Actively Error-Correcting Semi-Supervised Clustering Community Detection Algorithm Postprint

Authors: Xiankun Zhang, Liu Yuanbo, Ren Jing, Zhang Gaozhen

Date: 2018-05-24T00:00:00+00:00

Abstract

Classical unsupervised clustering algorithms are fast, simple, and can directly partition large-scale datasets; however, due to the relatively complex network structures, their partitioning accuracy is not high. To address this issue, we propose an active learning-based error correction semi-supervised community detection algorithm called ESCD (error correction semi-supervised community detection algorithm), which performs stepwise computation of the traditional K-means algorithm and incorporates pairwise constraints during the clustering process. Based on prior information, the algorithm retains correct partitions and corrects erroneous ones to alter the network' s connectivity relationships, thereby making the network exhibit a more pronounced block structure; the partitioning process terminates when the distances between nodes and cluster centers cease to change. Experimental results demonstrate that, compared with existing community detection algorithms, the ESCD algorithm achieves higher accuracy while requiring significantly less supervisory information than other semi-supervised algorithms.

Full Text

Active Error-Correcting Semi-Supervised Clustering Community Detection Algorithm

Zhang Xiankun, Liu Yuanbo†, Ren Jing, Zhang Gaozhen
(School of Computer Science & Information Engineering, Tianjin University of Science & Technology, Tianjin 300457, China)

Abstract

Classical unsupervised clustering algorithms are fast, simple, and capable of directly partitioning large-scale datasets. However, due to the complex structure of networks, their partitioning accuracy remains unsatisfactory. To address this limitation, we propose an Error-Correcting Semi-supervised Community Detection algorithm (ESCD) based on active learning. This approach decomposes the traditional K-means algorithm into stepwise computations while incorporating pairwise constraints during the clustering process. By preserving correct partitions and correcting erroneous ones according to prior information, the algorithm modifies network connectivity to produce a more pronounced block structure. The partitioning process terminates when the distances between nodes and cluster centers stabilize. Experimental results demonstrate that compared with existing community detection algorithms, ESCD achieves higher precision while requiring significantly less supervisory information than other semi-supervised methods.

Keywords: active learning; error-correcting semi-supervised community discovery; K-means algorithm; pairwise constraints

0 Introduction

Many real-world networks are composed of communities—groups of nodes with dense internal connections and sparse external connections. Community detection has become a research hotspot across numerous domains because it enables accurate identification of social groups, facilitating connections among individuals with similar interests for sharing ideas and expertise. Complex network clustering is fundamentally a graph partitioning problem, which has led to widespread adoption of traditional clustering algorithms such as K-means, K-medoids, spectral clustering, and graph clustering for community discovery.

The K-means algorithm, proposed by MacQueen in 1967, is a classic clustering method characterized by its efficiency and simplicity. It has been successfully applied to both general data clustering and network data processing. The core idea involves identifying K cluster centers that minimize the sum of squared distances between each data point and its nearest center. However, real-world networks exhibit high structural complexity with ambiguous community boundaries. Unsupervised clustering lacks prior information, relying solely on distance calculations between node feature vectors. Consequently, noise or irregular topological structures significantly influence node classification, resulting in unstable performance. In complex networks where intra-community connections are weak and inter-community connections are abundant, unclear network structures cause boundary nodes to be misclassified. Moreover, community detection results depend heavily on initial center selection, with accuracy degrading substantially when network structures are intricate.

To overcome these limitations, semi-supervised community detection algorithms have been proposed. Semi-supervised learning leverages both labeled and unlabeled data to substantially improve learning efficiency. However, existing semi-supervised methods suffer from several drawbacks: they require large amounts of labeled data, cannot truly alter complex network structures, and remain inefficient. As illustrated in [Figure 1: see original paper], conventional approaches randomly inject pairwise constraints outside the main community detection framework. Only through three pairs of constraints can node 5 be correctly assigned, demonstrating that random annotation is redundant and inefficient. This external injection mode constrains the effectiveness of semi-supervised learning for four primary reasons: (a) labeled data is added outside the complete algorithmic framework, which involves multiple iterations, failing to utilize each iterative step; (b) prior information intervenes in community partitioning through random annotation, requiring substantial prior knowledge that is typically expensive and difficult to obtain; (c) even with abundant prior information, correct network partitioning cannot be guaranteed because randomly added labels may not target the most critical nodes; and (d) pairwise constraints only guide the partitioning of specific nodes without fundamentally altering network structure to produce clearer block patterns.

