Knowledge-Based Systems 89 (2015) 160–172

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

NMFE-SSCC: Non-negative matrix factorization ensemble for semi-supervised collective classification

Qingyao Wu [a,1], Mingkui Tan [b,1], Xutao Li [c], Huaqing Min [a], Ning Sun [d,*]

[a] School of Software Engineering, South China University of Technology, Guangzhou, China
[b] School of Computer Science, The University of Adelaide, Australia
[c] School of Computer Engineering, Nanyang Technological University, Singapore
[d] School of Business and Management, The Hong Kong University of Science and Technology, Hong Kong

ARTICLE INFO

Article history:
Received 20 November 2014
Received in revised form 22 April 2015
Accepted 14 June 2015
Available online 17 July 2015

Keywords:
Collective classification
Non-negative matrix factorization
Semi-supervised collective classification
Ensemble classification

ABSTRACT

Collective classification (CC) is a task to jointly classifying related instances of network data. Enabling CC usually improves the performance of predictive models on fully-labeled training networks with large amount of labeled data. However, acquiring such labels can be difficult and costly, and learning a CC classifier with only a few labeled data can lead to poor performance. On the other hand, there are usually large amount of unlabeled data available in practical. This naturally motivates *semi-supervised collective classification* (SSCC) approaches for leveraging the unlabeled data to improve CC from a sparsely-labeled network. In this paper, we propose a novel *non-negative matrix factorization* (NMF) based SSCC algorithm, called NMF-SSCC, to effectively learn a data representation by exploiting both labeled and unlabeled data on the network. Our idea is to use matrix factorization to obtain a compact representation of network data which uncovers the class discrimination of the data inferred from the labeled instances and simultaneously respects the intrinsic network structure. To achieve this, we design a new matrix factorization objective function and incorporate a label matrix factorization term as well as a network regularization term into it. An efficient optimization algorithm using the multiplicative updating rules is then developed to solve the new objective function. To further boost the predicting performance, we extend the proposed NMF-SSCC method into an ensemble scheme, called NMFE-SSCC, in terms of building a classification ensemble with a set of NMF-SSCC collective classifiers using different constructed latent graphs. Each NMF-SSCC classifier is learnt from one latent graph generated with various latent linkages for effectively label propagation. Experimental results on real-world data sets have demonstrated the effectiveness of the new methods.

1. Introduction

For traditional supervised classification, data instances are assumed to be independently and identically distributed to each other. However, for network data classification tasks, instances are interrelated with each other by edges which is considered as a form of autocorrelation. Network data have become ubiquitous in many fields ranging from the Internet to social sciences, biology and many others. Generally, network data contain nodes (instances) interconnected with each other by edges (linkages) reflects the relation or dependence between the nodes. Information on the nodes is provides as a set of attribute features (e.g., words present in the web pages). Network data are obviously not independently and identically distributed. Instead, the class labels of a given node are related to the class labels of the nodes it is connected. For example, web pages are connected by hyperlinks and the interlinked pages are more likely to have the same class label than the unlinked pages. Another relevant example is the collaboration networks where researchers are connected by their joint publications. The collaborating researchers are more likely to share similar research topics than researchers without collaboration.

The problem of learning from network data is a challenging problem that has attracted growing attention in the literature due to its importance to many applications. This drives the need for effective approaches to cope with it, and *collective classification* (CC) [1] is such a task for jointly classifying related instances in the network by exploiting the link interactions between instances.

* Corresponding author. Tel./fax: +852 (65) 6790 4618.
E-mail address: lynnsun.sun@gmail.com (N. Sun).
[1] Indicates equal contributions (Co-first author).

http://dx.doi.org/10.1016/j.knosys.2015.06.026
0950-7051/© 2015 Elsevier B.V. All rights reserved.

Enabling CC usually improves the performance of classification on network data as prediction of one instance can be used to improve predictions on related instances. Such a performance improvement usually relies on using a fully-labeled network that consists of a sufficiently large amount of labeled instances. However, for many real-world network data applications, acquiring such labels can be costly and time-consuming.

The challenges of label deficiency problem in CC are particularly evident, when the amount of labeled instances available is limited, learning a CC model can lead to poor performance as the supervision knowledge cannot be obtained from the local connections. On the contrary, the unlabeled instances are much easier to obtain as compared to labeled instances. For example, one may only have a handful of known fraudsters and legitimate users in a telephone fraud detection tasks, say fewer than 10%, 5% or even 1%. But there are also copious amount of unlabeled data which are available in such sparsely-labeled network data. This naturally motivates developing new CC algorithms that are able to utilize both labeled and unlabeled data together to enhance the performances. Recently, *semi-supervised collective classification* (SSCC) has been examined to leverage unlabeled data to enhance the classification performance when learning from a sparsely-labeled network [2–5]. In Fig. 1, we show examples of "traditional supervised classification (TSC)", "semi-supervised classification (SSC)", "collective classification (CC)", and "semi-supervised collective classification (SSCC)" to illustrate their differences. We can observe from the figure that TSC only considers the labeled data to train a classification model, while SSC makes use of both labeled and unlabeled data to enhance the classification performance. CC aims to classify related instances in the network data, while SSCC leverages the labeled and unlabeled data together in the learning process.

In this paper, we propose a *non-negative matrix factorization* (NMF) based algorithm, called NMF-SSCC, to learn a data representation for network data classification by exploiting both labeled and unlabeled data on the network for SSCC. NMF has received considerable attention due to its psychological and physiological interpretation of naturally occurring data whose representation may be parts-based in the human brain [6]. A good representation typically contains "explanatory factors behind the data" [7] and reveals the structures within the data set [8]. For network data classification, our goal is to use matrix factorization to find a parts-based representation which uncovers the class discrimination of the data inferred from the labeled instances and simultaneously respects the intrinsic network structure, so that one can achieve very competitive classification performance with the computed network data representation.

To achieve this, we design a new matrix factorization objective function and incorporate a label matrix factorization term and a network regularization term into it to encode the label information from the labeled instances and the geometric information on the network (see Fig. 2). We also develop an optimization algorithm to solve the objective function based on iterative updates of the decomposed matrices and to learn a new parts-based data representation for classification prediction. Intuitively, an instance linked to neighbors with high probabilities to be a particular class label is likely to have this class label.

To further boost the CC performance, we extend the proposed NMF-SSCC approach into an ensemble scheme to build a classification ensemble with a set of NMF-SSCC base classifiers using a set of constructed latent graphs. The new algorithm, called NMFE-SSCC, is to learn one NMF-SSCC classifier on the basis of a latent graph that generated with various kinds of latent linkages to explicitly link (separate) the instances with similar (different) class labels. Then, the learnt NMF-SSCC models are combined into an ensemble for prediction by using a vote among the models as the ensemble classification decision (see Fig. 3).

In summary, the main contributions of this paper are given as follows.

- A new NMF based learning method, i.e., NMF-SSCC, that takes the label information and network structure into account to seek a matrix factorization for the SSCC problem.

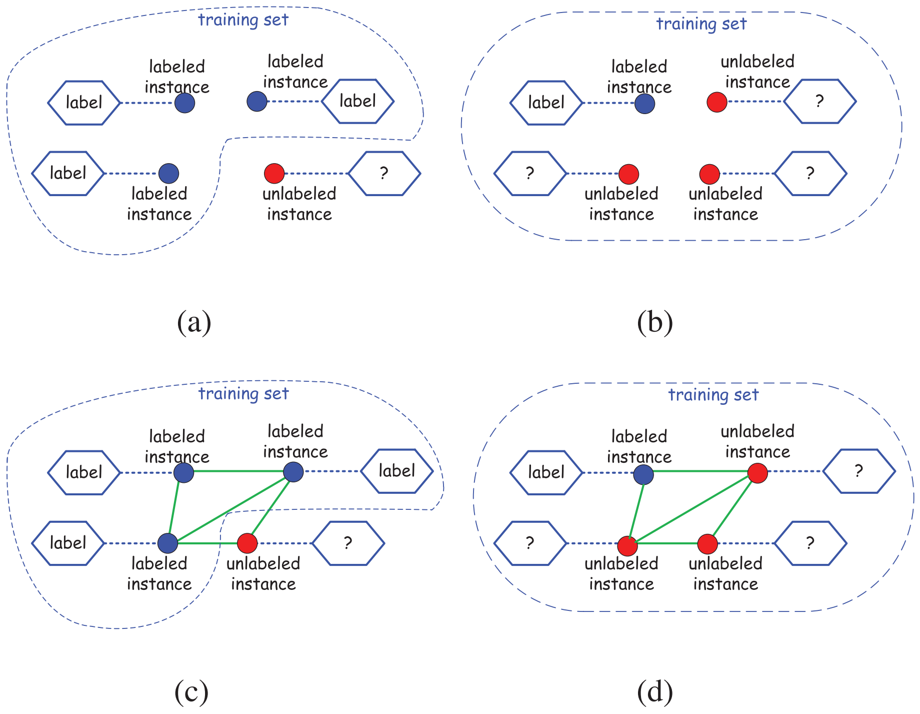

Fig. 1. (a) Traditional supervised classification; (b) semi-supervised classification; (c) collective classification; (d) semi-supervised collective classification. The links between the instances are their autocorrelations.

