Big Data Research ••• (••••) •••–•••

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks

Qingyao Wu [a], Jian Chen [a,*], Shen-Shyang Ho [b], Xutao Li [b], Huaqing Min [a], Chao Han [a]

[a] School of Software Engineering, South China University of Technology, China
[b] School of Computer Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 19 January 2015
Received in revised form 6 April 2015
Accepted 6 April 2015
Available online xxxx

Keywords:
Collective classification
Generative model
Semi-supervised learning
Multi-label learning
Large-scale sparsely labeled networks

ABSTRACT

The problem of *collective classification* (CC) for large-scale network data has received considerable attention in the last decade. Enabling CC usually increases accuracy when given a fully-labeled network with a large amount of labeled data. However, such labels can be difficult to obtain and learning a CC model with only a few such labels in large-scale sparsely labeled networks can lead to poor performance. In this paper, we show that leveraging the unlabeled portion of the data through *semi-supervised collective classification* (SSCC) is essential to achieving high performance. First, we describe a novel data-generating algorithm, called *generative model with network regularization* (GMNR), to exploit both labeled and unlabeled data in large-scale sparsely labeled networks. In GMNR, a network regularizer is constructed to encode the network structure information, and we apply the network regularizer to smooth the probability density functions of the generative model. Second, we extend our proposed GMNR algorithm to handle network data consisting of multi-label instances. This approach, called the *multi-label regularized generative model* (MRGM), includes an additional label regularizer to encode the label correlation, and we show how these smoothing regularizers can be incorporated into the objective function of the model to improve the performance of CC in multi-label setting. We then develop an optimization scheme to solve the objective function based on EM algorithm. Empirical results on several real-world network data classification tasks show that our proposed methods are better than the compared collective classification algorithms especially when labeled data is scarce.

1. Introduction

Networks have become ubiquitous in many application domains such as Internet, social, economical and scientific fields. Researchers in these fields have shown that systems of different nature can be represented as network data for which instances are interrelated. For example, web pages are connected to each other by hyperlinks, and telephone accounts are linked by calls. Generally, network data contain nodes (instances) interconnected with each other by edges reflects the relation or dependence between the nodes. Information on the nodes is provided as a set of attribute features (e.g., words present in the web page). Such network data are obviously not independent and identically distributed, and the class membership of an instance may influence the class membership of a related instance. Furthermore, many network data are large-scale and often involve the scenario where each node can be assigned a set of multiple labels simultaneously.

The problem of learning from large-scale network data is a challenging issue that has attracted growing attention from both academia and industry [1–3] due to its importance of related applications, ranging from web page classification to spatial data analysis and social network analysis. The *collective classification* (CC) is a task to jointly classifying interrelated instances in the network [4]. Enabling CC usually improves the performance of predictive models on network data as inference outcome for one instance can be used to improve inferences on related instances. However, such a performance improvement usually requires a fully-labeled network which contains a sufficiently large amount of labeled instances. For many large-scale network data, it is extremely expensive and time-consuming to obtain such labels especially when each instance has multiple class labels. In particular, the number of possible label assignments for an instance is exponential to the number of possible labels in a multi-label setting, which is extremely large even with a small number of possible labels. On the other hand, there are often large amount of unlabeled data available in the networks. Hence, it is of interest to develop learning algorithms that are able to utilize the large amount of unlabeled data together with the limited amount of labeled data in the large-

* Corresponding author. Tel.:/fax: +86 020 39380295.
E-mail address: ellachen@scut.edu.cn (J. Chen).

http://dx.doi.org/10.1016/j.bdr.2015.04.002
2214-5796/

scale sparsely labeled network data to avoid the expensive date-labeling effort and to enhance the learning performance.

In this paper, we study the problem of *semi-supervised collective classification* (SSCC) when one is given only with limited number of labeled data, which is common case in large-scale networks. Recently, various researchers have considered to examine the SSCC task using some forms of semi-supervised learning to improve the performance of CC [5,6]. It has been shown that leveraging the unlabeled portion of the data is essential to achieving high performance. The main aim of this paper is to find a generative representation for network data classification by exploiting information from both labeled and unlabeled data. To achieve this, we propose a new data generative algorithm, called *generative model with network regularization* (GMNR), based on the *probabilistic latent semantic analysis* (PLSA) method, and incorporate the network structure into it. In GMNR, a network regularizer is constructed to encode the network structure, and we apply the network regularizer to smooth the label probability distributions of the generative model. We find that the GMNR method is able to achieve a robust classification performance even in the paucity of labeled data.

Furthermore, we extend the GMNR method to the multi-label learning setting such that instances of the network data have multiple class labels. The new algorithm, called *multi-label regularized generative model* (MRGM) utilizes an additional label regularizer to explicitly encode the label correlation. This approach is able to capture the knowledge of the underlying network structure and the label correlation observed in the data to smooth the label probability density functions when learning the generative model. As a result, the predictions ensure consistency among interlinked instances and related labels. Intuitively, an instance connected to neighbors with high probabilities of related class labels also has a high chance for these class labels. In summary, the main contributions of this paper are as follows.

- A framework that utilizes the generative model that takes into account the network structure and label correlation for the collective classification problem in a semi-supervised and multi-label learning setting;
- An effective *expectation–maximization* (EM) algorithm to solve the maximum likelihood estimation problem in the proposed methods, and to compute the label probability distributions of the instances for classification;
- A theoretical discussion on the convergence of the proposed algorithm using an auxiliary function similar to that used in [7].
- An extensive statistical evaluation of the effectiveness of the proposed approach using various real-world network datasets.

This paper extends our preliminary work [8] which considers single-label learning problem in CC. In this paper, we focus on the multi-label learning problem in semi-supervised CC. We propose a novel data generative model which is able to exploit the network information and label correlation simultaneously for handling network data consisting of multi-label instances. Both of this work and the preliminary work are based on the PLSA data generative model, however, this paper significantly extends and upgrades the work presented there. The extensions and differences of this work and the preliminary work are summarized as follows:

1. Motivation of multi-label collective classification problem and collective inference techniques from the semi-supervised learning perspective is given.
2. An extensive discussion of the related work, including collective classification, semi-supervised learning, multi-label learning, as well as the PLSA model, is given.
3. We consider the CC task in a multi-label formulation, and extend the PLSA model for multi-label learning via incorporating a novel label regularizer into the model for smoothing the resulting label probability.
4. A new thresholding method for separating the relevant and irrelevant labels for a given multi-label instance is presented.
5. Additional experiments using new multi-label data networks are conducted. Experimental results with the Friedman and Nemenyi tests to assess the statistical significance of the differences in performance are reported.
6. A theoretical discussion on the convergence properties of the proposed method is given.

The rest of the paper is organized as follows. Section 2 describes the background and the related work. Section 3 presents the proposed methodology and its derivation in detail. Section 4 discusses the datasets, the experimental setup and experimental results. Finally, conclusions are drawn in Section 5.

2. Related work

2.1. Mining network data

Numerous approaches have been designed for learning from network data and label predicting for the unlabeled nodes. These approaches have been mainly studied in the research fields of collective inference, active inference, semi-supervised learning and multi-label learning. Details on these related works are described below.

Macskassy and Provost [9] provide a brief review of the previous work of collective classification in network data. Generally, the collective classification methods can be categorized into three groups: local classifier-based methods, global formulation-based methods, and relational-only methods. i) A local classifier-based method is based on an iterative process. The local classifier is trained for prediction using the attribute features derived from the content and additional relational features by aggregating the labels of neighbors. One example is the iterative classification algorithm (ICA) [4] which has been reported to be a fairly accurate method with robust performance to different network datasets. Gibbs sampling [1] is further used in the ICA framework to enrich the statistical properties of the algorithm. In recent years, there is a lot of work proposed to use a similar schema as ICA but with different local classifiers or different scenarios [1,10]. ii) A global method trains a classifier to optimize a global objective function for prediction. It is often based on a graphical model such as the loopy belief propagation of the relaxation labeling [10]. iii) A relational-only method uses only relational information for classification. Typically, the algorithm computes a new label distribution for an instance by averaging the current distributions of its neighbors. Weighted-vote relational neighbor with relaxation labeling (wvRN+RL) [9] is one such method. Recently, Macskassy and Provost [9] show that wvRN+RL performed very well in some cases. In fact, it should be considered as a baseline for CC evaluations. Sen et al. [2] provide an empirical study to analyze the strengths and weaknesses of different CC methods. One of the major disadvantages of these CC methods is that they are mainly studied in the scenario where there are a large amounts of labeled data in the network. However, it is difficult and time-consuming to acquiring such labels in many practical applications. On the other hand, there are usually large amount of unlabeled data available. As pointed out in [5], when the labeled data are limited, the performance of collective classification may be degraded due to the insufficient number of labeled neighbors [11].

In response, recent studies have examined semi-supervised collective classification methods on sparsely-labeled networks, using

Please cite this article in press as: Q. Wu et al., Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.04.002

some semi-supervised learning techniques to leverage the unlabeled data. For instance, McDowell and Aha [5] propose a semi-supervised ICA (semiICA) algorithm using a hybrid regularization to boost the performance of the ICA algorithm. McDowell and Aha [6] show that utilizing neighbor attributes are often more useful than collective classification based on neighbor labels when the network is sparsely labeled. They introduce a new method that enables discriminative classifiers to be used with neighbor attributes. Shi et al. [12] proposed a label propagation method with latent graph (LNP) constructed from the original network by adding various types of latent links including k-step links, label links, structure similarity links and attribute similarity links. Gallagher et al. [13] also propose a method adding edges to a sparsely-labeled network to improve classification performance. Bilgic et al. [14] provide an alternative solution to overcome the label sparsity issue using active learning approaches to take advantage of network structure. Kong et al. [15] recently propose the iterative classification of multiple labels (ICML) algorithm using the ICA scheme to handle multi-label network data. The main different is that label co-occurrences are considered as additional features in the model for multi-label learning. Specifically, each instance is represented as a feature vector in addition to the attribute features, label frequency of the neighbors, and label correlation features.

2.2. Probabilistic latent semantic analysis

Probabilistic latent semantic analysis (PLSA) [16] is one of the most popular statistical topic modeling approaches for the analysis of co-occurrence data, which has many applications, such as information retrieval and natural language processing. PLSA was originally developed for text data analysis where a document is represented as term frequencies. Therefore, we use text analysis to explain the model. Given a collection of co-occurrences (w, d) of document $d \in \mathcal{D} = \{d_1, \cdots, d_N\}$ and word $w \in \mathcal{W} = \{w_1, \cdots, w_M\}$. The key idea of PLSA is to introduce an unobserved topic variable $z \in \mathcal{Z} = \{z_1, \cdots, z_K\}$ associated with each observation so that documents and words are rendered conditionally independent on the state of the associated topic variable. One may define a generative model for word/document co-occurrences in the following way:

1. Select a document d_i with probability $P(d_i)$,
2. Pick a latent topic z_k with probability $P(z_k|d_i)$,
3. Generate a word w_j with probability $P(w_j|z_k)$.

A joint probability model over $\mathcal{D} \times \mathcal{W}$ is defined as follows

$$P(d_i, w_j) = P(d_i)P(w_j|d_i)$$

$$P(w_j|d_i) = \sum_{k=1}^{K} P(w_j|z_k)P(z_k|d_i).$$

A maximum likelihood formulation of the PLSA data generative model can be defined as

$$\begin{aligned}\mathcal{L} &= \sum_{i=1}^{N}\sum_{j=1}^{M} n(d_i, w_j) \log P(d_i, w_j) \\ &\propto \sum_{i=1}^{N}\sum_{j=1}^{M} n(d_i, w_j) \log \sum_{k=1}^{K} P(w_j|z_k)P(z_k|d_i) \end{aligned} \tag{1}$$

where N and M are the number of documents and words respectively, and $n(x_i, w_j)$ indicates the frequency of term w_j occurring in document d_i.

There are $NK + MK$ parameters $\{P(z_j|d_k), P(w_k|z_i)\}$ which need to be estimated in the PLSA generative model. The expectation maximization (EM) algorithm [17] is used for maximum likelihood estimation in the model. EM performs the following two steps alternatively until a termination condition is met: (i) Expectation step computes posterior probabilities for latent variables based on current estimates of the parameters. (ii) Maximization step updates the parameters estimation by maximizing the expected complete data log-likelihood derived in E-step. New parameter-estimates are then used in the next E-step.

