Joint Weighted Nonnegative Matrix Factorization for Mining Attributed Graphs

Zhichao Huang, Yunming Ye(✉), Xutao Li, Feng Liu, and Huajie Chen

Shenzhen Key Laboratory of Internet Information Collaboration,
Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China
{huangzhichao,liufeng,chenhuajie}@stmail.hitsz.edu.cn,
{yym,lixutao}@hitsz.edu.cn

Abstract. Graph clustering has been extensively studied in the past decades, which can serve many real world applications, such as community detection, big network management and protein network analysis. However, the previous studies focus mainly on clustering with graph topology information. Recently, as the advance of social networks and Web 2.0, many graph datasets produced contain both the topology and node attribute information, which are known as attributed graphs. How to effectively utilize the two types of information for clustering thus becomes a hot research topic. In this paper, we propose a new attributed graph clustering method, *JWNMF*, which integrates topology structure and node attributes by a new collective nonnegative matrix factorization method. On the one hand, JWNMF employs a factorization for topology structure. On the other hand, it designs a weighted factorization for nodes' attributes, where the weights are automatically determined to discriminate informative and uninformative attributes for clustering. Experimental results on seven real-world datasets show that our method significantly outperforms state-of-the-art attributed graph clustering methods.

Keywords: Attributed graph · Clustering · Weight · NMF

1 Introduction

Graph clustering is a widely studied research problem and receives considerable attention in data mining and machine learning recently [1–8]. It aims to partition a given graph into several connected components based on structural similarity. Vertices from the same component are expected to be densely connected, and the ones from different components are weakly tied. Graph clustering is popularly used in community detection, protein network analysis, etc. [4–6]. The previous work focused mainly on finding clusters by exploiting the topology structures. Recently, as the advance of social networks and Web 2.0, many graph datasets appear with both the topology and node attribute information. For example, a webpage (i.e., vertex) can be associated with other webpages via hyperlinks, and it may have some inherent attributes of itself, like the text description in

J. Kim et al. (Eds.): PAKDD 2017, Part I, LNAI 10234, pp. 368–380, 2017.
DOI: 10.1007/978-3-319-57454-7_29

the webpage. Such type of graphs are known as attributed graphs. Because the topology and attributes together offers us a better probability to find high-quality clusters, attributed graph clustering becomes a hot research topic.

However, finding clusters in attributed graphs is not trivial, and there are two important challenges we need to address. **Challenge 1**: how to effectively utilize the topology information and the attributes together. In conventional graph clustering methods, only the topology structure is exploited to find clusters. By contrast, conventional feature based clustering algorithms take merely the attributes into account. Different from the two types of approaches, attributed graph clustering algorithms should effectively use the two types of information together. **Challenge 2**: how to automatically determine the importance of different attributes? It is well-known that weighting features appropriately can help to find the inherent clusters, especially when there is a large portion of noisy features for clusters. We face the same challenge for attributed graph clustering. For example, in the aforementioned webpage example, each webpage may contain different textual information at different locations, e.g., title, body, advertisement, and features extracted thus may have distinct contributions to clusters. Although some methods have been put forward recently to address the first challenge [9–16], few of them notice the second challenge.

In this paper, we introduce a *Joint Weighted Nonnegative Matrix Factorization* method for clustering attributed graphs, namely JWNMF, which can address the two challenges. NMF [17,18] is a well-known technique, which could produce the promising performance in graph clustering [7,8,19]. For a given attributed graph, our method presents a mechanism by using joint-NMF to integrate the structural and attribute information. Specifically, we design two matrix factorization terms. One is modeling the topology structure and the other is for attributes. Meanwhile, we modify the NMF by introducing a weighting variable for each attribute, which can be automatically updated and determined in each iteration. Experiments are performed on seven real-world datasets, including two amazon information networks, one CMU email networks, one DBLP information network, one webpage links network and two citation information networks. Our experimental results show that the proposed JWNMF method outperforms state-of-the-art attributed graph clustering algorithms, like BAGC [11], PICS [13] and SANS [14].

The remainder of this paper is organized as follows: Sect. 2 reviews some existing work on attributed graph clustering. In Sect. 3, we introduce the proposed JWNMF method. Section 4 presents and discusses the experimental results. Finally, the conclusions are given in Sect. 5.

