Collaborative Topic Regression with Denoising AutoEncoder for Content and Community Co-Representation

Trong T. Nguyen
Singapore Management University
80 Stamford Road
Singapore 178902
ttnguyen.2014@smu.edu.sg

Hady W. Lauw
Singapore Management University
80 Stamford Road
Singapore 178902
hadywlauw@smu.edu.sg

ABSTRACT

Personalized recommendation of items frequently faces scenarios where we have sparse observations on users' adoption of items. In the literature, there are two promising directions. One is to connect sparse items through similarity in content. The other is to connect sparse users through similarity in social relations. We seek to integrate both types of information, in addition to the adoption information, within a single integrated model. Our proposed method models item content via a topic model, and user communities via an autoencoder model, while bridging a user's community-based preference to her topic-based preference. Experiments on public real-life data showcase the utility of the model, particularly when there is significant compatibility between communities and topics.

KEYWORDS

Cold-start recommendation; topic model; autoencoder; social collaborative filtering; collaborative deep learning

1 INTRODUCTION

Recommender systems are pervasive. Its expanse covers various spheres of life, influencing what advertisements are shown to us, what shows we watch, what news we read, etc. Its depth manifests in how it is becoming essentially the primary user interface when dealing with many services today, e.g., Netflix, Facebook News Feed. What drives these recommender systems is the historical behavioral data of users as they interact with the system. The more we know about a user, the better the recommendation is likely to be.

Despite the increasing reliance on recommendation for the targeted delivery of various services, getting recommendations right is very challenging due to the diversity of preferences among individuals. Crucially, for the vast majority of individuals, we have insufficient data to personalize their recommendations well due to various reasons. For one, a user may not identify herself during a transaction, e.g., guest checkout, and the behavioral data may not be recorded. Alternatively, she may interact with multiple systems, thus splintering her behavioral data across various "silos".

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM'17, November 6–10, 2017, Singapore.
© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ISBN 978-1-4503-4918-5/17/11…$15.00
DOI: https://doi.org/10.1145/3132847.3133128

This paper focuses on recommendation scenarios involving *sparse* users or items, with limited information on adoptions or purchases. Given a set of users $\mathcal{U}$ and a set of items $\mathcal{V}$, we observe historical interactions or adoptions involving some user $u_i \in \mathcal{U}$ and item $v_j \in \mathcal{V}$. This is indicated by a variable r_{ij}, which may be binary or continuous depending on whether we model adoption or rating respectively. In the sparse scenario, the set R of observed r_{ij}'s contains relatively few user-item pairs. The objective is to recommend to a user u_i, other items that u_i has not adopted in the past.

While there are other modeling paradigms [1, 11], the most popular way of modeling adoption or rating is matrix factorization [3, 10]. Every user and every item is respectively associated with a K-dimensional latent vector. Here, we overload the notations of users and items to also refer to their latent vectors. r_{ui} is expressed as a function of the inner product between the latent vectors, i.e., ${u_i}^T v_j$. However, in sparse scenarios, there are too few r_{ui} observations to learn the latent vectors well. Importantly for prediction, the model parameters learnt from a dearth of observations may overfit the insufficient training set, and do not generalize well to unseen cases.

The limited observations on adoption need to be supplemented with other information. There are primarily two broad directions. The first is to consider the similarity in content among items. Often, the items to be recommended come with textual description. Let d_j denote a text document that is the content of an item v_j. For instance, movies have synopses, research papers have abstracts, products have description, etc. For a sparse item, with little or no prior history, we assume that its behavior would be similar to other items with similar content. Some works explore integration of topic modeling and rating prediction [7, 14]. In particular, Collaborative Topic Regression or CTR [14] associates each item with a distribution θ_j over K topics, which is assumed to be similar to the latent vector v_j. Thus, we could learn a latent vector for an item based on its content, leveraging on similarity in topics with other items.

The second direction is to leverage on the similarity in behavior among users. Neighborhood-based collaborative filtering [12] determined similarity based on historical adoptions, which for some users are very sparse. A promising direction is to use another source of information, such as the social network. Social collaborative filtering [2, 4–6] assumes that users with similar social connections are more likely to have similar latent vectors. The principle of homophily suggests that friends tend to be alike [8]. A sparse user could essentially "borrow" some information from her friends.

