Seamlessly Integrating Effective Links with Attributes for Networked Data Classification

Yangyang Zhao[1(✉)], Zhengya Sun[1], Changsheng Xu[2], and Hongwei Hao[1]

[1] IDMTech, Institute of Automation, Chinese Academy of Sciences, Beijing, China
{yangyang.zhao,zhengya.sun,hongwei.hao}@ia.ac.cn
[2] NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
csxu@nlpr.ia.ac.cn

Abstract. Networked data is emerging with great amount in various fields like social networks, biological networks, research publication networks, etc. Networked data classification is therefore of critical importance in real world, and it is noticed that link information can help improve learning performance. However, classification of such networked data can be challenging since: 1) the original links (also referred as relations) in such networks, are always sparse, incomplete and noisy; 2) it is not easy to characterize, select and leverage effective link information from the networks, involving multiple types of links with distinct semantics; 3) it is difficult to seamlessly integrate link information with attribute information in a network. To address these limitations, in this paper we develop a novel Seamlessly-integrated Link-Attribute Collective Matrix Factorization (SLA-CMF) framework, which mines highly effective link information given arbitrary information network and leverages it with attribute information in a unified perspective. Algorithm-wise, SLA-CMF first mines highly effective link information via link path weighting and link strength learning. Then it learns a low-dimension link-attribute joint representation via graph Laplacian CMF. Finally the joint representation is put into a traditional classifier such as SVM for classification. Extensive experiments on benchmark datasets demonstrate the effectiveness of our method.

Keywords: Networked data classification · Heterogeneous information fusion · Collective matrix factorization

1 Introduction

In recent years, with the advance of the World Wide Web and social networks such as Twitter, YouTube, Facebook and Flickr, more and more networked data are available on the web. Compared with traditional data, the networked data brings us a lot of extra meaningful link information besides their attribute (content) information. In the majority cases, such data contains more than one types of entities and links, and is always referred as Heterogeneous Information Network [5,11].

T. Cao et al. (Eds.): PAKDD 2015, Part II, LNAI 9078, pp. 473–484, 2015.
DOI: 10.1007/978-3-319-18032-8_37

The link information in these networks has been proved beneficial for classification [2,6,9,11] in many works. However, classification of such networked data can be challenging since: 1) the original links (also referred as relations) in such networks, are always sparse, incomplete and noisy; 2) it is not easy to characterize, select and leverage effective link information from the networks, involving multiple types of links with distinct semantics; 3) it is difficult to seamlessly integrate link information with attribute information in a network.

A great deal of recent works have shown their interests in networked data classification and try to utilize link information in a network to enhance the classification performance, unfortunately, none of them simultaneously address the three challenges well, as far as we know. Information fusion based methods

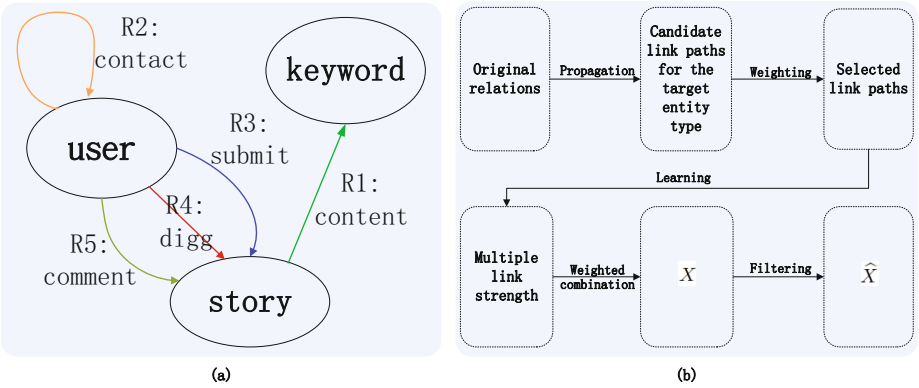

Fig. 1. (a)Entity types and original relations in Digg dataset;(b)The process of mining effective link information

