Latent Multi-group Membership Graph Model

Myunghwan Kim MYKIM@STANFORD.EDU
Stanford University, Stanford, CA 94305, USA

Jure Leskovec JURE@CS.STANFORD.EDU
Stanford University, Stanford, CA 94305, USA

Abstract

We develop the *Latent Multi-group Membership Graph* (*LMMG*) model, a model of networks with rich node feature structure. In the *LMMG* model, each node belongs to multiple groups and each latent group models the occurrence of links as well as the node feature structure. The *LMMG* can be used to summarize the network structure, to predict links between the nodes, and to predict missing features of a node. We derive efficient inference and learning algorithms and evaluate the predictive performance of the *LMMG* on several social and document network datasets.

1. Introduction

Network data, such as social networks of friends, citation networks of documents, and hyper-linked networks of webpages, play an increasingly important role in modern machine learning applications. Analyzing network data provides useful predictive models for recommending new friends in social networks (Backstrom & Leskovec, 2011) or scientific papers in document networks (Nallapati et al., 2008; Chang & Blei, 2009).

Research on networks has focused on various models of network link structure. Latent variable models (Airoldi et al., 2007; Hoff et al., 2002; Kemp et al., 2006) decompose a network according to hidden patterns of connections between the nodes, while models based on Kronecker products (Leskovec et al., 2010; Kim & Leskovec, 2012a; 2011a) accurately model the global network structure. Though powerful, these models account only for the structure of the network, while ignoring observed features of the nodes. For example, in social networks users have profile information, and in document networks each node also contains the text of the document that it represents. Above models can find patterns which account for the connections between nodes, but they cannot account for the node features.

Node features along with the links between them provide rich and complementary sources of information and should be used simultaneously for uncovering, understanding and exploiting the latent structure in the data. In this respect, we develop a network model that considers both the emergence of links of the network and the structure of node features such as user profile information or text of a document.

Considering both sources of data, links and node features, leads to more powerful models than those that only consider links. For example, given a new node with a few of its links, traditional network models provide a predictive distribution of nodes to which it might be connected. However, to predict links of a node, our model does not need to see any of its links. It can predict links using only the node's features. For example, we can suggest a user's friendships based only on the profile information, and recommend hyperlinks of a webpage based only on its textual information. Moreover, given a new node and its links, our model also provides a predictive distribution of node features. This can be used to predict features of a node given its links or even predict missing or hidden features of a node given its links. For example, in our model interests of a user or keywords of a webpage can be predicted using only the connections of the network. Such predictions are out of reach for traditional models of networks.

We develop a *Latent Multi-group Membership Graph* (*LMMG*) model of networks that explicitly ties nodes into groups of shared features and linking structure (Figure 1). Nodes belong to multiple latent groups and the occurrence of each node feature is determined by a logistic model based on the group memberships of the given node. Links of the network are then generated via link-affinity matrices. Each link-affinity matrix Θ_i represents a table of link tendencies, and an appropriate entry of Θ_i is chosen based on whether or not a pair of nodes share the membership in group i. We derive efficient algorithms for model pa-

Appearing in *Proceedings of the* 29th *International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012. Copyright 2012 by the author(s)/owner(s).

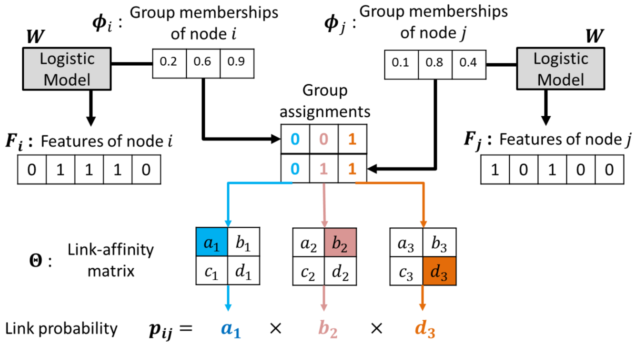


Figure 1. Latent Multi-group Membership Graph model. A node belongs to multiple latent groups at once. Based on group memberships features of a node are generated using a logistic model. Links are modeled via link-affinity matrices which allows for rich interactions between members and non-members of groups.

rameter estimation and prediction. We study the performance of *LMMG* on real-world social and document networks. We investigate the predictive performance on three different tasks: node feature prediction, link prediction, and supervised node classification. The *LMMG* provides significantly better performance on all three tasks than natural alternatives and the current state of the art.

2. *LMMG* Model Formulation

The *Latent Multi-group Membership Graph* (*LMMG*) model is a model of a (directed or undirected) network and nodes which have categorical features. Our model contains two important ingredients for innovation (See Figure 1).

First, the model assigns nodes to latent groups and allows nodes to belong to multiple groups at once. In contrast to multinomial models of group membership (Airoldi et al., 2007; Chang & Blei, 2009) where the membership of a node is shared among the groups (the probability over group memberships of a node sums to 1), we model group memberships as a series of Bernoulli random variables (ϕ_i in Figure 1), which indicates that nodes in our model can truly belong to multiple groups. Thus, in contrast to multinomial topic models, a higher probability of node membership to one group does not necessarily lower the probability of membership to some other group in the *LMMG*.

Second, for modeling the links of the network, each group k has associated a link-affinity matrix (Θ in Figure 1). Each link-affinity matrix represents a table of link affinities given that a pair of nodes belongs or does not belong to group k. Thus, depending on the combination of the memberships of nodes to group k, an appropriate element of Θ_k is chosen. For example, the entry $(0, 0)$ of Θ_k captures the link-affinity when none of the nodes belongs to group k, while $(1, 0)$ stores the link-affinity when first node belongs to the group (1) but the second does not (0). As we will later show that this allows for rich flexibility in modeling the links of the network as well as for uncovering and understanding the latent structure of the network.

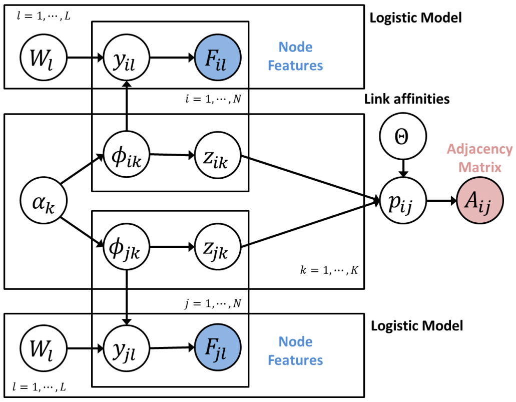


Figure 2. Plate model representation of *LMMG* model.