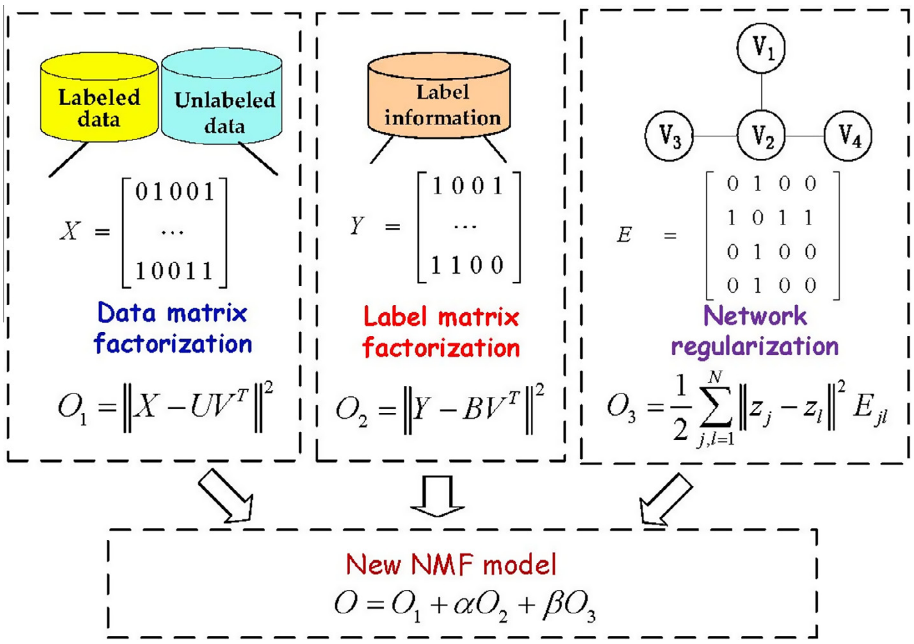
Labeled data
Unlabeled data
$X = \begin{bmatrix} 01001 \\ \cdots \\ 10011 \end{bmatrix}$
Data matrix factorization
$O_1 = \|X - UV^T\|^2$
Label information
$Y = \begin{bmatrix} 1001 \\ \cdots \\ 1100 \end{bmatrix}$
Label matrix factorization
$O_2 = \|Y - BV^T\|^2$
V1
V2
V3
V4
$E = \begin{bmatrix} 0&1&0&0 \\ 1&0&1&1 \\ 0&1&0&0 \\ 0&1&0&0 \end{bmatrix}$
Network regularization
$O_3 = \frac{1}{2}\sum_{j,l=1}^{N} \|z_j - z_l\|^2 E_{jl}$
New NMF model
$O = O_1 + \alpha O_2 + \beta O_3$

Fig. 2. An overview of the new NMF model for SSCC.

- An efficient iterative algorithm to optimize the objective function in the proposed NMF-SSCC method and to compute the class labels of the instances for classification.
- A theoretical discussion on the convergence of the proposed NMF-SSCC algorithm using an auxiliary function.
- An ensemble learning method for SSCC by learning multiple NMF-SSCC classifiers on various constructed latent graphs.
- Extensive experiments have been conducted to study the effectiveness of the proposed approach using various real-world network data sets.

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 describes the data sets and the experimental setup, followed by a discussion of the experimental results. Finally, conclusions are drawn in Section 5.

2. Related work

2.1. Learning from network data

In traditional supervised classification, instances are assumed to be independent of each other. However, in network data classification tasks, instances are interlinked with each other in a form of autocorrelation, and these relations among the instances can be helpful for the classification tasks. To exploit the data dependencies in the network, *collective classification* (CC) is proposed and it has received Macskassy and Provost [9] provide a brief review of the previous work of CC. In general, the methods can be categorized into three groups: relational-only methods, local classifier-based method and global formulation-based methods.

1. A relational-only method only uses relational information for classification. Typically, the algorithm computes a new label distribution for an instance based on the label distributions of its neighbors. Weighted-vote relational neighbor with relaxation labeling (wvRN + RL) [9] is one such method. Recently, Macskassy and Provost [9] showed that wvRN + RL performs very well in comparisons, and thus it is recommended to be used as a baseline for CC evaluations.
2. A local classifier-based method is based on an iterative process whereby a local classifier[2] is used to predict labels for unlabeled nodes using both original attribute features and newly constructed relational features. The iterative classification algorithm (ICA) [1] is one such example. Gibbs sampling [11] is further used in the ICA framework to enrich the statistical properties of the ICA algorithm. Previous studies reported that ICA is a fairly accurate method with robust performance for different network data and should be used as a baseline in CC evaluations. In recent years, there is a lot of work proposed to use a similar schema as ICA but with different learning strategies [11,12].
3. A global method trains a classifier to optimize a global objective function for prediction, often based on a graphical model such as loopy belief propagation of the relaxation labeling [12] (see [10] for more details).

Sen et al. [10] provide an empirical study of different CC methods. It is reported that enable CC often substantially increases classification accuracy when the class labels of linked instances are correlated. However, many current CC methods focus on learning from a fully-labeled network with large amount of labeled data, even though acquiring such labels is difficult and time-consuming. As pointed out in [2], when the labeled data are limited, the performance of CC may be degraded [13]. For example, consider predicting whether a web page belongs to a professor or a student. The ICA approach proposed in [1] uses the attribute features together with additional relational features constructed by using the labels of neighboring pages. The relational features of a given page can be constructed by counting the number of neighboring pages labeled student/professor that are linked to the page. However, the above schema may not work well when there are a limited number of labeled data available. In this scenario, most

[2] Various local classifiers have been used in previous studies, such as Logistic Regression (LR) [10] and Naive Bayes (NB) [11].

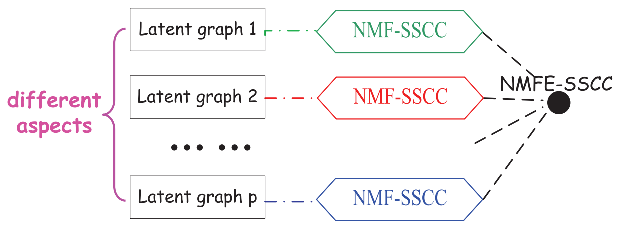

Fig. 3. An overview of the ensemble model for SSCC.

of the nodes may not connect to labeled data instances. Hence, the class summary of the neighboring instances tends to be 0 for the lack of labeled neighbor instances. As a consequence, the performance of CC may be degraded when the labeled data are limited [2].

In the task of CC, there are usually large amount of unlabeled data available. This naturally motivates the use of semi-supervised learning techniques to leverage the unlabeled data for CC. For instance, McDowell and Aha [2] propose a semi-supervised ICA (semiICA) algorithm using a hybrid regularization to boost the performance of the ICA algorithm. They investigate the performances of different semi-supervised learning variants in the study and developed a hybrid regularization to boost the performance gains. McDowell and Aha [3] show that utilizing neighbor attributes are often more useful than neighbor labels when the network is sparsely labeled. They introduce a new method that enable discriminative classifiers to be used with neighbor attributes. Gallagher et al. [14] propose a method that adds edges to a sparsely-labeled network to improve classification performance. Shi et al. [4] 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. Bilgic et al. [15,5] provide an alternative solution to overcome the label sparsity issue using active learning approaches. Rahmani et al. [16] use a transductive learning approach to predict the class labels of instances in instance-instance interaction networks. Other methods using the idea of label propagation or random walks [17–19] have been developed for the problem of semi-supervised collective classification.

2.2. Non-negative matrix factorization

Non-negative matrix factorization (NMF) is a matrix factorization technique for discovering low dimensional representations of non-negative data [20]. In many applications, the input data matrix (with non-negative elements) is of very high dimension, NMF seeks to find two lower dimensional matrices (also non-negative) whose product provides a good lower rank approximation to the original data matrix, and NMF has received much attention in such applications [21–26] because the learned bases can be interpreted as a natural parts-based representation of data. This interpretation is consistent with the psychological intuition of combining parts to form a whole, like face images and text documents [20,27]. That is, we can explain each data instance by additive linear combination of non-negative basis vectors since NMF allows only additive combinations. For this reason, NMF has been widely used in various real world applications such as face recognition [28], document clustering [29], recommendation [30] and collaborative filtering.

Let $\mathbf{X} = [x_1, \ldots, x_N] \in \mathbb{R}^{M\times N}$ denotes a data matrix formed by adjoining N non-negative column vectors (each vector is an input instance) of dimensionality M. NMF seeks to find two non-negative matrices $\mathbf{U} = [u_{ik}] \in \mathbb{R}^{M\times K}$ and $\mathbf{V} = [v_{jk}] \in \mathbb{R}^{N\times K}$ whose product provides a good approximation to the original data matrix $\mathbf{X}$, typically $K \ll M$ and $K \ll N$, such that

$$\mathbf{X} \approx \mathbf{U}\mathbf{V}^T \tag{1}$$

where $\mathbf{U}, \mathbf{V} \geqslant 0$, $\mathbf{U}$ is the basis matrix and $\mathbf{V}$ is the coefficient matrix.

The cost function that quantifies the quality of the approximation can be defined in different ways. Here we consider the square of the Euclidean distance of two matrices

$$\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|^2 = \sum_{i,j}\left(x_{ij} - \sum_{k=1}^{K} u_{ik}v_{jk}\right)^2. \tag{2}$$

The above objective function in Eq. (2) can be minimized by an iterative updating algorithm following the two steps

$$u_{ik} \leftarrow u_{ik}\frac{(\mathbf{X}\mathbf{V})_{ik}}{(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}} \tag{3}$$

$$v_{jk} \leftarrow v_{jk}\frac{(\mathbf{X}^T\mathbf{U})_{jk}}{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{jk}} \tag{4}$$

These updates are guaranteed to decrease the approximation cost at each iteration, and converge to a local minimum of the cost function.

Various researchers have considered introducing additional constraints to the original NMF. For instance, Li et al. [28] proposed the local NMF (LNMF) method that learns spatially localized subspace representation for visual patterns. Hoyer [31] presented the non-negative sparse coding (NSC) method to combine sparse coding and NMF for decomposing multivariante data into non-negative sparse components. Wang and Jia [32] proposed a Fisher NMF (FNMF) method for face recognition to impose Fisher constraints on the NMF algorithm. Yang et al. [33] proposed the non-negative graph embedding (NGE) method that simultaneously learns two subspaces in incorporating the marginal Fisher discrimiinformation with NMF.

Recent attention has been given to NMF on manifold learning. From the geometric perspective, the data usually reside on or close to an underlying sub-manifold embedded in a high dimensional ambient space. Learning from such intrinsic geometric structure modeled by a graph has been shown to be useful in various fields, including data clustering [34] and dyadic data analysis [35]. For this reason, Cai et al. [6] propose the graph regularized non-negative matrix factorization (GNMF) algorithm considering the manifold structure of the data in the NMF algorithm. A nearest-neighbor affinity graph is constructed to encode the data geometric structure. An optimization scheme is developed to find the parts-based data representation that respects the geometrical structure of the data space. But, GNMF was specifically designed for unsupervised data clustering tasks and thus cannot directly applied for collective classification.