In response to these challenges, we propose ESCD, an active error-correcting semi-supervised community detection algorithm that integrates active learning principles into semi-supervised clustering. Our approach decomposes traditional K-means into stepwise computations, treating each distance iteration result as a coarse clustering outcome. We compute node membership degrees based on current coarse clusters and actively inject minimal prior information through logical reasoning to modify network structure and obtain accurate partitions. ESCD fully exploits each iteration of K-means, automatically correcting misclassified nodes during every distance computation step to produce clearer block structures in complex networks. Experimental results confirm that ESCD achieves higher precision while substantially improving the efficiency of conventional semi-supervised clustering algorithms.

1.1 K-means Algorithm

This work employs the widely-used K-means algorithm, a partition-based unsupervised clustering method that uses distance as a similarity metric. The algorithm divides a dataset into K non-overlapping clusters, producing results with high intra-cluster similarity and low inter-cluster similarity.

Given a network graph $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ represents the node set and $E = \{e_1, e_2, ..., e_n\}$ represents the link set. The adjacency matrix $A = [a_{ij}]_{n\times n}$ directly reflects connectivity between nodes: $a_{ij} = 1$ if a link exists between v_i and v_j, and $a_{ij} = 0$ otherwise. During clustering, the adjacency matrix serves as algorithm input, with each node' s adjacency matrix vector representing its n-dimensional feature. Through multiple distance iterations between nodes and K cluster centers C_j, the algorithm converges to a final community partition when assignments stabilize. Distance is measured using Equation (1):

$$d(x_i, C_j) = \sqrt{\sum_{\alpha=1}^{m} (x_{i\alpha} - c_{j\alpha})^2}, \quad i = 1, 2, ..., n;\ j = 1, 2, ..., K$$

where x_i represents the feature vector of node i and c_j represents the center of cluster j.

1.2 Semi-Supervised Learning Algorithm

Semi-supervised learning incorporates various forms of supervisory information, most commonly sample class labels that explicitly specify each sample' s category. Beyond class labels, pairwise constraints represent relationships between sample pairs, including must-link and cannot-link constraints. A must-link relationship exists when two samples share the same class label, while a cannot-link relationship exists when they belong to different classes. Compared to class labels, pairwise constraints require less prior information and are easier to obtain.

Several approaches have utilized pairwise constraints in community detection. Liu et al. employed label propagation to spread labeled node information to neighboring nodes. Silva et al. integrated semi-supervised methods based on modularity maximization. Zhang et al. directly added node pairwise constraints to the decomposition of the adjacency matrix. Yang et al. established a unified semi-supervised community detection framework using latent space graph regularization. However, these methods share common limitations in semi-supervised learning efficiency.

1.3 Node Labeling Methods

To address the inability of semi-supervised learning to identify the most valuable data for labeling, active learning strategies have been applied to community detection. These strategies automatically and effectively select the most informative nodes, which domain experts then confirm and add to the labeled dataset. Yang et al. proposed an active semi-supervised community detection model based on non-negative matrix factorization that automatically selects the most unstable links by computing information entropy of community membership probability distributions. Cheng et al. identified important nodes through network weighting methods for efficient prior information utilization. However, entropy-based labeling requires obtaining probability distributions of community memberships, which is difficult in many community detection algorithms, limiting its applicability.

We propose an efficient and general node labeling approach that determines node stability within its community by computing membership degrees. For a node i in community k, its external degree $k_{out}^{(i)}$ counts connections to nodes outside the community, while its internal degree $k_{in}^{(i)}$ counts connections within the community. We define the degree of membership $MD_i^{(k)}$ for node i in community k based on coarse clustering results as:

$$MD_i^{(k)} = \frac{k_{in}^{(i)}}{k_{AN}^{(i)}}$$

where $k_{AN}^{(i)}$ represents all adjacent nodes of node i, equivalent to the node' s total degree ($k_{AN}^{(i)} = k_{in}^{(i)} + k_{out}^{(i)}$). By evaluating node membership degrees, we can identify boundary nodes and hub nodes within each community.