2 Related Work

Several clustering methods have been introduced for mining attributed graphs recently. They can mainly be categorized into two types, namely *distance-based methods* [9,10,14] and *model-based methods* [11–13,15,16]. The idea of distance-based methods is to design a unified distance which could combine and leverage

structural and attribute information, and then utilize existing clustering methods, e.g., k-means or spectral clustering, to cluster attributed graphs based on the unified distance. Model-based methods leverage the interactions between edges and node attributes to construct a joint model for clustering purpose.

2.1 Distance-Based Methods

Zhou *et al.* proposed a distance-based method **SA-Cluster** [9] in 2009 and its efficient version **Inc-Cluster** [10] in 2011. The key idea of the two methods is to construct a new graph by treating node attributes as new nodes and linking the original nodes to the new attribute nodes if the corresponding attribute values are non-zeros. A unified distance for the augmented graph is designed by using a random walk process. Finally, *k-mediods* is performed to partition the new augmented graph. As the augmenting step may increase the size of graphs considerable, the two methods are hard to run on large-scale attributed graphs.

SANS was introduced in 2015 [14], which partitions attributed graph leveraging both structural and node attribute information. In the method, a weighting vector is predefined. SANS chooses the node with the largest degree (out-degree plus in-degree) as a cluster center, then other nodes connected with this node are partitioned in the cluster. As a sequel, SANS assigns the clustered nodes whose attribute similarities with those assigned nodes are larger than a threshold into the cluster. After that, the weighting vector and attribute similarities are updated. The procedure is repeated until all nodes are clustered. This method can automatically partition attributed graph without pre-defined number of clusters.

2.2 Model-Based Methods

Xu *et al.* proposed a model-based approach **BAGC** in 2012 [11]. This method introduces a Bayesian probabilistic model by assuming that the vertices in same cluster should have a common multinomial distribution for each node attribute and a Bernoulli distribution for node connections. As a result, the attributed graph clustering problem can be transformed into a standard probabilistic inference problem. The clusters can be identified by using the node-to-cluster probabilities. The drawback is that this method cannot handle weighted attributed graphs. To overcome this problem, Xu extended BAGC and proposed **GBAGC** lately [12].

PICS was proposed by Akoglu in 2012 [13]. This method is a matrix compression based model clustering approach. It treats clustering problem as a data compression problem, where the structure matrix and attribute matrix are compressed at the same time. Each cluster is regarded as a compression of a dense connected subset, and the nodes in the same cluster have similar connectivity and attribute properties. Due to less computational complexity, PICS can deal with large-scale attributed graphs.

In 2014, Perozzi proposed a user interest based attributed graph clustering method, namely **FocusCo** [15]. The method utilizes the similarities of users'

interests to find an optimal clustering results for attributed graphs. **CESNA** [16] models the correlations between structures and node attributes to improve the intra-cluster similarities. The method differs from other attributed graph clustering methods in that it can detect overlapping communities in social networks.

Different from the existing studies, we propose a collective nonnegative matrix factorization method to leverage both the topology and attribute information. Moreover, we design a weighting vector to differentiate the contribution of attributes to clusters, which can be automatically determined. Our method addresses the two challenges mentioned in introduction.

3 Proposed Method

An attributed graph can be defined as $G = (V, E, A)$, where $V = \{v_1, v_2, \ldots, v_n\}$ denotes the set of nodes, $E = \{(v_i, v_j), 1 \leq i, j \leq n, i \neq j\}$ denotes the set of edges, and $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m]$ denotes the set of node attributes. In an attributed graph G, each node v_i in V is associated with an attribute vector $(a_1^i, a_2^i, \ldots, a_m^i)$, where each element of the vector is the attribute value of v_i on the corresponding attribute.

The key difference of attributed graph clustering to conventional graph clustering is that it needs take node attributes into account. Consequently, the ideal clustering results should follow two properties: (1) nodes in the same clusters are densely connected, and sparsely connected in different clusters; (2) and nodes in the same clusters have similar attribute values, and have diverse attribute values in different clusters.