Our proposed method is a NMF based algorithm but it is fundamentally different from these methods. We proposed a new NMF based objective function for semi-supervised collection classification (SSCC) problem by taking both network structure and label information in network data into account. To the best of the authors' knowledge, no previous work has been done on utilizing NMF for SSCC in network data where autocorrelations and label information are involved.

3. Methodology

We consider the network data as a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{X}, \mathbf{Y})$. $\mathbf{V}$ is the set of nodes $\{v_1, \ldots, v_N\}$, and $\mathbf{E} = [E_{ij}] \in \mathbb{R}^{N\times N}$ is the weight matrix on the graph such that $E_{ij} = 1$ if and only if nodes v_i and v_j are connected by an edge, and $E_{ij} = 0$ otherwise.

$\mathbf{X} = [x_1, \ldots, x_N] \in \mathbb{R}^{M \times N}$ is a data matrix with respect to N nodes' attribute feature vectors of dimensionality M, each $x_i \in \mathbf{X}$ is an attribute vector for a node v_i. $\mathbf{Y} = [Y_1, \ldots, Y_N] \in \mathbb{R}^{q \times N}$ is a label matrix where q is the number of all possible class labels, and each $Y_i = [Y_{i1}, \ldots, Y_{iq}] \in \{0,1\}^q$ such that $Y_{i,j} = 1$ means that v_i is associated with j-th class label, and $Y_{i,j} = 0$ otherwise. Assume that we have n' labeled nodes $\{(x_i, Y_i)\}_{i=1}^{n'}$ and n'' unlabeled nodes $\{(x_i)\}_{i=n'+1}^{n'+n''}$ with $N = n' + n''$. The task is to predict the class labels of unlabeled nodes. In SCCC task, there are only a limited number of labeled nodes on the network, i.e., $n' \ll n''$, most of the unlabeled nodes may not connect to the labeled ones, which makes the task very challenging.

3.1. NMF-SSCC: Non-negative matrix factorization for SSCC

The ordinary NMF method is developed for unlabeled data analysis in the context of Euclidean structure of the data. The updates in Eq. (3) and (4) derived from the objective function of NMF in Eq. (2) simply ignore the label information and network structure which play a crucial role for network data classification problems. The main aim of this paper is to use matrix factorization to seek a compact representation of all the (labeled and unlabeled) data nodes of the network in which the new data representation can uncover the class discrimination of the data inferred from the labeled instances and simultaneously respects the intrinsic network structure. To achieve this, we design a new matrix factorization objective function and incorporate a label matrix factorization term and a network regularization term into it.

Given the data matrix $\mathbf{X}$ and the label matrix $\mathbf{Y} = [Y_1, \ldots, Y_N] \in \mathbb{R}^{q \times N}$ which encodes the label information of all the data in which $Y_{ij} = 1$ if x_i is labeled with c_j, and $Y_{ij} = 0$ otherwise for labeled data, and $Y_{ij} = 0$ for unlabeled data, we extend the objective function of NMF by incorporating an additional label matrix factorization

$$\mathcal{O}_1 = \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|^2 + \alpha\|\mathbf{W} \odot (\mathbf{Y} - \mathbf{B}\mathbf{V}^T)\|^2 \tag{5}$$

where $\odot$ is the Hadamard product symbol which is a binary operation that takes two matrices of the same dimensions, and produces another matrix with elements given by $[A \odot B]_{ij} = [A]_{ij} \cdot [B]_{ij}$. The above objective function is divided into two terms. The first term is exactly the same as the objective function of ordinary in Eq. (2). The second term is the matrix factorization term in terms of $\mathbf{Y}$, α is a tradeoff parameter to determine the importance of the second term.

In the label matrix factorization term, $\mathbf{B} \in \mathbb{R}^{q \times K}$ is the basis matrix, $\mathbf{V}$ is the coefficient matrix, and $\mathbf{W} = [W_{ij}] \in R^{q \times N}$ is a weight matrix such that the elements of $\mathbf{W}$ are with nonzero values if the labels of corresponding instances are known. Otherwise, the elements of $\mathbf{W}$ are 0. Specifically, we have

$$W_{ij} = \begin{cases} \theta, & \text{if } Y_i \text{ is known and } Y_{ij} = 1 \\ 1 - \theta, & \text{if } Y_i \text{ is known and } Y_{ij} = 0 \\ 0, & \text{if } Y_i \text{ is unknown.} \end{cases} \tag{6}$$

By using the above constraints, the supervised knowledge inferred from the labeled instances can be preserved in solving the matrix factorization mathematical problem given by Eq. (5). In practical, the learning data sets may have large number of possible class labels. Note that the label matrix $\mathbf{Y}$ is a sparse matrix in this case because the number of zero elements(i.e., $Y_{ij} = 0$) is much larger than the remaining nonzero elements (i.e., $Y_{ij} = 1$). We can use a parameter θ to determine the weights to attach to these two types of elements.

Recall that the NMF method is a dimension reduction technique for factorizing a given matrix into the low-dimensional latent factor space, and is taken out through some constrained optimization process which can theoretically work on different forms of objective functions that quantify the quality of the factorization, such as Euclidean distance or Kullback–Leibler divergence [6]. In relational data sets and information network data, instances are correlated with complex dependencies, and exploiting the data dependencies can be used to enhance the performance of predicting class labels of related instances on the network. One hopes that the new data representation can respect the intrinsic network structure, rather than ambient Euclidean structure. To achieve this, we propose a new matrix factorization objective function which explicitly considers local invariance with respect to the geometrical information of the network, i.e., the neighboring nodes x_j and x_l on the network are likely to have similar classes. Specifically, we incorporate a network regularization term into the NMF objective function to seek a matrix factorization in which two data nodes are sufficiently close to each other if they are connected on the network. In other words, the distances of two data instances in terms of the new representation should be close if they are connected on the network where the measure of distance of two data instances can be measure by the Euclidean distance.

Given an adjacency matrix $\mathbf{E} = [E_{ij}] \in \mathbb{R}^{N \times N}$ to model the linkages of a network data with N vertices where $E_{ij} = 1$ if and only if nodes i and j are connected by an edge, and $E_{ij} = 0$ otherwise, The low dimensional representations of two neighboring nodes x_j and x_l with respect to the new basis are $\mathbf{z}_j = [v_{j1}, \ldots, v_{jK}]^T$ and $\mathbf{z}_l = [v_{l1}, \ldots, v_{lK}]^T$, respectively. The Euclidean distance of these two neighboring nodes is

$$d(\mathbf{z}_j, \mathbf{z}_l) = ||\mathbf{z}_j - \mathbf{z}_l||^2$$

With the adjacency matrix $\mathbf{E}$ and the distance measure $d(\mathbf{z}_j, \mathbf{z}_l)$, we can compute the smoothness of the instances on the intrinsic network structure using the following network regularizer

$$\begin{aligned} \mathcal{R} &= \frac{1}{2}\sum_{j,l=1}^{N} ||\mathbf{z}_j - \mathbf{z}_l||^2 E_{jl} = \sum_{j=1}^{N} z_j^T z_j D_{jj} - \sum_{j,l=1}^{N} z_j^T z_l E_{jl} \\ &= \mathrm{Tr}(\mathbf{V}^T\mathbf{D}\mathbf{V}) - \mathrm{Tr}(\mathbf{V}^T\mathbf{E}\mathbf{V}) = \mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \end{aligned} \tag{7}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix and $\mathbf{D}$ is a diagonal matrix whose elements are column sum of $\mathbf{E}$, i.e., $\mathbf{D}_{jj} = \sum_l E_{jl}$. $\mathbf{L} = \mathbf{D} - \mathbf{E}$ is the graph Laplacian.

Combining this network regularizer $\mathcal{R}$ with the objective function in Eq. (5), we obtain the following new matrix factorization objective function with an additional network regularization term

$$\mathcal{O} = ||\mathbf{X} - \mathbf{U}\mathbf{V}^T||^2 + \alpha||\mathbf{W} \odot (\mathbf{Y} - \mathbf{B}\mathbf{V}^T)||^2 + \beta\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \tag{8}$$

where β is the parameter controlling the importance of the network regularization term.

Similar to the standard NMF, multiplicative updates are derived for $\mathbf{U}, \mathbf{B}$ and $\mathbf{V}$ for minimizing the objective function. We introduce an iterative algorithm which can achieve a local minimum for the objective function $\mathcal{O}$ in Eq. (8). Using the matrix properties $\mathrm{Tr}(\mathbf{AB}) = \mathrm{Tr}(\mathbf{BA})$ and $\mathrm{Tr}(\mathbf{A}) = \mathrm{Tr}(\mathbf{A}^T)$, the objective function can be rewritten as follows.

$$\begin{aligned} \mathcal{O} &= \mathrm{Tr}((\mathbf{X} - \mathbf{U}\mathbf{V}^T)(\mathbf{X} - \mathbf{U}\mathbf{V}^T)^T) + \alpha\mathrm{Tr}(\mathbf{W} \odot ((\mathbf{Y} \\ &\quad - \mathbf{B}\mathbf{V}^T)(\mathbf{Y} - \mathbf{B}\mathbf{V}^T)^T)) + \beta\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \\ &= \mathrm{Tr}(\mathbf{X}\mathbf{X}^T) - 2\mathrm{Tr}(\mathbf{X}\mathbf{V}\mathbf{U}^T) + \mathrm{Tr}(\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T) + \alpha\mathrm{Tr}(\mathbf{W} \odot \mathbf{Y}\mathbf{Y}^T) \\ &\quad - 2\alpha\mathrm{Tr}(\mathbf{W} \odot YV\mathbf{B}^T) + \alpha\mathrm{Tr}(\mathbf{W} \odot \mathbf{B}\mathbf{V}^T\mathbf{V}\mathbf{B}^T) + \beta\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \end{aligned} \tag{9}$$

Let ψ_{ik}, γ_{ik} and ϕ_{jk} be the Lagrange multiplier for constraint $u_{ik} \geqslant 0, b_{ik} \geqslant 0$ and $v_{jk} \geqslant 0$, respectively. We need to minimize $\mathcal{O}$

with respect to $\mathbf{U}, \mathbf{B}$ and $\mathbf{V}$ subject to the Lagrange multiplier constraints. Then we have the Lagrangian $\mathcal{L}$ as follows.

$$\begin{aligned}\mathcal{L} &= \mathrm{Tr}(\mathbf{X}\mathbf{X}^T) - 2\mathrm{Tr}(\mathbf{X}\mathbf{V}\mathbf{U}^T) + \mathrm{Tr}(\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T) + \alpha\mathrm{Tr}(\mathbf{W}\odot\mathbf{Y}\mathbf{Y}^T) \\ &\quad - 2\alpha\mathrm{Tr}(\mathbf{W}\odot\mathbf{Y}\mathbf{V}\mathbf{B}^T) + \alpha\mathrm{Tr}(\mathbf{W}\odot\mathbf{B}\mathbf{V}^T\mathbf{V}\mathbf{B}^T) + \beta\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \\ &\quad + \mathrm{Tr}(\Psi\mathbf{U}^T) + \mathrm{Tr}(\Upsilon\mathbf{B}^T) + \mathrm{Tr}(\Phi\mathbf{V}^T)\end{aligned} \tag{10}$$

where $\Phi = [\psi_{ik}], \Upsilon = [\gamma_{ik}]$ and $\Phi = [\phi_{jk}]$.

