Pattern Recognition 100 (2020) 107126

Contents lists available at ScienceDirect

Pattern Recognition

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

Clustering social audiences in business information networks

Yu Zheng[a,1], Ruiqi Hu[b,1], Sai-fu Fung[c], Celina Yu[d], Guodong Long[b], Ting Guo[e], Shirui Pan[a,*]

[a] Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia
[b] Centre for Artificial Intelligence, University of Technology Sydney, NSW 2007, Australia
[c] Department of Applied Social Sciences, City University of Hong Kong, China
[d] Global Business College of Australia, Australia
[e] University of Technology Sydney, NSW 2007, Australia

ARTICLE INFO

Article history:
Received 11 January 2019
Revised 23 October 2019
Accepted 21 November 2019
Available online 28 November 2019

Keywords:
Machine learning
Clustering
Business information networks
Social networks

ABSTRACT

Business information networks involve diverse users and rich content and have emerged as important platforms for enabling business intelligence and business decision making. A key step in an organizations business intelligence process is to cluster users with similar interests into social audiences and discover the roles they play within a business network. In this article, we propose a novel machine-learning approach, called CBIN, that co-clusters business information networks to discover and understand these audiences. The CBIN framework is based on co-factorization. The audience clusters are discovered from a combination of network structures and rich contextual information, such as node interactions and node-content correlations. Since what defines an audience cluster is data-driven, plus they often overlap, predetermining the number of clusters is usually very difficult. Therefore, we have based CBIN on an overlapping clustering paradigm with a hold-out strategy to discover the optimal number of clusters given the underlying data. Experiments validate an outstanding performance by CBIN compared to other state-of-the-art algorithms on 13 real-world enterprise datasets.

1. Introduction

Due to advancements in web techniques and the rapid growth of social networks, many enterprises and organizations are seizing these opportunity to offer timely feedback and provide tailored services to their social media audiences. In this article, we define a business information network (BIN) as a social network where people share similar interests in business products or activities. There are two main kinds of BIN [1]: private BINs for example, networks designed for the employees of a company or members of an industry association; and public BINs, such as Second Life, Twitter, Facebook, or LinkedIn, where each entity has a user account provided by the platform. We have concentrated on public BINs in this paper since they are more customer facing.

By validating our framework on 13 real-world public BIN datasets, we attempt to: (1) prove that our work can be applied in real-world business applications and that it benefits the enterprise or organization; (2) explore the underlying knowledge of BINs and try to automatically provide meaningful explanations of each detected cluster.

BINs have many applications in real business world. For instance, a BIN is managed by an enterprise through their official account on a social media platform (e.g., Facebook or Twitter), where all its social audiences (i.e., customers, business partners, and followers) can post their views (textual content) and interact (links) with each another. BINs are also good tools for acquiring business intelligence. According to Barnetta [2], the majority of Fortune 500 companies like Toyota, IBM, DE, and Sears have significantly optimized their business workflow by virtue of BINs. Moreover, in the UK, the State of Social Enterprise Survey 2013 [3], reports that 32% of social enterprises increased their volume of business in 2012. BINs have also been successfully used for other practical applications, such as advertising [4], online recruitment [5], and marketing [6,7].

Fig. 1 provides an example of the business intelligence workflow behind Sony Pictures Twitter activity. The social audiences, their Twitter followers, and all their information, including their friendships, profiles, and post messages, are fed into an automatic clustering engine that segments the social audiences into groups. The social audiences in each group show similar interests or preferences for Sony Pictures products. For example, users that have discussed movies might form one audience, whiles users who have

* Corresponding author.
E-mail addresses: zhengyu511@gmail.com (Y. Zheng), ruiqi.hu@uts.edu.au (R. Hu), shirui.pan@monash.edu (S. Pan).
[1] Y. Zheng and R. Hu contributed equally to this work.

https://doi.org/10.1016/j.patcog.2019.107126
0031-3203/

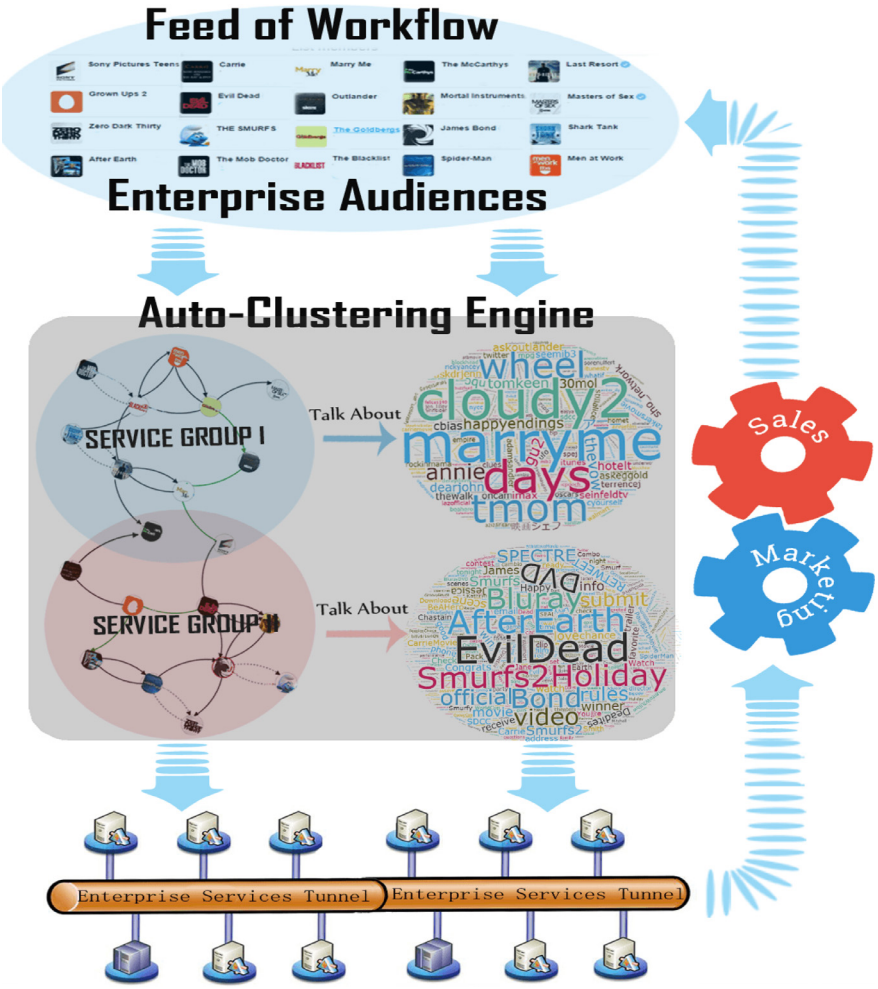

Fig. 1. An example of the business intelligence workflow behind the Twitter account of Sony Pictures Entertainment. The loop contains three main sections: a. automatic audiences, the Twitter followers, and their interests (derived from a clustering engine); b. an enterprise service tunnel; c. and sales and marketing analysis. Business partners and customers are divided into two groups according to the different services or products they are interested in with the people in each group sharing similar interests. For example, the members of one audience like the Evil Dead and After Earth two popular movies made by Sony Pictures, while the members another audience are mostly interested in its TV shows like Marry Me and Wheel. Note that customers may be members of two audiences simultaneously, and audiences may have overlapping interests. The different services or products of interest to each audience are delivered via an enterprise service tunnel, and the feedback data received through this tunnel are analyzed from a sales and marketing perspective to provide better services to the audience.

tweeted about TV shows could form another. Each group is then referred to the appropriate enterprise service tunnel based on their common preferences. Feedback from the enterprise service tunnel is delivered to the sales and marketing department, which is analyzed to further enhance service and product offerings to each audience. Obviously, the automatic clustering engine is the key to this entire business intelligence workflow. More specifically, accurately identifying the social audience clusters (SACs), and particularly their shared interests, makes a big difference in an enterprises ability to improve their offerings through tailored products and services to targeted users. Additionally, the ability to tailor products and services to specific audiences can dramatically reduce the cost of sales.

In this article, we focus on using a machine-learning approach to automatically discover SACs. A SAC is defined as a group of users who share similar interests or business roles within an enterprise. Intuitively, finding a SAC seems like it should be easy to accomplish with almost any network topology-based clustering algorithm, such as AgmFit [8] or BigClam [9] (both are available on SNAP[2]). However, these algorithms only consider a networks topology when clustering and so cannot provide much insight into the role a SAC plays in the network. Further, these approaches assume that the users (nodes) are densely connected, which is not usually the case with BINs. The poor performance of these clustering approaches on BIN datasets has proven this point.

More insight could be gained into the groups role with an algorithm based on relational topic discovery, like Balasubramanyan and Cohen [10] or a nonnegative matrix factorization method such as the one outlined in Cai et al. [11]. However, these approaches assume that node contents and networks are highly consistent and that the network (graph) follows manifold assumptions [11,12], i.e., the data reside smoothly over a low-dimensional space so the manifold regularization results in good clusters. Unfortunately, this is not true for BINs where node and network consistency is relatively low.

Beyond these disadvantages, even if existing tools could be used to find these audiences, there are still more limitations to overcome. Most tools would cluster each user into a single audience. Consequently, users could not hold two or more roles in the network, and this limitation does not fit many business models. Also, analysts would have to specify a threshold value for the number of clusters to find. But, in reality, users are often part of many au-

[2] http://snap.stanford.edu/.

diences, and the number of audiences is hard to define because they are data- and/or purpose-driven, and tend to have strong seasonal and event-centric characteristics. Hence, the clustering process needs to accommodate overlaps, and the optimal number of groups needs to be determined automatically.

In this article, we propose CBIN, a novel method for discovering SACs within BINs. CBIN is based on nonnegative matrix factorization (NMF), where network information and node content are integrated to produce clustering results via a consensus principle. CBIN naturally provides better insights into user interests whether customer or business partner as it simultaneously clusters data points and features from social media. We further propose a heuristic approach to set the threshold for overlapping clusters. The optimal number of groups is automatically determined by minimizing the reconstructed error with a hold-out method.

Our main contributions are summarized below:

1. We propose a novel algorithm for discovering SACs in BINs, which advances existing works that focus on generic information networks, like personal or academic network analysis.
2. We propose an effective overlapping co-clustering algorithm that simultaneously clusters social audiences and features into an automatically-determined number of optimal groups. Considering both audiences and features provides a deeper comprehension of the functional roles of business customers.
3. We conducted experiments on 13 real-world enterprise datasets. The results indicate that CBIN has outstanding performance compared to the current state-of-the-art approaches.

