Neurocomputing 565 (2024) 126989

Contents lists available at ScienceDirect

Neurocomputing

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

Overlapping community detection using expansion with contraction

Zhijian Zhuo, Bilian Chen *, Shenbao Yu, Langcai Cao

Department of Automation, Xiamen University, 361005, China
Xiamen Key Laboratory of Big Data Intelligent Analysis and Decision-making, Xiamen, 361005, China

ARTICLE INFO

Communicated by W.K. Wong

Keywords:
Community detection
Overlapping communities
Non-negative matrix factorization
Expansion and contraction

ABSTRACT

Numerous disjoint community detection methods have reached the state-of-the-art. Some overlapping community detection methods have been proposed in recent years, but they lack the ability to adjust the degree of overlap while maintaining detection quality. To well handle this issue, we in this paper propose a novel method, namely expansion with contraction method for overlapping community detection (ECOCD). Specifically, ECOCD obtains the disjoint communities through non-negative matrix factorization and proceeds to expansion with contraction process (including the expansion process and the contraction process). In each iteration of the process, we randomly select a community and then continuously conduct the expansion and contraction processes on this community. The former process absorbs nodes by the degree of affiliation that is newly defined, while the latter removes nodes by permanence. Moreover, we theoretically analyze the computational complexity of ECOCD. The advantage of ECOCD is that it is applicable to various networks with different properties by adjusting the degree of overlap, and enjoys high quality of overlapping community detection as well. Our experiments on both synthetic and real-world networks further verify this. Extensive experiments show that ECOCD is superior to the eleven state-of-the-art overlapping community detection methods in terms of four metrics, validating the effectiveness, efficiency and robustness of ECOCD.

1. Introduction

Benefiting from the explosive increase and variety of information resources, online social networks have been growing rapidly in recent years [1]. Community, usually defined as internal nodes in a social network that are tightly inter-connected and weakly connected to other external communities [2], provides a good opportunity for assessing valuable information concealed behind the social network. As a method for identifying the structure of large-scale network data sets based on graph theory, community detection aims to not only reveal a network's original graph with a macroscopic perspective but also uncover more semantic knowledge [3]. Not surprisingly, community detection has attracted great interest in network science.

Generally, previous approaches for the community detection problem have differed along two dimensions: disjoint community detection and overlapping community detection [4]. The disjoint community detection algorithms include spectral clustering [5], point-wise mutual information [6], graph attention auto-encoder [7] and mean path length [8]. Nowadays, the scale and complexity of networks hugely increase, leading to multiple relationships of nodes in the network, such as social networks, collaborative networks and biological neural networks [2]. Moreover, the degree of overlap varies in different networks with different properties. It thus becomes indispensable to learn the overlapping structure of networks. Through the efforts of scholars in recent years, there has been an increasing amount of literature on detecting overlapping communities, such as seed expansion [9], dynamics [10], and label propagation [11]. However, most existing algorithms do not support adjusting the degree of overlap while maintaining the quality of overlapping community detection.

Luckily, techniques of disjoint community detection are greatly developed, which shape pathways and thus provide guidance for discovering overlapping communities. In this paper, we focus on the overlapping community detection problem by extending the off-the-shelf disjoint community detection solution. Specifically, we first obtain high-quality disjoint communities and then design rules to enable these communities to absorb and remove certain nodes.

Our overlapping community detection method. We propose an expansion with contraction method for overlapping community detection (ECOCD) that provides the ability to adjust the degree of overlap, enjoying the virtue of good performance in networks with different overlap degrees. The proposed ECOCD method works through

* Corresponding author at: Department of Automation, Xiamen University, 361005, China.
E-mail addresses: 23220211151697@stu.xmu.edu.cn (Z. Zhuo), blchen@xmu.edu.cn (B. Chen), yushenbao@stu.xmu.edu.cn (S. Yu), langcai@xmu.edu.cn (L. Cao).

https://doi.org/10.1016/j.neucom.2023.126989
Received 1 February 2023; Received in revised form 27 July 2023; Accepted 30 October 2023
Available online 2 November 2023

a two-stage framework, i.e., the discovery of core communities and the expansion with contraction process.

The discovery of core communities. We employ non-negative matrix factorization (NMF) to get disjoint communities, which serve as the core communities.

The expansion with contraction process. After obtaining the core communities, ECOCD proceeds to multiple iterations. In each iteration, we first randomly select an overlapping community and then perform the expansion and contraction processes on this community. Finally, once the overlap degree of all communities satisfies the pre-defined overlap threshold, we return the final overlapping communities.

The goal of each cycle of the expansion process is to absorb a node with the highest degree of affiliation for a randomly selected community. To realize this goal, we design rigorous rules on the basis of the similarity, the distance and the factors decomposed by NMF to select nodes (see Section 4). Subsequently, we proceed to the contraction process, whose goal of each cycle is to remove the most alienated node from the community. The degree of alienation of each node is measured by permanence [12].

Summary of the evaluation. We evaluate the performance of the proposed ECOCD method on both synthetic networks and real-world networks. And four evaluation metrics and eleven state-of-the-art overlapping community detection algorithms are used to verify the effectiveness of ECOCD (see Section 5). Moreover, we compare the running time to measure the efficiency of the algorithms. Through these experiments, we summarize the performance of ECOCD as follows.

- ECOCD is well suited for high-overlap situations and outperforms other algorithms in most high-overlap synthetic networks in terms of the overlapping normalized mutual information [13] (see Fig. 2).
- ECOCD also performs well in low-overlap real-world networks in terms of the extended overlapping modularity [14], the modified permanence and conductance [15] (see Tables 4–6).
- The overall running time of ECOCD is less than the compared matrix computation-based algorithms, showing the high efficiency of ECOCD (see Fig. 9).

Contributions of the paper. The contributions of this article mainly include three aspects:

- ECOCD is able to adjust the degree of overlap to accommodate various detection requirements or network properties.
- We innovatively propose a rigorous selection process for selecting which nodes are absorbed into the community and which nodes remain in the community, taking into account not only the nodes that are locally appropriate for the community (see Eq. (1)), but also the characteristics of the community structure (see Eq. (2)).
- The proposed expansion with contraction process can be applied to any existing disjoint communities to discover overlapping communities with certain modifications.

The remainder of this paper is organized as follows. We review related works in Section 2. Section 3 introduces the preliminary of this work. The working principle of the proposed ECOCD method is presented in Section 4. Section 5 conducts experiments on synthetic and real-world networks to test the effectiveness and efficiency of ECOCD. A summary of ECOCD and a discussion of possible future improvements are described in Section 6.

2. Related work

In this section, we investigate the existing methods for overlapping community detection. Previous techniques have differed along several dimensions, including probability-based methods, basic community structure-based approaches, seed expansion for community detection algorithms, and others.

Overlapping community detection using probabilistic models. Mixed-membership stochastic block model (MMSB) [16] is a type of variance allocation model for modeling pairwise relations between nodes. MMSB combines global parameters on instantiating densely connected blocks with local parameters on instantiating node-specific variability. Although MMSB can handle directed overlapping networks, it is not good at handling multiple types of node information. Bayesian non-negative matrix factorization (BNMF) [17] is another probabilistic model that also considers both global and local parameters, which has the characteristics of soft classification and intuitiveness. BNMF avoids the shortcomings of maximizing modularity [18] and has a macroscopic vision in detecting overlapping communities. However, BNMF can only handle static networks. Different from the above two models, the cluster affiliation model for big networks (BigClam) [19] seeks to build the cluster affiliation model for overlapping community detection. The goal of BigClam is to estimate the non-negative latent variables that represent the membership strength of nodes to communities. Since BigClam assumes that the overlaps between communities are densely connected, the overlapping communities detected by BigClam are different from most algorithms.

Detection of overlapping communities based on basic community structure. The clique percolation method (CPM) is simply based on the local topology of a network structure for overlapping community detection [20]. Detecting k-clique structures at various weight thresholds is the mechanism of CPM. CPM has lower time complexity, but it is only suitable for detecting subgraphs that are fully connected. Similar to CPM, Ego-splitting [21] is also looking for a basic community structure by leveraging the local ego-nets. Ego-splitting consists of two steps: in the first step, the authors use a partitioning algorithm to partition the nodes' ego-nets and then use the calculated clusters to split each node into its persona nodes; in the second step, Ego-splitting partitions the newly created graph to obtain overlapping communities. A local-first discovery method for overlapping communities (DEMON) [22] democratically lets each node vote for the surrounding community in its limited view of the global system. DEMON also uses ego-net and the label propagation algorithm to find the elementary communities. Although ego-net pioneers the microcosm to solve complex network community discovery problems, the community discovery algorithm based on ego-net has the problems of the heavy burden of merging communities and low accuracy.

