A new approach to semi-supervised fuzzy graph-cut clustering

Vu Nam Anh[1], Mai Dinh Sinh[2], Dang Trong Hop[1*]

[1]School of Information and Technology, Hanoi University of Industry, 298 Cau Dien, Tay Tuu, Hanoi, Vietnam;
[2]Institute of Construction Technology, Le Quy Don Technical University, 236 Hoang Quoc Viet, Nghia Do, Hanoi, Vietnam.
[*]Corresponding author: hopdt@ haui.edu.vn

Received 4 Apr. 2026; Revised 26 May 2026; Accepted 1 Jun. 2026; Published 25 Aug. 2026.
DOI: https://doi.org/10.54939/1859-1043.j.mst.113.2026.158-167

ABSTRACT

Traditional feature-based clustering algorithms often fail to capture the geometric structure of data effectively. To address this limitation, this paper proposes a semi-supervised fuzzy graph cut clustering algorithm (SS-FGCC), which integrates anchor-point labels into fuzzy clustering and exploits graph structures to improve clustering quality while reducing computational complexity to linear time. Experiments on six synthetic datasets and five UCI datasets demonstrate that SS-FGCC consistently outperforms RCut, NCut, FCM, FFCAG, FGCC, and CEHM, achieving up to 82.76% accuracy. Moreover, the proposed method remains stable on large-scale datasets where conventional graph cut methods often suffer from memory limitations, making SS-FGCC an efficient and scalable clustering solution.

Keywords: SS-FGCC; Fuzzy clustering; Graph cut; Semi-supervised; Anchoring graphs.

1. INTRODUCTION

Traditional graph-cut algorithms (e.g., RCut [1], NCut [2]) capture complex structures but suffer from prohibitive computational costs. Alternatively, Fuzzy C-Means (FCM) [3] manages uncertainty via soft labeling but ignores geometric structures, rendering it noise-sensitive and prone to convergence deviations. To overcome this, Fast Fuzzy Graph Cutting (FGCC) [4] achieves linear complexity $O(n)$ by utilizing anchor-based graphs. However, its unsupervised nature lacks directional guidance, risking solution bias on complex manifolds or noisy data. To address this, we propose the Semi-Supervised Fast Graph-Cutting Fuzzy Clustering (SS-FGCC) algorithm. The primary novelty of SS-FGCC is the embedding of hard semi-supervised constraints directly into the fuzzy graph-cut objective function. By using a sparse subset of labeled anchors as semantic landmarks and propagating them via simplex projection, SS-FGCC strictly bounds the feasible search space. This mitigates unsupervised bias while preserving the optimal $O(n)$ linear complexity.

The main contributions of this paper are: Proposed SS-FGCC model: Integrates fuzzy indicators with graph cuts and hard semi-supervised anchor constraints, significantly enhancing clustering stability and accuracy in highly uncertain and noisy scenarios; Direct optimization: Solves the clustering indicator matrix directly without continuous relaxation or post-processing algorithms (e.g., K-means), thoroughly eliminating information loss and solution deviation; Performance and scalability optimization: Employs an alternating optimization scheme based on bipartite graph approximation, minimizing both computational and storage complexities to $O(n)$. This ensures excellent responsiveness for Big Data while maintaining the original manifold structure.

2. FUNDAMENTAL KNOWLEDGE

2.1. Theoretical basis

Traditional graph-cutting models, such as Ratio Cut (RCut) [1] and Normalized Cut (NCut) [2], optimize partition costs based on the graph Laplacian matrix but are strictly constrained by high computational complexity. To scale efficiently to large-scale datasets, the anchor graph model utilizes m representative landmarks ($m \ll n$) to approximate the global similarity matrix via a bipartite structure:

$$BB^T = Z\Delta^{-1}Z^T \quad (1)$$

where, $B \in R^{n\times m}$ is the transition matrix; Z is the local similarity matrix; and $\Delta \in R^{m\times m}$ is a diagonal matrix with the element p on the diagonal is $\Delta_{pp} = \sum_{i=1}^{n} z_{ip}$. This replacement helps reduce the computational load to a linear level $O(n)$.