The partial derivatives of $\mathcal{L}$ with respect to $\mathbf{U}, \mathbf{B}$ and $\mathbf{V}$ are

$$\frac{\partial\mathcal{L}}{\partial\mathbf{U}} = -2\mathbf{X}\mathbf{V} + 2\mathbf{U}\mathbf{V}^T\mathbf{V} + \Phi \tag{11}$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{B}} = -2\alpha[\mathbf{W}\odot\mathbf{Y}]\mathbf{V} + 2\alpha[\mathbf{W}\odot\mathbf{B}\mathbf{V}^T]\mathbf{V} + \Upsilon \tag{12}$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{V}} = -2\mathbf{X}^T\mathbf{U} + 2\mathbf{V}\mathbf{U}^T\mathbf{U} - 2\alpha[\mathbf{W}^T\odot\mathbf{Y}^T]\mathbf{B} + 2\alpha[\mathbf{W}^T\odot\mathbf{V}\mathbf{B}^T]\mathbf{B} + 2\beta\mathbf{L}\mathbf{V} + \Phi \tag{13}$$

By using the Karush–Kuhn–Tucker conditions $\psi_{ik}u_{ik} = 0, \gamma_{ik}b_{ik} = 0$ and $\phi_{jk}v_{jk} = 0$, we have

$$(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}u_{ik} - (\mathbf{X}\mathbf{V})_{ik}u_{ik} = 0 \tag{14}$$

$$([\mathbf{W}\odot\mathbf{B}\mathbf{V}^T]\mathbf{V})_{ik}b_{ik} - ([\mathbf{W}\odot\mathbf{Y}]\mathbf{V})_{ik}b_{ik} = 0 \tag{15}$$

$$(\mathbf{V}\mathbf{U}^T\mathbf{U} + \alpha[\mathbf{W}^T\odot\mathbf{V}\mathbf{B}^T]\mathbf{B} + \beta\mathbf{D}\mathbf{V})_{jk}v_{jk} - (\mathbf{X}^T\mathbf{U} + \alpha[\mathbf{W}^T\odot\mathbf{Y}^T]\mathbf{B} + \beta\mathbf{E}\mathbf{V})_{jk}v_{jk} = 0 \tag{16}$$

These equations lead to the following updating rules

$$u_{ik} \leftarrow u_{ik}\frac{(\mathbf{X}\mathbf{V})_{ik}}{(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}} \tag{17}$$

$$b_{ik} \leftarrow b_{ik}\frac{([\mathbf{W}\odot\mathbf{Y}]\mathbf{V})_{ik}}{([\mathbf{W}\odot\mathbf{B}\mathbf{V}^T]\mathbf{V})_{ik}} \tag{18}$$

$$v_{jk} \leftarrow v_{jk}\frac{(\mathbf{X}^T\mathbf{U} + \alpha[\mathbf{W}^T\odot\mathbf{Y}^T]\mathbf{B} + \beta\mathbf{E}\mathbf{V})_{ik}}{(\mathbf{V}\mathbf{U}^T\mathbf{U} + \alpha[\mathbf{W}^T\odot\mathbf{V}\mathbf{B}^T]\mathbf{B} + \beta\mathbf{D}\mathbf{V})_{ik}}. \tag{19}$$

When $\alpha = 0$ and $\beta = 0$, the above updating rules reduce to the updating rules of the original NMF.

Theorem 1. *The objective function $\mathcal{O}$ in Eq. (8) is non-increasing under the updating rules in Eqs. (17)–(19).*

The detailed proof for the above theorem is given in the Appendix. When $\alpha = 0$ and $\beta = 0$, the updating rules in Eqs. (17)–(19) reduce to the updating rules of the original NMF.

Algorithm 1. NMF-SSCC

```
Require: input matrices $\mathbf{X}, \mathbf{Y}$, and the parameters $\alpha$ and $\beta$
Ensure: output label matrix $\mathbf{Y}'$
 1: Initialize V using Eqs. (20) and (21).
 2: repeat
 3:    Update U using Eq. (17);
 4:    Update B using Eq. (18);
 5:    Update V using Eq. (19);
 6:    Reset V using Eq. (20) for labeled data;
 7: until stopping criteria is met
 8: for each unlabeled instance $x_i$ do
 9:    $k \leftarrow \arg\max_{\hat{k}}(v_{i\hat{k}})$;
10:    $Y'(i,k) \leftarrow 1$;
11: end for
```

3.2. NMF-SSCC Algorithm

We describe the NMF-SSCC algorithm in Algorithm 1 using the proposed NMF based method for SSCC. In the algorithm, the first step (line 1) initializes the value of $\mathbf{V}$ based on the class priors (using the labeled data). Specifically, for the labeled data we have

$$v_{jk} = \begin{cases} 1, & \text{if } y_{jk} = 1, \\ 0, & \text{if } y_{jk} = 0. \end{cases} \tag{20}$$

For unlabeled data, the values of v_{jk} are initialized as

$$v_{jk} = \frac{\sum_i n(c_k, x_i)}{\sum_{k'}\sum_i n(c_{k'}, x_i)} \tag{21}$$

where $n(c_k, x_i) = 1$ if x_i is labeled as c_k and 0 otherwise.

The matrices $\mathbf{U}, \mathbf{B}$ and $\mathbf{V}$ are then updated alternately until the objective value of Eq. (8) does not change or the maximum number of iterations is met (line 2–7). In this procedure, the v_{jk} values of the labeled data are reset at each iteration to encode the label information (line 6). In practice, only a small portion of elements of $\mathbf{V}$ will be reset when we have limited number of labeled data, and thus the reset action does not affect the convergence of the algorithm as we see in the experiment section.

Using these updating rules, we obtain a local optimum solution of the non-negative matrix $\mathbf{V}$ for the objective function $\mathcal{O}$ in Eq. (8). In the following, we describe how to use $\mathbf{V}$ for classification prediction. We specify the column dimension of the new representation $\mathbf{V} = [v_{jk}] \in \mathbb{R}^{N\times K}$ of the original data with respect to the new basis as the same as the number of possible class labels q, i.e., we set K equals to q, each column of $\mathbf{V}$ corresponds to one class label. For an unlabeled instance x_j, we assign it with the class with largest v_{jk} score,

$$Y_{jk} = \begin{cases} 1, & \text{if } k = \arg\max_{\hat{k}} v_{j\hat{k}}, \\ 0, & \text{otherwise}, \end{cases} \tag{22}$$

3.3. Complexity analysis

In this part, we discuss the computational cost of our proposed NMF-SSCC algorithm in comparison to standard NMF. We count the arithmetic operations for the NMF algorithm based on the updating rules in Eq. (3) and (4) and the NMF-SSCC algorithm based on the updating rules in Eq. (17)–(19). We count the arithmetic operations of each iteration in two algorithms and the result is given in Table 1. Suppose the multiplicative updates stops after t iterations, the overall cost for NMF is $O(tMNK)$ and the overall cost for NMF-SSCC is also $O(tMNK)$.

3.4. NMFE-SSCC: NMF-SSCC ensemble with latent graphs for SSCC

The basic assumption of the network regularization method in our proposed NMF-SMCC model is that the network structure can well respect the prediction relationships between the input instances and the output class labels. The proposed NMF-SSCC method naturally works better with network structure that can facilitate the prediction of the instances' labels. The relationship between instances and class labels are usually have semantic interpretability and thus it is desirable to obtain a graph structure which can well respect the relationships between instances and class labels, because it relates more directly to the classification tasks. This naturally motivates approaches for constructing various latent graphs in order to improve the quality of classification. This is achieved by a latent graph construction approach by which an undirected graph is constructed which utilizes different data sources that are available in the CC problem setting, including autocorrelation, attribute features, and label information.

The basic idea of constructing latent graphs is to link together the instance nodes, such that nodes which are closer in the graphs will tend to have the same class labels, and the nodes which are disconnected will tend to have different class labels. Various data sources of the CC data can be used to construct the latent graphs.

Table 1
Computational operation counts for each iteration in NMF and NMF-SSCC where f.p. refers to floating-point, N: the number of instances; M: the number of features; K: the number of factors; q: the number of labels, and $M \gg N \gg q$.

	f.p. addition	f.p. multiplication	f.p. division	Overall
NMF	$2MNK + 2(M+N)K^2$	$2MNK + 2(M+N)K^2 + (M+N)K$	$(M+N)K$	$O(MNK)$
NMF-SSCC	$2MNK + 2(M+N)K^2 + 2qNK + 2(q+N)K^2 + N^2K + NK$	$2MNK + 2(M+N)K^2 + (M+N)K + 2qNK + 2(q+N)K^2 + N^2K + NK$	$(M+N)K$	$O(MNK)$

For example, the CC instances are associated with a set of attribute features which can be used to detect the underlying network structure among instances based on the idea that instances with similar feature values may also be similar in their class labels. On the other hand, the labeled CC instances are associated with a set of class labels, which can be represented by a set of class features. The class features are also useful for evaluating the pairwise similarity of instance instances. Intuitively, instances with large class overlap on their class features should be linked together in the latent graph. In general, the copious availability of different data sources in the CC problem setting can be used to design for effective latent graphs for our classification model.