In prior work, these two directions have mostly been explored separately. We observe the potential synergies between the two. While content helps to bridge sparse items, social network helps to bridge sparse users. We seek to build a model that integrates

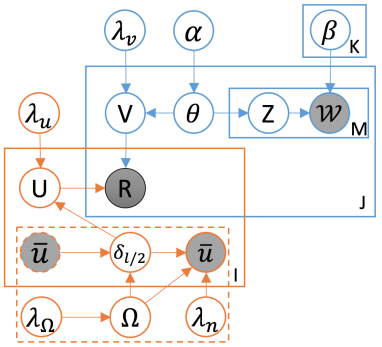

Figure 1: Our model CTR-DAE: a combination of topic model (blue), and autoencoder for community representation shown (red), in which all shaded nodes represent observations and plain nodes represent latent random variables.

both content and social network. The previous state of the art in this respect is CTR-SMF [9], which marries content-based CTR [14] and link-based SoRec [5]. However, the social component of CTR-SMF is based on social matrix factorization, which is oriented around user pairs. This "pair-wise" approach might lead to *uncertain* information being propagated among sparse users via spurious linkages, increasing the overfitting under very sparse scenarios.

We hypothesize that a better way to incorporate social network in a sparse scenario is to model "communities", i.e., patterns of connections involving multiple users (in other words "group-wise", instead of "pair-wise"). Instead of matrix factorization for link prediction, we propose using Denoising AutoEncoder (DAE) [13] to learn patterns of social connections, which are then used as a regularizer in a joint model with topic modeling for content CTR. Our proposed method is called CTR-DAE. On one hand, it enhances the "cooperation" among users, and on the other hand it prevents overfitting of users on their own adoptions. This is achieved through a mapping between the "community" distribution that a user belongs to, and the "topic" distribution learnt from her items (via the items' textual content). In this way, the user is expected to absorb topics shared by a majority of friends in her communities to a greater extent, while imbibing the less popular topics in her communities (that may be considered as noise in some cases) to a lesser extent.

Contributions. We make the following contributions. *First*, in Section 2, we design the CTR-DAE model that integrates topic modeling for item content, autoencoder for user community, as well as matrix factorization for user interaction with items in a single joint model. *Second*, through experiments in Section 3, we validate the utility of two various approaches ("pair-wise" versus "group-wise") over different scenarios of data sparsity.

2 MODEL

In this section, we describe our method Collaborative Topic Regression with Denoising AutoEncoder or CTR-DAE and how it seeks to better deal with sparse recommendation. The main principle being investigated in this work is the integration of communities and topics for co-representation, where CTR-DAE draws some inspiration from [14] in its topic modeling, as well as from [15] in its use of autoencoder. Notably, while [15] employs autoencoder for content alone, through experimentation with various designs, we discover that autoencoder is more useful in our context for modeling communities in social networks.

2.1 Generative Process

Figure 1 shows the plate diagram illustrating the dependencies among various variables of CTR-DAE. The overall generative process of CTR-DAE is as follows:

(1) For each item $v_j \in \mathcal{V}$:
- Draw its topic proportions: $\theta_j \sim \text{Dirichlet}(\alpha)$
- Draw its latent offset: $\epsilon_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K)$, and we have the item latent vector $v_j = \epsilon_j + \theta_j$
- For every word w_{jm} in document d_j:
  - Draw a topic assignment: $z_{jm} \sim \text{Multinomial}(\theta_j)$
  - Draw a word: $w_{jm} \sim \text{Multinomial}(\beta_{z_{jm}})$

(2) For each layer l of the autoencoder:
- Draw weight matrix: $\Omega_l \sim \mathcal{N}(0, \lambda_\Omega^{-1} I_{K_l})$
- Draw bias vector: $b_l \sim \mathcal{N}(0, \lambda_\Omega^{-1} I_{K_l})$
- $\forall i, \delta_{l,i} = \sigma(\delta_{l-1,i}\Omega_l + b_l)$

(3) For each user $u_i \in \mathcal{U}$:
- Draw her binary vector: $\bar{u}_i \sim \mathcal{N}(\delta_{L,i}, \lambda_n^{-1} I_{|\mathcal{U}|})$
- Draw her latent offset: $\epsilon_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K)$, and we have the user latent vector $u_i = \epsilon_i + \delta_{L/2,i}$

(4) For each rating r_{ij} in the set R:
- Draw a rating: $r_{ij} \sim \mathcal{N}(u_i^T v_j, C_{ij}^{-1})$

In its construction, CTR-DAE includes three main components as follows. The first component, illustrated by Step 1 of the generative process above is a topic model for items' documents. For each item, we extract its topic distribution θ_j to regularize its latent factor representation v_j. β_z is the distribution of words for topic z, and I is an identity matrix with dimensionality indicated in its subscript.