2. Related work

Business information networks (BINs) are important in a wide range of applications: as social media support systems, in marketing [13] and advertising [14], in customer relationship management [15], etc. Moreover, understanding the social audiences within BINs is a crucial aspect of business intelligence and decision making. In an attempt to find sought-after audiences, Lo et al. [16] proposed a method for ranking social audiences on Twitter according to their value. In this article, we propose to cluster social audiences in BINs with some understanding of their functional relevance to the company. From a technical perspective, discovering SACs in BINs is closely related to clustering problems in a network setting.

Some algorithms only leverage textual information during clustering for example, Dam and Veldens [17] MCA K-means approach to clustering Facebook users using collected profile information. Alternatively, community detection approaches, such as AgmFit [8], BigClam [9], GDPSO [18] and Diffusion [19], only rely on network structures to generate clusters. However, both these types of approaches only consider a single slice of information, which can lead to suboptimal results.

Co-clustering algorithms consider both textual information and network topologies. For instance, Gu and Zuo [12] and Wang et al. [20] both use NMF to factorize a user-feature matrix and then use the network structure to regularise the objective function. Relational topic model approaches, such as [10,21], use the network structure and the text simultaneously but, as mentioned above, there are consistency assumptions about the nodes and the network structure that may not be true in real-world BINs. NMF-based methods [11,22] assume that the data resides in a manifold, but network data is known to follow power law distributions. The joint matrix factorization (JMF) model [23] was specifically designed for pattern recognition and data integration. There are also emerging network embedding-based methods [24,25] or graph neural network-based methods [26], which learn the compact representation for each node so that clustering new representations can be done easily.

Algorithms that can create overlapping clusters of network data are a relatively recent advancement [27,28]. Leskovec and McAuley [27] studied the problem of discovering social groups in a user's personal network on social media platforms. By calculating the similarity between the users' profiles, the method proposed in [27] is capable of assigning one user to multiple clusters. Yang et al. [28] proposed the CENSA algorithm to model the interactions between the network structure and the node attributes. Although both meet the needs of overlapping clustering tasks, and they can specify which attributes are useful for forming a community, they do not group features into clusters to provide better interpretations of the topics that users are interested in.

Compared to existing studies, our research advances this work from generic information network analysis into BIN mining. Our framework has the following attractive properties: (1) it is specifically designed for BINs; (2) it enables increased freedom to address the inconsistencies in network structures and node content; and (3) it provides interpretable meanings for the SACs.

3. Problem definition

A BIN is defined as $G = \{V, E\}$. $V = \bigcup\{v_i\}_{i=1,\dots,n}$ denotes the set of social audiences in a BIN, and, $e_{i,j} =< v_i, v_j >\in E$ indicates an edge encoding the link between two audiences. A matrix $\mathbf{W}_s \in \mathbb{R}_+^{n\times n}$ is applied to further simplify the network information. If a link relationship exists between audiences v_i and v_j, $[\mathbf{W}_s]_{ij} = 1$; otherwise, $[\mathbf{W}_s]_{ij} = 0$. The user-feature vector $\mathbf{X}_{\cdot i} \in \mathbb{R}_+^m$ is associated with each audience v_i. Thus, the user-feature correlation of all audiences is embedded in a matrix $\mathbf{X} \in \mathbb{R}_+^{m\times n}$, where $\mathbf{X} = [\mathbf{X}_{\cdot 1}, \dots, \mathbf{X}_{\cdot n}]$ with each column in $\mathbf{X}$ indicating a node instance. Specifically, we leverage the TF-IDF to rank the m most significant features in the data set to construct a binary user-feature correlation matrix $\mathbf{X}$ where $\mathbf{X}_{ij} = 1$ if the j_{th} user contains the i_{th} feature, and $\mathbf{X}_{ij} = 0$ otherwise.

The **aim** of SAC discovering and understanding SACs is to automatically and simultaneously segment the audiences and features into an optimal number of clusters. Given that aim, the audiences $V = \bigcup\{v_i\}_{i=1,\cdots,n}$ in a BIN are clustered into k SACs $\mathcal{C} = [C_1, \dots, C_k]$. Following the similar operation of audience clustering, the features of the audiences V are also clustered into q clusters $\mathcal{Q} = [Q_1, \dots, Q_q]$ to understand the functions of SACs in a BIN. Every feature in a cluster is a special interest of the audiences in the BIN. To this end, a possible clustering result for an audience would be $\mathbf{G} = [\mathbf{G}_{1\cdot}; \dots; \mathbf{G}_{n\cdot}] \in \mathbb{R}_+^{n\times k}$, where $\mathbf{G}_{ij}$ corresponds to the degree of membership of the i-th audience for cluster C_j. Note that $\mathbf{G} \geq 0$, which means that all elements of $\mathbf{G}$ are nonnegative. Similarly, a possible clustering result for a feature would be the matrix $\mathbf{F} \in \mathbb{R}^{m\times q}$ where $\mathbf{F}_{ij}$ corresponds to the degree of membership of the ith feature for cluster Q_j. Again, $\mathbf{F} \geq 0$, indicating that all entries of $\mathbf{F}$ are nonnegative.

It is worth highlighting that SACs are groups of users with interactions over common interests. Unlike existing community detection algorithms [8], our proposed SAC discovery method, CBIN, has two unique features: (1) nodes can overlap multiple node groups, *i.e.*, $C_i \cap C_j \neq \emptyset$, which reflect situations in which a node plays different roles in different SACs; and (2) the optimal amount of SACs can be determined automatically.

4. CBIN algorithm

This section outlines the technical details of CBIN for SAC discovery, then extends CBIN into an overlapping clustering setting.

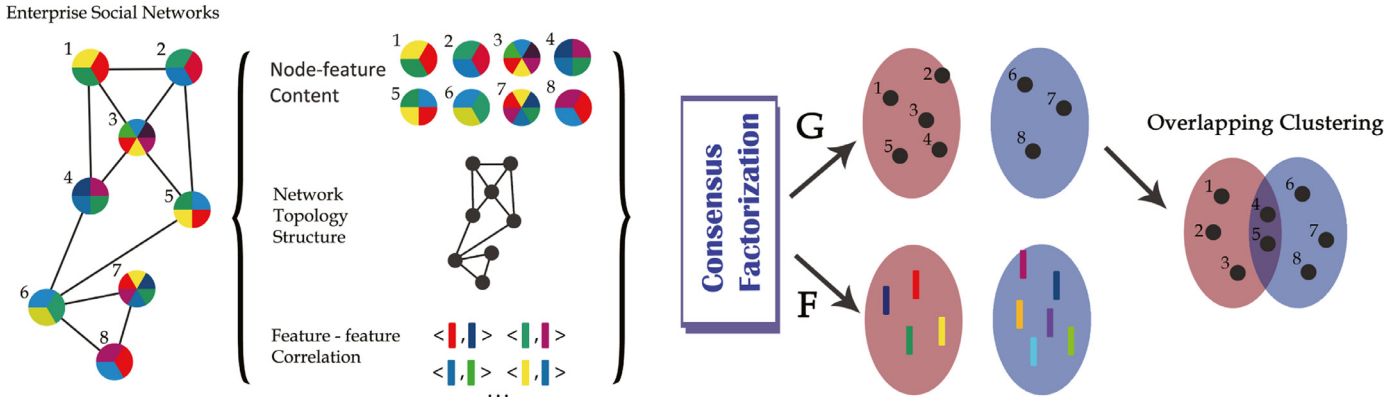

Fig. 2. Overall framework of the CBIN algorithm. Given a business social network, CBIN factorizes three channels of information via a consensus principle, i.e., the node-feature content matrix $\mathbf{X}$, the network topology structure $\mathbf{W}_s$, and the feature-feature correlations $\mathbf{W}_f$. After the factorization, CBIN clusters the nodes into ($\mathbf{G}$) and the features into ($\mathbf{F}$) and further extends $\mathbf{G}$ to account for overlapping clusters.

4.1. CBIN for SAC discovery

The data used to cluster the audiences is drawn from two main sources: the networks topological structures and the nodes contents. All audiences are represented as a user-feature matrix X, which comprises the actual feature values (contents) to stamp the user-feature of the BIN. To capture the links between audiences, pairwise connections inside the BIN are listed in an n × n matrix $\mathbf{W}_s$. More significantly, as the audiences in an SAC interact over similar interests in content and interactions, we explicitly explore the correlations between audience features ($\mathbf{W}_f$) need to be explicitly explored and used to inform the clustering process. Here, SACs can be discovered and clustered through a factorization method that factorizes the feature correlations matrix ($\mathbf{W}_f$), the user-feature ($\mathbf{X}$) and the network structure ($\mathbf{W}_s$), we use a factorization based methods to factorize $\mathbf{X}$, $\mathbf{W}_s$, and $\mathbf{W}_f$ separately, but then enforces consensus between the results. The overall framework for CBIN is shown in Fig. 2. The essential difference between CBIN and other existing NMF-based algorithms like [11,12,20] is that only the user-feature matrix is factorized, rather than comprehensively leveraging all the network information from different perspectives.

User-feature matrix factorization:

The user-feature matrix $\mathbf{X}$ provides a tabular mapping between users and features. Using NMF [29], $\mathbf{X}$ is factorized into two non-negative matrices $\mathbf{G}$ and $\mathbf{F}$ by minimizing the error function with a Frobenius norm:

$$\underset{\mathbf{F},\mathbf{G}}{\operatorname{argmin}}\ J_1 = ||\mathbf{X} - \mathbf{F}\mathbf{G}^\top||_F^2, \quad s.t.\ \mathbf{F} \geq 0, \mathbf{G} \geq 0, \tag{1}$$

where $\mathbf{F}\mathbf{G}^\top$ is the approximation of $\mathbf{X}$. The resulting clusters of users and features are naturally exposed in $\mathbf{G}$ and $\mathbf{F}$ [11,12,20,30]. For example, a node v_i can be assigned to the cluster $C_{j^\star}$, where $j^\star$ is determined by Eq. (2):

$$j^\star = \underset{j=1,\cdots,k}{\operatorname{argmax}} \mathbf{G}_{i,j} \tag{2}$$

In reality, because the two-factor NMF in Eq. (1) is restrictive, i.e., the number of clusters for q and k have to be equal, we have introduced an additional factor $\mathbf{S} \in \mathbb{R}_+^{q\times k}$ to compensate for the different scales of $\mathbf{X}$, $\mathbf{F}$ and $\mathbf{G}$. This leads to an extension of NMF, called NMTF [31,32]:

$$\underset{\mathbf{F},\mathbf{G}}{\operatorname{argmin}}\ J_2 = ||\mathbf{X} - \mathbf{F}\mathbf{S}\mathbf{G}^\top||_F^2, \quad s.t.\ \mathbf{F} \geq 0, \mathbf{G} \geq 0, \tag{3}$$

$\mathbf{S}$ provides increased degrees of freedom such that the low-rank matrix representation remains accurate, while the values of q and k can be different. The mapping information between node clusters and feature clusters is recorded in $\mathbf{S}$.

