CFOND: Consensus Factorization for Co-Clustering Networked Data

Ting Guo, Shirui Pan, Xingquan Zhu, *Senior Member, IEEE,* and Chengqi Zhang, *Senior Member, IEEE*

Abstract—Networked data are common in domains where instances are characterized by both feature values and inter-dependency relationships. Finding cluster structures for networked instances and discovering representative features for each cluster represent a special co-clustering task usefully for many real-world applications, such as automatic categorization of scientific publications and finding representative key-words for each cluster. To date, although co-clustering has been commonly used for finding clusters for both instances and features, all existing methods are focused on instance-feature values, without leveraging valuable topology relationships between instances to help boost co-clustering performance. In this paper, we propose CFOND, a consensus factorization based framework for co-clustering networked data. We argue that feature values and linkages provide useful information from different perspectives, yet they are not always consistent and therefore need to be carefully aligned for best clustering results. In the paper, we advocate a consensus factorization principle, which simultaneously factorizes information from three aspects: network topology structures, instance-feature content relationships, and feature-feature correlations. The consensus factorization ensures that the final cluster structures are consistent across information from the three aspects with minimum errors. Experimental results on real-life networks validate the performance of our algorithm.

Index Terms—Networked data, Networks, Co-clustering, Topology, Nonnegative Matrix Factorization.

1 Introduction

Recent advancements in networking and communication systems has witnessed an increasing number of domains with networked data representation [1], [2], [3], [4], [5], where instances are characterized by using (1) feature values to represent content of the instances; and (2) linkages to denote dependency relationships between instances. For example, in a citation network, nodes can denote publications and linkages represent citation relationships between papers. One can use bag-of-words as features to represent the content of each paper. It would be very useful if a tool exists to automatically separate papers into different groups, and also help identify representative key-words for each group of papers. For many other types of networks, such as human disease networks [6], protein interaction networks [7], [8], terrorist networks [9], [10] *etc.*, finding clusters and identifying representative features for each cluster can be very helpful for discovering nodes sharing similar content and structure information, as well as uncovering most representative features for each node group, so users or domain experts can understand the essential difference of the node clusters by comparing their representative features.

Existing research has shown that simultaneously clustering instances and features can be beneficial for discovering patterns from data with tabular instance-feature representations. Research in this field, commonly referred to as co-clustering (or bi-clustering), can be roughly categorized into two groups: (1) iterative partitioning or merging, and (2) factorization. For iterative partitioning or merging, co-clustering starts by partitioning (or merging) instances into groups, and then validate the utility of the clusters with respect to the features' values and then iteratively partition (or merge) instances and features to form co-clusters [11], [12], [13]. Such a partitioning or merging process is typically heuristic driven without considering a global objective, so may be stuck to local maximum and result in suboptimal results. Alternatively, factorization based approaches intend to factorize an instance-feature tabular matrix into instance and feature groups, respectively [14], [15]. Such a co-clustering process is guided by a well-defined objective function with sound theoretical foundations. In addition, the factorization results also directly specify the likelihood/probability of each instance (or feature) belonging to a specific cluster, so one can easily form fuzzy clusters without exclusively assigning instances and features into clusters (*i.e.*, hard membership assignments). As a result, factorization based methods have recently been used in co-clustering. This includes efforts to improve the robustness of factorization based co-clustering methods *w.r.t.* noise and outliers [16].

In networked settings, linkages provide useful information. A commonly observed phenomenon [17] is that nodes close to each other in the network topology structure space tend to share common content information. For example, friends in the same cohort group are likely to share similar experiences. In a citation network where nodes denote papers and edges represent their citation relationships, a paper belonging to the data mining field will have its content directly related to data mining, and majority references cited in the paper should also belong to the data mining field as

- *T. Guo, S. Pan and C. Zhang are with the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, New South Wales, Australia. E-mail: tng.guoi-1@student.uts.edu.au, shirui.pan@uts.edu.au, and chengqi.zhang@uts.edu.au.*
- *X. Zhu is with the Department of Computer & Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA. E-mail: xqzhu@cse.fau.edu.*

Manuscript received December xx, 2014.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2846555, IEEE Transactions on Knowledge and Data Engineering

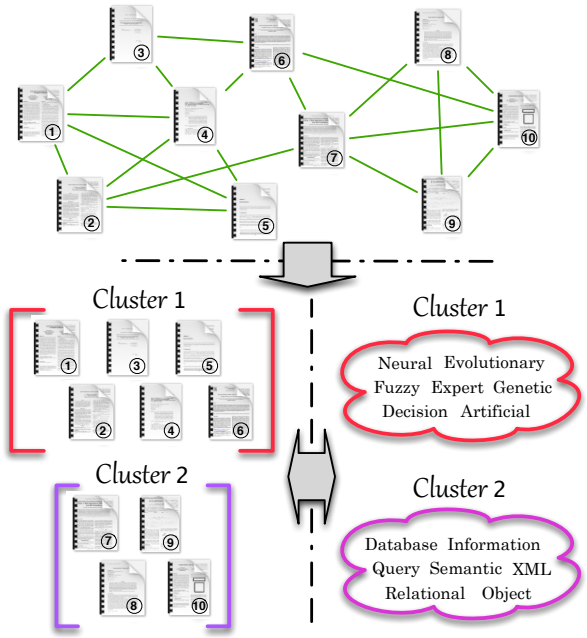


Fig. 1. An example of co-clustering on a citation network. The upper panel denotes a citation network, where each node represents a publication and green lines represent their citation relationships (each paper contains bag-of-words to represent the paper content). The lower panel shows co-clustering results on publications and keywords, respectively. Co-clustering on the citation network can help discover good node clusters (*i.e.* papers in different categories), and identify representative features (keywords) for each node group (*i.e.* keywords for a specific category of papers). Citation relationships are beneficial for inter-relationship finding and therefore help achieve more accurate clustering results.

well. As a result, linkage can help identify clusters which are incapable of being detected by using tabular instance-feature content matrix alone (*e.g.* Fig. (1)). Unfortunately, to date, all existing methods carry out co-clustering by exploring instance-feature relationships, without utilizing network topology structures to help find clusters from networked data. This observation raises a concern on what type of additional information linkages can provide for co-clustering, and how to leverage linkages to improve co-clustering results.

Indeed, using instance-instance graph relationships for co-clustering have been addressed in a number of studies, particularly in the context of manifold or k-NN graphs [18], [19]. For example, some works have proposed to build an instance-instance nearest neighbour graph by using k-NN relationships, and later enforce the k-NN graph in the objective function to discover cluster structures in a lower dimensional feature space (*i.e.*, manifold). In this context, the algorithm works on artificially created synthetic networked data. Intuitively, one can replace the synthetic instance-instance k-NN graph by using topology structure of the networks, and then apply existing co-clustering methods for networked data. However, as our experiments in Section 5 will soon demonstrate that existing k-NN graph based methods [18], [19] are ineffective in handling real-world networks. This is mainly because that instance-instance graphs created from the feature space are fundamentally different from real-world networks, in terms of the network characteristics and the consistency between network topology and node content. More specifically,

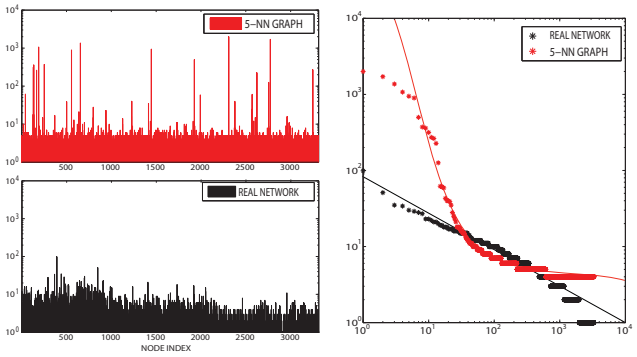


Fig. 2. The difference between real-world network topology structures *vs.* synthetic k-NN graph (affinity graph) on CiteSeer data set. Left figures show node degrees with logarithmic scale (y-axis) *w.r.t.* nodes indexed by the x-axis. Lower panel shows node degrees for genuine CiteSeer networks and the upper panel shows node degrees for the 5-NN graph. Results show that instances (nodes) with high degrees are different in real network *vs.* the 5-NN graph. Right figure shows that real network follows power-law distribution whereas the 5-NN graph follows geometric distribution, where the colored lines show the fitted functions. The results confirm that a k-NN graph does not capture real-world network distributions.

Scale-free *vs.* Geometric: One fundamental difference between a k-NN graph and a real-world network is that the former is built based on geometric distribution [20] whereas the latter usually follows *Scale-free* distribution [21], as shown in Fig. (2). For real-world networks with scale-free distributions ($P(d) \sim d^{-\gamma}$, where $P(d)$ denotes node degree distributions and γ is a parameter (typically within the range $2 < \gamma < 3$), majority nodes have very few connections. In comparison, each node in a k-NN graph (*i.e.* a random geometric graph) has at least k neighbors (assume using k-NN graphs) and some nodes have an extremely large number of node degree as shown in Fig. (2). This means that synthetic k-NN graphs are more smooth and have denser connections than real-world networks whereas real-world networks are more sparse and have severely biased node degree distributions. Therefore, directly replacing a smooth graph by using a network topology structure is ineffective, because all existing methods [18], [19] are based on k-NN graphs where majority nodes have similar node degree distributions.

Topology Structures *vs.* Node Content: The k-NN graphs used in existing co-clustering methods are often constructed by using feature based similarity between instances, under assumption that k-NN graph structure should be consistent with the instance feature values (*i.e.*, instances similar to each other in feature space are subject to a connection). In reality, network topology structures are, however, not always consistent with the node content, and nodes can be connected even if they do not share similar content. In Fig. (2), we build an 5-NN graph from CiteSeer data set by using feature values and compare 5-NN graph with the real CiteSeer citation network. The results show that instances (nodes) with high degrees are largely different between real CiteSeer network *vs.* 5-NN graph (Detailed information about CiteSeer is given in Experiments Section).

Therefore, directly utilizing linkage relationships to regularize the objective function, as most existing methods do (GNMF [22], DRCC [18], and LP-NMTF [23]), will propagate the inconsistency into the objective function and lead to deteriorated clustering results for networked data.

The above observations motivate our new co-clustering framework for networked data. To take special care of network topology structures, we advocate a *consensus factorization* principle to simultaneously factorize three types of relationships: network topology structures, feature-instance correspondences, feature-feature correlations, and further enforce the consensus of the factorized results for best clustering outcomes.

Compared to existing k-NN graph regularization based approaches, such as using regularization terms [18], our multi-relationship based factorization approach has two major advantages. First, simultaneously carrying out factorization for each relationship ensures that each factorization can best capture data distribution *w.r.t.* the underlying relationships, while a regularization can only restrict a solution but cannot discover new solutions. Second, our model uses a consensus approach to find the optimization factorization results which are consistent to all three types of relationships. This will eventually help solve the inconsistency between network topology structure and the node contents.

The key contribution of the paper is twofold:

- **Co-clustering for networked data:** We formulate a novel co-clustering research problem to simultaneously explore cluster structures for networked instances and features. Our solutions will help develop interesting approaches to find clusters and their respective features values under a network setting. Our research indicates that existing k-NN affinity graph based approaches cannot be directly utilized for networked data, mainly because they do not comply with real-world network characteristics. This observation will help interested readers design their own co-clustering methods by taking networked data distributions into consideration.
- **Consensus factorization:** We propose a consensus factorization model to factorize different types of relationships, and further explore their consensus for best clustering outcomes. Our consensus factorization approach can be extended to many other applications with a rich set of relationships in the data. Although some existing co-clustering methods have also considered instance and feature manifold (like DRCC and LP-FNMTF), these methods use strong constraints to force co-clustering results to be strictly consistent to the manifold. When topology structures and feature distributions are inconsistent, the results of these methods are severely inferior to our consensus factorization based approach.