The second component, illustrated by Step 2 and Step 3 of the generative process is an autoencoder. Step 2 describes an autoencoder model of L layers, where each layer l has K_l hidden units. In our model, each user u_i is associated with a set of visible units (or a binary vector $\bar{u}_i$). The number of visible units is the number of users in $\mathcal{U}$. The s^{th} element of $\bar{u}_i$ is activated if u_i has a social connection to user u_s in the social network graph $\mathcal{G}$. The key point in this approach is that through the hidden layers of the autoencoder, we seek some patterns of cooccurrences ("*community*") of *visible* units ("$friends$"). Step 3 describes how we use the mid-layer representation $\delta_{L/2,i}$ to regularize the user's latent vector u_i.

The third component, illustrated by Step 4 of the generative process is a matrix factorization model for ratings, expressing the rating r_{ij} as a function of the latent vectors u_i and v_j. Importantly, this brings together the community representation $\delta_{L/2,i}$ (via u_i) and the topic representation θ_j (via v_j) into a common latent space.

In CTR [14], a user is regularized independently via $\mathcal{N}(0, \lambda_u^{-1} I_K)$. As shown in Figure 2, our key idea is to represent the user in the community space to be closer to her friends, by producing $\mathcal{N}(\delta_{L/2,i}, \lambda_u^{-1} I_K)$, in which $\delta_{L/2,i}$ encodes the community of user i through DAE structure. Particularly, tying topic and community space is to let users move along with their community rather individually to prevent overfitting under very sparse scenarios.

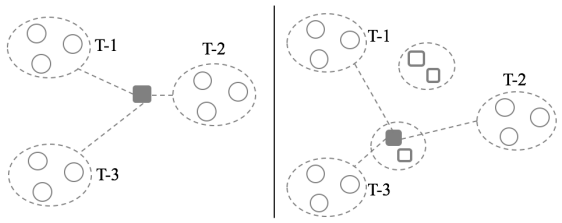

Figure 2: Illustration for CTR **(left) and** CTR-DAE **(right) interpretation: squared and circular nodes denote users and items respectively,** T**-**n **is for item topic** n**. Compared to** CTR **(on the left), where the user** i **is regularized via** $\mathcal{N}(0, \lambda_u^{-1} I_K)$**; our method** CTR-DAE **(on the right) places users into their community by producing** $\mathcal{N}(\delta_{L/2,i}, \lambda_u^{-1} I_K)$**, in which** $\delta_{L/2,i}$ **encodes the community of user** i **through DAE structure.**

2.2 Parameter Learning

CTR-DAE could be trained by maximizing the posterior, equivalent to maximizing the complete log-likelihood of all random variables $U, V, \theta, \Omega_l, b_l$ and $R, \bar{U}$ given hyperparameters $\lambda_u, \lambda_v, \lambda_\Omega, \lambda_n, \beta$.

$$\begin{aligned}\mathcal{L} = &-\frac{\lambda_u}{2}\sum_i \|u_i - \delta_{L/2}(\bar{u}_i; \Omega, b)\|^2 - \frac{\lambda_n}{2}\sum_i \|\bar{u}_i - \delta_L(\bar{u}_i; \Omega, b)\|^2 \\ &-\frac{\lambda_v}{2}\sum_j \|v_j - \theta_j\|^2 + \sum_j \sum_m \log\left(\sum_z \theta_{jz}\beta_{z,w_{jm}}\right) \\ &-\frac{\lambda_\Omega}{2}\sum_l \left(\|\Omega_l\|^2 + \|b_l\|^2\right) - \sum_{i,j}\frac{C_{ij}}{2}\left(r_{ij} - u_i^T v_j\right)^2,\end{aligned}$$

where $\delta_l(\bar{u})$ is the output at the layer l from the forward function δ with the input $\bar{u}$, and C represents for how much confidence on the observed data [14]. λ_u refers to how strong users depend on her "community" representation δ; the higher the value, the stronger the social effects. As discussed in [15], the ratio of $\lambda_u : \lambda_n$ indicates the effects between the decoder and the topic model on learning $\{\Omega, b\}$, while λ_v regularizes the dependency of v_j on θ.

The user and item latent vectors are updated as below:

$$\begin{aligned}u_i &\leftarrow \left(VC_iV^T + \lambda_u I\right)^{-1}\left(VC_iR_i + \lambda_u \delta_{L/2}(\bar{u}_i; \Omega, b)^T\right) \\ v_j &\leftarrow \left(UC_jU^T + \lambda_v I\right)^{-1}\left(UC_jR_j + \lambda_v \theta_j\right)\end{aligned}$$

For the remaining parameters $\{\theta, \beta, \Omega_l, b_l\}$, we follow the learning approaches discussed in [15] and [14].