In parallel, fuzzy clustering, notably Fuzzy C-Means (FCM) [3], introduces a soft-labeling mechanism where each data point belongs to multiple clusters with membership degrees $f_{ij} \in [0,1]$, effectively capturing data uncertainty. However, since FCM relies solely on coordinate-based attributes and disregards the underlying geometric manifold, it exhibits high sensitivity to noise and initialization bias. The proposed SS-FGCC framework bridges this gap by seamlessly embedding hard semi-supervised constraints within a unified, anchor-based fuzzy graph-cut objective function.

2.2. Proposed SS-FGCC algorithm

2.2.1. Overall architecture

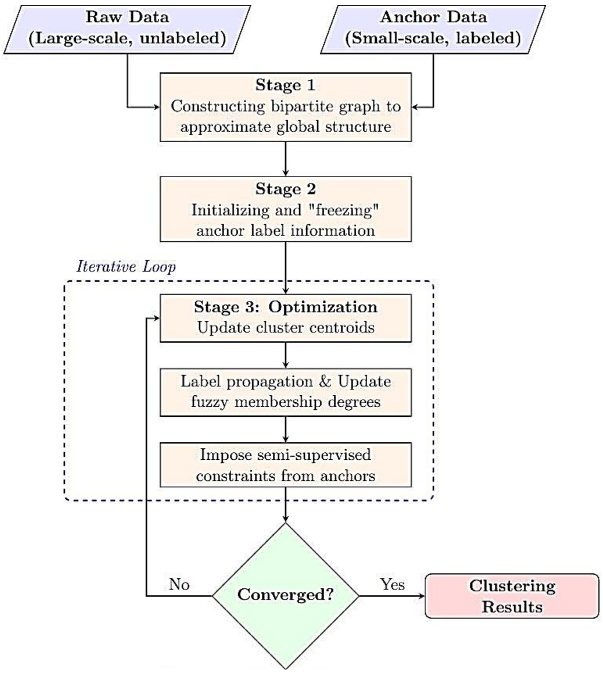

Figure 1. *Flowchart of the SS-FGCC algorithm.*

SS-FGCC innovatively integrates graph cutting, fuzzy logic, and semi-supervised learning into a linear framework. By utilizing labeled anchors to guide optimization along the original data manifold, it effectively overcomes the computational limitations of RCut/NCut and the noise sensitivity of unsupervised models like FCM and FGCC. This mechanism thoroughly resolves the shortcomings of its predecessors, delivering superior accuracy on large-scale datasets.

The SS-FGCC algorithm leverages a sparse set of labeled anchors as directional landmarks to accurately guide the clustering of vast unlabeled data. This tightly coupled process comprises three stages : **(1) Graph Approximation:** Instead of exhaustive pairwise calculations, it measures similarities only between data and representative anchors. This creates a compact bipartite graph to optimize computational costs while preserving the core geometric structure. **(2) Prior Initialization:** The system rigidly preserves the labels of known anchors to serve as structural seeds for the optimization process. **(3) Alternating Optimization:** This core cyclical process iteratively updates cluster centroids and fuzzy membership degrees. Here, labeled anchors act as hard constraints, preventing centroids from deviating from the true manifold and guiding fuzzy label propagation until stable convergence (see Figure 1).

2.2.2. Proposed algorithm

The workflow of the Semi-Supervised Fast Graph-Cutting Fuzzy Clustering (SS-FGCC) algorithm is realized through three core computational stages. The combination of matrix transformations and mathematical optimization is detailed in the following steps:

Phase 1: Constructing an Anchor-Based Graph Approximation

To reduce complexity from $O(n^2)$ to $O(n \cdot m)$, we extract m anchors $U \in \mathbb{R}^{m\times d}$ and compute the local similarity matrix

$$Z_{ij} = \exp\left(-\frac{\left\|x_i - u_j\right\|_2^2}{2\sigma^2}\right) \tag{2}$$

The normalized transition matrix is $B = Z\Delta^{-1/2}$, where $\Delta_{pp} = \sum_{i=1}^{n} Z_{ip}$, effectively approximating the global similarity as BB^T.

Phase 2: Semi-Supervised Prior Initialization

We encode the labeled anchor subset (indices L_{idx}) into a one-hot matrix Y_L.

