Expert Systems With Applications 121 (2019) 256–265

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Unsupervised binary feature construction method for networked data

Arzu Gorgulu Kakisim*, Ibrahim Sogukpinar

Gebze Technical University, Computer Engineering, Gebze, Kocaeli, Turkey

ARTICLE INFO

Article history:
Received 10 July 2018
Revised 17 December 2018
Accepted 18 December 2018
Available online 18 December 2018

Keywords:
Feature construction
Feature extraction
Feature selection
Link reconstruction
Social media
Networked data

ABSTRACT

Networked data is data composed of network objects and links. Network objects are characterized by high dimensional attributes and by links indicating the relationships among these objects. However, traditional feature selection and feature extraction methods consider only attribute information, thus ignoring link information. In the presented work, we propose a new unsupervised binary feature construction method (NetBFC) for networked data that reconstructs attributes for each object by exploiting link information. By exploring similar objects in the network and associating them, our method increases the similarities between objects with high probability of being in the same group. The proposed method enables local attribute enrichment and local attribute selection for each object by aggregating the attributes of similar objects in order to deal with the sparsity of networked data. In addition, this method applies an attribute elimination phase to eliminate irrelevant and redundant attributes which decrease the performance of clustering algorithms. Experimental results on real-world data sets indicate that NetBFC significantly achieves better performance when compared to baseline methods.

1. Introduction

With the increasing use of social networks such as Flickr, Twitter, and Facebook, huge amount of data are generated by users at an unprecedented rate. Therefore, extracting useful information from such massive data has become crucial to many applications, including recommendation systems offering services to consumers (Carrer-Neto, Hernández-Alcaraz, Valencia-García, & García-Sánchez, 2012; Nicholas & Carswell, 2012), and content providers which aim at finding the right purchasers for advertising (Domingos, 2005; Mostafa, 2013).

Networked data can be defined as data with correlations and dependencies (Kolaczyk, 2009). Therefore, it differs from plain data, which is assumed to be independent and identically distributed. For instance, in a citation network, links also exist between the articles citing each other, apart from the content information of the articles. Such links intrinsically occur in many networks, such as email networks (Diesner, Frantz, & Carley, 2005) and protein-protein interaction networks (Stelzl et al., 2005). Therefore, networked data consists of two types of information: structural data to represent the relations among network objects and attribute data to define the interests of each object. However, how to properly integrate these two information types to increase the performance of learning tasks such as clustering and classification remains unclear.

Due to the fundamental nature of networked data, network objects are characterized by high dimensionality. Therefore, many learning tasks become either computationally challenging or ineffective due to the curse of dimensionality (Duda, Hart, & Stork, 2012). In order to overcome this challenge, researchers use dimension reduction methods such as feature extraction and feature selection. Feature selection methods reduce the number of attributes by selecting a subset of relevant attributes, while feature extraction methods project data into a new space with lower dimension. However, most existing feature selection and feature extraction methods use only attribute data while ignoring structural data. In recent years, numerous efforts have focused on combining attribute data with link structure (Kumar & Daumé, 2011; Le & Lauw, 2014; Tang & Liu, 2014; Wei, Xie, & Yu, 2015). Some feature selection methods enhance their selection process by employing additional steps such as object regularization and social object dimension (Hoseini & Mansoori, 2016; Li, Hu, Wu, & Liu, 2016; Tang & Liu, 2012), while some feature extraction methods focus on projecting networked data into a low-dimensional space by considering the link and attribute correlations (Huang, Li, & Hu, 2017a; 2017b). However, the integration of attribute data and structural data can be challenging due to some of the characteristics of networked data. First, attributes in networked data are always sparse. This sparsity is especially prevalent in social media data, where some data objects have very few attributes and some attributes

* Corresponding author.
E-mail addresses: arzukakisim@gtu.edu.tr (A.G. Kakisim), ispinar@gtu.edu.tr (I. Sogukpinar).

https://doi.org/10.1016/j.eswa.2018.12.030
0957-4174/

are only available for a single object (Marsden & Friedkin, 1993). In this case, it is difficult to properly calculate the proximity between network objects in order to find similar object groups. Second, attributes and links in networked data can be irrelevant, redundant, and noisy. Such attributes and links underfit or overfit the learning methods. Third, data labelling is time-consuming and expensive task for a large number of network objects (Gundecha & Liu, 2012). Supervised methods require a considerable amount of labelled data for the training phase. Therefore, unsupervised methods are becoming increasingly important. In addition, physical meaning can be important to interpret the physical process of some applications. However, the physical meaning of original attributes no longer exists when feature extraction is applied on networked data.

In this work, we propose a new unsupervised binary feature construction method for networked data. This method constructs a structurally and semantically meaningful attribute space for network objects by integrating the structural and attribute proximity among objects. To model the proximity between objects, the proposed method associates similar objects that have high probability of being in the same group. The method enables local attribute enrichment and local attribute selection for each object by applying an attribute aggregation process on networked data in order to tackle with the sparsity of networked data. Moreover, an attribute elimination process is applied to eliminate irrelevant attributes. Additionally, our method preserves the physical meaning of attributes since it performs attribute elimination instead of projecting the original space into a new low-dimensional space. By iterating our method, different attribute representations can be obtained. Experimental results on four real-world data sets demonstrate that NetBFC outperforms baseline methods.

The remainder of this article is organized as follows: We provide a brief review of the related work in Section 2, define the problem statement in Section 3, and introduce our method in Section 4. In Section 5, we provide the experimental results. The paper is concluded in Section 6.

2. Related works

Various feature selection and feature extraction methods have been proposed in literature (Abualigah, Khader, Al-Betar, & Alomari, 2017; Bennasar, Hicks, & Setchi, 2015; Bharti & Singh, 2015; Guyon & Elisseeff, 2003; Hu, Gao, Zhao, Zhang, & Wang, 2018). Some feature selection methods select the most discriminative feature subsets using l_1-norm and l_2-norm regularization (He, Tan, Wang, & Zheng, 2012; Nie, Huang, Cai, & Ding, 2010; Yang, Hou, Nie, & Wu, 2012; Yang, Shen, Ma, Huang, & Zhou, 2011), while some methods evaluate the features which can best preserve local structure of the data (He, Cai, & Niyogi, 2006; Mitra, Murthy, & Pal, 2002; Zhao, Kwok, Wang, & Zhang, 2009). However, traditional feature selection and extraction methods (He et al., 2012; He et al., 2006; Mitra et al., 2002; Nie et al., 2010; Yang et al., 2012; Yang et al., 2011; Zhao et al., 2009) assume data objects to be independent and identically distributed. The methods ignore link structure of networked data which indicates the relations among data objects. Therefore, some recent efforts have focused on using both link structure and content information to improve feature representation of networked data (Chang et al., 2015; Hoseini & Mansoori, 2016; Qi, Aggarwal, Tian, Ji, & Huang, 2012; Tang & Liu, 2012; Zhu, Yu, Chi, & Gong, 2007). In feature selection, Gu and Han (2011) present a supervised feature selection method which adds graph regularization phase to a linear regression model in order to use link information of networked data. Tang and Liu (2012) propose a semi-supervised feature selection method which extracts various social relations (e.g., CoPost, CoFollowing, Following) from networked data, and integrates them to conventional feature selection. In unsupervised scenario, Tang and Liu (2014) define social dimension concept to model the relations in networked data. Wei et al. (2015) focus on selecting attributes that best preserve the similarity between network objects and their neighbors. Li et al. (2016) learn latent representations encoded in link structure of network data and integrate these representations into feature selection phase.