[2] try to combine the link information and attribute information together and perform quite well because of their ability to exploit correlation between the two aspects of the network. These methods treat the link information as another kind of attribute which directly decompose the original link information and attribute information simultaneously. But their results are always limited to noises and the sparsity of the links. Graph-based techniques [6] treat the link information in the view of manifolds which assume that the linked nodes have similar labels. They are sensitive to the quality of the graph and inclined to fail when the network is intrinsically of low label consistency. Collective classification based methods [7–9,11] predict the unlabeled nodes with the help of related labeled nodes. What makes it powerful for networked data classification is its great ability to learn and make use of various kinds of dependency structures. But at the same time the performance might be largely degraded due to the lack of neighbours when given a sparse network. Another obvious limitation is that most of these methods use link information and attribute information non-synchronously so that the links and attributes cannot be integrated well.

To summarize, most of the existing research suffers from three limitations: 1) only utilizing the original link information but ignoring mining richer semantics conveyed by the link paths; or 2) intuitively selecting and indiscriminately utilizing different types of link paths; and/or 3) utilizing link information and attribute information in distinct perspectives.

In this study, we mine highly effective link information given arbitrary information network (as shown in Fig.1(b)) and leverage it with attribute information in a unified perspective. First, we learn the weights of different link paths under the guidance of a few labeled nodes sampled from the training set. Second, we learn link strength of the selected link paths, and get a weighted combination of all the selected link paths. Then in a unified perspective, we learn a low-dimension link-attribute joint representation via CMF for networked data classification. Finally the joint representation is put into a traditional classifier such as SVM for classification. Experiments on three real datasets demonstrate the effectiveness of our SLA-CMF method.

The primary contributions of SLA-CMF are as follows.

- We propose to integrate link path weighting, link strength learning and graph-regularized CMF to select, characterize and leverage effective link information, and it is finally seamlessly integrated with attribute information.
- We adopt a simple but effective strategy to learn the weights of different link paths, which can be used for link path selection. Then we propose a general link path based similarity computing method, through which the strength of either symmetric or asymmetric link path can be accurately characterized. These two processes not only mine richer semantics conveyed by link paths, but also alleviate noise and sparsity of link information. (Section 4.1, 4.2).
- We treat both the links and the attributes in a unified perspective. Precisely, we treat the attributes as a kind of link path when characterizing the link structure. Meanwhile, we treat the link information as a kind of attribute when directly decomposing it in an attribute perspective. This scheme ensures the seamless integration by mutual penetration. (Section 4.3).

2 Related Work

In this section, we review some of the research literatures related to networked data classification.

To combine text content and the explicit links for classification, information fusion based method [2] simultaneously decomposes the adjacency matrix of the original cite relation and the bag-of-word attribute matrix by collective matrix factorization and gets great enhancement on the results. However, it is under the assumption of only one type of links and therefore does not apply in heterogeneous information networks.

Unlike [2] directly learning from the link information, graph-based technique [6] interprets it in a manifolds view and utilizes it to build up a graph Laplacian

regularization to constraint the encoding matrix of attribute information. This method is efficient and can outperform [2] in some cases. But it is still lack of mining link paths so that the result will inevitably be sensitive to the outliers in the original relation.

Recent researches [7,8] propose a tensor factorization method, which has the effect of collective classification and solve the classification problem by reconstructing the appropriate slice of the class relation. The three-way tensor puts all of the entities into the tensor and every frontal slice of the tensor is an observation matrix that describes one type of relation. So this model supports the representation of Heterogeneous Information Networks and linked data classification. Unfortunately, blindly utilizing different types of links without distinction and decomposing two many types of link information simultaneously leads to strong disturb from each other.

Collective classification based method [9] seeks to combine the explicit links with the the links mined from local attribute similarity to increase the information in the network, and it adopts a node-based assortativity coefficient to combine different edges with different weights, which greatly inspires us. However, the big difference is that, it adopts wvRN-RL as the classifier and just utilizes the link information and attribute information in a link perspective while we leverage both of them in both link perspective and attribute perspective.