3.1 Overview of NMF

Here, we will briefly review the *Nonnegative Matrix Factorization (NMF)* [17,18]. Let X denotes a $M \times N$ matrix whose data elements are all nonnegative. The goal of NMF is that to find two nonnegative matrix factors $V = (V_{i,j})_{M \times K}$ and $U = (U_{i,j})_{N \times K}$, where K denotes the desired reduced dimension of original matrix X. In general, $K \leq \min(M, N)$. After that, we can produce an approximation of X by $X \approx VU^T$. A commonly used objective function for NMF can be regarded as a *Frobenius norm* optimizing problem, as follows:

$$\min_{V,U \geq 0} \|X - VU^T\|_F^2$$

where $\|\cdot\|_F$ is the Frobenius norm and $V, U \geq 0$ represent the nonnegative constraints in matrix factorization.

3.2 Objective Function

Following the definition of attributed graphs above, we assume that S denotes the adjacency matrix for topology structure, and matrix A represents the attribute

information where rows denote nodes and columns represent attributes. In addition, we also introduce a diagonal matrix Λ to assign a weight for each attribute. Inspired by SymNMF [7,8], which often delivers promising results for graph clustering, we apply the idea for attributed graph clustering. Specifically, we have factorizations $S \approx VV^T$ and $A\Lambda \approx VU^T$, where V is a fusing representation of topology and attribute information for nodes.

In order to integrate the two approximation into the NMF framework, we propose a weighted joint NMF optimization problem over V, U, Λ:

$$\min_{V,U,\Lambda \geq 0} \|S - VV^T\|_F^2 + \lambda \|A\Lambda - VU^T\|_F^2 \tag{1}$$

where $S \in \mathbb{R}_+^{n\times n}$, $A \in \mathbb{R}_+^{n\times m}$, $\Lambda \in \mathbb{R}_+^{m\times m}$, $V \in \mathbb{R}_+^{n\times k}$, $U \in \mathbb{R}_+^{m\times k}$, $\mathbb{R}_+$ denotes the set of nonnegative real numbers, n denotes the number of nodes, m denotes the number of attribute categorizations, Λ is a diagonal matrix satisfying $\sum_{i=1}^{m} \Lambda_{i,i} = 1$ and $\lambda > 0$ is the weight to balance structural/attribute fusion and k is the number of clusters. Actually, before optimizing Eq. 1, we preprocess the adjacency matrix S and the attribute information matrix A as:

$$S = \frac{S}{\sum_{i=1}^{n}\sum_{j=1}^{n} S_{i,j}}, A = \frac{A}{\sum_{i=1}^{n}\sum_{j=1}^{m} A_{i,j}} \tag{2}$$

Next, we will derive the updating rules of V, U and Λ.

3.3 Updating Rules

Let α, β and γ denote respectively the Lagrange multiplier matrix for the constraints $V \geq 0$, $U \geq 0$ and $\Lambda \geq 0$. By using the Lagrange formulation, we obtain the loss function without constraints:

$$L = \frac{1}{2}(\|S - VV^T\|_F^2 + \lambda\|A\Lambda - VU^T\|_F^2) + Tr(\alpha^T V) + Tr(\beta^T U) + Tr(\gamma^T \Lambda)$$

Taking partial derivatives of L with respect to V, U and Λ, we have

$$\frac{\partial L}{\partial V} = -(SV + S^T V + \lambda A\Lambda U) + (2VV^T V + \lambda VU^T U + \alpha) \tag{3}$$

$$\frac{\partial L}{\partial U} = -\lambda \Lambda A^T V + \lambda UV^T V + \beta \tag{4}$$

$$\frac{\partial L}{\partial \Lambda} = -\lambda A^T VU^T + \lambda A^T A\Lambda + \gamma \tag{5}$$

In terms of *Karush-Kuhn-Tucker (KKT)* conditions $\alpha_{p,r}V_{p,r} = 0$, $\beta_{q,r}U_{q,r} = 0$ and $\gamma_{q,q}\Lambda_{q,q} = 0$, it follows that $\frac{\partial L}{\partial V} = 0$, $\frac{\partial L}{\partial U} = 0$ and $\frac{\partial L}{\partial \Lambda} = 0$. Base on these conditions, we can derive the following updating rules with respect to V, U and Λ:

$$V \longleftarrow V. * (SV + S^T V + \lambda A\Lambda U)./(2VV^T V + \lambda VU^T U) \tag{6}$$

$$U \longleftarrow U. * (\Lambda A^T V)./(UV^T V) \tag{7}$$

$$\Lambda \longleftarrow \Lambda. * (A^T V U^T)./(A^T A \Lambda) \tag{8}$$

where .* and ./ represent the elementwise multiplication and division, respectively. In order to assign the weights of Λ into a regular space, we normalize it as:

$$\Lambda = \frac{\Lambda}{\sum_{i=1}^{m} \Lambda_{i,i}} \tag{9}$$

Next, we briefly analyze the convergency and the computational complexity of above updating rules.