Seed expansion-based overlapping community detection. Community detection using seed set expansion (SE) [23] adopts a novel seeding selection strategy for overlapping community detection, which expands seeds using the personalized PageRank [24] clustering procedure. As an improved version of SE, neighborhood-inflated seed expansion (NISE) [9] aims to find more excellent seeds and then expand them with a greedy strategy using community metrics. When selecting the nodes as seeds, the seed expansion method often adopts relatively less complicated strategies. This leads to the fact that the seed expansion method may select not pretty well nodes as seeds at the beginning, which affects the accuracy of the algorithm. Computing the influence of nodes is another unique way to find seeds, which requires a relatively large amount of calculation. For example, global and local node influence-based community detection (LGIEM) [25] innovatively invents a way to combine global information and local information to find community centers as seeds. LGIEM first determines the most central nodes as initial communities and then uses the expansion strategy to add the nodes to the initial communities. LGIEM designs a set of rigorous calculation formulas for selecting central nodes, but it does not consider community density. Unlike LGIEM, GREESE [26] first finds the most similar pair of nodes and then expands them by maximizing a local community fitness function. In addition to seed

selection, improving expansion strategies is also important. Label propagation algorithm with neighbor node influence (LPANNI) [27] has the authority on expansion strategy. LPANNI adopts fixed label propagation sequence based on node importance and a label update strategy based on neighbor node influence and historical label preference strategy. However, LPANNI works only on homogeneous networks and is not tailored for heterogeneous networks.

Other overlapping community detection algorithms. Due to the fact that the community is characterized as dense within and sparse outside, density-based community detection methods have considerable applicability. For example, the overlapping community detection algorithm based on density peaks (OCDDP) uses a similarity-based approach to set distances between nodes, which employs a three-step process to pick the cores of communities and membership vectors to represent the belongings of nodes [28]. OCDDP requires manual confirmation of maximum distance and density. In addition, based on the improved density peak clustering, Lu et al. [29] used the modified PageRank of nodes as the density indexes, and then drew the decision graph to discover the communities. By triple factorization of the adjacency matrix, bounded non-negative matrix tri-factorization (BNMTF) [30] obtains the strength matrix of the relationship between nodes and communities and the interaction matrix between communities. Moreover, the adjacency matrix can actually play a role in matrix manipulations as pseudo-supervision information. Discrete non-negative matrix factorization (DNMF) explores discriminative information in a pseudo-supervised manner [31]. Unlike NMF which requires post-processing, DNMF can directly obtain discrete community memberships. The time complexity of most steps of DNMF is $O(n^2)$, so DNMF takes more running time to process large networks. Fang and Lin [32] proposed to use Bayesian rules to generate a normalized community index matrix to fit the node similarity matrix, which is also similar to DNMF's pseudo-supervision for community discovery. Minimum-volume non-negative matrix factorization (MNMF) [33] uses a volume regularizer that can be used for a rank-deficient NMF. Based on the existing excellent disjoint community detection methods, it is necessary to do an integration. Ensemble-based overlapping community detection using disjoint community structures (EnCoD) is an algorithm that integrates multiple disjoint community detection algorithms [34]. EnCoD uses the results of disjoint community detection algorithms to establish a feature vector for each node and finds overlapping communities through the aggregation of feature vectors. EnCoD has high detection accuracy, but it needs to use multiple disjoint community detection models and the ordering of network node pairs. Recently, deep learning models have demonstrated their powerful learning capabilities. Shchur and Günnemann [35] proposed a model based on graph neural network (GNN) for overlapping community detection. The core idea of this method is to combine the powerful capabilities of GNN with the Bernoulli-Poisson probability model.

As can be seen from the above, overlapping community detection algorithms have developed in different directions. However, the detection accuracy based on the NMF algorithm is still in a dominant position. Thus, it is crucial to make full use of the factors decomposed by NMF. Combining NMF with the advantages of seed expansion can bring a relatively large imagination space for creating new methods.

3. Preliminaries

Before going into details about the ECOCD method, we first briefly describe the overlapping community detection problem. Next, we introduce NMF and several classic strategies for seed expansion.

3.1. Problem statement

A given network can be modeled as a graph $G = (V, E)$, where V denotes the node set and E represents the edge set. Let $n = |V|$ be the number of total nodes. For undirected and unweighted networks, G can be represented by its adjacency matrix $A \in \mathbb{R}_+^{n\times n}$ such that $A_{ij} = 1$ if there is an edge between node v_i and node v_j, and $A_{ij} = 0$ otherwise. Traditional community detection problem considers that each node can only belong to a community. The goal of traditional community detection is to divide a graph into k highly cohesive disjoint communities $c_1, \ldots, c_k$ that satisfy $c_1 \cup \cdots \cup c_k = V$. In real-world networks, the relationships between nodes are often complex and interrelated, which guides us to consider overlapping communities. In the overlapping community detection problem, a node is allowed to belong to multiple communities. Therefore, the task of overlapping community detection is also to find k highly cohesive communities, but some nodes are allowed to belong to multiple communities such that $c_1 \cup \cdots \cup c_k \subseteq V$.

3.2. Non-negative matrix factorization

NMF has strong interpretability and dimensionality reduction ability, which is appropriate for community detection. Hence, a number of NMF-based techniques have been developed for the decomposition of an adjacency matrix.

Given an adjacency matrix A, we can find the non-negative factors $W \in \mathbb{R}_+^{n\times k}$ and $H \in \mathbb{R}_+^{k\times n}$ such that $A \approx WH$. Usually, the square of the Euclidean distance is a very simple and effective way to measure the quality of the estimated factors. Lee and Seung [36] invented "multiplicative update rules" to achieve a balance between computational speed and implementation difficulty. In this paper, we will use this rule to update the factors of NMF. The update rules for W and H are as follows:

$$H \leftarrow H \frac{W^T A}{W^T W H} \qquad W \leftarrow W \frac{A H^T}{W H H^T}$$

Remark that the above update rules are sensitive to the initial values of W and H, thus the initialization of the two factors needs to design certain rules. Boutsidis and Gallopoulos [37] proposed a method called non-negative double singular value decomposition (NNDSVD) to enhance the initialization for NMF. NNDSVD is based on two SVD processes: create a rank-k approximation and then have an SVD on the positive parts of each of the factors. NNDSVD can be combined with existing NMF algorithms to enable follow-up NMF algorithms to greatly reduce the initial residual in a few iterations at a lower overall cost. Therefore, in this paper, NNDSVD will be used to initialize the factors of NMF for synthetic networks.

3.3. Seed expansion

There are many ways to find good seeds, the more classic ones are "Graclus centers" and "Spread hubs" [9]. The strategy of "Graclus centers" is to first use a high-quality and fast community detection algorithm to find disjoint communities, and then calculate the most central nodes in the disjoint communities. In contrast, the approach of "Spread hubs" is to find the nodes that have relatively high coverage after a few expansions. After acquiring seeds, expanding these seeds to build communities is the next task. The easiest way is to do random walks on these seeds to expand the communities. And the stationary distribution of random walks can be represented by using a personalized PageRank [38] vector. The main idea of the proposed ECOCD method is similar to seed expansion. However, in seed selection process, we do not select some sporadic seeds but directly select a few clusters. Besides, in the expansion process, we innovatively propose the contraction process to make the expansion of the community more rigorous.

4. The proposed expansion with contraction method for overlapping community detection

In this section, we describe in detail the entire process of the proposed ECOCD method, as well as its time complexity.

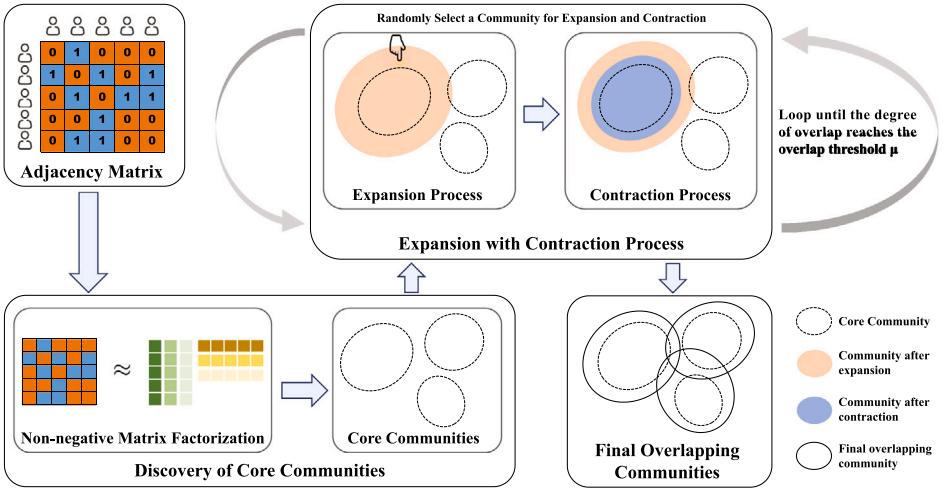


Fig. 1. The procedure of ECOCD method.

Table 1
The parameters of ECOCD method.

Parameter	Description
μ	The overlap threshold
k	The number of communities
k_s	The size of the candidate set
k_{ex}	The number of cycles of the expansion process
k_{con}	The number of cycles of the contraction process

4.1. The procedure of our method

Inspired by seed expansion, we propose the ECOCD method based on NMF. Fig. 1 graphically depicts the entire framework of the ECOCD method. As can be seen from the upper left corner of the figure, ECOCD starts from the adjacency matrix A. Subsequently, ECOCD enters two phases, one is the discovery of core communities, and the other one is the expansion with contraction process. In the former phase, we adopt the traditional NMF [36] method to find the core communities. While in each iteration of the latter phase, we first randomly select a core community and then perform the expansion process and the contraction process on it. The latter phase stops until the degree of overlap reaches the threshold value μ and then ECOCD yields the final overlapping communities.