Network topology structure matrix factorization:

The adjacency matrix $\mathbf{W}_s$ holds pairwise user connections, which provides topological information to discover the similarity between the users for co-clustering. In practice, the matrix $\mathbf{W}_s$ is factorized into an $n \times k$ matrix $\mathbf{G}_s$ and its transposition $\mathbf{G}_s^\top$, where $\mathbf{G}_s \in \mathbb{R}_+^{n\times k}$ is an indicator matrix that shows the potential clustering results if only the topological information were to be leveraged:

$$\underset{\mathbf{G}_s}{\operatorname{argmin}}\ J_3 = ||\mathbf{W}_s - \mathbf{G}_s\mathbf{G}_s^\top||_F^2, \quad s.t.\ \mathbf{G}_s \geq 0, \tag{4}$$

It is worth mentioning that $\mathbf{G} \in \mathbb{R}^{n\times k}$ in J_2 and $\mathbf{G}_s \in \mathbb{R}^{n\times k}$ in J_3 each contains separated factorization results for the entire networked but from different perspectives. In this way, $\mathbf{W}_s$ and $\mathbf{X}$ are factorized while retaining the maximum freedom to find the optimal results. The consensus function later enforces consensus between these two sets of results to produce the optimal outcome.

Feature correlation matrix factorization:

To enhance the feature clustering performance, pairwise feature correlations are also captured using the matrix $\mathbf{W}_f \in \mathbb{R}_+^{m\times m}$. We assume that the features $\mathbf{X}_{j\cdot}$ and $\mathbf{X}_{i\cdot}$ will be assigned to the same feature cluster, when they are highly associated, say, for words that always co-occur. Thus, correlation measurement approaches, such as a neighbor-based method [12] or heat kernels [33], are reasonable choices for constructing $\mathbf{W}_f$. In our experiments, we simply applied a linear kernel $[\mathbf{W}_f]_{ij} = < \mathbf{X}_{i\cdot}, \mathbf{X}_{j\cdot} >$, where $\mathbf{X}_{i\cdot}$ indicates the embedding of the i_{th} column of features across all users, while $< \mathbf{X}_{i\cdot}, \mathbf{X}_{j\cdot} >$ measures the similarity between $\mathbf{X}_{i\cdot}$ and $\mathbf{X}_{j\cdot}$.

The factorization of the feature matrix $\mathbf{W}_f$ is similar to Eq. (4):

$$\underset{\mathbf{F}_f}{\operatorname{argmin}}\ J_4 = ||\mathbf{W}_f - \mathbf{F}_f\mathbf{F}_f^\top||_F^2, \quad s.t.\ \mathbf{F}_f \geq 0, \tag{5}$$

Consensus factorization:

Through the above factorizations, J_2, J_3 and J_4 are cluster schemes of the entire network based on different sources of data. To unify the results, we jointly formulate J_2, J_3 and J_4 into a single objective with a consensus function:

$$\begin{aligned} J_5 = {} & ||\mathbf{X} - \mathbf{F}\mathbf{S}\mathbf{G}^\top||_F^2 + \alpha||\mathbf{W}_s - \mathbf{G}_s\mathbf{G}_s^\top||_F^2 + \beta||\mathbf{W}_f - \mathbf{F}_f\mathbf{F}_f^\top||_F^2 \\ & + \rho\left(||\mathbf{G} - \mathbf{G}_s||_F^2 + ||\mathbf{F} - \mathbf{F}_f||_F^2\right), \\ & s.t.\ \mathbf{F} \geq 0, \mathbf{G} \geq 0, \mathbf{F}_f \geq 0, \mathbf{G}_s \geq 0, \end{aligned} \tag{6}$$

The aim of Eq. (6) is to factorize $\mathbf{W}_f$ (user-features), $\mathbf{X}$ (feature correlations), and $\mathbf{W}_s$ (network structures) separately, then enforce the consensus between each set of results. For example, $||\mathbf{G} - \mathbf{G}_s||_F^2$ minimizes the difference between $\mathbf{G}$ and $\mathbf{G}_s$, which represents the

potential clusters from the user-features matrix and a purer topological structure respectively. Similarly, $||\mathbf{F}-\mathbf{F}_f||_F^2$ enforces that $\mathbf{F}$ should be maximally consistent with $\mathbf{F}_f$. α and β are responsible for balancing each factorization. ρ is a consistency trade-off. Intuitively, a large ρ would lead to a $\mathbf{G}$ that is close to $\mathbf{G}_s$ and an $\mathbf{F}$ that is close to $\mathbf{F}_f$; however, a small ρ would mean $\mathbf{G}$ and $\mathbf{G}_s$ are independent of each other. Our approach provides increased degrees of freedom to exploit the inconsistencies between content and the topological structure, which a typical characteristic of BINs as discussed in the Introduction.

Algorithm optimization

The objective function Eq. (6) is processed with regard to $\mathbf{F}_f$, $\mathbf{F}$, $\mathbf{G}_s$, $\mathbf{G}$ and $\mathbf{S}$, which is not convex when simultaneously considering all variables. In this case, Eq. (6) is optimized with regard to one variable while fixing the others. This procedure is repeated until the function converges.

If $\mathbf{G}$ is optimized first while fixing the others, the Lagrange function for Eq. (6) is:

$$L = ||\mathbf{X}-\mathbf{FSG}^\top||_F^2 + \rho||\mathbf{G}-\mathbf{G}_s||_F^2 + \lambda_G\mathbf{G} \tag{7}$$

By conducting the partial derivatives with respect to $\mathbf{G}$ zero, we have

$$\frac{\partial L}{\partial \mathbf{G}} = -2\mathbf{X}^\top\mathbf{FS} + 2\mathbf{GS}^\top\mathbf{F}^\top\mathbf{FS} + 2\rho\mathbf{G} - 2\rho\mathbf{G}_s + \lambda_G = 0 \tag{8}$$

which leads to an update of $\mathbf{G}$ with following rule:

$$\mathbf{G} \leftarrow \mathbf{G} \odot \frac{\mathbf{X}^\top\mathbf{FS} + \rho\mathbf{G}_s}{\mathbf{GS}^\top\mathbf{F}^\top\mathbf{FS} + \rho\mathbf{G}}; \tag{9}$$

Here "$\odot$" means the Hadamar product (".*" in MatLab), *i.e.* $(\mathbf{A}\odot\mathbf{B})_{ij} = (\mathbf{A})_{ij}\cdot(\mathbf{B})_{ij}$. Similarly, we can update $\mathbf{G}_s$, $\mathbf{F}_f$, $\mathbf{F}$, and $\mathbf{S}$ are updated as follows:

$$\mathbf{F} \leftarrow \mathbf{F} \odot \frac{\mathbf{XGS}^\top + \rho\mathbf{F}_f}{\mathbf{FSG}^\top\mathbf{GS}^\top + \rho\mathbf{F}}; \tag{10}$$

$$\mathbf{G}_s \leftarrow \mathbf{G}_s \odot \frac{\rho\mathbf{G} + 2\alpha\mathbf{W}_s^\top\mathbf{G}_s}{2\alpha\mathbf{G}_s\mathbf{G}_s^\top\mathbf{G}_s + \rho\mathbf{G}_s}; \tag{11}$$

$$\mathbf{F}_f \leftarrow \mathbf{F}_f \odot \frac{\rho\mathbf{F} + 2\beta\mathbf{W}_s^\top\mathbf{F}_f}{2\beta\mathbf{F}_f\mathbf{F}_f^\top\mathbf{F}_f + \rho\mathbf{F}_f}; \tag{12}$$

$$\mathbf{S} \leftarrow \mathbf{S} \odot \frac{\mathbf{F}^\top\mathbf{XG}}{\mathbf{F}^\top\mathbf{FSG}^\top\mathbf{G}}, \tag{13}$$

These parameters will be iteratively updated until reaching the convergence to obtain the clustering result matrices:

$$\hat{\mathbf{F}} = \mathbf{F} + \mathbf{F}_f; \quad \hat{\mathbf{G}} = \mathbf{G} + \mathbf{G}_s \tag{14}$$

We assign a user v_i to a cluster $C_{j^\star}$ where $j^\star$ is defined as:

$$j^\star = \underset{j=1,\cdots,k}{\operatorname{argmax}} \hat{\mathbf{G}}_{i,j} \tag{15}$$

4.2. Automatic overlapping clusters

The formulation in Eq. (15) only represents partial progress because it can only deal with single hard cluster membership problems, i.e., where each node is assigned to a single cluster. In BINs, users often belong to different audiences and audiences often play different roles, i.e., nodes belong to more than one cluster and clusters overlap. Hence, this section discusses the process for determining how to the clusters overlap.

Overlapping clusters. The strategy in most current clustering algorithms is to assign a node to the cluster with the highest probability of membership. Our strategy is to condition a possibility threshold for each cluster and assign a node to a cluster if it exceeds that possibility threshold. Technically, the initial clustering results could be derived using Eqs. (6) and (15), which could then be used to fine-tune the clustering overlap results. However, using a threshold means the process can be automated. In specific terms, the scheme operates as follows: a threshold γ_j is assigned to each cluster C_j and, rather than assigning node v_i to a single cluster $C_{j^\star}$ via Eq. (15), v_i is assigned to any cluster C_j as long as $\hat{\mathbf{G}}_{i,j} \geq \gamma_j$.