---

2 Active Error-Correcting Semi-Supervised Community Discovery Algorithm

The proposed ESCD algorithm actively selects the most uncertain nodes and incorporates minimal labeled data to improve community partitioning accuracy. The core methodology comprises three main steps:

a) Selecting initial cluster centers based on available prior information.

b) Decomposing K-means into stepwise computations:
   - Computing node membership degrees in current coarse clusters based on distance-based partitioning results.
   - Actively adding prior information through pairwise constraints and logical reasoning to modify the adjacency matrix structure and cluster indices.
   - Re-clustering using the modified adjacency matrix until distances between nodes and their community centers stabilize.

c) Obtaining final community partitioning results.

The following sections provide detailed analysis of key technical components.

2.1 Initial Cluster Center Selection Principle

K-means performance heavily depends on initial cluster center selection. Most research focuses on optimizing this selection: Kaufman et al. proposed selecting data points with highest local density; Rodriguez et al. introduced a fast density peak search algorithm identifying cluster centers as points with higher density than neighbors and relatively large distances from other high-density points;

Basu et al. utilized labeled data for k-means initialization; Leng et al. proposed a novel initial center selection strategy using labeled datasets to ensure at least one data object is selected from each class.

To avoid instability from random initial center selection and ensure full utilization of prior information, we adopt a semi-supervised initialization scheme. Gu et al. proposed a semi-supervised initial center selection strategy that uses labeled datasets to assist in selecting initial cluster centers. In a dataset containing s distinct labels, the average of several nodes sharing the same label serves as that class' s initial cluster center, yielding s initial centers. The remaining $K - s$ centers are selected from unlabeled data as the nodes farthest from existing centers. This approach eliminates randomness and ensures center dispersion. For experimental accuracy, we report the average NMI across ten clustering runs as final results.

2.2 Algorithm Implementation

The ESCD algorithm proceeds as follows:

Input: Network $G = (V, E)$, adjacency matrix A, number of communities K, convergence condition (node indices no longer change), maximum iterations $Iter$.

Output: Community set $C = \{C_1, C_2, ..., C_K\}$.

a) Select K nodes as initial cluster centers $\{c_1, c_2, ..., c_K\}$ according to the initialization principle described above.

b) Compute distances from each node to all cluster centers, assign each node to its nearest center, and update cluster centers using Equation (4).

c) Based on the current assignment, identify hub nodes $\{V_{hub}\}$ and boundary nodes $\{V_{mar}\}$ in each community. Using must-link and cannot-link constraints from prior information, reconstruct edges between these nodes according to three rules to update network structure, yielding a new adjacency matrix A_h.

d) Recompute distances from each node to cluster centers using the updated adjacency matrix A_h, reassign nodes to nearest centers, and update cluster centers again using Equation (4).

e) Repeat steps c) and d) until node assignments stabilize, then terminate.

2.3 Active Pairwise Constraint Addition Method

During clustering, misclassified nodes exhibit common characteristics, particularly when network modular structure is ambiguous. Two primary factors cause misclassification: (a) numerous inter-community connections create unclear boundaries and weak block structures, with connection endpoints typically being boundary nodes; (b) some intra-community nodes have large external degrees k_{out} and small internal degrees k_{in}, indicating stronger connections to external nodes than internal ones. By identifying these boundary nodes, pairwise constraints can efficiently guide whether edges between them should be connected or disconnected, actively enhancing community clarity and significantly improving correct partitioning.

As described in Section 2.2, we add minimal labels based on each clustering iteration result, injecting prior information as pairwise constraints into the adjacency matrix. Decomposing a complete K-means execution into steps and adding constraints after the first distance computation allows active neighbor node selection to maximize the utility of each obtained prior information. The active semi-supervised process follows three rules, with the framework illustrated in [Figure 2: see original paper].

Rule 1: Selection Rule
Based on the first coarse clustering result, we traverse each community to select hub nodes and some boundary nodes, as edges between boundary nodes contain cross-community connections.