In feature extraction, recent efforts rely on obtaining a joint representation by embedding link structure and content information into a unified space (Zhu et al., 2007). Qi et al. (2012) aim to learn a latent semantic space for multimedia networks. Le and Lauw (2014) formulate a probabilistic generative topic model tying together various representations of a document (words, links, topics, and coordinates) to generate a unified low dimensional representation of networked data. Huang, Li, and Hu (2017b) present a network embedding method which projects the networked data into a low-dimensional space by considering the link and attribute correlations of the data. Huang et al. (2017a) also present a distributed optimization algorithm based on decomposition of attribute and link proximity to construct an attributed network embedding model. Most recently, some methods based on deep learning algorithms are proposed to learn better representations for networked data. Chang et al. (2015) define a deep learning model that captures the relations among network objects based on content and link information, and transfers these relations into unified vector representations. Liao, He, Zhang, and Chua (2017) propose a social network embedding method to model interrelations between link structure and features using deep neural networks. In addition, multi-view learning methods (De Sa, 2005; Kumar & Daumé, 2011) are used to learn a statistical model for networked data by modelling each data obtained from multiple information sources.

There also exist some methods which propose feature enrichment approaches based on neighborhood relationships between network objects in order to create a better attribute representation for networked data. Perlich and Provost (2006) aim to enrich feature space by aggregating feature space of network objects using some operators such as min, max and sum. Cataltepe, Sonmez, and Senliol (2014) present a supervised feature enrichment approach for transductive classification. Their method aggregates feature spaces of network objects by considering neighborhood relationships in order to learn unknown labels of some network objects.

3. Problem definition

Networked data is defined as a graph with node attributes. As a mathematical abstraction, networked data can be represented as $G = \{V, E, F\}$ where $V = \{v_1, v_2, \ldots, v_N\}$ is a finite set of nodes, $E \subseteq V \times V$ is a finite set of edges, and $F = \{f_1, f_2, \ldots, f_D\} \in R^{N \times D}$ denotes the attributes of each node in G. f_{v_i} is the D-dimensional attribute vector of node $v_i \in V$. Each edge $e_{v_i,v_j} \in E$ denotes the link between node v_i and node v_j. The link structure of G is represented by a link matrix $L \in R^{N \times N}$ where $l_{v_i,v_j} = 1$ if $e_{v_i,v_j} \in E$, and $l_{v_i,v_j} = 0$ otherwise.

Using the terminologies above, we define the problem of unsupervised feature construction as follows: Given an attribute matrix F and a link matrix L, the aim is to obtain constructed attribute matrix $X = \{x_{v_1}, x_{v_2}, \ldots, x_{v_N}\} \in [\{0, 1\}]^{N \times S}$ where x_i represents S-dimensional vector of node v_i. As a result, it is expected that more efficient learning will be achieved with X, when compared to the original attribute matrix F.

In networked data, the links among nodes and node attributes tend to correlate with each other (Kolaczyk, 2009). To explain this correlation, researchers have revealed two main effects: homophily and social influence. Homophily is defined as the tendency of nodes with similar characteristics to form links with each other

(McPherson, Smith-Lovin, & Cook, 2001). Social influence is defined as the tendency of nodes to establish similar attributes with the nodes in the same group (Friedkin, 2006). Therefore, two proximity definitions are provided in the literature (Huang et al., 2017a; Liao et al., 2017) to observe these two effects in networked data. These are structural proximity and attribute proximity. As the name suggests, structural proximity is the proximity determined by the similarity of links, and attribute proximity is the proximity determined by the similarity of attributes. The proximity between two nodes can be observed either directly or indirectly. The direct proximity exists between two linked nodes, and it indicates first-order proximity. The indirect proximity between two nodes corresponds to the second-order proximity, which indicates neighborhood similarity.

In this work, we strive to reconstruct attributes and links for each node by utilizing the direct and indirect relationships of the node. Thus, by exposing contextual and structural relationships between similar objects, we can increase the similarity between nodes that have a high probability of being in the same cluster. Therefore, we provide Def. 1 that redefines the proximity between nodes to obtain second-order proximity based on attribute proximity and structural proximity.

Definition 1. (attribute-link proximity). The attribute-link proximity ρ_{AS} is the second-order proximity determined by both attribute similarity ρ_A and neighborhood similarity ρ_S. The attribute-link proximity $\rho_{AS}(v_i, v_j)$ between node $v_i \in V$ and node $v_j \in V$, is equal to $\rho_A(v_i, v_j)^* \rho_S(v_i, v_j)$.

Considering a network evolving with the effect of homophily and social influence, a higher degree of second-order proximity can be observed between two nodes that are likely to be in the same cluster. Therefore, we can capture structural and contextual relationships between nodes based on attribute-link proximity. To explore the indirect relationships in networked data, we reveal two notions. First, an indirect relationship exist between two nodes that are likely to be in the same cluster; in other words, between those with high attribute-proximity. Second, a node has some indirect attributes that make it more compatible with a group of nodes. Therefore, we provide the definition of the two notions in Def. 2 and Def. 3 in order to explore indirect attributes and relationships for a node.

Definition 2. (top-k-proximity neighbors). The top-k-proximity neighbors of node $v_i \in V$ are the members of the set of k nodes possessing the highest attribute-link proximity with node v_i.

Definition 3. (z-common attributes). The z-common attributes of node $v_i \in V$ are the members of the set of the most frequent z attributes belonging to top-k-proximity neighbors of v_i.

By obtaining top-k-proximity neighbors and z-common attributes for each node in network, nodes are associated with a group of nodes and with a group of attributes.

4. Proposed feature construction method: NETBFC

In this section, we first describe and formulate our proposed work. Then, we present the algorithm of the proposed work.

In this work, to reconstruct attribute space of each node, we aim to find its indirect relationships and indirect attributes in an iterative manner. To obtain the indirect relationships belonging to each node, nodes are associated with a node group to which it is similar. Therefore, attribute-link proximity is used to find the node group belonging to each node in networked data. To calculate attribute-link pairwise proximity values, firstly, attribute proximity matrix ρ_A and structural proximity matrix ρ_S are calculated by using Cosine similarity metric. The value of attribute proximity $\rho_A(v_i, v_j)$ between node v_i and node v_j is obtained with:

$$\rho_A(v_i, v_j) = \begin{cases} \dfrac{f_{v_i} f_{v_j}}{\|f_{v_i}\| \, \|f_{v_j}\|}, & if \ \dfrac{f_{v_i} f_{v_j}}{\|f_{v_i}\| \, \|f_{v_j}\|} > 0 \\ \gamma, & otherwise. \end{cases} \tag{1}$$

where γ is a non-zero constant and $\|f_{v_i}\|$ is the Frobenius norm of f_{v_i}. In a similar manner, $\rho_S(v_i, v_j)$ is obtained with:

$$\rho_S(v_i, v_j) = \begin{cases} \dfrac{l_{v_i} l_{v_j}}{\|l_{v_i}\| \, \|l_{v_j}\|}, & if \ \dfrac{l_{v_i} l_{v_j}}{\|l_{v_i}\| \, \|l_{v_j}\|} > 0 \\ \gamma, & otherwise. \end{cases} \tag{2}$$

where $\|l_{v_i}\|$ is the Frobenius norm of l_{v_i}. γ is used to avoid the effect of zero value. γ must be greater than zero to hold the value of the link proximity and the value of the attribute proximity of the node. The attribute-link proximity matrix ρ_{AS} is the n-square matrix obtained by the Hadamard multiplication ($\bullet$) of ρ_A and ρ_S.

$$\rho_{AS} = ((\rho_A)^{\alpha} \bullet (\rho_S)^{\beta})_N \tag{3}$$

The product of $\rho_A(v_i, v_j)$ and $\rho_S(v_i, v_j)$ indicates the value of attribute-link proximity between node v_i and node v_j. The parameters α and β are used to control the level of contribution of ρ_A and ρ_S, respectively. To generate a more contextual representation for the nodes, the value of α can be chosen higher than the value of β. On the other hand, the value of β can be chosen higher than the value of α to ensure that more structural representation is generated. It is expected that different α and β values provide different outcomes.