The fuzzy membership matrix $F \in \mathbb{R}^{n\times c}$ is randomly initialized ($F1 = 1, F \geq 0$) and hard-constrained by $F[L_{idx}, :] := Y_L$. The objective function minimizes data-centroid distances with a fuzzy regularization term:

$$J(F, M) = \sum_{i=1}^{n}\sum_{j=1}^{c} f_{ij}\left\|b_i - m_j\right\|_2^2 + \gamma\|F\|_F^2 \quad \text{s.t.} \quad F[L_{idx}, :] \coloneqq Y_L Z_{ij} = \exp\left(-\frac{\left\|x_i - u_j\right\|_2^2}{2\sigma^2}\right) \tag{3}$$

Phase 3: Alternating Optimization & Label Propaganda: We iteratively update M and F until convergence ($\|F^t - F^{t-1}\|_F < \varepsilon$).

+ **Update M:** Fixing F^{t-1}, cluster centroids are updated as $M^t = \Lambda^{-1}(F^{t-1})^T B$, where $\Lambda_{kk} = \sum_{i=1}^{n} F_{ik}^{t-1}$.

+ **Update F:** Fixing M^t, F is updated independently for each data point via Simplex Projection: $F_{ij}^t = \max\left(-\frac{1}{2\gamma}\left\|b_i - m_j\right\|_2^2 - \theta_i, 0\right)$, where θ_i is the projection translation threshold ensuring $\sum_{j=1}^{c} f_{ij} = 1$.

+ **Impose Constraints:** Crucially, we enforce $F^t[L_{idx}, :] := Y_L$ immediately after updating F. This persistent lock forces centroids to shift around precise anchor coordinates, navigating the fuzzy propagation toward the true manifold structure.

2.2.3. Proof of convergence

The convergence of the SS-FGCC algorithm is mathematically guaranteed by Theorem 1.

Theorem 1. *Under the alternating optimization rules for the cluster centroids M and the fuzzy membership matrix F, the objective function $J(F, M)$ decreases monotonically at each iteration and is guaranteed to converge to a local minimum.*

Proof. First, since $F \geq 0$, $\gamma > 0$, and $\left\|b_i - m_j\right\|_2^2 \geq 0$, the objective function (8) is naturally bounded below by zero ($J(F, M) \geq 0$). Second, when fixing F^t, the subproblem w.r.t M is an unconstrained convex quadratic function. Setting the first derivative $\partial J/\partial M = 0$ yields the unique global optimal solution M^{t+1} via Eq. (10) , which guarantees:

$$J(F^t, M^{t+1}) \leq J(F^t, M^t) \tag{4}$$

Third, when fixing M^{t+1}, the optimization w.r.t F decomposes into independent convex subproblems under simplex constraints. The Simplex Projection (Algorithm 1) ensures the optimal convergence coordinates are found on the simplex surface. Crucially, the hard semi-supervised intervention $F_{L_{idx},:} := Y_L$ strictly restricts the search space to a valid convex subspace where prior labels are preserved. Since the update reduces the objective function over the entire space, imposing boundary conditions on this valid subspace still strictly preserves the monotonic property, ensuring:

$$J(F^{t+1}, M^{t+1}) \leq J(F^t, M^{t+1}) \quad (5)$$

Combining (4) and (5), yields the strict monotonic inequality chain:

$$J(F^t, M^t) \geq J(F^t, M^{t+1}) \geq J(F^{t+1}, M^{t+1}) \geq 0 \quad (6)$$

Since the sequence $\{J(F^t, M^t)\}$ is monotonically decreasing and bounded below, according to the Monotone Convergence Theorem, the SS-FGCC algorithm inevitably converges to a stable local minimum.