Next we formalize the *LMMG* model illustrated in Figure 2 and describe it in a generative way. Formally, each node $i = 1, 2, \cdots, N$ has a real-valued group membership $\phi_{ik} \in [0, 1]$ for each group $k = 1, 2, \cdots, K$. ϕ_{ik} represents the probability that node i belongs to group k. Assuming the Beta distribution parameterized by α_{k1}, α_{k2} as a prior distribution of group membership ϕ_{ik}, we model the latent group assignment z_{ik} for each node as follows:

$$\begin{aligned} \phi_{ik} &\sim \text{Beta}(\alpha_{k1}, \alpha_{k2}) \\ z_{ik} &\sim \text{Bernoulli}(\phi_{ik}) \quad \text{for } k = 1, 2, \cdots, K\,. \end{aligned} \tag{1}$$

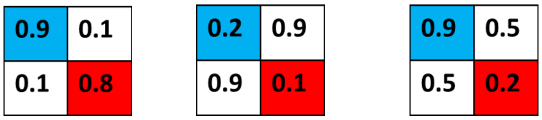
Since each group membership z_{ik} of a node is independent, a node can belong to multiple groups simultaneously.

The group memberships of a node affect both node features and its links. With respect to node features, we limit our focus to binary-valued features and use a logistic function to model the occurrence of node's features based on the groups it belongs to. For each feature F_{il} of node i ($l = 1, \cdots, L$), we consider a separate logistic model where we regard group memberships $\phi_{i1}, \cdots, \phi_{iK}$ as input features of the model. In this way, the logistic model represents the relevance of each group membership to the presence of a node feature. For convenience, we refer to the input vector of node i for the logistic model as $\phi_i = [\phi_{i1}, \cdots, \phi_{iK}, 1]$, where $\phi_{i(K+1)} = 1$ represents the intercept term. Then,

$$\begin{aligned} y_{il} &= \frac{1}{1 + \exp(-w_l^T \phi_i)} \\ F_{il} &\sim \text{Bernoulli}(y_{il}) \quad \text{for } l = 1, 2, \cdots, L \end{aligned} \tag{2}$$

where $w_l \in \mathbb{R}^{K+1}$ is the logistic model parameter for the l-th node feature. The value of each w_{lk} indicates the contribution of group k to the presence of node feature l.

In order to model the links of the network, we build on the idea of the Multiplicative Attribute Graph (MAG)



0.9	0.1
0.1	0.8

(a) Homophily

0.2	0.9
0.9	0.1

(b) Heterophily

0.9	0.5
0.5	0.2

(c) Core-periphery

Figure 3. Link structures modeled by link-affinity matrices.

model (Kim & Leskovec, 2012a). Here each latent group k has associated a link-affinity matrix $\Theta_k \in [0,1]^{2\times 2}$. Each entry of the link-affinity matrix indicates a tendency of linking between a pair of nodes depending on whether they belong to the group k or not. In other words, given the group assignments z_{ik} and z_{jk} of nodes i and j, z_{ik} "selects" a row and z_{jk} "selects" a column of Θ_k and the linking tendency from node i to node j is captured by $\Theta_k[z_{ik}, z_{jk}]$. After acquiring such link-affinities from all the groups, we define the link probability p_{ij} as the product of the link-affinities. Therefore, based on latent group assignments and link-affinity matrices, we determine each entry of the adjacency matrix $A \in \{0,1\}^{N\times N}$ of the network as follows:

$$\begin{aligned} p_{ij} &= \prod_k \Theta_k[z_{ik}, z_{jk}] \\ A_{ij} &\sim \text{Bernoulli}(p_{ij}) \quad \text{for } i,j = 1,2,\cdots N\,. \end{aligned} \tag{3}$$

The parameter matrix Θ_k represents the link affinity with respect to the particular group k. The model offers flexibility in a sense that we can represent many types of linking structures. In Figure 3, by varying the link-affinity matrix, the model can capture heterophily (love of the different), homophily (love of the same), or core-periphery structure. This way the affinity matrix allows us to discover the effects of node features on links of the network.

The node features and the links of the network are connected via group memberships ϕ_i. For instance, suppose that w_{lk} is large for some feature l and group k. Then, as the node i belongs to group k with high probability (ϕ_{ik} is close to 1), the feature l of node i, F_{il}, is more likely to be 1. By modeling group memberships using multiple Bernoulli random variables (instead of using multinomial distribution (Airoldi et al., 2007; Chang & Blei, 2009)), we achieve greater modeling flexibility which allows for making predictions about links given features and features given links. In Section 4, we empirically demonstrate that the *LMMG* outperforms traditional models on these tasks.

Moreover, if we divide the nodes of the network into two sets depending on the membership to group k, then we can discover how members of group k link to other members as well as non-members of group k, based on the structure of Θ_k. For example, when Θ_k has large values on diagonal entries like in Figure 3(a), members or non-members are likely to link among themselves, while there is low affinity for links between members and non-members. Figure 3(b) captures exactly the opposite behavior where links are most likely between members and non-members. While the core-periphery structure is captured by link-affinity matrix in Figure 3(c) where nodes that share group memberships (the "core") are most likely to link, while nodes in the periphery are least likely to link among themselves.

3. Inference, Estimation and Prediction

We now turn our attention to *LMMG* model estimation. Given a set of binary node features F and the network A, we aim to find node group memberships ϕ, parameters W of node feature model, and link-affinity matrices Θ.

3.1. Problem formulation

When the node features $F = \{F_{il} : i = 1,\cdots,N,\;\; l = 1,\cdots,L\}$ and the adjacency matrix $A \in \{0,1\}^{N\times N}$ are given, we aim to find the group memberships $\phi = \{\phi_{ik} : i = 1,\cdots,N,\;\; k = 1,\cdots,K\}$, the logistic model parameters $W = \{w_{lk} : l = 1,\cdots,L,\;\; k = 1,\cdots,K+1\}$, and the link-affinity matrices $\Theta = \{\Theta_k : k = 1,\cdots K\}$. We apply the maximum likelihood estimation, which finds the optimal values of ϕ, W, and Θ so that they maximize the likelihood $P(F, A, \phi | W, \Theta, \alpha)$, where $\alpha = \{(\alpha_{k1}, \alpha_{k2}) : k = 1,\cdots,K\}$ represents hyper parameters for the Beta prior distributions. In the end, we aim to solve

$$\max_{\phi, W, \Theta} \log P(F, A, \phi | W, \Theta, \alpha)\,. \tag{4}$$

Now we compute the objective function in the above optimization problem. Since the *LMMG* independently generates F and A given group memberships ϕ, we decompose the log-likelihood $\log P(F, A, \phi | W, \Theta, \alpha)$ as follows:

$$\begin{aligned} &\log P(F, A, \phi | W, \Theta, \alpha) \\ &= \log P(F|\phi, W) + \log P(A|\phi, \Theta) + \log P(\phi|\alpha)\,. \end{aligned} \tag{5}$$

Hence, to compute $\log P(F, A, \phi | W, \Theta, \alpha)$, we separately calculate each term of Equation (5). We obtain $\log P(\phi|\alpha)$ and $\log P(F|\phi, W)$ from Equations (1) and (2):

$$\begin{aligned} \log P(\phi|\alpha) &= \sum_{i,k} (\alpha_{k1} - 1) \log \phi_{ik} \\ &\qquad + \sum_{i,k} (\alpha_{k2} - 1) \log(1 - \phi_{ik}) \\ \log P(F|\phi, W) &= \sum_{i,l} F_{il} \log y_{il} + (1 - F_{il}) \log(1 - y_{il}) \end{aligned}$$

where y_{il} are defined in Equation (2).