The remainder of the paper is structured as follows. Notations and problem formulation are given in Sec. 2. Sec. 3 introduces the proposed CFOND algorithm and the derivation of the optimal solutions. The convergence analysis is reported in Sec. 4. Experiments are reported in Sec. 5, followed by literature review and related work in Sec. 6. We conclude the paper in Sec. 7.

2 Notations and Problem Formalization

In networked data setting, we are given a network $\mathcal{G} = \{V, E\}$, where each node (or instance) $v_i \in V$ is represented by a feature vector $\mathbf{x}_{\cdot i} \in \mathbb{R}_+^d$, and each edge $\{v_i, v_j\} \subseteq E$ encodes the relationship between nodes v_i and v_j. Typically, network G can be formulated by using two matrices $\mathbf{X} \in \mathbb{R}_+^{d\times n}$ and $\mathbf{W}_s \in \mathbb{R}_+^{n\times n}$. $\mathbf{X} = [\mathbf{x}_{\cdot 1}, \mathbf{x}_{\cdot 2}, \cdots, \mathbf{x}_{\cdot n}] = [\mathbf{x}_{1\cdot}^\top, \cdots, \mathbf{x}_{d\cdot}^\top]^\top$ is a *Feature-instance Adjacency Matrix* representing the content of each node, and $\mathbf{W}_s$ is a *Topology Matrix* which encodes the linkages between all nodes (*i.e.*, $[\mathbf{W}_s]_{ij} = 1$, if $\{v_i, v_j\} \subseteq E$ otherwise $[\mathbf{W}_s]_{ij} = 0$).

In the context of clustering, the goal is to cluster instances into different groups, with similar instances being assigned in one group. An instance may be exclusively assigned to only one cluster (*i.e.* hard clustering) or multiple clusters (*i.e.* soft clustering). We use an indicator matrix $\mathbf{G} = [\mathbf{g}_{1\cdot}^\top, \cdots, \mathbf{g}_{n\cdot}^\top] \in \mathbb{R}_+^{n\times c}$ to represent potential clustering result of instances. $\mathbf{g}_{ij}$ is the cluster membership of the i_{th} node corresponding to cluster $\mathcal{C}_j$. By using indicator matrix $\mathbf{G}$, the final clustering results can be obtained by choosing the cluster to which each node has the highest membership value. Similarly, we use another indicator matrix $\mathbf{F} \in \mathbb{R}_+^{d\times k}$ to represent clustering membership values of features.

The **aim** of co-clustering for networked data is to optimally group all nodes (instances) $\{\mathbf{x}_{\cdot 1}, \cdots, \mathbf{x}_{\cdot n}\}$ into c clusters $\{\mathcal{C}_j\}_{j=1}^c$, and simultaneously group features $\{\mathbf{x}_{1\cdot}, \cdots, \mathbf{x}_{d\cdot}\}$ into k clusters $\{\mathcal{K}_j\}_{j=1}^k$ ($c \ll n$ and $k \ll d$), such that the clusters have the best quality with respect to certain assessment criteria (we use clustering accuracy and normalized mutual information NMI in our experiments).

3 CFOND Algorithm

For networked data, information can be obtained from two major channels: network node content and topology structures. Network nodes provide detailed feature values to characterize node content information. We can collect all nodes as independent tabular instance-feature matrix (denoted by $\mathbf{X}$), named *Instance-feature Content Matrix*, to characterize network node content. Many existing co-clustering methods [18], [22], [23] can be applied to explore co-clustering results by factorizing $\mathbf{X}$. Meanwhile, network topology structures characterize dependency relationships between nodes in networks, and such relationships can be explicitly captured by using as an $n \times n$ matrix (denoted by $\mathbf{W}_s$), named *Network Topology Structure Matrix*.

Although some works have considered k-NN similarity based relationships between instances [24], [25], they cannot reveal the characteristic of each group but can only provide one-side clustering result on instances rather than co-clustering for both instances and features. More importantly, since we are trying to explore cluster structures for both instances and features, it is necessary to explicitly capture correlations between features (denoted by $\mathbf{W}_f$), named *Feature-feature Correlation Matrix*, and further integrate such relationships into the co-clustering process.

In this paper, we propose a consensus factorization method, CFOND, to incorporate all three types of information, node content ($\mathbf{X}$), network topology structures ($\mathbf{W}_s$), and feature-feature correlations $\mathbf{W}_f$, into consideration for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2846555, IEEE Transactions on Knowledge and Data Engineering

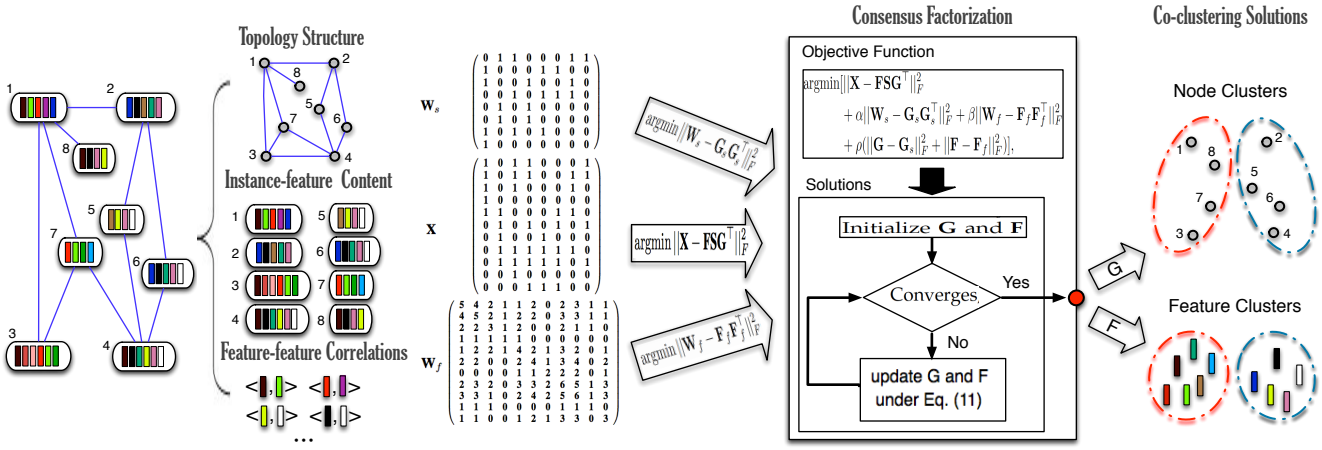


Fig. 3. An overview of the proposed CFOND Framework. CFOND carries out co-clustering by considering information from three aspects: network topology structures, feature-instance content relationships, and feature-feature correlations. CFOND factorizes each of them and then explores the consensus of their factorized results by using squared error terms to achieve optimal co-clustering results.

co-clustering. CFOND employs a factorization based approach to factorize $\mathbf{X}$, $\mathbf{W}_s$, and $\mathbf{W}_f$ separately, and further enforces constrains to ensure the consensus of their factorization results. This is essentially different from existing factorization based approaches [24], [25], which only factorize instance-feature matrix $\mathbf{X}$ and add other information as regularization terms to filter factorized results from $\mathbf{X}$.

In the following, we will introduce individual factorization components of CFOND and address its consensus factorization process. The overall framework of CFOND is shown in Fig. (3).

Instance-feature Content Matrix Factorization: Instance-feature content matrix ($\mathbf{X}$) provides tabular relationships between instances (nodes) and features. We can use Nonnegative Matrix Factorization (NMF) to factorize $\mathbf{X}$ into two nonnegative matrices $\mathbf{F}$ and $\mathbf{G}$, with the objective of minimizing squared errors between $\mathbf{X}$ and its approximation,

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

Where $||\mathbf{A}||_F$ is the Frobenius norm of the matrix $\mathbf{A}$ [26]. In reality, because two-factor NMF in Eq. (1) is restrictive, in which the cluster numbers c and k have to be equal, one can introduce an additional factor $\mathbf{S} \in \mathbb{R}^{c \times k}$ to absorb the different scales of $\mathbf{X}$, $\mathbf{F}$ and $\mathbf{G}$. This leads to an extension of NMF, named NMTF [14], [15]:

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

In Eq. (2), latent matrix $\mathbf{S}$ provides increased degrees of freedom such that the low-rank matrix representation remains accurate, while c and m can have different values.

Our factorization process is different from a previous orthogonal NMF [14]. In [14], the encoding matrix needs to satisfy both orthogonality and non-negativity constraints, so their encoding matrix has a form of a cluster indicator matrix, with only one non-zero element existing in each row. This results in hard-clustering and has been further improved for k-NN method [15]. However, our method is a soft-clustering method that gives the confidence degree of a node belonging each cluster, so we can further discover intra-relationships between different clusters.

Network Topology Structure Matrix Factorization: Network topology structure matrix $\mathbf{W}_s$ contains pairwise node topology relationships in the structure space, which offers additional information for characterizing similarity between nodes for co-clustering. Accordingly, we can factorize matrix $\mathbf{W}_s$ as an $n \times c$ matrix $\mathbf{G}_s$, where $\mathbf{G}_s \in \mathbb{R}_+^{n \times c}$ is an indicator matrix showing potential clustering results of network nodes by only using topology structures:

$$\underset{\mathbf{G}_s}{\operatorname{argmin}} J_3 = ||\mathbf{W}_s - \mathbf{G}_s\mathbf{G}_s^\top||_F^2, \;\; s.t.\; \mathbf{G}_s \geq 0, \tag{3}$$

It is noteworthy that $\mathbf{G} \in \mathbb{R}^{n \times c}$ in J_2 and $\mathbf{G}_s \in \mathbb{R}^{n \times c}$ in J_3 each contains separated factorization results for all network nodes. By using this approach, we allow factorization for $\mathbf{X}$ and $\mathbf{W}_s$ to have maximum freedom to explore its optimal results, respectively. The consensus factorization process will latter enforce these two sets of results to be consistent for optimal outcomes.

Feature-feature Correlation Matrix Factorization: Similarly, to enhance feature clustering results, CFOND also uses a feature-feature correlation matrix $\mathbf{W}_f \in \mathbb{R}_+^{d \times d}$ to capture pair-wise feature correlations. Intuitively, if features $\mathbf{x}_{i\cdot}$ and $\mathbf{x}_{j\cdot}$ are highly correlated (*e.g.* two keywords always co-occur), they should be more likely being clustered to the same feature cluster. Therefore, we can use correlation measures, such as heat Kernels [27] or Neighbor-based method [18], to construct $\mathbf{W}_f$. For simplicity, we use linear kernel $[\mathbf{W}_f]_{ij} =< \mathbf{x}_{i\cdot}, \mathbf{x}_{j\cdot} >$ in our experiments, where $\mathbf{x}_{i\cdot}$ is a vector representation of the i_{th} feature across all nodes, and $< \mathbf{x}_{i\cdot}, \mathbf{x}_{j\cdot} >$ is the similarity degree of $\mathbf{x}_{i\cdot}$ and $\mathbf{x}_{j\cdot}$.