3. Methodology

In the SSCC problem, we consider the given network data as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{Y})$. $\mathcal{V}$ is a set of nodes $\{v_1, \ldots, v_N\}$. $\mathcal{E}$ is the set of edges. $\mathcal{X} = \{x_1, x_2, \cdots, x_N\} \subset \mathbb{R}^M$ consists of N instances where $x_i = [w_{i1}, \ldots, w_{iM}]$ is the M-dimensional attribute feature vector for a node $v_i \in \mathcal{V}$. $\mathcal{Y}$ contains the set of label set Y_i corresponding to instance x_i. $\{c_1, c_2, \cdots, c_K\}$ is the set of K possible class labels. Each $Y_i = [Y_{i1}, \ldots, Y_{iK}]^T \in \{0, 1\}^K$ such that $Y_{ik} = 1$ means that x_i is associated with c_l, otherwise $Y_{ik} = 0$. Given a set of labeled nodes, the task is to predict the class labels of the remaining unlabeled nodes. We assume the label information of the first n' instances $\{(x_i, Y_i)\}_{i=1}^{n'}$ is available, the rest n'' instances $\{x_i\}_{i=n'+1}^{n'+n''}$ are unlabeled data, and we have $N = n' + n''$. In a sparsely-labeled network, which is the primary interest in this study, there are only a limited number of labeled nodes, i.e. $n' \ll n''$, which makes the task challenging.

3.1. Generative model with network regularization (GMNR)

To tackle the label deficiency problem, we propose to use both the labeled and unlabeled portion of the data, and judiciously integrate them to learn a generative model. We model the learning problem as a data generating process utilizing PLSA to compute the probabilities of the classes to the instances. Generally, PLSA is an unsupervised learning algorithm, if the label information is provided, we can use such supervised knowledge to guide the learning process. In this subsection, we introduce our *generative model with network regularization* (GMNR) algorithm that generates a label probability distribution for a given instance such that its relevant labels receive higher probabilities than the irrelevant labels.

Given a collection of co-occurrences (w, x) of node instances $x \in \mathcal{X} = \{x_1, \cdots, x_N\}$ and data features $w \in \mathcal{W} = \{w_1, \cdots, w_M\}$. We specify that the class label corresponds to the latent topics in PLSA. That is, each latent topic z_k represents a class of interest c_k $(1 \le k \le K)$. With the given label information, the PLSA generative model can be written in the following way.

$$\begin{aligned}\mathcal{L} &= \sum_{i=1}^{N}\sum_{j=1}^{M} n(x_i, w_j) \log \sum_{k=1}^{K} P(w_j|c_k)P(c_k|x_i) \\ &= \sum_{i=1}^{n'}\sum_{j=1}^{M} n(x_i, w_j) \log \sum_{k=1}^{K} P(w_j|c_k)P(c_k|x_i) \\ &\quad + \sum_{i=n'+1}^{n'+n''}\sum_{j=1}^{M} n(x_i, w_j) \log \sum_{k=1}^{K} P(w_j|c_k)P(c_k|x_i) \end{aligned} \tag{2}$$

The above objective function is divided into two terms depending on the availability of labels for the data. The label information is encoded in the first term of Eq. (2) and applied to the label probability distribution $P(c|x)$ as follows.

$$P(c_k|x_i) = \begin{cases} 1/l_x & \text{if } x_i \text{ is labeled with } c_k \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Please cite this article in press as: Q. Wu et al., Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.04.002

where l_x is the number of classes assigned to the labeled instance x_i. $P(c_k|x_i)$ is filled with either $1/l_x$ or 0, depending on whether x_i belongs to class c_k or not.

The second term corresponds to the unlabeled data. Our objective is to seek a good estimation of the label probability distribution $P(c|x)$ for classification prediction. By using the above constraints to learn the PLSA generative model, the supervised knowledge of labeled data can be effectively encoded in the model. However, PLSA assumes that instances are independent and identical distributed. Hence, it fails to capture the intrinsic autocorrelation between instances in the network and discriminating power of network topology which is essential to the collective classification tasks in which the data instances are connected with dependency. Towards this end, we take the network structure into account when we compute the label probability distribution $P(c|x)$ by adding a network regularizer into the objective function. Our goal is to find a better estimation of $P(c|x)$ for classification of instances under the assumption that the distributions of $P(c|x)$ for two data instances are close to each other if they are connected in the network. In particular, the network regularizer performs smoothing to the label probability distribution.

A network dataset with N instances can be represented by an N-by-N adjacency matrix $\mathbf{E} = [E_{is}]$ with entries $E_{is} = 1$ if there is an edge connecting two instances x_i and x_s, and $E_{is} = 0$ otherwise. The basic assumption is that if two nodes x_i and x_s are connected in the network, these nearby nodes tend to share similar class labels, i.e., the distance of the conditional distributions $P(c|x_i)$ and $P(c|x_s)$ should be close to each other. In order to measure the distance between two distributions, we consider the Kullback–Leibler divergence (KL-divergence) of two vectors. Suppose the vector representation of $P(c_k|x_i)$ with respect to different class labels is $\mathbf{z}_i = [P(c_1|x_i), \cdots, P(c_K|x_i)]^T$. Then, the KL-divergence between $\mathbf{z}_i$ and $\mathbf{z}_s$ is defined as

$$D(\mathbf{z}_i||\mathbf{z}_s) = \sum_{k=1}^{K} P(c_k|x_i) \log \frac{P(c_k|x_i)}{P(c_k|x_s)}$$

However, KL-divergence is not symmetric. Therefore, we use the symmetric KL-divergence $\frac{1}{2}\Big(D(\mathbf{z}_i||\mathbf{z}_s)) + D\big(D(\mathbf{z}_s||\mathbf{z}_i)\big)\Big)$ to measure the pairwise distance of two distributions.

Using the adjacency matrix and the symmetric KL-divergence distance measure, we define the network regularizer as

$$\begin{aligned}\mathcal{R} &= \frac{1}{2}\sum_{i,s=1}^{N}\Big(D(\mathbf{z}_i||\mathbf{z}_s)) + D\big(D(\mathbf{z}_s||\mathbf{z}_i)\big)\Big)E_{is} \\ &= \frac{1}{2}\sum_{i,s=1}^{N}\sum_{k=1}^{K}\Big(P(c_k|x_i)\log\frac{P(c_k|x_i)}{P(c_k|x_s)} \\ &\quad + P(c_k|x_s)\log\frac{P(c_k|x_s)}{P(c_k|x_i))}\Big)E_{is} \end{aligned} \tag{4}$$

to measure the smoothness of label probability distribution $P(c|x)$ over the instances in the network. The value of $\mathcal{R}$ ranges from 0 to ∞. By minimizing $\mathcal{R}$, we get a label probability distribution which satisfies the assumption that connected nodes tend to share similar label probability distributions.

Combining this network regularizer $\mathcal{R}$ with the objective function $\mathcal{L}$ in Eq. (2), we obtain

$$\mathcal{O}_1 = \mathcal{L} - \lambda\mathcal{R} \tag{5}$$

where $\lambda \geq 0$ is the regularization parameter controlling the smoothness of the prediction model.

By maximizing the above objective function, GMNR can output a label probability distribution $[P(c_1|x_i), \cdots, P(c_K|x_i)]^T$ indicating the relevance of each label to a given instance x_i. For single-label classification prediction, the instance x_i is then assigned with a class label with the largest probability value, i.e., $Y_{ik} = 1$ if $k = \arg\max_{k'} P(c_{k'}|x_i)$ and $Y_{ik} = 0$ otherwise.

Remark 1. Note that topic modeling methods has been exploited in a number of real-world big data research problems [18]. From the geometric perspective, the learning data are usually reside on or close to an underlying sub-manifold embedded in a high dimensional ambient space. In recent work, Cai et al. [19–21] propose topic modeling methods with manifold regularization for document clustering. The topic modeling methods explicitly take the underlying manifold structure into account by constructing a nearest neighbor graph on a scatter of data points. Although previous works do take manifold local consistency into consideration to smooth the topic distribution estimation, most of these methods mainly focus on unsupervised learning and thus cannot be directly applied to collective classification tasks.

3.2. Regularized generative model for multi-label SSCC

For many network data applications, an instance may be associated with multiple class labels. As a consequence, we further generalized the GMNR algorithm to support SSCC under this general setting.

Recall that GMNR obtains a smoothed label probability distribution over the intrinsic network structure. One further hopes that the resulting label probability distribution respects the label correlations. A natural assumption here could be that if two class labels c_k and c_l are related, then the label probability distributions $P(c_k|x_i)$ and $P(c_l|x_i)$ with respect to the instances are also close to each other. In this subsection, we describe how the label correlation knowledge is incorporated into GMNR to train the generative model for SSCC. At the same time, it also solves the multi-label SSCC problem. We call the generalized approach *multi-label regularized generative model* (MRGM) algorithm.

First, a label-to-label affinity graph is constructed to encode the label correlation information. There are many choices to define the weight matrix $\mathbf{F} = [F_{kl}]$ on the affinity graph. Specifically, we use the commonly used dot-product weighting to measure the similarity between two labels on the graph. We define

$$F_{kl} = Y_k^T Y_l,$$

where $Y_k = [Y_{1k}, \cdots, Y_{Nk}]^T$ such that $Y_{i,k}$ is nonzero if x_i belongs to class c_k and the remaining elements are zero. Here, Y_k and Y_l are normalized to unit length, thus the dot product of the two vectors is equivalent to their cosine similarity.

Similar to Section 3.1, we use the KL-divergence to measure the distance of distributions. Given the vector representation of $P(c_k|x_i)$ with respect to different data instances as $\mathbf{r}_k = [P(c_k|x_1), \cdots, P(c_k|x_N)]^T$, the KL-divergence between $\mathbf{r}_k$ and $\mathbf{r}_l$ is defined as

$$D(\mathbf{r}_k||\mathbf{r}_l) = \sum_{i=1}^{N} P(c_k|x_i)\log\frac{P(c_k|x_i)}{P(c_l|x_i)}$$

By using the weighting matrix $\mathbf{F}$ and the symmetric KL-divergence measure, we can define the following label regularizer

$$\begin{aligned}\mathcal{H} &= \frac{1}{2}\sum_{k,l=1}^{K}\Big(D(\mathbf{r}_k||\mathbf{r}_l)) + D(\mathbf{r}_l||\mathbf{r}_k)\Big)F_{kl} \\ &= \frac{1}{2}\sum_{i=1}^{N}\sum_{k,l=1}^{K}\Big[P(c_k|x_i)\log\frac{P(c_k|x_i)}{P(c_l|x_i)} + P(c_l|x_i)\log\frac{P(c_l|x_i)}{P(c_k|x_i))}\Big]F_{kl}\end{aligned} \tag{6}$$

Please cite this article in press as: Q. Wu et al., Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.04.002

Combining this label regularizer with the objective function in Eq. (5), we obtain the new objective function

$$\mathcal{O}_2 = \mathcal{L} - \lambda(\mathcal{R} + \mathcal{H}) \tag{7}$$

where $\mathcal{L}$, $\mathcal{R}$ and $\mathcal{H}$ are defined in Eqs. (2), (4) and (6) respectively. $\lambda \geq 0$ is the regularization parameter controlling the smoothness of the prediction model.

Remark 2. It is worth mentioning that MRGM can be reduced to the single label collective classification by considering $\mathcal{H} = 0$, in this case the label correlation is not used in learning the generative model. In addition, MRGM can be reduced to the conventional multi-label learning setting by considering $\mathcal{R} = 0$, in this case the instance correlation is not used in learning the generative model.

3.3. EM algorithms for GMNR and MRGM

To learn the prediction model, we optimize the objective functions $\mathcal{O}_1$ and $\mathcal{O}_2$ in Eqs. (5) and (7) for GMNR and MRGM, respectively. In this subsection, we develop an iteration scheme to solve these objective functions based on an effective EM algorithm. The E-step computes the posterior probabilities $P(c_k|x_i, w_j)$ for the latent variable c_k. The M-step updates the $NK + MK$ parameters $\{P(w_j|c_k), P(c_k|x_i)\}$. The EM algorithm performs these two steps alternatively until the maximum number of iterations is reached.