With regard to the second term in Equation (5),

$$\log P(A|\phi, \Theta) = \log \sum_Z P(A|Z, \phi, \Theta) P(Z|\phi, \Theta) \tag{6}$$

for $Z = \{z_{ik} : i = 1, \cdots, N, \;\; k = 1, \cdots, K\}$. Note that A is independent of ϕ given Z. To exactly calculate $\log P(A|\phi, \Theta)$, we sum $P(A|Z, \Theta)P(Z|\phi)$ over every instance of Z given Θ and ϕ. However, this requires the sum over 2^{NK} instances. As this exact computation is infeasible, we approximate $\log P(A|\phi, \Theta)$ using its lower bound obtained by applying Jensen's Inequality to Equation (6):

$$\begin{aligned} \log P(A|\phi, \Theta) &= \log \mathbb{E}_{Z\sim\phi}\left[P(A|Z, \Theta)\right] \\ &\geq \mathbb{E}_{Z\sim\phi}\left[\log P(A|Z, \Theta)\right] . \end{aligned} \tag{7}$$

Now that we are summing up over N^2 terms, the computation of the lower bound is feasible. We thus maximize the lower bound $\mathcal{L}$ of the log-likelihood $\log P(A, F, \phi|W, \Theta, \alpha)$. To sum up, we aim to maximize

$$\min_{\phi, W, \Theta} -(\mathcal{L}_\phi + \mathcal{L}_F + \mathcal{L}_A) + \lambda|W|_1 \tag{8}$$

where $\mathcal{L}_\phi = \log P(\phi|\alpha), \mathcal{L}_F = \log P(F|\phi, W)$, and $\mathcal{L}_A = \mathbb{E}_{Z\sim\phi}\left[\log P(A|Z, W)\right]$. To avoid overfitting, we regularize the objective function by the L1-norm of W.

3.2. Parameter estimation

To solve the problem in Equation (8), we alternately update the group memberships ϕ, the model parameters W, and Θ. Once ϕ, W, and Θ are initialized, we first update the group memberships ϕ to maximize $\mathcal{L}$ with fixing W and Θ. We then update the model parameters W and Θ to minimize the function $(-\mathcal{L} + \lambda|W|_1)$ in Equation (8) by fixing ϕ. Note that $\mathcal{L}$ is decomposed into $\mathcal{L}_A$, $\mathcal{L}_F$, and $\mathcal{L}_\phi$. Therefore, when updating W and Θ given ϕ, we separately maximize the corresponding log-likelihoods $\mathcal{L}_F$ and $\mathcal{L}_A$. We repeat this alternate updating procedure until the solution converges. In the following we describe the details.

Update of group memberships ϕ. Now we focus on updating group memberships ϕ given the model parameters W and Θ. We use the coordinate ascent algorithm which updates each membership ϕ_{ik} by fixing the others so to maximize the lower bound $\mathcal{L}$. By computing the derivatives of $\mathcal{L}_\phi$, $\mathcal{L}_F$, and $\mathcal{L}_A$ we apply the gradient method to update each ϕ_{ik}:

$$\begin{aligned} \frac{\partial \mathcal{L}_\phi}{\partial \phi_{ik}} &= \frac{\alpha_{k1} - 1}{\phi_{ik}} - \frac{\alpha_{k2} - 1}{1 - \phi_{ik}} \\ \frac{\partial \mathcal{L}_F}{\partial \phi_{ik}} &= \sum_l (F_{il} - y_{il}) w_{lk} \\ \frac{\partial \mathcal{L}_A}{\partial \phi_{ik}} &= \mathbb{E}_{Z\sim\phi}\left[\sum_{j:A_{ij}=1} \frac{\partial \log p_{ij}}{\partial \phi_{ik}} + \sum_{j:A_{ij}=0} \frac{\partial \log(1 - p_{ij})}{\partial \phi_{ik}} \right. \\ &\quad \left. + \sum_{j:A_{ji}=1} \frac{\partial \log p_{ji}}{\partial \phi_{ik}} + \sum_{j:A_{ji}=0} \frac{\partial \log(1 - p_{ji})}{\partial \phi_{ik}} \right] \end{aligned} \tag{9}$$

where F_{il} is either 0 or 1, and y_{il} and p_{ij} is respectively defined in Equation (2) and (3). Due to the brevity, we describe the details of Equation (9) in the full version of this paper (Kim & Leskovec, 2012b). Hence, by adding up $\frac{\partial \mathcal{L}_\phi}{\partial \phi_{ik}}$, $\frac{\partial \mathcal{L}_F}{\partial \phi_{ik}}$, and $\frac{\partial \mathcal{L}_A}{\partial \phi_{ik}}$, we complete computing the derivative of the lower bound of log-likelihood $\frac{\partial \mathcal{L}}{\partial \phi_{ik}}$ and update the group membership ϕ_{ik} using the gradient method:

$$\phi_{ik}^{new} = \phi_{ik}^{old} + \gamma_\phi \left(\frac{\partial \mathcal{L}_A}{\partial \phi_{ik}} + \frac{\partial \mathcal{L}_F}{\partial \phi_{ik}} + \frac{\partial \mathcal{L}_A}{\partial \phi_{ik}} \right) \tag{10}$$

for a given learning rate γ_ϕ. By updating each ϕ_{ik} in turn by fixing the others, we can find the optimal group memberships ϕ given the model parameters W and Θ.