Prediction. We perform the in-matrix prediction with the parameters $\{U^*, V^*, \theta, \beta, W\}$ locally optimized after the learning stage. Let D be the observed data, we can use the point estimate of $\{\Omega, \epsilon_i\}$ and $\{\theta_j, \epsilon_j\}$ to approximate the ratings expectation:

$$\begin{aligned}\mathcal{E}\left[r_{ij}|D\right] &\approx \left(\mathcal{E}\left[\delta_{l/2}(\bar{u}_i; \Omega)|D\right]^T + \mathcal{E}\left[\epsilon_i|D\right]\right)\left(\mathcal{E}\left[\theta_j|D\right] + \mathcal{E}\left[\epsilon_j|D\right]\right) \\ r_{ij}^* &\approx \left(u_i^*\right)^T v_j^*\end{aligned}$$

In the case of out-matrix prediction for new users or items, where $\mathcal{E}[\epsilon_i|D] = \mathcal{E}[\epsilon_j|D] = 0$, we can simply replace $\{u_i, v_j\}$ by $\{\delta_{L/2}(\bar{u}_i; \Omega), \theta_j\}$ respectively.

3 EXPERIMENTS

The objective is to evaluate the integration of users' social network and items' content for recommendation in sparse scenarios.

	Delicious
No. of users	1,867
No. of items	69,226
No. of adoptions (user, item)	104,220
No. of social links (user, user)	15,328
Adoption density	0.08%
Social network density	0.44%

Table 1: Dataset Sizes

Datasets. We experiment with Delicious dataset[1] (see Table 1), whereby users bookmark a set of URL's associated with tags (the item content). Each user has a list of friends (the social network). We model adoptions, i.e., $r_{ij} = 1$ if u_i adopts v_j, and zero otherwise.

Training vs. Testing. Our main focus is the sparse scenario. We keep 50% of each user's ratings as test set, sufficiently large so as to reduce the potential variance in performance across samples. For the training set, we experiment with a few settings of different degrees of sparsity, i.e., {5%, 10%, 15%, 50%} of each user's ratings. We ensure that a larger training set contains a smaller training set, i.e., 5% training data is a subset of 10% training data. The expectation is that in the very sparse cases of {5%,10%,15%}, most users have very few observations, and therefore they need to rely more on the communities, drawing from their communities' mapping to the item topics. In contrast, for a higher density, i.e., {50%}, a user may have sufficient data to express a higher reliance on personalization.

Comparative Methods. Our CTR-DAE models both content and social networks. The most comparable baseline is CTR-SMF [15], which extends CTR [14] with a link model based on social matrix factorization (SMF). We also include a comparison to the base CTR, which models topic-based content but not social networks.

To make the effect of social networks even clearer, independent of content, we create another version of CTR-DAE by removing the content effect, which we refer to as CF-DAE. As a baseline to this, we include a matrix factorization-based collaborative filtering CF.

Metrics. For each method, we derive the predictions for each user, and present it as a ranked list. For these datasets, we only observe positive examples, i.e., the items adopted by users. The unobserved adoptions may not mean that the user dislikes the items, but may be due to the sparsity of the data. Therefore, an appropriate evaluation metric in such cases is recall [14]. The recall 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 all methods, the number of latent factors or topics K is set to 200 as in [9]. Both datasets are trained within 300 iterations with the learning rate of 0.003 and the autoencoder is pretrained for 500 iterations before joint-learning with CTR. We also found the best performance with a single layer, and the ratio $\lambda_n : \lambda_u$ is also found to be 1:1 for the better performance. For hyperparameter tuning, we found that the λ_u tends to reduce from 0.1 to 0.01 corresponding to from the sparse- to dense-settings, while the λ_v is always at 100 for the best performance. This trend is also consistent with our hypothesis above, where the effect of community reduces from stronger to weaker corresponding in sparse- to dense-settings.

