A graphical model for multi-relational social network analysis

Mohammad Khoshneshin MKHOSH@BGSU.EDU
Bowling Green State University, Bowling Green, Ohio 43403 USA

Nick Street NICK-STREET@UIOWA.EDU
University of Iowa, Iowa City, Iowa 52242 USA

Abstract

In this paper, we propose a graphical model for multi-relational social network analysis based on latent variable models. Latent variable models are one of the successful approaches for social network analysis. These models assume a latent variable for each entity and then the probability distribution over relationships between entities is modeled via a function over latent variables. Here, we use latent feature networks (LFN) — a general purpose framework for multi-relation learning via latent variable models. The experimental results show that using the side information via the proposed model can drastically improve the link prediction task in a social network.

1. Introduction

One of the successful approaches to modeling social network data is latent variable models (Hoff et al., 2002). To learn a social network via latent variable models, a latent variable is assigned to each person. Here, the latent variable of each person can be interpreted as features that are related to their friendship. The key idea is that people who are friends have similar features. Although we do not know these hidden features a priori, we can learn them given the network. Using statistical methods, we try to estimate a set of features for each person so that these features can predict the friendship links accurately. This method has been found to be one of the most successful approaches for recommending friends.

Now assume that we have some other relational information about people in the social network. This is the situation in social networking websites such as facebook.com, twitter.com, or delicious.com. In Facebook, people like different pages, posts by other users, hyperlinks, etc. In Delicious, users bookmark various webpages of interest. Now the question is: can we use the information from one network to learn other networks better? For example, if we know the bookmarking network of users — bookmarking webpages in the Delicious example — can we use this information to predict friendship more accurately? This problem is known as multi-relational learning.

In this paper, we use latent feature networks (LFN) as a framework for multi-relational social network analysis (Khoshneshin, 2012). The LFN approach extends latent variable models to multi-relational learning. The LFN first assumes a local latent variable model for each relational network. Then it connects those local latent variable models by enforcing correlation between the local features of entities participating in both networks. In the Delicious example, we have two relational networks: one is friendships among users and the other is bookmarking between users and webpages. Users have two sets of latent features given that they participate in both friendship and bookmarking relationships. However, using modeling links, we enforce correlation between local hidden feature sets of users for friendship and bookmarking relationships. This way, we can learn the social network relationships more accurately as reported in the experimental study section.

2. Latent feature networks

The latent feature networks (LFN) (Khoshneshin, 2012) — a general framework for multi-relational learning — is explained in this section. In latent feature networks, each relationship is represented by a component. Each component is a latent variable model in which the relationship between two entities is modeled as a function of latent variables. We call these

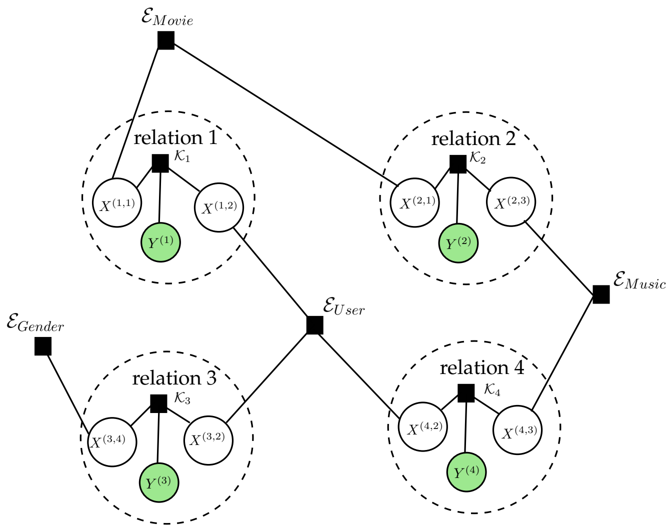

Figure 1. A latent feature network for 4 relations among 4 entity types

latent variables latent features since they are expected to represent the intrinsic features of the entities. If an entity is participating in different relationships, then it will have different local latent feature variables for each relation. However, different latent feature variables for an entity are forced to be dependent. This dependency can be an equality constraint in one extreme — the same latent variable for each relation — which underfits the data. On the other extreme, these local latent variables could be independent which results in overfitting the data. The main goal of LFN is designing a general model for statistical relational learning that can model arbitrary relation types and is powerful in learning.