Update of node feature model parameters W. Now we update the parameters W of node feature model while keeping group memberships ϕ fixed. Note that given the group memberships ϕ the node feature model and the network model are independent of each other. Therefore, finding parameters W is identical to running the L1-regularized logistic regression given input ϕ and output F data as we penalize the objective function in Equation (8) on the L1 value of model parameters W. We basically use the gradient method to update W but make it sparse by applying the technique similar to *LASSO*:

$$\begin{aligned} \frac{\partial \mathcal{L}_F}{\partial w_{lk}} &= \sum_i (F_{il} - y_{il}) \phi_{ik} \\ w_{lk}^{new} &= w_{lk}^{old} + \gamma_F \frac{\partial \mathcal{L}_F}{\partial w_{lk}} - \lambda(k) \mathrm{Sign}(w_{lk}) \end{aligned} \tag{11}$$

if $w_{lk}^{old} \neq 0$ or $|\frac{\partial \mathcal{L}_F}{\partial w_{lk}}| > \lambda(k)$ where $\lambda(k) = \lambda$ for $k = 1, \cdots, K$ and $\lambda(K+1) = 0$ (*i.e.*, we do not regularize on the intercepts). γ_F is a constant learning rate. Furthermore, if w_{lk} crosses 0 while being updated, we assign 0 to w_{lk}, same as in *LASSO*. By this procedure, we can update the node feature model parameters W to maximize the lower bound of log-likelihood $\mathcal{L}$ as well as to maintain the small number of relevant groups for each node feature.

Update of network model parameters Θ. Next we focus on updating network model parameters, Θ, when the group memberships ϕ are fixed. Again, note that the network model is independent of the node feature model given the group memberships ϕ, so we do not need to consider $\mathcal{L}_\phi$ or $\mathcal{L}_F$. We thus update Θ to maximize $\mathcal{L}_A$ given ϕ using the gradient method.

$$\begin{aligned} \nabla_{\Theta_k} \mathcal{L}_A &\approx \nabla_{\Theta_k} \mathbb{E}_{Z\sim\phi}\left(\sum_{A_{ij}=1} \log p_{ij} + \sum_{A_{ij}=0} \log(1 - p_{ij}) \right) \\ \Theta_k^{new} &= \Theta_k^{old} + \gamma_A \nabla_{\Theta_k} \mathcal{L}_A \end{aligned}$$

for a constant learning rate γ_A. We explain the computation of $\nabla_{\Theta_k} \mathbb{E}_{Z\sim\phi} \log p_{ij}$ and $\nabla_{\Theta_k} \mathbb{E}_{Z\sim\phi} \log(1 - p_{ij})$ in detail in the full version of this paper (Kim & Leskovec, 2012b).

3.3. Prediction

With a fitted model, our ultimate goal is to make predictions about new data. In real-world application, the node features are often missing. Our approach is able to nicely handle such missing node features by fitting *LMMG* only to the observed features. In other words, when we update the group memberships ϕ or the feature model parameters W by the gradient method from Equation (9) and (11), we only average the terms corresponding to the observed data. For example, when there is missing feature data, Equation (9) can be converted into as:

$$\frac{\partial \mathcal{L}_F}{\partial \phi_{ik}} = \frac{\sum_{l:F_{il}\in\mathbb{O}} (F_{il} - y_{il}) w_{lk}}{\sum_{l:F_{il}\in\mathbb{O}} 1} \tag{12}$$

for the observed data $\mathbb{O}$.

Similarly, for link prediction we modify the model estimation method as follows. While updating the node feature model parameters W based on the features of all the nodes including a new node, we estimate the network model parameters Θ only on the observed network by holding out the new node. This way, the observed features naturally update the group memberships of a new node. We then predict the missing node features or network links by using the estimated group memberships and model parameters.

Source code of our algorithms is available at http://snap.stanford.edu.

4. Experiments

First, we run various prediction tasks: missing node feature prediction, missing link prediction, and supervised node classification. In all tasks our model outperforms natural baselines. Second, we qualitatively analyze the relationships between the node features and the network structure by a case study of a Facebook ego-network and show how the *LMMG* identifies useful and interpretable latent structures.

Datasets. For our experiments, we used the following datasets containing networks and node features.

- AddHealth (AH): School friendship network (458 nodes, 2,130 edges) with 35 school-related node features such as GPA, courses taken, and placement (Bearman et al., 1997).
- Egonet (EGO): Facebook ego-network of a particular user (227 nodes, 6,348 edges) and 14 binary features (*e.g.* same high school, same age, and sports club), manually assigned to each friend by the 'ego' user.
- Facebook100 (FB): Facebook network of Caltech (769 nodes, 33,312 edges) and 24 university-related node features like major, gender, and dormitory (Traud et al., 2011).
- WebKB (WKB): Hyperlinks between computer science webpages of Cornell University in the WebKB dataset (195 nodes, 304 edges). We use occurrences of 993 words as binary features (Craven et al., 1998).

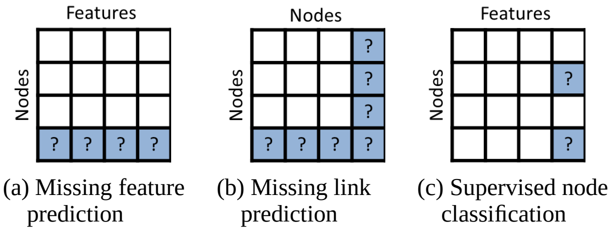

Features
Nodes
?
(a) Missing feature prediction
Nodes
Nodes
(b) Missing link prediction
Features
Nodes
(c) Supervised node classification

Figure 4. Three link and feature based predictive tasks.

We binarized discrete valued features (*e.g.* school year) based on whether the feature value is greater than the median value. For the non-binary categorical features (*e.g.* major), we used an indicator variable for each possible feature value. Some of these datasets are available at http://snap.stanford.edu.

Predictive tasks. We investigate the predictive performance of the *LMMG* based on three different tasks. We visualize them in Figure 4. Note that the column represents either features or nodes depending on the type of the task. For each matrix, given 0/1 values in the white area, we predict the values of the entries with question marks. First, assuming that all node features of a given node are completely missing, we predict all the features based on the links of the node (Figure 4(a)). Second, when all the links of a given node are missing, we predict the missing links by using the node feature information (Figure 4(b)). Last, we assume only few features of a node are missing and perform supervised classification of a specific node feature given all the other node features and the network (Figure 4(c)).

Baseline models. Next we introduce natural baseline as well as the state-of-the-art methods. First, for the most basic baseline, when predicting some missing value (node feature or link) of a given node, we average the corresponding values of all the other nodes and regard it as the probability of value 1. We refer to this algorithm as AVG. Second, as we can view each of the three prediction tasks as the classification task, we use Collective Classification (CC) algorithms that exploit both node features and network dependencies (Sen et al., 2008). For the local classifier of CC algorithms, we use Naive-Bayes (CC-N) as well as logistic regression (CC-L). We also compare the *LMMG* to the state of the art, Relational Topic Model (RTM) (Chang & Blei,

LL					
	AVG	CC-N	CC-L	RTM	*LMMG*
AH	-23.0	-17.6	-16.8	-63.4	**-15.6**
EGO	-5.4	-6.6	-5.1	-9.9	**-3.7**
FB	-8.7	-11.6	-8.9	-19.0	**-7.4**
WKB	-179.3	-186.8	-179.2	-336.8	**-173.6**
ACC					
	AVG	CC-N	CC-L	RTM	*LMMG*
AH	0.53	0.61	0.56	0.59	**0.64**
EGO	0.79	0.81	0.78	0.74	**0.86**
FB	0.77	0.76	0.75	0.77	**0.80**
WKB	0.88	0.88	0.89	0.88	**0.90**

Table 1. Prediction of missing node attributes. The *LMMG* performs the best in terms of the log-likelihood as well as the classification accuracy on the held-out data.