After obtaining attribute-link proximity, each node is associated with a node group to which it is similar in order to obtain the indirect relations belonging to the node. Therefore, we determine the set of top-k-proximity neighbors for each node. For node v_i, the top-k-proximity neighbors kp^i are selected as given in Eq. (4). The function of $max(u, k)$ returns the indices of k nodes that are closest to u. In some cases, the node may not have k neighbors whose proximity values are greater than γ^2. Therefore, k' indicates the number of obtained neighbors for the node.

$$[kp^i_1, kp^i_2, \ldots, kp^i_{k'}] = max(\rho_{AS}(v_i, :) > \gamma^2, k) \tag{4}$$

The set of top-k-proximity neighbors for each node contains direct or indirect neighbors of the node. The links are retained if the node is linked with the nodes in the set of top-k-proximity neighbors. A link is created between two nodes, if a node is not directly adjacent to the node. Therefore, the link vector of the node is rearranged by preserving some direct links and adding some indirect links. For node v_i, there is a link from node v_i to node v_j if node v_j is one of the top-k-proximity neighbors of node v_i. Thus, new link vector l_{v_i} is constructed. l_{v_i, v_j} is equal to 1 if there is a link from node v_i to node v_j, and is zero otherwise.

Some nodes in network have very few attributes that make them difficult to be clustered or be classified correctly. Therefore, the node can be associated with some indirect attributes to be enriched its attribute space. We first create a joint attribute vector by aggregating the attribute vectors of top-k-proximity neighbors. Aggregated attribute vector AF for node v_i is obtained as follows:

$$AF_{v_i} = \sum_{j \in [kp^i_1, kp^i_2, \ldots, kp^i_{k'}]}^{k'} f_{v_j} \tag{5}$$

By utilizing aggregated attribute vector, new binary attribute vector is constructed for node v_i. Therefore, the z-common attributes from the aggregated attribute vector AF_{v_i} are selected. To decide which attributes to select for a node, the probability of selecting each attribute is calculated. The probabilities of selecting

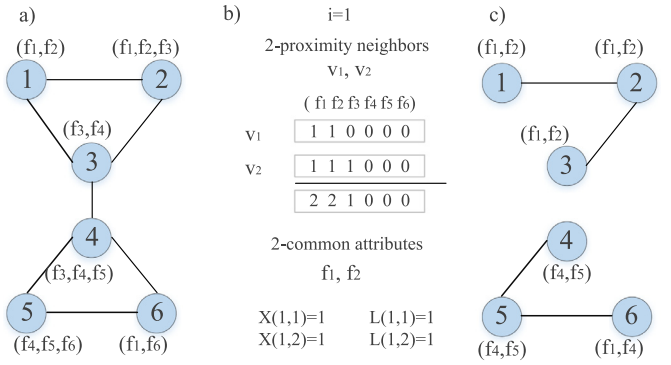

Fig. 1. An example to illustration the process of attribute and link reconstruction.

attributes belonging to node v_i are calculated as:

$$sp_f^i = \frac{AF_{v_i,f}}{k'} \tag{6}$$

where sp_f^i is the probability of selecting attribute f for node v_i. For node v_i, the set of z-common attributes are obtained with:

$$[zp_1^i, zp_2^i, \ldots, zp_{z'}^i] = max([sp_1^i, sp_2^i, \ldots, sp_D^i] > 0, z) \tag{7}$$

where zp^i refers to the set of selected attributes for node v_i. The set of z-common attributes contain the direct or indirect attributes for the node. There may not always be z attributes whose probabilities are greater than zero. Therefore, z' indicates the number of obtained common attributes. To rearrange the attribute vector x_{v_i} of node v_i, the values of indirect attributes are specified, and the value of the direct attributes are preserved. $x_{v_i,f}$ is equal to 1 if the attribute f is one of the z-common attributes of node v_i, and is zero otherwise.

Algorithm 1 NetBFC.

Require: $E, F, N, k, z, iter, \alpha, \beta$
Ensure: X
  $L \leftarrow E + I$, $X \leftarrow F$
  for $t = 1$ to $iter$ **do**
    $D \leftarrow$ dimension of X
    Obtain ρ_A^t by Eq. 1 using X^t
    Obtain ρ_S^t by Eq. 2 using L^t
    Obtain ρ_{AS}^t by Eq. 3 using α and β
    for $i = 1$ to N **do**
      Obtain $[kp_1^i, kp_2^i, \ldots, kp_{k'}^i]^t$ by Eq. 4 using k
      $l^{t+1}_{v_i, v_{[kp_1^i, kp_2^i, \ldots, kp_{k'}^i]^t}} \leftarrow 1$
      $l^{t+1}_{v_i, v_{\sim[kp_1^i, kp_2^i, \ldots, kp_{k'}^i]^t}} \leftarrow 0$
      Obtain $[sp_1^i, sp_2^i, \ldots, sp_D^i]^t$ by Eq. 5 and by Eq. 6
      Obtain $[zp_1^i, zp_2^i, \ldots, zp_{z'}^i]^t$ by Eq. 7 using z
      $x^{t+1}_{v_i, [zp_1^i, zp_2^i, \ldots, zp_{z'}^i]^t} \leftarrow 1$
      $x^{t+1}_{v_i, \sim[zp_1^i, zp_2^i, \ldots, zp_{z'}^i]^t} \leftarrow 0$
    end for
    for $i = 1$ to D **do**
      if $\sum_{j=1}^{N} x^{t+1}_{v_j,i} == 0$ **then**
        Mark attribute i to eliminate
      end if
    end for
    Delete marked attributes from X^{t+1}
    $D \leftarrow$ dimension of X^{t+1}
  end for
  Return X

Table 1
Detailed information of the data sets.

Property	WebKB	Cora	Citeseer	BlogCatalog
# of nodes	647	2708	3312	5196
# of features	1703	1433	3703	8189
# of links	1063	5429	4598	171,743
# of classes	5	7	6	6
Avg. homophily	0.14	0.82	0.71	0.39

In Algorithm 1, the proposed method NetBFC is given in detail. Given $G = \{V, E, F\}$ with N nodes and D attributes, the algorithm generates a constructed attribute matrix $X \in [\{0, 1\}]^{N \times S}$ according to the parameters k, z, α and β. First, the link matrix L is equalized to the sum of the edge matrix E and the identity matrix I of E, and the constructed attribute matrix X is equalized to the attribute matrix F. We initially set $i = 1$. Thus, by starting from node v_1, the set of top-k-proximity neighbors kp^1 and the aggregated attribute vector AF_{v_1} are obtained. The probabilities of selecting attributes sp^1 for node v_1 are calculated. According to probabilities, the z-common attributes of node v_1 are selected. Then, constructed attribute vector x_{v_1} and constructed link vector l_{v_1} are obtained for node v_1. The above mentioned aggregation and selection steps are applied locally for each node in network. By using the parameter $iter$, these steps can be repeated. X^{t+1} indicates the constructed matrix in $(t + 1)$th iteration. These processes can be iterated until L is converged to a fixed link structure. At the end of each iteration, if some attributes are not associated with any nodes in the network, these attributes are eliminated from X. The attribute f is deleted from the constructed matrix X if $\sum_{j=1}^{N} x_{v_j,f}$ is equal to 0. The reduction in the dimensionality of X varies depending on the selected k, z and $iter$ parameters. For this reason, a desired level of reduction in the dimensionality of X is achieved by selecting appropriate parameters.

An illustration of a simple networked data is shown in Fig. 1. Fig. 1(a) is a network consisting six nodes and six attributes. Fig. 1(b) shows how the link structure and the attributes for node v_1 are reconstructed when both k and z are set to 2. When $iter$ is equal to 3, the obtained final attributes and the reconstructed link structure are given in Fig. 1(c).

5. Experimental results

In this section, we evaluate the performance of the proposed feature construction method by using a clustering algorithm. We test if our method generates an improved attribute space for a clustering task compared to baseline methods. Thereafter, we also investigate how the results of these methods are affected when our method is used by these methods as a pre-processing step. Finally, we analyze the parameters of the proposed method.