Here we exploit the data sources available in CC data to construct a set of latent graphs for more effective label prediction. Via the latent linkages in the latent graphs we constructed, knowledge from labeled nodes can be propagated to unlabeled nodes more effectively, such as the example in Fig. 4. Specifically, we introduce three types of latent graphs that can be easily computed from the data. For each individual latent graph, we compute a weight $\mathbf{E}_{ij}$ for each element of its adjacency matrix where $\mathbf{E}_{i,j}$ is large indicates two nodes are close together, and vice versa.

Autocorrelation latent graph: We consider the autocorrelation network as a latent graph, and construct the adjacency matrix $\mathbf{E}^{(1)}$ of the autocorrelation latent graph as follows

$$\mathbf{E}^{(1)}_{ij} = \mathbf{E}(i,j)$$

where $\mathbf{E}(i,j) = 1$ if node v_i and node v_j are connected in the autocorrelation network, and $\mathbf{E}(i,j) = 0$ otherwise.

Random walk latent graph: Some first-order neighborhoods may not share the same class label in the autocorrelation network. In such cases, the performance of learning methods simply based on first-order linkages will be degraded. On the other hand, it is observed that second-order neighborhoods can have a great likelihood of sharing similar characteristics with the instance of interest [36]. Thus, we use the idea of *even-step* random walk with restart (ERWR) [14] to compute such latent linkages. We assume that the linkages linked to first-order neighborhoods with the same class label with the instance of interest (i.e., v_1) typically have triangle structures (see Fig. 4(b)). These neighbors (i.e., v_2 and v_3) are able to obtain high scores using ERWR because they are well-connected in the autocorrelation network. On the other hand, ERWR can avoid the immediate neighbors (e.g., v_1 and v_2) with inconsistent linkages that negatively influence the prediction of v_1 because such inconsistent linkages are often sparsely-connected (i.e., without triangle structures among these nodes). Also, ERWR is able to exploit the second-order neighborhoods (e.g., v_4) which are well-connected with level-1 neighbors.

Formally, we compute $\mathbf{P} = \mathbf{EE}$ and normalize its elements with respect to each column to obtain a normalized transition probability matrix $\mathbf{P}$, where $\mathbf{E}$ is the adjacency matrix of the autocorrelation network of the data. With transition probability given in $\mathbf{P}$, the ERWR random walker iteratively visits neighborhood nodes to compute the steady-state probability matrix as the adjacency matrix of the random walk latent graph. Also at each step, it has probability λ (e.g., $\lambda = 0.1$) to return to the start node. Concretely, we define the adjacency matrix $\mathbf{E}^{(2)}$ of the random walk latent graph as follows

$$\mathbf{E}^{(2)}_{ij} = R(i,j)$$

where $R = \sum_{t=1}^{T} \lambda(1-\lambda)^t P^t$ is the steady-state probability matrix after T steps.

Prediction similarity latent graph: We consider the values of class labels of the labeled instances as input features to build a classifier and use the learnt classifier to predict the labels of the remaining unlabeled instances. Specifically, we use SVM classifier with probability outputs implemented in the LIBSVM library [37] to compute the probabilities of different labels to an instance, $Y_i = [P(c_1|x_i), P(c_2|x_i), \ldots, P(c_q|x_i)]^T$, such that $P(c_j|x_i)$ is the probability (or confidence) of an instance x_i belongs to the class c_j. With the computed prediction confidences of the instances, the adjacency matrix $\mathbf{E}^{(3)}$ of the prediction similarity latent graph is defined as follows

$$\mathbf{E}^{(3)}_{ij} = Y_i^T Y_j$$

Here, Y_i and Y_j are normalized to unit length, thus the dot product of the two vectors is equivalent to their cosine similarity.

In the prediction similarity latent graph, there are many elements being close to zero. It may not be necessary to consider these elements. Therefore, we use a kNN construction scheme for graph. We connect two nodes v_i and v_j if v_j is among the k-nearest neighbors of v_i or if v_j is among the k-nearest neighbors of v_i. We define a sparse adjacency matrix for kNN graph as follows

$$\hat{\mathbf{E}}^{(3)}_{ij} = \begin{cases} 1, & \text{if } v_i \in \mathcal{N}_k(j) \text{ or } v_j \in \mathcal{N}_k(i) \\ 0, & \text{otherwise.} \end{cases}$$

where $\mathcal{N}_k(i)$ is the set of k nearest neighbors of v_i. In practice, we find that k does not need tuning. We use $k = 10$ nearest neighbors as default setting.

Algorithm 2. NMFE-SMCC

```
Input: latent graphs $\{\mathbf{E}^{(i)}\}_{i=1}^{p}$, input matrices $\mathbf{X}, \mathbf{Y}$, and the
  parameters $\alpha$ and $\beta$
Output: the label matrix $\mathbf{Y}'$
Procedure:
1: for $i = 1$ to $p$ do
2:    Learn a NMF-SMCC model using the constructed latent
      graph $\mathbf{E}^{(i)}$. In the NMF-SMCC model, compute the network
      regularizer $\mathcal{R}$ in Eq. (2) according to $\mathbf{E}^{(i)}$;
3:    Use EM algorithm to optimize the NMF-SMCC model to
      compute the output label matrix $\mathbf{Y}'^{(i)}$ as in Algorithm 1;
4: end for
5: Combine the results of $p$ learned models $\mathbf{Y}'^{(i)}, \mathbf{Y}'^{(i)}, \ldots, \mathbf{Y}'^{(p)}$
   into an ensemble prediction as $\overline{\mathbf{Y}}' = \frac{1}{p}\sum_{i=1}^{p}\mathbf{Y}'^{(i)}$
6: for each unlabeled instance $x_j$ do
7:    $k \leftarrow \arg\max_{\hat{k}}(\overline{Y}'(j\hat{k}))$;
8:    $Y'(j,k) \leftarrow 1$;
9: end for
```

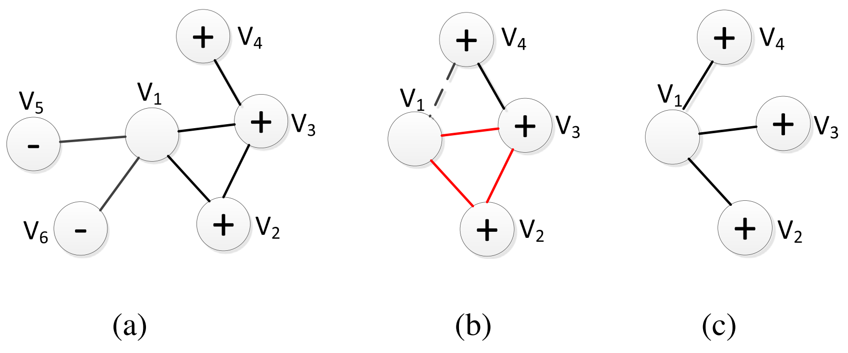

Fig. 4. An example of latent graphs. (a) autocorrelation latent graph, i.e., the original interaction network, where the ground-truth label of the center (unknown) node v_1 is "+", but it is difficult to predict the label (+ or −) for node v_1 since it has the same number of positive and negative neighboring nodes; (b) even-step random walk latent graph, the first-order neighborhoods with the same label ("+") to v_1 have triangle edges (red lines), hence they are reachable from v_1 using even-step random walks. On the other hand, the second-order neighborhoods (from the + nodes) in the network are also linked together by creating edges (dash line) using even-step random walks; (c) prediction similarity latent graph using kNN graph construction. In the kNN graph, a node pair shares an edge if the two nodes are k-nearest neighbors. In the example, we set $k = 3$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Via the latent graphs constructed, the supervised knowledge can be propagated from labeled to unlabeled nodes more effectively. Here, we introduce the NMFE-SSCC method to build a classification ensemble with a set of NMF-SSCC classifiers using various latent graphs. Given adjacency matrices $\{\mathbf{E}^{(i)}\}_{i=1}^{p}$ of p latent graphs, the NMFE-SMCC algorithm is described in Algorithm 2. In the algorithm, each NMF-SSCC model is learnt from one latent graph, and we combine the learned NMF-SSCC models into an ensemble classifier to effectively compute the prediction for network data.

4. Experiments

In this section, we compare the performance of our proposed NMF-SSCC algorithm with the autocorrelation network to a number of baselines (SVM, wvRN + RL, ICA, semiICA, and LNP) on three data sets, and show that the proposed NMF-SSCC algorithm outperforms these algorithms.

4.1. Data sets

We use three real network data sets for our performance comparison. A summary of the characteristics of these data sets are shown in Table 2 and described below.

The **Cora** data set [10][3] is a paper publication data set which is used frequently in collective classification studies. 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 in the network graph represents a paper document described by a 0/1 valued bag-of-word vector with 1433 dimensions. The citations provide links between the instances and we ignore their directions similar to Bilgic et al. [5]. This citation network consists of 5278 links.

The **Citeseer** data set [10] is a collection of research papers drawn from the Citeseer collection. The data set 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 dimension of the vector is 3703. There are 4598 links representing the citation relations between the instances.

[3] http://linqs.cs.umd.edu/projects/projects/lbc/index.html.

Table 2
A summarized description of the three data sets.

Characteristics	Cora	Citeseer	COIL20
#Instances	2708	3312	1440
#Features	1433	3703	1024
#Links	5278	4598	8402
#Classes	7	6	20

The **COIL20** image data set [6][4] contains 32×32 gray scale images of 20 objects viewed from varying angles. A p nearest neighbor graph is constructed to encode the interaction among the data instances. For each data node x_j, we find its $p = 5$ nearest neighbors and put edges between x_j and its neighbors.

4.2. Existing methods for comparison and experimental setting

We compare the accuracy and sensitivity of our proposed approach to five existing algorithms using the three data sets. These baseline algorithms are as follows:

1. **SVM**. This classifier only uses the attribute feature information for learning without considering using any link relationship. The linear kernel SVM in LibSVM[5] is used.
2. **wvRN + RL**. This algorithm 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 performs quite well on the network data. It is considered as a good baseline learner for CC.
3. **ICA**. The Iterative Collective classification Algorithm (ICA) [1] is one of the most popular CC methods that is 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 Naive Bayes and k-Nearest Neighbors, for ICA [5]. Therefore, we use LR as the local classifier for ICA in our experiments.
4. **semiICA**. This algorithm is a SSCC method. It uses the idea of label regularization to learn hybrid local classifiers, enabling them to leverage the unlabeled data to improve the learning performance of the ICA algorithm.