Let $X_i^{(r,k)}$ represent the latent feature variable of entity i with type k for relation r. $Y_{ij}^{(r)}$ shows the relation between entity i from the first entity type and entity j from the second entity type participating in r. Y may be fully or partially observed while X is always hidden. Here, we consider dyadic relations — between two entity types. Generalizing to n-ary relations is straightforward from the modeling viewpoint. We define two functions to capture the dependencies among the variables. The first one is the relation kernel function $\mathcal{K}_r(X^{(r,k_{r1})}, X^{(r,k_{r2})}, Y^{(r)})$ which models the relationship r between entity type k_{r1} and k_{r2}. Another dependency regards correlation between the same entities participating in different relationships. Let $\mathcal{E}_k()$ be a function of $X^{(r,k)}$ for all relations r in which entity type k participates. This function enforces dependency among the same entities of type k.

To clarify the notation, we give a fictional example about a multi-context recommendation system. In this example, entity types are:

1. Movie
2. User
3. Book
4. Gender.

Note that gender is usually considered as a feature in machine learning algorithms but we can model it as a relationship as well. Assume that 4 relation types are observed:

1. Movie-User: rating of a user for a movie
2. Movie-Book: is true if a movie is based on a book
3. User-Gender: gender of a user
4. Book-User: rating of a user for a book.

Using the representative indexes above, $X^{(1,1)}$ and $X^{(2,1)}$ represent the local latent variables of movies in Movie-User and Movie-Book relationships. $X^{(1,2)}$, $X^{(3,2)}$, and $X^{(4,2)}$ represent the local latent variables of users in Movie-User, User-Gender, and Book-User relationships respectively. $X^{(2,3)}$ and $X^{(4,3)}$ represent the local latent variables of books in Movie-Book and Book-User relationships. $X^{(3,4)}$ represents the latent variable of gender categories (male and female) for the User-Gender relationship. $Y^{(1)}$ is the rating of a user for a movie, $Y^{(2)}$ is one if a movie is based on a book, $Y^{(3)}$ is the assignment of a gender to a user and $Y^{(4)}$ is is the rating of a user for a book.

A latent feature network can be represented by a factor graph. Random variable nodes represent latent feature variables X and relationship or observed variables Y. Function nodes represent relationship kernels $\mathcal{K}$ and dependencies between latent feature variables $\mathcal{E}$.

Figure 1 represents a latent feature network for the example explained above. Movie participates in relations 1 and 2 and therefore a latent variable for each relation is defined — $X^{(1,1)}$ and $X^{(2,1)}$. Similarly, user participates in relations 1, 3, and 4, $X^{(1,2)}$, $X^{(3,2)}$, and $X^{(4,2)}$ for each relation. Factor function $\mathcal{K}_r$ for $r = 1, ..., 4$ models the relationship as a function of latent feature variables. Factor function $\mathcal{E}_k$ for $k = 1, .., 3$ models the dependency among the latent feature variables of the same entity participating in different relationships.

Given the defined variables and relations for the latent feature network, the joint probability distribution over all random variables is given by

$$P(X,Y,\Theta) = \frac{1}{Z}\exp\left(\sum_r \mathcal{K}_r + \sum_k \mathcal{E}_k\right), \quad (1)$$

where $Z = \int_{X,Y,\Theta} \exp\left(\sum_r \mathcal{K}_r + \sum_k \mathcal{E}_k\right)$ is the normalizing or partition function. This may be a sum instead of integral for some relation variables Y. Θ represents the parameters that might be used in the modeling. In the example in Figure 1, the joint probability distribution is

$$\begin{aligned} &P(X,Y,\Theta) = \\ &\frac{1}{Z}\exp[\mathcal{K}_1(X^{(1,1)},X^{(1,2)},Y^{(1)}) + \mathcal{K}_2(X^{(2,1)},X^{(2,3)},Y^{(2)}) \\ &+ \mathcal{K}_3(X^{(3,4)},X^{(3,2)},Y^{(3)}) + \mathcal{K}_4(X^{(4,2)},X^{(4,3)},Y^{(4)}) \\ &+ \mathcal{E}_1(X^{(1,1)},X^{(2,1)}) + \mathcal{E}_2(X^{(1,2)},X^{(3,2)},X^{(4,2)}) \\ &+ \mathcal{E}_3(X^{(2,3)},X^{(4,3)})]. \quad (2)\end{aligned}$$

Note that some of the parameter variables Θ may exist in any factor.