- **Boundary nodes** $\{V_{mar}\}$: These are the most uncertain and informative nodes, containing cross-community connections. As shown in Equation (5), boundary nodes are defined as those with minimum membership degree in a community. If multiple nodes share the same minimum membership value, all are marked as boundary nodes for subsequent guidance by prior information.
- **Hub nodes** $\{V_{hub}\}$: These are the most stable nodes in a community. As shown in Equation (6), hub nodes are defined as those with maximum membership degree—nodes where the proportion of intra-community connections is highest. If multiple nodes share the maximum membership value, preference is given to nodes not connected to boundary points; if all connect to boundary nodes, one is randomly selected from those with highest membership.

Since boundary node uncertainty may affect the true membership of their neighbors, [FIGURE:3-(a)] illustrates selected hub nodes and boundary nodes $\{A, C\}, \{B\}$.

Rule 2: Query Rule
This rule uses prior information to guide error-correction operations. We traverse each community, comparing them pairwise. As shown in [FIGURE:3-(b)], we first mark all adjacency edges between boundary nodes $\{A, B\}$ and $\{C, B\}$

of two communities, as edges connected to cross-community links are also uncertain. We then query the pairwise constraints of nodes $\{A, B\}$ and $\{C, B\}$ based on true labels. If a must-link relationship exists, the connection is preserved. Simultaneously, we check the true community labels of the boundary node pairs: if they match node A' s label, all marked edges connecting to B are disconnected; if they match B' s label, the opposite operation occurs. As shown in [FIGURE:3-(c)], this process successfully corrects node B, which was previously misassigned to the wrong community.

Rule 3: Strengthening Rule
To prevent excessive sparsity at nodes where edges were disconnected in Rule 2, we strengthen connections for these nodes based on true labels of boundary and hub nodes. As shown in [FIGURE:3-(d)], if a must-link exists between disconnected node B and hub node V_{hub}, they are reconnected. Similarly, connections are strengthened on the right side. This reinforcement prevents individual nodes from becoming isolated after disconnection, which could lead to misclassification into other communities, especially in smaller communities with sparse connections. The strengthening operation makes community block structures more pronounced while preventing sparsely-connected nodes from becoming noise points and avoiding incorrect partitioning of small-scale communities. After this correction, the community structure becomes clearer. The updated cluster centers and reconstructed network structure then serve as input for the next K-means distance iteration, repeating until all node-to-center distances stabilize.

---

3 Experimental Analysis

We evaluated the ESCD method on both synthetic and real-world networks, comparing it against several widely-used semi-supervised community detection algorithms: Spin [20], SNMF [16], and CL-ML [15]. Spin employs pairwise node constraints, SNMF is a symmetric non-negative matrix factorization model with implicit space graph regularization, and CL-ML incorporates logical reasoning with prior information in a non-negative matrix factorization framework. We used Normalized Mutual Information (NMI) as the evaluation metric.

3.1 Evaluation Criteria

NMI compares the detected communities with ground-truth communities to assess accuracy. Let K be the number of communities in the current partition and K' be the number in the ground-truth partition. NMI is calculated as:

$$\mathrm{NMI} = \frac{-2\sum_{i=1}^{K}\sum_{j=1}^{K'} N_{ij}\log\left(\frac{N_{ij}N}{N_{i\cdot}N_{\cdot j}}\right)}{\sum_{i=1}^{K} N_{i\cdot}\log\left(\frac{N_{i\cdot}}{N}\right)+\sum_{j=1}^{K'} N_{\cdot j}\log\left(\frac{N_{\cdot j}}{N}\right)}$$

where N_{ij} is the number of nodes in community i of the detected partition that belong to community j of the ground-truth, $N_{i.}$ is the total number of nodes in detected community i, $N_{.j}$ is the total number of nodes in ground-truth community j, and N is the total number of nodes.

3.2 Experimental Design

Experiments were conducted on six real-world networks and synthetic networks. The real network datasets are summarized in . Synthetic networks simulate the scale-free properties of node degrees and community sizes commonly found in real networks, with parameters including: number of nodes N, average degree k, minimum community size $minc$, maximum community size $maxc$, and mixing parameter μ (inter-community connection probability). Higher μ values indicate more complex community structure and greater clustering difficulty. We tested on GN networks with $K_{out} = 7$ and $K_{out} = 8$, and LFR networks with $\mu = 0.65$ and $\mu = 0.75$, as detailed in .