Suppose the initialized groups obtained from Eq. (15) are $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$, where $C_j = \bigcup\{v_i\}$ is the set of nodes in each cluster. The membership value for each node $v_i \in C_j$, in $\hat{\mathbf{G}}$ is $\hat{\mathbf{G}}_{i,j}$. C_j will have a set of membership values $P_j = \bigcup_{v_i\in C_j}\{\hat{\mathbf{G}}_{i,j}\}$. Next, P_j is sorted to get its minimum value:

$$\gamma_j = \min P_j \tag{16}$$

where γ_j is the minimum threshold for assigning a node to cluster C_j. Thus the final clustering result of the j-th cluster is:

$$f(v_i) = C_j \quad : \quad \text{if} \quad \hat{\mathbf{G}}_{i,j} \geq \gamma_j, j = 1\cdots k \tag{17}$$

Thus, through Eq. (17), CBIN can automatically generate overlapping clusters.

This SAC discovery method is outlined in Algorithm 1. The inputs are the network structure matrix $\mathbf{W}_s$, the user-feature matrix $\mathbf{X}$, the number of feature clusters q, and the number of node clusters k as inputs. The algorithm initializes $n \times k$ matrix $\mathbf{G}$ and $m \times q$ matrix $\mathbf{F}$ using a k-means algorithm in Steps 2-3. In Steps 4-10, $\mathbf{G}_s$, $\mathbf{G}$, $\mathbf{F}_s$, $\mathbf{F}$ and $\mathbf{S}$ are iteratively updated until convergence or the maximum number of iterations S_{max} is reached. (In our experiments, S_{max} was 80.) The clustering results for the nodes from matrix $\hat{\mathbf{G}}$,and the clustering results for the features from matrix $\hat{\mathbf{F}}$ are determined in Step 11. In Steps 14-16, CBIN fine-tunes the clustering overlap results by first finding the threshold γ_j for each cluster group C_j in Steps 15-16, and then assigning the nodes with membership values not less than γ_j to cluster C_j.

Determining the optimal number of clusters. To automatically determine the optimal number of clusters k, we propose empirical testing with a hold-out strategy. A subset of $\mathbf{X}$ and $\mathbf{W}_s$ is held-out as the test set and the rest is used for training. The training subset for $\mathbf{X}$ and $\mathbf{W}_s$ is factorized and the reconstruction errors for the test entries are predicted, which leads to a revised objective function:

$$\begin{aligned} J_6 = {} & ||\mathbf{Y}\odot\mathbf{X} - \mathbf{FSG}^\top||_F^2 + \alpha||\mathbf{Y}\odot\mathbf{W}_s - \mathbf{G}_s\mathbf{G}_s^\top||_F^2 \\ & + \beta||\mathbf{W}_f - \mathbf{F}_f\mathbf{F}_f^\top||_F^2 \\ & + \rho\left(||\mathbf{G}-\mathbf{G}_s||_F^2 + ||\mathbf{F}-\mathbf{F}_f||_F^2\right), \\ & s.t.\ \mathbf{F}\geq 0, \mathbf{G}\geq 0, \mathbf{F}_f \geq 0, \mathbf{G}_s \geq 0, \end{aligned} \tag{18}$$

where $\mathbf{Y} \in \mathbb{R}_+^{n\times n}$ is binary. If the entry $\mathbf{Y}_{i,j} = 1$, it is used for training, otherwise it is used for testing.

Then the reconstruction error on the test entries ($\mathbf{Y}_{i,j} = 0$) is computed as follows:

$$ReconsErr = ||\neg\mathbf{Y}\odot\mathbf{X} - \mathbf{FSG}^\top||_F^2 + ||\neg\mathbf{Y}\odot\mathbf{W}_s - \mathbf{G}_s\mathbf{G}_s^\top||_F^2 \tag{19}$$

where $\mathbf{Y}$ is the negation of $\neg\mathbf{Y}$. Specifically, $\mathbf{Y}_{ij} = 0$ if $\neg\mathbf{Y}_{ij}$=1, otherwise 1. During these parameter tuning-style experiments, k is varied from k_{min} to k_{max} until the reconstruction error *ReconsErr* does not decrease. Hence, the number of clusters k is determined empirically.

Time complexity. In brief, the time complexity of Algorithm 1 is as follows. Initializing the k-means algorithm in Step 2 has linear time complexity, i.e., $O(nmkt)$ [34], where n is the number of m-dimensional vectors, k is the number of clusters and t is the number of iterations needed until convergence. Steps 5 to 9 employ multiplicative update rules, and the time complexity for each step is $O(nmr)$ [35,36], where $r = \max(k, q)$. Thus, the time complexity for the matrix factorization is #interation $\times$ $O(nmr)$. In Step 21, sorting costs $O(n\log n)$. Thus, the total time complexity is

$$O(nmkt) + \#\text{interation} \times O(nmr) + O(n\log n).$$

Algorithm 1 CBIN: Clustering Business Information Networks.

Require: user-feature matrix $\mathbf{X} = [\mathbf{x}_{\cdot 1}, \ldots, \mathbf{x}_{\cdot n}] \in \mathbb{R}_+^{d \times n}$,
Network structure matrix $\mathbf{W}_s \in \mathbb{R}_+^{n \times n}$;
k: number of node clusters;
q: number of feature clusters.

Constructed $\mathbf{W}_f \in \mathbb{R}_+^{m \times m}$, i.e., $[\mathbf{W}_f]_{ij} = < \mathbf{x}_{i\cdot}, \mathbf{x}_{j\cdot} >$
Initialize $\mathbf{G} \in \mathbb{R}_+^{n \times k}$ and $\mathbf{F} \in \mathbb{R}_+^{m \times q}$ using k-means on $\mathbf{X}$ and $\mathbf{X}^\top$, respectively;
Initialize $\mathbf{G}_s = \mathbf{G}$ and $\mathbf{F}_f = \mathbf{F}$;
repeat
  // Update potential audiences clustering matrix G with Eq. (9)
  $\mathbf{G} \leftarrow \mathbf{G} \odot \frac{\mathbf{X}^\top \mathbf{F}\mathbf{S} + \rho \mathbf{G}_s}{\mathbf{G}\mathbf{S}^\top \mathbf{F}^\top \mathbf{F}\mathbf{S} + \rho \mathbf{G}}$;
  // Update potential feature clustering matrix F with Eq. (10)
  $\mathbf{F} \leftarrow \mathbf{F} \odot \frac{\mathbf{X}\mathbf{G}\mathbf{S}^\top + \rho \mathbf{F}_f}{\mathbf{F}\mathbf{S}\mathbf{G}^\top \mathbf{G}\mathbf{S}^\top + \rho \mathbf{F}}$;
  // Update audience indicator matrix $\mathbf{G}_s$ with Eq. (11)
  $\mathbf{G}_s \leftarrow \mathbf{G}_s \odot \frac{\rho \mathbf{G} + 2\alpha \mathbf{W}_s^\top \mathbf{G}_s}{2\alpha \mathbf{G}_s \mathbf{G}_s^\top \mathbf{G}_s + \rho \mathbf{G}_s}$;
  // Update feature indicator matrix $\mathbf{F}_f$ with Eq. (12)
  $\mathbf{F}_f \leftarrow \mathbf{F}_f \odot \frac{\rho \mathbf{F} + 2\beta \mathbf{W}_s^\top \mathbf{F}_f}{2\beta \mathbf{F}_f \mathbf{F}_f^\top \mathbf{F}_f + \rho \mathbf{F}_f}$;
  // Update feature-audience mapping matrix **S** with Eq. (13)
  $\mathbf{S} \leftarrow \mathbf{S} \odot \frac{\mathbf{F}^\top \mathbf{X}\mathbf{G}}{\mathbf{F}^\top \mathbf{F}\mathbf{S}\mathbf{G}^\top \mathbf{G}}$;
until Converges;
// Final clustering results for audiences and features
$\hat{\mathbf{G}} = \mathbf{G} + \mathbf{G}_s$;~~~$\hat{\mathbf{F}} = \mathbf{F} + \mathbf{F}_f$;

// Overlapping Clustering
Get the initial clustering result $\{C_1, C_2, \ldots, C_k\}$, where $C_j = \bigcup\{v_i\}$ according to Eq. (15);
Get the threshold γ_j for cluster C_j according to Eq. (16);
Overlapping Clustering according to Eq. (17).

5. Experiments

We evaluated CBIN on a series of real-world business social networks. Our aim was to show that: (1) CBIN offers outstanding performance when clustering users (SAC discovery); (2) CBIN clusters features and provides a meaningful understanding of the topics in networks; and (3) the correlations between the feature clusters and the node clusters provides insights into the role of each SAC in the BIN.

5.1. Experimental setup

5.1.1. Business information networks datasets

To evaluate CBIN and conduct the experiments, we assembled 13 real-world datasets from Twitter. These datasets cover various industries, including sports associations, motor enterprises, political parties, and news agencies. Each enterprise has an official Twitter homepage, and their online followers are manually managed with *Twitter lists*. Each of the followers on a list shares some common interests or have a similar relationship to the enterprise. Each enterprise hires dedicated employees to manually categorize the users into the different lists. Refer to Fig. 1 for an example of a business intelligence workflow based on a Twitter list.

As Twitter users have limited profile information, we shaped each followers interests by collecting and processing their most recent 100 tweets (retrieved in March 2015). Specifically, we took the top 1000 most frequent words or phrases marked with a hashtag # (to indicate an event) or the @ sign (to mention a user) to denote all the interests in one dataset. Table 1 contains the details of the datasets: the screen name of each organization, the number of followers, the number of labeled followers, the number of edges (links) between followers, the number of tweets by a follower, the number of lists (clusters), and a brief description of each enterprise. To evaluate CBINs performance, we regarded the Twitter lists from each enterprise as the ground truth user labels. If two users belonged to the same list, we considered them as one cluster.

5.1.2. Baselines

To validate CBINs performance, we compared it to nine relevant algorithms. These algorithms included approaches that only leverage either textual information or topological information, as well as approaches that consider both. The algorithms that only consider either textual or topological information were:

- *BigClam* [9], which combines NMF with the block stochastic gradient descent to detect communities based on topological information.
- *AgmFit* [8] is a community detection algorithm based on community-affiliation models; and
- *K-means*, a classical and effective unsupervised approach. We only used textual information with k-means in these experiments.

The baseline algorithms that consider both textual and topological information include relational topic models [10], NMF based algorithms [11,12,20], and the state-of-the-art algorithms in [27,28]. The specific algorithms selected for comparison were:

- *Censa* [28], a statistics-based model that uses the interactions between node content and the network structure for more accurate community detection;
- *Circle* [27], an attributed graph clustering algorithm that handles overlapping hard-membership for graph clustering;
- *Block-LDA* [10], an LDA based relational topic model method that considers both textual and topological information for clustering tasks;
- *FNMTF* [20], an NMF-based co-clustering method, which targets large scale of datasets;
- *DRCC* [12], which extracts nodes and features from network information to construct a user graph and a feature graph for clustering; and
- *GNMF* [11], which adds an encoded k-NN graph as a regularization term to the objective function of the classical NMF method.