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

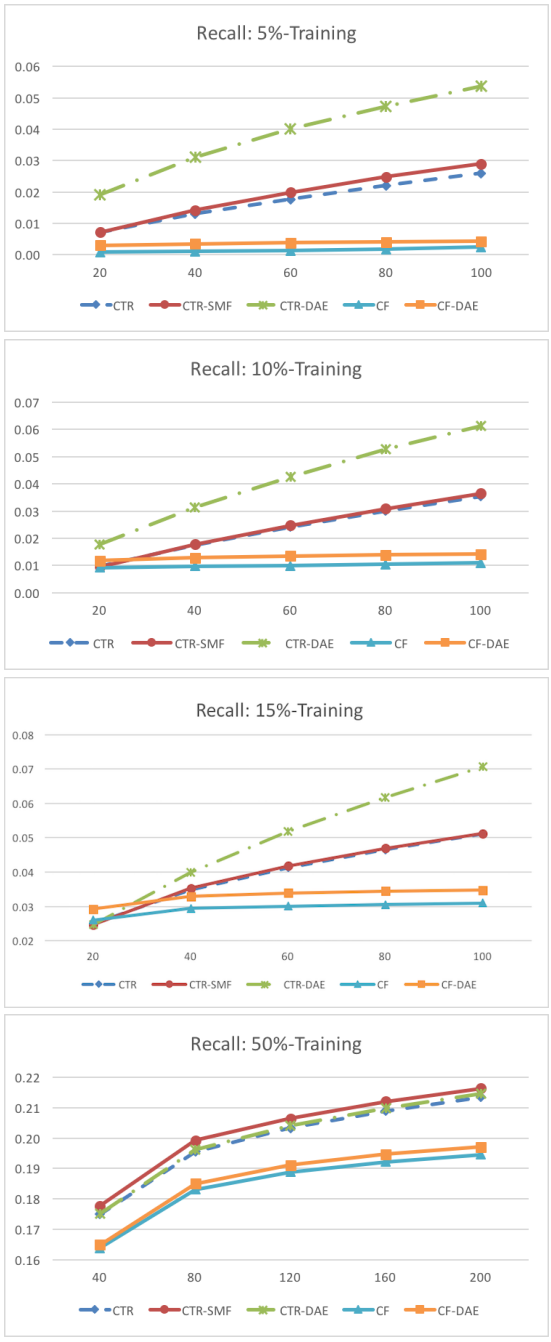

Figure 3: Recall@M on Delicious for various Top M.

Comparison. Figure 3 shows the performance of the comparative methods for various degrees of sparsity. In each plot, we show the *recall@M* performance for various $M \in [20, 100]$.

In very sparse cases, i.e., {5%,10%,15%}, each user has relatively few ratings. In these cases, evidently CTR-DAE outperforms the other methods. CTR-DAE outperforms CTR by a factor of 2.0X~2.7X and 1.7X~1.8X in very sparse settings (for 5% and 10% training splits respectively), while these numbers are 1.9X~2.7X and 1.6X~1.8X as compared to CTR-SMF. Correspondingly, for the methods without content, CF-DAE that is aware of social networks also outperforms the basic CF, further validating the effects of social network. However, the difference in performance CF-DAE vs. CF is smaller than that between CTR-DAE vs. CTR-SMF or CTR, which suggests that social network jointly with content has an even stronger impact.

Under very dense setting (e.g. 50%), the performance of various CTR-based methods tend to converge. Similarly for CF-based models. This trend in performance is due to the increasing amount of adoption information for each individual user. As each user gains "self-sufficiency", which leads to higher diversity in users' preferences, bridging community and topic has less utility. As alluded to earlier, CTR-SMF suffers more in very sparse scenarios due to the propagation of *uncertain* topics over very sparse users.

4 CONCLUSION

We explore the effects of communities on learning users' topical preferences, especially in sparse scenarios. CTR-DAE combines topic modeling for content and autoencoder for community representation for regularizing item and user latent factors respectively. We validate CTR-DAE across different degrees of sparsity. For very sparse cases, CTR-DAE outperforms the baselines, especially on datasets where communities play a key role in bridging social preferences and topic mixtures. For future work, we would investigate different mappings between communities and topics.

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).

REFERENCES

[1] Thomas Hofmann. 2004. Latent semantic models for collaborative filtering. *TOIS* 22, 1 (2004), 89–115.
[2] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*. 135–142.
[3] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.
[4] Hao Ma, Irwin King, and Michael R Lyu. 2009. Learning to recommend with social trust ensemble. In *SIGIR*. 203–210.
[5] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*. 931–940.
[6] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *WSDM*. 287–296.
[7] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*. 165–172.
[8] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual Review of Sociology* 27, 1 (2001), 415–444.
[9] Sanjay Purushotham and Yan Liu. 2012. Collaborative Topic Regression with Social Matrix Factorization for Recommendation Systems.. In *ICML*.
[10] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization.. In *NIPS*, Vol. 1. 2–1.
[11] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *ICML*. 791–798.
[12] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*. 285–295.
[13] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *ICML*. 1096–1103.
[14] Chong Wang and David M. Blei. 2011. Collaborative Topic Modeling for Recommending Scientific Articles. In *KDD*. 448–456.
[15] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative Deep Learning for Recommender Systems. In *KDD*. 1235–1244.