Robust Classification of Information Networks by Consistent Graph Learning

Shi Zhi[1](✉), Jiawei Han[1], and Quanquan Gu[2]

[1] Department of Computer Science,
University of Illinois at Urbana-Champaign, Champaign, IL, USA
{shizhi2,hanj}@illinois.edu
[2] Department of Systems and Information Engineering,
University of Virginia, Charlottesville, VA, USA
qg5w@virginia.edu

Abstract. Graph regularization-based methods have achieved great success for network classification by making the label-link consistency assumption, i.e., if two nodes are linked together, they are likely to belong to the same class. However, in a real-world network, there exist links that connect nodes of different classes. These inconsistent links raise a big challenge for graph regularization and deteriorate the classification performance significantly. To address this problem, we propose a novel algorithm, namely *Consistent Graph Learning*, which is robust to the inconsistent links of a network. In particular, given a network and a small number of labeled nodes, we aim at learning a consistent network with more consistent and fewer inconsistent links than the original network. Since the link information of a network is naturally represented by a set of relation matrices, the learning of a consistent network is reduced to learning consistent relation matrices under some constraints. More specifically, we achieve it by joint graph regularization on the nuclear norm minimization of consistent relation matrices together with ℓ_1-norm minimization on the difference matrices between the original relation matrices and the learned consistent ones subject to certain constraints. Experiments on both homogeneous and heterogeneous network datasets show that the proposed method outperforms the state-of-the-art methods.

Keywords: Robust classification · Information network · Consistent link · Consistent network · Consistent Graph Learning

1 Introduction

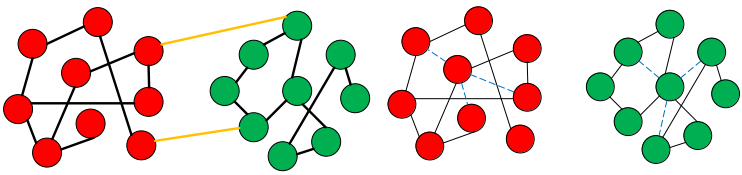
Information networks have been found to play increasingly important role in real-life applications. Generally speaking, information networks can be categorized into two families: (1) homogeneous information networks where there is only one type of nodes and links. Examples include friendship network in Facebook[1], co-author and citation network in DBLP[2], and the World Wide Web;

[1] http://www.facebook.com
[2] http://www.informatik.uni-trier.de/~ley/db/

A. Appice et al. (Eds.): ECML PKDD 2015, Part II, LNAI 9285, pp. 752–767, 2015.
DOI: 10.1007/978-3-319-23525-7_46

and (2) heterogeneous information networks where there exist multiple types of nodes and links. A bibliographic information network is an example of heterogeneous information network, which contains four types of objects: papers, authors, conferences and terms. Papers and authors are linked by the relation of "written by" and "write". Papers and conferences are linked by "published in" and "publish". Papers and terms are linked by "contain" and "contained in".

In the past decade, many methods have been proposed for classification of both homogeneous information networks [5,9,11,13,14,18–21] and heterogeneous information networks [7,8], which are based on the link structure and the node content of networks. Among these methods, graph regularization-based methods [5,8,9,18,19,21] have achieved superior performance over other methods. These methods assume that if two nodes are linked in a network, their labels are likely to be the same. Start from a small number of labeled nodes, labels are propagated along linking nodes to preserve the local consistency. Therefore, they heavily depend on the link structure of a network and implicitly require the links of the network to be consistent with node labels. However, in many cases, this requirement is not satisfied. For example, in Figure 1(a), there are two classes of nodes denoted by different colors. The black edge links two nodes of the same class, while the yellow edge links node from different classes. We define black link as *Consistent Link*, and yellow link as *Inconsistent Link*. Due to the existence of inconsistent links, graph regularization-based methods may fail to correctly classify the nodes residing on both sides of the inconsistent edges. In our study, we call the network with inconsistent links as *Inconsistent Network*. Since inconsistent links are prevalent in real-world networks, it is of central importance to develop learning models for classification of inconsistent networks.



(a) Original Inconsistent Network (b) Consistent Network

Fig. 1. An example of (a) Inconsistent Network, and (b) Consistent Network. There are two classes of nodes denoted by red and green. The black links are consistent with the labels, while the nodes linked by yellow links have different labels. The blue dashed links are added consistent links. The goal is to remove red links and add blue links.

Intuitively, if there are no inconsistent links, e.g., the yellow edges in Figure 1(a), graph regularization-based classification methods [18] can achieve good results. This motivates us to handle an inconsistent network in the following two ways. First, if we can detect which links are inconsistent, we can delete these inconsistent links. Second, if we can add more consistent links between the nodes of the same classes, we can compensate the effect of the inconsistent links as well. It is desirable to get a network as shown in Figure 1 (b), where we

remove the inconsistent yellow edges and add the dashed blue edges, i.e., consistent links. Based on this modified network, graph regularization-based methods may work better than using original relation matrices. In our study, we will show that we can learn an approximately consistent network using a small number of labeled data under certain constraints.

