequent, and may be more prone to social influence, explaining the greater effects of homophily seen in *Delicious*.

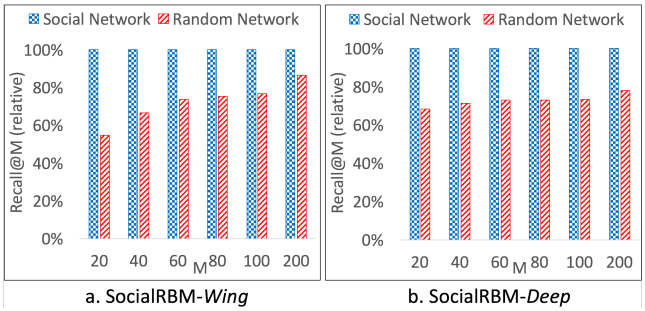

Figure 5: Relative Comparison of Social Network vs. Random Network for Delicious.

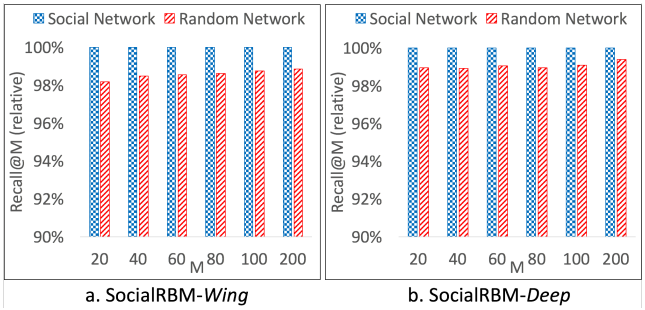

Figure 6: Relative Comparison of Social Network vs. Random Network for LastFM.

6.2 Randomization Study

We try to keep the models comparable by using the same dimensionality of latent representation across models. To investigate the effects of the social network itself, in this section we conduct another set of experiments, which attempt to control the effects of the model structure, and isolate the effects of homophily alone. The best way to do so is to compare *within* the same model, but swap the social network graph with a random graph of a similar structure.

Randomization. Ascertaining data mining results via randomization was advocated by [9]. Here, we follow the way they generate random graphs, so as to maintain the structure of the original graph in terms of the degree of each node. Essentially, a random graph is obtained by randomizing the adjacency matrix of the original graph, while keeping the same row and column sums. For each dataset, we generate ten random networks from the original social network, and compare the model run with these random networks with the same model run on the social network.

Analysis. For this study, we still use the recall@M metric described above. However, our focus here is not on the absolute performance, but on whether the same model will produce different levels of performance when run with social network or random network. To present these results clearly, we peg the performance by the model with social network at 100% (the absolute values can be seen in Table 2 and Table 3), and present the relative results by the same model but with random network as a percentage of the former.

For *Delicious*, Figure 5(a) shows the relative comparison of recall by SocialRBM-*Wing*, when social network is replaced by random network. Figure 5(b) shows that of SocialRBM-*Deep*. For both, we witness significant drops in recall for random networks, e.g., recall@20 drops to 55% for SocialRBM-*Wing*, and 65% for SocialRBM-*Deep*.

Figure 6(a) and Figure 6(b) show similar experiments on *LastFM*. They tell a similar story, though the drop is modest, with random networks attaining 98% of the performance of social network. This is another evidence of the relatively weaker effect of homophily on *LastFM* than on *Delicious*, which we trace to the different natures of the items. If the patterns of adopting popular artists are similar among users, changing the network structure would not lead to significant changes in adoptions, thus explaining the smaller drop rates.

For both *Delicious* and *LastFM*, the drops due to random networks are statistically significant at 0.01 level in all cases. This supports the hypothesis of the measurable effects of homophily on the RBM-based models for item adoptions.

Dataset	RBM	SocialRBM-*Wing*	SocialRBM-*Deep*
Delicious	0.0084 ± 0.0004	$0.0250 \pm 0.0008^{\dagger\dagger}$	$0.0210 \pm 0.0031^{\dagger\dagger}$
LastFM	0.0486 ± 0.0015	$0.0629 \pm 0.0121^{\dagger\dagger}$	$0.0821 \pm 0.0024^{\dagger\dagger}$

Table 4: Comparison of MAP for Similarity Ranking of Social Links. †† (0.01 level) indicates statistically significant improvement over RBM**.**

6.3 Similarity in Latent Representations

So far we learn that improvements arise from the right model architecture, as well as the right network structure. This comes from the intuition built into the models to bring together the representations of socially connected users, either via hidden units (SocialRBM-*Deep*) or visible units (SocialRBM-*Wing*). In this final experiment, we briefly explore this intuition, by using the similarities among the learnt representations of users to rank one's friends.

Task. The user representation is a vector of hidden units inferred from Sections 4 and 5. For each user, we rank other users based on the cosine similarity to their learnt representations. The aim is to see whether the user's friends would be ranked highly in this list, i.e., friends have similar representations. This study is not meant to be predictive, but rather *reflective*, whether the learnt representations may be correlated to the social connections in the training set.