5.1. Data sets

Experiments were conducted on real data sets from the literature (Cataltepe et al., 2014; Huang et al., 2017b; Li et al., 2016). We used four publicly available real-world data sets with different domains ranging from citation networks to social networks. The statistics of the data sets are summarized in Table 1. The WebKB data set consists of web pages belonging to the computer science departments of three universities (Cornell University, The University of Texas and The University of Wisconsin). Two web pages are interlinked if a hyperlink exists between them. The Cora data set (Sen et al., 2008) includes machine learning papers and their relations. The Citeseer data set (Sen et al., 2008) is composed of a set of scientific publications and their relationships. For Cora and Citeseer data sets, each node corresponds to one pub-

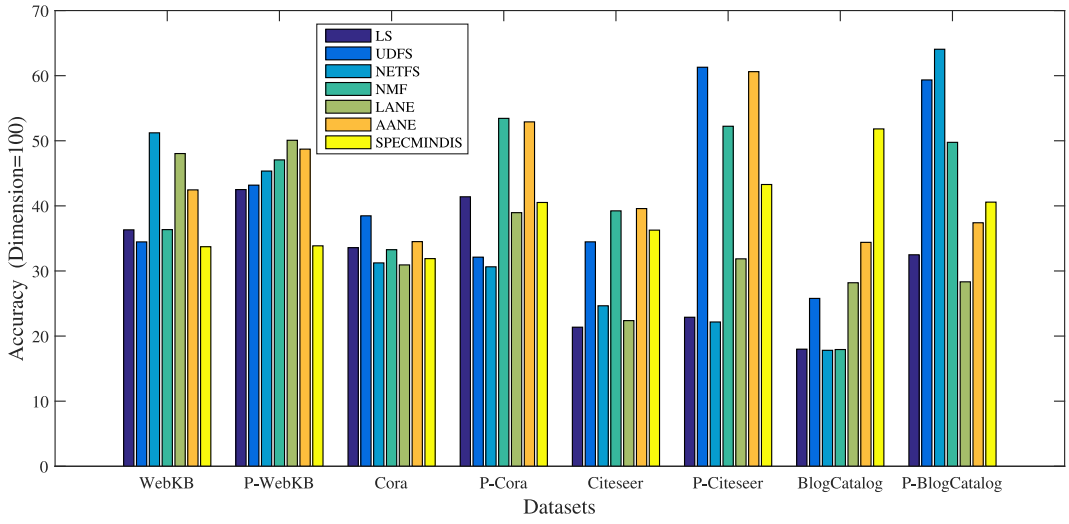

Fig. 2. Clustering accuracy results of different methods with and without NetBFC.

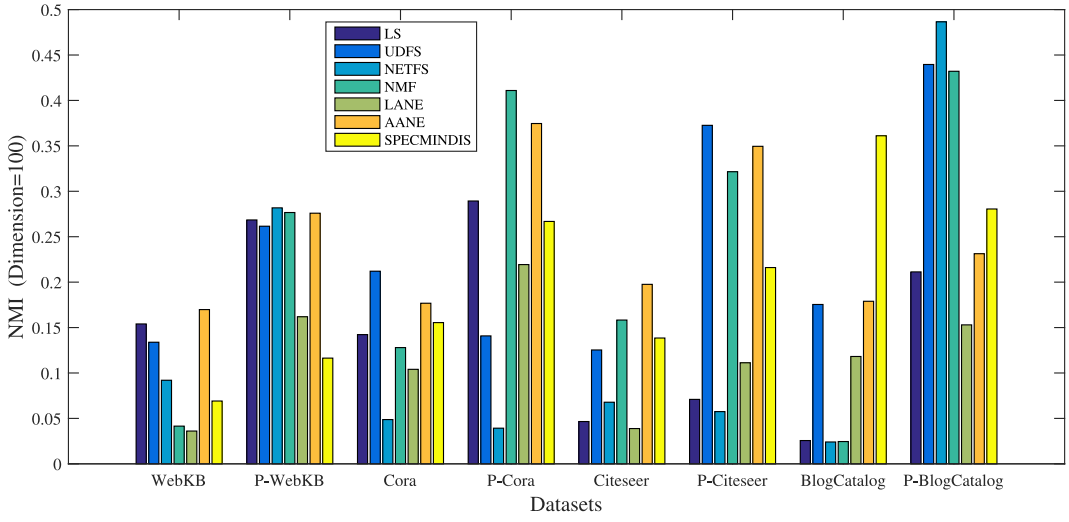

Fig. 3. Clustering NMI results of different methods with and without NetBFC.

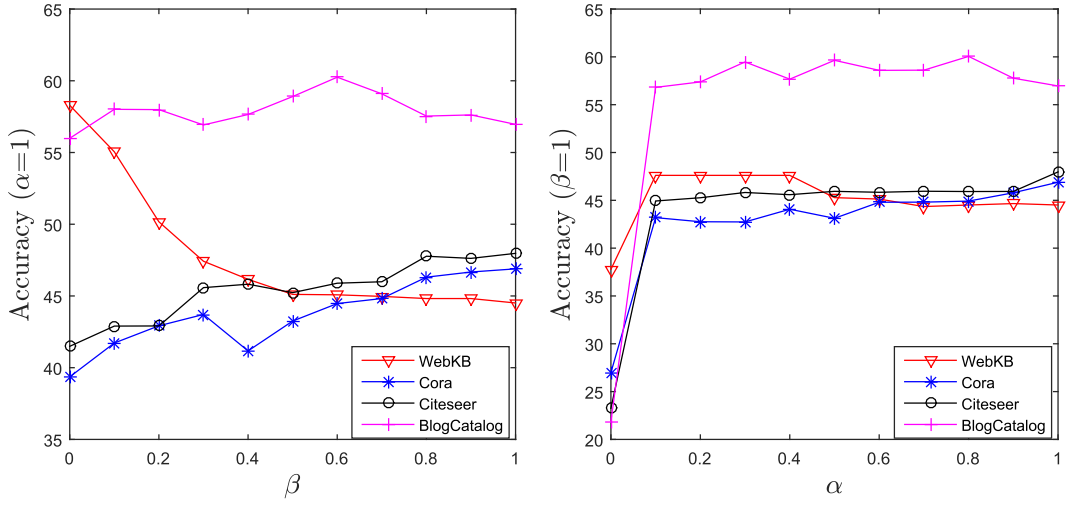

Fig. 4. Clustering accuracy results of NetBFC with different values of α and β.

Table 2
Clustering accuracy results of different algorithms on different data sets.

	WebKB				Cora				Citeseer				BlogCatalog			
Methods	100	200	400	600	100	200	400	600	100	200	400	600	100	200	400	600
Features only	44.61	44.61	44.61	44.61	32.39	32.39	32.39	32.39	38.53	38.53	38.53	38.53	18.12	18.12	18.12	18.12
LS	35.47	38.67	38.51	43.14	35.41	35.79	36.89	38.21	21.35	21.37	21.68	29.07	17.98	17.99	18.20	18.78
UDFS	51.78	53.45	54.73	**62.09**	30.68	31.24	35.47	35.65	24.66	28.46	42.91	43.11	17.80	17.81	17.83	18.01
NETFS	39.57	41.19	46.57	49.29	38.49	38.64	41.14	42.23	38.63	40.12	42.01	42.82	29.31	28.41	26.62	26.61
NMF	36.35	36.15	38.21	41.51	33.27	36.80	36.29	35.41	39.24	43.78	47.28	46.91	17.84	18.59	18.27	17.96
LANE	45.75	48.23	44.32	41.49	41.62	35.06	37.23	38.76	41.63	37.17	35.18	38.52	44.77	48.74	42.92	54.34
AANE	44.15	44.24	46.45	46.97	39.42	37.48	40.83	42.33	39.52	43.48	47.90	45.91	44.12	43.41	42.35	43.56
SPECMINDIS	33.72	35.92	39.31	32.92	31.89	25.78	24.85	23.44	36.27	29.61	24.98	24.08	51.82	35.71	31.13	28.64
NetBFC	**62.75**	**68.31**	**62.53**	51.91	**62.58**	**62.24**	**62.26**	**62.94**	**56.78**	**58.62**	**59.59**	**60.12**	**54.26**	**58.96**	**61.19**	**63.74**

Table 3
Clustering NMI results of different algorithms on different data sets.