Similar to Eq. (3), the factorization of feature matrix $\mathbf{W}_f$ is as follows,

$$\underset{\mathbf{F}_f}{\operatorname{argmin}} J_4 = ||\mathbf{W}_f - \mathbf{F}_f\mathbf{F}_f^\top||_F^2, \;\; s.t.\; \mathbf{F}_f \geq 0, \tag{4}$$

Consensus Factorization: In the above factorization processes, the objective functions J_2, J_3 and J_4 each provides clustering results from different aspects (node content, topology structures, and feature correlations). To ensure that final results are consistent, CFOND proposes a consensus

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2846555, IEEE Transactions on Knowledge and Data Engineering

factorization objective function to jointly formulate J_2, J_3 and J_4 into a unified objective function:

$$\begin{aligned} J_5 = &||\mathbf{X} - \mathbf{F}\mathbf{S}\mathbf{G}^\top||_F^2 + \alpha||\mathbf{W}_s - \mathbf{G}_s\mathbf{G}_s^\top||_F^2 + \beta||\mathbf{W}_f - \mathbf{F}_f\mathbf{F}_f^\top||_F^2 \\ &+ \rho(||\mathbf{G} - \mathbf{G}_s||_F^2 + ||\mathbf{F} - \mathbf{F}_f||_F^2), \\ &s.t.\ \mathbf{F} \geq 0, \mathbf{G} \geq 0, \mathbf{G}_s \geq 0, and\ \mathbf{F}_f \geq 0, \end{aligned} \tag{5}$$

The objective function in Eq.(5) is to factorize $\mathbf{X}$, $\mathbf{W}_g$, and $\mathbf{W}_f$ separately, and enforce the factorization consensus among all three aspects: node content, network structures, and feature correlations. For instance, $\mathbf{G}$ and $\mathbf{G}_s$ provide clustering results from instance-feature and topology structures, respectively. $||\mathbf{G} - \mathbf{G}_s||_F^2$ enforces that $\mathbf{G}$ should be maximally consistent with $\mathbf{G}_s$. Similarly, $||\mathbf{F} - \mathbf{F}_f||_F^2$ makes $\mathbf{F}$ and $\mathbf{F}_f$ close to each other. α and β in Eq.(5) are regularization parameters to balance each factorization part. ρ trade-offs the consistent degree. Intuitively, a very large ρ value will make $\mathbf{G} = \mathbf{G}_s$ and $\mathbf{F} = \mathbf{F}_f$, while a small ρ would make $\mathbf{G}$ and $\mathbf{G}_s$ totally independent (*e.g.*, $\rho = 0$). As a result, our method provides increased degrees of freedom to exploit different information encoded in networked data.

Latent matrix $\mathbf{S}$ not only absorbs the different scales of $\mathbf{X}$, $\mathbf{F}$ and $\mathbf{G}$, but also reveals the corresponding relationships between the node clustering and feature clustering results. $\mathbf{S}_{ij}$ uncovers the relative weight between feature cluster i and node cluster j. In the experiments, we will report the detailed analysis of $\mathbf{S}$.

3.1 CFOND Optimal Solutions

Minimizing Eq.(5) is with respect to $\mathbf{G}$, $\mathbf{G}_s$, $\mathbf{F}$, $\mathbf{F}_f$, and $\mathbf{S}$, and the function is not convex in all variables together. We will present an alternating scheme to optimize the objective. In other words, we will optimize the objective *w.r.t.* one variable while fixing the other variables. This procedure repeats until convergence.

To optimize Eq.(5) *w.r.t.* $\mathbf{G}$, $\mathbf{G}_s$, $\mathbf{F}$, $\mathbf{F}_f$, and $\mathbf{S}$, five Lagrangian multipliers are introduced as follows:

$$\begin{aligned} &\lambda_G \in \mathbb{R}^{n\times c},\ \lambda_{G_\star} \in \mathbb{R}^{n\times c}, \lambda_F \in \mathbb{R}^{d\times k},\ \lambda_{F_\star} \in \mathbb{R}^{d\times k}, \\ &and\ \lambda_S \in \mathbb{R}^{k\times c}, \end{aligned} \tag{6}$$

Then the Kuhn-Tucker condition (KKT condition) [28] characterizes the necessary and sufficient condition that the optimal solutions need to satisfy:

$$\begin{aligned} &\lambda_G \odot \mathbf{G} = 0; \quad \lambda_{G_\star} \odot \mathbf{G}_s = 0; \quad \lambda_F \odot \mathbf{F} = 0; \\ &\lambda_{F_\star} \odot \mathbf{F}_f = 0; \quad \lambda_S \odot \mathbf{S} = 0 \end{aligned} \tag{7}$$

and "$\odot$" is the Hadamard product operator (as the operator ".*" in matlab), *i.e.* $[\mathbf{A} \odot \mathbf{B}]_{ij} = \mathbf{A}_{ij} \cdot \mathbf{B}_{ij}$. Thus the Lagrangian function is

$$\begin{aligned} L = &J_5 - tr(\lambda_G\mathbf{G}) - tr(\lambda_{G_\star}\mathbf{G}_s) - tr(\lambda_F\mathbf{F}) \\ &- tr(\lambda_{F_\star}\mathbf{F}_f) - tr(\lambda_S\mathbf{S}) \\ = &tr((\mathbf{X} - \mathbf{F}\mathbf{S}\mathbf{G}^\top)^\top(\mathbf{X} - \mathbf{F}\mathbf{S}\mathbf{G}^\top)) \\ &+ \alpha tr((\mathbf{W}_s - \mathbf{G}_s\mathbf{G}_s^\top)^\top(\mathbf{W}_s - \mathbf{G}_s\mathbf{G}_s^\top)) \\ &+ \beta tr((\mathbf{W}_f - \mathbf{F}_f\mathbf{F}_f^\top)^\top(\mathbf{W}_f - \mathbf{F}_f\mathbf{F}_f^\top)) \\ &+ \rho tr((\mathbf{G} - \mathbf{G}_s)^\top(\mathbf{G} - \mathbf{G}_s)) \\ &+ \rho tr((\mathbf{F} - \mathbf{F}_f)^\top(\mathbf{F} - \mathbf{F}_f)) \\ &- tr(\lambda_G\mathbf{G}) - tr(\lambda_{G_\star}\mathbf{G}_s) \\ &- tr(\lambda_F\mathbf{F}) - tr(\lambda_{F_\star}\mathbf{F}_f) \\ &- tr(\lambda_S\mathbf{S}) \end{aligned} \tag{8}$$

Setting partial derivatives of $\mathbf{G}$, $\mathbf{G}_s$, $\mathbf{F}$ and $\mathbf{F}_f$ to zero, we have

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{G}} &= -2\mathbf{X}^\top\mathbf{F}\mathbf{S} + 2\mathbf{G}\mathbf{S}^\top\mathbf{F}^\top\mathbf{F}\mathbf{S} + 2\rho\mathbf{G} - 2\rho\mathbf{G}_s - \lambda_\mathbf{G} = 0 \\ \frac{\partial L}{\partial \mathbf{F}} &= -2\mathbf{X}\mathbf{G}\mathbf{S}^\top + 2\mathbf{F}\mathbf{S}\mathbf{G}^\top\mathbf{G}\mathbf{S}^\top + 2\rho\mathbf{F} - 2\rho\mathbf{F}_f - \lambda_F = 0 \\ \frac{\partial L}{\partial \mathbf{G}_s} &= -2\alpha\mathbf{W}_s^\top\mathbf{G}_s - 2\alpha\mathbf{W}_s\mathbf{G}_s + 4\alpha\mathbf{G}_s\mathbf{G}_s^\top\mathbf{G}_s \\ &\quad - 2\rho\mathbf{G} + 2\rho\mathbf{G}_s - \lambda_{G_\star} = 0 \\ \frac{\partial L}{\partial \mathbf{F}_f} &= -2\beta\mathbf{W}_s^\top\mathbf{F}_f - 2\beta\mathbf{W}_s\mathbf{F}_f + 4\beta\mathbf{F}_f\mathbf{F}_f^\top\mathbf{F}_f \\ &\quad - 2\rho\mathbf{F} + 2\rho\mathbf{F}_f - \lambda_{F_\star} = 0 \\ \frac{\partial L}{\partial \mathbf{S}} &= -2\mathbf{F}^\top\mathbf{X}\mathbf{G} + 2\mathbf{F}^\top\mathbf{F}\mathbf{S}\mathbf{G}^\top\mathbf{G} - \lambda_S = 0 \end{aligned} \tag{9}$$

To eliminate Lagrangian multipliers by using Eq. (7), we have

$$\begin{aligned} (\mathbf{X}^\top\mathbf{F}\mathbf{S} + \rho\mathbf{G}_s) \odot \mathbf{G} &= (\mathbf{G}\mathbf{S}^\top\mathbf{F}^\top\mathbf{F}\mathbf{S} + \rho\mathbf{G}) \odot \mathbf{G} \\ (\mathbf{X}\mathbf{G}\mathbf{S}^\top + \rho\mathbf{F}_f) \odot \mathbf{F} &= (\mathbf{F}\mathbf{S}\mathbf{G}^\top\mathbf{G}\mathbf{S}^\top + \rho\mathbf{F}) \odot \mathbf{F} \\ (\rho\mathbf{G} + 2\alpha\mathbf{W}_s^\top\mathbf{G}_s) \odot \mathbf{G}_s &= (2\alpha\mathbf{G}_s\mathbf{G}_s^\top\mathbf{G}_s + \rho\mathbf{G}_s) \odot \mathbf{G}_s \\ (\rho\mathbf{F} + 2\beta\mathbf{W}_s^\top\mathbf{F}_f) \odot \mathbf{F}_f &= (2\beta\mathbf{F}_f\mathbf{F}_f^\top\mathbf{F}_f + \rho\mathbf{F}_f) \odot \mathbf{F}_f \\ \mathbf{F}^\top\mathbf{X}\mathbf{G} \odot \mathbf{S} &= \mathbf{F}^\top\mathbf{F}\mathbf{S}\mathbf{G}^\top\mathbf{G} \odot \mathbf{S} \end{aligned} \tag{10}$$

Eq. (10) leads to the following updating formulas

$$\begin{aligned} &\mathbf{G} \leftarrow \mathbf{G} \odot \frac{\mathbf{X}^\top\mathbf{F}\mathbf{S} + \rho\mathbf{G}_s}{\mathbf{G}\mathbf{S}^\top\mathbf{F}^\top\mathbf{F}\mathbf{S} + \rho\mathbf{G}}; \quad \mathbf{F} \leftarrow \mathbf{F} \odot \frac{\mathbf{X}\mathbf{G}\mathbf{S}^\top + \rho\mathbf{F}_f}{\mathbf{F}\mathbf{S}\mathbf{G}^\top\mathbf{G}\mathbf{S}^\top + \rho\mathbf{F}} \\ &\mathbf{G}_s \leftarrow \mathbf{G}_s \odot \frac{\rho\mathbf{G} + 2\alpha\mathbf{W}_s^\top\mathbf{G}_s}{2\alpha\mathbf{G}_s\mathbf{G}_s^\top\mathbf{G}_s + \rho\mathbf{G}_s} j \\ &\mathbf{F}_f \leftarrow \mathbf{F}_f \odot \frac{\rho\mathbf{F} + 2\beta\mathbf{W}_s^\top\mathbf{F}_f}{2\beta\mathbf{F}_f\mathbf{F}_f^\top\mathbf{F}_f + \rho\mathbf{F}_f}; \quad \mathbf{S} \leftarrow \mathbf{S} \odot \frac{\mathbf{F}^\top\mathbf{X}\mathbf{G}}{\mathbf{F}^\top\mathbf{F}\mathbf{S}\mathbf{G}^\top\mathbf{G}} \end{aligned} \tag{11}$$

After updating, the final clustering indicator matrices are the results of consensus factorizations, *i.e.* $\mathbf{G}_{Final} = \mathbf{G} + \mathbf{G}_s$ and $\mathbf{F}_{Final} = \mathbf{F} + \mathbf{F}_f$, respectively. In summary, we present the alternating iterative algorithm for optimizing Eq. (5) in Algorithm 1. The convergence analysis of Algorithm 1 is in Section 4.