2009). We give further details about these models in the full version of this paper (Kim & Leskovec, 2012b).

Task 1: Predicting missing node features. First, we examine the task of predicting missing features of a node where features of other nodes and all the links are observed. We randomly select a node and remove all the feature values of that node and then try to recover them. We quantify the performance by using the log-likelihood of the true feature values over the estimated distributions as well as the predictive accuracy (the probability of correctly predicting the missing features) of each method.

Table 1 shows the results of the experiments by measuring the average of log-likelihood (LL) and prediction accuracy (ACC) for each algorithm and each dataset. We notice that *LMMG* model exhibits the best performance in the log-likelihood for all datasets. While CC-L in general performs the second best, our model outperforms it by up to 23%. The performance gain over the other models in terms of accuracy seems smaller when compared to the log-likelihood. However, *LMMG* model still predicts the missing node features with the highest accuracy on all the datasets.

In particular, the *LMMG* exhibits the most improvement in node feature prediction on the ego-network dataset (30% in LL and 7% in ACC) over the next best method. Here node features are derived by manually labeling community memberships of each person in the ego-network. Thus, a group of people in the network intrinsically share some node feature value (community membership). In this sense, the node features and the links in the ego-network are directly related to each other and our model successfully exploits this relationship to predict missing node features.

Task 2: Predicting missing links. Second, we also consider the task of predicting missing links of a specific node while the features of the node are given. Similarly to the previous task, we select a node at random, but here we remove all its links while observing its features. We then aim to recover the missing links. For evaluation, we use the log-likelihood (LL) of missing links as well as the area under the ROC curve (AUC) of missing link prediction.

LL					
	AVG	CC-N	CC-L	RTM	*LMMG*
AH	-40.2	-57.2	-38.9	-100.6	**-36.1**
EGO	-142.7	-134.3	-157.6	-149.9	**-125.9**
FB	**-320.8**	-330.7	-345.6	-359.1	-328.3
WKB	-54.2	-185.5	-39.6	-25.8	**-13.7**
AUC					
	AVG	CC-N	CC-L	RTM	*LMMG*
AH	0.51	0.69	0.39	0.56	**0.72**
EGO	0.61	**0.89**	0.55	0.49	**0.89**
FB	**0.73**	0.70	0.57	0.46	**0.73**
WKB	0.70	0.86	0.55	0.50	**0.89**

Table 2. Prediction of missing links of a node. The *LMMG* performs best in all but one case.

We give the experimental results for each dataset in Table 2. Again, the *LMMG* outperforms the baseline models in the log-likelihood except for the Facebook100 data. Interestingly, while RTM was relatively competitive when predicting missing features, it tends to fail predicting missing links, which implies that the flexibility of link-affinity matrices is needed for accurate modeling of the links.

We observe that Collective Classification methods look competetive in some performance metrics and datasets. For example, CC-N gives good results in terms of classification accuracy, and CC-L performs well in terms of the log-likelihood. As CC-N is a discriminative model, it does not perform well in missing link probability estimation. However, the *LMMG* is a generative model that produces a joint probability of node features and network links, so it is also very good at estimating missing links. Hence, in overall, the *LMMG* nicely exploits the relationship between the network structure and node features to predict missing links.

Task 3: Supervised node classification. Finally, we examine the performance on the supervised classification task. In many cases, we aim to classify entities (nodes) based on their feature values under the supervised setting. Here the relationships (links) between the entities are also provided. For this experiment, we hold out one feature of nodes as the output class, regarding all other features of nodes and the network as input data. We divide the nodes into a 70% training and 30% test set. Similarly, we measure the average of the log-likelihood (LL) as well as the average classification accuracy (ACC) on the test set.

We illustrate the performance of various models in Table 3. The *LMMG* model performs better than the other models in both the log-likelihood and the classification accuracy. It improves the performance by up to 20% in the log-likelihood and 5% in the classification accuracy. We also notice that exploiting the relationship between node

LL					
	AVG	CC-N	CC-L	RTM	*LMMG*
AH	-84.5	-486.6	-60.5	-236.0	**-55.3**
EGO	-24.8	-54.0	-22.2	-41.7	**-21.2**
FB	-97.6	-254.6	-79.2	-181.7	**-63.4**
WKB	-17.5	-254.6	-15.4	-193.6	**-15.0**

ACC					
	AVG	CC-N	CC-L	RTM	*LMMG*
AH	0.52	0.58	**0.63**	0.51	**0.63**
EGO	0.76	0.76	0.77	0.75	**0.79**
FB	0.69	0.71	**0.77**	0.72	**0.77**
WKB	0.82	0.81	0.84	0.84	**0.85**

Table 3. Supervised node classification. The *LMMG* gives the best performance on both metrics and all four datasets.

features and the global network structure improves the performance of supervised node classification compared to the models focusing on local network dependencies (*e.g.*, Collective Classification methods).

Case study: Analysis of a Facebook ego-network. Last we qualitatively analyze the Facebook ego-network example to provide insights into the relationship between node features and network structure. By investigating model parameters (W and Θ), we can find not only what features are important for each group but also how each group affects the link structure.

We begin by introducing the ego-user which we used to create a network between his Facebook friends. We asked our user to label each of his friends with a number of labels. He chose to use 14 different labels. They correspond to his high school (HS), undergraduate university (University), math olympiad camp (Camp), computer programming club (KProg) and work place (KComp) friends. The user also assigned labels to identify friends from his graduate program (CS) and university (ST), basketball (Basketball) and squash (Squash) clubs, as well as travel mates (Travel), summer internship buddies (Intern), family (Family) and age group (Age).

We fit the *LMMG* to the ego-network and each friend's labels, and obtained the model parameters W and Θ. We set the number of latent groups to 5 since the previous prediction tasks worked well when $K = 5$. In Table 4, for each of 5 latent groups, we represent the top 3 features with the largest absolute value of model parameter $|w_{lk}|$ and the corresponding link-affinity matrices Θ_k.

We begin by investigating the first group. The top three labels the most correlated to the first group are ST, Age, and Intern. However, notice that Intern is negatively correlated. This means that group 1 contains students from the same graduate school and age, but not people with whom our user worked together at the summer internship (even though they may be of the same school/age). We also note that Θ_1 exhibits homophily structure. From this we learn that summer interns, who met our Facebook user neither because of shared graduate school nor because of age, form a group within which people are densely connected. On the other hand, people of the same age at the same university also exhibit homophily, but are less densely connected with each other. Such variation in link density that depends on the group memberships agrees with our intuition. Those who worked at the same company actively interact with each other so almost everyone is linked in Facebook. However, as the group of people of the same university or age group is large and each pair of people in that group does not necessarily know each other, the link affinity in this group is naturally smaller than in the intern's group.