For proving the convergency, we just need adopt the auxiliary function described in [18]. In addition, the KKT conditions, which suffice the stationary point of the objective function, also imply the convergency of those updating rules.

Here, the computational complexity is discussed. Supposing the algorithm stops after t iterations, the overall cost for SymNMF [7,8] is $O(n^2kt)$. As the objective function adds one more linear matrix factorization term, the overall cost for updating rules is $O((n^2 + m^2 + mn)kt)$.

3.4 The Joint Weighted NMF Algorithm

By combining the parts above, our attributed graph clustering algorithm JWNMF can be summarized as follows: Firstly, we preprocess the adjacency matrix S and attribute matrix A, and randomly initialize the matrices U, V and assign the values of diagonal matrix Λ with $1/m$. Then we iteratively update matrices U, V and Λ as Eqs. (6)–(9) until it converges. Finally, *LiteKmeans*[1] is performed on the factorization result V to identify k clusters.

4 Experimental Study

In this section, we evaluate the performance of our algorithm, and compare it with three state-of-the-art attributed graph clustering methods: BAGC [11], PICS [13] and SANS [14], and a benchmark clustering approach S-Cluster which is implemented by using *LiteKmeans* and focuses only on structure information. All algorithms were implemented in Matlab R2014b, and tested on a Windows 10 PC, Intel Core i5-4460 3.20 GHz CPUs with 32 GB memory.

4.1 Datasets

Seven real-world datasets are employed in our experiments, where four of them do not have ground truth and three of them have ground truth. The datasets without ground truth include two amazon information networks (Amazon Fail[2] and Disney[3]), a CMU email address network (Enron (see footnote 2)) and a

[1] http://www.zjucadcg.cn/dengcai/Data/Clustering.
[2] http://www.ipd.kit.edu/~muellere/consub/.
[3] http://www.perozzi.net/projects/focused-clustering/.

network of DBLP information (DBLP-4AREA (see footnote 3)). On the other hand, the datasets with ground truth (WebKB, Citeseer, Cora)[4] are all from text categorization applications. We represent all of these datasets as undirected networks. Table 1 summarizes the characteristics of the seven datasets.

Table 1. Description of seven real-world datasets

Dataset	#Nodes	#Edges	#Attributes	#Clusters
Amazon Fail	1,418	3,695	21	-
Enron	13,533	176,987	18	-
Disney	124	335	25	-
DBLP-4AREA	27,199	66,832	4	-
WebKB	877	174	1,703	5
Citeseer	3,312	117	3,703	6
Cora	2,708	151	1,433	7

4.2 Evaluation Measures

The goal of attributed graph clustering is to effectively leverage the topology and attribute information. Hence, we evaluate the attributed graph clustering based on the two aspects. Specially, to evaluate clustering results from the topology structure and the attribute points of view, we employ *modularity* and *average entropy*. Modularity [20] is a widely used evaluation measure for graph partition, and average entropy is often used in evaluating feature based clustering results.

Let $C = (C_1, C_2, \ldots, C_k)$ represents the k partitions of an attributed graph, the modularity Q and average entropy $Avg\_entropy$ are defined as:

$$Q = \sum_{i=1}^{k} (e_{i,i} - c_i^2) \tag{10}$$

$$Avg\_entropy = \sum_{t=1}^{m} \sum_{j=1}^{k} \frac{|C_j|}{nm} entropy(a_t, C_j) \tag{11}$$