The pseudocode corresponding to the entire framework of ECOCD is shown in Algorithm 1. Specifically, the discovery of core communities is described in Steps 1–5 and the expansion with contraction process is shown in Steps 8–15. Function 1 is the most important function proposed in this paper, which contains the expansion process (Steps 8–23) and the contraction process (Steps 24–29). In order to illustrate ECOCD more clearly, Table 1 summarizes the parameters used in this method.

4.1.1. Discovery of core communities

We first use NMF to virtually factorize the adjacency matrix A into two selfsame low-dimensional non-negative matrix $U \in \mathbb{R}_{+}^{n\times k}$ ($A \approx UU^T$). Each column of U represents a community, and each row of U indicates the strength of a node belonging to different communities. Next, to obtain the disjoint communities, we assign each node $v_i \in V$ into $c_{\arg\max_{j\in[1,k]}U_{ij}}$ and then get k disjoint communities $C = \{c_1, c_2, \ldots, c_k\}$ (Steps 1–5 of Algorithm 1). Then, we define C as the core communities (CC). Here, we assume that a node cannot be removed from the core community to which it belongs, but it can be assigned to other communities with multiple memberships.

4.1.2. Expansion with contraction process

After obtaining the core communities, we proceed to the expansion with contraction process, including the expansion process and the contraction process. This process is described in Function 1. In each iteration of this function, an overlapping community oc_r ($r \in [1, K]$) is randomly selected to perform the expansion process to absorb k_{ex} nodes and the contraction process to remove k_{con} nodes. During the iterations, we set that k_{ex} is always larger than k_{con} to ensure the growth of oc_r. To measure the size of the overlaps between communities, we define the degree of overlap as follows:

$$d = \frac{\sum_{i=1}^{k} |oc_i|}{n}$$

where oc_i represents the ith overlapping community and n is the number of total nodes. Function 1 loops until d reaches the overlap threshold μ. The remainder of this section refers exclusively to the steps of Function 1.

Algorithm 1: ECOCD: Overlapping Community Detection Using Expansion with Contraction

Input: n, μ, k, A
Output: The final overlapping communities OC

```
(U, U^T) = NMF (A, k);
C = {c_1 = ∅, ..., c_k = ∅};
for i ← 1 to n do
    c_{arg max_{j∈[1,k]} U_ij} ← c_{arg max_{j∈[1,k]} U_ij} ∪ {v_i};
end
Define C as core communities CC;
Create a copy of CC named overlapping communities OC;
while True do
    d = Σ_{i=1}^{k} |oc_i| / n;
    if d < μ then
        OC = ExpansionWithContraction(OC, CC);
    else
        return OC;
    end
end
```

(1) Expansion Process

The goal of the expansion process is to absorb nodes. We only absorb one node into oc_r in each cycle of the process, and thus absorb k_{ex} nodes in total through k_{ex} cycles. Therefore, we need to design a criterion to choose which node to absorb. This criterion is called the degree of affiliation. Given a node v_i and an overlapping community oc_r, it is

Function 1: *ExpansionWithContraction*

Input: OC, CC
Output: Transitional overlapping communities OC

1 Randomly pick an integer $r \in [1, K]$
2 $k_s = \lceil |oc_r|/10 \rceil$;
3 $k_{ex} = \lceil k_s/2 \rceil$;
4 $k_{con} = \lceil k_s/6 \rceil$;
5 **if** $k_{con} >= k_{ex}$ **then**
6 $\quad k_{con} = k_{ex} - 1$;
7 **end**
8 **for** $i \leftarrow 1$ **to** k_{ex} **do**
9 $\quad NE = \emptyset$;
10 $\quad$ **for** *each* $v_j \in oc_r$ **do**
11 $\quad\quad NE \leftarrow NE \cup neighbor(v_j)$;
12 $\quad$ **end**
13 $\quad NE \leftarrow NE - oc_r$;
14 $\quad$ **for** *each* $v_j \in NE$ **do**
15 $\quad\quad$ Count the number of edges connecting v_j to oc_r;
16 $\quad$ **end**
17 $\quad CA = \emptyset$;
18 $\quad$ Assign k_s nodes $\in NE$ with the most edges connected to oc_r into CA;
19 $\quad$ **for** *each* $v_j \in CA$ **do**
20 $\quad\quad$ Calculate $affi(v_j, oc_r)$;
21 $\quad$ **end**
22 $\quad$ Assign a node $v_j \in CA$ with the highest $affi(v_j, oc_r)$ into oc_r;
23 **end**
24 **for** $i \leftarrow 1$ **to** k_{con} **do**
25 $\quad$ **for** *each* $v_j \in oc_r$ **do**
26 $\quad\quad$ Calculate $Perm(v_j)$;
27 $\quad$ **end**
28 $\quad$ Remove a node $v_j \in oc_r$ that is not in core community cc_r and has the lowest value of negative $Perm(v_j)$;
29 **end**
30 **return** OC.

defined as follows:

$$affi(v_i, oc_r) = \frac{|neigh(v_i) \cap oc_r|}{|oc_r|} + \frac{|oc_r|}{\sum_{v_j \in oc_r} dist(v_i, v_j)} + U[index(v_i), index(oc_r)] \quad (1)$$

where $neigh(v_i)$ obtains the adjacent nodes of node v_i, $dist(v_i, v_j)$ gets the distance of the shortest path from node v_i to node v_j, $index(v_i)$ and $index(oc_r)$ returns the index of node v_i and the index of community oc_r, respectively. Note that the criterion consists of three terms: the first term is similar to the Jaccard similarity [39]; the second term is based on the inverse of the average of the distance of the shortest path, which means the strength of involvement of a node v_i and an overlapping community oc_r [40]; and the last term is based on the factor U decomposed by NMF, which represents the strength of the relationship between nodes and communities [31].

To reduce the amount of calculation, we split each cycle of the expansion process into two parts, namely, quickly finding the candidate nodes and absorbing the only node with the highest degree of affiliation. Take the randomly selected community oc_r in Function 1 as an example, the two parts run as follows:

Quickly find the candidate nodes. To avoid considering all the nodes outside oc_r, we need to limit the search scope of nodes. Hence, we first get all the neighbor nodes NE of each node in oc_r and delete the nodes in NE that already exist in oc_r (Steps 9–13). Then we count the number of edges connecting each node in NE to oc_r (Steps 14–16). Finally, we assign k_s nodes with the most edges connected with oc_r in NE into the candidate set CA (Steps 17–18).

Absorb the only node with the highest degree of affiliation. We first calculate $affi(v_i, oc_r)$ between each node $v_i \in CA$ and oc_r (Steps 19–21), and then assign a node v_i with the highest $affi(v_i, oc_r)$ in CA into oc_r (Step 22).

(2) Contraction Process

The goal of the contraction process is to remove nodes. We only remove one node from oc_r in each cycle of the process, and thus remove k_{con} nodes in total through k_{con} cycles. After k_{ex} cycles of the expansion process, the internal degree, the maximum external connections and the local transitivity of some nodes in oc_r are partially changed, thus we need to contract oc_r to clear out some nodes that are not so belonging to oc_r. Here we use permanence [12] as a criterion to calculate how permanent each node remains in oc_r. Given a node v_i, it is defined as follows:

$$Perm(v_i) = \frac{I(v_i)}{E_{max}(v_i)D(v_i)} - (1 - c_{in}(v_i)) \quad (2)$$

where $I(v_i)$ is the internal degree of node v_i in a community, $D(v_i)$ is the degree of v_i, $E_{max}(v_i)$ is the maximum external community connected with node v_i, and $c_{in}(v_i)$ is the local transitivity of node v_i in a community.

To sum up, in each cycle of the contraction process, we first calculate $Perm(v_i)$ of each node $v_i \in oc_r$ (Steps 25–27) and then remove the only one node $v_i \in oc_r$ that is not in the core community cc_r and has the lowest value of negative $Perm(v_i)$ (Step 28).

4.2. Time complexity analysis

The discovery of core communities and the expansion with contraction process are the two phases of the entire algorithm. And the time complexity of each step in these two phases is analyzed in detail as follows.

- **The time complexity of the discovery of core communities.**
  (*a*) *The time complexity of NMF.* High-performance parallel NMF [41] has been able to achieve time complexity $O(kn/\sqrt{p})$, where p is the number of parallel processes.
  (*b*) *The time complexity of assigning nodes to the communities.* In order to get the core communities, we assign each node $v_i \in V$ into $c_{\arg\max_{j \in [1,k]} U_{ij}}$. Hence, obtaining the core communities from the matrix U consumes $O(kn)$.
- **The time complexity of the expansion with contraction process.**
  (*a*) *The time complexity of the expansion process.* Finding the neighbors of all nodes in oc_r takes $O(en/k)$, where e is the average degree of nodes. The time complexity of calculating the number of edges connecting each node in NE to oc_r is $O(ek_sn/k)$. Calculating the degree of affiliation of each node in CA consumes $O(k_s(n+n^2)/k)$. Hence, the time complexity of each cycle of the expansion process is $O(k_sn^2/k)$. In total, the time complexity of the expansion process is $O(k_{es}k_sn^2/k)$.
  (*b*) *The time complexity of the contraction process.* The time complexities of calculating c_{in} and E_{max} of a node are $O(e^2)$ and $O(n)$, respectively. This process requires calculating n/k nodes, thus the time complexity of each cycle of the contraction process is $O(n^2/k)$. In total, the time complexity of the contraction process is $O(k_{con}n^2/k)$.
  In each iteration of the expansion with contraction process, oc_r absorbs k_{ex} nodes and removes k_{con} nodes. The process stops when the sum of all community sizes reaches $n\mu$. As a result, the process goes through $T = n(\mu - 1)/(k_{ex} - k_{con})$ iterations. In all, the time complexity of this process is $O(T(k_{ex}k_s + k_{con})n^2/k)$.