The aim of the latent feature networks is providing a unified model for learning arbitrary relational data with any number of relations and any type of relationships. Although a few other latent variable models which are capable of learning multi-relational data have been proposed, they are usually limited to one type of relationships. Furthermore, one can deduce those models from LFN as LFN is more general. The most important related model in the literature is collective matrix factorization (Singh & Gordon, 2008; Singh, 2009) which assumes the same latent variable for an entity that participates in different relationships. Such approach limits the learning power as different relationships demand different features. LFN resolves this problem by assuming a latent variable for each relationship, and then it learns the dependency between relationships.

3. Multi-relational social network analysis

In this section, a latent network model is proposed for link prediction in social networks using information from a side network. Latent variable models are successful approaches for social network analysis (Hoff et al., 2002). A social network consists of social actors and edges between them which usually convey concepts such as friendship. In latent variable models, individuals are mapped into a latent space and the relationship between them is a function of the position of individuals in the latent space.

In this paper, we exploit side information for better social network analysis. Side information might be the interaction of individuals with other entities. One example is the ratings of individuals for items as we have seen in the collaborative filtering problem. Here, we propose a latent feature network for modeling a social network with a side network of bookmarked URLs. We derive a Gibbs sampling algorithm for Bayesian inference and run experiments on a real world dataset — Delicious dataset[1]. We use the two networks that exist in this dataset: a bookmarking network and a social network. Based on the experiments, using both networks significantly outperforms using only the social network for link prediction.

3.1. Model

In the latent feature network for social network analysis, we introduce one random variable per entity. In link prediction with a side network of bookmarking, there are two groups of entities: individuals and URLs. We use latent variable $X_i^{(1)}$ — a $1 \times k_1$ vector — for features of individual i participating in the social network relationship, latent variable $X_i^{(2)}$ — a $1 \times k_2$ vector — for features of individual i participating in the bookmarking relationship, and Y_h — a $1 \times k_2$ vector — for features of URL h participating in the bookmarking relationship. Note that the social network relationship is modeled by a k_1-dimensional latent variable feature space while the bookmarking relationship is modeled by a k_2-dimensional latent feature space. Here, two types of relationships exist: the friendship between individuals i and i' denoted by the friendship link binary variable $l_{ii'}$ which is one if there are friends and zero otherwise, and the bookmarking relationship between individual i and URL h denoted by the the bookmarking binary variable b_{ih} which is one if user i has bookmarked URL h and zero otherwise.

The generative model for the proposed model is depicted in Figure 2 which is as follows:

1. Choose precision matrix $\Lambda_X \sim \mathcal{W}ishart(W_0, \nu_0)$
2. Choose precision matrix $\Lambda_Y \sim \mathcal{W}ishart(W_0, \nu_0)$
3. For each individual i:
   (a) chose latent variables $\begin{pmatrix} X_i^{(1)} \\ X_i^{(2)} \end{pmatrix} \sim \mathcal{N}ormal(0, \Lambda_X)$
4. For each URL h:
   (a) Choose latent variable $Y_h \sim \mathcal{N}ormal(0, \Lambda_Y)$

[1] http://www.grouplens.org/node/462

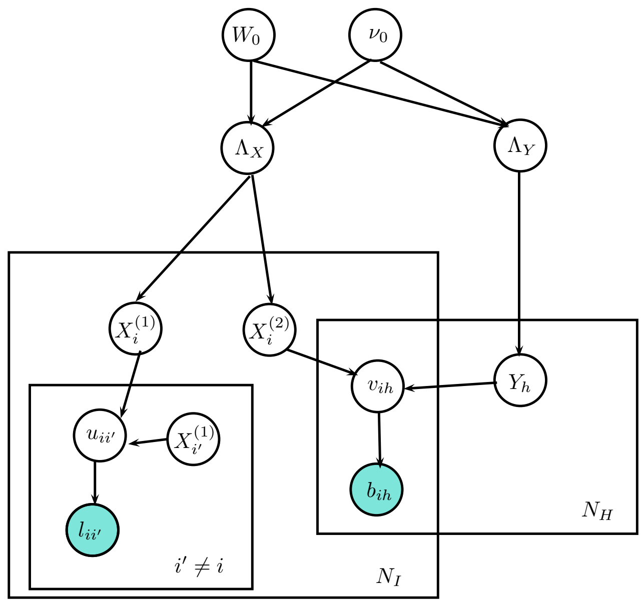

Figure 2. The graphical model of the latent feature network for social network analysis with side information