[4] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html.
[5] Available at http://www.csie.neu.edu.tw/cjlin/libsvm/.

5. **LNP**. This algorithm is another SSCC method [4]. 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.

To quantify the classification performance for the various classification approaches at each experimental trial,

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

is used.

In the experiments, a small number of examples are randomly selected as labeled data. The remaining ones are used as unlabeled data for testing the quality of the classification. This is a challenging problem from the classifier training perspective because the labeled data is scarce. The labeled/unlabeled data split is repeated 10 times for 10 runs. For each algorithm, the *average accuracy* as well as the standard deviation over 10 runs is reported in order to compare the effectiveness of the classifiers for the learning tasks.

Note that it is generally more difficult to determine the classifier parameter values when the number of labeled data available is smaller. Thus, we simply set the penalty $C = 1.0$ for the SVM with linear kernel as default. The maximum number of iterations for ICA and wvRN + RL is set to 10 and 100 as the setting used in [38]. The number of neighbors used in wvRN + RL is depended on the percentages of the labeled data used in the experiments. Specifically, for each unlabeled node, wvRN + RL finds all its labeled neighbors and computes the most common label among these neighbors as the new label. The number of labeled neighbors used increases as the percentages of labeled data increases.

The percentages of the labeled data used in the experiments is small, the number of possible labels of the data sets is large, and the label matrix **Y** in Eq. (6) is a sparse matrix, i.e., **Y** has large quantity of zero elements ($Y_{ij} = 0$). In this case, we set the parameter θ in Eq. (6) to be a small value (0.01 in this study) to ensure the zero elements with a large weight. The parameters α and β in Eq. (8) are set default to 10 and 5, respectively. The parameter selection will be discussed in Section 4.6.

4.3. NMF-SSCC results

Figs. 5–7 show the accuracy results of different compared algorithms on the Cora, Citeseer and COIL20 data sets with respect to different percentages of labeled data ranging from 1% to 10%. For each percentage of labeled data in each data set, we average accuracies of each algorithm over 10 runs and report the results of average accuracies and standard deviation. The x-axis of the figures indicates the different percentages of data used as labeled instances, and y-axis of the figures indicates the results of accuracy of different algorithms. We see from the figures that, over all percentages of labeled/unlabeled data splits, our proposed NMF-SSCC method consistently outperforms the other baselines. In general, the smaller the percentage of labeled data is, the larger improvement NMF-SSCC can achieve.

We note that only a small number of labeled data available is the focus for our study. From the figures, one observes that NMF-SSCC achieves a better performance than the other algorithm by a large margin when the percentage of labeled data is less than 5%. NMF-SSCC achieves the largest improvement against the other baselines while learning with only 1% labeled data. For instance, NMF-SSCC achieves accuracy improvement of 16% against the second best method on the Cora data set (NMF-SSCC with accuracy 0.67 versus semiICA with 0.51), and 22% on the Citeseer data set (NMF-SSCC with accuracy 0.66 versus semiICA with 0.45). This result illustrates the advantages of the proposed NMF-SSCC method when there are an extremely small number of labeled data, and it is consistent with our earlier assertions that our proposed approach can work well in the paucity of labeled data. A closer examination of the results in the figures shows that the second best performing methods are semiICA and LNP which leveraging the unlabeled data using some form of semi-supervised learning.

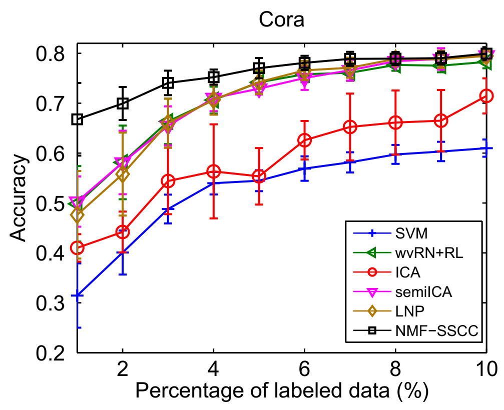

Fig. 5. Results for the Cora data set.

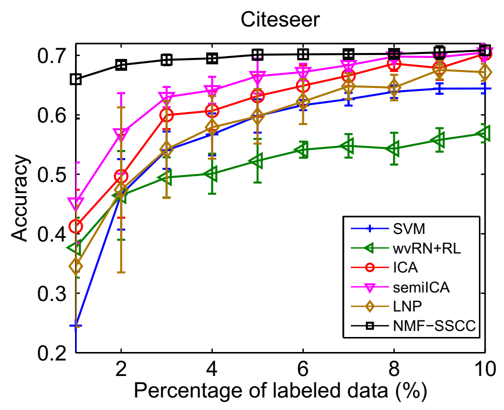

Fig. 6. Results for the Citeseer data set.

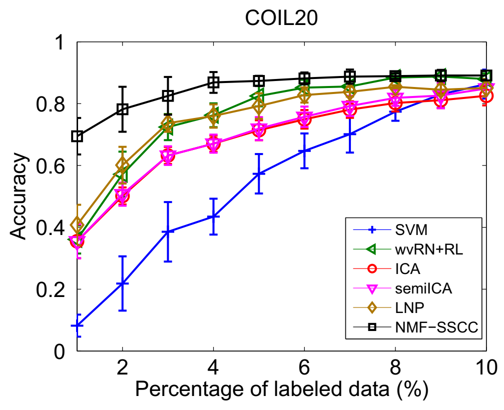

Fig. 7. Results for the COIL20 data set.

Table 3
The results of win/tie/loss counts.

Data set	SVM	wvRN + RL	ICA	semiICA	LNP
Cora	5/0/0	4/1/0	5/0/0	4/1/0	4/1/0
Citeseer	5/0/0	5/0/0	5/0/0	5/0/0	5/0/0
COIL20	5/0/0	4/1/0	5/0/0	5/0/0	5/0/0

The SVM and wvRN + RL methods only using the attribute features or relational information perform is not competitive with the other algorithms. They perform poorly when there are limited number of labeled data.

We further analyze the performance difference between NMF-SSCC and other methods and count the results of the win-tie-loss with pairwise *t*-tests at 0.10 significance level. In the statistical significance evaluation, the percentages of labeled data used for learning are 1%, 2%, 3%, 4% and 5%. For each labeled/unlabeld data split, a win (or loss) is counted when NMF-SSCC is significantly better (or worse) than the compared algorithm over 10 runs. Otherwise, a tie is recorded. Table 3 shows the win/tie/lose counts with pairwise *t*-test for NMF-SSCC against the baselines. The results of win/tie/loss counts are 5/0/0 over all comparison except the comparisons on the Cora and COIL20 data sets where the win/tie/loss counts on Cora data set are 4/1/0 when compared with wvRN + RL, semiICA, and LNP with 5% of labeled data and the win/tie/loss counts on COIL20 data set are 4/1/0 when compared with wvRN + RL with 5% of labeled data. This result reveals that NMF-SSCC is statistically superior to other compared methods when there is only limited number of labeled data.

We also conduct experiment to compare the running time of NMF-SSCC with the SVM, wvRN + RL, ICA, and semiICA algorithms on the Citeseer dataset. 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 are given in Table 4, where the best performance on each percentage of labeled data is bolded. We can see from the table that NMF-SSCC is able to achieve better running time performance against compared algorithms. Similar running time results on the other datasets are also observed.

4.4. NMFE-SMCC results

In the case of NMFE-SMCC, three types of latent graphs are utilized to build the ensemble model. It is desirable to investigate which latent graph will work better with the proposed NMF method, and whether the ensemble method will improve the equality of classification. In this experiment, we compare the performance of the MFE-SMCC method using a single latent graph with the NMFE-SMCC ensemble method using multiple latent graphs on the Cora data set with different label ratio from 1% to 5%. For each label ratio, we report the average accuracy of each comparison method over 10 runs. The experimental results are given in Fig. 8, where NMF-SMCC-1, NMF-SMCC-2 and NMF-SMCC-3 denote the NMF-SMCC method using the autocorrelation latent graph ($\mathbf{E}^{(1)}$), the random walk latent graph ($\mathbf{E}^{(2)}$) and the prediction similarity latent graph ($\mathbf{E}^{(3)}$), respectively. While NMFE-SMCC denotes the ensemble method using all the latent graphs.

From Fig. 8, we observe that NMFE-SMCC using multiple latent graphs is able to achieve better performance against the NMF-SMCC method using a single latent graph. A reasonable explanation for this finding is that the different latent graphs have complementary relationship for classification, because these latent graphs are derived on the basis of different data sources. The performance of an ensemble learner is highly dependent on two factors: one is the accuracy of each component learner; the other is the diversity among these components. Examining the results in Fig. 8 shows that the performances of the NMF-SMCC methods built from different graphs are able to achieve good overall performance. This result indicates that different latent graphs provide prediction knowledge from different aspects, and thus the ensemble of such latent graphs is able to achieve better performance than the single latent graph scheme.

We also conduct experiment to investigate how the prediction similarity latent graph affects the performance of the ensemble. Table 5 shows the accuracy results of NMFE-SMCC with all latent graphs and NMFE-SMCC without using the prediction latent graph on the Cora dataset. We find that NMFE-SSCC using all the latent graphs has better performance against the NMFE-SSCC only using the autocorrelation latent graph and the random walk latent graph. These results are reasonable because the autocorrelation latent graph and the random walk latent graph are based on the network topology information. The prediction latent graph considers the availability of label information to learn a latent graph such that the supervision knowledge can be applied in the ensemble. Such strategy is found to be useful in learning tasks with limited labeled data in the experiments.

Table 4
The running time (in seconds) of different learning algorithms on the Citeseer dataset.