Similarly, groups 2 and 3 form the two sports groups (Basketball, Squash). People are connected densely within each of the groups, but less connected to the outside of the groups. Furthermore, we notice that those who graduated from the same high school (HS) as well as the same undergraduate school (University) form another community but the membership to high school is more important than to the undergraduate university (8.7 vs. 2.3).

Last, for groups 4 and 5, we note that the corresponding link-affinity matrices are nearly flat (*i.e.* values are nearly uniform). This implies that groups 4 and 5 are related to general node features. In this sense, we hypothesize that features like CS, family, math camp, and the company, have relatively little effect on the network structure.

5. Related Work and Discussion

The *LMMG* builds on previous research in machine learning and network analysis. Many models have been developed to explain network link structure (Airoldi et al., 2007; Hoff et al., 2002; Kemp et al., 2006; Leskovec et al., 2010) and extensions that incorporate node features have also been proposed (Getoor et al., 2001; Kim & Leskovec, 2011b; Taskar et al., 2003). However, these models do not consider latent groups and thus cannot provide the benefits of dimensionality reduction or produce interpretable clusters useful for understanding network community structure.

The *LMMG* provides meaningful clustering of nodes and their features in the network. Network models of similar flavor have been proposed in the past (Airoldi et al., 2007; Hoff et al., 2002; Kemp et al., 2006), and some even incorporate node features (Chang & Blei, 2009; Nallapati et al., 2008; Miller et al., 2009). However, such models have been mainly developed for document networks where they assume multinomial topic distributions for each word in the document. We extend this by learning a logistic model of occurrence of each feature based on node group memberships. We highlight this difference to previous models. For topic memberships. previous models use multinomial dis-

Group	Top 1	Top 2	Top 3	Link-affinity matrix
1	ST (9.0)	Age (4.5)	Intern (-3.7)	[0.67 0.08; 0.08 0.17]
2	HS (-8.7)	University (-2.3)	Basketball (2.2)	[0.26 0.18; 0.18 0.38]
3	University (-7.1)	KorST (-2.6)	Squash (2.2)	[0.22 0.23; 0.23 0.32]
4	CS (7.3)	Family (7.0)	Camp (6.9)	[0.25 0.24; 0.24 0.27]
5	KComp (5.2)	KorST (4.4)	Intern (-3.8)	[0.29 0.22; 0.22 0.27]

Table 4. Logistic model parameter values of top 3 features and the link-affinity matrix associated with each group in the ego-network.

tributions, where a node has a mass of 1 to split among various topics. In contrast, in the *LMMG*, a node can belong to multiple topics at once without any constraint.

While previous work tends to explore only the network or only the features, the *LMMG* jointly models both so that it can make predictions of one given the other. The *LMMG* models the interaction between links and group memberships via link-affinity matrices which provide great flexibility and interpretability of obtained groups and interactions.

The *LMMG* is a new probabilistic model of links and nodes in networks. It can be used for link prediction, node feature prediction and supervised node classification. We have demonstrated qualitatively and quantitatively that the *LMMG* proves useful for analyzing network data. The *LMMG* significantly improves on previous models, integrating both node-specific information and link structure to give better predictions.

Acknowledgments

Myunghwan Kim was supported by the Kwanjeong Educational Foundation fellowship. This research has been supported in part by NSF CNS-1010921, IIS-1016909, IIS-1149837, IIS-1159679, Albert Yu & Mary Bechmann Foundation, Boeing, Allyes, Samsung, Yahoo, Alfred P. Sloan Fellowship and the Microsoft Faculty Fellowship.

References

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. Mixed membership stochastic blockmodels. *JMLR*, 9: 1981–2014, 2007.

Backstrom, L. and Leskovec, J. Supervised random walks: Predicting and recommending links in social networks. In *WSDM*, 2011.

Bearman, P. S., Jones, J., and Udry, J. R. The national longitudinal study of adolescent health: Research design. *http://www.cpc.unc.edu/addhealth*, 1997.

Chang, J. and Blei, D. M. Relational topic models for document networks. In *AISTATS*, 2009.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S. Learning to extract symbolic knowledge from the world wide web. In *AAAI '98*, 1998.

Getoor, L., Segal, E., Taskar, B., and Koller, D. Probabilistic models of text and link structure for hypertext classification. In *IJCAI Workshop on Text Learning: Beyond Supervision*, 2001.

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

Kemp, C., Tenebaum, J. B., and Griffiths, T. L. Learning systems of concepts with an infinite relational model. In *AAAI '06*, 2006.

Kim, M. and Leskovec, J. Network completion problem: Inferring missing nodes and edges in networks. In *SDM*, 2011a.

Kim, M. and Leskovec, J. Modeling social networks with node attributes using the multiplicative attribute graph model. In *UAI*, 2011b.

Kim, M. and Leskovec, J. Multiplicative attribute graph model of real-world networks. *Internet Mathematics*, 8 (1-2):113–160, 2012a.

Kim, M. and Leskovec, J. Latent multi-group membership graph model. *arXiv:1205.4546v1*, 2012b.

Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., and Ghahramani, Z. Kronecker Graphs: An Approach to Modeling Networks. *JMLR*, 11:985–1042, 2010.

Miller, K. T., Griffiths, T. L., and Jordan, M. I. Nonparametric latent feature models for link prediction. In *NIPS '09*, 2009.

Nallapati, R., Ahmed, A., Xing, E., and Cohen, W. W. Joint latent topic models for text and citations. In *KDD*, 2008.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., and Eliassi-rad, T. Collective classification in network data. *AI Magazine*, 29(3), 2008.

Taskar, B., Wong, M. F., Abbeel, P., and Koller, D. Link prediction in relational data. In *NIPS*, 2003.

Traud, Amanda L., Mucha, Peter J., and Porter, Mason A. Social structure of facebook networks. *arXiv:1102.2166v1*, 2011.

A. Mathematical Details

A.1. Update of Group Membership ϕ

In Equation (9), we proposed the gradient ascent method which updates each group membership ϕ_{ik} to maximize the lower bound of log-likelihood $\mathcal{L}$. To complete its computation, we further take a look at $\frac{\partial \mathbb{E}_{Z\sim\phi}\log p_{ij}}{\partial \phi_{ik}}$ and $\frac{\partial \mathbb{E}_{Z\sim\phi}\log(1-p_{ij})}{\partial \phi_{ik}}$ in detail. Then, we can also compute $\frac{\partial \mathbb{E}_{Z\sim\phi}\log p_{ji}}{\partial \phi_{ik}}$ and $\frac{\partial \mathbb{E}_{Z\sim\phi}\log(1-p_{ji})}{\partial \phi_{ik}}$ in the same way.