For GMNR, we have the following updating rules for $P^{(t)}(c_k|x_i, w_j)$ in the t-th iteration for the E-step

$$P^{(t)}(c_k|x_i, w_j) = \frac{P^{(t-1)}(w_j|c_k)P^{(t-1)}(c_k|x_i)}{\sum_{l=1}^{K} P^{(t-1)}(w_j|c_l)P^{(t-1)}(c_l|x_i)} \tag{8}$$

as well as the following equations for the M-step

$$P^{(t)}(w_j|c_k) = \frac{\sum_{i=1}^{N} n(x_i, w_j)P^{(t)}(c_k|x_i, w_j)}{\sum_{m=1}^{M}\sum_{i=1}^{N} n(x_i, w_m)P^{(t)}(c_k|x_i, w_m)} \tag{9}$$

Let $\mathbf{y}_k^{(t)} = [P^{(t)}(c_k|x_1), \cdots, P^{(t)}(c_k|x_N)]^T$ be the label probabilities in the t-th iteration. We have

$$\mathbf{y}_k^{(t)} = (\mathbf{\Lambda} + \lambda\mathbf{L})^{-1} \begin{bmatrix} \sum_{j=1}^{M} n(x_1, w_j) \log P^{(t)}(c_k|x_1, w_j) \\ \vdots \\ \sum_{j=1}^{M} n(x_1, w_j) \log P^{(t)}(c_k|x_N, w_j) \end{bmatrix} \tag{10}$$

where $\mathbf{\Lambda}$ is an N-by-N diagonal matrix, and $\mathbf{\Lambda}$ denotes an N-by-N diagonal matrix with (i, i)-th entry as $\rho_i = \sum_{j=1}^{M} n(x_i, w_j)$, and $\mathbf{L}$ is the graph Laplacian matrix defined as $\mathbf{L} = \mathbf{C} - \mathbf{E}$. Here, $\mathbf{E}$ is the adjacency matrix of the network data and $\mathbf{C}$ is an N-by-N diagonal matrix whose entries are column sums of $\mathbf{E}$, i.e., $C_{i,i} = \sum_s E_{is}$.

Theorem 1. *The objective function $\mathcal{O}_1$ is non-decreasing under the updating rules in Eqs. (8), (9) and (10).*

For MRGM, the updating rules for $P^{(t)}(c_k|x_i, w_j)$ in the E-step and $P^{(t)}(w_j|c_k)$ in the M-step are the same as Eqs. (8) and (9), respectively. For the update rule of $P^{(t)}(c_k|x_i)$ in the M-step, we have

$$\mathbf{y}^{(t)} = (\mathbf{\Omega} + \boldsymbol{\lambda}(\mathbf{D} - \mathbf{B} + \mathbf{U} - \mathbf{R}))^{-1}\mathbf{Z}^{(t)}, \tag{11}$$

where $\mathbf{y}^{(t)}$ denotes a K-by-1 block matrix $[\mathbf{y}_1^{(t)}, \cdots, \mathbf{y}_K^{(t)}]^T$ with $\mathbf{y}_k^{(t)} = [P^{(t)}(c_k|x_1), \cdots, P^{(t)}(c_k|x_N)]^T$ for update rules in the t-th iteration, and $\mathbf{\Omega}$, $\mathbf{D}$, $\mathbf{B}$, $\mathbf{U}$, and $\mathbf{R}$, $\mathbf{Z}^{(t)}$ are NK-by-NK sparse matrices described as follows.

$\mathbf{\Omega} = [\mathbf{\Omega}_{i,j}]$ is a K-by-K block matrix where its (i, j)th block is an N-by-N diagonal matrix. All the non-diagonal entries of $\mathbf{\Omega}$ are equal to 0, while the diagonal entries $\Omega_{i,i,s,s} = \rho_i = \sum_{j=1}^{M} n(x_i, w_j)$. $\mathbf{D} = [\mathbf{D}_{i,j}]$ is a K-by-K block diagonal matrix, where the (i, j)th block of $\mathbf{D}$ is an N-by-N matrix $\mathbf{D}_{i,j} = [d_{i,j,s,t}]_{s,t=1,\ldots,N}$. All the entries of $\mathbf{D}$ are equal to 0 except the diagonal entries $d_{i,i,s,s} = \sum_s E_{is}$. Here E is the adjacency matrix of the network data. $\mathbf{B} = [\mathbf{B}_{i,j}]$ is another K-by-K block diagonal matrix, where its (i, j)th block is also an N-by-N matrix $\mathbf{B}_{i,j} = [b_{i,j,s,t}]_{s,t=1,\ldots,N}$. The entries of $\mathbf{B}$ are equal to 0 when $i \neq j$; otherwise, if $i = j$, then we have $b_{i,j,s,t} = E_{st}$. $\mathbf{U} = [\mathbf{U}_{i,j}]$ is an N-by-N block diagonal matrix, where the (i, j)th block of $\mathbf{U}$ is a K-by-K matrix $\mathbf{U}_{i,i} = [u_{i,i,s,t}]_{s,t=1,\ldots,K}$. All non-diagonal entries of $\mathbf{U}$ are equal to 0 and the diagonal entries $u_{i,i,s,s} = \sum_s F_{sl}$. Here F is the label affinity matrix. $\mathbf{R} = [\mathbf{R}_{i,j}]$ is another N-by-N block matrix where its (i, j)-th block is a K-by-K matrix $\mathbf{R}_{i,j} = [r_{i,j,s,t}]_{s,t=1,\ldots,N}$. Indeed, each $\mathbf{R}_{i,j}$, for $i, j = 1, \ldots, K$, is a diagonal matrix $r_{i,j,s,s} = F_{ij}$. $\mathbf{Z}^{(t)}$ is a K-by-1 block matrix for update rules in the t-th iteration, where its k-th entry $\mathbf{Z}_k^t$ is an N dimensional column vector defined as follows:

$$\mathbf{Z}_k^t = \begin{bmatrix} \sum_{j=1}^{M} n(x_1, w_j)P^t(c_k|x_1, w_j) \\ \cdots \\ \sum_{j=1}^{M} n(x_N, w_j)P^t(c_k|x_N, w_j) \end{bmatrix}$$

In GMNR and MRGM, the values of $P(w_i|c_k)$ and $P(c_k|x_i)$ are set to the class prior (of the known label data) during initialization. We assume that each feature w_j is conditionally independent to each other given the label c_k. $P(w_j|c_k)$ are initialized as $P(w_j|c_k) = \frac{n(w_j, c_k)}{\sum_i n(w_i, c_k)}$, where $n(w_j, c_k)$ is the frequency of w_j and c_k co-occurred. The label distribution $P(c_k|x_i)$ for unlabeled data x_i are initialized as $P(c_k|x_i) = \frac{\sum_i n(c_k, x_i)}{\sum_l \sum_i n(c_l, x_i)}$, where $n(c_k, x_i) = 1$ if x_i is labeled as c_k and $n(c_k, x_i) = 0$ otherwise.

Theorem 2. *The objective function $\mathcal{O}_2$ is non-decreasing under the updating rules in Eqs. (8), (9) and (11).*

Proofs for the above two theorems are in Appendix A.

3.4. Thresholding scheme for multi-label classification

Consider a confidence vector $\mathbf{w}_i = [w_i(1), \cdots, w_i(q)] \in \mathbb{R}^q$ for a given instance x_i where each element corresponds to a confidence for one class label. Given $\mathbf{w}_i$, the task of making a multi-label prediction Y_i to x_i is to find a bipartition of relevant and irrelevant labels based on a threshold function $f_t(\mathbf{w}_i)$, such that

$$Y_i(j) = \begin{cases} 1, & \text{if } w_i(j) \geq t, \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

where t is a predefined threshold in the range of $[0, 1]$, and there are many choices to define the threshold t. For instance, $t = 0.5$ is usually used as the threshold for traditional binary classification [22]. This is the simplest method and is very easy to compute. This method uses one threshold for different data sets. Here, we introduce a new thresholding scheme for threshold selection in multi-label classification, i.e., maximum drop thresholding method. The proposed thresholding method selects one threshold for each instance. For a given multi-label instance x_i, we compute a new threshold value t_i based on the confidence vector values $\mathbf{w}_i$ of x_i. Intuitively, one hopes that the confidences of the relevant labels should be much larger than the confidences of the remaining irrelevant labels. Given the confidences $\mathbf{w}_i$ of a given instance x_i, we first sort the labels according to $\mathbf{w}_i$, find two sorted classes with largest drop in terms of their confidence values, and then use the median value of these two classes as a threshold to create a separation of relevant and irrelevant labels for x, where relevant labels are the ones with confidences larger than the threshold, and irrelevant labels are the remaining ones.

Please cite this article in press as: Q. Wu et al., Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.04.002

Table 1
An example of multi-label classification procedure, where the number of possible labels is $K = 7$ and the threshold is $t = 0.6$ for the function $f_t(\mathbf{w})$.

	$P(c_1\|x)$	$P(c_2\|x)$	$P(c_3\|x)$	$P(c_4\|x)$	$P(c_5\|x)$	$P(c_6\|x)$	$P(c_7\|x)$
$\mathbf{w} =$	[0.95	0.05	0.25	0.15	0.1	0.98	0.01]
$Y = f_{t=0.6}(\mathbf{w}) =$	[1	0	0	0	0	1	0]

In MRGM, the algorithm outputs a label probability distribution $[P(c_1|x_i), \cdots, P(c_K|x_i)]^T$ for each given multi-label instance where $P(c_k|x_i)$ indicates the confidence of a label c_k to an instance x_i. For a given multi-label instance x_i, the obtained label probability distribution is equivalent to confidence vector $\mathbf{w}_i$. We show this with an example in Table 1, where the largest drop of the confidences for the sorted classes is observed between 0.95 and 0.25, and their medial value, say $t = 0.6$, is used as a threshold to separate relevant labels (i.e., c_1 and c_6) and irrelevant labels (i.e., c_2, c_3, c_4, c_5 and c_7).

Remark 3. Various semi-supervised learning strategies have been proposed and developed to perform semi-supervised CC. An important component of semi-supervised learning is to build a mechanism to make use of both the labeled and unlabeled data to enhance the classification performance. Most approaches [23–25,5], which all use some variant of ICA [4], first learn an "attribute-only classifier" M_A, then predict labels for the unlabeled nodes with M_A. The known labels (of labeled data) and predicted labels (of unlabeled data) are used together to compute relational features and learn one "node classifier" M_{AR} using both attributes and relational feature values. These relevant studies are [23–25,5]. The variants are different in how they use the attributes and relational features to build M_{AR}. Other approaches to semi-supervised CC have explored how to perform learning without needing to learn an explicit classifier for relational features. For example, Shi et al. create latent links that enable label propagation to classify the nodes. In [26], Tang and Liu use the links to extract latent social dimensions that enable node classification. These existing relevant studies only deal with single-label semi-supervised CC so far. They do not propose for multi-label semi-supervised CC where each node is associated with multiple labels.

The main contribution of this paper is to develop a novel PLSA generative model for multi-label semi-supervised CC. To the best of the authors' knowledge, no other previous work has been done on utilizing PLSA for multi-label SSCC tasks. For our proposed method, the supervised knowledge of the labeled data is used to guide the procedure of learning the generative model for classification, and the network regularizer is used as a smoothing operator on the label probability distributions $P(c|x)$ of instances in the network. In this way, two connected instances have similar label probability distributions. The combination effect of these two aspects results in the effectiveness of the proposed GMNR model for SSCC tasks. Our proposed method is different from existing ICA variants and label propagation methods. In ICA variants, the predicted labels of unlabeled nodes are used to build the relational features. Such predictions are then iteratively refined by an iteration method. However, this iteration method is non-convex. The solution is not unique and the performance highly depends on the initial predictions. In label propagation methods, the latent links are evaluated by evaluating the loss of predictions using different links. However, such loss function is difficult to minimize and these methods usually have time complexity at least quadratic in the number of nodes (N), and thus do not scale to large, realistic graphs. In the proposed method, we only need to solve $\mathcal{O}_1$ for single-label semi-supervised CC and $\mathcal{O}_2$ for multi-label semi-supervised CC via using a network regularizer and a label regularizer to encode the network structure and the label correlation, respectively. Iterative algorithms are formulated for solving the objective functions. The algorithms we use are only linear in the number of nodes. The solutions are unique and the algorithms always converge to these solutions.