In summary, the overall time complexity of ECOCD is $O(kn/\sqrt{p} + kn + T(k_{ex}k_s + k_{con})n^2/k)$.

5. Experimental evaluation

In this section, we conduct several experiments to verify the effectiveness and efficiency of the proposed ECOCD method. All the experiments are conducted on a 64-bit architecture computer with 64 GB RAM, Intel(R) Core(TM) E5-2640v3 2.60 GHz processor. The source code and network datasets are available at https://github.com/OliverZhuo/ECOCD.

5.1. Baseline methods

We pick eleven different types of state-of-the-art overlapping community detection methods as the baselines for our proposed ECOCD method.

BNMF [17]: BNMF is a probabilistic method for community detection, which utilizes a Bayesian non-negative matrix factorization model to obtain overlapping communities. BNMF can give the soft classification of a network in a computationally efficient way.

DEMON [22]: DEMON adopts a local-first approach that is capable of uncovering the modular organization of complex networks. Each node selects neighbor clusters by voting democratically.

BNMTF [30]: BNMTF decomposes the adjacency matrix into three factors, allowing it to accurately learn the community membership of each node and the interaction among communities.

BigClam [19]: BigClam is a cluster affiliation model that is capable of quickly detecting community structure in large networks. This method proposes an innovative assumption that the overlaps between communities are densely connected.

SE [23]: SE is a seed set expansion approach. This method focuses on finding suitable seeds and expands them to optimize conductance [15] using the PageRank [24] strategy.

EgoSplit [21]: EgoSplit is a highly scalable and flexible community detection algorithm. It converts the overlapping community detection problem to a simpler and more test-worthy non-overlapping problem.

OCDDP [28]: OCDDP utilizes a similarity-based approach to set the distance among nodes. It goes through three steps to choose the core of the communities and uses membership vectors to represent the belongings of nodes.

DNMF [31]: DNMF designs a new way without setting thresholds in post-processing and uses the adjacency matrix as pseudo-supervision information to further improve the robustness of the algorithm.

NOCD [35]: NOCD is a deep learning-based model, which retains the essence of graph neural networks and probabilistic model for overlapping community detection.

MNMF [33]: MNMF has the ability to process rank deficient matrices to reveal communities in both synthetic and real-world networks.

GREESE [26]: GREESE uses a coupled-seed expansion approach for overlapping community detection with reasonable execution time.

5.2. Experiments on synthetic networks

5.2.1. Datasets

Capitalizing on the flexibility of the LFR benchmark toolkit [42], we can generate various synthetic networks with ground-truth overlapping communities.

We generate five sets of LFR networks (from LFR1 to LFR5), each of which produces several subnetworks by adjusting a particular parameter. In detail, we tune the mixing parameter for topology u, the number of memberships of overlapping nodes om, and the number of overlapping nodes on in the set of $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, $\{2, 3, 4, 5, 6\}$, and $\{100, 200, 300, 400, 500\}$ to generate the subnetworks in LFR1, LFR3, and LFR4, respectively. Similarly, we adjust the number of total nodes n in the range of 1000 to 10 000 with an interval of 1000 and in the range of 10 000 to 50 000 with an interval of 5000 to generate ten and nine subnetworks in LFR2 and LFR5, respectively. The average degree d_{avg}, the maximum degree d_{max}, the minimum size of a single community c_{min}, and the maximum size of a single community c_{max} of all LFR networks are set to be 25, 50, 20, and 50, respectively. Most of the synthetic networks generated above are high-overlap undirected and unweighted networks, which can verify the performance of ECOCD in the high-overlap case. A summary of the above LFR networks is shown in Table 2.

Table 2
The parameters of LFR benchmark networks.

Network	n	on	om	u	d_{avg}	d_{max}	c_{min}	c_{max}
LFR1	1000	200	3	0.1–0.5	25	50	20	50
LFR2	1000–10 000	200	3	0.4	25	50	20	50
LFR3	1000	200	2–6	0.4	25	50	20	50
LFR4	1000	100–500	3	0.4	25	50	20	50
LFR5	10 000–50 000	200	3	0.4	25	50	20	50

5.2.2. Evaluation metrics

For networks having ground-truth communities, we use the overlapping normalized mutual information (ONMI) [13], which is a variant of normalized mutual information [43], to evaluate overlapping community detection results. The ONMI is able to quantify the difference between the overlapping communities derived from the detection algorithm and the true ones, which is defined as follows:

$$ONMI(OC, OC^*) = \frac{H(OC) + H(OC^*) - H(OC, OC^*)}{\max(H(OC), H(OC^*))}$$

where OC and OC^* denote the detected overlapping communities and the ground-truth ones, respectively. $H(OC)$ and $H(OC^*)$ are the entropy of the detected overlapping communities and the ground-truth ones, respectively. And $H(OC, OC^*)$ represents the joint entropy of OC and OC^*.

5.2.3. Parameters settings

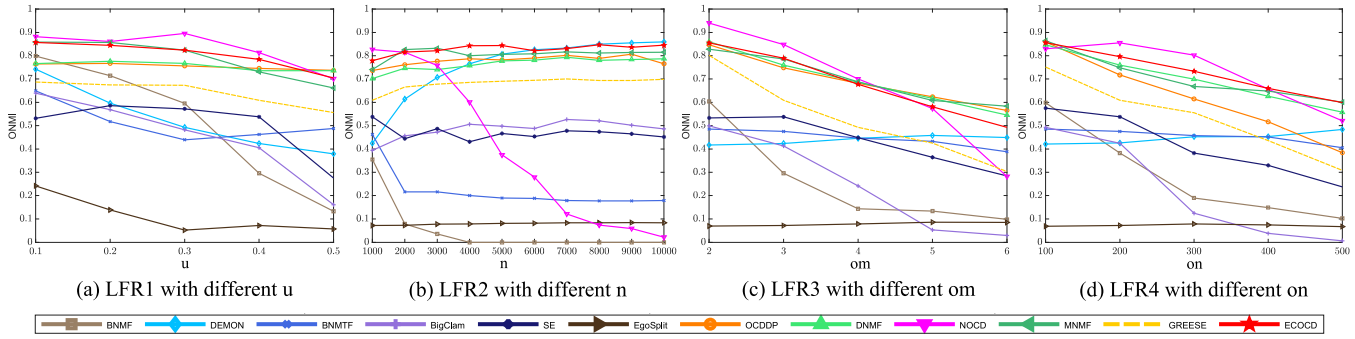
For all algorithms. We adjust the number of communities detected by each algorithm as close to the number of ground-truth communities as possible. And we also adjust the parameters of each baseline algorithm to achieve the best ONMI scores. Each algorithm is repeated 10 times for each synthetic network, and finally the average value of the ONMI is reported.

For our proposed ECOCD method. We use the NNDSVD strategy to initialize the latent factors of NMF for synthetic networks. The optimal overlap threshold values μ for LFR1 and LFR2 are set as 1.386 and 1.393, respectively. For LFR3 with different values of om varied in {2, 3, 4, 5, 6}, we set the values μ as 1.199, 1.393, 1.408, 1.432, and 1.465, respectively. The optimal values μ for LFR4 with different values of on varied in {100, 200, 300, 400, 500} are set as 1.203, 1.387, 1.536, 1.771, and 1.995, respectively. Also, we set $k_s = \lceil |oc_r|/10 \rceil$, $k_{ex} = \lceil k_s/2 \rceil$ and $k_{con} = \lfloor k_s/6 \rfloor$.

5.2.4. Experimental results

We first experiment with the performance of the proposed method on LFR1 with different values of u. As shown in Fig. 2(a), we can see that ECOCD outperforms most algorithms except NOCD. Then we proceed to illustrate the performance of ECOCD via increasing the number of total nodes n on LFR2. Fig. 2(b) shows that the proposed ECOCD method dominates the ONMI scores on LFR2. In particular, we observe that as n increases, the ONMI value of DEMON increases, and the ONMI values of OCDDP and DNMF remain stable, while the ONMI value of NOCD decreases sharply.

Experiments on LFR3 and LFR4 respectively verify the performance of the algorithms with different values of om and on. An increase in



(a) LFR1 with different u (b) LFR2 with different n (c) LFR3 with different om (d) LFR4 with different on

Fig. 2. Performance comparison in terms of ONMI.

Table 3
Description of the real-world networks.