In this paper, based on the above discussion, we propose a novel regularization technique, namely *Consistent Graph Learning*, which is robust to those inconsistent links of a network. Our goal is to learn an approximately consistent network based on a small number of labeled data. Since the link information of a network can be naturally represented by a set of relation matrices, the learning of a consistent network can be transformed into learning consistent relation matrices. More specifically, we assume that each original relation matrix can be decomposed into a consistent relation matrix and a residue matrix. In a fully consistent network, each pair of nodes of the same class are linked while those of different classes are not linked. Though the real-world network is usually sparse, nodes in a consistent network connect much more to the nodes of the same class rather than a different class. Thus, consistent relation matrix intrinsically has the low-rank property. We can achieve this low-rank characteristics by applying nuclear-norm minimization on consistent relation matrix. By doing this, more consistent links are added to the original inconsistent network. On the other hand, since in real-world network nodes of the same class tends to have much more links than those of different classes do, the consistent network should be similar to the original network and the norm of the residue matrix should be small. To remove inconsistent links and keep consistent links, we aim to have a sparse residue matrix with non-zero elements as fewer as possible instead of changing the value of every element in the original relation matrix. It can be achieved by minimizing ℓ_1-norm of the residue matrix. In summary, to satisfy both requirements, we perform a joint graph regularization on the consistent relation matrix with nuclear-norm minimization, and the residue matrix with ℓ_1-norm minimization, subject to the constraint that the sum of the consistent relation matrix and the residue matrix equals to the original relation matrix, and each element of consistent matrix is within a certain range. Given a set of labeled data, our model can learn the consistent network by alternating direction method of multipliers [2] (ADMM) method that solves a convex optimization problem by breaking it into smaller pieces, each of which is easier to handle. We can use the consistent network to classify all the other nodes by any network classification method. Experiments on both homogeneous and heterogeneous network datasets show that the proposed method outperforms the state-of-art methods.

The main contributions of this paper are as follows: (1) We raise and analyze the inconsistency of real-world networks; (2) we propose a consistent graph learning technique which is able to learn an approximately consistent network given a small number of labeled data; and (3) we validate the effectiveness of the proposed method on both homogeneous and heterogeneous networks. The remainder of this paper is organized as follows. In Section 2 we present a model for classification of the information networks with inconsistent links. In Section 3,

we discuss several related work to our method. The experiments on Cora and DBLP datasets are demonstrated in Section 4. Finally, we draw a conclusion and point out the future work in Section 5.

2 The Proposed Method

In this section, we present Consistent Graph Learning for semi-supervised classification of information networks. Before going deep into the proposed method, we first present some preliminary definitions of information network.

2.1 Preliminary Definitions

Definition 1. *An information network consists of m types of objects $\mathcal{X}^{kl} = \{\mathcal{X}^k\}_{k=1}^m$, where $\mathcal{X}^k$ is a set of objects belonging to the k-th type. A weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, R)$ is called an* ***information network*** *on objects $\mathcal{X}$, if $\mathcal{V} = \mathcal{X}$, $\mathcal{E}$ is a binary relation on $\mathcal{V}$, and $R : \mathcal{E} \to \mathbb{R}$ is a weight function mapping from an edge $e \in \mathcal{E}$ to a real number $w \in \mathbb{R}$. Specially, we call such an information network* ***heterogeneous network*** *when $m \geq 2$; and* ***homogeneous network*** *when $m = 1$.*

We can treat homogeneous information network as a special case of heterogeneous information network. The crucial difference of using heterogeneous network is that we work on each relation matrix between two types of nodes instead of working on a large relation matrix between all nodes of different types. Therefore, we will introduce the proposed method in the context of heterogeneous network, which is more general. Now we present the formal definitions of *Consistent Link* and *Consistent Network*.

Definition 2. *A link is* ***consistent*** *if the nodes it connects belong to the same class. An information network is* ***consistent*** *if and only if all of its links are consistent.*

The definitions of *Inconsistent Link* and *Inconsistent Network* can be deduced analogously, hence we omit them. Note that the definitions in this paper are specific to our problem, i.e., classification of networks. There may exist other definitions of *Consistent Link* and *Consistent Network* in the literature.

2.2 Notation

A heterogeneous network can be represented by a collection of relation matrices, each of which models the pairwise relation between a node in one type and another node in a different type. Mathematically speaking, in a heterogeneous information network, suppose there are m types of entities, i.e., $\mathcal{X}^k, 1 \leq k \leq m$, where $\mathcal{X}^k = \{x_1^k, \ldots, x_n^k\}$. A relation graph $\mathcal{G}^{kl}$ can be built corresponding to each type of link relationships between two types of data entities $\mathcal{X}^k$ and $\mathcal{X}^l$, $1 \leq k \leq m$. Let $\mathbf{R}^{kl}$ be an $n_k \times n_l$ relation matrix corresponding to graph $\mathcal{G}$,

in which R_{ij}^{kl} denotes the weight on link from x_i^k to x_j^l. Note that $\mathbf{R}^{kl}$ is not symmetric. One possible definition of $\mathbf{R}^{kl}$ is as follows.

$$R_{ij}^{kl} = \begin{cases} 1 \text{ if there is a link from } x_i^k \text{ to } x_j^l \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{1}$$

If we consider a weighted graph, the definition of $\mathbf{R}^{kl}$ can be extended to

$$R_{ij}^{kl} = \begin{cases} m \text{ if there are } m \text{ links from } x_i^k \text{ to } x_j^l \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{2}$$

Suppose there are c classes, in order to encode label information of each type, we basically define a label matrix for each type, i.e., $\mathbf{Y}^k \in \mathbb{R}^{n_k \times c}$, such that

$$Y_{il}^k = \begin{cases} 1 \text{ if } x_i^k \text{ is labeled to the } l\text{-th class} \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{3}$$

Note that if x_i^k is unlabeled, then $Y_{il}^k = 0$ for $\forall l$.