First, we calculate the derivative of expected log-likelihood for edges, $\mathbb{E}_{Z\sim\phi}\log p_{ij}$. When all the group memberships except for ϕ_{ik} are fixed, we can derive $\frac{\partial \mathbb{E}_{Z\sim\phi}\log p_{ij}}{\partial \phi_{ik}}$ from definition of p_{ij} in Equation (3) as follows:

$$\begin{aligned}\frac{\partial \mathbb{E}_{Z\sim\phi}\log p_{ij}}{\partial \phi_{ik}} &= \frac{\partial}{\partial \phi_{ik}}\mathbb{E}_{Z\sim\phi}\left[\sum_{k'}\log\Theta_{k'}[z_{ik'}, z_{jk'}]\right]\\ &= \sum_{k'}\left[\frac{\partial}{\partial \phi_{ik}}\mathbb{E}_{Z\sim\phi}\log\Theta_{k'}[z_{ik'}, z_{jk'}]\right]\end{aligned} \tag{13}$$

Here we use the following property. Since z_{ik} is an independent Bernoulli random variable with probability ϕ_{ik}, for any function $f : \{0,1\}^2 \to \mathbb{R}$,

$$\begin{aligned}\mathbb{E}_{Z\sim\phi}f(z_{ik}, z_{jk}) &= \phi_{ik}\phi_{jk}f(1,1) + \phi_{ik}(1-\phi_{jk})f(1,0)\\ &+(1-\phi_{ik})\phi_{jk}f(0,1) + (1-\phi_{ik})(1-\phi_{jk})f(0,0)\,.\end{aligned} \tag{14}$$

Hence, by applying Equation (14) to (13), we obtain

$$\begin{aligned}&\frac{\partial \mathbb{E}_{Z\sim\phi}\log p_{ij}}{\partial \phi_{ik}} = \frac{\partial}{\partial \phi_{ik}}\mathbb{E}_{Z\sim\phi}\log\Theta_k[z_{ik}, z_{jk}]\\ &= \phi_{jk}\log\Theta_k[1,1] + (1-\phi_{jk})\log\Theta_k[1,0]\\ &\quad - \phi_{jk}\log\Theta_k[0,1] - (1-\phi_{jk})\log\Theta_k[0,0]\,.\end{aligned} \tag{15}$$

Next, we compute the derivative of expected log-likelihood for unlinked node pairs, *i.e.* $\mathbb{E}_{Z\sim\phi}\log(1-p_{ij})$. Here we approximate the computation using the Taylor's expansion, $\log(1-x) \approx -x - 0.5x^2$ for small x:

$$\frac{\partial \mathbb{E}_{Z\sim\phi}\log(1-p_{ij})}{\partial \phi_{ik}} \approx -\frac{\partial \mathbb{E}_{Z\sim\phi}p_{ij}}{\partial \phi_{ik}} - 0.5\frac{\partial \mathbb{E}_{Z\sim\phi}p_{ij}^2}{\partial \phi_{ik}}\,.$$

To compute $\frac{\partial \mathbb{E}_{Z\sim\phi}p_{ij}}{\partial \phi_{ik}}$,

$$\begin{aligned}&\frac{\partial \mathbb{E}_{Z\sim\phi}p_{ij}}{\partial \phi_{ik}}\\ &= \frac{\partial}{\partial \phi_{ik}}\mathbb{E}_{Z\sim\phi}\prod_{k'}\Theta_{k'}[z_{ik'}, z_{jk'}]\\ &= \frac{\partial}{\partial \phi_{ik}}\mathbb{E}_{Z\sim\phi}\Theta_k[z_{ik}, z_{jk}]\prod_{k'\neq k}\Theta_{k'}[z_{ik'}, z_{jk'}]\\ &= \prod_{k'\neq k}\mathbb{E}_{Z\sim\phi}\Theta_{k'}[z_{ik'}, z_{jk'}]\frac{\partial}{\partial \phi_{ik}}\mathbb{E}_{Z\sim\phi}\Theta_k[z_{ik}, z_{jk}]\,.\end{aligned}$$

By Equation (14), each $\mathbb{E}_{Z\sim\phi}\Theta_k[z_{ik}, z_{jk}]$ and its derivative can be obtained. Similarly, we can calculate $\frac{\partial \mathbb{E}_{Z\sim\phi}p_{ij}^2}{\partial \phi_{ik}}$, so we complete the computation of $\frac{\partial \mathbb{E}_{Z\sim\phi}\log(1-p_{ij})}{\partial \phi_{ik}}$.

As we attain $\frac{\partial \mathbb{E}_{Z\sim\phi}\log p_{ij}}{\partial \phi_{ij}}$ and $\frac{\partial \mathbb{E}_{Z\sim\phi}\log(1-p_{ij})}{\partial \phi_{ij}}$, we eventually calculate $\frac{\partial \mathcal{L}_A}{\partial \phi_{ik}}$. Hence, by adding up $\frac{\partial \mathcal{L}_\phi}{\partial \phi_{ik}}$, $\frac{\partial \mathcal{L}_F}{\partial \phi_{ik}}$, and $\frac{\partial \mathcal{L}_A}{\partial \phi_{ik}}$, we complete computing the derivative of the lower bound of log-likelihood $\frac{\partial \mathcal{L}}{\partial \phi_{ik}}$:

$$\frac{\partial \mathcal{L}}{\partial \phi_{ik}} = \frac{\partial \mathcal{L}_A}{\partial \phi_{ik}} + \frac{\partial \mathcal{L}_F}{\partial \phi_{ik}} + \frac{\partial \mathcal{L}_A}{\partial \phi_{ik}}\,.$$

A.2. Update of MAG Model Parameters Θ

Next we focus on the update of parameters of the network model, Θ, where the group membership ϕ is fixed. Since the network model is independent of the node attribute model given the group membership ϕ, we do not need to consider $\mathcal{L}_\phi$, $\mathcal{L}_F$, or $|W|_1$. We thus update Θ to maximize only $\mathcal{L}_A$ given ϕ using the gradient method.