Another collective classification method [11] is an two-stage method, which utilizes the attributes only for bootstrap, simply adopts the generation scheme of link path defined in [5] and iteratively infer the unlabeled entities based on the neighbours with respect to different link paths. It is limited because it pays no attention to link path weights and utilizes the links and attributes separately.

3 Problem Formulation

The networked data classification in this paper is in a supervised or semi-supervised setting. Given a network with known labels for the nodes in training set, we predict the labels of the nodes in testing set. In a certain network, the content matrix and the original link (relation) adjacency matrices of different types can be easily extracted. To leverage the link information and attribute information for an enhanced classification performance, the main idea behind this paper is selecting and characterizing effective link information, and then seamlessly integrating it with attribute information to get a link-attribute joint representation that is more appropriate for classification.

First, we need to specify the type of objects in the network we will classify, and it is called the **target entity type**. If the number of entity type or that of the relation type a certain network is more than one, the network can be called a **heterogeneous information network** [5], otherwise homogeneous information network. And we call the adjacency matrix of the original link path **original adjacency matrix**.In this paper, **Link path**, or referred as meta path [5], is a path that connects object types via a sequence of relations. It includes the original relations whose length is one. In this paper, in order to

facilitate the weighted combination of different link paths, we integrate all the relations related to the target entity type, including **inter-type relation**s (e.g., $R4 : User \xrightarrow{comment} Story$ in Fig.1) and **inner-type relation**s (e.g., $R2 : User \xrightarrow{contact} User$ in Fig.1), to construct the link information of the **inner-type link path**s (e.g., $Story \xrightarrow{comment^{-1}} User \xrightarrow{comment} Story$ and $story \xrightarrow{comment^{-1}} user \xrightarrow{contact} user \xrightarrow{comment} story$, they can also be represented by multiplying adjacency matrices for each relation along the link path, i.e., ${R_4}^T \times R_4$ and ${R_4}^T \times R_2 \times R_4$). Note that, $comment^{-1}$ is the inverted relation of $comment$, which may means *commentedby*.

In the process of seamless link-attribute integration, $X_k(k = 1, \cdots, m)$ is the adjacency matrix of a certain selected link path with the size $n \times n$, where n is the number of objects of target entity type. and m is the number of the selected link paths. X is the weighted combination of selected link paths. D is the attribute information matrix with the size $n \times l$, where l is the number of attributes. Z is the latent relation space matrix, whose size is $r \times r$. V is the basis matrix in attribute information matrix factorization with the size $l \times r$. A is the encoding matrix in which every row present one object of target entity type, and the size of it is $n \times r$.

4 Details of SLA-CMF

In this section, we will detail the proposed $SLA - CMF$ model. We first introduce a link path weighting strategy by which we can characterize the importance of a certain link path and filter out less important ones whose weights are less than the threshold. Then we learn the link strength by the method proposed by us. Finally we get a weighted combination of the selected link paths, and utilize a Laplacian regularized CMF to integrate the effective link information with attribute information, and get a joint link-attribute representation.

4.1 Link Path Weighting and Selection

Different link paths always have different semantics so that have different degree of impact in label consistency. Unlike many existing work intuitively selects and indiscriminately utilizes different types of link paths, we employ a novel strategy to learn the weights of different link paths under the guidance of some sampled labeled nodes from the training set. The weight is evaluated by the correlation between a certain link path and the label consistency of the sampled nodes, which can also be seen as the conditional probability of the label consistency w.r.t. (**w**ith **r**espect **t**o) a certain link path. The strategy is very simple but effective, and the weighting function and the threshold function of the k-th link path are defined as follows.