For each type of objects, we are going to learn a class assignment matrix $\mathbf{F}^k \in \mathbb{R}^{n_k \times c}$, whose definition is similar to $\mathbf{Y}^k$. We denote the i-th row of $\mathbf{Y}^k$ by $\mathbf{Y}_{i\cdot}^k$, and the i-th row of $\mathbf{F}^k$ by $\mathbf{F}_{i\cdot}^k$. For a matrix $\mathbf{E}^{kl}$, its ℓ_1-norm is defined as $||\mathbf{E}||_1 = \sum_{ij} |E_{ij}|$. For a $m \times n$ matrix $\mathbf{W}$, its nuclear norm is defined as $||\mathbf{W}||_* = \sum_i^{\min\{m,n\}} \sigma_i$, where $\mathbf{W} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ is the Singular Value Decomposition (SVD) of $\mathbf{W}$, $(\boldsymbol{\Sigma})_{ii} = \sigma_i$. For a matrix $\mathbf{D}$, its Frobenius norm is defined as $||\mathbf{D}||_F = \sqrt{\sum_{ij} D_{ij}^2}$. Notation $\circ$ is used to get the entry-wise product of two matrices, e.g., $\mathbf{D} \circ \mathbf{E}$ is a matrix whose each element equals to $D_{ij}E_{ij}$. Matrix $\mathbf{0}$ is a matrix of all zeros, and matrix $\mathbf{1}$ is a matrix of all ones.

2.3 Standard Graph Regularization

The basic assumption of graph regularization is that if two objects x_i^k and x_j^l are linked together, then their labels F_{ip}^k and F_{jp}^l are likely to be the same. It can be mathematically formulated as [16],

$$\min_{\mathbf{F}^k, \mathbf{F}^l} \frac{1}{2} \sum_{i=1}^{n_k} \sum_{j=1}^{n_l} ||\mathbf{F}_{i\cdot}^k - \mathbf{F}_{j\cdot}^l||_2^2 R_{ij}^{kl}, \tag{4}$$

where $\mathbf{R}^{kl}$ could either be the original relation matrix or the normalized one. As we can see, if $R_{ij}^{kl} > 0$, Eq. (4) will push the label of x_i^k and the label of x_j^l close. If $R_{ij}^{kl} = 0$, the labels are determined by other terms of the objective function. This is the rationale of graph regularization. To give an example, in a homogeneous citation network ($k = l = 1$), if the i-th paper cites the j-th paper, standard graph regularization tends to classify these two papers into the same class. However, we will show that it is not true in real world citation dataset when there are inconsistent links, as we will show in the experiments. In this case, graph regularization would fail. This shows the drawback of standard graph regularization technique for network classification.

2.4 Consistent Graph Learning

The basic idea of our method is to learn an approximately consistent network, based on which we apply graph regularization and learn a classifier. Since a heterogeneous information network can be represented by a set of relation matrices, i.e., $\{\mathbf{R}^{kl}\}$, the learning of a consistent network can be transformed into learning a set of consistent relation matrices, i.e., $\{\mathbf{W}^{kl}\}$. Here we say a relation matrix is consistent if and only if its corresponding network is consistent. For each relation matrix $\mathbf{R}^{kl}$ where there may exist some entries which are inconsistent with the labels, we decompose it into a consistent relation matrix $\mathbf{W}^{kl}$ whose entries are consistent with the node labels, and a residue matrix $\mathbf{E}^{kl}$ whose entries are inconsistent with the labels. Hereafter, we call $\mathbf{W}^{kl}$ as *Consistent Relation Matrix* and $\mathbf{E}^{kl}$ as *Residue Matrix*. It is mathematically described as

$$\mathbf{R}^{kl} = \mathbf{W}^{kl} + \mathbf{E}^{kl}, \mathbf{0} \leq \mathbf{W}^{kl} \leq \max(\mathbf{1}, \mathbf{R}^{kl}), \tag{5}$$

where the function max takes the larger value of 1 and each element in $\mathbf{R}^{kl}$. We add a box constraint on $\mathbf{W}^{kl}$ to make it both lower and upper-bounded. The reason is that the original relation matrix $\mathbf{R}^{kl}$ is bounded. We hope that the learned consistent relation matrix $\mathbf{W}^{kl}$ is also bounded.

In order to make the learned relation matrix $\mathbf{W}^{kl}$ consistent, we need to specify additional constraints as follows:

1. $\mathbf{W}^{kl}$ should be consistent with the labeled data $\mathbf{Y}$. It can be achieved by standard graph regularization on $\mathbf{W}^{kl}$ with respect to $\mathbf{Y}$.
2. We assume that the number of inconsistent links is only a portion of the links in $\mathbf{R}^{kl}$. Hence, we require the residue matrix to be sparse. To obtain this goal, we apply ℓ_1-norm minimization to $\mathbf{E}^{kl}$.
3. In principal, we prefer not to remove links that connect nodes of large degree because removing such links may take a risk to disconnect more unlabeled nodes with labeled ones such that some labels cannot be propagated through the consistent links. To handle this issue, we take entry-wise product of $\mathbf{E}^{kl}$ in the ℓ_1-norm term and $\mathbf{D}^{kl}$, where $D^{kl}_{ij} = \sqrt{d_i d_j}$, $d_i = \sum_i R^{kl}_{ij}$ is the out-degree of node x^k_i and $d_j = \sum_j R^{kl}_{ij}$ is the in-degree of node x^k_j.
4. As we mentioned before, nodes in a consistent relation matrix connect much more to the nodes of the same class rather than different class. To pursue the low-rank property of consistent relation matrix $\mathbf{W}^{kl}$, we apply nuclear norm minimization to $\mathbf{W}^{kl}$. Note that there may be some extreme cases when a low-rank matrix is not necessarily a consistent matrix (e.g., an all-ones matrix). However, we can prevent our method from converging to these cases by balancing different regularizations.