Table 2
The description of the five datasets.

Characteristics	Cora	Citeseer	Genes	DBLP-A	DBLP-B
#Instances	2708	3312	1243	23 806	16 020
#Features	1433	3703	461	12 588	8595
#Links	5278	4598	1326	86 895	55 526
#Classes	7	6	2	6	6

4. Empirical evaluation

In this section, we compare the performance of the proposed GMNR and MRGM algorithms with other previously proposed algorithms: wvRN+RL, ICA, semiICA, LNP, and ICML on various network datasets, and show that the proposed algorithm outperforms these algorithms. For the purpose of reproducibility, we provide the code and data sets at: https://sites.google.com/site/qysite/.

4.1. Datasets

Before we proceed to presenting empirical results, we provide a description of the used datasets and experimental settings. We use three network datasets for single-label collective classification and two network datasets for multi-label collective classification. The characteristics of the datasets are summarized in Table 2. They are described below.

4.1.1. Single-label collective classification data

The three benchmark collective classification datasets (Cora, Citeseer, and Genes)[1] are from different application domains. These datasets have been widely used in prior research on collective classification [27,2,28].

The **Cora** dataset is a paper publication dataset which is used frequently in collective classification studies [2]. It consists of 2708 machine learning papers classified into one of seven classes: "Case Based", "Genetic Algorithms", "Neural Networks", "Probabilistic Methods", "Reinforcement Learning", "Rule Learning" and "Theory". Each node on the collective network represents a paper document described by a 0/1 valued bag-of-word vector with 1433 dimensions. The citations in a paper are used to construct links to the cited papers.

The **Citeseer** dataset is a collection of research papers drawn from the Citeseer collection [2]. The dataset consists of 3312 instances taken from six classes as follows: "AI", "Agents", "DB", "HCI", "IR" and "ML". Each instance is described as a 0/1 bag-of-word vector indicating the absence or presence of particular words in the corresponding paper. The links between the instances represent their citation relations.

The **Genes** dataset is a protein interaction network dataset released in KDD cup 2001 for gene localization prediction [29]. The

[1] Available at http://linqs.cs.umd.edu/projects//projects/lbc/index.html.

Please cite this article in press as: Q. Wu et al., Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.04.002

task is to predict "nucleus" or "non-nucleus" using features including Essential, Class, Complex, Phenotype, Motif and Chromosome. Each protein is represented as a 461 dimensional binary vector by binarizing these features. The links represent the protein–protein interactions.

4.1.2. Multi-label collective classification data

We use two multi-label collective classification datasets (DBLP-A and DBLP-B) derived from the DBLP computer science bibliography website. These datasets have been used in [15] for multi-label collective classification.

In the **DBLP-A** dataset, the nodes represent the research authors. They are described by the attributes of the papers published by the authors with respect to different conferences. The DBLP-A dataset consists of a collection of 23 800 research authors who have published papers from 2000 to 2010. Each node (author) is represented by a 12,588 dimensional bag-of-word feature vector of the paper published by the authors. Two authors who have collaborated with each other are linked together. Each author has one (or multiple) research topic(s) of interests from 6 research areas in terms of different representative conferences. The conferences (classes) for DBLP-A are given as follows:

1. Database: ICDE, VLDB, SIGMOD, PODS, EDBT;
2. Data Mining: KDD, ICDM, SDM, PKDD, PAKDD;
3. Artificial Intelligence: IJCAI, AAAI;
4. Information Retrieval: SIGIR, ECIR;
5. Computer Vision: CVPR;
6. Machine Learning: ICML, ECML.

Again, the **DBLP-B** dataset is derived from the DBLP website. It contains 16,020 instances (authors) represented by feature vectors with 8595 dimensions. There are 55,526 links in the network and 6 classes in terms of conferences as follows:

1. Algorithms & Theory: STOC, FOCS, SODA, COLT;
2. Natural Language Processing: ACL, ANLP, COLING;
3. Bioinformatics: ISMB, RECOMB;
4. Networking: SIGCOMM, MOBICOM, INFOCOM;
5. Operating Systems: SOSP, OSDI;
6. Parallel Computing: POD, ICS.

4.2. Evaluation metrics

The evaluation is performed on the two collections of datasets described above. For the single-label collective classification datasets, we evaluate the performance by classification accuracy which is measured as follows:

$$\text{Accuracy} = \frac{\#\text{Test data labeled correctly}}{\#\text{Test data}}$$

For the multi-label collective classification datasets, we evaluate the performance in terms of six multi-label learning evaluation measures (*Hamming loss*, *one-error*, *coverage*, *average precision* *macro-F1* and *micro-F1*). These measures are defined as follows.

(1) *Hamming loss* evaluates how many times the class label is misclassified, i.e., a label not belonging to the instance is predicted or a label belonging to the instance is not included. It is defined as

$$hamming\_loss(h) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{K}|h(x_i)\triangle Y_i|$$

where $h(x_i)$ denotes the predictions for the instance x_i, and $h(x_i)\triangle Y_i$ denotes the symmetric difference between the predicted labels $h(x_i)$ and the ground-truth labels Y_i.

(2) *One-error* evaluates how many times the top-ranked label is *not* in the set of true labels of the instance. Define a classifier that assigns a single label for an instance x_i by $H(x_i) = \arg\max_{c\in\mathcal{C}} h(x_i, c)$, then the one-error is

$$one\_error(H) = \frac{1}{N}\sum_{i=1}^{N}[\![H(x_i)\notin Y_i]\!],$$

where $[\![\pi]\!]$ is 1 if π holds and 0 otherwise. For single label classification problems, the one-error is identical to classification error.

(3) *Coverage* evaluates how far we need, on the average, to go down the list of labels in order to cover all the true labels of an instance

$$\text{coverage}(f) = \frac{1}{N}\sum_{i=1}^{N}\max_{c\in Y_i} rank_s(x_i, c) - 1,$$

where $rank_s(x_i, c)$ is the rank of the class label c derived from the function $s(x_i, c)$ which returns a real-value indicating the confidence for the class label c to be a proper label of x_i, so that $rank_s(x_i, c') < rank_s(x_i, c)$ if $s(x_i, c') > s(x_i, c)$.

(4) *Average precision* evaluates the average fraction of labels ranked above a particular label $c \in Y_i$ which are actually in Y_i

$$avg\_prec(f) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i|}\sum_{c\in Y_i}\frac{|\mathcal{P}_i|}{rank_s(x_i, c)},$$

where $\mathcal{P}_i = \{c' \in Y_i | rank_s(x_i, c') \le rank_s(x_i, c)\}$.

(5) *Macro-F1* is the harmonic mean between precision and recall, where the average is calculated per label and then averaged across all labels. It is defined as

$$macro\_F1 = \frac{1}{K}\sum_{k=1}^{K}\frac{2\times p_k \times r_k}{p_k + r_k}$$

where $p_k = \frac{tp_k}{tp_k + fp_k}$ and $r_k = \frac{tp_k}{tp_k + fn_k}$ are the precision and recall of the k-th label c_k, and tp_k, fp_k and fn_k are defined as the number of true positive, false positive and false negative for the label c_k (considered in a binary class setting).

(6) *Micro-F1* is the harmonic mean between the *micro-precision* and *micro-recall*.

$$micro\_F1 = \frac{2\times micro\_precision \times micro\_recall}{micro\_precision + micro\_recall}$$

where $micro\_precision = \frac{\sum_{k=1}^{K} tp_k}{\sum_{k=1}^{K} tp_k + \sum_{k=1}^{K} fp_k}$ and $micro\_recall = \frac{\sum_{k=1}^{K} tp_k}{\sum_{k=1}^{K} tp_k + \sum_{k=1}^{K} fn_k}$. Here, *micro-precision* (*micro-recall*) is the results of precision (recall) averaged over all the instance/label pairs.

One notes that the smaller the values of *hamming loss*, *one-error* and *coverage*, the better the performance. On the other hand, the bigger the values of *average precision*, *macro-F1* and *micro-F1*, the better the performance.

4.3. Comparison methods

We compare our proposed method with the following collective classification algorithms:

1. **wvRN+RL**. This baseline is a relational-only CC method using only relational information. This method computes the label of an instance by averaging the labels of its neighbors. Macskassy and Provost [9] have shown that wvRN+RL preforms quite well in some network datasets.

Please cite this article in press as: Q. Wu et al., Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.04.002

Table 3
The performance (mean ± standard deviation) of the algorithms with varying percentages of labeled data on Cora dataset, Citeseer dataset and Genes dataset, the best performance achieved among different compared algorithms is bolded.

Dataset	Percentage	wvRN+RL	ICA	semiICA	LNP	GMNR
Cora	0.01	0.493 ± 0.08	0.410 ± 0.03	0.503 ± 0.05	0.476 ± 0.09	**0.773 ± 0.02**
	0.02	0.573 ± 0.08	0.442 ± 0.04	0.581 ± 0.06	0.558 ± 0.08	**0.785 ± 0.01**
	0.03	0.698 ± 0.05	0.544 ± 0.07	0.675 ± 0.04	0.701 ± 0.05	**0.797 ± 0.02**
	0.04	0.706 ± 0.03	0.563 ± 0.09	0.710 ± 0.03	0.707 ± 0.03	**0.805 ± 0.01**
	0.05	0.740 ± 0.02	0.553 ± 0.06	0.729 ± 0.01	0.742 ± 0.02	**0.824 ± 0.02**
	0.06	0.757 ± 0.02	0.626 ± 0.04	0.750 ± 0.02	0.766 ± 0.02	**0.827 ± 0.01**
	0.07	0.760 ± 0.02	0.652 ± 0.07	0.766 ± 0.02	0.771 ± 0.02	**0.828 ± 0.01**
	0.08	0.776 ± 0.01	0.661 ± 0.06	0.784 ± 0.01	0.789 ± 0.01	**0.832 ± 0.01**
	0.09	0.775 ± 0.01	0.665 ± 0.06	0.789 ± 0.02	0.788 ± 0.02	**0.835 ± 0.01**
	0.1	0.786 ± 0.01	0.714 ± 0.03	0.789 ± 0.01	0.798 ± 0.01	**0.837 ± 0.01**
	0.2	0.812 ± 0.01	0.809 ± 0.02	0.834 ± 0.01	0.823 ± 0.01	**0.851 ± 0.01**
Citeseer	0.01	0.358 ± 0.06	0.431 ± 0.05	0.476 ± 0.05	0.324 ± 0.10	**0.664 ± 0.03**
	0.02	0.437 ± 0.04	0.507 ± 0.04	0.584 ± 0.04	0.447 ± 0.09	**0.690 ± 0.02**
	0.03	0.479 ± 0.03	0.566 ± 0.03	0.621 ± 0.03	0.527 ± 0.07	**0.707 ± 0.01**
	0.04	0.499 ± 0.04	0.607 ± 0.03	0.641 ± 0.02	0.579 ± 0.05	**0.710 ± 0.01**
	0.05	0.524 ± 0.04	0.632 ± 0.04	0.665 ± 0.03	0.597 ± 0.05	**0.716 ± 0.02**
	0.06	0.541 ± 0.01	0.649 ± 0.03	0.672 ± 0.01	0.622 ± 0.04	**0.719 ± 0.01**
	0.07	0.549 ± 0.02	0.666 ± 0.02	0.684 ± 0.01	0.648 ± 0.02	**0.721 ± 0.01**
	0.08	0.543 ± 0.03	0.686 ± 0.01	0.698 ± 0.01	0.646 ± 0.02	**0.721 ± 0.01**
	0.09	0.558 ± 0.02	0.679 ± 0.02	0.697 ± 0.01	0.676 ± 0.02	**0.724 ± 0.01**
	0.1	0.560 ± 0.02	0.694 ± 0.02	0.702 ± 0.01	0.676 ± 0.01	**0.726 ± 0.01**
	0.2	0.608 ± 0.02	0.731 ± 0.01	0.720 ± 0.01	0.701 ± 0.01	**0.741 ± 0.01**
Genes	0.01	0.539 ± 0.074	0.637 ± 0.05	0.653 ± 0.06	0.529 ± 0.06	**0.713 ± 0.08**
	0.02	0.604 ± 0.02	0.699 ± 0.05	0.695 ± 0.04	0.573 ± 0.03	**0.770 ± 0.05**
	0.03	0.581 ± 0.07	0.744 ± 0.06	0.757 ± 0.02	0.583 ± 0.09	**0.813 ± 0.03**
	0.04	0.627 ± 0.07	0.764 ± 0.06	0.777 ± 0.04	0.621 ± 0.09	**0.814 ± 0.02**
	0.05	0.642 ± 0.05	0.780 ± 0.05	0.779 ± 0.03	0.641 ± 0.07	**0.822 ± 0.03**
	0.06	0.643 ± 0.07	0.786 ± 0.05	0.795 ± 0.03	0.639 ± 0.08	**0.827 ± 0.03**
	0.07	0.671 ± 0.02	0.793 ± 0.03	0.809 ± 0.01	0.644 ± 0.03	**0.831 ± 0.01**
	0.08	0.674 ± 0.04	0.818 ± 0.03	0.819 ± 0.02	0.706 ± 0.05	**0.839 ± 0.02**
	0.09	0.683 ± 0.03	0.820 ± 0.03	0.818 ± 0.02	0.708 ± 0.03	**0.836 ± 0.01**
	0.1	0.672 ± 0.03	0.838 ± 0.02	0.830 ± 0.01	0.707 ± 0.04	**0.848 ± 0.01**
	0.2	0.729 ± 0.01	0.866 ± 0.02	**0.867 ± 0.01**	0.772 ± 0.02	0.849 ± 0.02