$$Weight_k = \frac{\sum_{i=1}^{n_s} \sum_{j=1}^{n_s} Consistency(o_i, o_j) Strength(o_i \xrightarrow{LinkPath_k} o_j)}{\sum_{i=1}^{n_s} \sum_{j=1}^{n_s} Strength(o_i \xrightarrow{LinkPath_k} o_j)} \quad (1)$$

$$Threshold_k = \frac{\sum_{i=1}^{n_s}\sum_{j=1}^{n_s} Consistency(o_i, o_j) NotLink(o_i \xrightarrow{LinkPath_k} o_j)}{\sum_{i=1}^{n_s}\sum_{j=1}^{n_s} NotLink(o_i \xrightarrow{LinkPath_k} o_j)} \quad (2)$$

where n_s is the number of sampled nodes, $Strength(\ )$ is the link strength of a certain link between two nodes; $Consistency(\ )$ is a two-valued function that sets the value to be 1 if two nodes have the same label, 0 otherwise; and $NotLink(\ )$ is also is a two-valued function that sets the value to be 1 if two nodes are not linked by a certain link path, 0 otherwise. Note that, if $Weight_k$ is less than the $Threshold_k$, this link path has negative impact in label consistency and should be filtered out because it means the k-th link path makes the nodes linked by it less possible to have the same label than the nodes not linked by it.

4.2 Link Strength Learning

As the selected meta-paths are all inner-typed, we propose a link path based similarity calculation method *g-PathSim* (general Path Similarity) to characterize the strength of pairwise interactions among the objects via calculating the similarity of the two objects connected along a certain link path. The basic idea is that similar objects are not only strongly connected but also have few connection with others. Given an arbitrary link path $P : P_1 \times P_2 \times \cdots \times P_n$ of length $n(n>1)$, it can be decomposed into two shorter link paths $P_L : P_1 \times \cdots \times P_m$ and $P_R : P_{m+1} \times \cdots \times P_n$. The *g-PathSim* is defined as follows,

$$g\text{-}PathSim(o_i \xrightarrow{P} o_j) = \frac{2 * (|o_i \xrightarrow{P} o_j| + |o_j \xrightarrow{P} o_i|)}{|O(o_i \mid P_L)| + |O(o_j \mid P_L)| + |I(o_i \mid P_R)| + |I(o_j \mid P_R)|} \quad (3)$$

where $|\ |$ is a counter function, $|o_i \xrightarrow{P} o_j|$ is the number of link path instances from object o_i to o_j along the link path P_{LR}, $|O(o_i \mid P_L)|$ is the weighted out-degree of object o_i along the link path P_L, and $|I(o_i \mid P_R)|$ is the weighted in-degree of object o_i along the link path P_R. Note that if o_i and o_j are the same object, $g\text{-}PathSim(o_i, o_j \mid P_{LR})$ is directly set to 1.

W.r.t. objects of the same entity type, the *g-PathSim* of them can be calculated in matrix or vector manner as follows, where $\|*\|$ is the L2-norm function, L and R are the adjacency matrices corresponding to the left and the right link paths respectively.

$$g\text{-}PathSim(o_i \xrightarrow{P} o_j) = \frac{2 * (L_{i,*}R_{*,j} + L_{j,*}R_{*,i})}{\|L_{i,*}\|^2 + \|L_{j,*}\|^2 + \|R_{*,i}\|^2 + \|R_{*,j}\|^2} \quad (4)$$

W.r.t. a single relation, or referred as one-length link path, we add an imaginary entity type between the real object type and decompose the atomic relation into two relations as applied in [10]. The *g-PathSim* of one-length link path is calculated as follows, where P is the original adjacency matrix of the one-length link path.

$$g\text{-}PathSim(o_i \xrightarrow{P} o_j) = \frac{2(P_{i,j} + P_{j,i})}{\|P_{i,*}\|^2 + \|P_{j,*}\|^2 + \|P_{i,*}\|^2 + \|P_{j,*}\|^2} \quad (5)$$

PathSim [5] is a special case of *g-PathSim* with symmetry link paths. Compared with *PathSim* and *HeteSim* [10], the advantages of *g-PathSim* are as follows: 1) our method refers to the two objects' information of both the left and right sides, so the similarity search result is more synthesized; 2) the result maintains symmetry in arbitrary link path, so we only need to search the similarities of the upper triangular matrix at half of the computational cost.