We bring in an auxiliary $\mathbf{Q}^{kl}$ and let it be equal to $\mathbf{W}^{kl}$. The advantage of it is to allows us to solve nuclear norm minimization and box constraint in separate steps, which is easy to solve. Putting all the above constraints together,

we obtain

$$\begin{aligned} \min_{\{\mathbf{Q}^{kl},\mathbf{W}^{kl},\mathbf{E}^{kl}\}} \quad & \sum_{i=1}^{n_k}\sum_{j=1}^{n_l} ||\mathbf{Y}^k_{i\cdot} - \mathbf{Y}^l_{j\cdot}||_2^2 W^{kl}_{ij} + \gamma_{kl}||\mathbf{D}^{kl} \circ \mathbf{E}^{kl}||_1 + \beta_{kl}||\mathbf{W}^{kl}||_* \\ \text{subject to} \quad & \mathbf{R}^{kl} = \mathbf{W}^{kl} + \mathbf{E}^{kl}, \mathbf{W}^{kl} = \mathbf{Q}^{kl}, \mathbf{0} \le \mathbf{Q}^{kl} \le \max(\mathbf{1}, \mathbf{R}^{kl}) \end{aligned} \tag{6}$$

where $\gamma_{kl} > 0$ and $\beta_{kl} > 0$ are regularization parameters that controls the sparsity of $\mathbf{E}^{kl}$ and the low-rank property of $\mathbf{W}^{kl}$, respectively. These two parameters essentially control the balance among the label-link consistency, sparsity and low-rank property. The larger γ_{kl} is, the sparser $\mathbf{E}^{kl}$ will be. Larger β_{kl} forces the nuclear norm of $\mathbf{W}^{kl}$ to be smaller. We call the model in Eq. (6) as *Consistent Graph Learning*. Note that if we set $\gamma_{kl} = \infty$ and $\beta_{kl} = 0$, $\mathbf{W}^{kl}$ will be exactly equal to $\mathbf{R}^{kl}$. If we have prior knowledge indicating some of the relation matrices $\mathbf{R}^{kl}$ are consistent, we can set the corresponding γ_{kl} to ∞ and $\beta_{kl} = 0$. Note that we cannot guarantee that learned $\mathbf{W}^{kl}$ is totally consistent because we only have partial labels of the nodes, but the learned relation matrix has fewer inconsistent links and more consistent links than the original one. In the following, we will introduce how to solve it.

2.5 Optimization

Due to the decomposition equality constraint in Eq. (6), we use the alternating direction method of multipliers [2] (ADMM). We will derive an algorithm based on ADMM for solving Eq. (6). Before that, we first briefly introduce augmented Lagrangian multiplier [3] method. Augmented Lagrangian [3] (ALM) is a method for solving equality constrained optimization problem. It reformulates the problem into an unconstrained one by adding Lagrangian multipliers and an extra quadratic penalty term for each equality constraint.

As to our method, the augmented Lagrangian function is as follows by ignoring the inequality constraints on $\mathbf{E}^{kl}$

$$\begin{aligned} L(\mathbf{Q}^{kl}, \mathbf{W}^{kl}, \mathbf{E}^{kl}, \mathbf{Z}^{kl}) = & \sum_{i=1}^{n_k}\sum_{j=1}^{n_l} ||\mathbf{Y}^k_{i\cdot} - \mathbf{Y}^l_{j\cdot}||_2^2 W^{kl}_{ij} + \beta_{kl}||\mathbf{W}^{kl}||_* + \gamma_{kl}||\mathbf{D}^{kl} \circ \mathbf{E}^{kl}||_1 \\ & + \mathrm{tr}((\mathbf{Z}^{kl})^T(\mathbf{W}^{kl} + \mathbf{E}^{kl} - \mathbf{R}^{kl})) + \frac{\mu}{2}||\mathbf{W}^{kl} + \mathbf{E}^{kl} - \mathbf{R}^{kl}||_F^2 \\ & + \mathrm{tr}((\mathbf{X}^{kl})^T(\mathbf{W}^{kl} - \mathbf{Q}^{kl})) + \frac{\zeta}{2}||\mathbf{W}^{kl} - \mathbf{Q}^{kl}||_F^2, \end{aligned} \tag{7}$$

where $\mathbf{Z}^{kl}$ and $\mathbf{X}^{kl}$ are Lagrangian multipliers, μ and ζ are penalty parameters.

In the following, we will derive the updating formula for each variable. In other words, we solve each variable when fixing the other variables. This is also known as alternating direction method of multipliers [2] (ADMM).

Computation of $\mathbf{W}^{kl}$. Given other variables fixed, the optimization of Eq. (7) with respect to $\mathbf{W}^{kl}$ is reduced to

$$\begin{aligned}\min_{\mathbf{W}^{kl}} \quad & \mathrm{tr}((\mathbf{S}^{kl})^T\mathbf{W}^{kl}) + \beta_{kl}||\mathbf{W}^{kl}||_* \\ & +\mathrm{tr}((\mathbf{Z}^{kl})^T(\mathbf{W}^{kl}+\mathbf{E}^{kl}-\mathbf{R}^{kl})) + \frac{\mu}{2}||\mathbf{W}^{kl}+\mathbf{E}^{kl}-\mathbf{R}^{kl}||_F^2 \\ & +\mathrm{tr}((\mathbf{X}^{kl})^T(\mathbf{W}^{kl}-\mathbf{Q}^{kl})) + \frac{\zeta}{2}||\mathbf{W}^{kl}-\mathbf{Q}^{kl}||_F^2, \end{aligned} \tag{8}$$

where the matrix $\mathbf{S}^{kl}$ is defined as $S_{ij}^{kl} = ||\mathbf{Y}_{i\cdot}^{k} - \mathbf{Y}_{j\cdot}^{l}||_2^2$. Eq. (8) is equivalent to

$$\min_{\mathbf{W}^{kl}} \quad \frac{\beta_{kl}}{\mu+\zeta}||\mathbf{W}^{kl}||_* + \frac{1}{2}||\mathbf{W}^{kl}-\mathbf{A}^{kl}||_F^2, \tag{9}$$

where

$$\mathbf{A}^{kl} = \frac{\zeta\mathbf{Q}^{kl} - \mu(\mathbf{E}^{kl}-\mathbf{R}^{kl}) - \mathbf{S}^{kl} - \mathbf{Z}^{kl} - \mathbf{X}^{kl}}{\mu+\zeta}. \tag{10}$$

Eq. (9) has a closed-form solution

$$\mathbf{W}^{kl} = \mathbf{U}\boldsymbol{\Sigma}_*\mathbf{V}^T, \tag{11}$$

where $\mathbf{A}^{kl} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ is the SVD of $\mathbf{A}^{kl}$, and $\boldsymbol{\Sigma}_*$ is the diagonal with $(\boldsymbol{\Sigma}_*)_{ii} = \max\{0, (\boldsymbol{\Sigma})_{ii} - \beta_{kl}/(\mu+\zeta)\}$. By setting small singular values to zero, the nuclear norm of $\mathbf{W}^{kl}$ is reduced.