	SSCC-NMF	SVM	wvRN + RL	ICA	semiICA
2%	**3.50**	4.20	5.87	6.15	6.75
4%	**6.93**	12.15	14.13	30.19	32.07
6%	**7.05**	7.75	8.23	9.25	9.18
8%	**12.18**	13.79	17.19	18.04	19.63
10%	**25.72**	30.35	39.92	41.31	45.99

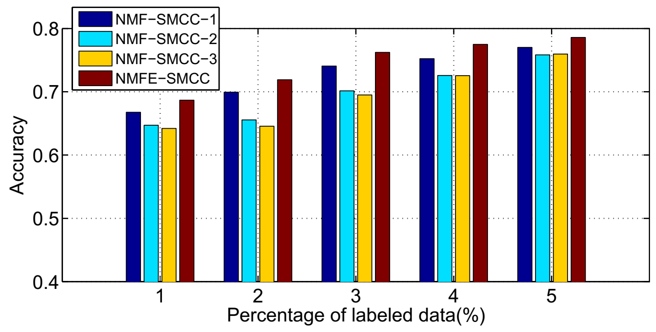

Fig. 8. The accuracy of NMF-SMCC (with different latent graphs) and NMFE-SMCC (with all latent graphs) against different percentages of labeled data (%) on the Cora data set.

Table 5
The accuracy of NMFE-SMCC with all latent graphs and NMFE-SMCC without using the prediction graph on the Cora dataset.

Ensemble methods	1%	2%	3%	4%	5%
NMFE-SMCC with all graphs	0.687	0.719	0.762	0.775	0.786
NMFE-SMCC without prediction graph	0.672	0.701	0.750	0.758	0.770

4.5. Convergence study

The iterative algorithm in Algorithm 1 minimizes the objective functions $\mathcal{O}$ in Eq. (8) using iterative updates. Here, we investigate the convergence of the iterative algorithm. Figs. 9 and 10 shows the convergence curve of the NMF-SSCC algorithm on the Cora and Citeseer data sets (with 5% labeled data). The *x*-axis is the

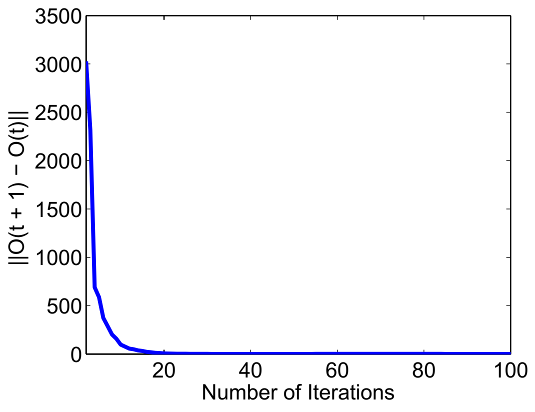

Fig. 9. Convergence curve of NMF-SSCC on the Cora data set.

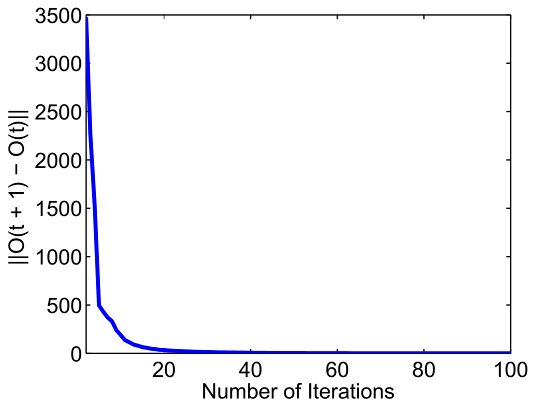

Fig. 10. Convergence curve of NMF-SSCC on the Citeseer data set.

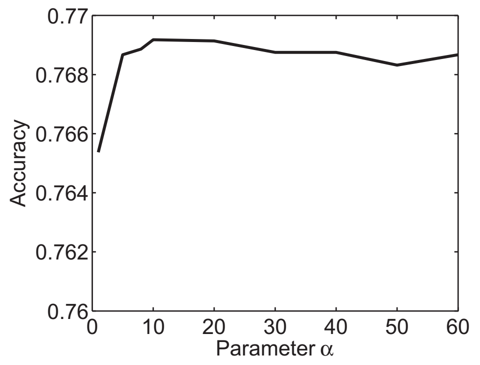

Fig. 11. Classification accuracy of different α on the Cora data set (the parameter β is fixed at 5). The accuracy of NMF-SSCC with respect to the value of α between 5 and 60 does not change significantly.

number of iterations in the optimization procedure for the objective function $\mathcal{O}$, and the y-axis is the difference of successively computed objective value $||\mathcal{O}(t-1) - \mathcal{O}(t)||$ at t and $t-1$ iteration steps. We observe from the figures that the difference of the successive computed objective values decreases as the iteration number increases. The algorithm converges after about 20 iterations.

4.6. Parameter sensitivity

For our proposed NMF-SSCC method, we need to set the parameters α and β which quantify the importance of the label matrix factorization term and network regularizer term in the objective function (8). In this experiment, we investigate how different values of the parameters α and β affect the classification accuracy of the proposed method. Fig. 11 shows the classification accuracy of NMF-SSCC using different values of label matrix factorization term parameter α. The parameter β is fixed at 5. From the figure, we observe that when α is small the accuracy is poor because the proposed algorithm becomes an unsupervised NMF. The accuracy of the proposed NMF-SSCC method increases as the value of α increases. However, the accurate for α between 5 and 60 does not change significantly. Such a plateau in the accuracy curve indicates that the proposed method is insensitive to the specific setting of α across a wide range of parameter setting. One observes that the best performance is achieved at $\alpha = 10$. Fig. 12 shows the classification accuracy of NMF-SSCC using different values of network regularization term parameter β. The parameter α is fixed at 10. When one has a small β value, i.e., no network regularization is used, the classification accuracy is degraded. As the parameter β increases, the accuracy reaches a plateau between 5 and 60, and does not change significantly. One observes that the best performance is achieved at $\beta = 5$. The convergence behaviors of NMF-SSCC on the other data sets are similar.

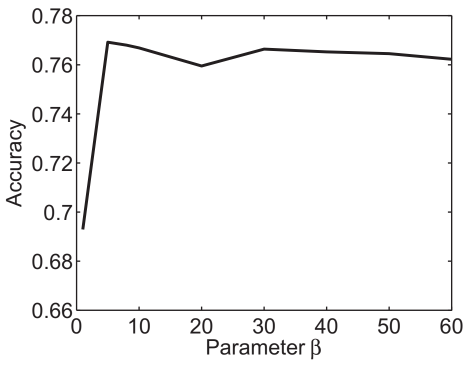

Fig. 12. Classification accuracy of different β on the Cora data set (the parameter α is fixed at 10). The accuracy of NMF-SSCC with respect to the value of β between 5 and 60 does not change significantly.

5. Conclusions

In this paper, we present an effective non-negative matrix factorization (NMF) based learning method for semi-supervised collective classification (SSCC). This method alleviates the issue of scarce labels inherent in the domain of network data classification. A factorization term contains the label information and a network regularizer term respects the network structure are incorporated into the NMF model to seek a data representation which respects the label information and the intrinsic network structure in the network data for SSCC. An efficient iterative algorithm using multiplicative updating rules is developed to optimize the objective function of the proposed NMF method, and to effectively compute the class labels of the unlabeled instances. We also extend the proposed NMF method into an ensemble classification scheme by constructing various latent graphs that utilize different data sources that are available in the CC problem setting. We evaluate the proposed algorithms on three real world network data sets. The proposed algorithms show superior results compared to a number of baseline classification algorithms. The improvement of the ensemble method against the single latent graph methods is not significant because the proposed ensemble algorithm uses a simple equally-weighting method for each latent graph. The equally-weighting method is not able to reflect the different degrees of importance from different latent graphs. In future, we plan to investigate a more sophisticated weight setting for building the ensemble, such that the latent graph with more discriminative power for classification tends to have higher weights. A suitable weight for each latent graph is critical to our ensemble method. It remains unclear how to do weight selection theoretically and effectively. It is interesting to investigate the weight selection method in our future work. We also plan to use our proposed method for semi-supervised classification tasks in heterogeneous networks.

Acknowledgements

This research was supported by the Guangzhou Key Laboratory of Robotics and Intelligent Software under Grant No. 201509010013, and Fundamental Research Funds for the Central Universities under Grant No. D215048w and 2015ZZ029, and National Natural Science Foundation of China (NSFC) under Grant No. 61005061.

Appendix A. Proofs of Theorem 1

To prove Theorem 1, we need to show that $\mathcal{O}$ is non-increasing under the updating rules Eqs. (17)–(19). Since the second and the third terms in $\mathcal{O}$ are not related to **U**, we have exactly the same update formula for **U** under Eq. (8) as in the original NMF. On the other hand, the first and the third terms in $\mathcal{O}$ are not related to **B**. By reversing the roles of **X** and **Y**, and reversing the roles of **U** and **B** in Eq. (8), $\mathcal{O}$ can similarly be shown to be non-increasing under the update rules for **B**.

Thus we only need to prove that $\mathcal{O}$ is non-increasing under the updating rules for **V** in Eq. (19). Our convergence proof will follow the similar procedure in [6] by making use of an auxiliary function similar to that used in [39]. The definition of the auxiliary function is given as follows.

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

$$\mathcal{G}(\theta,\theta') \geqslant \mathcal{L}(\theta), \quad \mathcal{G}(\theta,\theta) = \mathcal{F}(\theta) \tag{23}$$

Lemma 1. *If $\mathcal{G}$ is an auxiliary function of $\mathcal{F}$, then $\mathcal{F}$ is non-increasing under the update*

$$\theta^{(t+1)} = \arg\min_{\theta} \mathcal{G}(\theta,\theta^{(t)}) \tag{24}$$

Proof.

$$\mathcal{F}(\theta^{(t+1)}) \leqslant \mathcal{G}(\theta^{(t+1)},\theta^{(t)}) \leqslant \mathcal{G}(\theta^{(t)},\theta^{(t)}) = \mathcal{F}(\theta^{(t)}). \qquad \square$$

In the following, we will show that the update rules for **V** are exactly the updates in Eq. (24) with a proper auxiliary function. For simplicity's sake, we rewrite the objective function **O** in Eq. (8) without using the weight matrix **E** as follows

$$\begin{aligned}\mathcal{O} &= \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|^2 + \alpha\|\mathbf{Y}-\mathbf{B}\mathbf{V}^T\|^2 + \beta\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V})\\ &= \sum_{i=1}^{M}\sum_{j=1}^{N}\left(x-\sum_{k=1}^{K}u_{ik}v_{jk}\right) + \alpha\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x-\sum_{k=1}^{K}u_{ik}v_{jk}\right)\\ &\quad + \beta\sum_{k=1}^{K}\sum_{j=1}^{N}\sum_{l=1}^{M}v_{jk}L_{jl}v_{lk}\end{aligned} \tag{25}$$

We use F_{ab} to denote the part of $\mathcal{O}$ which is relevant to **V**, and we have

$$F'_{ab} = \left(\frac{\partial\mathcal{O}}{\partial\mathbf{V}}\right)_{ab} = (-2\mathbf{X}^T\mathbf{U}+2\mathbf{V}\mathbf{U}^T\mathbf{U}-\alpha2\mathbf{Y}^T\mathbf{B}+2\alpha\mathbf{V}\mathbf{B}^T\mathbf{B}+2\beta\mathbf{L}\mathbf{V})_{ab} \tag{26}$$

$$F''_{ab} = 2(\mathbf{U}^T\mathbf{U})_{aa} + 2\alpha(\mathbf{B}^T\mathbf{B})_{aa} + 2\beta\mathbf{L}_{aa} \tag{27}$$

The update in Eq. (19) for **V** is element-wise, thus it is sufficient to show that each F_{ab} is non-increasing under the update step of Eq. (19).