2. **ICA**. The Iterative Collective classification Algorithm (ICA) [4] is one of the most popular CC methods that frequently used as a baseline for CC evaluation in previous studies. ICA uses a local classifier for classification prediction. Prior work has found Logistic Regression (LR) to be superior to other classifiers, such as NB and *k*NN, for ICA [14]. Therefore, we use LR as the local classifier for ICA in our experiments.
3. **semiICA**. This baseline is a semi-supervised collective classification method. It uses the idea of label regularization to learn hybrid local classifiers, enabling them to leverage the unlabeled data to bias the learning process of the ICA algorithm.
4. **LNP**. This baseline is another semi-supervised collective classification method [12]. It explores latent linkages among the nodes to generate a latent graph for label propagation. There may exist various latent linkages for latent graph construction. Semantic similarity is one of the most commonly used methods for latent graph generation. In our experiments, we use the semantic similarity linkages for the LNP algorithm. Such linkages can be obtained by connecting the nearest neighbor of the instances based on their attribute similarity.
5. **ICML**. The Iterative Classification of Multiple Labels (ICML) algorithm [15] is a multi-label collective classification method. This method uses an iterative classification algorithm similar to ICA. The main difference between ICML and ICA is that ICML uses the dependencies among the labels in the learning process.

The maximum number of iterations for ICA, semiICA, and ICML is set to 10, the default used in [15], while wvRN+RL is set to 1000. The parameter λ for our proposed methods is set to 5. The parameter selection will be discussed in the later section.

4.4. Collective classification results

We first consider using the single-label GMNR model to perform the learning tasks on network data. We conduct experiments on the Cora, Citeseer, and Genes datasets to evaluate the performance of the proposed GMNR method by comparing with the learning algorithms: wvRN+RL, ICA, semiICA and LNP. For each dataset, a small number of examples are randomly selected for each category as labeled data. The remaining ones are used as unlabeled data for testing the quality of the classification. Specifically, we randomly select different percentage of data ranging from 1% to 20% as labeled data. This is a challenging problem from the classifier training perspective because only a small number of examples are used as labeled data. The performance is measured in classification accuracy rate by averaging 10 trials (randomly selection of labeled/unlabeled data) for each data percentage.

The performance results for each algorithm on the Cora, Citeseer and Genes datasets are shown in Table 3 to compare the effectiveness of the algorithms as the percentage of labeled data varies. These experimental results reveal a number of interesting points:

- Regardless of the datasets, our proposed GMNR method consistently outperforms the other algorithms. This illustrates the advantages of our method in learning the network data.
- The smaller the number of labeled data is, the larger improvement GMNR can achieve. We see that GMNR outperforms the other algorithms by a large margin when the labeled data is less than 10%. GMNR obtains the largest improvement when learning with only 1% labeled data. This result illustrates the advantages of our GMNR approach when there is only small number of labeled data available.

Please cite this article in press as: Q. Wu et al., Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.04.002

Table 4
Experimental results in terms of Accuracy evaluation measure. The performance of the methods are ranked in decreasing order and shown in brackets. The average rank is the average of the ranks across all datasets.

Dataset	wvRN+RL	LNP	ICA	semiICA	GMNR
Cora 1	0.654(3)	0.682(2)	0.526(5)	0.626(4)	**0.787(1)**
Cora 2	0.695(3)	0.728(2)	0.609(5)	0.688(4)	**0.808(1)**
Cora 3	0.647(4)	0.684(2)	0.530(5)	0.673(3)	**0.772(1)**
Cora 4	0.611(4)	0.676(2)	0.577(5)	0.641(3)	**0.777(1)**
Cora 5	0.662(4)	0.687(2)	0.550(5)	0.681(3)	**0.782(1)**
Cora 6	0.659(4)	0.682(2)	0.585(5)	0.671(3)	**0.798(1)**
Cora 7	0.675(3)	0.698(2)	0.543(5)	0.653(4)	**0.786(1)**
Cora 8	0.657(4)	0.660(3)	0.593(5)	0.666(2)	**0.777(1)**
Cora 9	0.677(3)	0.716(2)	0.591(5)	0.669(4)	**0.785(1)**
Cora 10	0.649(4)	0.686(3)	0.580(5)	0.691(2)	**0.788(1)**
Citeseer 1	0.433(5)	0.560(4)	0.628(3)	0.651(2)	**0.691(1)**
Citeseer 2	0.393(5)	0.565(4)	0.613(3)	0.644(2)	**0.697(1)**
Citeseer 3	0.416(5)	0.561(4)	0.631(3)	0.639(2)	**0.703(1)**
Citeseer 4	0.367(5)	0.543(4)	0.616(3)	0.653(2)	**0.710(1)**
Citeseer 5	0.447(5)	0.597(4)	0.633(3)	0.642(2)	**0.694(1)**
Citeseer 6	0.440(5)	0.609(4)	0.618(3)	0.632(2)	**0.689(1)**
Citeseer 7	0.359(5)	0.591(4)	0.613(3)	0.641(2)	**0.690(1)**
Citeseer 8	0.415(5)	0.563(4)	0.604(3)	0.630(2)	**0.694(1)**
Citeseer 9	0.447(5)	0.542(4)	0.634(3)	0.645(2)	**0.684(1)**
Citeseer 10	0.485(5)	0.596(4)	0.625(3)	0.654(2)	**0.706(1)**
Genes 1	0.618(5)	0.677(4)	0.727(2)	0.713(3)	**0.785(1)**
Genes 2	0.615(4)	0.575(5)	0.753(3)	0.786(2)	**0.820(1)**
Genes 3	0.644(4)	0.630(5)	0.783(2)	0.782(3)	**0.825(1)**
Genes 4	0.580(5)	0.640(4)	0.723(2)	0.717(3)	**0.805(1)**
Genes 5	0.620(5)	0.650(4)	0.724(3)	0.738(2)	**0.777(1)**
Genes 6	0.577(5)	0.651(4)	0.693(3)	0.697(2)	**0.749(1)**
Genes 7	0.606(5)	0.702(4)	0.779(3)	0.784(2)	**0.799(1)**
Genes 8	0.632(5)	0.692(4)	0.758(3)	0.787(2)	**0.815(1)**
Genes 9	0.622(4)	0.621(5)	0.751(3)	0.754(2)	**0.807(1)**
Genes 10	0.597(5)	0.687(4)	0.756(3)	0.787(2)	**0.807(1)**
Avg. rank	4.433	3.5	3.567	2.5	**1**

- Both semiICA, ICA and LNP can achieve good classification performance when there are 30% of labeled data. If there are sufficient informative and labeled data, these methods can work very well. However, such labeled data are expensive to obtain in real-world applications. In additional, it is usually difficult to estimate how many labeled data are needed to get a good result. By leveraging the unlabeled data, GMNR can achieve a more robust performance across a wide range of label/unlabeled data splits.

In summary, the experimental results demonstrate the effectiveness of the proposed GMNR method. In particular, it can be used for solving collective classification problems even in the paucity of labeled data.

Our proposed algorithm performs well when there are only a small number of labeled instances. It is effective for SSCC in sparsely labeled networks. In order to validate this point further, we employed the corrected Friedman test and the post-hoc Nemenyi test as recommended by Demsar [30] to assess whether the differences in performance across the compared algorithms are statistically significant when there is only limited number of labeled data available. A Friedman test for the null hypothesis that all learners have equal performance is first used. In the case when this hypothesis is rejected, a Nemenyi test is used to compare the algorithms in a pairwise way. For this procedure, the algorithms are ranked according to their performance for each task, so that the best performing algorithm has the rank of 1, the second best the rank of 2, etc. The performance of two algorithms is significantly different to each other if their average ranks across all the learning datasets differ by more than some critical distance (CD). Such CD depends on the number of algorithms, the number of datasets, and the significance level. The experiment is

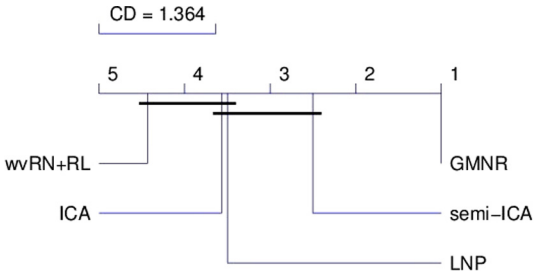

Fig. 1. The graphical presentation of results from the Nemenyi post-hoc test at 0.05 significance level in terms of accuracy.

conducted on 30 classification tasks derived from the Cora, Citeseer and Genes datasets. For each dataset D with N instances, we use bootstrap sampling method to generate 10 new data subsets D_i, each of size $N_i < N$, by sampling from D uniformly and with replacement. Then, we compare the performance of GMNR and the other methods on all 30 datasets with 5% of labeled data using the two-step statistical test procedure for statistical evaluation.

Table 4 shows the results for all 30 datasets. The values in each row in the table correspond to the accuracy values of different methods on one dataset. The methods are ranked in decreasing order in terms of their performances. The number in brackets next to the accuracy values is the rank of the method on the corresponding dataset. The average of the ranks across all datasets (the average rank) is given in the bottom line of the table. Fig. 1 show the results from the Nemenyi post-hoc test with average rank diagrams as suggested in [30]. In the figure, the top line is the axis on which the average ranks of methods are drawn. The algorithms are depicted along the axis in such a manner that the best ranking ones are at the right-most side of the diagram. The lines for the average ranks of the algorithms that do not differ significantly (at the 0.05 significant level) are connected with a line.

From Table 4 and Fig. 1, we come to the following conclusions. First, the Friedman test suggests that the proposed GMNR performs significantly better against other methods in the situation where there is only limited number of labeled data (here, we present the results when one has only 5% of labeled data due to page limitation. The same test has been performed with varying number of labeled data from 1% to 20% and similar conclusions are obtained). Second, although there are no significant differences between the semiICA, LNP, and ICA methods at a significance level of 5%, the overall picture taken from the experiments is clearly in favor of the semiICA and LNP methods using semi-supervised techniques. Third, the relational-only method, wvRN+RL, does not perform competitively because it ignores the attribute information of the instances. Enabling the methods to use both attribute and relational information allows significantly higher accuracies compared to relational-only approach.

4.5. Multi-label collective classification results

We compare the MRGM algorithm with the wvRN+RL, ICA, semiICA and ICML methods for the multi-label collective classification task. Among the four compared methods, ICA, semiICA, and wvRN+RL are the approaches which focus on single-label collective classification. To enable comparison for these baselines, we decompose the multi-label problem into a set of K single-label classification problems using the one-against-all strategy, where K is the total number of possible classes. Then, we train an independent classifier for each one-against-all classification problem. This approach is known as the binary relevance (BR) method [31]. The predictions for all K single-label problems are combined to make the final prediction for multi-label classification.