Network	# nodes	# edges	# communities	Node type	Edge type	Community type	Reference
Karate	34	78	2	Trainee	Collaboration	Sport team	[44]
Dolphins	62	159	2	Animal	Friendship	Social network	[44]
Polbooks	105	441	3	Book	Similar preference	Shopping preference	[45]
Jazz	198	2,742	5	Musician	Collaboration	Orchestra	[45]
Netscience	379	914	33	Scientist	Collaboration	Research area	[45]
Polblogs	1490	16,726	2	Politician	Tendency	Political group	[46]
Citeseer	3312	4,732	6	Publication	Citation	Research area	[46]
LastFM	7624	27,806	18	User	Friendship	Social network	[47]

either *om* or *on* represents an increase in the number of overlapping nodes. (When *om* is $2, 3, 4, 5$, or 6, it corresponds to the degree of overlap of $1.2, 1.4, 1.6, 1.8$, or 2 respectively.) The numerical results are presented in Figs. 2(c) and 2(d). As shown, some algorithms perform poorly when *om* (or *on*) values rise. NOCD performs well when $om \in \{2, 3, 4\}$ (or $on \in \{200, 300\}$), but its ONMI value drops sharply with the increase of *om* (or *on*). While for our method, ECOCD does not show a rapid decline in terms of the metric ONMI and ranks in the first echelon when $om \in \{2, 3\}$ (or $on \in \{100, 200, 300, 400, 500\}$).

In summary, the above performance results suggest that our ECOCD method reaches the highest average score of ONMI (i.e., 0.773) among all LFR networks, e.g., it is 1.58% higher than MNMF (0.761), 5.17% higher than DNMF (0.735), and 6.62% higher than OCDDP (0.725).

5.3. Experiments on real-world networks

5.3.1. Datasets

We now evaluate the performance of the proposed ECOCD method using eight real-word low-overlap undirected and unweighted networks in different domains. The eight networks are summarized in Table 3, where the 4th column denotes the number of communities, the 5th and 6th columns indicate the node type and edge type in the network respectively, and the 7th column describes the community type of each network.

5.3.2. Evaluation metrics

We use three representative evaluation metrics, namely the extended overlapping modularity, the modified permanence, and conductance to evaluate the performance of the algorithms.

The extended overlapping modularity EQ [14]. EQ is an overlapping version of modularity [18], which is a high-quality evaluation metric widely used to compare the performance of the community detection algorithms. EQ is defined as follows:

$$EQ = \frac{1}{2m}\sum_{r=1}^{k}\sum_{v_i,v_j\in oc_r}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\frac{1}{O_i O_j}$$

where O_i is the number of memberships of node i, k_i denotes the degree of node i, and m denotes the number of total edges.

The modified permanence $Perm_{ov}(G)$. Permanence [12] is an excellent indicator to measure the degree of modularity of a network and the anti-disturbance ability of an algorithm. Given a graph G with a set of nodes V, the permanence of the graph is $Perm(G) = \sum_{v_i\in V} Perm(v_i)/n$, where $Perm(v_i)$ is the permanence of node v_i. In order to make $Perm(G)$ suitable for evaluating overlapping communities, we slightly modified the definition to include the nodes in overlaps. The modified permanence, denoted as $Perm_{ov}(G)$, is defined as follows:

$$Perm_{ov}(G) = \frac{\sum_{i=1}^{k}\sum_{v_j\in oc_i} Perm(v_j)}{\sum_{i=1}^{k}|oc_i|}$$

Since both $Perm(G)$ and $Perm_{ov}(G)$ calculate the average permanence of all nodes, they have similar evaluation capabilities.

Conductance $\varphi(G)$ [15]. Conductance measures the time of a random walk from a community to out-of-community nodes to converge to a stationary distribution. In addition to being used in circuits, conductance can also be used to measure the conductivity of a community. Given a graph G, conductance $\varphi(G)$ is calculated as below:

$$\varphi(G) = \min_{C\subseteq V}\frac{\sum_{v_i\in C, v_j\in\bar{C}} A_{ij}}{\min(a(C), a(\bar{C}))}$$

where $\bar{C}$ includes the total nodes outside the community C and $a(C) = \sum_{v_i\in C}\sum_{v_j\in V} A_{ij}$.

For EQ and $Perm_{ov}(G)$, a higher score indicates a better overlapping community detection result. For $\varphi(G)$, a lower value means weaker conductivity between communities, which leads to better community division.

5.3.3. Parameters settings

For all algorithms. The number of communities detected by each algorithm is adjusted as close as possible to the values given in Table 3. Each algorithm runs 10 times on each real-world network, and finally we report the mean value and standard deviation of EQ, $Perm_{ov}(G)$, and $\varphi(G)$. Since different degrees of overlap of communities result in different evaluation metric scores, and to be fair, the overlap threshold values μ of communities detected by each algorithm for Karate, Dolphins, Polbooks, Jazz, Netscience, Polblogs, Citeseer, and LastFM are set as 1.02, 1.02, 1.01, 1.01, 1.02, 1.01, 1.01, and 1.02, respectively.

Table 4
Performance comparison in terms of the extended overlapping modularity.

Network	Karate	Dolphins	Polbooks	Jazz	Netscience	Polblogs	Citeseer	LastFM
BNMF	0.328 ± 0.032	**0.385 ± 0.002**	0.491 ± 0.017	0.414 ± 0.014	**0.879 ± 0.005**	0.424 ± 0.002	0.542 ± 0.013	**0.746 ± 0.019**
DEMON	0.315 ± 0.012	0.253 ± 0.027	0.400 ± 0.014	0.079 ± 0.007	0.468 ± 0.017	0.289 ± 0.006	0.294 ± 0.001	0.529 ± 0.001
BNMTF	0.361 ± 0.001	0.357 ± 0.007	0.355 ± 0.001	0.312 ± 0.006	0.515 ± 0.032	0.313 ± 0.001	0.237 ± 0.004	0.256 ± 0.014
BigClam	0.307 ± 0.003	0.374 ± 0.003	0.487 ± 0.024	0.350 ± 0.027	0.806 ± 0.021	0.198 ± 0.019	0.341 ± 0.021	0.381 ± 0.007
SE	0.093 ± 0.001	0.093 ± 0.001	0.113 ± 0.001	0.189 ± 0.001	0.029 ± 0.001	0.009 ± 0.001	0.258 ± 0.001	0.347 ± 0.001
EgoSplit	0.296 ± 0.001	0.320 ± 0.001	0.465 ± 0.002	0.318 ± 0.026	0.798 ± 0.001	0.412 ± 0.001	0.294 ± 0.001	0.529 ± 0.001
OCDDP	0.352 ± 0.001	0.347 ± 0.026	0.430 ± 0.047	0.223 ± 0.034	0.725 ± 0.009	0.332 ± 0.001	0.399 ± 0.001	0.456 ± 0.006
DNMF	0.364 ± 0.017	0.377 ± 0.008	0.494 ± 0.029	0.401 ± 0.006	0.864 ± 0.015	0.421 ± 0.001	0.417 ± 0.029	0.617 ± 0.001
NOCD	0.354 ± 0.005	0.369 ± 0.004	0.457 ± 0.009	0.362 ± 0.007	0.856 ± 0.006	0.335 ± 0.026	**0.612 ± 0.044**	0.574 ± 0.031
MNMF	**0.371 ± 0.001**	0.379 ± 0.001	0.425 ± 0.086	0.402 ± 0.001	0.747 ± 0.034	**0.425 ± 0.001**	0.409 ± 0.080	0.672 ± 0.007
GREESE	0.367 ± 0.001	0.382 ± 0.001	0.467 ± 0.001	0.326 ± 0.001	0.799 ± 0.001	0.291 ± 0.001	0.505 ± 0.001	0.354 ± 0.001
ECOCD	**0.371 ± 0.001**	**0.385 ± 0.001**	**0.516 ± 0.001**	**0.417 ± 0.003**	0.764 ± 0.028	**0.425 ± 0.001**	0.582 ± 0.018	0.706 ± 0.001

Table 5
Performance comparison in terms of the modified permanence.