In order to be able to use NMF-based methods [11,20],the manifold graphs (k-NN graphs) needed to be replaced with network structures for regularization. However, in practice, the network structures may be not the same as the k-NN graphs. It is worth noting that although NMF-based methods and relational topic models [10] can cluster nodes and features simultaneously, they are not designed to overlap clusters. However, Circle [27] and Censa [28] can overlap clusters, but cannot provide any interpretation of the resulting clusters at the feature level. By comparison, CBIN not only simultaneously clusters nodes and features in a business social network, but it also enables clusters to be overlapped for SAC discovery.

Table 2 summarily compares the all ten conducted algorithms.

5.1.3. Evaluation metrics

Although CBIN is unsupervised, we evaluated its performance by comparing the predicted clusters $\mathcal{C} = \{C_1, \ldots, C_k\}$ with the manually-labeled ground truth $\overline{\mathcal{C}} = \{\overline{C_1}, \ldots, \overline{C_k}\}$. Ideally, the predicted clusters should be consistently aligned with the ground truth.

We applied a balanced error rate (BER) [27,37] to calculate the error between the predicted clusters $\mathcal{C}$ and the ground truth $\overline{\mathcal{C}}$. The

Table 1
Business social network datasets used in this article.

Companies	Total Nodes	Labeled	Edges	Tweets	List	Industry	Description
ABCNews	845,196	729	43,032	69,018	17	Media	News division of Australian Broadcasting Corpo.
NBA	17,387,126	137	2,036	12,402	7	Sport	National Basketball Association
TwitterAU	210,194	531	13,718	45,324	13	Internet Media	Twitter Inc. Australia Branch
SonyPictures	1,454,582	129	186	10,801	5	Film Entertainment	Sony Pictures Entertainment Corporation Australian
Labor	111,447	346	11,484	9,646	5	Political Party	Labour Party in Australia
WhiteHouse	7,377,128	161	3,925	13,503	5	Political Organization	White House
						Automobile	
MercedesBenz	1,434,283	142	1,174	11,846	7	Manufacturer	Mercedes-Benz Automobile Manufacturer Corpo
Techreview	452,766	155	1,220	14,024	7	Technical Media	MIT Technology Review
Cambridge_Uni	214,640	529	4,125	47,329	12	University	Cambridge University
The_Nationals	22,253	27	173	1,941	3	Political Party	The National Party in Australia
Greens	68,607	154	2,384	13,078	9	Political Party	The Greens Party in Australia
MGM_Studios	835,751	68	621	6,414	6	Film Entertainment	Disney Metro Goldwyn Mayer Hollywood Studio
BBCNews	5,019,860	624	16,974	61,533	11	Media	British Broadcasting Corporation

Table 2
Comparison of CBIN with other algorithms.

	Ours	K-Means	BigClam	AgmFit	Censa	FNMTF	GNMF	DRCC	Nips	BLOCK-LDA
Content	*	*			*	*	*	*	*	*
Structure	*		*	*	*	*	*	*	*	*
Overlapping	*		*	*	*					
Explain Clusters	*									
Auto-determine Clusters number	*		*	*	*					

calculation is defined as follow:

$$\mathrm{BER}\left(C,\overline{C}\right)=\frac{1}{2}\left(\frac{\left|C/\overline{C}\right|}{|C|}+\frac{\left|C^c/\overline{C^c}\right|}{|C^c|}\right) \tag{20}$$

BER assigns equal importance to false positives and false negatives, so that trivial or random predictions incur an error rate of 0.5 on average. Such a measure is preferable to, say, a 0/1 loss, which assigns an extremely low error rate to trivial predictions.

We also used F_1 scores as an evaluation metric (F-measure) [38]. F_1 scores consider both the precision and recall of the clustering result. Only results with a high precision and recall rate produce good F_1 scores.

5.1.4. Parameter study

For a fair comparison, following [11,20], we conducted experiments with varied parameter settings for each baseline and took the best settings for each. With CBIN, we varied α, β, and σ from 0.2 to 2 in steps of 0.2. For DRCC [12], GNMF [11], and FNMTF [20], we set $\lambda=\mu$ where λ was tuned by searching the grid {0.1, 1, 10, 100, 500, 1000} as described in their paper. For AgmFit, we set the parameter e (edge probability between the nodes that do not share any community) by searching the grid{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0} and chose the best value as the final result. For the Circle method, we set the regularization parameter $\lambda_\epsilon \in$ {0, 1, 10, 100} as described in their paper.

We conducted the experiments to let CBIN empirically determine the number of clusters k and compared its performance with the existing overlapping algorithms: BigClam [9], AgmFit [8], and Censa [28].

We also fixed the value of k in the range of 3, 5, 7, 9, 11 to compare a broad range of algorithms on clustering tasks. We tested each setting for every algorithm on each dataset 30 times and used the average of the values as the report score.

Unless otherwise specified, we set k_{min}=2, $k_{max}=10$ and Z=30 for CBIN to automatically determine the value of k.

5.2. Experimental results

This section presents the results of the experiments on a user clustering task. We assumed the number of clusters k was unknown and compared CBIN with the overlapping clustering algorithms first, and then specified the value of k, and compared the performance of all algorithms.

Table 3
BER scores for auto-detecting K methods.

Auto Ber_loss	Ours	BigClam	AgmFit	Censa
ABCNews	0.19	0.303	0.484	0.413
NBA	0.007	0.125	0.368	0.105
Twitte rAU	0.045	0.313	0.467	0.249
SonyPictures	0.357	0.376	0.461	0.414
AustralianLabor	0.132	0.229	0.332	0.23
WhiteHouse	0.037	0.281	0.447	0.315
MercedesBenz	0.046	0.227	0.403	0.295
Techreview	0.317	0.297	0.435	0.342
Cambridge_Uni	0.091	0.264	0.421	0.242
The_Nationals	0.102	0.133	0.212	0.228
Greens	0.222	0.295	0.44	0.327
MGM_Studios	0.126	0.269	0.261	0.263
BBCNews	0.116	0.233	0.138	0.138

5.2.1. Overlapping clustering

This section compares the results for the overlapping clustering methods: BigClam, AgmFit, and Censa, all of which are available on the SNAP platform[3]. Note that the Circle source code does not provide automatic clustering functionality. Therefore, even though it is technically an overlapping method, we have not included its results in this section. Instead, we have only reported Circles results with a given k in the next subsection.

Looking at Tables 3 and 4, we can see that CBIN dramatically outperformed the other algorithms. For example, CBIN achieved F_1 scores of 0.988 and 0.925 on the NBA and White House datasets, while Censa only achieved F_1 scores of 0.791 and 0.295.

Overall, CBIN outperformed the other algorithms in 12 of the 13 real-world datasets according to BER and surpassed all the baselines in terms of F_1 scores. One reason could be that algorithms like BigClam and AgmFit only leverage topological structures when clustering users, and ignore any contextual information

[3] http://snap.stanford.edu/.

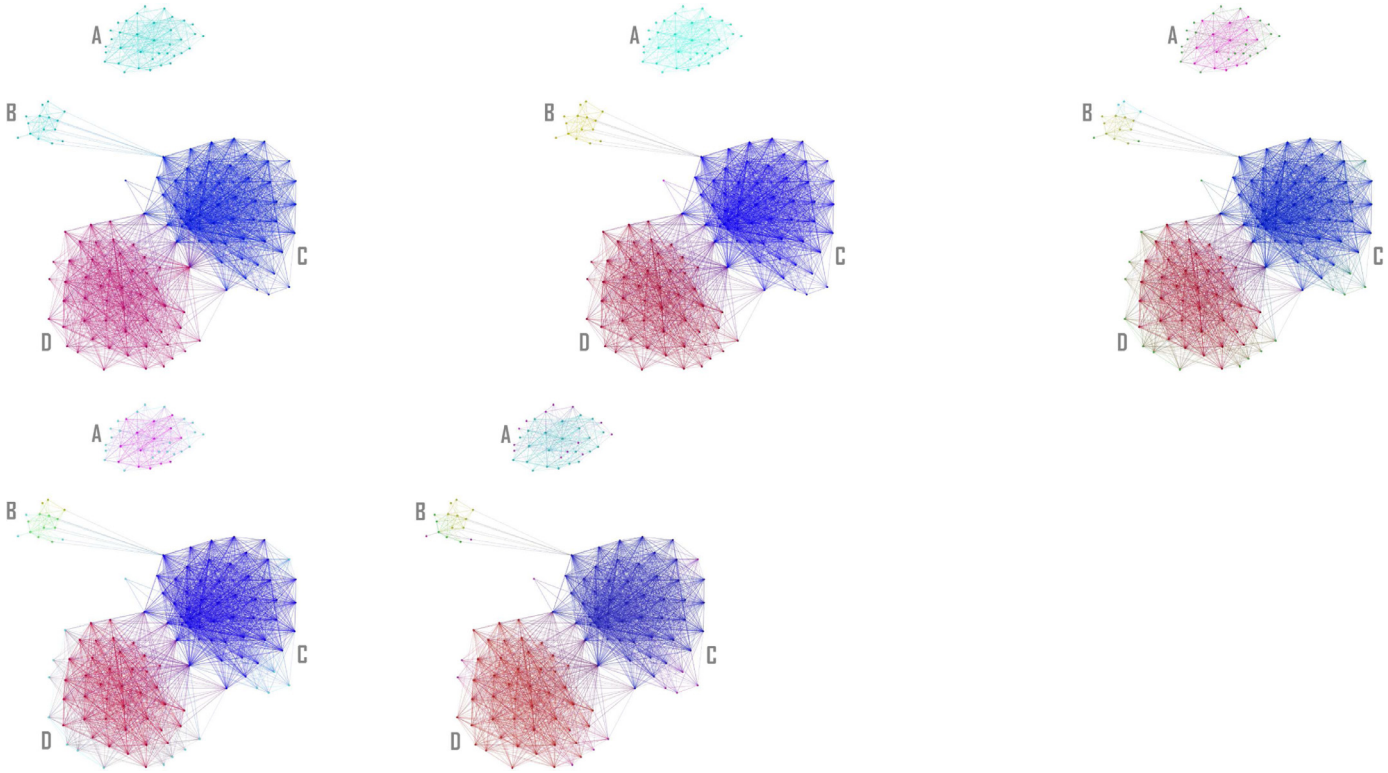

Fig. 3. White House data set result comparison. From left to right, the network graph comes from our algorithm, the ground truth, BigClam, Censa and AMGFit. Each color represents a cluster. The purer color of a cluster, the better performance.