Computation of $\mathbf{E}^{kl}$. Given other variables fixed, the optimization of Eq. (7) with respect to $\mathbf{E}^{kl}$ boils down to

$$\begin{aligned}\min_{\mathbf{E}^{kl}} \quad & \gamma_{kl}||\mathbf{D}^{kl}\circ\mathbf{E}^{kl}||_1 \\ & +\mathrm{tr}((\mathbf{Z}^{kl})^T(\mathbf{W}^{kl}+\mathbf{E}^{kl}-\mathbf{R}^{kl})) + \frac{\mu}{2}||\mathbf{W}^{kl}+\mathbf{E}^{kl}-\mathbf{R}^{kl}||_F^2, \end{aligned} \tag{12}$$

which is equivalent to

$$\min_{\mathbf{E}^{kl}} \quad \frac{\gamma_{kl}}{\mu}||\mathbf{D}^{kl}\circ\mathbf{E}^{kl}||_1 + \frac{1}{2}||\mathbf{E}^{kl}+\mathbf{W}^{kl}-\mathbf{R}^{kl}+\frac{1}{\mu}\mathbf{Z}^{kl}||_F^2. \tag{13}$$

Eq. (13) has a closed-form solution as follows,

$$E_{ij}^{kl} = \begin{cases} B_{ij}^{kl} - \frac{\gamma_{kl}D_{ij}^{kl}}{\mu} & \text{if } B_{ij}^{kl} \geq \frac{\gamma_{kl}D_{ij}^{kl}}{\mu} \\ 0 & \text{if } -\frac{\gamma_{kl}D_{ij}^{kl}}{\mu} < B_{ij}^{kl} < \frac{\gamma_{kl}D_{ij}^{kl}}{\mu} \\ B_{ij}^{kl} + \frac{\gamma_{kl}D_{ij}^{kl}}{\mu} & \text{if } B_{ij}^{kl} \leq -\frac{\gamma_{kl}D_{ij}^{kl}}{\mu} \end{cases}, \tag{14}$$

where $\mathbf{B}^{kl} = -\mathbf{W}^{kl} + \mathbf{R}^{kl} - \frac{1}{\mu}\mathbf{Z}^{kl}$. This step essentially only allows E_{ij}^{kl} to be non-zero when falling out of a certain range. We can see that by introducing matrix $\mathbf{D}_{ij}^{kl}$, E_{ij}^{kl} is more likely to be zero if D_{ij}^{kl} is larger. Thus, the link between nodes of large in-degree and out-degree will be less likely to be removed.

Computation of $\mathbf{Q}^{kl}$. Given other variables fixed, the optimization of Eq. (7) with respect to $\mathbf{Q}^{kl}$ boils down to

$$\begin{aligned} \min_{\mathbf{Q}^{kl}} \quad & \mathrm{tr}((\mathbf{X}^{kl})^T(\mathbf{W}^{kl} - \mathbf{Q}^{kl})) + \frac{\zeta}{2}||\mathbf{W}^{kl} - \mathbf{Q}^{kl}||_F^2 \\ \text{subject to} \quad & \mathbf{0} \leq \mathbf{Q}^{kl} \leq \max(\mathbf{1}, \mathbf{R}^{kl}), \end{aligned} \tag{15}$$

which has a closed-form solution

$$Q_{ij}^{kl} = \begin{cases} R_{ij}^{kl} & Q_{ij}^{kl} \geq \max(1, R_{ij}^{kl}) \\ W_{ij}^{kl} + \frac{1}{\zeta} X_{ij}^{kl} & 0 < Q_{ij}^{kl} < \max(1, R_{ij}^{kl}) \\ 0 & Q_{ij}^{kl} \leq 0 \end{cases}. \tag{16}$$

By making $\mathbf{Q}^{kl} = \mathbf{W}^{kl}$ and adding a box constraint on $\mathbf{Q}^{kl}$, $\mathbf{W}^{kl}$ is essentially upper and lower-bounded.

Computation of $\mathbf{Z}^{kl}$ and $\mathbf{X}^{kl}$. Taking the derivative of L with respect to $\mathbf{Z}^{kl}$ and $\mathbf{X}^{kl}$, we obtain

$$\frac{\partial L}{\partial \mathbf{Z}^{kl}} = \mathbf{W}^{kl} + \mathbf{E}^{kl} - \mathbf{R}^{kl} \quad \text{and} \quad \frac{\partial L}{\partial \mathbf{X}^{kl}} = \mathbf{W}^{kl} - \mathbf{Q}^{kl}, \tag{17}$$

which leads to the following updating formula for Lagrangian multiplier $\mathbf{Z}^{kl}$,

$$\mathbf{Z}^{kl} = \mathbf{Z}^{kl} + \mu(\mathbf{W}^{kl} + \mathbf{E}^{kl} - \mathbf{R}^{kl}). \tag{18}$$

Similarly, the updating formula for Lagrangian multiplier is $\mathbf{X}^{kl}$,

$$\mathbf{X}^{kl} = \mathbf{X}^{kl} + \zeta(\mathbf{W}^{kl} - \mathbf{Q}^{kl}). \tag{19}$$

In summary, we present the algorithm in Algorithm 1. In our experiments, we set $\mu = 10$ and $\zeta = 10$, which leads to fast convergence. In addition, we initialize $\mathbf{W}^{kl}$ as the original relation matrix $\mathbf{R}^{kl}$ with a small perturbation by adding a random matrix $\mathbf{M}^{kl}$ to $\mathbf{W}^{kl}$. Note that the random perturbation matrix helps the convergence of the ADMM algorithm [17]. We can see that in each outer iteration of Algorithm 1, it learns the underlying consistent network between $\mathcal{X}^k$ and $\mathcal{X}^l$, i.e., $\mathbf{W}^{kl}$ by ADMM. Note that k and l can be either the same or different. We can see later the learned consistent matrices can improve the accuracy of classification in the next step.