Network	Karate	Dolphins	Polbooks	Jazz	Netscience	Polblogs	Citeseer	LastFM
BNMF	0.401 ± 0.161	0.217 ± 0.001	0.304 ± 0.021	0.096 ± 0.001	0.517 ± 0.001	0.130 ± 0.001	−0.209 ± 0.001	**0.097 ± 0.001**
DEMON	0.368 ± 0.001	0.017 ± 0.005	0.327 ± 0.003	0.112 ± 0.002	0.596 ± 0.013	0.002 ± 0.007	−0.059 ± 0.003	−0.211 ± 0.008
BNMTF	0.488 ± 0.008	−0.324 ± 0.001	0.243 ± 0.090	0.044 ± 0.054	0.129 ± 0.010	−0.249 ± 0.145	−0.577 ± 0.001	−0.317 ± 0.001
BigClam	0.505 ± 0.031	0.192 ± 0.016	0.330 ± 0.044	0.016 ± 0.003	0.532 ± 0.008	−0.221 ± 0.064	−0.221 ± 0.006	−0.293 ± 0.010
SE	0.026 ± 0.001	0.025 ± 0.001	−0.155 ± 0.001	−0.165 ± 0.001	0.022 ± 0.001	−0.021 ± 0.001	−0.050 ± 0.001	-0.010 ± 0.001
EgoSplit	−0.044 ± 0.001	−0.415 ± 0.006	−0.025 ± 0.001	0.113 ± 0.009	0.514 ± 0.001	0.073 ± 0.005	−0.448 ± 0.001	−0.568 ± 0.001
OCDDP	0.446 ± 0.001	−0.177 ± 0.053	0.194 ± 0.019	−0.081 ± 0.017	0.499 ± 0.001	−0.231 ± 0.004	−0.207 ± 0.004	−0.216 ± 0.005
DNMF	0.509 ± 0.042	0.185 ± 0.025	0.325 ± 0.068	0.065 ± 0.041	0.519 ± 0.017	0.082 ± 0.014	−0.318 ± 0.018	−0.297 ± 0.011
NOCD	0.478 ± 0.009	0.185 ± 0.030	0.223 ± 0.014	−0.043 ± 0.015	0.585 ± 0.012	−0.173 ± 0.017	−0.043 ± 0.014	−0.046 ± 0.013
MNMF	0.538 ± 0.001	0.217 ± 0.001	0.211 ± 0.001	0.098 ± 0.001	0.471 ± 0.018	0.130 ± 0.001	−0.216 ± 0.014	−0.085 ± 0.001
GREESE	0.522 ± 0.001	0.139 ± 0.001	0.213 ± 0.001	−0.088 ± 0.001	**0.743 ± 0.001**	−0.163 ± 0.001	**0.034 ± 0.001**	−0.084 ± 0.001
ECOCD	**0.539 ± 0.001**	**0.221 ± 0.001**	**0.368 ± 0.001**	**0.116 ± 0.019**	0.636 ± 0.013	**0.136 ± 0.003**	−0.015 ± 0.003	−0.018 ± 0.001

Table 6
Performance comparison in terms of conductance.

Network	Karate	Dolphins	Polbooks	Jazz	Netscience	Polblogs	Citeseer	LastFM
BNMF	0.179 ± 0.046	0.071 ± 0.001	0.073 ± 0.011	0.146 ± 0.007	0.015 ± 0.006	0.092 ± 0.004	0.150 ± 0.026	0.049 ± 0.007
DEMON	0.211 ± 0.001	0.200 ± 0.004	0.103 ± 0.001	0.414 ± 0.027	0.011 ± 0.001	0.316 ± 0.02	0.136 ± 0.001	0.136 ± 0.001
BNMTF	0.147 ± 0.001	0.075 ± 0.001	0.064 ± 0.001	0.188 ± 0.001	0.060 ± 0.001	0.203 ± 0.001	0.175 ± 0.015	0.275 ± 0.005
BigClam	0.208 ± 0.013	0.076 ± 0.014	0.055 ± 0.008	0.233 ± 0.030	0.011 ± 0.001	0.370 ± 0.028	0.303 ± 0.028	0.186 ± 0.023
SE	0.133 ± 0.001	0.128 ± 0.001	0.056 ± 0.001	0.236 ± 0.001	0.040 ± 0.001	0.787 ± 0.001	0.025 ± 0.001	0.029 ± 0.001
EgoSplit	0.250 ± 0.001	0.265 ± 0.001	0.109 ± 0.001	0.253 ± 0.055	0.011 ± 0.001	0.091 ± 0.001	0.044 ± 0.001	0.136 ± 0.001
OCDDP	0.151 ± 0.001	0.087 ± 0.024	0.075 ± 0.028	0.229 ± 0.016	0.034 ± 0.001	0.081 ± 0.001	0.039 ± 0.001	**0.013 ± 0.001**
DNMF	0.138 ± 0.020	0.086 ± 0.013	0.055 ± 0.001	0.157 ± 0.020	0.014 ± 0.005	0.082 ± 0.001	0.285 ± 0.023	0.127 ± 0.009
NOCD	0.149 ± 0.006	0.103 ± 0.005	0.062 ± 0.007	0.184 ± 0.007	0.016 ± 0.007	0.154 ± 0.003	0.052 ± 0.002	0.072 ± 0.007
MNMF	**0.132 ± 0.001**	**0.067 ± 0.001**	0.061 ± 0.008	**0.123 ± 0.001**	0.011 ± 0.001	**0.078 ± 0.015**	0.073 ± 0.001	0.086 ± 0.001
GREESE	0.189 ± 0.001	0.121 ± 0.001	0.062 ± 0.001	0.159 ± 0.001	0.037 ± 0.001	0.111 ± 0.001	0.048 ± 0.001	0.030 ± 0.001
ECOCD	**0.132 ± 0.001**	**0.067 ± 0.001**	**0.054 ± 0.001**	**0.123 ± 0.001**	**0.009 ± 0.002**	**0.078 ± 0.013**	**0.017 ± 0.008**	0.074 ± 0.001

For our proposed ECOCD method. We use the random strategy to initialize the latent factors of NMF for real-world networks (except for Polbooks and LastFM, in which we employ the NNDSVD strategy). Here we set $k_s = \lceil |oc_r|/10 \rceil$, $k_{ex} = \lceil k_s/2 \rceil$, and $k_{con} = \lceil k_s/6 \rceil$.

5.3.4. Experimental results

The performance results in terms of EQ, $Perm_{ov}(G)$ and $\varphi(G)$ are presented in Tables 4, 5, and 6, respectively. The best results are in boldface, while the top-2 results are highlighted in gray. We conclude that ECOCD performs best on Karate, Dolphins, Polbooks, Jazz and Polblogs, and performs the second best on Citeseer and LastFM in terms of the metric EQ as shown in Table 4. Besides, we find that, in terms of the metrics $Perm_{ov}(G)$ and $\varphi(G)$, ECOCD takes lead in seven networks except the LastFM network as shown in Tables 5 and 6. It is remarked that ECOCD ranks third on the LastFM network in terms of $Perm_{ov}(G)$, which is very close to the performance of SE in the second place.

ECOCD does not perform well on the Netscience dataset in terms of the metric EQ. This is because one of the size of the initial communities generated by the traditional NMF [36] is larger than 200, while the rest are smaller than 25. Large differences in initial community size heavily affect the EQ score. While the metric $Perm_{ov}(G)$ represents the average Permanence value of all nodes, thereby it is less affected by the number and scale of initial communities. Therefore, ECOCD performs better on the Netscience dataset in terms of $Perm_{ov}(G)$.

In addition, ECOCD does not perform well on the LastFM dataset in terms of the metric $\varphi(G)$. Of the 18 overlapping communities detected by ECOCD on this dataset, 15 communities did not absorb nodes with the fewest connections to out-of-community nodes for the best overlaps, thereby the $\varphi(G)$ score of ECOCD is a moderate 0.074. However, OCDDP and SE propose new edge density calculations based on local density and connectivity, achieving good overlaps but not well the entire overlapping communities. Therefore, they perform top-2 in terms of $\varphi(G)$, but perform poorly in terms of EQ.

In summary, among the twenty-four experiments (eight real-world networks × three evaluation metrics), ECOCD takes the first place in seventeen cases and the second place in four cases. That is, our proposed ECOCD method leads in 87.5% of the experiments, so there is no doubt that ECOCD is capable of revealing high-quality real-world overlapping communities.

5.4. Parameters sensitivity analyses

- **Initialization strategy for NMF.** We employ the NMF method to obtain the core communities, where different initialization

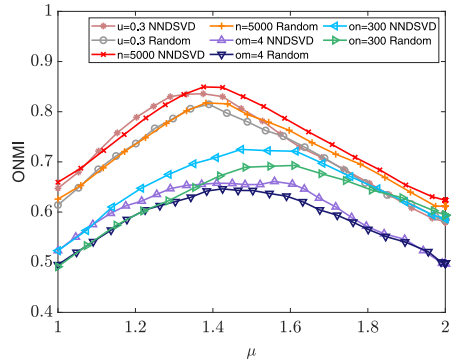


Fig. 3. Effect of initialization strategy on synthetic networks.