Please cite this article in press as: Q. Wu et al., Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.04.002

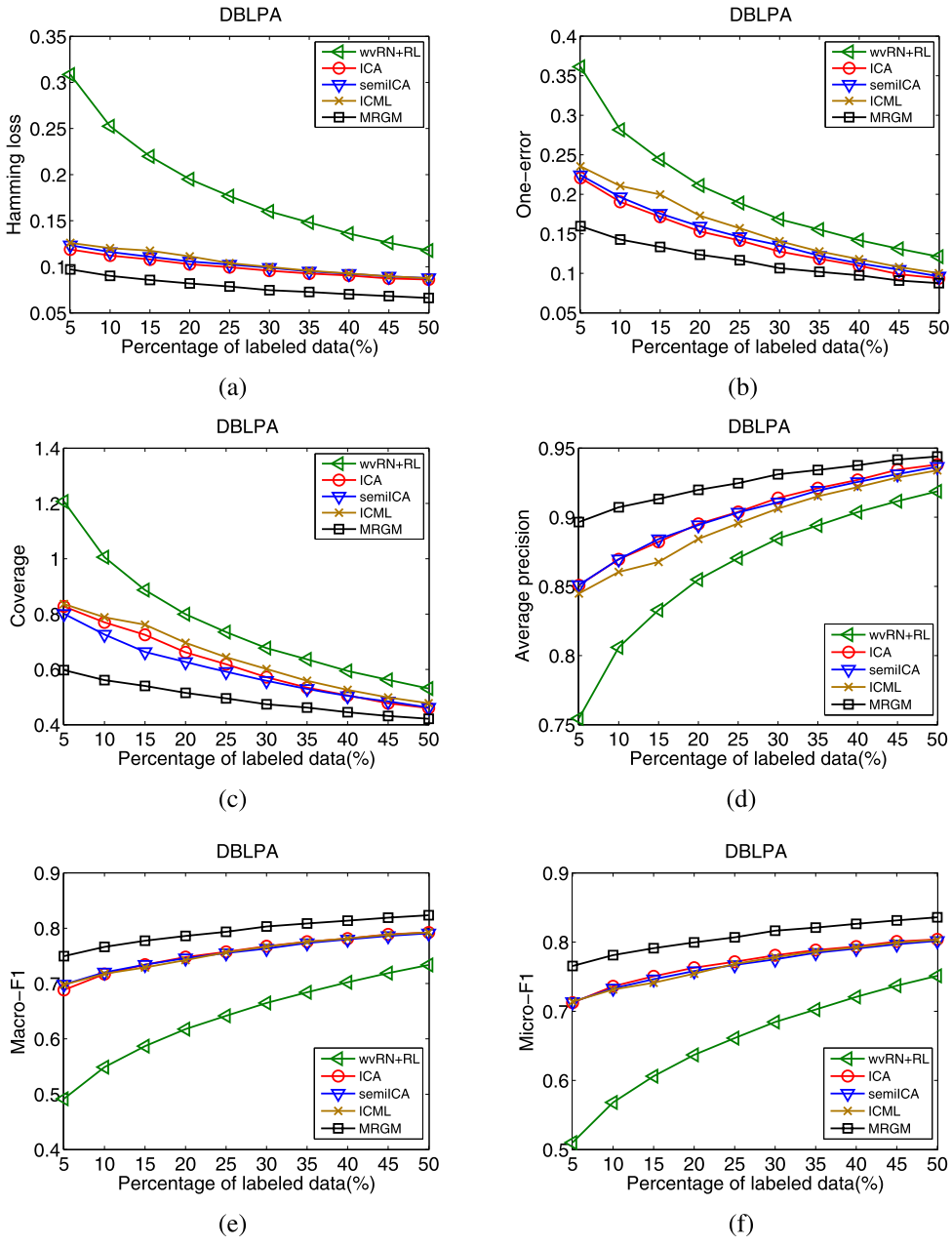

Fig. 2. The performance of the algorithms with varying percentages of labeled data. (a)–(c) Results with respect to Hamming loss, One-error, and Coverage. In these subfigures, the lower value the curve is, the better the performance is. (d)–(f) Results with respect to Average precision, Macro-F1, and Micro-F1. In these subfigures, the larger value the curve is, the better the performance is.

Figs. 2 and 3 show the classification results on the DBLP-A and DBLP-B datasets, respectively. We conduct the evaluations with varying percentage of labeled data ranging from 5% to 50%. For each percentage of labeled data, the performance of each algorithm is measured by averaging 10 trials (randomly selection of labeled/unlabeled data splits) on each dataset. From Figs. 2 and 3, one observes that MRGM outperforms the other algorithms across all the datasets and metrics. It is shown in recent study [32] that one algorithm rarely outperforms another algorithm on all multi-label classification evaluation criteria as they measure the learning performance from different aspects. In the experiments, we find that MRGM is able to consistently produce better results across all the evaluation metrics. This provides evidence to demonstrate the effectiveness of our MRGM algorithm for multi-label collective classification. We also find that MRGM is able to consistently produce better results across different percentages of labeled data. The smaller the number of labeled data, the larger improvement MRGM achieves. Compared with the other algorithms, the performance of MRGM is more stable and it is able to achieve a good classification performance even in the situation of learning with extremely small percentage of labeled examples, e.g., 5%.

4.6. Convergence study

We investigate how fast the GMNR and NRGM algorithms converge for the objective functions $\mathcal{O}_1$ and $\mathcal{O}_2$ in Eqs. (5) and (7). Figs. 5(a) and 5(b) show the convergence curves of the GMNR and MRGM algorithms on the Cora and DBLP-A dataset (at 5% labeled data), respectively. The x-axis is the number of iteration number in the process of optimizing the objective value and the y-axis is the value of successive computed objective value $||\mathcal{O}(t+1) - \mathcal{O}(t)||/||\mathcal{O}(t)||$. We can see that both algorithms converge within

Please cite this article in press as: Q. Wu et al., Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.04.002

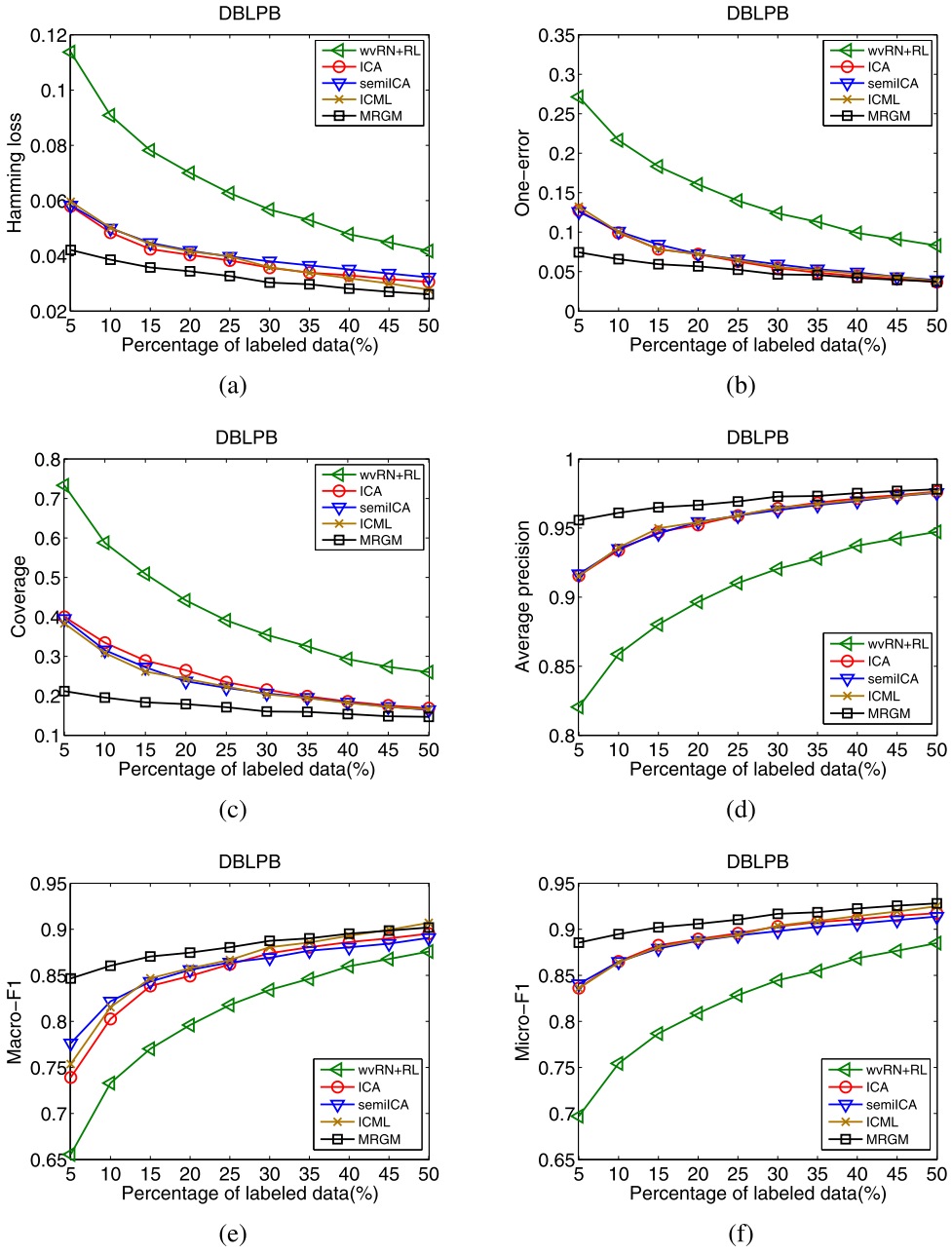

Fig. 3. Similar to Fig. 2, but for DBLP-B dataset.

10 iterations. The required computational time is about 10 s using our MATLAB implementation. We also conduct experiment to compare the running time of wvRN+RL, ICA, semiICA and GMNR algorithms on the Citeseer dataset. The LNP algorithm is not included in the comparison because the generation of latent graphs involved in LNP is very time consuming. The comparison is performed in a computer with 2.40 GHz CPU and 4.0 GB memory. The results of different compared algorithms against different percentages of labeled data on the Citeseer dataset is given in Fig. 4. We can see from the figure that the running time of GMNR is much faster than those of the compared algorithms. Similar fast convergence and running time results on the other datasets are also observed.

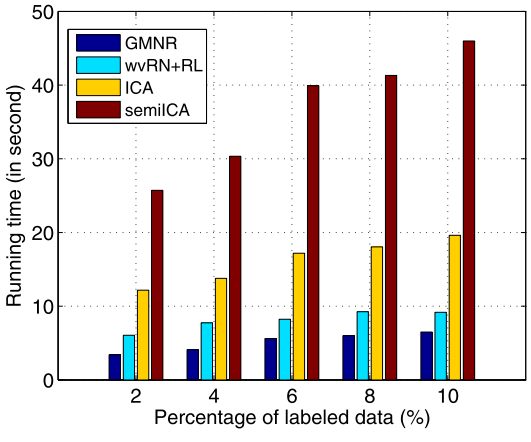

Fig. 4. The running time of different learning algorithms on the Citeseer dataset.

4.7. Parameter selection

Parameter λ is used to weigh the importance of the regularizer for the GMNR and MRGM algorithms. To illustrate the sensitivity

Please cite this article in press as: Q. Wu et al., Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.04.002

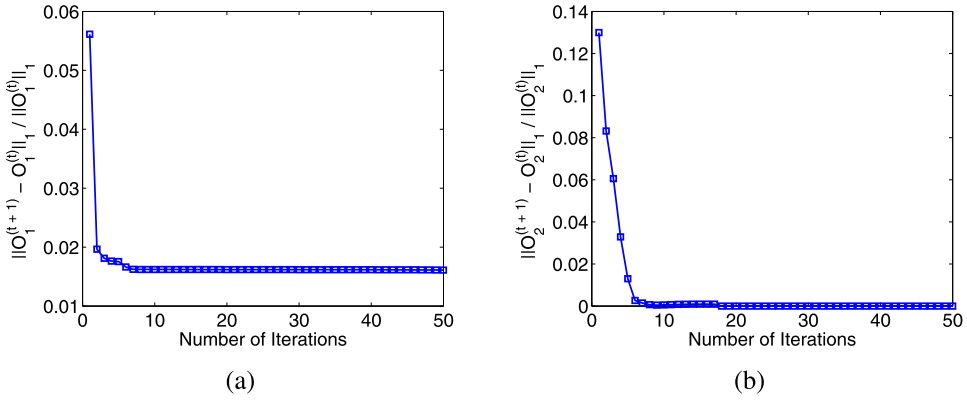

Fig. 5. (a) Convergence curve of GMNR on the Cora dataset. (b) Convergence curve of MRGM on the DBLP-A dataset.