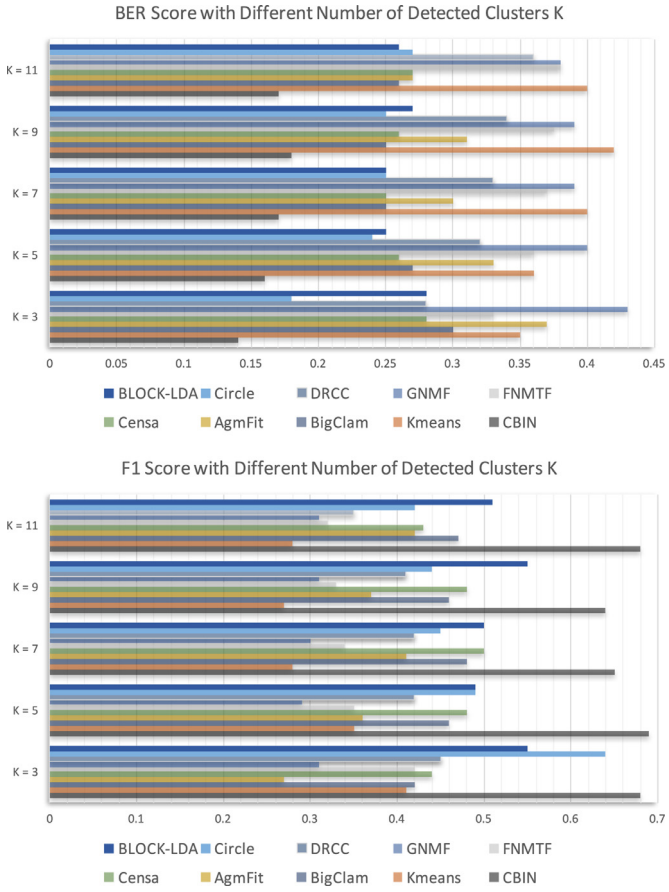

Fig. 4. Average performance on different numbers of detected k clusters. (A) BER score (smaller is better), (B) F_1 score (larger is better).

Table 4
F1 scores for auto-detecting K methods.

Auto F1 Score	Ours	BigClam	AgmFit	Censa
ABCNews	0.515	0.317	0.038	0.171
NBA	0.988	0.729	0.27	0.791
TwitterAU	0.829	0.376	0.072	0.44
SonyPictures	0.419	0.257	0.085	0.167
AustralianLabor	0.765	0.487	0.345	0.478
WhiteHouse	0.925	0.244	0.115	0.295
MercedesBenz	0.85	0.528	0.202	0.405
Techreview	0.445	0.371	0.138	0.255
Cambridge_Uni	0.692	0.453	0.159	0.521
The_Nationals	0.869	0.549	0.593	0.403
Greens	0.589	0.45	0.137	0.358
MGM_Studios	0.715	0.456	0.485	0.427
BBCNews	0.752	0.447	0.179	0.712

that might provide complementary knowledge. As a consequence, their results were suboptimal. Alternatively, algorithms like Censa, which consider both topological information and textual information, are based on the assumption that the communities generate both the topological structure and its textual features. However, Censa specifically assumes that the networks links are highly consistent with or high dependent on the users attributes, which is not always true in BINs. As previously discussed, the network links and user attributes in BINs tend to be the opposite âhighly inconsistent. As a result, Censas performance was only comparable to BigClam.

5.2.2. Case study

The clustering results in Fig. 3 show that the clusters predicted by CBIN were almost perfectly aligned with the ground truth. However, the results from the other baselines contained noticeable errors. For example, BigClam misclassified a large group of users in Cluster A.

Table 5
BER score with K = 5.

K = 5 Ber_loss	Ours	K-Means	BigClam	AgmFit	Censa	FNMTF	GNMF	DRCC	Nips	BLOCK-LDA
ABCNews	0.23	0.324	0.409	0.468	0.411	0.343	0.431	0.324	0.188	0.343
NBA	0.027	0.192	0.153	0.186	0.138	0.291	0.302	0.175	0.215	0.185
Twitte rAU	0.039	0.288	0.402	0.436	0.401	0.213	0.359	0.137	0.185	0.418
SonyPictures	0.290	0.343	0.344	0.449	0.338	0.413	0.469	0.307	0.415	0.113
AustralianLabor	0.126	0.348	0.27	0.197	0.279	0.409	0.364	0.36	0.214	0.079
WhiteHouse	0.037	0.381	0.36	0.268	0.363	0.392	0.462	0.413	0.21	0.125
MercedesBenz	0.115	0.407	0.245	0.281	0.259	0.398	0.352	0.343	0.234	0.193
Techreview	0.271	0.436	0.116	0.416	0.122	0.429	0.454	0.44	0.335	0.263
Cambridge_Uni	0.092	0.426	0.355	0.403	0.358	0.351	0.472	0.348	0.206	0.317
The_Nationals	0.036	0.419	0.094	0.176	0.119	0.241	0.22	0.322	0.247	0.308
Greens	0.333	0.343	0.261	0.287	0.256	0.365	0.364	0.333	0.183	0.301
MGM_Studios	0.083	0.349	0.239	0.203	0.202	0.394	0.468	0.204	0.187	0.265
BBCNews	0.113	0.394	0.221	0.371	0.223	0.398	0.458	0.386	0.187	0.303
Average	0.156	0.358	0.267	0.318	0.267	0.357	0.398	0.321	0.231	0.247

Table 6
F1 score with K = 5.

K = 5 F1 Score	Ours	K-Means	BigClam	AgmFit	Censa	FNMTF	GNMF	DRCC	Nips	BLOCK-LDA
ABCNews	0.432	0.215	0.181	0.075	0.177	0.294	0.216	0.222	0.46	0.686
NBA	0.922	0.641	0.698	0.633	0.728	0.467	0.393	0.61	0.458	0.368
Twitte rAU	0.834	0.444	0.203	0.133	0.207	0.44	0.28	0.687	0.484	0.833
SonyPictures	0.527	0.42	0.33	0.113	0.337	0.338	0.192	0.501	0.348	0.224
AustralianLabor	0.783	0.366	0.457	0.603	0.442	0.302	0.341	0.397	0.632	0.157
WhiteHouse	0.924	0.297	0.285	0.459	0.279	0.311	0.217	0.265	0.504	0.248
MercedesBenz	0.757	0.238	0.516	0.42	0.486	0.291	0.363	0.36	0.482	0.383
Techreview	0.462	0.224	0.757	0.176	0.746	0.248	0.242	0.218	0.367	0.522
Cambridge_Uni	0.713	0.229	0.295	0.197	0.289	0.244	0.155	0.237	0.366	0.633
The_Nationals	0.962	0.356	0.737	0.622	0.716	0.663	0.658	0.528	0.676	0.593
Greens	0.379	0.346	0.51	0.447	0.523	0.318	0.319	0.34	0.517	0.597
MGM_Studios	0.776	0.458	0.515	0.552	0.583	0.297	0.189	0.578	0.462	0.522
BBCNews	0.748	0.291	0.577	0.252	0.559	0.303	0.16	0.285	0.51	0.605
Average	0.681	0.348	0.466	0.396	0.467	0.347	0.286	0.402	0.482	0.49

Table 7
F1 score with K = 9.

K = 9 Ber_loss Ours	K-Means	BigClam	AgmFit	Censa	FNMTF	GNMF	DRCC	Nips	BLOCK-LDA	
ABCNews	0.274	0.374	0.353	0.37	0.344	0.345	0.411	0.359	0.223	0.365
NBA	0.043	0.322	0.129	0.16	0.144	0.342	0.352	0.287	0.231	0.248
TwitterAU	0.142	0.268	0.276	0.328	0.277	0.261	0.371	0.201	0.225	0.271
SonyPictures	0.339	0.473	0.387	0.442	0.391	0.411	0.36	0.366	0.357	0.259
AustralianLabor	0.118	0.36	0.256	0.291	0.227	0.375	0.32	0.336	0.234	0.3455
WhiteHouse	0.089	0.486	0.266	0.281	0.302	0.397	0.458	0.383	0.227	0.334
MercedesBenz	0.148	0.481	0.234	0.281	0.28	0.418	0.392	0.355	0.274	0.181
Techreview	0.313	0.475	0.3	0.306	0.319	0.421	0.468	0.405	0.304	0.21
Cambridge_Uni	0.054	0.447	0.16	0.371	0.131	0.397	0.481	0.384	0.231	0.235
The_Nationals	0.272	0.377	0.202	0.237	0.227	0.259	0.271	0.279	0.197	0.308
Greens	0.34	0.428	0.295	0.328	0.301	0.402	0.441	0.409	0.249	0.337
MGM_Studios	0.135	0.398	0.239	0.324	0.225	0.394	0.27	0.219	0.237	0.242
BBCNews	0.179	0.418	0.118	0.262	0.13	0.427	0.475	0.402	0.239	0.213
Average	0.19	0.408	0.247	0.306	0.254	0.373	0.39	0.337	0.248	0.273

5.2.3. Experimental results on node clustering with user-specified k

To compare CBIN with all baselines, including both the overlapping and non-overlapping methods, we specified the number of clusters (k) in a user clustering task. The results for K=5, 9, and 11 are shown in Tables 5, 6, 7, 8, 9 and 10. These results show that CBIN outperformed the other baselines in most cases.

Algorithms such as AgmFit, BigClam, and k-means only consider textual information about users or the topological information of the network, ignoring the other sources of information. As a result, these algorithms cannot fully leverage different sources of information available in the BIN and, therefore, produced suboptimal performance.

By contrast, our algorithm maximizes the degree of freedom to explore the distinct information encoded in the networked data, which leads to an increase in performance.