4.3 Seamless Link-Attribute Integration

To seamlessly integrate the selected link information and attribute information, we design a graph Laplacian regularized CMF method, whose structure and the setting of X enables it to realize this goal. The objective function is

$$\begin{aligned} f &= f_{link}(A,Z) + f_{attrubute}(A,V) + f_{graphLapl.}(A) \\ &= \frac{1}{2}\|X - AZA^T\|_F^2 + \frac{\alpha}{2}\|D - AV^T\|_F^2 + \frac{\beta}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\widehat{X}_{i,j}\|A_{i,*} - A_{j,*}\|^2 \\ &= \frac{1}{2}tr(XX^T - 2AZA^TX^T + AZA^TAZ^TA^T) \\ &+ \frac{\alpha}{2}tr(DD^T - DVA^T - AV^TD^T + AV^TVA^T) + \frac{\beta}{2}tr(A^TLA) \end{aligned} \tag{6}$$

where $X = \sum_{k=1}^{m} Weight_k \cdot g\text{-}PathSim_k (k = 1, \cdots, m)$. Note that, to ensure the weighted adjacency matrix for graph Laplacian regularization highly reliable, we choose the top-K highest-weighted links for every node in X and construct a filtered adjacency matrix $\widehat{X}$ to construct L [1].

This model simultaneously decomposes link information in attribute perspective and utilizes it as graph Laplacian regularization in link perspective. Meanwhile we treat the attributes as a kind of link path (i.e., the attribute similarity) when constructing the combined graph X, so attribute information is also taking part in graph Laplacian regularization and is utilized in link perspective. Therefore the seamless integration of link information and attribute information is ensured by mutual penetration. (Our regularization means that, the more effective links and similar attributes two nodes have, the more closer the encoding vectors of them will be.)

W.r.t. optimization, we adopt an alternating projection method to learn the parameters A, Z, V . More specifically, each time we update one parameter and fix the others. This procedure will be repeated for several iterations until the termination condition is satisfied. One straightforward way to learn the parameters is to set the gradient of f w.r.t. A, Z, V to 0 and solve the corresponding linear system or nonlinear system. And the gradients of the objective function w.r.t. variable A, Z and V are as follows.

[1] $L = S - \widehat{X}$ is known as the Laplacian matrix with S being a diagonal matrix whose diagonal elements $S_{i,i} = \sum_j \widehat{X}_{i,j}$.

$$\begin{aligned}\frac{\partial f}{\partial A} &= \underline{A}ZA^TAZ + \underline{A}ZA^TAZ^T - XAZ^T - X^TAZ \\ &\quad + \alpha(\underline{A}V^TV - DV) + \beta(L^T + L)A, \\ \frac{\partial f}{\partial Z} &= A^TAZA^TA - A^TXA, \quad \frac{\partial f}{\partial V} = VA^TA - D^TA\end{aligned} \tag{7}$$

R , V can be updated directly by solving the linear system as follows.

$$Z \Leftarrow (A^TA)^{-1}A^TXA(A^TA)^{-1}, \quad V \Leftarrow D^TA(A^TA)^{-1} \tag{8}$$

As this equation of A can not be solved directly, an alternative approach is to approximate this nonlinear problem by solving only for the left A with underlines while holding the right A constant in the same way as [7,8]. The experiments show the viability of the update of A in this situation. A can be updated by

$$A \Leftarrow [XAZ^T + X^TAZ + \alpha DV - \beta(L^T + L)A][Z^TA^TAZ + ZA^TAZ^T + \alpha V^TV]^{-1} \tag{9}$$

Alternatively, the update of A can also be implemented through gradient methods, such as the conjugate gradient method and quasi-Newton methods or just the gradient descent method. In this paper we choose the gradient descent method to update the coding matrix A since the equation solution has a bit large errors especially in the early steps of updating, and the trick initializing A from the eigendecomposition of X is adopted.