Lemma 2. *If F_{ab} is the part of $\mathcal{O}$ which is relevant to **V**, then the auxiliary function for F_{ab} is given as follows*

$$\begin{aligned}G(v,v_{ab}^{(t)}) &= F_{ab}(v_{ab}^{(t)}) + F'_{ab}(v_{ab}^{(t)})(v-v_{ab}^{(t)})\\ &\quad + \frac{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{aa} + \alpha(\mathbf{V}\mathbf{B}^T\mathbf{B})_{aa} + \beta(\mathbf{D}\mathbf{V})_{ab}}{v_{ab}^{(t)}}(v-v_{ab}^{(t)})^2\end{aligned} \tag{28}$$

Proof. It is obvious that $G(v,v) = F_{ab}(v)$. In the follows, we show that $G(v,v_{ab}^{(t)}) \geqslant F_{ab}(v)$. First, we compare the Taylor series expansion of $F_{ab}(v)$

$$F_{ab}(v) = F_{ab}(v_{ab}^{(t)}) + F'_{ab}(v_{ab}^{(t)})(v-v_{ab}^{(t)}) \tag{29}$$

$$+ [(\mathbf{U}^T\mathbf{U})_{bb} + \alpha(\mathbf{B}^T\mathbf{B})_{bb} + \beta\mathbf{L}_{aa}](v-v_{ab}^{(t)})^2 \tag{30}$$

with Eq. (28) to find that $G(v,v_{ab}^{(t)}) \geqslant F_{ab}(v)$ is equivalent to

$$\frac{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{aa} + \alpha(\mathbf{V}\mathbf{B}^T\mathbf{B})_{aa} + \beta(D\mathbf{V})_{ab}}{v_{ab}^{(t)}} \geqslant (\mathbf{U}^T\mathbf{U})_{bb} + \alpha(\mathbf{B}^T\mathbf{B})_{bb} + \beta\mathbf{L}_{aa} \tag{31}$$

Because we have

$$(\mathbf{V}\mathbf{U}^T\mathbf{U})_{aa} = \sum_{l=1}^{K} v_{al}^{(t)}(\mathbf{U}^T\mathbf{U})_{lb} \geqslant v_{ab}^{(t)}(\mathbf{U}^T\mathbf{U})_{bb} \tag{32}$$

and

$$\alpha(\mathbf{V}\mathbf{B}^T\mathbf{B})_{aa} = \alpha\sum_{l=1}^{K} v_{al}^{(t)}(\mathbf{B}^T\mathbf{B})_{lb} \geqslant \alpha v_{ab}^{(t)}(\mathbf{B}^T\mathbf{B})_{bb} \tag{33}$$

and

$$\beta(\mathbf{D}\mathbf{V})_{ab} = \beta\sum_{j=1}^{M}\mathbf{D}_{aj}v_{jb}^{(t)} \geqslant \beta\mathbf{D}_{aa}v_{ab}^{(t)} \geqslant \beta(\mathbf{D}-\mathbf{E})_{aa}v_{ab}^{(t)} = \beta\mathbf{L}_{aa}v_{ab}^{(t)} \tag{34}$$

Thus, Eq. (31) holds and $G(v,v_{ab}^{(t)}) \geqslant F_{ab}(v)$. $\square$

Proof of Theorem 1. Replacing $G(v,v_{ab}^{(t)})$ in Eq. (24) by Eq. (28) resulting in the update rule:

$$\begin{aligned}v_{ab}^{(t+1)} &= v_{ab}^{(t)} - v_{ab}^{(t)}\frac{F'_{ab}(v_{ab}^{(t)})}{2(\mathbf{V}\mathbf{U}^T\mathbf{U})_{aa} + 2\alpha(\mathbf{V}\mathbf{B}^T\mathbf{B})_{aa} + 2\beta(\mathbf{D}\mathbf{V})_{ab}}\\ &= v_{ab}^{(t)}\frac{(\mathbf{X}^T\mathbf{U}+\alpha\mathbf{Y}^T\mathbf{B}+\beta\mathbf{E}\mathbf{V})_{ab}}{(\mathbf{V}\mathbf{U}^T\mathbf{U}+\alpha(\mathbf{V}\mathbf{B}^T\mathbf{B})_{aa}+\beta(\mathbf{D}\mathbf{V}))_{ab}}\end{aligned} \tag{35}$$

Since Eq. (28) is an auxiliary function, F_{ab} is non-increasing under this update rule.

References

[1] 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.

[2] 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.

[3] 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.
[4] 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.
[5] 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.
[6] 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.
[7] Y. Bengio, A.C. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[8] M. Keikha, A. Khonsari, F. Oroumchian, Rich document representation and classification: an analysis, Knowl.-Based Syst. 22 (1) (2009) 67–71.
[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] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective classification in network data, AI Mag. 29 (3) (2008) 93.
[11] 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.
[12] 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.
[13] J. Neville, D. Jensen, Relational dependency networks, J. Mach. Learn. Res. 8 (2007) 653–692.
[14] B. Gallagher, H. Tong, T. Eliassi-Rad, C. Faloutsos, Using ghost edges for classification in sparsely labeled networks, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 256–264.
[15] M. Bilgic, L. Getoor, Reflect and correct: a misclassification prediction approach to active inference, ACM Trans. Knowl. Discovery Data (TKDD) 3 (4) (2009) 20.
[16] H. Rahmani, H. Blockeel, A. Bender, Predicting the functions of proteins in protein-protein interaction networks from global information, in: JMLR: Workshop and Conference Proceedings, vol. 8, 2010, pp. 82–97.
[17] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, in: Advances in Neural Information Processing Systems, vol. 16, 2003, pp. 321–328.
[18] X. Zhu, Z. Ghahramani, J. Lafferty, et al., Semi-supervised learning using gaussian fields and harmonic functions, in: Proc. of the 20th International Conference on Machine Learning, vol. 3, 2003, pp. 912–919.
[19] Q. Wu, M.K. Ng, Y. Ye, X. Li, R. Shi, Y. Li, Multi-label collective classification via markov chain based learning method, Knowl.-Based Syst. 63 (2014) 1–14.
[20] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.
[21] H. Lee, J. Yoo, S. Choi, Semi-supervised nonnegative matrix factorization, Signal Process. Lett., IEEE 17 (1) (2010) 4–7.
[22] N. Lopes, B. Ribeiro, A fast optimized semi-supervised non-negative matrix factorization algorithm, in: International Joint Conference on Neural Networks (IJCNN), 2011, pp. 2495–2500.
[23] F. Lin, W.W. Cohen, Semi-supervised classification of network data using very few labels, in: International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2010, pp. 192–199.
[24] J.J. Pan, S.J. Pan, J. Yin, L.M. Ni, Q. Yang, Tracking mobile users in wireless networks via semi-supervised colocalization, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 587–600.
[25] M.-F. Weng, Y.-Y. Chuang, Collaborative video reindexing via matrix factorization, ACM Trans. Multimedia Comput. Commun. Appl. (TOMCCAP) 8 (2) (2012) 1–20.
[26] X. Luo, M. Zhou, Y. Xia, Q. Zhu, An efficient non-negative matrix-factorization-based approach to collaborative-filtering for recommender systems, IEEE Trans. Industr. Inf. 10 (2) (2014) 1273–1284.
[27] M.W. Berry, M. Browne, A.N. Langville, V.P. Pauca, R.J. Plemmons, Algorithms and applications for approximate nonnegative matrix factorization, Comput. Statist. Data Anal. 52 (1) (2007) 155–173.
[28] S.Z. Li, X. Hou, H. Zhang, Q. Cheng, Learning spatially localized, parts-based representation, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, CVPR 2001, vol. 1, 2001, pp. I–207.
[29] W. Xu, X. Liu, Y. Gong, Document clustering based on non-negative matrix factorization, in: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003, pp. 267–273.
[30] C.-X. Yin, Q.-K. Peng, A careful assessment of recommendation algorithms related to dimension reduction techniques, Knowl.-Based Syst. 27 (2012) 407–423.
[31] P.O. Hoyer, Non-negative sparse coding, in: Proceedings of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing, 2002, pp. 557–565.
[32] Y. Wang, Y. Jia, Fisher non-negative matrix factorization for learning local features, in: Proceedings of Asian Conference on Computer Vision, 2004, pp. 27–30.
[33] J. Yang, S. Yang, Y. Fu, X. Li, T. Huang, Non-negative graph embedding, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, CVPR 2008, 2008, pp. 1–8.
[34] X. He, D. Cai, Y. Shao, H. Bao, J. Han, Laplacian regularized gaussian mixture model for data clustering, IEEE Trans. Knowl. Data Eng. 23 (9) (2011) 1406–1418.
[35] 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.
[36] H.N. Chua, W.-K. Sung, L. Wong, Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions, Bioinformatics 22 (13) (2006) 1623–1630.
[37] C. Chang, C. Lin, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011) 27.
[38] X. Kong, X. Shi, P.S. Yu, Multi-label collective classification, in: SIAM International Conference on Data Mining (SDM), 2011, pp. 618–629.
[39] 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.