3.3.1 Real Network Experimental Results

[Figure 4: see original paper] presents results across six real-world networks, with each curve representing algorithm performance. To test stability, we ran each method ten times, with data points indicating average NMI values and vertical bars showing variance. Results demonstrate that ESCD achieves excellent performance with minimal labels. For instance, in the Football network, NMI improves from 0.921 to 1.0 with only 2.32% labeled data. All six real network experiments validate ESCD’ s effectiveness and stability, showing that actively injecting pairwise constraints during stepwise distance computation significantly enhances K-means accuracy and stability.

3.3.2 Synthetic Network Experimental Results

To verify ESCD’ s effectiveness on synthetic networks, we first tested on GN networks, with [Figure 5: see original paper] showing averaged results over ten runs. The x-axis represents the proportion of added pairwise constraints, and the y-axis shows NMI values (1.0 indicates perfect alignment with ground truth). ESCD demonstrates superior accuracy compared to alternative methods. For GN networks with $K_{out} = 7$, ESCD achieves perfect performance (NMI=1.0) with only 2% labels. When $K_{out} = 8$, ESCD attains NMI values of 0.713 and 0.807 with 1% and 2% labels respectively, outperforming other semi-supervised algorithms.

Further validation on two LFR networks shows ESCD and SNMF both perform well, achieving ideal results with very few labels. ESCD surpasses SNMF when approximately 5% labels are added, as shown in [Figure 6: see original paper].

3.4 Partitioning Process Demonstration

This section presents the active error-correcting semi-supervised partitioning process using a real community example. Based on experimental observations, the active adjacency-based prior information addition continuously corrects misclassified nodes through iterative refinement.

In the Football network, the first distance computation yields six correctly partitioned communities, with errors concentrated in several central communities. [Figure 7: see original paper] shows the initial coarse clustering result. To illustrate the computational process, we examine a subset of communities in detail.

Based on the current coarse clustering, we traverse community pairs sequentially. [Figure 8: see original paper] shows two primary communities with multiple interleaved nodes; in ground truth, nodes of the same color belong to the same community. Each pair of communities undergoes three rounds of evaluation to maximize assessment of uncertain nodes. First, selection rules identify hub and boundary nodes. Using the membership formula, the hub nodes are identified as 29 and 50 (enlarged in [Figure 8: see original paper] for visibility). Boundary nodes are 59, 74, and 89 (74 and 89 share identical membership values).

The query rule then checks whether nodes 59 and 74/89 belong to the same community based on true labels. The must-link relationships {59=74} and {59=89} are established. Since {59=74} are in the same community but unconnected, a red edge is added. For {59=89}, the existing connection is preserved (shown in green). According to the query rule, all edges connecting node 59 to its current community should be disconnected; however, since node 59 has no connections within its assigned community, no disconnection is needed—demonstrating that boundary nodes are those least connected to their own community.

After this correction, node 59 is properly assigned. The same rules are applied to correct the next community pair. As shown in [FIGURE:9-(a)], three nodes (64, 60, 70) are misclassified in the left blue community. In the first iteration, nodes 64, 60, and 70 are selected as boundary nodes. Through semi-supervised rules, connections are established between 64, 60, and 70 while disconnecting edges to the left community. Since the right community' s hub node has a different true label, this indicates that hub node connections also require pairwise constraint guidance. However, this does not necessitate many labels due to logical reasoning within constraint pairs. As shown in [FIGURE:9-(b)], the second iteration selects nodes 98 and 70 as boundary nodes, ultimately correcting nodes 60, 64, and 98 to the right community. This active error-correction process between two communities is now complete, with subsequent communities processed similarly.

4 Conclusion

Building upon analysis of existing clustering algorithms' characteristics and limitations, the proposed ESCD algorithm decomposes traditional clustering into stepwise distance computations. At each step, it actively selects uncertain nodes for prior information injection and employs three rules to reorganize complex network connections, substantially improving network modularity. Theoretical analysis and experiments on real datasets demonstrate that ESCD significantly enhances community detection accuracy and stability.