Metric. We borrow a metric from information retrieval: Mean Average Precision or MAP [25]. This is computed as the mean of the average precision across all users, as follows.

$$MAP \leftarrow \frac{1}{U} \sum_{j=1}^{U} AvgPrecision_j$$

$AvgPrecision_j$ is computed by averaging the precisions at each of u_j's recall points (the ranks of u_j's friends).

Analysis. The main expectation is that if the proposed models do indeed absorb the social network information well during the learning phase, they would perform better at this task than the baseline RBM. Table 4 shows that on both datasets, SocialRBM-*Wing* and SocialRBM-*Deep* have significantly higher MAP than RBM. For *LastFM*, they increase by a factor of 1.3X to 1.7X. For *Delicious*, they increase by an even larger factor of 2.5X to 3X, supporting the hypothesis of a stronger homophily effect on *Delicious*. Overall, these results support that the learnt representations of friends become more correlated as a result of the homophily assumptions built into our proposed models.

7. CONCLUSION

We study the homophily effect on RBM-based models for preference datasets, proposing two models for incorporating social network. The first, SocialRBM-*Wing*, operates by fitting two sets of observations: item adoptions and social network. The second, SocialRBM-*Deep*, uses social network as a form of sharing hidden units at the top layer. These models are verified on two publicly available real-life item adoption datasets. The main conclusions are two-fold. First is the importance of the right architecture, as evidenced by SocialRBM-*Deep*'s outperformance over SocialRBM-*Wing* and RBM. Second is the importance of the right network information, as evidenced by the outperformance by social network over random network for each model. For future work, we plan to investigate enrichments to the proposed models, such as additional modalities.

Acknowledgments

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its NRF Fellowship Programme (Award No. NRF-NRFF2016-07).

8. REFERENCES

[1] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *TKDE*, 17(6):734–749, 2005.

[3] X. Amatriain and J. Basilico. Recommender systems in industry: A netflix case study. In *Recommender Systems Handbook*, pages 385–419. Springer, 2015.

[4] R. M. Bell and Y. Koren. Lessons from the Netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[5] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *TPAMI*, 35(8):1798–1828, 2013.

[6] I. Cantador, P. Brusilovsky, and T. Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (HetRec 2011). In *RecSys*, 2011.

[7] S. Deng, L. Huang, G. Xu, X. Wu, and Z. Wu. On deep learning for trust-aware recommendations in social networks. *TNNLS*, (99):1–14, 2016.

[8] K. Georgiev and P. Nakov. A non-iid framework for collaborative filtering with restricted Boltzmann machines. In *ICML*, pages 1148–1156, 2013.

[9] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas. Assessing data mining results via swap randomization. *TKDD*, 1(3), 2007.

[10] I. Guy. Social recommender systems. In *Recommender Systems Handbook*, pages 511–543. Springer, 2015.

[11] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *TOIS*, 22(1):5–53, 2004.

[12] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:2002, 2000.

[13] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine*, 29(6):82–97, 2012.

[14] M. Jahrer and A. Töscher. Collaborative filtering ensemble. In *KDD Cup*, pages 61–74, 2012.

[15] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, pages 135–142, 2010.

[16] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[17] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *TACL*, 2015.

[18] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[20] H. Lee, C. Ekanadham, and A. Y. Ng. Sparse deep belief net model for visual area v2. In *NIPS*, pages 873–880, 2008.

[21] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *SIGIR*, pages 203–210, 2009.

[22] H. Ma, I. King, and M. R. Lyu. Learning to recommend with explicit and implicit social relations. *TIST*, 2(3):29, 2011.

[23] H. Ma, H. Yang, M. R. Lyu, and I. King. SoRec: social recommendation using probabilistic matrix factorization. In *CIKM*, pages 931–940, 2008.

[24] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, pages 287–296, 2011.

[25] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[26] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, pages 415–444, 2001.

[27] P. Pu, L. Chen, and R. Hu. A user-centric evaluation framework for recommender systems. In *RecSys*, pages 157–164, 2011.

[28] S. Purushotham, Y. Liu, and C.-c. J. Kuo. Collaborative topic regression with social matrix factorization for recommendation systems. In *ICML*, pages 759–766, 2012.

[29] S. Reed, K. Sohn, Y. Zhang, and H. Lee. Learning to disentangle factors of variation with manifold interaction. In *ICML*, pages 1431–1439, 2014.

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

[31] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In *ICML*, pages 791–798, 2007.

[32] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie. AutoRec: Autoencoders meet collaborative filtering. In *WWW*, pages 111–112, 2015.

[33] N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep Boltzmann machines. In *NIPS*, pages 2222–2230, 2012.

[34] T. Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *ICML*, pages 1064–1071, 2008.

[35] T. Tran, D. Phung, and S. Venkatesh. Thurstonian Boltzmann machines: Learning from multiple inequalities. In *ICML*, pages 46–54, 2013.

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

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

[38] H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In *KDD*, pages 1235–1244, 2015.

[39] E. P. Xing, R. Yan, and A. G. Hauptmann. Mining associated text and images with dual-wing harmoniums. In *UAI*, 2005.

[40] X. Yang, Y. Guo, Y. Liu, and H. Steck. A survey of collaborative filtering based social recommender systems. *Computer Communications*, 41:1–10, 2014.