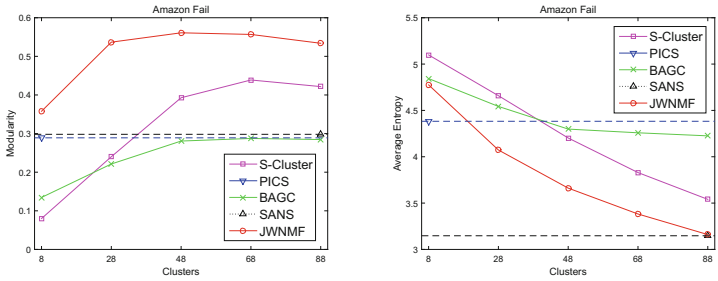
where $e_{i,j}$ is the fraction of edges with the start node in cluster i and the end node in cluster j, and c_i denotes the fraction of ends of edges that are attached to nodes in cluster i, and $entropy(a_t, C_j)$ is the information entropy of attribute a_t in cluster C_j. The value with respect to modularity and average entropy falls within the range of $[-1, 1]$ and the range of $[0, +\infty)$, where higher modularity indicates dense connections between nodes within clusters but sparse connections between nodes in different clusters, and lower average entropy indicates we have similar attribute values within clusters but dissimilar attribute values in different clusters, i.e., a better clustering result.

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

In addition to modularity and average entropy, we also utilize *Normalized Mutual Information (NMI)* to evaluate the clustering performance for the datasets with ground truth. Generally, higher NMI values indicate better clustering results.

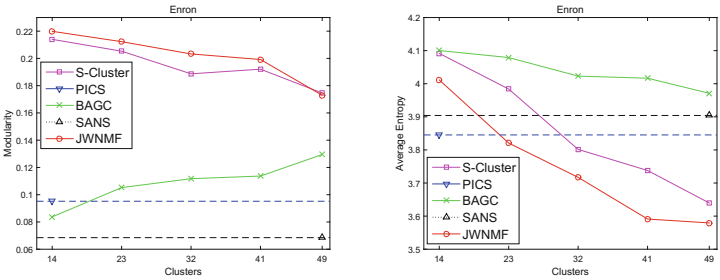
4.3 Performance on Datasets Without Ground Truth

Effectiveness Evaluation. We show how the modularity and average entropy change with respect to different number of clusters on Amazon Fail in Fig. 1. We observe JWNMF outperforms the four baseline methods in terms of modularity when varying the number of clusters. Meanwhile, in terms of average entropy, JWNMF performs the best, except when the number of clusters is set as 8. Similar observations can be found on Enron and Disney (in Figs. 2 and 3). From Fig. 4, we can see that our method achieves the lowest average entropy on DBLP-4AREA. However, according to modularity, JWNMF is inferior to PICS. The reason is that PICS treats attributed graph clustering problem as a data compression problem, thus it prefers datasets which consist of large number of nodes but sparse topology structures. Moreover, we can see from Figs. 1, 2, 3 and 4 that average entropy has a descending trend as the number of clusters is



(a) Modularity (b) Average Entropy

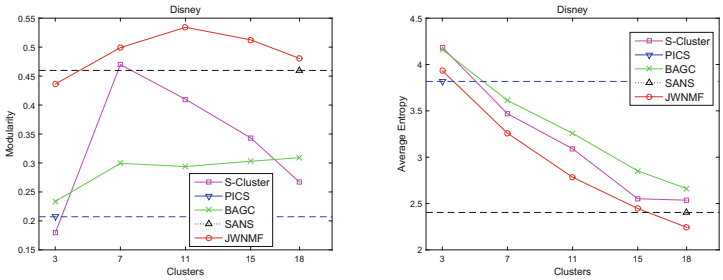
Fig. 1. Clustering qualities on Amazon Fail



(a) Modularity (b) Average Entropy

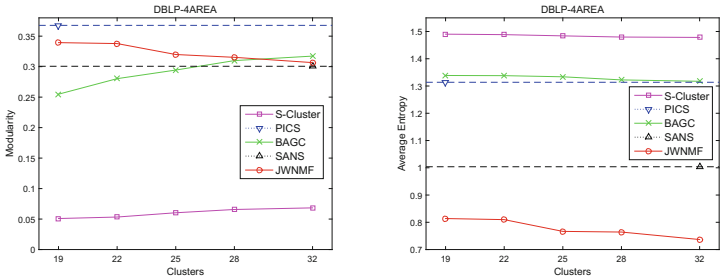
Fig. 2. Clustering qualities on Enron

increased. This is because increasing the number of clusters improves the chances that the nodes with similar attributes are put into the same cluster.