Future work will extend this semi-supervised model to social networks such as Weibo, incorporating additional reference factors for community detection, node importance metrics, and seed node selection strategies. We will also investigate the impact of introducing multiple reference factors on real community partitioning and consider integrating deep learning for feature extraction from complex community structures to enhance practical applicability.

References

[1] Lin Youfang, Wang Tianyu, Tang Rui. An effective model and algorithm for community detection in social networks [J]. Journal of Computer Research and Development, 2012, 49(2): 337-345.

[2] Newman M E. The Structure and Function of Complex Networks [J]. SIAM Review, 2003, 45(2): 168-256.

[3] Perozzi B, Akoglu L. Focused clustering and outlier detection in large attributed graphs [C]// Proc of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2014: 1346-1355.

[4] MacQueen J. Some methods for classification and analysis of multivariate observations [C]// Proc of the 5th Berkeley Symposium on Mathematical Statistics and Probability. 1967: 281-297.

[5] Allahverdyan A E, Steeg G V, Galstyan A. Community detection with and without prior information [J]. Europhysics Letter, 2009, 90(1): 983-995.

[6] Ma Xiaoke, Gao Lin, Rong Xue, et al. Semi-supervised clustering algorithm for community structure detection in complex networks [J]. Physica A: Statistical Mechanics & Its Applications, 2012, 389(1): 187-197.

[7] Leng Mingwei, Yao Yukai, Cheng Jianjun, et al. Active semi-supervised community detection algorithm with label propagation [C]// Proc of International Conference on Database Systems for Advanced Applications. Berlin: Springer, 2013: 324-338.

[8] Zhu Xiaojin. Semi-supervised learning with graphs [D]. Pittsburgh, PA: Carnegie Mellon University, 2005.

[9] Liu Dong, Bai Hongyu, Li Huijia, et al. Semi-supervised community detection using label propagation [J]. International Journal of Modern Physics B, 2014, 28(29): 145-208.

[10] Silva T C, Liang Zhao. Semi-supervised learning guided by the modularity measure in complex networks [J]. Neurocomputing, 2012, 78(1): 30-37.

[11] Zhang Zhongyuan, Sun Kaidi, Wang Siqi. Community structure detection in complex networks with partial background information [J]. Scientific Reports, 2013, 3(11): 32-41.

[12] Yang Liang, Cao Xiaochun, Jin Di, et al. A unified semi-supervised community detection framework using latent space graph regularization [J]. IEEE Trans on Cybernetics, 2017, 45(11): 2585-2598.

[13] Yang Liang, Jin Di, Wang Xiao, et al. Active link selection for efficient semi-supervised community detection [J]. Scientific Reports, 2015, 5: 9039.

[14] Cheng Jianjun, Leng Mingwei, Li Longjie, et al. Active semi-supervised community detection based on must-link and cannot-link constraints [J]. PLOS One, 2014, 9(10): 88-110.

[15] Kaufman L, Rousseeuw P J. Finding groups in data: an introduction to cluster analysis [J]. The Astrophysical Journal, 1998, 508(1).

[16] Rodriguez A, Laio A. Machine learning: Clustering by fast search and find of density peaks [J]. Science, 2014, 344(6191): 1492-1496.

[17] Basu S, Banerjee A, Mooney R J. Semi-supervised clustering by seeding [C]// Proc of the 19th International Conference on Machine Learning. 2002.

[18] Leng Mingwei, Huang Liang, Li Longjie, et al. Active semi-supervised community detection based on asymmetric similarity measure [J]. International Journal of Modern Physics B, 2015, 29(13): 1550078.

[19] Gu Lei, Lu Xianling. Semi-supervised subtractive clustering by seeding [C]// Proc of International Conference on Fuzzy Systems & Knowledge Discovery. 2012: 738-741.

[20] Eaton E, Mansbach R. A spin-glass model for semi-supervised community detection [C]// Proc of the 26th AAAI Conference on Artificial Intelligence. 2012: 900-906.

Note: Figure translations are in progress. See original paper for figures.