5. For each pair of individuals (i, i') where $i \neq i'$:
   (a) Choose latent variable $u_{ii'} \sim \mathcal{N}ormal(X_i^{(1)} {X_{i'}^{(1)}}^T, 1)$
   (b) Set link variable $l_{ii'} = \begin{cases} 1 & \text{if } u_{ii'} > 0 \\ 0 & \text{otherwise} \end{cases}$
6. For each pair of individual and URL (i, h):
   (a) Choose latent variable $v_{ih} \sim \mathcal{N}ormal(X_i^{(2)} {Y_h}^T, 1)$
   (b) Set bookmark variable $b_{ih} = \begin{cases} 1 & \text{if } v_{ih} > 0 \\ 0 & \text{otherwise} \end{cases}$

Note that we dropped one of the X's superscripts for simplicity. Here we use auxiliary random variables $u_{ii'}$ and v_{ih} similar to those used in (Albert & Chib, 1993). The benefit of such a modeling is that the conditional probability distributions are known which is very helpful in inference. Note that the precision matrix Λ_X enforces transferring knowledge between two different relationships. In the following, we derive the Gibbs sampling algorithm to derive the posterior distribution of needed statistics.

3.1.1. Gibbs sampling

To infer the latent variables we use Gibbs sampling. In Gibbs sampling, we need to derive the conditional probability of each latent variable given the rest of latent variables. The conditional distribution of precision matrices is as follows:

$$\Lambda_X | X \sim \mathcal{W}ishart\left((W_0 + \sum_i X_i^T X_i)^{-1}, N_I + \nu_0 \right), \tag{3}$$

where $X_i = \begin{pmatrix} X_i^{(1)} \\ X_i^{(2)} \end{pmatrix}$, and

$$\Lambda_Y | Y \sim \mathcal{W}ishart\left((W_0 + \sum_h X_h^T X_h)^{-1}, N_H + \nu_0 \right). \tag{4}$$

The conditional distribution of auxiliary random variables $u_{ii'}$ and v_{ih} are as follows:

$$u_{ii'} | l_{ii'}, X_i^{(1)}, X_{i'}^{(1)} \sim$$
$$1(u_{ii'} \geq 0)^{l_{ii'}} 1(u_{ii'} < 0)^{1-l_{ii'}} \mathcal{N}ormal\left(X_i^{(1)} {X_{i'}^{(1)}}^T, 1 \right), \tag{5}$$

and

$$v_{ih} | b_{ih}, X_i^{(2)}, Y_h \sim$$
$$1(v_{ih} \geq 0)^{b_{ih}} 1(v_{ih} < 0)^{1-b_{ih}} \mathcal{N}ormal\left(X_i^{(2)} Y_h^T, 1 \right). \tag{6}$$

Finally, the conditional distribution of latent feature variables Y_h and X_i are as follows:

$$Y_h | X^{(2)}, v, \Lambda_Y \sim \mathcal{N}ormal\left(\left(\sum_i v_{ih} X_i^{(2)} \right) \Sigma_h, \Sigma_h \right), \tag{7}$$

where

$$\Sigma_h = (\Lambda_Y + \sum_i {X_i^{(2)}}^T X_i^{(2)})^{-1}, \tag{8}$$

and

$$X_i | Y, X_{-i}, u, v, \Lambda_X \sim$$
$$\mathcal{N}ormal\left(\left(\sum_{i' \neq i} u_{ii'} X_{i'}^{(1)} \quad \sum_h v_{ih} Y_h \right) \Sigma_i, \Sigma_i \right), \tag{9}$$

where X_{-i} includes all latent variables X except for X_i and

$$\Sigma_i = \left(\Lambda_X + \begin{pmatrix} \sum_{i' \neq i} {X_{i'}^{(1)}}^T X_{i'}^{(1)} & 0 \\ 0 & \sum_h Y_h^T Y_h \end{pmatrix} \right)^{-1}. \tag{10}$$