Interestingly, the algorithms that use both topological information and textual information, like Censa, Block-LDA, Circle, FNMTF, DRCC, and GNMF did not outperform algorithms like BigClam and k-means, which only leverage one source of information. These experiments confirm that there is a risk of confusing the model when integrating multiple sources of information. The GNMF, DRCC, and FNMTF methods are all NMF-based algorithms regularized by a manifold graph (k-NN graph). However, real-world network structures usually follow a power law distribution, which is different from a k-NN graph. In addition, in a k-NN graph, a node is supposed to be connected to its k nearest neighbors, which is calculated using feature values, so the links are highly consistent with the node contents. This is, unfortunately, not the case for BINs where topological information is typically highly inconsistent with the textual information. As a result, the algorithms that draw on multiple sources of information did not perform as well. By com-

Table 8
BER score with K = 11.

K = 9 F1 Score	Ours	K-Means	BigClam	AgmFit	Censa	FNMTF	GNMF	DRCC	Nips	BLOCK-LDA
ABCNews	0.366	0.192	0.257	0.226	0.272	0.174	0.231	0.159	0.319	0.729
NBA	0.941	0.437	0.709	0.652	0.689	0.419	0.399	0.501	0.432	0.493
TwitterAU	0.625	0.434	0.455	0.35	0.453	0.415	0.285	0.529	0.329	0.54
SonyPictures	0.451	0.172	0.24	0.116	0.224	0.308	0.441	0.414	0.426	0.513
Australian Labor	0.798	0.405	0.433	0.378	0.49	0.325	0.462	0.445	0.574	0.689
WhiteHouse	0.825	0.122	0.357	0.356	0.32	0.297	0.207	0.321	0.494	0.665
MercedesBenz	0.714	0.115	0.521	0.41	0.43	0.249	0.263	0.3750	0.396	0.359
Techreview	0.384	0.161	0.3480	0.321	0.303	0.261	0.193	0.253	0.371	0.418
Cambridge_Uni	0.81	0.187	0.684	0.261	0.741	0.205	0.097	0.274	0.315	0.469
The_Nationals	0.549	0.385	0.586	0.468	0.404	0.624	0.566	0.578	0.734	0.593
Greens	0.375	0.165	0.295	0.379	0.414	0.256	0.177	0.211	0.351	0.669
MGM_Studios	0.723	0.275	0.239	0.258	0.534	0.27	0.473	0.671	0.389	0.478
BBCNews	0.678	0.244	0.76	0.459	0.732	0.237	0.093	0.279	0.406	0.426
Average	0.634	0.254	0.453	0.402	0.462	0.311	0.299	0.385	0.426	0.542

Table 9
BER score with K = 11.

K = 11 Ber_loss	Ours	K-Means	BigClam	AgmFit	Censa	FNMTF	GNMF	DRCC	Nips	BLOCK-LDA
ABCNews	0.307	0.387	0.323	0.316	0.335	0.386	0.424	0.386	0.238	0.197
NBA	0.036	0.296	0.129	0.142	0.148	0.349	0.329	0.241	0.241	0.27
TwitterAU	0.063	0.229	0.299	0.202	0.324	0.289	0.349	0.26	0.249	0.222
SonyPictures	0.338	0.469	0.387	0.442	0.389	0.401	0.369	0.368	0.362	0.245
Australian Labor	0.106	0.357	0.238	0.261	0.23	0.398	0.354	0.389	0.272	0.385
WhiteHouse	0.092	0.479	0.276	0.228	0.315	0.393	0.407	0.399	0.212	0.328
MercedesBenz	0.118	0.483	0.231	0.288	0.295	0.428	0.389	0.378	0.276	0.193
Techreview	0.298	0.464	0.342	0.352	0.342	0.416	0.469	0.421	0.333	0.233
Cambridge_Uni	0.159	0.444	0.238	0.343	0.238	0.416	0.483	0.397	0.272	0.165
The_Nationals	0.113	0.356	0.122	0.108	0.228	0.264	0.259	0.28	0.222	0.212
Greens	0.314	0.403	0.314	0.343	0.327	0.418	0.433	0.401	0.258	0.36
MGM_Studios	0.157	0.39	0.218	0.324	0.263	0.345	0.272	0.251	0.24	0.311
BBCNews	0.244	0.432	0.162	0.25	0.138	0.433	0.479	0.402	0.262	0.191
Average	0.18	0.399	0.252	0.277	0.275	0.38	0.386	0.352	0.264	0.255

Table 10
F1 score with K = 11.

K = 11 F1 Score	Ours	K-Means	BigClam	AgmFit	Censa	FNMTF	GNMF	DRCC	Nips	BLOCK-LDA
ABCNews	0.338	0.132	0.281	0.294	0.318	0.168	0.198	0.164	0.271	0.393
NBA	0.95	0.461	0.709	0.698	0.633	0.393	0.433	0.565	0.418	0.5368
TwitterAU	0.808	0.539	0.406	0.601	0.359	0.363	0.275	0.44	0.278	0.443
SonyPictures	0.439	0.21	0.24	0.119	0.233	0.332	0.42	0.394	0.43	0.487
Australian Labor	0.86	0.395	0.458	0.396	0.478	0.306	0.408	0.346	0.494	0.767
WhiteHouse	0.863	0.149	0.31	0.43	0.295	0.291	0.313	0.297	0.502	0.652
MercedesBenz	0.787	0.103	0.52	0.367	0.405	0.233	0.271	0.318	0.383	0.383
Techreview	0.442	0.201	0.267	0.221	0.255	0.247	0.176	0.248	0.342	0.463
Cambridge_Uni	0.652	0.167	0.527	0.316	0.527	0.174	0.087	0.256	0.261	0.329
The_Nationals	0.85	0.403	0.619	0.661	0.403	0.624	0.624	0.558	0.697	0.407
Greens	0.413	0.249	0.389	0.335	0.358	0.222	0.169	0.246	0.341	0.714
MGM_Studios	0.704	0.257	0.515	0.298	0.427	0.318	0.463	0.463	0.413	0.612
BBCNews	0.554	0.205	0.658	0.495	0.712	0.223	0.079	0.28	0.363	0.381
Average	0.666	0.267	0.454	0.402	0.416	0.299	0.301	0.352	0.399	0.505

parison, our algorithm extracts the best ingredients from both the topological structure and the textual information through the proposed consensus factorization framework. Hence, the performance is outstanding compared to the other nine algorithms.

The average performance of each algorithm is illustrated in Fig. 4, which shows that our algorithm had low *BER* scores and higher F_1 scores for different K values.

5.3. Word cloud: understanding the feature groups

One noticeable property of CBIN is its capacity to group node features into clusters to provide a better understanding of a SACs interests in relation to the organization.

The clustering results are shown in Fig. 5 demonstrate that each word cloud is indeed very meaningful in practice. Table 11 lists the most frequent word in each word cloud along with its meaning.

Table 11
Keywords and explanation in word clouds.

TCNathan	A tropical cyclone which lashed North Queensland
qanda	Question and Answer, an ABC politic TV show
Auspol	Australian Policy/Politic
qld	Queensland, the second largest state in Australia
ausunions	Australian Unions

The results in Table 11 indicate that the feature clusters discovered by CBIN indeed provide an alternative understanding of the users interests. For instance, the keyword qld(meaning Queensland, the second-largest and third-most-populated state in Australia) is larger than other words. This means that *Queensland* is frequently discussed on Twitter because the members of the group

Fig. 5. Feature word clouds on the *AustralianLabor* dataset. Each word cloud represents a word cluster. The larger a word in a cloud, the more frequent it is discussed online.

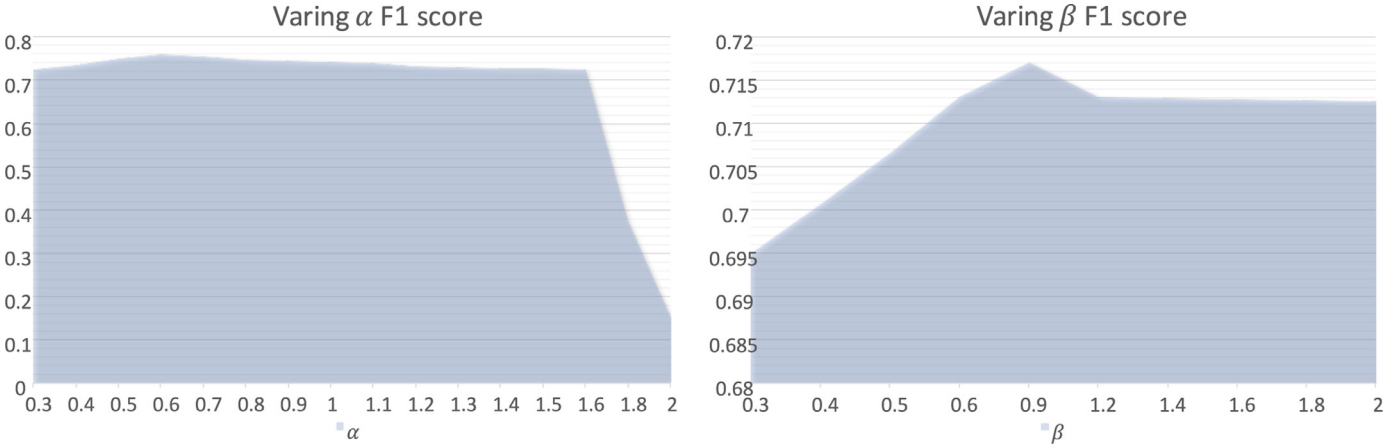

Fig. 6. Clustering performance *w.r.t.* different α, β values.

mostly come from Queensland or care about things that happen in Queensland.

5.4. Parameter sensitivity

We varied α, β, and σ from 0.3 to 2.0, to validate the performance of CBIN in terms of F_1 score, and the results are shown in Fig. 6. We have omitted the diagram for σ due to space limitations. As the value of α increased from 0.3 to 1.5, each dataset generated very similar results in terms of both BER and F_1 scores. However, performance plummeted when α was further increased. The results of the F_1 and BER scores in terms of β were similar to the results observed when changing the values for α.

6. Conclusion and future work

We argue that business intelligence networks (BINs), which are driven by relationships, policies, and business interests, are significant enablers of business intelligence and decision making for companies. However, the ability to accurately identify multiple audiences and to treat users as complex entities with overlapping interests is key to how intelligent a business intelligence system is. To this end, we devised a factorization-based co-clustering approach to identify and group audiences with similar interests in a BIN. The method, called CBIN, discovers social audience based on both network topology and user features, such as interests, posts, and profiles, to provide overlapping audience clusters with insights into the functional relevance those audiences have to the organization.