(a) Modularity (b) Average Entropy

Fig. 3. Clustering qualities on Disney



(a) Modularity (b) Average Entropy

Fig. 4. Clustering qualities on DBLP-4AREA

Efficiency Evaluation. Table 2 shows the running time of all the methods on the four datasets without ground truth. We can see JWNMF runs much faster than three sate-of-the-art attributed graph clustering methods, PICS, BAGC and SANS. The reason is that JWNMF is a quite efficient method whose iterate computation converges very fast (usually in 100 iterations). Although S-cluster achieves the best efficiency, its clustering results can be pretty poor as in Fig. 4.

Parameter Setting. In our experiments, we search the parameter λ in the set $\{10^{-10}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 0.5\}$ to find its optimal settings on Amazon Fail, Enron, Disney and DBLP-4AREA. According to our experience, we advise to set the parameter λ in terms of the sparsity of topology structures. Specifically, it is more appropriate to use a small value of λ for datasets with dense topology structure.

Table 2. Running time (sec) on datasets without ground truth

Dataset	Clusters	S-Cluster	PICS	BAGC	SANS	JWNMF
Amazon Fail	8	0.0033	9.2490	0.6575	–	0.3254
	28	0.0075	-	1.4442	–	0.3351
	48	0.0120	-	1.7249	–	0.3755
	68	0.0133	-	1.6153	–	0.4369
	88	0.0216	-	2.0932	4.2059	0.4320
Enron	14	0.4001	385.5467	360.9523	–	101.1421
	23	0.7470	-	349.1356	–	103.0285
	32	0.8329	-	319.1372	–	73.9748
	41	1.5420	-	280.1238	–	103.1282
	49	1.2942	-	250.1443	481.8581	105.3710
Disney	3	0.0017	0.1792	0.0138	–	0.0049
	7	0.0015	-	0.0201	–	0.0062
	11	0.0017	-	0.0287	–	0.0061
	15	0.0015	-	0.0137	–	0.0081
	18	0.0014	-	0.0350	0.0414	0.0074
DBLP-4AREA	19	0.2162	762.0975	1666.9570	–	367.9434
	22	0.2454	-	1663.7425	–	306.7181
	25	0.2548	-	1601.4241	–	290.4555
	28	0.2931	-	1544.3671	–	291.9064
	32	0.3422	-	1540.0352	2182.5214	367.0382

4.4 Performance on Datasets with Ground Truth

Since PICS and SANS cannot output the ground-truth of number of clusters, we do not compare with them in this section. Table 3 reports the performance for S-Cluster, BAGC and JWNMF on the three datasets with ground truth. For JWNMF, we set λ =1.5, 0.5 and 4.5 for the three datasets, respectively. Overall, our method has better performance than the baseline methods. In particular, the improvements are significant in terms of modularity and NMI. In terms of average entropy, however, the superiority of JWNMF is slight. The reason is that the textual attribute is with huge dimensions but very sparse, which makes the computed entropies more or less equal.

In JWNMF, we introduced a weighting matrix Λ to handle noisy features. To demonstrate the merits of the weighting scheme, we inject *30%* noisy attributes of random 0/1 distribution into the three datasets. Table 4 reports the results on those noisy datasets, where JNMF represents the variant of our method by removing the weighting matrix. We find that JWNMF significantly outperforms other methods including JNMF. The results show that the weighting scheme of our model is very useful, especially in the presence of noisy attributes.

Table 3. Performance on three textual datasets (%)

Dataset	Methods	Modularity	Average entropy	NMI
WebKB	S-Cluster	0.1633	23.1949	1.4282
	BAGC	−0.0260	23.2986	0.3313
	JWNMF	**33.7672**	**23.0107**	**2.1891**
Citeseer	S-Cluster	2.2419	5.9691	0.2895
	BAGC	0	5.9791	0
	JWNMF	**23.999**	**5.9565**	**0.6178**
Cora	S-Cluster	−0.2060	8.3762	0.4014
	BAGC	0	8.3963	0
	JWNMF	**25.8493**	**8.3427**	**1.5033**

Table 4. Performance on three noisy textual datasets (%)

Dataset	Methods	Modularity	Average entropy	NMI
WebKB	S-Cluster	0.1633	35.8873	1.4282
	BAGC	0	36.1062	0
	JNMF	33.7928	35.7491	**2.3884**
	JWNMF	**37.3405**	**35.7283**	2.1879
Citeseer	S-Cluster	2.2419	21.6018	0.2895
	BAGC	0	21.6352	0
	JNMF	27.4264	21.5915	0.6725
	JWNMF	**32.1148**	**21.5890**	**0.7623**
Cora	S-Cluster	−0.2060	23.5784	0.4014
	BAGC	0	23.6321	0
	JNMF	38.7895	23.5460	1.6905
	JWNMF	**41.3449**	**23.5439**	**1.8291**