Algorithm 1 CFOND

Require: Data Matrix $\mathbf{X} = [\mathbf{x}_{\cdot 1}, \cdots, \mathbf{x}_{\cdot n}] \in \mathbb{R}_+^{d\times n}$, Structure Matrix $\mathbf{W}_s \in \mathbb{R}_+^{n\times n}$, and Clustering number c and k.

Constructed $\mathbf{W}_f$, i.e., $[\mathbf{W}_f]_{ij} =< \mathbf{x}_{i\cdot}, \mathbf{x}_{j\cdot} >$
Initialize $\mathbf{G}$ and $\mathbf{F}$ using K-means on $\mathbf{X}$ and $\mathbf{X}^\top$, respectively;
Initialize $\mathbf{G}_s = \mathbf{G}$ and $\mathbf{F}_f = \mathbf{F}$;
repeat
  $\mathbf{G} \leftarrow \mathbf{G} \odot \frac{\mathbf{X}^\top\mathbf{F}\mathbf{S} + \rho\mathbf{G}_s}{\mathbf{G}\mathbf{S}^\top\mathbf{F}^\top\mathbf{F}\mathbf{S} + \rho\mathbf{G}}$;
  $\mathbf{F} \leftarrow \mathbf{F} \odot \frac{\mathbf{X}\mathbf{G}\mathbf{S}^\top + \rho\mathbf{F}_f}{\mathbf{F}\mathbf{S}\mathbf{G}^\top\mathbf{G}\mathbf{S}^\top + \rho\mathbf{F}}$;
  $\mathbf{G}_s \leftarrow \mathbf{G}_s \odot \frac{\rho\mathbf{G} + 2\alpha\mathbf{W}_s^\top\mathbf{G}_s}{2\alpha\mathbf{G}_s\mathbf{G}_s^\top\mathbf{G}_s + \rho\mathbf{G}_s}$;
  $\mathbf{F}_f \leftarrow \mathbf{F}_f \odot \frac{\rho\mathbf{F} + 2\beta\mathbf{W}_s^\top\mathbf{F}_f}{2\beta\mathbf{F}_f\mathbf{F}_f^\top\mathbf{F}_f + \rho\mathbf{F}_f}$;
  $\mathbf{S} \leftarrow \mathbf{S} \odot \frac{\mathbf{F}^\top\mathbf{X}\mathbf{G}}{\mathbf{F}^\top\mathbf{F}\mathbf{S}\mathbf{G}^\top\mathbf{G}}$;
until Converges;
$\mathbf{G}_{Final} = \mathbf{G} + \mathbf{G}_s$;
$\mathbf{F}_{Final} = \mathbf{F} + \mathbf{F}_f$;
Output: Cluster indicator matrices $\mathbf{G}$, $\mathbf{G}_s$, $\mathbf{F}$ and $\mathbf{F}_f$ for instance and feature clustering tasks, respectively.

3.2 Differentiation from other NMF/NMTF Objectives

In Table 1, we summarize the objective functions of three state-of-the-art NMF/NMTF based co-clustering methods: GNMF [22], DRCC [18], and LP-NMTF [23]. It is clear that GNMF is an extension of the NMF method [14], with one additional constraint enforcing the feature clustering indicator matrix F to be consistent to the network topology structures (W_s). For DRCC method, it factorizes instance-feature content matrix X, and also enforces the consistency between feature partitioning matrix G and feature affinity matrix (L_G), as well as the consistency between instance partitioning matrix F and instance affinity matrix (L_F). LP-NMTF has a similar objective function as DRCC but uses a locality preserved way to make it computationally efficient.

Compared to GNMF, DRCC, and LP-NMTF, CFOND has the following noticeable differences. First, the factorization of CFOND is explicitly carried out on three sources: instance-feature content relationships (X), network topology structures (W_s), and feature-feature correlations (W_f), whereas the other three methods' factorization is only limited to the instance-feature content matrix X. Second, GNMF, DRCC, and LP-NMTF have all considered instance (or feature) correlations in their regularization, which are captured in the second and/or the third terms of their objective functions. However, their instance correlations are based on affinity matrix derived k-NN graphs. As we have explained in Sec. 1, k-NN graphs are synthetic, and are fundamentally different from real-world networks. Therefore these methods cannot effectively accommodate network topology structure matrix W_s as regularizations to co-cluster networked data (the restrictions of real-world networks will be too strong for them to discover suitable solutions). As a result, the consensus factorization of multiple relationship matrices allows CFOND to maximally consider both content and structure information in networked data for best clustering results.

TABLE 1
Objective Functions of GNMF, DRCC and LP-NMTF

Method	*Objective Function*
GNMF	$\mathcal{O}_1 = \|\|X - GF^\top\|\|_F^2 + \lambda Tr(F^\top L F)$ where $D_{jj} = \sum_l (W_s)_{jl}, L = D - W_s$
DRCC	$\mathcal{O}_2 = \|\|X - GSF^\top\|\|_F^2 + \lambda Tr(F^\top L_F F) + \mu Tr(G^\top L_G G)$
LP-NMTF	$\mathcal{O}_3 = \|\|X - FSG^T\|\|_F^2 + \alpha\|\|G - B_d Q_d\|\|_F^2 + \beta\|\|F - B_f Q_f\|\|_F^2$ s.t. $Q_d^\top Q_d = I, Q_f^\top Q_f = I$

4 CFOND Convergence Analysis

In the following, we will use the auxiliary function approach [29] to analyze the convergence of the updating rule in Eq.(11).

Definition 1. *$Z(h, h')$ is an auxiliary function for $P(h)$ if the following conditions are satisfied.*

$$Z(h, h') \geq P(h), Z(h, h) = P(h)$$

The auxiliary function is a useful concept because of the following lemma, which is also graphically illustrated in Fig. 4.

Lemma 1. *If Z is an auxiliary function for P, then P is non-increasing under the update*

$$h^{(t+1)} = \arg\min_h Z(h, h^t)$$

Proof.

$$P(h^{(t+1)}) \leq Z(h^{(t+1)}, h^{(t)}) \leq Z(h^{(t)}, h^{(t)}) = P(h^t)$$

□

We will show that by defining the appropriate auxiliary functions $Z(h, h^t)$ for J_5, the update rules in Eq. (11) easily follow from Lemma 1. The objective value J_5 in Eq. (5) will monotonically decreasing during iterations. Take the update rule of G_s for instance, for any element $g_{s(ij)}$ in $\mathbf{G}_s$, We use P_{ij} to indicate the part of L which is relevant to $g_{s(ij)}$. The first and second order of derivatives of P_{ij} are computed as

$$\begin{aligned} P'_{ij} &= (\tfrac{\partial L}{\partial \mathbf{G}_s})_{ij} \\ P''_{ij} &= -2\alpha(\mathbf{W}_s + \mathbf{W}_s^\top)_{jj} + 4\alpha(\mathbf{G}_s^\top \mathbf{G}_s)_{jj} + 2\rho \\ &= 4\alpha(\mathbf{G}_s^\top \mathbf{G}_s)_{jj} + 2\rho \end{aligned}$$

The last equation holds because the topological proximity matrix $[\mathbf{W}_s]_{jj} = 0$ as there is no link for a node to itself.

Lemma 2. *Function*

$$\begin{aligned} Z(g, g_{s(ij)}^{(t)}) = \; & P_{ij}(g_{s(ij)}^{(t)}) + P'_{ij}(g_{s(ij)}^{(t)})(g - g_{s(ij)}^{(t)}) \\ & + \tfrac{(2\alpha \mathbf{G}_s \mathbf{G}_s^\top \mathbf{G}_s + \rho \mathbf{G}_s)_{ij}}{g_{s(ij)}^{(t)}} (g - g_{s(ij)}^{(t)})^2 \end{aligned}$$

is a proper auxiliary function for $P_{ij}(g)$.

Proof. It is straight-forward that $Z(g, g) = P_{ij}(g)$, and thus we only need to verify that $Z(g, g_{s(ij)}^{(t)}) \geq P_{ij}(g)$. Using Taylor series,

$$\begin{aligned} P_{ij}(g) = \; & P_{ij}(g_{s(ij)}^{(t)}) + P'_{ij}(g_{s(ij)}^{(t)})(g - g_{s(ij)}^{(t)}) \\ & + (2\alpha(\mathbf{W}_s^T \mathbf{W}_s)_{jj} + \rho)(g - g_{s(ij)}^{(t)})^2 \end{aligned}$$

Because,

$$(2\alpha \mathbf{G}_s \mathbf{G}_s^\top \mathbf{G}_s + \rho \mathbf{G}_s)_{ij} \geq g_{s(ij)}^{(t)} (2\alpha(\mathbf{W}_s^\top \mathbf{W}_s)_{jj} + \rho)$$

Thus $Z(g, g_{s(ij)}^{(t)}) \geq P_{ij}(g)$, and Lemma 2 holds. □

Theorem 1. *The objective value J_5 in Eq. (5) is nonincreasing under the updated rules of Eq. (11).*

Proof. Replacing the auxiliary function in Lemma 2 into Lemma 1, we can get g by minimizing $Z(g, g_{s(ij)}^{(t)})$. Setting the derivative $\frac{\partial Z(g, g_{s(ij)}^{(t)})}{\partial g} = 0$, we have:

$$\begin{aligned} g = g_{s(ij)}^{(t+1)} &= g_{s(ij)}^{(t)} - \tfrac{g_{s(ij)}^{(t)} P'_{ij}}{4\alpha \mathbf{G}_s \mathbf{G}_s^\top \mathbf{G}_s + 2\rho \mathbf{G}_s} \\ &= g_{s(ij)}^{(t)} \tfrac{(\rho \mathbf{G} + 2\alpha \mathbf{W}_s^\top \mathbf{G}_s)_{ij}}{(2\alpha \mathbf{G}_s \mathbf{G}_s^\top \mathbf{G}_s + \rho \mathbf{G}_s)_{ij}} \end{aligned}$$

Since Lemma 2 is an auxiliary function, J_5 is nonincreasing under this update rule, according to Lemma 1. This updating rule is essentially consistent with Eq. (11). Similarly, J_5 can be shown to be nonincreasing under the

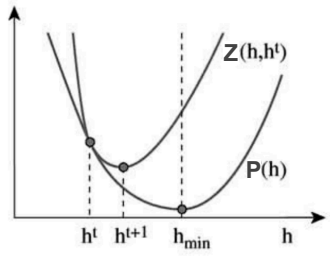


Fig. 4. Minimizing the auxiliary function $Z(h, h^t) \geq P(h)$ guarantees that $P(h^{t+1}) \leq P(h^t)$ for $h^{n+1} = argmin_h Z(h, h^t)$.

updating rule for **G**, **F**, $\mathbf{F}_s$, and **S** in Eq. (11). As the objective function is lower bounded by 0, the convergence aspect is proved. □

It is worthy noting that our multiplicative update rules in Eq. (11) follow the similar ideas of Lee and Seung's proof in the original NMF paper [29] and GNMF in [22]. A recent study [30] shows that Lee and Seung's multiplicative algorithm [29] cannot guarantee the convergence to a stationary point. Particularly, Lin [30] suggests minor modifications on Lee and Seung's algorithm which can converge. Our updating rules in Eq. (11) are essentially similar to the updating rules for NMF and therefore Lin's modifications can also be applied.

5 Experiments

Benchmark Methods: We compare CFOND with the following state-of-the-art co-clustering methods: GNMF [22], DRCC [18], LP-NMTF [23], and iTopicModel [31]. Meanwhile, we also report the clustering results from k-means and NMF [29] (although they are not co-clustering methods, but they are used as baseline to justify the performance of all co-clustering methods).