The main benefits of this approach, as compared to similar algorithms, are that CBIN is an advancement to current clustering research on generic social information networks, such as personal or academic network analysis. Further, CBIN simultaneously cofactorizes information from several perspectives: topological structure (audiences), textual features (user data), and the correlations between features. A consensus principle integrates these three different sets of results to effectively overcome inconsistencies in the BIN for optimal clustering performance. The end result is support for a business intelligence tool that can provide an in-depth functional comprehension of a companys customers and other social audiences.

We validated the effectiveness of CBIN through a series of experiments on 13 real-world enterprise datasets against nine other classical and state-of-the-art algorithms.

As with all studies, this research has some limitations that are opportunities for future research. To date, we have only evaluated our algorithm on business networks with ground truth labels. In the future, we plan to conduct evaluations with larger-scale labeled BINs. As enterprises and organizations prefer the simple but effective principle for decision-making, we also plan to further explore how to automatically determine the models clustering parameters with automatic machine learning techniques.

Declaration of Competing Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the Australian Government through the Australian Research Council (ARC) under grants 1) LP160100630 partnership with Australia Government Department of Health and 2) LP150100671 partnership with Australia Research Alliance for Children and Youth (ARACY) and Global Business College Australia (GBCA).

References

[1] E. Turban, N. Bolloju, T.-P. Liang, Enterprise social networking: opportunities, adoption, and risk mitigation, J. Organ. Comput. Electron. Commerce 21 (3) (2011) 202–220.
[2] A. Barnetta, Fortune 500 companies in Second Life–Activities, their success measurement and the satisfaction level of their projects, Master thesis ETH Zürich, 2009 Ph.D. thesis.
[3] S.E. UK, State of Social Enterprise Survey 2013, Social Enterprise UK, 2013.
[4] W.-S. Yang, J.-B. Dia, H.-C. Cheng, H.-T. Lin, Mining social networks for targeted advertising, in: Proceedings of the Thirty-Ninth Annual Hawaii International Conference on System Sciences, HICSS'06, volume 6, IEEE, 2006, p. 137a.
[5] E. Butow, K. Taylor, How to Succeed in Business Using LinkedIn: Making Connections and Capturing Opportunities on the World's# 1 Business Networking Site, AMACOM Division of American Management Association, 2008.
[6] G. Drury, Opinion piece: social media: should marketers engage and how can it be done effectively? J. Direct Data Digit. Market. Pract. 9 (3) (2008) 274–277.
[7] L. Wang, Z. Yu, F. Xiong, D. Yang, S. Pan, Z. Yan, Influence spread in geo-social networks: a multiobjective optimization perspective, IEEE Trans. Cybern. (2019) In press, doi:10.1109/TCYB.2019.2906078.
[8] J. Yang, J. Leskovec, Community-affiliation graph model for overlapping network community detection, in: Proceedings of the IEEE Twelfth International Conference on Data Mining (ICDM), IEEE, 2012, pp. 1170–1175.
[9] J. Yang, J. Leskovec, Overlapping community detection at scale: a nonnegative matrix factorization approach, in: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, ACM, 2013, pp. 587–596.
[10] R. Balasubramanyan, W.W. Cohen, Block-LDA: Jointly modeling entity-annotated text and entity-entity links, in: Proceedings of the SDM, volume 11, SIAM, 2011, pp. 450–461.
[11] D. Cai, X. He, X. Wu, J. Han, Non-negative matrix factorization on manifold, in: Proceedings of the Eighth IEEE International Conference on Data Mining, 2008. ICDM'08, IEEE, 2008, pp. 63–72.
[12] Q. Gu, J. Zhou, Co-clustering on manifolds, in: Proceedings of the Fifteenth ACM SIGKDD international conference on Knowledge Discovery and Data Mining, ACM, 2009, pp. 359–368.
[13] M. Trusov, R.E. Bucklin, K. Pauwels, Effects of word-of-mouth versus traditional marketing: findings from an internet social networking site, J. Market. 73 (5) (2009) 90–102.
[14] T.L. Tuten, Advertising 2.0: Social Media Marketing in a Web 2.0 World, Greenwood Publishing Group, 2008.
[15] C. Heller Baird, G. Parasnis, From social media to social customer relationship management, Strategy Leadersh. 39 (5) (2011) 30–37.
[16] S.L. Lo, R. Chiong, D. Cornforth, Ranking of high-value social audiences on twitter, Decis. Support Syst. 85 (2016) 34–48.
[17] J.-W. van Dam, M. van de Velden, Online profiling and clustering of facebook users, Decis. Support Syst. 70 (2015) 60–72.
[18] Q. Cai, M. Gong, L. Ma, S. Ruan, F. Yuan, L. Jiao, Greedy discrete particle swarm optimization for large-scale social network clustering, Inf. Sci. 316 (2015) 503–516.
[19] S. Yong, L. Minglong, Y. Hong, N. Lingfeng, Diffusion network embedding, Pattern Recognit. 88 (4) (2018) 518–531.
[20] H. Wang, F. Nie, H. Huang, F. Makedon, Fast nonnegative matrix tri-factorization for large-scale data co-clustering, in: Proceedings of the IJCAI Proceedings-International Joint Conference on Artificial Intelligence, volume 22, 2011, p. 1553.
[21] J. Gibert, E. Valveny, H. Bunke, Graph embedding in vector spaces by node attribute statistics, Pattern Recognit. 45 (9) (2012) 3072–3083.
[22] T. Guo, S. Pan, X. Zhu, C. Zhang, Cfond: consensus factorization for co-clustering networked data, IEEE Trans. Knowl. Data Eng. 31 (4) (2019) 706–719.
[23] L. Zhang, S. Zhang, A unified joint matrix factorization framework for data integration, (2017) arXiv:1707.08183.
[24] S. Pan, R. Hu, S.-f. Fung, G. Long, J. Jiang, C. Zhang, Learning graph embedding with adversarial training methods, (2019) arXiv:1901.01250.
[25] C. Wang, S. Pan, G. Long, X. Zhu, J. Jiang, Mgae: Marginalized graph autoencoder for graph clustering, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM, 2017, pp. 889–898.
[26] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks, (2019) arXiv:1901.00596.
[27] J. Leskovec, J.J. Mcauley, Learning to discover social circles in ego networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 539–547.
[28] J. Yang, J. McAuley, J. Leskovec, Community detection in networks with node attributes, in: Proceedings of the IEEE Thirteenth International Conference on Data Mining (ICDM), 2013, IEEE, 2013, pp. 1151–1156.
[29] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Proceedings of the Advances in Neural Information Processing Systems, 2001, pp. 556–562.
[30] W. Xu, X. Liu, Y. Gong, Document clustering based on non-negative matrix factorization, in: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2003, pp. 267–273.
[31] C. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix t-factorizations for clustering, in: Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 126–135.
[32] C. Ding, T. Li, M. Jordan, et al., Convex and semi-nonnegative matrix factorizations, IEEE Trans. Pattern Anal. Mach. Intell. 32 (1) (2010) 45–55.
[33] X. Niyogi, Locality preserving projections, in: Proceedings of the Neural Information Processing Systems, volume 16, MIT, 2004, p. 153.
[34] J.A. Hartigan, M.A. Wong, Algorithm AS 136: a K-means clustering algorithm, J. R. Stat. Soc. Ser. C (Appl. Stat.) 28 (1) (1979) 100–108.
[35] L.-X. Li, L. Wu, H.-S. Zhang, F.-X. Wu, A fast algorithm for nonnegative matrix factorization and its convergence, IEEE Trans. Neural Netw. Learn. Syst. 25 (10) (2014) 1855–1863.
[36] C.-J. Lin, On the convergence of multiplicative update algorithms for nonnegative matrix factorization, IEEE Trans. Neural Netw. 18 (6) (2007) 1589–1596.
[37] Y.-W. Chen, C.-J. Lin, Combining SVMs with various feature selection strategies, in: Feature Extraction, Springer, 2006, pp. 315–324.
[38] D.M. Powers, Evaluation: from precision, recall and f-measure to ROC, informedness, markedness and correlation, J. Mach. Learn. Technol. 2 (1) (2011) 37–63.

Yu Zheng received the masters degree in computer science from the Northwest A&F University, Yangling, Shaanxi, China, in 2011. She is now a research assistant at the Faculty of Information Technology, Monash University, Australia. Her research interests include data mining and machine learning.

Ruiqi Hu received the bachelor degree in software engineering from the Tianjin Polytechnic University (TJPU), Tianjin, China, in 2013. Since January 2016, he has been working toward the PhD degree in the Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney (UTS), Australia. His research interests include data mining and machine learning.

Sai-fu Fung received the Ph.D. degree from the University of Warwick, Coventry, U.K. He is currently with the Department of Applied Social Sciences, City University of Hong Kong, Hong Kong. His current research interests include comparative methods and democratisation and politics of East and Southeast Asia.

Celina P. Yu received the Ph.D. degree in finance from RMIT University, Melbourne, VIC, Australia. She is currently the Managing Director with the Global Business College of Australia (GBCA), Melbourne, VIC, Australia, and plays an indispensable role in the partnership with University of Canberra, Canberra, ACT, Australia, to deliver tertiary programs in Melbourne. Dr. Yu was a recipient of RMIT Universitys prestigious Best Doctoral Research Excellence Award.

Guodong Long received his Ph.D. degree in computer science from University of Technology, Sydney (UTS), Australia, in 2014. He is a Senior Lecturer in the Centre for Quantum Computation and Intelligent Systems (QCIS), Faculty of Engineering and Information Technology, University of Technology, Sydney (UTS). His research focuses on machine learning, data mining and cloud computing.

Ting Guo received the PhD degree in computer science from the University of Technology Sydney (UTS), Australia. He is a Lecturer with the Faculty of Engineering and Information Technology, UTS. His research interests mainly include data mining and machine learning. To date, he has published research papers in top-tier journals and conferences, including the IEEE Transactions on Knowledge and Data Engineering, IJCAI, AAAI and NIPS.

Shirui Pan received the Ph.D. degree in computer science from the University of Technology Sydney (UTS), Ultimo, NSW, Australia. He is currently a Lecturer with the Faculty of Information Technology, Monash University, Australia. Prior to that, he was a Lecturer with the School of Software, University of Technology Sydney. His research interests include data mining and machine learning. To date, Dr Pan has published over 70 research papers in top-tier journals and conferences, including the IEEE Transactions on Neural Networks and Learning Systems (TNNLS), IEEE Transactions on Knowledge and Data Engineering (TKDE), IEEE Transactions on Cybernetics (TCYB), ICDE, AAAI, IJCAI, and ICDM.