	WebKB				Cora				Citeseer				BlogCatalog			
Methods	100	200	400	600	100	200	400	600	100	200	400	600	100	200	400	600
Features only	0.192	0.192	0.192	0.192	0.150	0.150	0.150	0.150	0.167	0.167	0.167	0.167	0.038	0.038	0.038	0.038
LS	0.154	0.206	0.229	0.231	0.152	0.161	0.175	0.184	0.046	0.044	0.041	0.092	0.025	0.026	0.036	0.043
UDFS	0.093	0.117	0.143	0.276	0.049	0.041	0.163	0.179	0.068	0.067	0.196	0.211	0.024	0.022	0.024	0.025
NETFS	0.176	0.236	0.261	0.271	0.205	0.242	0.259	0.185	0.239	0.234	0.255	0.201	0.212	0.187	0.187	0.178
NMF	0.042	0.031	0.026	0.027	0.128	0.181	0.176	0.157	0.159	0.210	0.239	0.235	0.025	0.028	0.026	0.034
LANE	0.027	0.025	0.017	0.019	0.261	0.158	0.178	0.145	0.242	0.177	0.165	0.171	0.335	0.407	0.384	0.341
AANE	0.193	0.161	0.204	0.251	0.216	0.226	0.248	0.252	0.222	0.228	0.271	0.249	0.289	0.264	0.250	0.261
SPECMINDIS	0.069	0.056	0.064	0.059	0.156	0.092	0.077	0.054	0.139	0.073	0.037	0.027	0.361	0.186	0.138	0.099
NetBFC	**0.262**	**0.310**	**0.347**	**0.341**	**0.379**	**0.436**	**0.444**	**0.457**	**0.295**	**0.319**	**0.340**	**0.348**	**0.374**	**0.417**	**0.449**	**0.477**

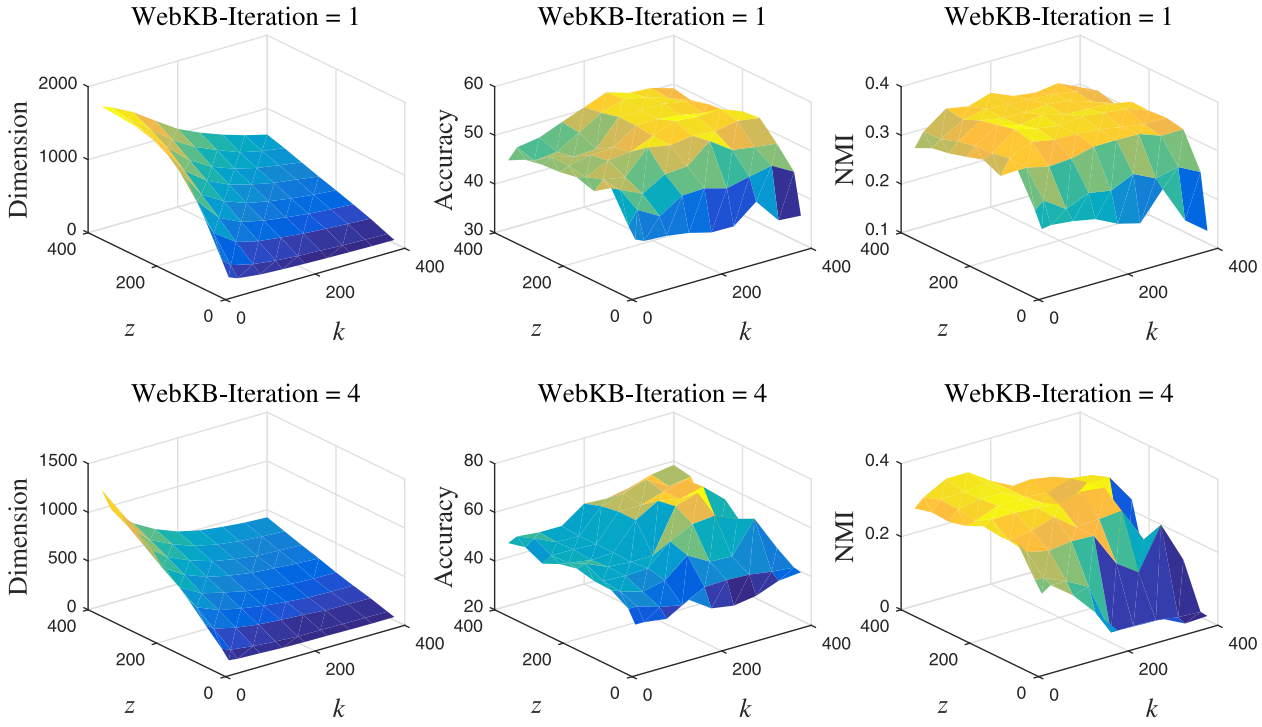

Fig. 5. Clustering performance of NetBFC on WebKB with different values of k and z.

lication, while each relation refers to the citation. The BlogCatalog data set is a subset of the data that is crawled from an online blogger community. A link exists between two bloggers if they follow each other. The bloggers use tags to specify their content. Each page/word/tag is described by a 0/1-valued word vector indicating the absence/presence of the corresponding page/word/tag from the dictionary. In Table 1, we also provide the values of average homophily for all data sets. The degree of homophily for a node in a network is determined by the number of its neighbors that possess the same label as that node (Cataltepe et al., 2014).

5.2. Evaluation criteria

In this section, we evaluate the effectiveness of the proposed feature construction method by comparing its impact on a known clustering algorithm. To measure the result of a clustering task, Normalized Mutual Information (NMI) and Accuracy metrics are used. Accuracy is calculated as follows:

$$Accuracy = \frac{\sum_{i=1}^{n} \delta(c_i = map(p_i))}{n} \tag{8}$$

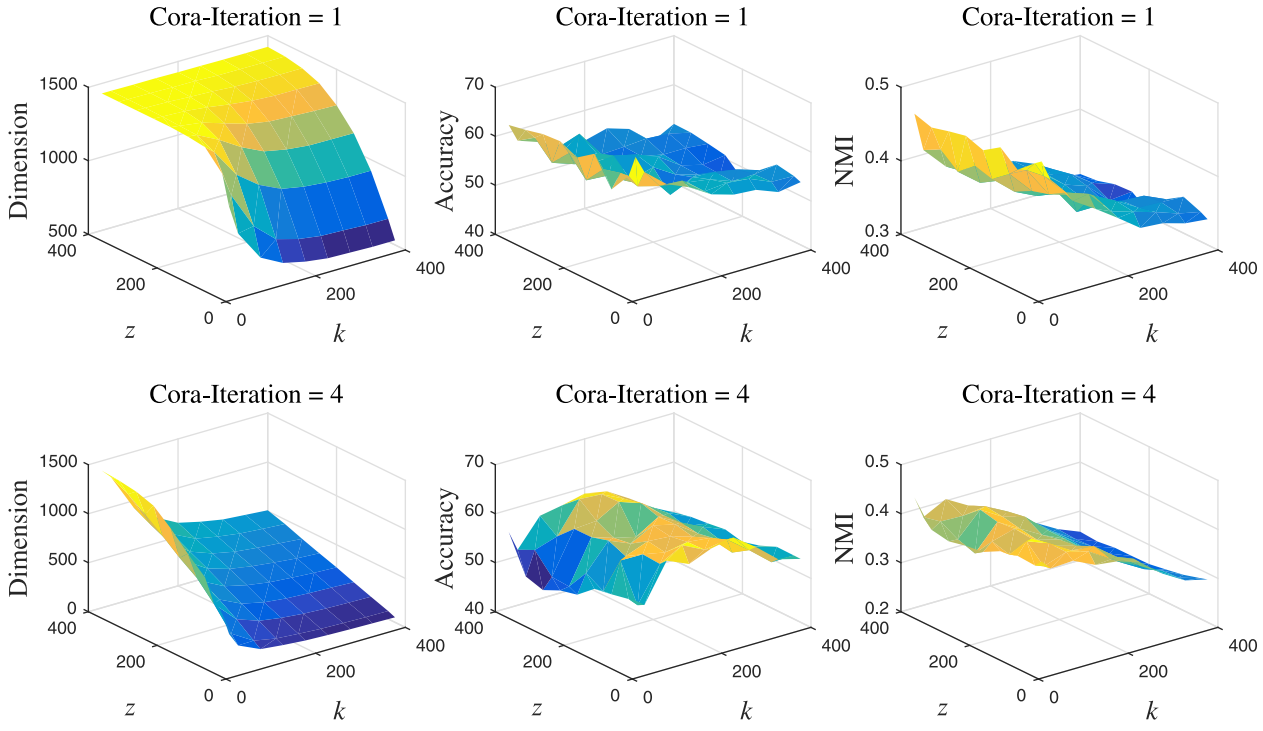

Fig. 6. Clustering performance of NetBFC on Cora with different values of k and z.