2.6 Estimation of Unlabeled Data

After we compute the consistent relation matrix $\mathbf{W}^{kl}$, we can apply existing semi-supervised classification algorithms to estimate the unlabeled data. In the experiment, we use LLGC [18] for homogeneous network classification and GNetMine [8] for heterogeneous network classification. In the experiments, for the citation and co-author sub-networks, we transform the learned consistent relation matrix into a symmetric one by setting W_{ij}^{kl} to the larger element between W_{ij}^{kl} and W_{ji}^{kl}. We do the same symmetrization on original relation matrix for input of LLGC and GNetMine.

Algorithm 1 Robust Classification of Network by *Consistent Graph Learning* (CGL)

Input: $\mathbf{R}^{kl}$, $\beta_{kl} > 0$, $\gamma_{kl} > 0$, $\mathbf{Y}$, μ, ζ;
Output: $\mathbf{W}^{kl}$, $\mathbf{E}^{kl}$, $k, l = 1, \ldots, m$;
for $k, l = 1 \rightarrow m$ **do**
  Initialize $\mathbf{W}^{kl} = \mathbf{R}^{kl} + \mathbf{M}^{kl}$, $\mathbf{Z}^{kl}$, $\mathbf{X}^{kl}$
  repeat
    Compute $\mathbf{W}^{kl}$ as in Eq. (11)
    Compute $\mathbf{E}^{kl}$ as in Eq. (14)
    Compute $\mathbf{Q}^{kl}$ as in Eq. (16)
    Compute $\mathbf{Z}^{kl}$ as in Eq. (18)
    Compute $\mathbf{X}^{kl}$ as in Eq. (19)
  until Convergence
end for

2.7 Analysis

The convergence of Algorithm 1 is stated in the following theorem.

Theorem 1. *Algorithm 1 is theoretically guaranteed to converge to the global minima of the problem in Eq. (6).*

Proof sketch: The global convergence of the algorithm can be proved using the technique in [10] [6].

Now we analyze the time complexity of Algorithm 1. Let c be the number of classes, $|V|$ denote the total number of objects, and $|E|$ denote the total number of links in the information network. In each inner iteration of Algorithm 1, it takes $O(n^2c)$ to update $\mathbf{W}^{kl}$, $O(|E|)$ to update $\mathbf{Q}^{kl}$, $O(|E|)$ to update $\mathbf{E}^{kl}$, and $O(|E|)$ to update $\mathbf{X}^{kl}$ and $\mathbf{Z}^{kl}$. Hence the total time complexity of Algorithm 1 is $O\big(T(|E| + n^2c)\big)$, where T is the average number of inner iterations. In our empirical study, we found that algorithm usually converges within 30 iterations.

3 Related Work

In this section, we review some work which are closely related to our study.

Classification of information networks has been extensively studied in the past decade. Earlier studies mainly focus on the homogeneous network. For example, [18,21] studied classification of undirected networks while [19] studied classification of directed networks. [20] proposed link-content matrix factorization (LCMF) method, which integrates content and link information into a joint matrix factorization framework. Sen et al. [14] studied collective classification of networked data. Li and Yeung [9] proposed probabilistic relational principal component analysis (PRPCA), which is the state-of-the-art subspace learning method for networks. More recently, [1,15] suggested active learning for networked data, whose goal is to minimize the labeling effort while maximize the classification accuracy. Gu and Han [5] proposed a feature selection approach for

homogeneous networked data, which selects a subset of features, such that they are consistent with the link structure of the network. Recently, classification of heterogeneous information networks received increasing attention. For instance, as a natural generalization of [18], Ji et al. [8] proposed a model for classification of heterogeneous networks. Later, Ji et al. [7] proposed to integrate ranking and classification for heterogeneous networks, where they pay more attention to the nodes whose ranking scores are higher. All the methods mentioned above are heavily depending on the link structure of the network. They should perform well if we remove inconsistent links and add consistent links. However, their classification performance is limited when the networks are inconsistent. This motivates us to develop a new model which is robust to the inconsistent links and performs well on inconsistent networks.

We notice that Chen et. al [4] proposed a similar technique for sparse graph clustering. However, their method does not take into account the label information and heavily relies on the planted partition model assumption. Luo et. al [12] proposed a similar method namely forging the graph, while their method does not leverage label information either.

4 Experiments

In this section, we empirically evaluate the effectiveness of the proposed method. All the experiments are performed on a PC with Intel Core i5 3.20G CPU and 48GB RAM.

4.1 Data Sets

In our experiments, we use two benchmark datasets: one is a homogeneous citation network, the other is a heterogeneous bibliographic network.

Cora: It contains the abstracts and references of about 34,000 research papers from the computer science community. The task is to classify each paper into one of the subfields of data structure (DS), hardware and architecture (HA), machine learning (ML), and programming language (PL), based on the citation relation between the papers. We only use the link information of this dataset. The statistics about the Cora data set are summarized in Table 1. Before we run all the baselines and our algorithm, we first make adjacent matrices symmetric, i.e. set $r'_{ij} = \max(r_{ij}, r_{ji})$.