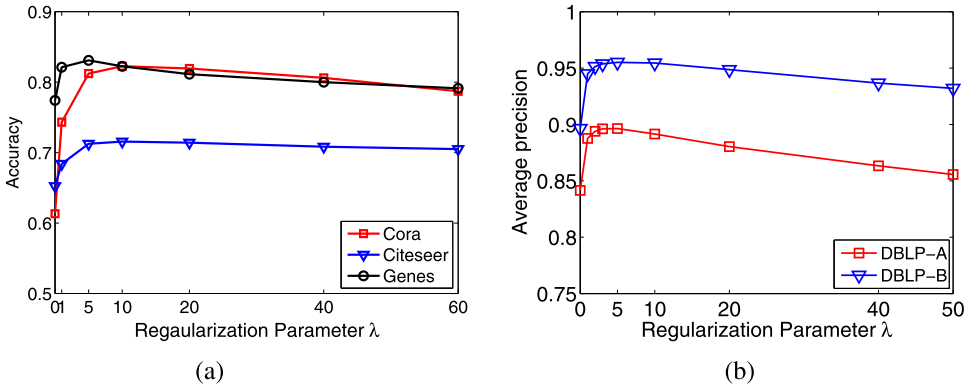

Fig. 6. (a) The performance trend (accuracy) of GMNR for various λ values on the Cora, Citeseer and Genes datasets; (b) The performance trend (average precision) of MRGM for various λ values on the DBLP-A and DBLP-B datasets.

of this parameter, we show the performance of the proposed algorithms with various λ on different datasets in Fig. 6.

The performance of GMNR on Cora, Citeseer, and Genes datasets with λ varied from 0 to 60 is shown in Fig. 6(a). We observe from Fig. 6(a) that when $\lambda = 0$, the accuracy is low. In fact, the proposed GMNR algorithm boils down to the original PLSA when $\lambda = 0$. No network structure or label correlation knowledge is used in this case. When λ increases, the accuracy increases. On the other extreme, with a high value of λ (e.g., $\lambda = 60$), the objective function is dominated by the smoothness on the regularization term. It does not use sufficient attribute features of the instances to learn the generative model for collective classification, but just the relational information. Generally, the plateaus in the accuracy curves indicate that the proposed GMNR is quite insensitive to the specific setting of λ. GMNR achieves consistently good performance when λ varies from 5 to 60 on all the datasets. This implies that the method can be used in a robust way across a wide range of parameters. From Fig. 6(a), one observes that the best performance is achieved at $\lambda = 5$. The average precision of MRGM on DBLP-A and DBLP-B datasets with λ varied from 0 to 50 is shown in Fig. 6(b). The performance trend for MRGM shown in Fig. 6(b) is similar to the performance trend for GMNR.

5. Conclusions

In this paper, we first present a novel generative model with network regularization (GMNR) algorithm for semi-supervised collective classification (SSCC). For GMNR, a network regularizer encodes the network structure, and it is incorporated into the PLSA generative model to learn from network data. The resulting model provides local smoothness of the label probability distributions for classification predictions. Then, we extend the GMNR to handle the SSCC when the instances have multi-labels. The new generative model, called multi-label regularized generative model (MRGM) utilizes an additional label regularizer to explicitly encode the label correlation. The predictions of MRGM ensure consistency among interlinked instances and related labels. We evaluate the proposed GMNR and MRGM algorithms on an extensive set of real world network datasets. Empirical results show that the proposed methods perform significantly better than the other baseline collective classification methods, especially when there are only limited number of labeled data available. Future work includes the development of automated selection method for λ which controls the smoothness of our GMNR model. We will also extend the proposed methods to handle the heterogeneous network data classification problem.

Acknowledgements

This research was supported by the Fundamental Research Funds for the Central Universities and AcRF Grant RG-41/12.

Appendix A

To prove Theorems 1 and 2, we need to show that $\mathcal{O}_1$ is non-decreasing under the updating rules in Eqs. (8), (9) and (10), and $\mathcal{O}_2$ is non-decreasing under the updating rules in Eqs. (8), (9) and (11). Since the regularization terms $\mathcal{R}$ and $\mathcal{H}$ in $\mathcal{O}_1$ and $\mathcal{O}_2$ are only related to $P(c|x)$, we have exactly the same update formula for $P(c|x, w)$ and $P(w|c)$ under Eqs. (8) and (9) as in the original PLSA model. We only need to prove that $\mathcal{O}_1$ and $\mathcal{O}_2$ are non-decreasing under the updating rules in Eqs. (10) and (11), respectively. We make use of an auxiliary function similar to that used in the EM algorithm [17] in our proof.

Please cite this article in press as: Q. Wu et al., Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.04.002

Definition 1. $\mathcal{Q}(\theta, \theta')$ is an auxiliary function for $\mathcal{L}(\theta)$ if the following conditions are satisfied

$$\mathcal{Q}(\theta, \theta') \leq \mathcal{L}(\theta), \quad \mathcal{Q}(\theta, \theta) = \mathcal{L}(\theta) \tag{13}$$

Lemma 1. *If $\mathcal{Q}$ is an auxiliary function of $\mathcal{L}$, then $\mathcal{L}$ is non-decreasing under the update*

$$\theta^{(t+1)} = \arg\max_{\theta} \mathcal{Q}(\theta, \theta^{(t)}) \tag{14}$$

Proof. $\mathcal{L}(\theta^{(t+1)}) \geq \mathcal{Q}(\theta^{(t+1)}, \theta^{(t)}) \geq \mathcal{Q}(\theta^{(t)}, \theta^{(t)}) = \mathcal{L}(\theta^{(t)})$. □

Lemma 2. *Function*

$$Q(\theta, \theta^{(t)}) = \sum_{c} P(c|x, \theta^{(t)}) \log \frac{P(x, c|\theta)}{P(c|x, \theta^{(t)})}$$

is an auxiliary function for the likelihood function $\mathcal{L}(\theta) = \log P(x|\theta)$, where x is the observed data, θ is the parameters, and c is the hidden variables in our data generation problem.

Proof.

$$\begin{aligned}
&Q(\theta, \theta^{(t)}) \\
&= \sum_{c} P(c|x, \theta^{(t)}) \log \frac{P(x|c, \theta) P(c|\theta)}{P(c|x, \theta^{(t)})} \\
&= \sum_{c} P(c|x, \theta^{(t)}) \log \frac{P(c|x, \theta) P(x|\theta)}{P(c|x, \theta^{(t)})} \\
&= \sum_{c} P(c|x, \theta^{(t)}) \log P(x|\theta) + \sum_{c} P(c|x, \theta^{(t)}) \log \frac{P(c|x, \theta)}{P(c|x, \theta^{(t)})} \\
&= \log P(x|\theta) - \underbrace{\sum_{c} P(c|x, \theta^{(t)}) \log \frac{P(c|x, \theta^{(t)})}{P(c|x, \theta)}}_{D(P(c|x,\theta^{(t)}) || P(c|x,\theta))}
\end{aligned}$$

It is straightforward to verify that $\mathcal{Q}(\theta, \theta) = \mathcal{L}(\theta)$. The last KL-divergence term is always non-negative. Therefore, we have $\mathcal{Q}(\theta, \theta^{(t)}) \leq \mathcal{L}(\theta)$. □

Lemma 3. *Maximizing the expected complete data log-likelihood*

$$\theta^{(t+1)} = \arg\max_{\theta} \sum_{c} P(c|x, \theta^{(t)}) \log P(x, c|\theta) \tag{15}$$

is equivalent to maximizing the update in Eq. (14).

Proof.

$$\begin{aligned}
Q(\theta, \theta^{(t)}) &= \sum_{c} P(c|x, \theta^{(t)}) \log \frac{P(x, c|\theta)}{P(c|x, \theta^{(t)})} \\
&= \sum_{c} P(c|x, \theta^{(t)}) \log P(x, c|\theta) \\
&\quad - \sum_{c} P(c|x, \theta^{(t)}) \log P(c|x, \theta^{(t)})
\end{aligned}$$

The second term is independent of θ, it can be treated as constant. Thus, maximizing the expected complete data log-likelihood function is equivalent to maximizing $Q(\theta, \theta^{(t)})$. □

Proof of Theorem 1. With simple derivations [16], one obtains the relevant part of the expected complete data log-likelihood function for the objective function in Eq. (2) as:

$$\bar{\mathcal{L}} = \sum_{i=1}^{N} \sum_{j=1}^{M} n(x_i, w_j) \sum_{k=1}^{K} P(c_k|x_i, w_j) \log \Big[(w_j|c_k) P(c_k|x_i) \Big]$$

and the relevant part of the expected complete data log-likelihood function for the objective function $\mathcal{O}_1 = \mathcal{L} - \lambda \mathcal{R}$ in Eq. (5) as:

$$\begin{aligned}
\bar{\mathcal{Q}}_1 = &\sum_{i=1}^{N} \sum_{j=1}^{M} n(x_i, w_j) \sum_{k=1}^{K} P(c_k|x_i, w_j) \log \Big[(w_j|c_k) P(c_k|x_i) \Big] \\
&- \frac{\lambda}{2} \sum_{i,s=1}^{N} \sum_{k=1}^{K} \Big(P_i(c_k) \log \frac{P_i(c_k)}{P_s(c_k)} + P_s(c_k) \log \frac{P_s(c_k)}{P_i(c_k)} \Big) E_{ij}
\end{aligned}$$

According to Lemma 3, it follows that maximizing the above $\bar{\mathcal{Q}}_1$ function results in exactly the update rule in Eq. (14) to the objective function $\mathcal{Q}_1$ in (5) for SSCC. In the following, we show how to obtain the update rule in Eq. (10) that maximizes the $\bar{\mathcal{Q}}_1$ function.

We determine the re-estimation of $\{P(c_k|x_i)\}$ in M-step by maximizing the $\bar{\mathcal{Q}}_1$ with the constraint: $\sum_{k=1}^{K} P(c_k|x_i) = 1$. Therefore, we augment $\bar{\mathcal{Q}}_1$ by the appropriate Lagrange multipliers ρ_i to obtain

$$\mathcal{F} = \bar{\mathcal{Q}}_1 + \sum_{i=1}^{N} \rho_i (1 - \sum_{k=1}^{K} P(c_k|x_i)). \tag{16}$$

Maximization of $\mathcal{F}$ with respect to $P(c_k|x_i)$ leads to the following set of equations:

$$\begin{aligned}
&\frac{\sum_{j=1}^{M} n(d_i, w_j) P(c_k|x_i, w_j)}{P(c_k|x_i)} - \rho_i \\
&\quad - \frac{\lambda}{2} \sum_{s=1}^{N} \Big(\log \frac{P(c_k|x_i)}{P(c_k|x_s)} + 1 - \frac{P(c_k|x_s)}{P(c_k|x_i)} \Big) E_{is} = 0 \\
&1 \leq i \leq N, \ 1 \leq k \leq K
\end{aligned} \tag{17}$$

We expect that if two instances x_i and x_s are connected (i.e., $E_{is} = 1$), then the distributions $P(c_k|x_i)$ and $P(c_k|x_s)$ are similar to each other, i.e., $P(c_k|x_i)$ will be close to $P(c_k|x_s)$. We have

$$\Big(\frac{P(c_k|x_i)}{P(c_k|x_s)} \Big)^{E_{is}} \approx 1.$$

By using the approximation

$$\log(x) \approx 1 - \frac{1}{x}, \ x \to 1, \tag{18}$$

Eq. (17) is rewritten as

$$\begin{aligned}
&\sum_{j=1}^{M} n(d_i, w_j) P(c_k|x_i, w_j) - \rho_i P(c_k|x_i) \\
&\quad - \lambda \sum_{s=1}^{N} \Big(P(c_k|x_i) - P(c_k|x_s) \Big) W_{is} = 0, \\
&1 \leq i \leq N, 1 \leq k \leq K.
\end{aligned} \tag{19}$$