- **k-means** is a method of vector quantization, originally from signal processing, popularly used as a baseline for clustering analysis [32]. k-means aims to partition n instances into k clusters with each instance being assigned to the cluster whose center has the smallest distance to the instance.
- **NMF** is a relaxation technique for clustering. It has shown remarkable progress in the past decade [15], [29], [33]. NMF finds a low-rank approximating matrix to the input non-negative data matrix, where the most popular approximation criterion or divergence in NMF is the Least Square Error (LSE).
- **GNMF** is a graph based approach for parts-based data representation in order to overcome the limitation that NMF fails to consider geometric structures in the data. GNMF constructs an affinity graph to encode geometrical information and seeks a matrix factorization consistent to the graph structures [22].
- **DRCC** is a Dual Regularized Co-Clustering method based on semi-nonnegative matrix tri-factorization. It constructs two synthetic graphs, data graph and feature graph, to explore the geometric structure of data manifold and feature manifold. By using two graph regularizers, DRCC formulates a semi-nonnegative matrix tri-factorization objective function, requiring that cluster labels of data points are smooth with respect to the data manifold, while the cluster labels of features are smooth with respect to the feature manifold [18].
- **LP-NMTF** is a Locality Preserved Fast Nonnegative Matrix Tri-Factorization approach to constrain the factor matrices of NMF to be cluster indicator matrices. As a result, the optimization problem can be decoupled, which results in much smaller size subproblems requiring much less matrix multiplications. This approach was claimed to work well for large-scale input data [23].
- **iTopicModel** follows the traditional topic model to characterize the generation of text for each document, by formulating a joint distribution function which considers texts and inter-connection relationships between documents. iTopicModel seeks to maximize the log-likelihood of the joint probability in order to estimate topic models [31].

Performance Metrics: In order to assess the performance of different algorithms, we employ two commonly used clustering performance metrics: clustering accuracy and normalized mutual information (NMI) [22]. More specifically, each node of our benchmark data sets (networks) has a ground truth label (because they are built for classification purposes). In our experiments, we set the number of clusters as the same number of classes of the network. For each node cluster, we will find majority class label of nodes in this cluster, and divide the number of nodes with the majority class label by the cluster size, which will result in a clustering accuracy. The total clustering accuracy is based on the average clustering accuracy across all clusters. Meanwhile, $NMI = MI(\mathcal{C}, P)/\sqrt{H(\mathcal{C})H(P)}$, where the random variables $\mathcal{C}$ and P denote the cluster and class sizes, respectively. The value of NMI is in the interval: $[0, 1]$, and a larger value indicates a better clustering result.

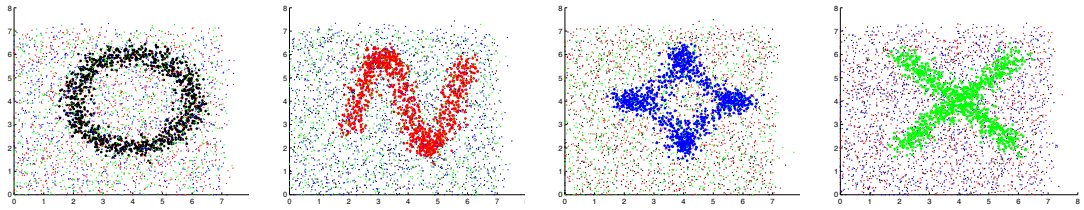
TABLE 2
Description of benchmark data

Data Sets	# *instance*	# *feature*	# *edge*	# *class*
Cora	2,708	1,433	5,429	7
CiteSeer	3,312	3,703	4,732	6
PubMed	19,717	500	44,338	3
Attack$_1$	1,293	106	3,172	6
Attack$_2$	1,293	106	571	6
Synthetic	4,000	8	3,057	4

Benchmark Networks: In our experiments, we use five real-world networks and one synthetic network to evaluate the algorithm performance. Table 2 summarizes their data characteristics.

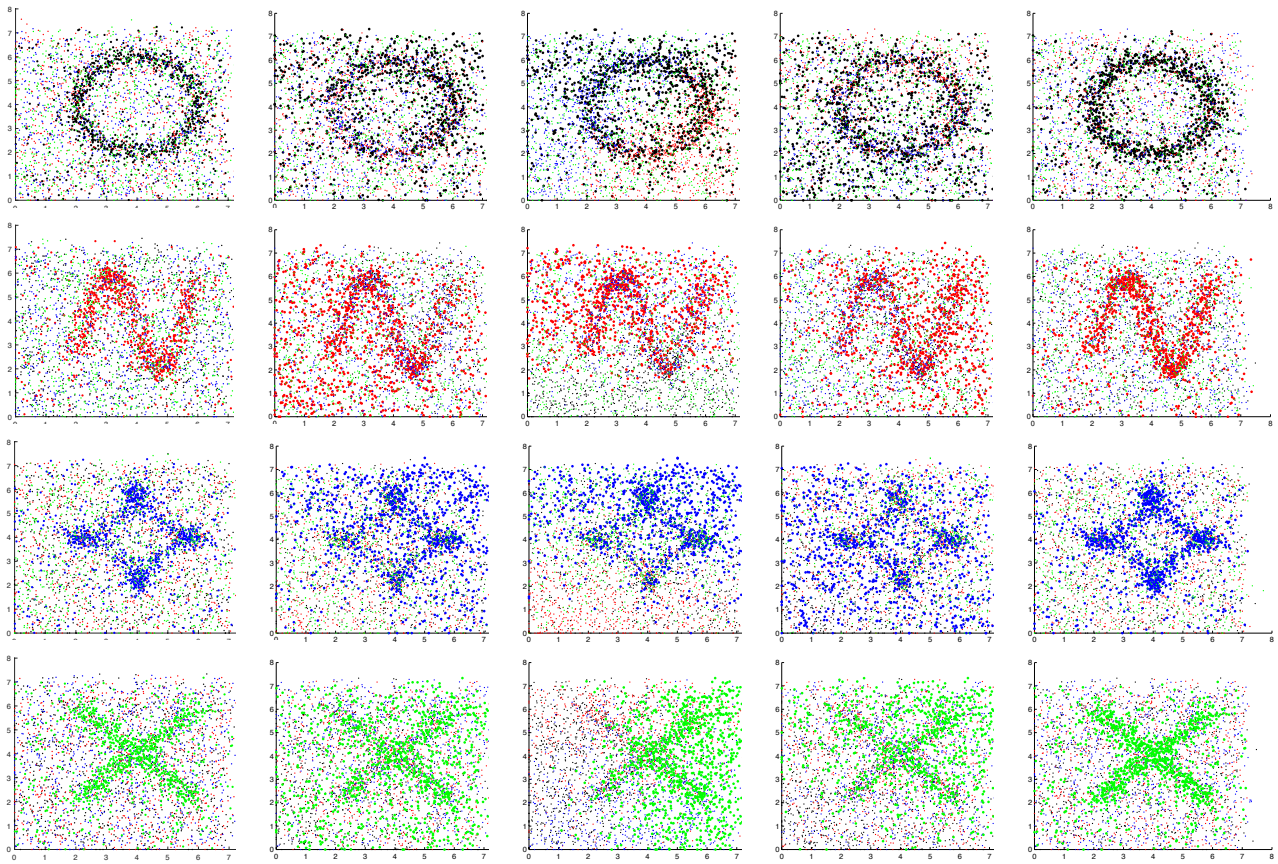
Synthetic Network: In order to visually examine the co-clustering quality, we design a synthetic network with 4000 nodes. We equally divide instances into four clusters, so each cluster has 1000 nodes (the four clusters are shown in Fig. 5). In addition, each instance has eight features, which are also equally divided into four parts with each part containing two features. For each feature part, the two features are unique for one instance cluster but randomly

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2846555, IEEE Transactions on Knowledge and Data Engineering



(a) (b) (c) (d)

Fig. 5. The genuine clusters of the synthetic network (best viewed in color). (a) The distribution of features 1 and 2 is a circle; (b) The distribution of features 3 and 4 is a sine function; (c) The distribution of features 5 and 6 is a star function, and (d) The distribution of features 7 and 8 is a absolute value function.



(p) iTopicModel (q) DRCC (r) GNMF (s) LP-FNMTF (t) CFOND

Fig. 6. Co-clustering results from iTopicModel, DRCC, GNMF, LP-FNMTF, and CFOND on the synthetic network. The genuine clusters are a circle function, a sine function, a star function, and a absolute value function showing in Fig. 5. Each column shows co-clustering results of one method.

appearing in other three clusters (we refer to the two unique features as the effective features of the instance cluster).

For each instance cluster $\mathcal{C}_i$, the values of the allocated two effective features of the cluster form a specific shape (which defines the underlying cluster). In other words, the two features' values in the instances in cluster $\mathcal{C}_i$ follow a given distribution, including a circle function, a sine function, a star function, and an absolute value function, respectively, which are shown in Fig. 5). Meanwhile, we also add white Gaussian noise with the signal-to-noise ratio $snr = 25\ dB$ to gently perturb the data distributions in the effective features, which will make the clustering tasks more challenging. The appearances of the two features in the instances with other clusters $\mathcal{C}_j$, where $j \neq i$, follow a random distribution. For example, (a) and (b) in Fig. 5 show the genuine cluster structures corresponding to features 1 & 2, and features 3 & 4, respectively. By doing so, we can visually show the clustering results in a two-dimensional space to compare the performance of different methods.

The topology structures of the synthetic network follow a scale-free distribution. The probability $P(m)$ of nodes (instances) in the network having m connections follow a power-law distribution $P(m) \sim m^{-\gamma}$, where γ is the dependency parameter (we set $\gamma = 2.5$ in our experiments). More specifically, we randomly set the fraction $P(m)$ of nodes having m connections to other nodes in the synthetic network, where $1 \leq m \leq 10$. For each node with m edges, we set the fraction 0.7 of its edges connecting to the nodes

within the same class and 0.3 of its edges being connected to the nodes in other classes, randomly. By doing so, the edge connections will ensure majority nodes within the same class to have a better chance of being connected to nodes within the same class, and the random connects to nodes from other classes will simulate the real-world scenario where edges are not always consistent with the node content and therefore complicate the clustering tasks.

Real-world Networks: The Cora[1], CiteSeer[1] and PubMed[1] networks consist of scientific publications from different domains. For Cora and CiteSeer, each publication in the networks is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. For pubMed, each publication in the dataset is described by a TF/IDF weighted word vector from the dictionary. The citation relations are used to construct the network structures. Attack1[1] and Attack2[1] data sets contain two types of information related to terrorism attack entities: the attributes of the entities and the links that connect various entities together to form a graph structure. Attack1 is based on co-located attacks and Attack2 is based on co-located attacks organized by the same terrorist organization.

For all benchmark networks, each instance/node has a true class label, which is used for validation only. In other words, the class labels are unseen (not exposed) for all co-clustering methods (including parameter tuning process). When validating the clustering accuracy, we compare the clustering results with the instance labels and validate the algorithm performance.

Parameters Setting: Each clustering algorithm has one or more parameters to be tuned. In order to make fair comparisons, we run these algorithms under different parameter settings and select the best average result for comparisons (using normalized mutual information NMI). For all clustering methods, we set the number of clusters equal to the true number of classes for all the data sets. We construct nearest-neighbor graph following [18], where the neighborhood size for graph construction is set by searching the grid of $\{1, 2, \cdots, 10\}$, and the regularization parameters (*i.e.*, α, β and ρ in Eq.(5)) are set by searching the grid of $\{0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000\}$. For iteration-based method, we set the iteration number to 80 in order to make sure all the compared methods can fully reach their convergence.