Table 1. Description of the Cora dataset

Data Sets	#samples	#links	#classes
DS	751	1283	9
HA	400	793	7
ML	1617	4046	7
PL	1575	4918	9

DBLP: We extract a sub-network of the DBLP data set on four areas: database, data mining, information retrieval and artificial intelligence, which naturally form four classes. By selecting five representative conferences in each area, papers published in these conferences, the authors of these papers and the terms that appeared in the titles of these papers, we obtain a heterogeneous information network that consists of four types of objects: paper, conference, author and term. Within that heterogeneous information network, we have four types of link relationships: paper-conference, paper-author, paper-term and author-term. The data set we used contains 14376 papers, 20 conferences, 6401 authors and 4483 terms, with a total number of 192003 links. For evaluation, we use a labeled data set of 2876 authors, 100 papers and all 20 conferences. The statistics about the DBLP data set are summarized in Table 2.

Table 2. The statistics of the DBLP dataset

#paper	14376	#paper-author	33720
#author	6401	#paper-conference	14376
#conference	20	#paper-term	110187
#term	4483	#author-term	33720

4.2 Baselines and Parameter Settings

We compare the proposed method with the state-of-the-art network classification algorithms. The methods and their parameter settings are summarized as follows.

Network-only Link-based Classification (nLB). [13] We use network-only derivative of nLB because local features are not available in our problem. We use the implementation from NetKit-SRL[3].

Weighted-vote Relational Neighbor Classifier (wvRN). [13] We only create a feature vector for each node based on the structure information and use the implementation from NetKit-SRL.

Learning with Local and Global Consistency (LLGC). [18] LLGC is a graph-based transductive classification algorithm. The regularization parameter is tuned by searching the grid $\{0.01, 0.1, 1, 10, 100\}$.

GNetMine. [8] GNetMine is a heterogeneous generalization of LLGC. According to [8], we set the regularization parameters λ to be the same for every pair of k, l, and tune it by searching the grid $\{0.01, 0.1, 1, 10, 100\}$. It uses three relation matrices: paper-author, paper-conference and paper-term.

Consistent Graph Learning (CGL). The regularization parameters β_{kl} are set to be the same for all k, l, and tuned by searching the grid $\{0.1 : 0.1 : 1\}$ and $\{1 : 1 : 20\}$. Similarly, we turn regularization parameters γ_{kl} by searching the grid $\{1 : 1 : 10\}$ and $\{10 : 10 : 100\}$. After learning the consistent relation matrices,

[3] http://netkit-srl.sourceforge.net

Table 3. Classification Accuracy (%) on the Cora dataset

subset	DS			HA			ML			PL		
#labeled node	20%	50%	80%	20%	50%	80%	20%	50%	80%	20%	50%	80%
nLB	39.49	41.08	57.87	38.50	48.27	64.13	45.14	55.32	60.09	46.82	58.75	61.40
wvRN	45.56	58.97	64.93	42.56	57.24	67.38	49.97	60.77	63.69	50.90	59.76	64.84
LLGC	62.77	73.39	77.32	70.38	81.71	83.50	73.54	81.05	82.79	66.66	75.33	76.16
CGL	**64.39**	**77.02**	**81.45**	**79.72**	**87.31**	**88.38**	**76.26**	**82.68**	**84.94**	**69.54**	**76.90**	**80.68**

we use LLGC for Cora and GNetMine for DBLP to estimate the unlabeled data. Parameters are tuned in the same way as LLGC and GNetMine.

The searching grids are set based on heuristics. We found for CGL, the balance of different regularization terms is more important than the absolute values. Note that we do not compare our method with [1,5,7,9,20] because they either need the content information of the nodes or come from a different line of ideas.

4.3 Classification Results on Cora

For each subset of Cora dataset, we randomly choose 20%, 50%, 80% objects as labeled samples, and the rest as test samples. We repeat the selection 10 times and report the average result.

The semi-supervised classification results on the Cora data are shown in Table 3. We can see that the proposed method outperforms the other methods significantly on all the subsets with different proportion of labeled samples. More specifically, considering that LLGC is the special case of our method without relation matrix learning, it indicates that finding the consistent network is of essential importance for classification of network.

4.4 Classification Results on DBLP

For DBLP dataset, according to [8], we randomly choose 0.1%, 0.2%, 0.3%, 0.4%, 0.5% of authors and papers, and use their label information in the classification task. When applying LLGC to heterogeneous network, we tried different settings. In detail, when classifying authors and papers, we tried constructing homogeneous author-author (A-A) and paper-paper (P-P) subnetworks in various ways, where the best results reported for author are given by the co-author network, and the best results for papers are generated by linking two papers if the are published in the same conference. The above two approaches are referred as LLGC (A-A) and LLGC (P-P). Note that we do not use labels of conferences in training, so we cannot build a conference-conference (C-C) sub-network for classification. We also try to apply LLGC on all the objects without considering their different types. It is denoted by LLGC (A-C-P-T). The key difference between LLGC (A-C-P-T) and GNetMine (A-C-P-T) is the normalization of the relation matrix: the former one normalizes the whole big relation matrix, while the latter one normalizes the small relation matrices respectively. The semi-supervised classification results of paper, author and conference on DBLP

Table 4. Classification Accuracy (%) of Paper on the DBLP dataset

% of authors and papers labeled	LLGC (P-P)	CGL (P-P)	LLGC (A-C-P-T)	GNetMine (A-C-P-T)	CGL (A-C-P-T)
0.1%	63.26±2.81	67.02±2.97	58.49±2.23	71.74±2.93	**74.42±2.50**
0.2%	69.58±2.49	74.15±1.98	61.27±2.41	80.01±2.67	**82.94±2.53**
0.3%	80.70±2.68	82.47±1.73	69.82±2.78	84.91±2.31	**87.75±2.23**
0.4%	79.76±2.29	82.29±2.39	67.38±1.90	83.81±1.86	**87.62±2.29**
0.5%	79.64±1.76	82.57±2.37	74.64±2.15	83.57±2.18	**87.00±2.75**