5 Experiments

5.1 Datasets and Evaluation Scheme

The Cora1 [3] and Cora2 [4] datasets contain research papers from the computer science community. And we adopt the whole Cora2 and the subset EC, OS, NW, DB of Cora1. In Cora2,there is only one type of original links and the original adjacency matrix M_{PP} describes the relation $Paper \xrightarrow{cite} Paper$. We first characterize Cora2 by the link paths of $M_{PP}, M_{PP} \times {M_{PP}}^T, {M_{PP}}^T \times M_{PP}, M_{PP} \times M_{PP}$ and augment them by $M_{PAttri} \times {M_{PAttri}}^T$. And then we do link path weighting and selection, link strength learning via g-PathSim and finally get a weighted combination of the selected link paths. Although Cora2 is originally a sparse homogeneous information network, we can mine abundant and effective link information through learning. Meanwhile, the subsets of Cora1 have one more original adjacency matrix M_{PA} which is corresponding to the relation $Paper \xrightarrow{write^{-1}} Author$. It is a simple heterogeneous information network, and we augment the candidate link paths by $M_{PA} \times {M_{PA}}^T$.

The Digg [1] dataset we utilize in this paper consists of stories, users and their actions $(submit, digg, comment)$ w.r.t. the stories, as well as the explicit *contact* relation among these users, and the attribute of Digg stories is made up of keywords extracted from the story titles. In this paper we choose stories of five topics (i.e., pc games, space, pets/animals, linux/unix, political news) as

the objects of target entity type, 200 from each, as well as the related users. As described in Fig.1, $R_k(k = 1, \cdots, 5)$ are original adjacency matrices of the five original relations. The candidate link paths of this network are quite various, including symmetric link path such as $R_1 \times {R_1}^T, {R_3}^T \times R_3, {R_3}^T \times R_2 \times R_2 \times R_3$, as well as many asymmetric ones such as ${R_3}^T \times R_2, {R_3}^T \times R_2 \times R_3, {R_3}^T \times R_2 \times {R_2}^T \times R_4$. We traverse the link paths with the length constraint 4 according to cross-validation. And it is processed just via the schema described in Section 4.

W.r.t. evaluation scheme, we take accuracy as evaluation criteria, we adopt 5-fold cross validation to evaluate our method, set the rank of latent factor A to be 50, and put A into Linear SVM for classification (all as the same as [2,6]) after seamless link-attribute integration through graph regularized CMF.

5.2 Baselines and Parameters Setting

The compared approaches include the state-of-art information fusion based method Link-content MF [2], Graph based method RRMF [6], Collective classification based method RESCAL [7,8], HCC [11], wvRN-RL [9]. Meanwhile, we also compare SLA-CMF with several variants of it. Each variant differs from SLA-CMF just in one aspect while consistent in others. Among them, SLA-CMF(naive-link) is a variant with only one link path without link strength learning (just as the same as Link-content MF and RRMF), SLA-CMF(naive-link+attri.-simil.) is the variant with attribute similarity directly added to the naiva link described above, SLA-CMF(PathSim) and SLA-CMF(HeteSim) are the variants replacing g-PathSim with PathSim and HeteSim correspondingly for link strength learning, and SLA-CMF(NAC) is the one replacing our link path weighting method with node-based assortativity coefficient [9]. Note that, there is no naive link between *Story*, it is selected from ${R_3}^T \times R_3, {R_4}^T \times R_4, {R_5}^T \times R_5$.

To ensure the weighted adjacency matrix for graph Laplacian regularization highly reliable, we choose the top-K highest-weighted links for every node in X, where the K is set to be 6 through cross validation, and the relative importance parameters (i.e., α and β) are set by searching the grid of $\{0.01, 0.03, 0.1, 0.3, 1, 3\}$. And the parameters in baselines are set to respect the original settings as much as possible. In all the methods that need labeled seeds, the ratio of the sampled labeled nodes are set to be 20%.

5.3 Performance and Result Analysis

What motivates *SLA-CMF* most is the assumption that both mining effective link information and seamlessly integrating links with attributes will enhance networked data classification. This is the primary hypothesis we want to verify. Second, we want to test the validity of the link path weighting method and the link strength learning method proposed by us. Finally, we try to observe its sensitivity to the ratio of the sampled labeled nodes and the rank of the latent latent factor A.