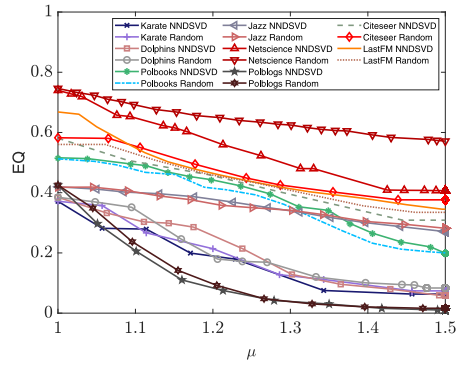
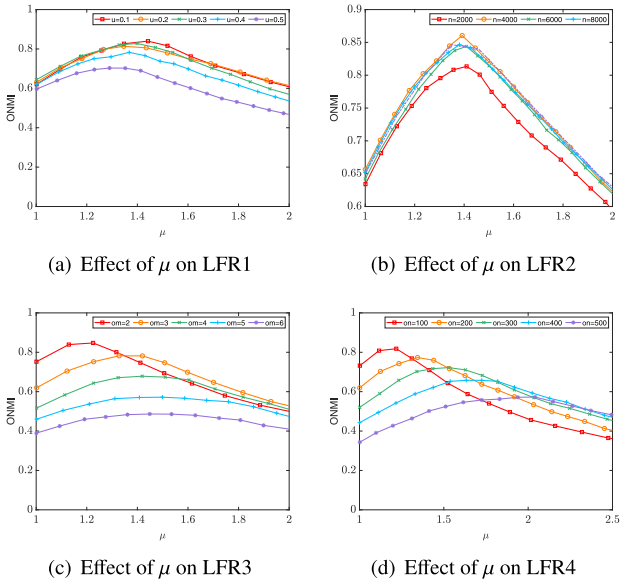


Fig. 4. Effect of initialization strategy on real-world networks.

strategies for NMF are adopted to generate different core communities, so we need to analyze the impact of different initialization strategies. For synthetic networks, we analyze the effect of initialization strategies on LFR1 ($u = 0.3$), LFR2 ($n = 5000$), LFR3 ($om = 4$), and LFR4 ($on = 300$). From Figs. 3 and 4, we are aware that the NNDSVD strategy is more suitable for synthetic networks and the random strategy is more applicable to real-world networks. Therefore, we use NNDSVD and random strategies to initialize the latent factors of NMF for synthetic and real-world networks, respectively.

- **The overlap threshold μ.** The value of μ indicates the degree of overlap of the final overlapping communities. Figs. 5 and 6 demonstrate the influence of varying degrees of overlap in the final overlapping communities on different networks. Here, we empirically analyze the parameter sensitivity of μ.
  (a) *For synthetic networks.* We show the effect of different values of μ on the first four LFR networks in Table 2 in terms of ONMI. Specifically, for the experiment on the effect of μ on LFR1, we only adjust the parameter u and keep the other parameters (i.e., n, om, and on) unchanged. The experiments are similar on LFR2, LFR3, and LFR4, where the parameters n, om, and on are fixed, respectively. The results are presented in Fig. 5. From Figs. 5(a) and 5(b), we can see that the ONMI value first increases and then decreases with the increase of μ, and ECOCD achieves the highest ONMI score when μ is approximately equal to 1.4 on both LFR1 and LFR2. From Figs. 5(c) and 5(d), we can see that the ONMI value shows a similar trend as mentioned above. The difference is that as om or on value increases, the optimal values μ also increase on both LFR3 and LFR4.
  (b) *For real-world networks.* Fig. 6 reveals the effect of μ in terms of EQ in different real-world networks. We can see that the EQ scores continue to decrease, and the reason is that all the real-world networks discussed here are low-overlap.
- **The size of the candidate set k_s.** The value k_s controls the number of nodes added to the candidate set. A larger candidate



(a) Effect of μ on LFR1 (b) Effect of μ on LFR2

(c) Effect of μ on LFR3 (d) Effect of μ on LFR4

Fig. 5. Effect of μ on synthetic networks.

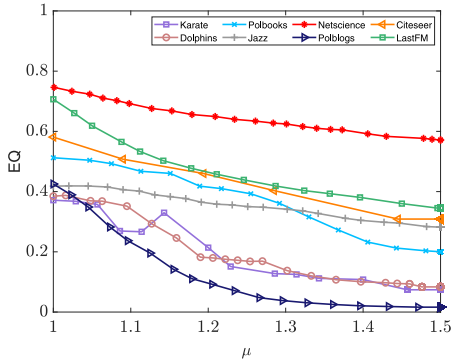


Fig. 6. Effect of μ on real-world networks.

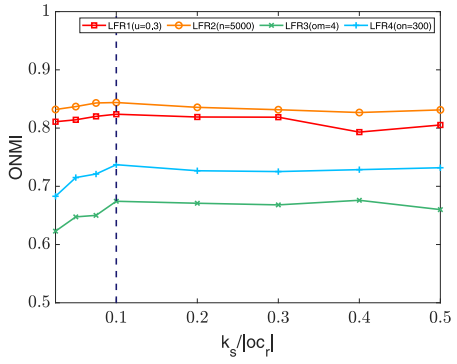


Fig. 7. Effect of k_s.

set means a greater amount of computation, so we need to set the size of the candidate set as small as possible while ensuring the quality of community detection. For clarity, we analyze the effect of k_s on LFR1 ($u = 0.3$), LFR2 ($n = 5000$), LFR3 ($om = 4$), and LFR4 ($on = 300$). The results are presented in Fig. 7, where the values of $k_s/|oc_r|$ are in the range of $[0, 0.5]$. As can be seen from the figure, we determine k_s as $\lceil |oc_r|/10 \rceil$, since it achieves the best quality of community detection and reduces computational complexity (see Section 4.2).

- **The number of cycles of the expansion process k_{ex} and the contraction process k_{con}.** In each iteration, when the expansion

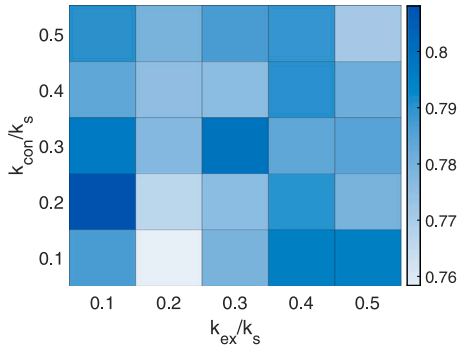


Fig. 8. Joint effect of k_{ex} and k_{con}.

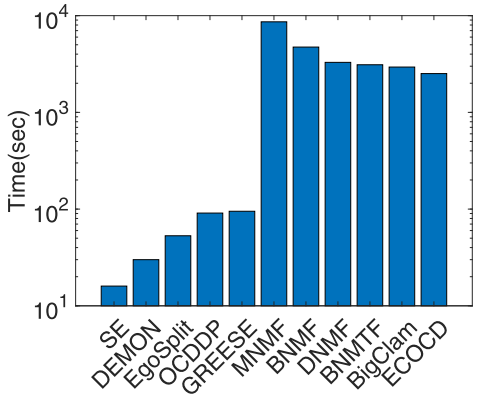


Fig. 9. Running time comparison.

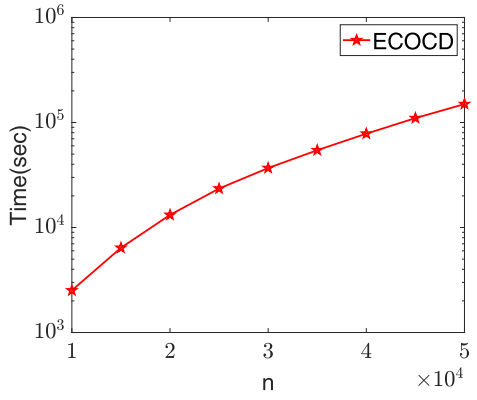


Fig. 10. Scalability testing.

process and the contraction process are performed, oc_r absorbs k_{ex} nodes and removes k_{con} nodes. That is, in each iteration, k_{ex} controls the number of nodes that join oc_r, and k_{con} controls how many nodes are removed from oc_r. In order to ensure that each community continues to grow, k_{con} needs to be set as an integer that is less than k_{ex}. Fig. 8 illustrates the joint effect of parameters k_{ex} and k_{con} on LFR1 when $u = 0.4$. This heat map takes ONMI as the evaluation metric. The X and Y axes in the figure record the values of k_{ex}/k_s and k_{con}/k_s respectively, where $k_s = \lceil |oc_r|/10 \rceil$. It can be observed that ECOCD is robust to k_{ex} and k_{con}, when k_{ex} and k_{con} are in the range of $\lceil k_s/10 \rceil$ - $\lceil k_s/2 \rceil$.

5.5. Running time analysis

We compare the running time of ECOCD with ten baseline algorithms on LFR5 when $n = 10\,000$. In all the experiments in this part, the parameters of ECOCD are set as $\mu = 1.4$, $k_s = \lceil |oc_r|/10 \rceil$, $k_{ex} = \lceil k_s/2 \rceil$, and $k_{con} = \lceil k_s/6 \rceil$. Note that we ignore the NOCD algorithm because it requires GPU resources for deep learning. As shown in Fig. 9, the running time of ECOCD is shorter than that of MNMF, BNMF, DNMF, BNMTF and BigClam requiring matrix computation, but much longer than that of SE based on seed expansion, DEMON based on local-first approach, EgoSplit based on ego-net, and OCDDP based on density peak. This demonstrates that the proposed ECOCD method has a trade-off between detection accuracy and running time.

We further test the running time of ECOCD on LFR5 with various n. As shown in Fig. 10, the growth trend of the running time of ECOCD gradually slows down as the network size n becomes larger. EC takes 149 734 s to perform overlapping community detection on LFR5 with $n = 50\,000$, which demonstrates the scalability of ECOCD on large networks.

6. Conclusion

In this paper, we propose an expansion with contraction approach, namely the ECOCD method for overlapping community detection. This method is proposed on the basis of traditional NMF [36], which is intuitive and easy to follow. Besides, ECOCD runs without complex parameters. We evaluate the ECOCD method on both synthetic and real-world networks, and experimental results show that our ECOCD method outperforms eleven state-of-the-art overlapping community detection methods in terms of four metrics. Despite the excellent performance, there is still room for further improvements, such as how to add regularization terms to the traditional NMF to control the number and size of initial communities.