Table 5. Classification Accuracy (%) of Author on DBLP dataset

% of authors and papers labeled	LLGC (A-A)	CGL (A-A)	LLGC (A-C-P-T)	GNetMine (A-C-P-T)	CGL (A-C-P-T)
0.1%	41.31±3.04	45.39±3.18	58.68±2.89	80.42±2.62	**83.77±2.55**
0.2%	45.51±2.83	51.47±2.93	60.86±3.41	81.24±3.46	**84.71±3.58**
0.3%	47.72±2.72	55.53±2.75	66.39±3.08	84.50±2.55	**87.31±2.44**
0.4%	47.47±3.81	55.50±3.41	70.32±2.19	85.14±2.05	**88.97±1.94**
0.5%	50.64±2.16	59.11±2.31	71.17±1.72	86.84±1.55	**89.93±1.67**

Table 6. Classification Accuracy (%) on Conference

% of authors and papers labeled	LLGC (A-C-P-T)	GNetMine (A-C-P-T)	CGL (A-C-P-T)
0.1%	73.50±4.84	80.50±3.25	**82.00±3.94**
0.2%	77.50±2.35	83.50±2.30	**86.50±2.12**
0.3%	82.00±3.37	87.00±2.89	**90.00±2.82**
0.4%	78.00±2.83	88.00±2.22	**92.00±2.59**
0.5%	82.50±2.17	90.00±2.77	**94.50±2.80**

dataset are shown in Tables 4, 5 and 6 respectively. Since wvRN and nLB perform much worse than the other methods on this dataset, we omit their results due to space limit. Similar observations are reported in [8].

We can observe that:

1. CGL (A-C-P-T) outperforms the state-of-the-art method, i.e., GNetMine, significantly on all the types of objects. The reason is that CGL is able to learn an approximately consistent heterogeneous network. This indicates the effectiveness of CGL on heterogeneous information networks.
2. CGL (P-P) is better than LLGC (P-P) and CGL (A-A) is better than LLGC (A-A). This strengthens the effectiveness of CGL on homogeneous networks.

5 Conclusions and Future Work

In this paper, we proposed a *Consistent Graph Learning*, which is robust to inconsistent links in networks. Experiments on both homogeneous and heterogeneous network datasets show that the proposed method outperforms the state-of-the-art methods. In the future, we plan to develop theoretical analysis on the

conditions under which the relation matrices can be recovered. Also, it is interesting to analyze how the percentage of inconsistent links in a network affect the classification performance, and test the algorithm in data set with large number of classes.

Acknowledgments. Research was sponsored in part by the U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), National Science Foundation IIS-1017362, IIS-1320617, and IIS-1354329, HDTRA1-10-1-0120, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative, and MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC.

References

1. Bilgic, M., Mihalkova, L., Getoor, L.: Active learning for networked data. In: ICML, pp. 79–86 (2010)
2. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning **3**(1), 1–122 (2011)
3. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press, Cambridge (2004)
4. Chen, Y., Sanghavi, S., Xu, H.: Clustering sparse graphs. In: Advances in Neural Information Processing Systems, pp. 2204–2212 (2012)
5. Gu, Q., Han, J.: Towards feature selection in network. In: CIKM, pp. 1175–1184 (2011)
6. Hong, M., Luo, Z.-Q.: On the linear convergence of the alternating direction method of multipliers. arXiv preprint arXiv:1208.3922 (2012)
7. Ji, M., Han, J., Danilevsky, M.: Ranking-based classification of heterogeneous information networks. In: KDD, pp. 1298–1306 (2011)
8. Ji, M., Sun, Y., Danilevsky, M., Han, J., Gao, J.: Graph regularized transductive classification on heterogeneous information networks. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010, Part I. LNCS, vol. 6321, pp. 570–586. Springer, Heidelberg (2010)
9. Li, W.-J., Yeung, D.-Y., Zhang, Z.: Probabilistic relational pca. In: NIPS, pp. 1123–1131 (2009)
10. Lin, Z., Chen, M., Wu, L.: The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Analysis, math.OC:-09-2215 (2010)
11. Lu, Q., Getoor, L.: Link-based classification. In: ICML, pp. 496–503 (2003)
12. Luo, D., Huang, H., Nie, F., Ding, C.H.: Forging the graphs: a low rank and positive semidefinite graph learning approach. In: Advances in Neural Information Processing Systems, pp. 2960–2968 (2012)
13. Macskassy, S.A., Provost, F.J.: Classification in networked data: A toolkit and a univariate case study. Journal of Machine Learning Research **8**, 935–983 (2007)
14. Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in network data. AI Magazine **29**(3), 93–106 (2008)
15. Shi, L., Zhao, Y., Tang, J.: Combining link and content for collective active learning. In: CIKM, pp. 1829–1832 (2010)
16. Smola, A.J., Kondor, R.: Kernels and regularization on graphs. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS (LNAI), vol. 2777, pp. 144–158. Springer, Heidelberg (2003)

17. Sun, R., Luo, Z.-Q., Ye, Y.: On the expected convergence of randomly permuted admm. arXiv preprint arXiv:1503.06387 (2015)
18. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: NIPS (2003)
19. Zhou, D., Huang, J., Schölkopf, B.: Learning from labeled and unlabeled data on a directed graph. In: ICML, pp. 1036–1043 (2005)
20. Zhu, S., Yu, K., Chi, Y., Gong, Y.: Combining content and link for classification using matrix factorization. In: SIGIR, pp. 487–494 (2007)
21. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-supervised learning using gaussian fields and harmonic functions. In: ICML, pp. 912–919 (2003)