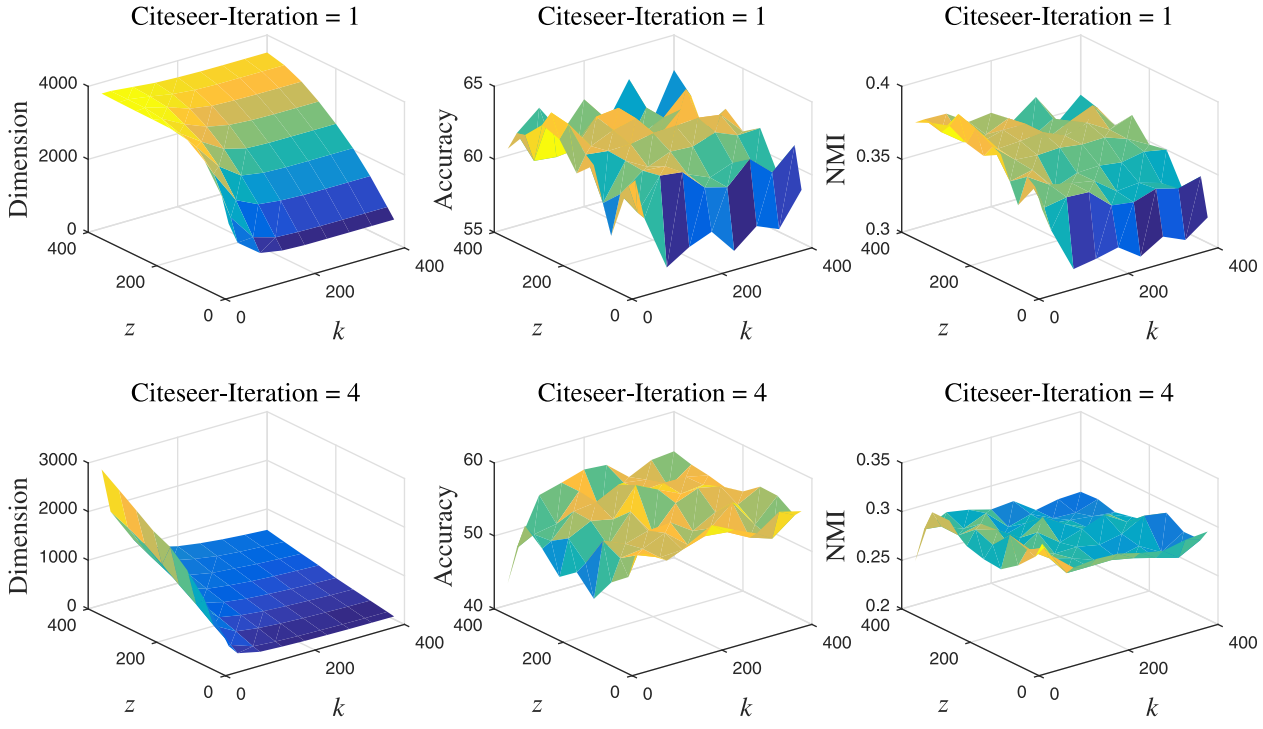

Fig. 7. Clustering performance of NetBFC on Citeseer with different values of k and z.

where p_i is the generated label for instance i, and c_i is its ground truth label. The $map(p_i)$ is the permutation mapping function that maps p_i to a cluster label using Kuhn-Munkres algorithm (Frank, 2005).

Let C be the set of cluster from the ground truth, and C' predicted by our method. NMI is defined as follows:

$$NMI(C, C') = \frac{MI(C, C')}{max(H(C), H(C'))} \tag{9}$$

where $H(C)$ and $H(C')$ are the entropies of C and C'. $MI(C, C')$ is their mutual information.

5.3. Baseline methods and experimental setup

Our method is compared to three unsupervised feature selection methods LS (He et al., 2006), UDFS (Yang et al., 2011) and NETFS (Li et al., 2016), one classical feature extraction method NMF (Berry, Browne, Langville, Pauca, & Plemmons, 2007), two

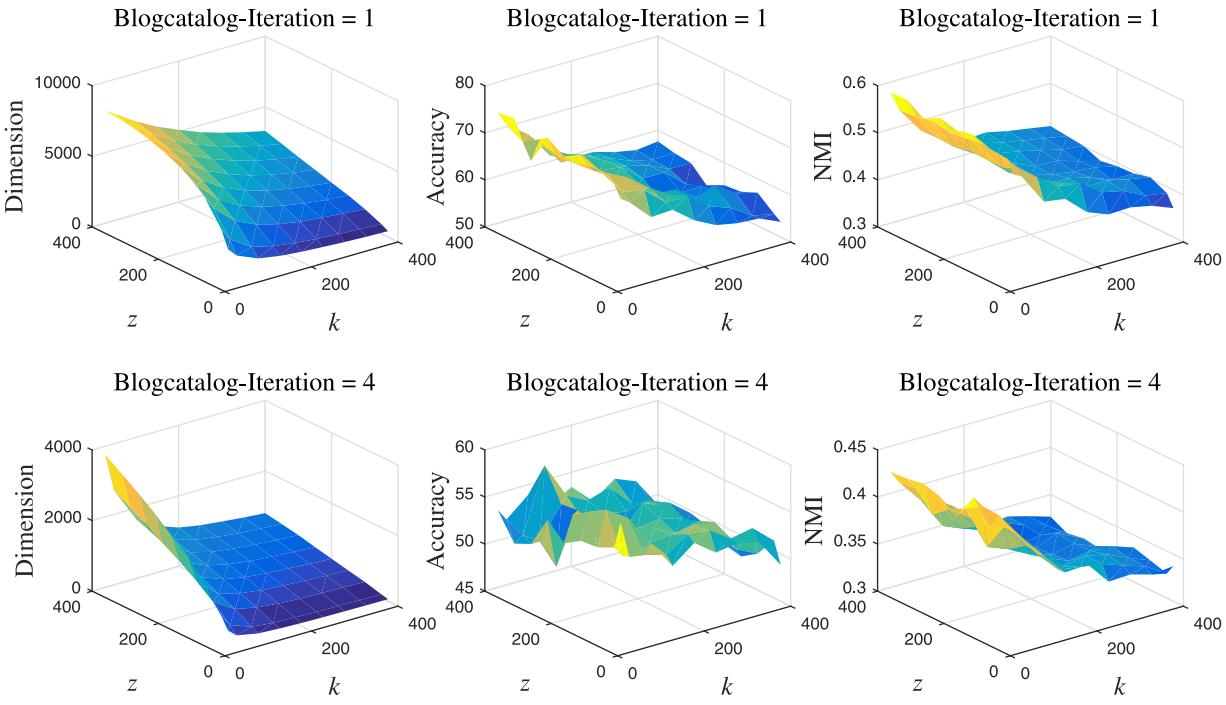

Fig. 8. Clustering performance of NetBFC on BlogCatalog with different values of k and z.