We independently repeat the experiments for 10 times and report the best average result in Fig.2 and Fig.3. As the degree of label consistency varies in dif-

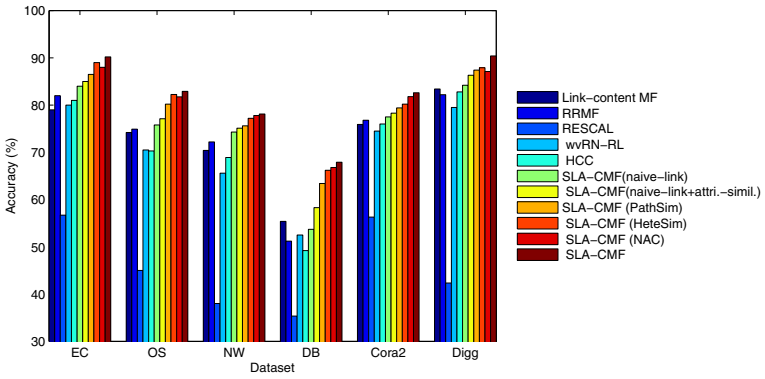

Fig. 2. Average classification accuracies of the compared methods

ferent networks, Link-content MF and RRMF have strong or weak performance respectively, which should be decided by their different mechanisms for utilizing link information. SLA-CMF(naive-link) outperforms Link-content MF and RRMF in all the datasets. As the link information and the attribute information of all the three methods are set in the same way, the result shows that, simultaneously decomposing the link information in attribute perspective and utilizing it as graph Laplacian regularization in link perspective, will lead to better result than separately processing it in either perspective. That is to say, the structure of graph Laplacian regularized CMF, which is designed to seamlessly integrating links with attributes, will indeed improve the robustness to label consistency and enhance networked data classification. Meanwhile, SLA-CMF(naive-link+attri.-simil.) performs a little better than SLA-CMF(naive-link), which indicates that utilizing attributes in link perspective can increase the information in the network and improve the performance. When it comes to other baselines, their performance is roughly consistent with the analysis of their strengths and weaknesses which can be referred in Section 2. RESCAL and HCC fail due to their ignorance of weighting different links, while HCC is also suffers from utilizing attributes and links separately. wvRN-RL is limited by only utilizing all the information in link perspective.

We can easily see that SLA-CMF, SLA-CMF(PathSim), SLA-CMF(HeteSim) and SLA-CMF(NAC) all outperform the former two SLA-CMFs, which confirms the validity of link path weighting and link strength learning. And the advantage of HeteSim and g-PtahSim to PathSim is obvious, because of their ability to computing the strength of asymmetric links and g-PathSim characterizes the link strength best. Moreover, the comparison between the performance of SLA-CMF and SLA-CMF(NAC) proves that our link path weighting method with the threshold check is more suitable for link path selection. In a word, the stable advantages shown in the experiments confirms that SLA-CMF indeed solve the 3 challenges well.

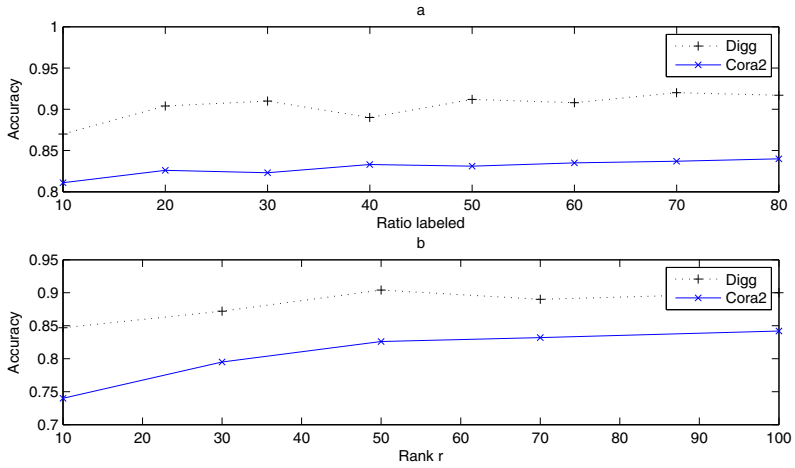

Fig. 3. (a) Average classification accuracies w.r.t. the ratio of sampled labeled nodes; (b) Average cassification accuracies w.r.t. the rank of the latent relation space