For each method (including CFOND), clustering is repeated multiple times by using all assortments of parameters with the values of $\{0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000\}$, and report the best NMI result (Similar to the experimental setting in [23]). We also report the co-clustering results of CFOND by changing parameters on real-world networks.

GNMF, DRCC and LP-NMTF deal with co-clustering on manifold, so we use the same $\mathbf{W}_s$ and $\mathbf{W}_f$ as the data manifold and feature manifold, respectively.

For co-clustering methods, including GNMF, DRCC, LP-NMTF, and our CFOND methods, the number of feature clusters is set to be the same as that of the data clusters, *i.e.*, $c = k$. The same constrain is applied to iTopicModel method as well.

1. http://linqs.cs.umd.edu/projects//projects/lbc/index.html

All experiments are conducted on a cluster machine with 16GB RAM and Intel Core^{TM} i7 3.20 GHZ CPU.

5.1 Experimental Comparisons on Synthetic Network

In Fig. 6, we visually report the results of major compared methods: iTopicModel, DRCC, GNMF, LP-FNMTF and CFOND on Synthetic Data (Each column in Fig. 6 corresponds to one method). Fig. 6 shows that CFOND and iTopicModel have the best clustering results, and their outputs are mostly close to the true distributions. iTopicModel is a Bayesian-based method and has shown good performance on the synthetic network. Indeed, the synthetic networks and the noise in the network are generated following given distributions which can be better fitted by Bayesian-based methods. In Section 5.5, we will further compare CFOND and iTopicModel on real-world networks.

Because GNMF does not consider feature-feature correlations, its clustering results are mainly influenced by the structure information (or data manifold). For DRCC and LP-FNMTF, although both methods are claimed to consider data and feature manifold, they use strong constraints to force co-clustering results to be consistent with the manifolds. Their results are severely deteriorated when topology structures and feature distributions are inconsistent (which are common for real-world networks).

5.2 Experimental Results on Real Networks

Node Clustering Results: For each comparison method (including CFOND), we repeat clustering 50 times for each data set, and calculate the average clustering results. We report the best average result with optimal parameters for each method on six data sets in Table 3.

The results in Table 3 show that CFOND consistently outperforms other methods, with noticeable performance gain, which demonstrate its advantage in terms of clustering performance. A more careful examination on the results shows that, the co-clustering methods, including GNMF, DDRC, and LP-FNMTF methods, somehow exploit the geometric structures in data or feature spaces and generally achieve better clustering results comparing with traditional clustering methods, like k-means and NMF, in some data sets. In addition, we observed that iTopicModel performs very well on Synthetic and Cora data, but its performance on other data is inferior to GNMF, DRCC, LP-FNMTF, and CFOND with quite significant loss. This suggests that iTopicModel is likely sensitive to feature distributions and noise distributions. In Section 5.5, we will further investigate iTopicModel's performance *w.r.t.* different network characteristics.

Indeed, CFOND considers instance-feature, instance-instance, feature-feature as three separated relationships, and simultaneously carries out factorization on each relationship to ensure that factorization can best capture data distributions *w.r.t.* the underlying relationship. This is essentially better than using regularization terms (such as GNMF, DRCC, and LP-FNMTF do), because a regularization can only restrict a solution but cannot discover new solutions.

Feature Clustering Results: Because there is no feature cluster ground truth, we list Top-20 keywords of each topics of our results on PubMed data set and compared them with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2846555, IEEE Transactions on Knowledge and Data Engineering

TABLE 3
Clustering results on instances measured by accuracy/NMI of the compared methods.

Data Sets	Kmeans (ACC-NMI)	NMF (ACC-NMI)	iTopicModel (ACC-NMI)	GNMF (ACC-NMI)	DRCC (ACC-NMI)	LP-FNMTF (ACC-NMI)	CFOND (ACC-NMI)
Attack1	45.33%-0.2185	44.03%-0.2055	41.05%-0.1796	47.07%-0.2371	49.60%-0.2508	49.65%-0.2176	**68.36%-0.4693**
Attack2	45.42%-0.2243	43.45%-0.2036	40.10%-0.1627	47.85%-0.2339	49.71%-0.2541	45.63%-0.2191	**70.07%-0.5046**
Cora	34.90%-0.1609	33.56%-0.1351	47.33%-0.3014	39.07%-0.1719	42.71%-0.2198	28.61%-0.0261	**54.91%-0.3425**
CiteSeer	47.15%-0.2289	40.10%-0.1567	48.59%-0.2302	49.93%-0.2471	55.12%-0.2852	23.27%-0.0143	**56.34%-0.3696**
PubMed	56.99%-0.2451	60.17%-0.2506	55.78%-0.2367	53.90%-0.1531	61.75%-0.2618	54.37%-0.1532	**64.14%-0.4550**
Synthetic	49.55%-0.3599	49.37%-0.3479	66.38%-0.3927	50.85%-0.4025	50.48%-0.3516	55.12%-0.3756	**68.65%-0.4103**

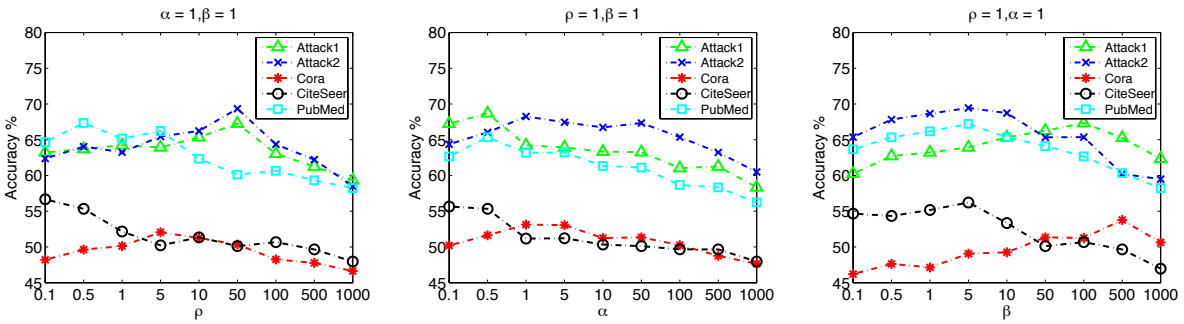


Fig. 7. Clustering results of using CFOND on five real-world data sets respect to changing parameter values.

TABLE 4
The comparisons of Top-20 features (words) for each topic in the PubMed network using CFOND, NMI, and RELIEF, respectively.

Topic a			Topic b			Topic c		
CFOND	**NMI**	**RELIEF**	**CFOND**	**NMI**	**RELIEF**	**CFOND**	**NMI**	**RELIEF**
Patient	*syndrom*	*develop*	*rat*	*cell*	*glucose*	*group*	*group*	*group*
type	*Patient*	*type*	*cell*	*glucose*	*insulin*	*children*	*children*	*male*
iddm	*type*	*Patient*	*mice*	*acid*	*inhibit*	*subject*	*male*	*subject*
gene	*develop*	*mass*	*glucose*	*rat*	*acid*	*niddm*	*min*	*plasma*
develop	*care*	*syndrom*	*islet*	*beta*	*antibodies*	*plasma*	*subject*	*correl*
disease	*associate*	*age*	*control*	*increase*	*nod*	*rate*	*year*	*factor*
associate	*import*	*care*	*active*	*impair*	*mice*	*compar*	*fat*	*sex*
age	*low*	*active*	*increase*	*express*	*resist*	*year*	*factor*	*year*
risk	*mellitus*	*risk*	*nod*	*mice*	*hyperglycemia*	*dure*	*higher*	*ml*
factor	*data*	*low*	*express*	*produce*	*kidney*	*nondiabetic*	*heart*	*rate*
differ	*marker*	*associate*	*protein*	*transport*	*depress*	*serum*	*ratio*	*similar*
use	*clinic*	*direct*	*animal*	*animal*	*liver*	*albuminuria*	*baselin*	*excret*
mellitus	*detect*	*insulindepend*	*effect*	*antibodies*	*increase*	*weight*	*index*	*mean*
function	*high*	*data*	*antibodies*	*depress*	*data*	*heart*	*ml*	*albumin*
excret	*gene*	*marker*	*kidney*	*resist*	*pathway*	*albumin*	*preval*	*niddm*
clinic	*provid*	*gene*	*reduce*	*nod*	*year*	*hyperglycemia*	*bmi*	*children*
relate	*risk*	*aim*	*significant*	*year*	*active*	*ratio*	*similar*	*detect*
studi	*direct*	*screen*	*inhibit*	*active*	*transport*	*adolesc*	*mm*	*reduce*
high	*aim*	iddm	*human*	*significant*	*rat*	*cpeptid*	*sex*	*bmi*
data	*studi*	*inject*	*response*	*inject*	*cell*	*baselin*	*excret*	*heart*
intervention	*differ*	*provid*	*insulin*	*data*	*express*	*mass*	*rate*	*onset*

the Top-20 words selected by using Normalized Mutual Information (NMI) and RELIEF algorithm in Table 4.

NMI is a commonly used measure to compare feature selection methods [34]. In our case, the normalized mutual information of two discrete random variables: the distribution of feature (keyword) i among nodes (instances) $\mathbf{x}_{\cdot i} \in \mathbb{R}_+^d$ and the **one-against-all ground-truth labelling** of group k, $\mathbf{y}_{\cdot k} \in \{0,1\}_+^d$, is defined as follows:

$$NMI(\mathbf{x}_{\cdot i}, \mathbf{y}_{\cdot k}) = \frac{\mathbb{I}(\mathbf{x}_{\cdot i}, \mathbf{y}_{\cdot k})}{[\mathbb{H}(\mathbf{x}_{\cdot i}), \mathbb{H}(\mathbf{y}_{\cdot k})]/2}, \tag{12}$$

where $\mathbb{I}$ is the mutual information function and $\mathbb{H}$ is the entropy function. Because $\mathbf{y}_{\cdot k}$ denotes the ground-truth labels of topic k, for each topic, the Top-20 keywords listed in Table 4 represent the most distinguished features selected by using NMI.

RELIEF is a feature selection algorithm used in binary classification (applicable to polynomial classification by decomposition into a number of binary problems) [35]. Similar to NMI method in our case, RELIEF repetitively calculates the weights of features. At each iteration, it considers the feature vector $\mathbf{X}$ of one random instance, and the feature vectors of the instance closest to $\mathbf{X}$ (by Euclidean distance) from each class. The closest same-class instance is called 'near-hit', and the closest different-class instance is called 'near-miss'. The weight vector of the feature is updated as

follows:

$$\mathbf{W}_i = \mathbf{W}_i - (\mathbf{x}_{\cdot i} - \mathbf{nearHit}_i)^2 + (\mathbf{x}_{\cdot i} - \mathbf{nearMiss}_i)^2 \quad (13)$$

Table 4 shows that there are a fair share of overlapping (colored words) between top features selected by using CFOND and NMI (12/20 for Topic a, 11/20 for Topic b and 8/20 for Topic c), which means clustering tasks carried on the data and features are strongly correlated and clearly not independent. In addition, feature clustering results also match to the three instance clusters with very good correspondence. Similarly, the overlapping of top features selected by using CFOND and RELIEF is high as well: 9/20 for Topic a, 12/20 for Topic b and 9/20 for Topic c.

Parameter Analysis: For our CFOND method, we have three parameters in Eq. (5), where α and β are regularization parameters to balance each factorization part, and ρ trade-offs the consistent degree. Fig. (7) shows that parameter values have different effect on real-world data sets. For Attack data set, a relatively larger constraint on ($\mathbf{G}$, $\mathbf{G}_s$) and ($\mathbf{F}$, $\mathbf{F}_s$) is needed, $\mathbf{G}$ should be close to $\mathbf{G}_s$, and $\mathbf{F}$ should be close to $\mathbf{F}_s$ in Eq. (5) to achieve high clustering performance. While for CiteSeer data set, the constraint should not be too strong. This is because the nodes' feature distances are not always consistent with the topology distances. Because CFOND is an iterative co-clustering method using feature and node cluster results in each iteration to improve the co-clustering results on the next iteration, the regularisation parameters α and β also effect algorithm performance. The choose of regularisation parameters are based on users' preference on whether to focus on node clustering results or feature clustering results.