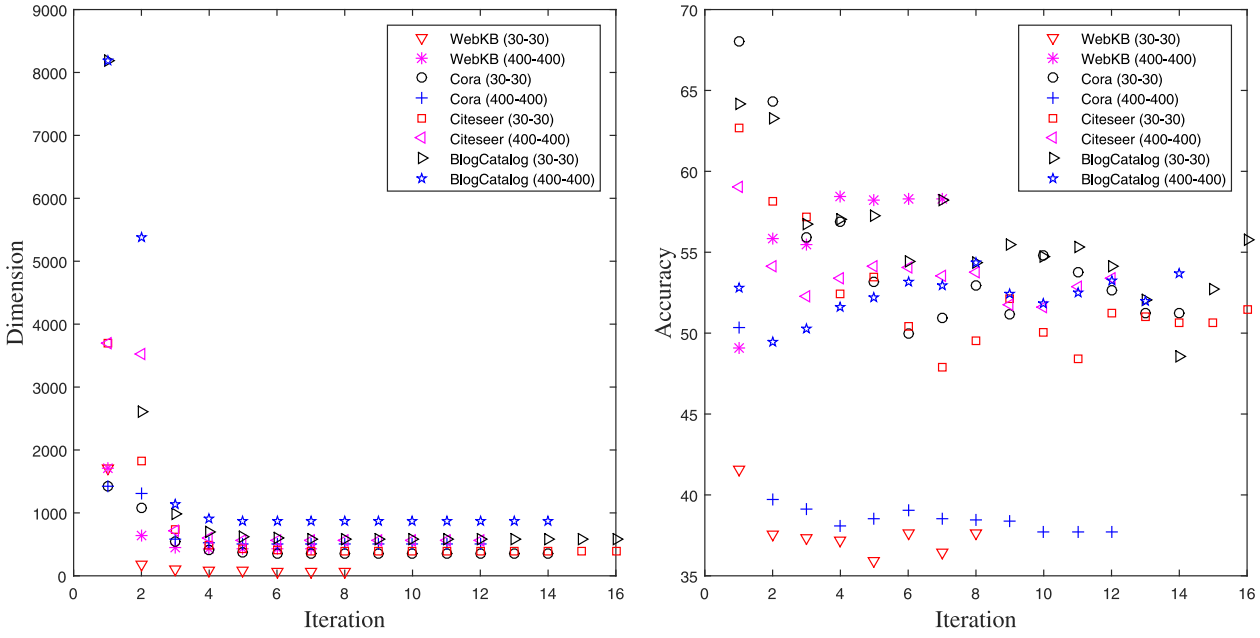

Fig. 9. The dimension and accuracy variations of NetBFC with different values of k and z.

network embedding methods LANE$_{w/o}$ (Huang et al., 2017b) and AANE (Huang et al., 2017a), and one multi-view spectral clustering method SPECMINDIS (De Sa, 2005). Brief descriptions of these methods are provided in Section 2.

We use the parameters as described in their respective papers for the baseline methods. The Cosine method was used for similarity metric for LS. We varied the parameters of NetFS, LANE and AANE methods from 0.01 to 100 with an increment step of 10. The number of iteration is set to 5 for LANE and AANE. LS, UDFS, NMF and SPECMINDIS are run with their default parameters.

For our method, the value of k for obtaining top-k-proximity neighbors and the value of z for selecting z-common attributes were varied as [30, 50, 100, 150, 200, 250, 300, 350, 400]. We varied the parameters α and β from 0 to 1 with an increment step of 0.1. The other parameter γ was determined using search strategy. We tested γ among the range of [0.001, 0.01, 0.1, 1]. Thereafter, we empirically set γ as 0.1. The number of iterations was set to 6. K-means (provided in MATLAB) were used as the clustering method. We repeated the K-means algorithm for 10 times since K-means

algorithm is affected by the initialization. We reported the average clustering performance.

5.4. Performance evaluation

In the experiments, we varied the number of selected features (dimensions) as [100, 200, 400, 600]. The method was iterated until the desired level of reduction in the dimensionality of the constructed attribute matrix was achieved. Average clustering performances are shown in Tables 2 and 3. A higher value for NMI and accuracy indicates improved performance. Compared to all baseline methods, we observed that the proposed NetBFC method consistently and significantly achieves higher performance for all data sets. For the WebKB data set, the UDFS method provided higher accuracy than our method when the dimension was 600. However, our method provided better NMI result for the WebKB data set. Considering the results in Tables 2 and 3, NetBFC tends to yield higher accuracy and NMI values with a small number of attributes when the appropriate parameters are selected. With only 100 attributes, NetBFC obtains 40.66%, 93.21%, 47.36%, and 199.45% improvements in accuracy for WebKB, Cora, Citeseer, and BlogCatalog, respectively. These results also reveal that NetBFC provides notable reduction in the dimension of networked data without a considerable performance decrease.

We analyze the effectiveness of all baseline methods with and without our method. NetBFC is evaluated as a pre-processing algorithm. We pre-processed each of the data sets by applying NetBFC, and then the constructed attribute matrices for all data sets were obtained. Thereafter, we ran all baseline methods on the constructed attribute matrices. We used P-WebKB, P-Cora, P-Citeseer and P-BlogCatalog labels to denote the pre-processed WebKB, Cora, Citeseer and BlogCatalog data sets. We set $k = 100$, $z = 100$, $\gamma = 0.1$, and $iter = 1$. α and β were set to 1. The results presented in Figs. 2 and 3 demonstrate that LS, NMF, LANE, and AANE with NetBFC provided better accuracy and NMI results compared to using these methods alone. UDFS with NetBFC yields higher accuracy and NMI values for almost all data sets compared to UDFS, with the exception of the Cora data set. NETFS with NetBFC achieves a notable increase in accuracy and NMI values for the BlogCatalog data set compared to NETFS. SPECMINDIS with NetBFC obtains better results for the Cora and Citeseer data sets compared to SPECMINDIS. Although these baseline methods differ from each other, it was observed that a considerable increase in the accuracy and NMI values existed for nearly all baseline methods when these methods were combined with NetBFC.

5.5. Parameter analysis

We can balance the contributions of contextual and structural information by controlling α and β, respectively. Moreover, two other parameters are used: k (which adjusts the number of top-k proximity neighbors), and z (which refers to the number of attributes belonging to each node). We first explored the effects of α and β by setting k to 400, and z to 400. To investigate the impacts of these parameters, we fixed one parameter to 1 and varied the other parameter from 0 to 1 with an increment step of 0.1. Fig. 4 demonstrates that the parameter α for NetBFC must be greater than zero in order to achieve effective results. The effectiveness of the method is sensitive to the parameter β. The highest accuracy results for WebKB, Cora, Citeseer, and BlogCatalog were obtained with $\beta = 0$, $\beta = 1$, $\beta = 1$, and $\beta = 0.6$, respectively when α was equal to 1. Considering the degree of homophily for the data sets presented in the Table 1, we observed that the parameter β is helpful when the homophily of the networked data is high. The degree of homophily for the Cora and Citeseer data sets are considerably higher than that of the WebKB and BlogCatalog data sets. Therefore, the best accuracy values were obtained for the Cora and Citeseer data sets when $\alpha = 1$ and $\beta = 1$.

To analyze how k and z affect accuracy, NMI, and dimensionality, we first set α and β to the values where the best performance for each data set was obtained. Performance variations are depicted in Figs. 5–8 for the WebKB, Cora, Citeseer, and BlogCatalog data sets, respectively. We observed that higher k and lower z values provided a significant amount of reduction in dimension. The highest accuracy values obtained were 68.18% with dimension 228 at $iter = 5$ for WebKB, 67.71% with dimension 1086 at $iter = 1$ for Cora, 63.92% with dimension 3694 at $iter = 1$ for Citeseer, and 73.96% with dimension 7846 at $iter = 1$ for BlogCatalog.

Fig. 9 presents the dimension and accuracy variations for all data sets at different iterations. NetBFC was iterated until the method was converged. The method converged rapidly, in fewer than 14 iterations. However, the reduction in dimension for all data sets were fixed at certain values in approximately 4 iterations. In a similar manner, the accuracy values for each data sets were fixed at certain intervals. As evident in Fig. 9, the method was converged in 8 iterations for the WebKB data set when k and z are set to 30. The obtained dimension of WebKB was 76 when *iter* is equal to 8. As such, the results indicate that our method converges rapidly for all data sets.