We can learn from Fig.3 that, the performance of SLA-CMF slightly improves with the increasing of the ratio of the sampled labeled nodes, but it changes very little, which may be due to the stability of our link path weighting method, or because we use a variety of information and they are related and redundancy. The performance of SLA-CMF w.r.t. the rank r of the latent factor A changes greatly when k is small, and then tend to remain unchanged. This feature indicates that SLA-CMF is suitable for networked data dimensionality reduction.

6 Conclusion and Discussion

In this paper, we propose a novel SLA-CMF framework for networked data classification, which mines highly effective link information given arbitrary information network by integrating link path weighting with link strength learning, and leverages it with attribute information in a unified perspective. First, we learn the weights of different link paths under the guidance of a few labeled nodes sampled from the training set, and utilizing these weights we select effective link paths. Second, we learn link strength of the selected link paths, and get a weighted combination of all the selected link paths. Finally, in a unified perspective, we learn a low-dimension link-attribute joint representation via CMF for networked data classification. Through these our method is enabled to solve the 3 challenges well and the experiments demonstrate its superiority for networked data classification compared with state-of-the-art approaches.

In our study, to facilitate the weighted combination of different link paths, we integrate the relations related to the target entity type to construct the link

information of the inner-type link path. This schema works well, however, there may be the other schemas that work better, and it is worth deeper researching.

Acknowledgments. The authors thank the anonymous reviewers for their valuable comments. This research work was funded by the National Natural Science Foundation of China under Grant No. 61303179.

References

1. Lin, Y., Sun, J., Castro, P., Konuru, R., Sundaram, H., Kelliher, A.: Metafac: community discovery via relational hypergraph factorization. KDD **15**, 527–536 (2009)
2. Zhu, S., Yu, K., Chi, Y., Gong, Y.: Combining content and link for classification using matrix factorization. ACM SIGIR **30**, 487–494 (2007)
3. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Automating the construction of internet portals with machine learning. Kluwer Academic Publishers Hingham. Inf. Retr. **3**(2), 127–163 (2000)
4. Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T.M., Nigam, K., Slattery, S.: Learning to extract symbolic knowledge from the world wide web. In: AAAI/IAAI, pp. 509–516 (1998)
5. Sun, Y., Han, J., Yan, X., Yu, P., Wu, T.: PathSim : Meta path-based top-k similarity search in heterogeneous information networks. In: VLDB (2011)
6. Li, W., Yeung, D.Y.: Relation regularized matrix factorization. In: IJCAI, pp. 1126–1131 (2009)
7. M. Nickel, V. Tresp and H. P. Kriegel: A three-way model for collective learning on multi-relational data. In: ICML, pp. 809–816 (2011)
8. M. Nickel, V. Tresp and H. P. Kriegel: Factorizing yago: scalable machine learning for linked data. In: WWW, pp. 271–280 (2012)
9. Sofus, A.: Macskassy: Improving Learning in Networked Data by Combining Explicit and Mined Links. In: AAAI (2007)
10. Shi, C., Kong, X., Huang, Y., Yu, P.S., Wu, B.: HeteSim: A General Framework for Relevance Measure in Heterogeneous Networks. IEEE TKDE (2013). doi:10.1109/TKDE.2013.2297920
11. Kong, X., Yu, P.S., Ding, Y., Wild, D.J.: Meta path-based collective classification in heterogeneous information networks. In: CIKM, pp. 1567–1571 (2012)
12. J. Liu, C. Wang, J. Gao and J. Han: Multi-view clustering via joint nonnegative matrix factorization. In Proceedings of the 13th SIAM International Conference on Data Mining, 252–260 (2013)