5.3 Convergence and efficiency Analysis

We also report the convergence analysis by setting the number of iterations to 80 for each method, with optimal parameter setting for each data set.

Because the updating rules of minimizing the objective function for CFOND are iterative, we need to show that these rules are indeed empirically convergent. In order to investigate the actual convergence performance of these rules, we report the convergence curves of all state-of-the-art co-clustering methods (CFOND, DRCC, GNMF and LP-FNMTF) and one topic model method (iTopicModel) on all the five real-world data sets in Fig. (8), where the y-axis is the normalized value of the objective function and the x-axis denotes the iteration number. The results in Fig. (8) show that the multiplicative updating rules for both CFOND and LP-FNMTF converge very fast, usually within 20 iterations.

In addition, we also report the average convergence time of the compared iteration-based methods on real-world data sets in Fig. 9. From the results, we can observe that CFOND is only slightly slower than LP-FNMTF, but is much faster than all other state-of-the-art co-clustering methods. This is mainly because LP-FNMTF constrains factor matrices of NMF to be cluster indicator matrices, therefore requires much less matrix multiplications. GNMF and DRCC require much more iterations in order to reach convergence, and are therefore more time-consuming. If we take the clustering results in Table 3 and the runtime performance in Fig. 9 into consideration, CFOND demonstrates clear advantage for co-clustering on large-scale data.

The runtime performance in Fig. 9 shows that iTopicModel is the most time-consuming method, and its convergence curves on real-world data are not as smooth as others. This is because iTopicModel is an EM-based method which requires the more iterative times, and much more space-time consumption for each iteration. As a result, its execution speed and convergence spped are slow as shown in Fig. 8, comparing to other NMF/NMTF based methods.

5.4 The Relationship Between Node Clusters and Feature Clusters

An inherent advantage of CFOND is that the latent matrix $\mathbf{S}$ in consensus factorisation function Eq. 5 can also reveal the corresponding relationship between the node clustering and feature clustering results. In this subsection, we report the results of latent matrix $\mathbf{S}$ on PubMed dataset (Table 5) and Synthetic network (Table 6).

As we have described in Section 2, $\mathbf{S}_{ij}$ represents the relative weight between feature cluster i and node cluster j. In Tables 5 and 6, we use best matching method to evaluate the accuracy results. Based on the best matching principle, for PubMed dataset, node clusters 1, 2, 3 by using CFOND are corresponding with real classes "Diabetes Mellitus Type 1", "Diabetes Mellitus Type 2", and "Diabetes Mellitus, Experimental", respectively. Feature clusters 1, 2, 3 are shown as Topics (a), (b), (c) in Table 4.

From Table 5, we can see that node cluster 1 is most related to feature cluster 2 ($\mathbf{S}_{12}$), node cluster 2 is the closest to feature cluster 1 ($\mathbf{S}_{21}$), and node cluster 3 is most related to feature cluster 3. This is, in fact, consistent with the data domain characteristics. For example, "Diabetes Mellitus, Experimental" studies wet lab diabetes mellitus models used for different experiments, such as the type of procedures used to cause a lab animal, such as a mice, becoming a diabetes test bed. Because this category is closely related to the lab and experiments, the words "plasma" and "non-diabet(ic)" etc. are representative words to this cluster, so Topic *c* is matched to the "Diabetes Mellitus, Experimental". Therefore, the larger the $\mathbf{S}_{ij}$ value, the higher the correlation between node cluster i and feature cluster j is. Similar conclusion can also be derived from synthetic network in Table 6.

TABLE 5
The latent matrix $\mathbf{S}$ on PubMed Network ($\mathbf{S} \in \mathbb{R}^{c \times k}$ where $c = 3$ and $k = 3$ representing the number of node clusters (rows) and the number of feature clusters (columns), respectively).

5.67E-03	0.008456	0.000812
0.028237	1.70E-05	0.001195
0.000415	0.000826	0.031455

TABLE 6
The latent matrix $\mathbf{S}$ on Synthetic Network ($\mathbf{S} \in \mathbb{R}^{c \times k}$ where $c = 4$ and $k = 4$ representing the number of node clusters (rows) and the number of feature clusters (columns), respectively).

0.64732	0.00273	0.02784	0.00673
0.03654	0.00876	0.59272	0.00219
0.00145	0.29562	0.07365	0.00227
0.03497	0.01162	0.00758	0.39654

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2846555, IEEE Transactions on Knowledge and Data Engineering

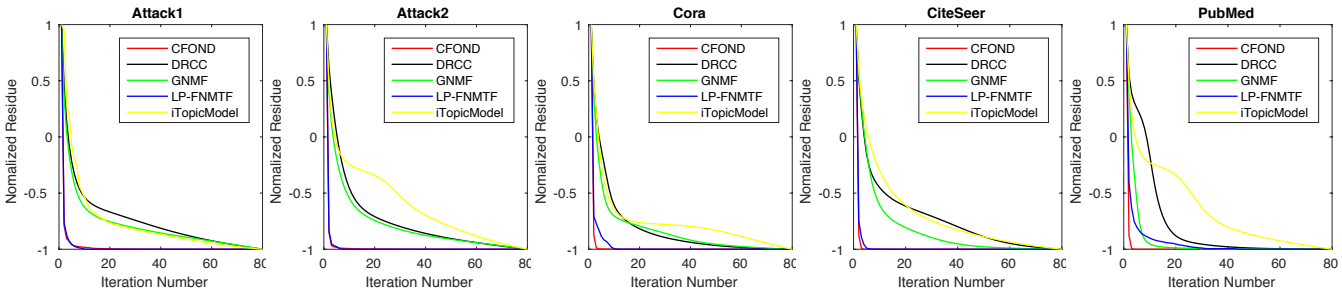


Fig. 8. Convergence comparisons of different co-clustering methods on real-world data sets. The $x-$axis denotes the number of iterations, and the $y-$axis denotes the normalized residue of the objective function.

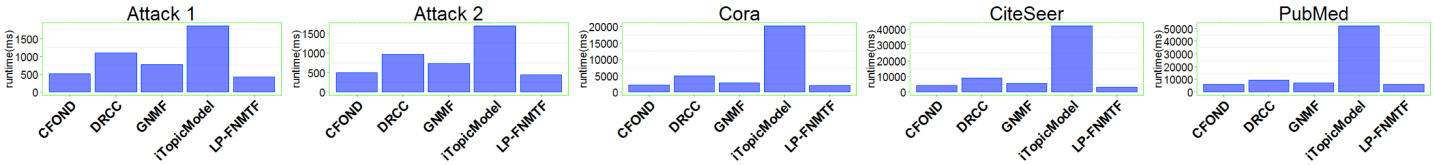
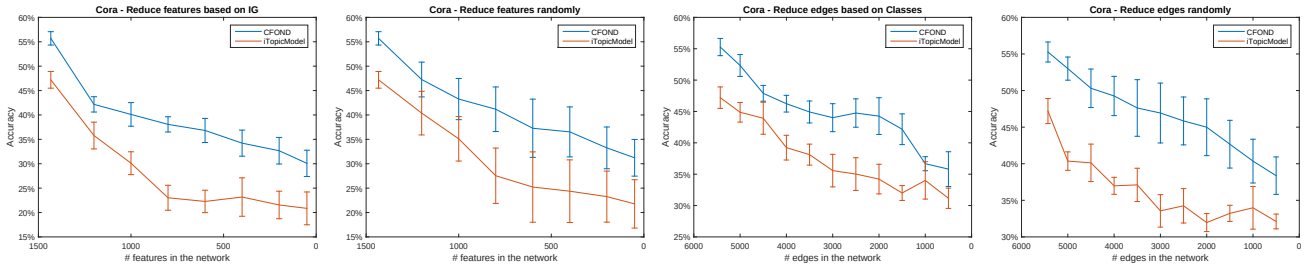


Fig. 9. Runtime comparisons of different co-clustering methods on real-world data sets.



(a) (b) (c) (d)

Fig. 10. Case study of CFOND *vs.* iTopicModel on networks with various degree of consistency between node content and topology structures (a) and (b), and networks with different degree of edge density (c) and (d).

5.5 Case Study

In this section, we further compare Bayesian-based method (iTopicModel) and NMF/NMTF based method (CFOND) on the Cora data set, by varying the network characteristics. Our purpose is to observe how do iTopicModel and CFOND behave (1) for networks with different edge density, and (2) for networks with various degree of consistency between node content and topology structures.

To generate networks with various consistency between node content and topology structures, we sort all node features in a descending order according to their Information Gain (IG) scores. In Fig. 10(a), we continuously remove features based on their IG scores from high to small and generate networks whose node content is less and less consistent to structures. In comparison, we also randomly remove the same number of node features and report the results in Fig. 10(b).

To generate networks with different edge density, we gradually reduce edges between nodes in the same class, followed by removing edges between nodes in different classes, and report the results in Fig. 10(c). This will help generate networks with less and less edge density (so topology structures is playing less and less important role). In comparison, we also randomly remove the same number of edges and report the results in Fig. 10(d).

The results from Fig. 10 show that with the reduction of node features and edges, iTopicModel's performance deteriorate dramatically. It is more sensitive to node features rather than edges, because comparing to feature reduction *vs.* edge reduction, the former results in a larger performance loss. In comparison, CFOND's performance is relatively balanced between features and edges.

6 Related Work

In traditional clustering, the aim is to divide an unlabeled data set into groups of similar data points. This can be achieved by comparing feature based similarities/distances between instance pairs, and assigning each instance to the group mostly similar to. k-means [36] is the classical clustering method which follows the traditional clustering principle. From a geometrical point of view, a data set can be seen as a set of nodes connected with structure relationships, and clustering aims to finding intrinsic groups of the data. Spectral clustering [37], [38], [39] and Non-negative Matrix Factorization(NMF) [29] are typical methods which carry out clustering from the geometrical point of view. Some studies have also been proposed to combine traditional clustering and geometrical relationships between instances for better clustering results (commonly referred to as attributed graph clustering [40], [41], [42]).

The above clustering methods mainly focus on one-side clustering. In other words, clustering is based on the similarities along either the feature or the structure relationships, respectively. Motivated by the duality between data points (*e.g.* documents) and features (*e.g.* words), several co-clustering algorithms have been proposed to cluster data based on their distributions in the feature space, as well as cluster features into groups by using their distributions in

the sample space. Such two-side co-clustering approaches have demonstrated better performance than traditional one-side clustering. For example, [43] employs a bipartite spectral graph partition approach to co-cluster words and documents, which requires that each document cluster is associated to a word cluster (which is a rather restrictive constraint). To overcome this drawback, [13] presents a co-clustering algorithm that monotonically increases the preserved mutual information by intertwining both the row and column clusterings at all stages, which is an information theoretic method and can be seen as the extension of information bottleneck method [44] to two-side clustering. Alternatively, factorization based approaches are also used to factorize an instance-feature matrix into instance and feature groups respectively [15], [16]. [14] proposed an orthogonal nonnegative matrix tri-factorization (ONMTF) to co-cluster words and documents, with sound mathematical form and encouraging performance.