5 Conclusion

In this paper, we develop a joint weighted nonnegative factorization method, namely JWNMF, to solve the attributed graph clustering problem. By using two joint factorization terms, JWNMF nicely fuses the topology and attribute information of attributed graphs for clustering. Moreover, a weighting scheme is incorporated into JWNMF to differentiate attribute importance to clusters. An iterative algorithm is proposed to find solutions of JWNMF. Extensive experimental results show that our method outperforms state-of-the-art attribute graph clustering algorithms.

Acknowledgments. The research was supported in part by NSFC under Grant Nos. 61572158 and 61602132, and Shenzhen Science and Technology Program under Grant Nos. JCYJ20160330163900579 and JSGG20150512145714247, Research Award Foundation for Outstanding Young Scientists in Shandong Province, (Grant No. 2014BSA10016), the Scientific Research Foundation of Harbin Institute of Technology at Weihai (Grant No. HIT(WH)201412).

References

1. Van Dongen, S.M.: Graph clustering by flow simulation (2001)
2. Schaeffer, S.E.: Graph clustering. Comput. Sci. Rev. **1**(1), 27–64 (2007)
3. Ng, A.Y., Jordan, M.I., Weiss, Y., et al.: On spectral clustering: analysis and an algorithm. Adv. Neural Inf. Process. Syst. **2**, 849–856 (2002)
4. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3), 75–174 (2010)
5. Brohee, S., Van Helden, J.: Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinf. **7**(1), 1 (2006)
6. Wu, Z., Leahy, R.: An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **15**(11), 1101–1113 (1993)
7. Kuang, D., Ding, C., Park, H.: Symmetric nonnegative matrix factorization for graph clustering. In: SDM, vol. 12, pp. 106–117. SIAM (2012)
8. Kuang, D., Yun, S., Park, H.: SymNMF: nonnegative low-rank approximation of a similarity matrix for graph clustering. J. Glob. Optim. **62**(3), 545–574 (2015)
9. Zhou, Y., Cheng, H., Yu, J.X.: Graph clustering based on structural/attribute similarities. Proc. VLDB Endow. **2**(1), 718–729 (2009)
10. Zhou, Y., Cheng, H., Yu, J.X.: Clustering large attributed graphs: an efficient incremental approach. In: 2010 IEEE 10th International Conference on Data Mining (ICDM), pp. 689–698. IEEE (2010)
11. Xu, Z., Ke, Y., Wang, Y., Cheng, H., Cheng, J.: A model-based approach to attributed graph clustering. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 505–516. ACM (2012)
12. Xu, Z., Ke, Y., Wang, Y., Cheng, H., Cheng, J.: GBAGC: a general bayesian framework for attributed graph clustering. ACM Trans. Knowl. Discov. Data (TKDD) **9**(1), 5 (2014)
13. Akoglu, L., Tong, H., Meeder, B., Faloutsos, C.: PICS: Parameter-free identification of cohesive subgroups in large attributed graphs. In: SDM, pp. 439–450. SIAM (2012)
14. Parimala, M., Lopez, D.: Graph clustering based on structural attribute neighborhood similarity (SANS). In: 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), pp. 1–4. IEEE (2015)
15. Perozzi, B., Akoglu, L., Iglesias Sánchez, P., Müller, E.: Focused clustering and outlier detection in large attributed graphs. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1346–1355. ACM (2014)
16. Yang, J., McAuley, J., Leskovec, J.: Community detection in networks with node attributes. In: 2013 IEEE 13th International Conference on Data mining (ICDM), pp. 1151–1156. IEEE (2013)
17. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)

18. Lee, D.D.: Algorithms for non-negative matrix factorization. Adv. Neural Inf. Process. Syst. **13**(6), 556–562 (2001)
19. Jin, D., Gabrys, B., Dang, J.: Combined node and link partitions method for finding overlapping communities in complex networks. Sci. Rep. **5**, 8 p. (2015)
20. Newman, M.E.J.: Modularity and community structure in networks. Proc. Natl Acad. Sci. **103**(23), 8577–8582 (2006)