The Gibbs sampling algorithm is depicted in Figure 3. First, we iteratively sample random variables for a specific number of iterations which is known as the burn-in phase. Then we sample needed statistics to score a link — $X_i^{(2)} {X_{i'}^{(2)}}^T$ for each $i' \neq i$.

```
Input [{l_{ii'}}, {b_{ih}}, W_0, ν_0]
Output [Pr(l_{ii'} = 1)]
Initialize [Λ_X,Λ_Y,{X_i^(1)},{X_i^(2)},{Y_h},{u_{ii'}},{v_{ih}}]
Repeat
      Sample Λ_X|X
      Sample Λ_Y|Y
      ∀_{i<i'} sample u_{ii'}|l_{ii'}, X_i^(1), X_{i'}^(1)
      ∀_{i,h} sample v_{ih}|b_{ih}, X_i^(2), Y_h
      ∀_h sample Y_h|X^(2), v, Λ_Y
      ∀_i ∈ RandomizedOrder sample X_i|Y, X_{-i}, u, v, Λ_X
Until (Sufficient samples achieved)
```

Figure 3. Gibbs sampling algorithm for the social network analysis with side bookmarking network

4. Experimental results

For experimental evaluations, we used a subset of the Delicious dataset. We sampled from this dataset so that remaining individuals and URLs have enough support in the social network matrix and the bookmarking network. In the subset, there are $N_I = 869$ individuals and $N_H = 2214$ URLs. The main goal is predicting the friendship link between two individuals. For evaluation, for each individual i we randomly selected a friend and eliminated the link from the social network. Then for each selected link, 50 individuals who were not friends with individual i were randomly selected. We use metric average rank for evaluation:

$$AverageRank = \frac{\sum_i Rank(j_i, S_i)}{N_I} \tag{11}$$

where j_i is the index of the individual who is friends with individual i and S_i is the set of individuals who are not friends with individual i, and $Rank(j_i, S_i)$ computes the rank of the individual with true friendship among all individuals. The rank can be between 1 and 51.

Here, we compare two algorithms: the LFN algorithm explained in the previous section which exploits both friendship and bookmarking relationship — shown by LFN-FB — and an LFN algorithm that only exploits the friendship relationship with the same setup of LFN-FB but without the bookmarking component — shown by LFN-F. LFN-F is very similar to the algorithm proposed in (Hoff et al., 2002) but with Probit model instead of Logit model.

The results are given in Figure 4. The lower average rank is better. Using all dimensions, LFN-FB outperforms LFN-F (significant at p-value$< 10^{-4}$).

4.1. Conclusion

In this paper, we derive a model and inference algorithm for social network analysis with side information. We show that link prediction via latent space models can be improved by using extra information given latent feature network setup.

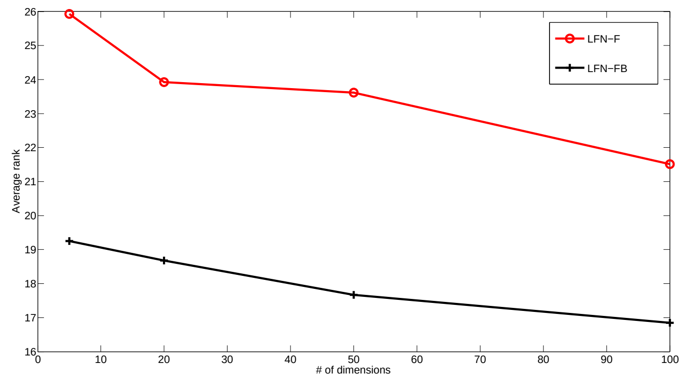

Figure 4. The average rank results for LFN-F versus LFN-FB with different dimensions

References

Albert, J.H. and Chib, S. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, pp. 669–679, 1993.

Hoff, P.D., Raftery, A.E., and Handcock, M.S. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460): 1090–1098, 2002.

Khoshneshin, M. *Latent Feature Networks for Statistical Relational Learning*. PhD thesis, University of Iowa, Department of Management Sciences, 2012.

Singh, A.P. *Efficient Matrix Models for Relational Learning*. PhD thesis, Carnegie Mellon University, School of Computer Science, Machine Learning Department, 2009.

Singh, A.P. and Gordon, Geoffrey J. Relational learning via collective matrix factorization. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pp. 650–658, New York, NY, USA, 2008. ACM.