6. Conclusion

In this work, we proposed an unsupervised binary feature construction method that generates contextually and structurally meaningful attributes for networked data by exploiting link and attribute information. We aimed to increase the similarity between network objects with a high probability of being in the same group. Experiments were conducted on four publicly available real-world data sets. For all data sets, we observed that the proposed method, NetBFC, achieves better performance than the baseline methods. NetBFC also tends to yield high accuracy and high NMI values with a small number of attributes. We focus our future work on three aspects. Firstly, we intend to explore different aggregation and selection strategies to increase the performance of our method. Secondly, we plan to investigate how the value of the parameters can be estimated in an unsupervised manner from the structure of network data. Lastly, we are interested in making our method workable for dynamic networks.

CRediT authorship contribution statement

Arzu Gorgulu Kakisim: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft, Data curation, Visualization. **Ibrahim Sogukpinar:** Writing - review & editing, Resources, Supervision, Project administration.

References

Abualigah, L. M., Khader, A. T., Al-Betar, M. A., & Alomari, O. A. (2017). Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering. *Expert Systems with Applications, 84*, 24–36.

Bennasar, M., Hicks, Y., & Setchi, R. (2015). Feature selection using joint mutual information maximisation. *Expert Systems with Applications, 42*(22), 8520–8532.

Berry, M. W., Browne, M., Langville, A. N., Pauca, V. P., & Plemmons, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis, 52*(1), 155–173.

Bharti, K. K., & Singh, P. K. (2015). Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering. *Expert Systems with Applications, 42*(6), 3105–3114.

Carrer-Neto, W., Hernández-Alcaraz, M. L., Valencia-García, R., & García-Sánchez, F. (2012). Social knowledge-based recommender system. Application to the movies domain. *Expert Systems with Applications, 39*(12), 10990–11000.

Cataltepe, Z., Sonmez, A., & Senliol, B. (2014). Feature enrichment and selection for transductive classification on networked data. *Pattern Recognition Letters, 37*, 41–53.

Chang, S., Han, W., Tang, J., Qi, G.-J., Aggarwal, C. C., & Huang, T. S. (2015). Heterogeneous network embedding via deep architectures. In *Proceedings of the 21th ACMSIGKDDinternational conference on knowledge discovery and data mining* (pp. 119–128). ACM.

De Sa, V. R. (2005). Spectral clustering with two views. In *ICML workshop on learning with multiple views* (pp. 20–27).

Diesner, J., Frantz, T. L., & Carley, K. M. (2005). Communication networks from the enron email corpus â;;it's always about the people. enron is no differentâ;;. *Computational & Mathematical Organization Theory, 11*(3), 201–228.

Domingos, P. (2005). Mining social networks for viral marketing. *IEEE Intelligent Systems, 20*(1), 80–82.

Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

Frank, A. (2005). On kuhn's hungarian methodâ;;a tribute from hungary. *Naval Research Logistics (NRL), 52*(1), 2–5.

Friedkin, N. E. (2006). *A structural theory of social influence*: 13. Cambridge University Press.

Gu, Q., & Han, J. (2011). Towards feature selection in network. In *Proceedings of the 20th ACM international conference on information and knowledge management* (pp. 1175–1184). ACM.

Gundecha, P., & Liu, H. (2012). Mining social media: A brief introduction. *Tutorials in Operations Research, 1*(4), 1–17.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*(Mar), 1157–1182.

He, R., Tan, T., Wang, L., & Zheng, W.-S. (2012). l 2, 1 regularized correntropy for robust feature selection. In *Computer vision and pattern recognition (CVPR), 2012 ieee conference on* (pp. 2504–2511). IEEE.

He, X., Cai, D., & Niyogi, P. (2006). Laplacian score for feature selection. In *Advances in neural information processing systems* (pp. 507–514).

Hoseini, E., & Mansoori, E. G. (2016). Selecting discriminative features in social media data: An unsupervised approach. *Neurocomputing, 205*, 463–471.

Hu, L., Gao, W., Zhao, K., Zhang, P., & Wang, F. (2018). Feature selection considering two types of feature relevancy and feature interdependency. *Expert Systems with Applications, 93*, 423–434.

Huang, X., Li, J., & Hu, X. (2017a). Accelerated attributed network embedding. In *Proceedings of the 2017 SIAM international conference on data mining* (pp. 633–641). SIAM.

Huang, X., Li, J., & Hu, X. (2017b). Label informed attributed network embedding. In *Proceedings of the tenth ACM international conference on web search and data mining* (pp. 731–739). ACM.

Kolaczyk, E. D. (2009). *Statistical analysis of network data: Methods and models*. Springer Science & Business Media.

Kumar, A., & Daumé, H. (2011). A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 393–400).

Le, T. M., & Lauw, H. W. (2014). Probabilistic latent document network embedding. In *Data mining (ICDM), 2014 ieee international conference on* (pp. 270–279). IEEE.

Li, J., Hu, X., Wu, L., & Liu, H. (2016). Robust unsupervised feature selection on networked data. In *Proceedings of the 2016 SIAM international conference on data mining* (pp. 387–395). SIAM.

Liao, L., He, X., Zhang, H., & Chua, T.-S. (2017). Attributed social network embedding. arXiv:1705.04969,.

Marsden, P. V., & Friedkin, N. E. (1993). Network studies of social influence. *Sociological Methods & Research, 22*(1), 127–151.

McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual Review of Sociology, 27*(1), 415–444.

Mitra, P., Murthy, C., & Pal, S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(3), 301–312.

Mostafa, M. M. (2013). More than words: Social networks; text mining for consumer brand sentiments. *Expert Systems with Applications, 40*(10), 4241–4251.

Nicholas, F. C., & Carswell, I. B. (2012). *Method and system for providing network based target advertising*. US Patent 8,131,585

Nie, F., Huang, H., Cai, X., & Ding, C. H. (2010). Efficient and robust feature selection via joint ;2, 1-norms minimization. In *Advances in neural information processing systems* (pp. 1813–1821).

Perlich, C., & Provost, F. (2006). Distribution-based aggregation for relational learning with identifier attributes. *Machine Learning, 62*(1–2), 65–105.

Qi, G.-J., Aggarwal, C., Tian, Q., Ji, H., & Huang, T. (2012). Exploring context and content links in social media: a latent space method. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(5), 850–862.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., & Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine, 29*(3), 93.

Stelzl, U., Worm, U., Lalowski, M., Haenig, C., Brembeck, F. H., Goehler, H., et al. (2005). A human protein-protein interaction network: A resource for annotating the proteome. *Cell, 122*(6), 957–968.

Tang, J., & Liu, H. (2012). Feature selection with linked data in social media. In *Proceedings of the 2012 siam international conference on data mining* (pp. 118–128). SIAM.

Tang, J., & Liu, H. (2014). An unsupervised feature selection framework for social media data. *IEEE Transactions on Knowledge and Data Engineering, 26*(12), 2914–2927.

Wei, X., Xie, S., & Yu, P. S. (2015). Efficient partial order preserving unsupervised feature selection on networks. In *Proceedings of the 2015 SIAM international conference on data mining* (pp. 82–90). SIAM.

Yang, S., Hou, C., Nie, F., & Wu, Y. (2012). Unsupervised maximum margin feature selection via l 2, 1-norm minimization. *Neural Computing and Applications, 21*(7), 1791–1799.

Yang, Y., Shen, H. T., Ma, Z., Huang, Z., & Zhou, X. (2011). l2, 1-norm regularized discriminative feature selection for unsupervised learning. In *IJCAI proceedings-international joint conference on artificial intelligence: 22* (p. 1589).

Zhao, B., Kwok, J., Wang, F., & Zhang, C. (2009). Unsupervised maximum margin feature selection with manifold regularization. In *Computer vision and pattern recognition, 2009. CVPR 2009. IEEE conference on* (pp. 888–895). IEEE.

Zhu, S., Yu, K., Chi, Y., & Gong, Y. (2007). Combining content and link for classification using matrix factorization. In *Proceedings of the 30th annual international ACMSIGIR conference on research and development in information retrieval* (pp. 487–494). ACM.