As we previously did in computing $\frac{\partial \mathcal{L}_A}{\partial \phi_{ik}}$ by separating edge and non-edge terms, we compute each $\frac{\partial \mathcal{L}_A}{\partial \Theta_k[x_1,x_2]}$ for $k = 1, \cdots, K$ and $x_1, x_2 \in \{0,1\}$. To describe mathematically,

$$\begin{aligned}\frac{\partial \mathcal{L}_A}{\partial \Theta_k[x_1,x_2]} &= \sum_{A_{ij}=1}\frac{\partial \mathbb{E}_{Z\sim\phi}\log p_{ij}}{\partial \Theta_k[x_1,x_2]}\\ &+ \sum_{A_{ij}=0}\frac{\partial \mathbb{E}_{Z\sim\phi}\log(1-p_{ij})}{\partial \Theta_k[x_1,x_2]}\,.\end{aligned} \tag{16}$$

Now we compute each term in the above calculation by the definition of p_{ij}. First, we compute the former term by using Equation (14) For instance,

$$\frac{\partial \mathcal{L}_A}{\partial \Theta_k[0,1]} = (1-\phi_{ik})\phi_{jk}\frac{\partial\log\Theta_k[0,1]}{\partial \Theta_k[0,1]} = \frac{(1-\phi_{ik})\phi_{jk}}{\Theta_k[0,1]}\,.$$

Hence, we can properly compute Equation (16) depending on the values of x_1 and x_2.

Second, we use the same Taylor's expansion technique for the latter term in Equation (16) as follows:

$$\frac{\partial \mathbb{E}_{Z\sim\phi}\log(1-p_{ij})}{\partial \Theta_k[x_1,x_2]} \approx \frac{\partial}{\partial \Theta_k[x_1,x_2]}\mathbb{E}_{Z\sim\phi}\left(-p_{ij} - 0.5p_{ij}^2\right)\,.$$

Similarly to $\frac{\mathbb{E}_{Z\sim\phi}p_{ij}}{\partial \phi_{ik}}$, $\frac{\mathbb{E}_{Z\sim\phi}p_{ij}}{\partial \Theta_k[x_1,x_2]}$ is computed by

$$\prod_{k'\neq k}\mathbb{E}_{Z\sim\phi}\Theta_{k'}[z_{ik'}, z_{jk'}]\frac{\partial}{\partial \Theta_k[x_1,x_2]}\mathbb{E}_{Z\sim\phi}\Theta_k[z_{ik}, z_{jk}]$$

where each term is obtained by Equation (14). Similarly, we compute $\frac{\mathbb{E}_{Z\sim\phi}p_{ij}^2}{\partial \Theta_k[x_1,x_2]}$ so that we can obtain $\frac{\partial \mathcal{L}_A}{\partial \Theta_k[x_1,x_2]}$.

B. Implementational Details

B.1. Initialization

Since the objective function in Equation (8) is non-convex, the final solution might be dependent on the initial values of ϕ, W, and Θ. For reasonable initialization, as the node attributes F are given, we run the Singular Vector Decomposition (SVD) by regarding F as an $N \times L$ matrix and obtain the singular vectors corresponding to the top K singular values. By taking the top K components, we can approximate the node attributes F over K latent dimensions. We thus assign the l-th entry of the k-th right singular vectors multiplied by the k-th singular value into w_{lk} for $l = 1, \cdots, L$ and $k = 1, \cdots, K$. We also initialize each group membership ϕ_{ik} based on the i-th entry of the k-th left singular vectors. This approximation can in particular provide good enough initial values when the top K singular values dominate the others. In order to obtain the sparse model parameter W, we reassign 0 to w_{lk} of small absolute value such that $|w_{lk}| < \lambda$.

Finally, to initialize the link-affinity matrices Θ, we introduce the following way. When initializing the k-th link-affinity matrix Θ_k, we assume that the group other than group k has nothing to do with network structure, *i.e.* every entry in the other link-affinity matrices has the equal value. Then, we compute the ratio between entries $\Theta_k[x_1, x_2]$ for $x_1, x_2 \in \{0, 1\}$ as follows:

$$\Theta_k[x_1, x_2] \propto \sum_{i,j:A_{ij}=1} \mathbb{E}_{Z\sim\phi} P[z_{ik} = x_1, z_{ik} = x_2]$$

As the group membership ϕ is initialized above and z_{ik} and z_{jk} are independent of each other, we are able to compute the ratio between entries of Θ_k. After computing the ratio between entries for each link-affinity matrix, we adjust the scale of the link-affinity matrices so that the expected number of edges in the MAG model is equal to the number of edges in the given network, *i.e.* $\sum_{i,j} p_{ij} = \sum_{i,j} A_{ij}$.

B.2. Selection of the Number of Groups K

Another issue in fitting the *LMMG* to the given network and node feature data is to determine the number of groups, K. We can find the insight about the value of K from the MAG model. It has been already proved that, in order for the MAG model to reasonably represent the real-world network, the value of K should be in the order of $\log N$ where N represents the number of nodes in the network (Kim & Leskovec, 2012a). Since in the *LMMG* the network links are modeled similarly to the MAG model, the same argument on the number of groups K still holds.

However, the above argument cannot determine the specific value of K. To select one value of K, we use the cross-validation method as follows. For instance, suppose that we aim to predict all the features of a node where its links to the other nodes are fully observed (Task 1 in Section 4). While holding out the test node, we can set up the same prediction task in a way that we select one at random from the other nodes (training nodes) and regard it as the validation test node. We then perform the missing node feature prediction on this validation node and obtain the log-likelihood result. By running this procedure with varying the validation test node, we can attain the average log-likelihood on the missing node features given the specific value of K (*i.e.* N-fold cross-validation). Finally, we compare the average log-likelihood values according the value of K and pick up the best one to maximize the log-likelihood. This method can be done by the other prediction tasks, missing link prediction and supervised node classification.

B.3. Baseline Models

Here we briefly describe how we implemented each baseline method depending on the type of prediction task.

AVG. In this baseline method, we regard each l-th node feature and a link to the i-th node as an independent random variable, respectively. In other words, we assume that missing node features or links do not depend on each other. Hence, we predict the l-th missing node feature by finding the probability that the l-th node feature of all the other nodes have value 1. We then regard the found probability as that of the missing l-th node feature taking value 1.

Similarly, when we predict missing links (in particular, the link to the i-th node) of a given node, we average the probability that all the other nodes are linked to the i-th node and take it as the probability of link from the given node to the i-th node (*i.e.* preferential attachment).

CC-N. For this method, we basically use the Naive-Bayes method using node features of each node as well as those of neighboring nodes. To represent each node feature of neighboring nodes by a single value, we select the majority value (either 0 or 1) from the neighbors' feature values.

However, we cannot use the node features when predicting all the node features of a given node. Furthermore, the node features of neighboring nodes are unattainable when we predict missing links. Therefore, depending on the type of prediction task, we exploit only achievable information among node features and those of neighboring nodes.

CC-L. We employ the similar approach to the CC-N. However, here we use the logistic regression rather than the Naive-Bayes and average the feature values of neighboring nodes rather than pick up the majority value.

RTM. We use the lda-R package to run RTM (*http://cran.r-project.org/web/packages/lda/index.html*).