Recently, several studies have shown that many real-world data are actually sampled from an intrinsic network structure [1], [45], [46], in which linkages provide useful information for clustering. Co-clustering algorithms [18], [19] try to build instance-instance nearest neighbours graph and enforce the k-NN graph in the objective function to discover cluster structures with respect to low dimensional feature space (*i.e.*, manifold). However, as we have elaborated in Section 1, these methods are ineffective for co-clustering networked data, mainly because k-NN graphs have different characteristics from real-world networks, and the topology of k-NN graphs are consistent with the node similarity assessed in the feature space. The unique characteristics of real-world networks and the inconsistency of the node content and topology structure of the networks suggest that existing manifold based co-clustering methods [18], [19] are ineffective for networked data.

Our work is also related to relational topic models. Jonathan and David developed a relational topic model (RTM) which is a model of documents and links between them [47], with a hierarchical model of links and node attributes. However, RTM models nodes and links separately and therefore results in information loss. In [31] and [48], the inter-dependence of a set of high-level topics and the documents are considered to develop a Bayesian hierarchical approach. Unfortunately, Bayesian-based methods are too sensitive to samples selected and therefore often lead to sensitive result as shown in Section 5.

Although the factorization framework employed in CFOND is similar to the factorization approach in [18], [19], the differences between CFOND and existing works are fundamental: (1) CFOND considers three-factor relationship matrices, instance-feature, instance-instance, and feature-feature, in the factorization framework, whereas existing methods [18], [19] only consider two-factor relationships (instance-feature and instance-instance relationships). The integration of feature-feature matrix in the factorization allows CFOND to explicitly capture feature-to-feature relationships for finding optimal feature clustering results; and (2) CFOND employs a consensus factorization principle where three relationship matrices are factorized simultaneously, conditioned by the consensus objective function, whereas existing methods [18], [19] only factorizes one (instance-feature) matrix and uses other relationships as hard constraints. The three independent factorizations in CFOND provide maximum degree of freedom for CFOND to explore solutions best fit for each individual relationship matrix, and the consensus factorization further ensures that the solutions are consistent across all relationship matrices for optimal clustering results.

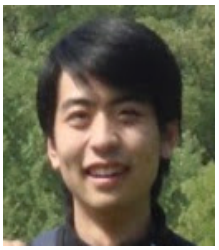
7 Conclusion

In this paper, we proposed a consensus factorization based method, CFOND, to simultaneously cluster networked instances (nodes) and features which represent node content in the network. CFOND is rooted on NMF/NMTF based co-clustering, but has its uniqueness in (1) leveraging auxiliary information in networked data for simultaneous factorization of three types of relationships: Instance-feature, instance-instance, feature-feature; and (2) enforcing the consensus of the factorized results for optimal clustering results. Compared to existing proximity graph regularization based methods, the consensus factorization ensures that the final cluster structures are consistent across information from different types of relationships, and therefore results in minimum errors. Theoretical analysis confirms the convergence of the solutions derived from CFOND. Extensive experiments and comparisons on benchmark data sets demonstrate that CFOND consistently outperforms baseline approaches for co-clustering networked data.

References

[1] J. McAuley and J. Leskovec, "Image labelling on a network: using social-network metadata for image classification," in *ECCV*, vol. 4, 2012, pp. 828–841.
[2] H. T. Shen, Y. F. Shu, and B. Yu, "Efficient semantic-based content search in p2p network," *IEEE Trans. on Knowledge and Data Eng.*, vol. 16, no. 7, pp. 813–826, 2004.
[3] M. Li, Y. Liu, and L. Chen, "Nonthreshold-based event detection for 3D environment monitoring in sensor networks," *IEEE Trans. on Knowledge and Data Eng.*, vol. 20, no. 12, pp. 1699–1711, 2008.
[4] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang, "Tri-party deep network representation," in *IJCAI*, 2016, pp. 1895–1901.
[5] C. Wang, S. Pan, S.-F. Fung, C. P. Yu, X. Zhu, and J. Jiang, "MGAE: Marginalized graph autoencoder for graph clustering," in *CIKM*, 2017.
[6] K.-I. Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and A.-L. Barábasi, "The human disease network," in *PNAS*, vol. 104, no. 21, 2007, pp. 8685–8690.
[7] M. Pellegrini, D. Haynor, and J. M. Johnson, "Protein interaction networks," *Expert Review of Proteomics*, vol. 1, no. 2, pp. 89–99, 2004.
[8] J. Ji, A. Zhang, C. Liu, and X. Quan, "Survey: Functional module detection from protein-protein interaction networks," *IEEE Trans. on Knowledge and Data Eng.*, vol. 26, no. 2, pp. 261–277, 2014.
[9] F. Spezzano, V. S. Subrahmanian, and A. Mannes, "Reshaping terrorist networks," *Communications of the ACM (CACM)*, vol. 57, no. 8, pp. 60–69, 2014.
[10] L. Cao, Y. Zhao, and C. Zhang, "Mining impact-targeted activity patterns in imbalanced data," *IEEE Trans. on Knowledge and Data Eng.*, vol. 20, no. 8, pp. 1053–1066, 2008.
[11] B. Mirkin, Ed., *Mathematical classification and clustering: From how to what and why*. Berlin Heidelberg: Springer, 1998, vol. 1.
[12] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *SIGKDD*, 2001, pp. 269–274.
[13] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic coclustering," in *SIGKDD*, 2003, pp. 89–98.
[14] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *SIGKDD*, 2006, pp. 126–135.
[15] C. Ding and T. Li, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, 2010.

[16] L. Du and Y.-D. Shen, "Towards robust co-clustering," in *IJCAI*, 2013, pp. 1317–1322.
[17] R. Reagans and B. McEvily, "Network structure and knowledge transfer: The effects of cohesion and range," *ASQ*, vol. 48, no. 2, pp. 240–267, 2003.
[18] Q. Gu and J. Zhou, "Co-clustering on manifolds," in *SIGKDD*, 2009, pp. 359–368.
[19] D. Kuang, C. Ding, and H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *ICDM*, vol. 12, 2012, pp. 106–117.
[20] M. Maier, M. Hein, and U. von Luxburg, "Cluster identification in nearest-neighbor graphs," *Algorithmic Learning Theory (ALT)*, pp. 196–210, 2007.
[21] M. L. Goldstein, S. A. Morris, and G. G. Yen, "Problems with fitting to the power-law distribution," *The European Physical Journal B-Condensed Matter and Complex Systems (EPJ B)*, vol. 41, no. 2, pp. 255–258, 2004.
[22] D. Cai, X. He, X. Wu, and J. Han, "Non-negative matrix factorization on manifold," in *ICDM*, 2008, pp. 63–72.
[23] H. Wang, F. Nie, H. Huang, and F. Makedon, "Fast nonnegative matrix tri-factorization for large-scale data co-clustering," in *JCAI*, 2011, pp. 1553–1558.
[24] N. Mishra, R. Schreiber, I. Stanton, and R. E. Tarjan, "Clustering social networks," *Algorithms and Models for the Web-Graph*, pp. 56–67, 2007.
[25] J. J. Whang, D. F. Gleich, and I. S. Dhillon, "Overlapping community detection using seed set expansion," in *CIKM*, 2013, pp. 2099–2108.
[26] G. Golub and C. V. Loan, Eds., *Matrix computations 4th edition*. Baltimore Maryland: Johns Hopkins University Press, 2013, vol. 3.
[27] X. Niyogi, "Locality preserving projections," in *NIPS*, vol. 16, 2004, p. 153.
[28] D. G. Luenberger, Ed., *Linear and nonlinear programming*. Springer, 2008, vol. 116.
[29] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *NIPS*, vol. 13, pp. 556–562, 2001.
[30] C. J. Lin, "On the convergence of multiplicative update algorithms for nonnegative matrix factorization," *IEEE TNN*, vol. 18, no. 6, pp. 1589–1596, 2007.
[31] Y. Sun, J. Han, J. Gao, and Y. Yu, "iTopicModel: Information network-integrated topic modeling," in *ICDM*, 2009, pp. 493–502.
[32] J.MacQueen, "Some methods for classification and analysis of multivariate observations," pp. 281–297, 1967.
[33] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.
[34] R. Cilibrasi and P. Vitanyi, "Clustering by compression," *IEEE Transactions on Information theory*, vol. 51, no. 4, pp. 1523–1545, 2005.
[35] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *AAAI*, 1992, pp. 129–134.
[36] C. M. Bishop, Ed., *Pattern recognition and machine learning*. New York, USA: Springer, 2006, vol. 1.
[37] U. von Luxburg, "A tutorial on spectral clustering," in *Statistics and Computing*, vol. 17, no. 4, 2007, pp. 395–416.
[38] J. Shi and J. Malik, "Normalised cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
[39] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *NIPS*, vol. 2, pp. 849–856, 2001.
[40] H. Cheng, Y. Zhou, and J. X. Yu, "Clustering large attributed graphs: A balance between structural and attribute similarities," *ACM Trans. on Knowledge Discovery from Data*, vol. 5, no. 2, 2011.
[41] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng, "A model-based approach to attributed graph clustering," in *SIGKDD*, 2012, pp. 505–516.
[42] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," in *VLDB*, vol. 2, no. 1, 2009, pp. 718–729.
[43] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *SIGKDD*, 2001, pp. 269–274.
[44] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *arXiv preprint physics/0004057*, 2000.
[45] J. Staiano, B. Lepri, N. Aharony, F. Pianesi, N. Sebe, and A. Pentland, "Friends don't lie—inferring personality traits from social network structure," in *Ubicomp*, 2012, pp. 321–330.
[46] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *ACSAC*, 2010, pp. 1–9.
[47] J. Chang and D. M. Blei, "Relational topic models for document networks," in *AISTATS*, 2009, pp. 81–88.
[48] Y. Liu and A. Niculescu-Mizil, "Topic-link lda:joint models of topic and author community," in *ICML*, 2009, pp. 665–672.



Ting Guo received the master's degree in computer science from the Jilin University (JLU), Jilin, China, in 2012. Since February 2012, 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.

Shirui Pan received his PhD degree in computer science from University of Technology Sydney (UTS), Australia. He is a Research Fellow in the Centre for Artificial Intelligence (CAI), UTS. His research interests include data mining and machine learning. To date, Dr. Pan has published over 35 research papers in top-tier journals and conferences, including the IEEE Transactions on Knowledge and Data Engineering, the IEEE Transactions on Cybernetics, and IEEE International Conference on Data Engineering.

Xingquan Zhu received the PhD degree in computer science from Fudan University, Shanghai, China. He is an associate professor in the Department of Computer & Electrical Engineering and Computer Science, Florida Atlantic University, and a Distinguished Visiting Professor (Eastern Scholar) at Shanghai Institutions of Higher Learning. His research interests mainly include data analytics, machine learning, and bioinformatics. Since 2000, he has published more than 230 refereed journal and conference papers in these areas, including two Best Paper Awards and one Best Student Paper Award. He is an associate editor of the IEEE Trans. on Knowledge and Data Engineering (2008-2012, 2014-date).

Chengqi Zhang received the PhD degree from the University of Queensland, Brisbane, Australia, in 1991 and the DSc degree (higher doctorate) from Deakin University, Geelong, Australia, in 2002. Since December 2001, he has been a professor of information technology with the University of Technology, Sydney (UTS), Sydney, Australia, where he has been the director of the UTS Priority Investment Research Centre for Quantum Computation and Intelligent Systems since April 2008. Since November 2005, he has been the chairman of the Australian Computer Society National Committee for Artificial Intelligence. He has published more than 200 research papers, and has attracted 11 Australian Research Council grants. He was as an associate editor for IEEE Trans. on Knowledge and Data Engineering (2005-2008); and served as the general chair, PC chair, or organising chair for five international Conferences including KDD 2015, ICDM 2010, and WI/IAT 2008.