CRediT authorship contribution statement

Zhijian Zhuo: Conceptualization, Methodology, Software, Writing – original draft, Data curation, Writing – review & editing. **Bilian Chen:** Methodology, Funding acquisition, Validation, Supervision, Writing – review & editing. **Shenbao Yu:** Methodology, Writing – review & editing. **Langcai Cao:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We have shared our code via a link mentioned in our paper. The data used are all public.

Acknowledgments

The work was supported in part by National Natural Science Foundation of China (Grants nos. 12371515, 61836005, 62176225 and 62171391).

References

[1] S.-M. Horng, C.-L. Wu, How behaviors on social network sites and online social capital influence social commerce intentions, Inf. Manag. 57 (2) (2020) 103176.
[2] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (12) (2002) 7821–7826.
[3] M. Plantié, M. Crampes, Survey on social community detection, in: Social Media Retrieval, Springer, 2013, pp. 65–85.
[4] M.A. Javed, M.S. Younis, S. Latif, J. Qadir, A. Baig, Community detection in networks: a multidisciplinary review, J. Netw. Comput. Appl. 108 (2018) 87–111.
[5] X. Li, B. Kao, Z. Ren, D. Yin, Spectral clustering in heterogeneous information networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 4221–4228.
[6] X. Luo, Z. Liu, M. Shang, J. Lou, M. Zhou, Highly-accurate community detection via pointwise mutual information-incorporated symmetric non-negative matrix factorization, IEEE Trans. Netw. Sci. Eng. 8 (1) (2020) 463–476.

[7] C. He, Y. Zheng, X. Fei, H. Li, Z. Hu, Y. Tang, Boosting nonnegative matrix factorization based community detection with graph attention auto-encoder, IEEE Trans. Big Data 8 (4) (2021) 968–981.
[8] A.K. Ghoshal, N. Das, S. Das, Disjoint and overlapping community detection in small-world networks leveraging mean path length, IEEE Trans. Comput. Soc. Syst. 9 (2) (2021) 406–418.
[9] J.J. Whang, D.F. Gleich, I.S. Dhillon, Overlapping community detection using neighborhood-inflated seed expansion, IEEE Trans. Knowl. Data Eng. 28 (5) (2016) 1272–1284.
[10] T.P. Peixoto, Network reconstruction and community detection from dynamics, Phys. Rev. Lett. 123 (12) (2019) 128301.
[11] S.E. Garza, S.E. Schaeffer, Community detection with the label propagation algorithm: a survey, Physica A 534 (2019) 122058.
[12] T. Chakraborty, S. Srinivasan, N. Ganguly, A. Mukherjee, S. Bhowmick, On the permanence of vertices in network communities, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 1396–1405.
[13] A.F. McDaid, D. Greene, N. Hurley, Normalized mutual information to evaluate overlapping community finding algorithms, 2011, CoRR abs/1110.2515.
[14] H. Shen, X. Cheng, K. Cai, M. Hu, Detect overlapping and hierarchical community structure in networks, Physica A 388 (8) (2009) 1706–1712.
[15] J. Leskovec, K.J. Lang, M. Mahoney, Empirical comparison of algorithms for network community detection, in: Proceedings of the 19th International Conference on World Wide Web, 2010, pp. 631–640.
[16] E.M. Airoldi, D. Blei, S. Fienberg, E. Xing, Mixed membership stochastic blockmodels, Adv. Neural Inf. Process. Syst. 21 (2008).
[17] I. Psorakis, S. Roberts, M. Ebden, B. Sheldon, Overlapping community detection using bayesian non-negative matrix factorization, Phys. Rev. E 83 (6) (2011) 066114.
[18] M.E. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. 103 (23) (2006) 8577–8582.
[19] 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, 2013, pp. 587–596.
[20] P. Sun, X. Wu, Y. Quan, Q. Miao, Influence percolation method for overlapping community detection, Physica A 596 (2022) 127103.
[21] A. Epasto, S. Lattanzi, R. Paes Leme, Ego-splitting framework: from non-overlapping to overlapping clusters, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 145–154.
[22] M. Coscia, G. Rossetti, F. Giannotti, D. Pedreschi, Demon: a local-first discovery method for overlapping communities, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 615–623.
[23] J.J. Whang, D.F. Gleich, I.S. Dhillon, Overlapping community detection using seed set expansion, in: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, 2013, pp. 2099–2108.
[24] R. Andersen, F. Chung, K. Lang, Local graph partitioning using pagerank vectors, in: 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), IEEE, 2006, pp. 475–486.
[25] T. Ma, Q. Liu, J. Cao, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, Lgiem: global and local node influence based community detection, Future Gener. Comput. Syst. 105 (2020) 533–546.
[26] K. Asmi, D. Lotfi, A. Abarda, The greedy coupled-seeds expansion method for the overlapping community detection in social networks, Computing 104 (2) (2022) 295–313.
[27] M. Lu, Z. Zhang, Z. Qu, Y. Kang, Lpanni: overlapping community detection using label propagation in large-scale complex networks, IEEE Trans. Knowl. Data Eng. 31 (9) (2018) 1736–1749.
[28] X. Bai, P. Yang, X. Shi, An overlapping community detection algorithm based on density peaks, Neurocomputing 226 (2017) 7–15.
[29] H. Lu, Z. Shen, X. Sang, Q. Zhao, J. Lu, Community detection method using improved density peak clustering and nonnegative matrix factorization, Neurocomputing 415 (2020) 247–257.
[30] Y. Zhang, D.-Y. Yeung, Overlapping community detection via bounded nonnegative matrix tri-factorization, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 606–614.
[31] F. Ye, C. Chen, Z. Zheng, R. Li, J.X. Yu, Discrete overlapping community detection with pseudo supervision, in: 2019 IEEE International Conference on Data Mining (ICDM), IEEE, 2019, pp. 708–717.
[32] C. Fang, Z. Lin, Overlapping communities detection based on cluster-ability optimization, Neurocomputing 494 (2022) 336–345.
[33] V. Leplat, A.M. Ang, N. Gillis, Minimum-volume rank-deficient nonnegative matrix factorizations, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 3402–3406.
[34] T. Chakraborty, S. Ghosh, N. Park, Ensemble-based overlapping community detection using disjoint community structures, Knowl.-Based Syst. 163 (2019) 241–251.
[35] O. Shchur, S. Günnemann, Overlapping community detection with graph neural networks, Comput. Sci. 50 (2.0) (2019) 49.
[36] D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, Adv. Neural Inf. Process. Syst. 13 (2000).
[37] C. Boutsidis, E. Gallopoulos, Svd based initialization: a head start for nonnegative matrix factorization, Pattern Recognit. 41 (4) (2008) 1350–1362.
[38] R. Andersen, F. Chung, K. Lang, Local graph partitioning using pagerank vectors, in: 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), IEEE, 2006, pp. 475–486.
[39] S. Bag, S.K. Kumar, M.K. Tiwari, An efficient recommendation generation using relevant jaccard similarity, Inform. Sci. 483 (2019) 53–64.
[40] T. Chakraborty, N. Park, V. Subrahmanian, Ensemble-based algorithms to detect disjoint and overlapping communities in networks, in: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2016, pp. 73–80.
[41] R. Kannan, G. Ballard, H. Park, A high-performance parallel algorithm for nonnegative matrix factorization, ACM SIGPLAN Not. 51 (8) (2016) 1–11.
[42] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (4) (2008) 046110.
[43] A. Lancichinetti, S. Fortunato, J. Kertész, Detecting the overlapping and hierarchical community structure in complex networks, New J. Phys. 11 (3) (2009) 033015.
[44] J. Xie, S. Kelley, B.K. Szymanski, Overlapping community detection in networks: the state-of-the-art and comparative study, ACM Comput. Surv. (CSUR) 45 (4) (2013) 1–35.
[45] J. Xie, B.K. Szymanski, X. Liu, Slpa: uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process, in: 2011 IEEE 11th International Conference on Data Mining Workshops, IEEE, 2011, pp. 344–349.
[46] Z. Ding, X. Zhang, D. Sun, B. Luo, Low-rank subspace learning based network community detection, Knowl.-Based Syst. 155 (2018) 71–82.
[47] B. Rozemberczki, R. Sarkar, Characteristic functions on graphs: birds of a feather, from statistical descriptors to parametric models, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 1325–1334.

Zhijian Zhuo received his B.Eng. degree from Xiamen University, China, in 2019. He is currently pursuing a master's degree in engineering at Xiamen University. His research interests include overlapping community detection and optimization.

Bilian Chen received her Ph.D. degree from The Chinese University of Hong Kong in 2012. Now she is an Associate Professor in Xiamen University. Her research interests include machine learning, optimization theory and recommendation system. Her publications appear in IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Neural Networks and Learning Systems, and so on.

Shenbao Yu received his master degree in automation from Xiamen University, China, in 2017. He is now a Ph.D. candidate in Xiamen University. His research interests include recommendation system, educational data mining and probabilistic graphical models.

Langcai Cao received the Ph.D. degree in automation from Xiamen University, in 2011. He is currently an Associate Professor in Xiamen University. His research interests include research and development of information systems, process intelligence and machine learning.