By summing the above equations over all k values with respect to a given instance x_i, we obtain the Lagrange multipliers

$$\rho_i = \sum_{j=1}^{M} n(x_i, w_j), \ 1 \leq i \leq N. \tag{20}$$

Let $\mathbf{y}_k = [P(c_k|x_1), \cdots, P(c_k|x_N)]$, $\mathbf{\Lambda}$ denote the diagonal matrix, and $\mathbf{L}$ denote the graph Laplacian matrix defined in Section 3.3, the system of equations in (19) can be rewritten as

Please cite this article in press as: Q. Wu et al., Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.04.002

$$\begin{bmatrix} \sum_{j=1}^{M} n(x_1, w_j) P(c_k|x_1, w_j) \\ \vdots \\ \sum_{j=1}^{M} n(x_1, w_j) P(c_k|x_N, w_j) \end{bmatrix} - \mathbf{\Lambda} \mathbf{y}_k - \lambda \mathbf{L} \mathbf{y}_k = 0.$$

Thus, we have

$$\mathbf{y}_k = (\mathbf{\Lambda} + \lambda \mathbf{L})^{-1} \begin{bmatrix} \sum_{j=1}^{M} n(x_1, w_j) \log P(c_k|x_1, w_j) \\ \vdots \\ \sum_{j=1}^{M} n(x_1, w_j) \log P(c_k|x_N, w_j) \end{bmatrix}.$$

Setting $\mathbf{y}_k = \mathbf{y}_k^{(t)}$ and $P(c_k|x_1, w_j) = P^{(t)}(c_k|x_1, w_j)$ in the t-th iteration, the update rule takes the form as in Eq. (10).

Since $\bar{\mathcal{Q}}_1$ is the expected complete data log-likelihood function for Q_1, and the update rule in Eq. (10) is obtained by maximizing $\bar{Q}_1$, Q_1 is non-decreasing under this update. □

Proof of Theorem 2. Similarly, with simple derivations [16], one obtains the relevant part of the expected complete data log-likelihood function for the objective function $\mathcal{O}_2$ in Eq. (7) as follows:

$$\begin{aligned} \bar{\mathcal{Q}}_2 = & \sum_{i=1}^{N} \sum_{j=1}^{M} n(x_i, w_j) \sum_{k=1}^{K} P(c_k|x_i, w_j) \log \big[P(w_j|c_k) P(c_k|x_i) \big] \\ & - \frac{\lambda}{2} \sum_{i,s=1}^{N} \sum_{k,l=1}^{K} \Bigg(\Big(P_i(c_k) \log \frac{P_i(c_k)}{P_s(c_k)} + P_s(c_k) \log \frac{P_s(c_k)}{P_i(c_k)} \Big) E_{is} \\ & + \Big(P_i(c_k) \log \frac{P_i(c_k)}{P_i(c_l)} + P_i(c_l) \log \frac{P_i(c_l)}{P_i(c_k)} \Big) F_{kl} \Bigg). \end{aligned} \tag{21}$$

Again, we augment $\bar{\mathcal{Q}}_2$ by the appropriate Lagrange multipliers ρ_i to obtain

$$\mathcal{F} = \bar{\mathcal{Q}}_2 + \sum_{i=1}^{N} \rho_i \big(1 - \sum_{k=1}^{K} P(c_k|x_i) \big) \tag{22}$$

Maximizing $\mathcal{F}$ with respect to $P(c_k|x_i)$ leads to the following set of equations:

$$\begin{aligned} & \frac{\sum_{j=1}^{M} n(x_i, w_j) P(c_k|x_i, w_j)}{P(c_k|x_i)} - \rho_i \\ & - \frac{\lambda}{2} \Bigg[\sum_{s=1}^{N} \Big(\log \frac{P(c_k|x_i)}{P(c_k|x_s)} + 1 - \frac{P(c_k|x_s)}{P(c_k|x_i)} \Big) E_{is} \\ & + \sum_{l=1}^{K} \Big(\log \frac{P(c_k|x_i)}{P(c_l|x_i)} + 1 - \frac{P(c_l|x_i)}{P(c_k|x_i)} \Big) F_{kl} \Bigg] = 0 \\ & 1 \le i \le N, \ 1 \le k \le K. \end{aligned} \tag{23}$$

Based on local consistency assumptions, we have the following equations

$$\Big(\frac{P(c_k|x_i)}{P(c_k|x_s)} \Big)^{E_{is}} \approx 1, \quad \Big(\frac{P(c_k|x_i)}{P(c_l|x_i)} \Big)^{F_{kl}} \approx 1.$$

Using the approximation in Eq. (18), we rewrite Eq. (23) as

$$\begin{aligned} & \frac{\sum_{j=1}^{M} n(x_i, w_j) P(c_k|x_i, w_j)}{P(c_k|x_i)} - \rho_i - \frac{\lambda}{P(c_k|x_i)} \mathcal{A} = 0 \\ & 1 \le i \le N, \ 1 \le k \le K \end{aligned} \tag{24}$$

where $\mathcal{A}$ is

$$\begin{aligned} & \sum_{s=1}^{N} \big(P(c_k|x_i) - P(c_k|x_s) \big) E_{is} + \sum_{l=1}^{K} \big(P(c_k|x_i) - P(c_l|x_i) \big) F_{kl} \\ & = P(c_k|x_i) \sum_{s=1}^{N} E_{is} - \sum_{s=1}^{N} P(c_k|x_s) E_{is} + P(c_k|x_i) \sum_{l=1}^{K} F_{kl} \\ & \quad - \sum_{l=1}^{K} P(c_l|x_i) F_{kl}. \end{aligned}$$

Let $\mathbf{y}$ denote an NK length label probabilities vector $[\mathbf{y}_1, \cdots, \mathbf{y}_K]^T$ with $\mathbf{y}_k = [P(c_k|x_1), \cdots, P(c_k|x_N)]$, and $\mathbf{\Omega}$, $\mathbf{D}$, $\mathbf{B}$, $\mathbf{U}$, and $\mathbf{R}$, $\mathbf{Z}$ be the NK-by-NK sparse matrices defined in Section 3.3.

The system of equations in (22) is solved using the following matrix form:

$$\mathbf{Z} - \mathbf{\Omega} \mathbf{y} - \lambda (\mathbf{D} - \mathbf{B} + \mathbf{U} - \mathbf{R}) \mathbf{y} = 0. \tag{25}$$

Thus, we have

$$\mathbf{y} = (\mathbf{\Omega} + \lambda (\mathbf{D} - \mathbf{B} + \mathbf{U} - \mathbf{R}))^{-1} \mathbf{Z}. \tag{26}$$

Setting $\mathbf{y} = \mathbf{y}^{(t)}$ and $\mathbf{Z} = \mathbf{Z}^{(t)}$ in the t-th iteration, the update rule for $P^{(t)}(c_k|x_i)$ takes the form as in Eq. (11).

Since $\bar{Q}_2$ is the expected complete data log-likelihood function for Q_2, and the update rule in Eq. (11) is obtained by maximizing $\bar{Q}_2$, Q_2 is non-decreasing under this update. □

References

[1] D. Jensen, J. Neville, B. Gallagher, Why collective inference improves relational classification, in: Proc. of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 593–598.
[2] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective classification in network data, AI Mag. 29 (3) (2008) 93.
[3] Y. Xia, J. Shang, J. Chen, G.-P. Liu, Networked data fusion with packet losses and variable delays, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 39 (5) (2009) 1107–1120.
[4] J. Neville, D. Jensen, Iterative classification in relational data, in: Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data, 2000, pp. 13–20.
[5] L. McDowell, D. Aha, Semi-supervised collective classification via hybrid label regularization, in: Proc. of the 29th International Conference on Machine Learning, 2012, pp. 975–982.
[6] L.K. McDowell, D.W. Aha, Labels or attributes? Rethinking the neighbors for collective classification in sparsely-labeled networks, in: Proc. of the 22nd ACM International Conference on Information and Knowledge Management (CIKM 2013), 2013.
[7] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems, 2000, pp. 556–562.
[8] R. Shi, Q. Wu, Y. Ye, H. Shen-Shyang, A generative model with network regularization for semi-supervised collective classification, in: Proc. of the SIAM International Conference on Data Mining, 2004.
[9] S.A. Macskassy, F. Provost, Classification in networked data: a toolkit and a univariate case study, J. Mach. Learn. Res. 8 (2007) 935–983.
[10] B. Taskar, P. Abbeel, D. Koller, Discriminative probabilistic models for relational data, in: Proc. of the Eighteenth Conference on Uncertainty in Artificial Intelligence, 2002, pp. 485–492.
[11] J. Neville, D. Jensen, Relational dependency networks, J. Mach. Learn. Res. 8 (2007) 653–692.
[12] X. Shi, Y. Li, P. Yu, Collective prediction with latent graphs, in: Proc. of the 20th ACM International Conference on Information and Knowledge Management, 2011, pp. 1127–1136.
[13] B. Gallagher, H. Tong, T. Eliassi-Rad, C. Faloutsos, Using ghost edges for classification in sparsely labeled networks, in: Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 256–264.
[14] M. Bilgic, L. Mihalkova, L. Getoor, Active learning for networked data, in: Proc. of the 27th International Conference on Machine Learning, 2010, pp. 79–86.
[15] X. Kong, X. Shi, P.S. Yu, Multi-label collective classification, in: SIAM International Conference on Data Mining (SDM), 2011, pp. 618–629.
[16] T. Hofmann, Unsupervised learning by probabilistic latent semantic analysis, Mach. Learn. 42 (1–2) (2001) 177–196.

Please cite this article in press as: Q. Wu et al., Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.04.002

[17] A.P. Dempster, N.M. Laird, D.B. Rubin, et al., Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. 39 (1) (1977) 1–38.
[18] X. Jin, B.W. Wah, X. Cheng, Y. Wang, Significance and challenges of big data research, Big Data Research 2 (2) (2015) 59–64.
[19] D. Cai, X. Wang, X. He, Probabilistic dyadic data analysis with local and global consistency, in: Proc. of the 26th Annual International Conference on Machine Learning, 2009, pp. 105–112.
[20] D. Cai, Q. Mei, J. Han, C. Zhai, Modeling hidden topics on document manifold, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM'08), 2008, pp. 911–920.
[21] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1548–1560.
[22] G. Tsoumakas, I. Katakis, I. Vlahavas, Random k-label sets for multilabel classification, IEEE Trans. Knowl. Data Eng. 23 (7) (2011) 1079–1089.
[23] M. Bilgic, Cost-sensitive information acquisition in structured domains, Ph.D. thesis, University of Maryland at College Park, 2010.
[24] Q. Lu, L. Getoor, Link-based classification using labeled and unlabeled data, in: International Conference on Machine Learning, 2003.
[25] R. Xiang, J. Neville, Pseudolikelihood EM for within-network relational learning, in: Proc. of International Conferences on Data Mining, 2008, pp. 1103–1108.
[26] L. Tang, H. Liu, Relational learning via latent social dimensions, in: Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 817–826.
[27] Q. Lu, L. Getoor, Link-based classification, in: Proc. of the 20th International Conference on Machine Learning (ICML), vol. 20, 2003, pp. 496–503.
[28] L.K. McDowell, K.M. Gupta, D.W. Aha, Cautious collective classification, J. Mach. Learn. Res. 10 (2009) 2777–2836.
[29] J. Cheng, C. Hatzis, H. Hayashi, M.-A. Krogel, S. Morishita, D. Page, J. Sese, KDD Cup 2001 report, ACM SIGKDD Explor. Newsl. 3 (2) (2002) 47–64.
[30] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[31] M.-L. Zhang, Z.-H. Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng., http://doi.ieeecomputersociety.org/10.1109/TKDE.2013.39.
[32] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, Mach. Learn. 85 (3) (2011) 333–359.

Please cite this article in press as: Q. Wu et al., Multi-Label Regularized Generative Model for Semi-Supervised Collective Classification in Large-Scale Networks, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.04.002