Table 1. *Simplex projection.*

Algorithm 1: Simplex Projection
Input: Intermediate vector$v \in R^c$.
Output: Probability vector$f \in R^c$ satisfied $f \geq 0$ and $\sum f_j = 1$.
Initialization:$c = length(v)$.
Applying projection:
1. Sorting vector v in descending order is called $\mu \in R^c$
2. Calculate the following cumulative total: $cssv_j = \sum_{k=1}^{j} \mu_k$ $(\forall j \in 1, \dots, c)$
3. Find the effective dimension position:

$$\rho = \left\{\mu_j - \frac{cssv_j - 1}{j} > 0\right\}$$

4. Calculate the penalty threshold for translation (threshold): $\theta = \frac{cssv_\rho - 1}{\rho}$
5. **FOR**j = 1 **TO**c**DO**
Calculate the projected probability value: $f_j = (v_j - \theta, 0)$
ENDFOR
RETURN vector f

Table 2. *SS-FFGS algorithm pseudocode.*

Algorithm 2: SS-FGCC Algorithm
Input: Original dataset$X \in R^{n \times d}$; labeled data index set L_{idx}, standard label matrix $Y_L \in R^{|L_{idx}| \times c}$; number of clusters c, number of anchors m.
Output: Membership level matrix$F \in R^{n \times c}$and centroid matrix $M \in R^{c \times m}$.
Step 1: Initialize parameters
Set the fuzzy coefficient γ, bandwidth σ, maximum number of loops T_{max}, and convergence error ε. Set $t = 1$.
Step 2: Data Preprocessing (Phases 1 & 2)
2.1 Set anchor sets $U \in R^{m \times d}$ from the label dataset.
2.2 Calculate the similarity matrix $Z \in R^{n \times m}$ and normalize to transition matrix $B \in R^{n \times m}$
2.3 Random initialization $F^{(0)} \in R^{n \times c}$ satisfied $F^{(0)} 1 = 1, F^{(0)} \geq 0$.
2.4 "Setting up semi-supervised boundary conditions": $F^{(0)}[L_{idx}, :] = Y_L$.
Step 3: Optimization (Phase 3)
WHILE$\left|F^{(t)} - F^{(t-1)}\right|_F < \varepsilon$**AND**$t \leq T_{max}$**DO**

3.1 Calculate the matrix $\Lambda \in R^{c\times c}$với $\Lambda_{kk} = \sum_{i=1}^{n} F_{ik}^{(t-1)}$.
3.2 Update the centroid matrix $M^{(t)}$.
3.3 **FOR** $i = 1$ **TO** n **DO**
IF$i \notin L_{idx}$**THEN** //Updated template, not yet a label
a. Calculate the distance vector E_i and vector $V_i = \frac{-1}{2\gamma} E_i$
b. Update the probability distribution $F_{i,:}^{(t)}$ by **Algorithm 1**
END IF
3.4 **END FOR**
3.5 Semi-supervised constraint: $F^{(t)}[L_{idx}, :] = Y_L$
3.6 $t \leftarrow t + 1$.
END WHILE

2.2.4. Computational complexity analysis

Instead of performing eigenvalue decomposition on the adjacency matrix $n \times n$ with calculated costs $O(n^3)$ Unlike standard spectral clustering, the SS-FGCC algorithm has a significant advantage. In each iteration, updating M requires $O(n \cdot c^2 + n \cdot m \cdot c)$, and update F via the required simplex projection $O(m \cdot c)$. Combined with the graph construction process, the overall computational complexity of the method is maintained at a certain level $O(n \cdot m \cdot d)$. The storage complexity is also only at a certain level $O(n \cdot m)$ instead of $O(n^2)$, this confirms its ability to perfectly handle large-scale datasets.

3. RESULTS AND DISCUSSION

3.1. Prepare for the experiment

3.1.1. Installation environment

All experiments were executed on Google Colaboratory (Intel Xeon 2.00GHz CPU, 12GB RAM, NVIDIA Tesla T4 16GB GPU). Models were implemented in Python 3.12.13, utilizing CUDA 13.0 for accelerated matrix optimization.

3.1.2. Experimental data

The proposed algorithm is evaluated on two dataset types: six 2D synthetic datasets (testing boundary delineation and complex manifold handling), and five real-world UCI benchmark datasets [5] (varying in size, dimensions, and classes to assess scalability).

Table 3. *Characteristics of the simulated datasets.*

Symbol	Dataset name	Number of data points	Characteristics
D1	Cluster-in-cluster	2024	Nested core and ring structure; tests boundary separation.
D2	Crescent-and-fullmoon	875	Non-uniform dispersion between crescent and full-moon shapes.
D3	Two-moon	300	Interlocking half-moons with blurred boundaries; tests manifold preservation.
D4	Multi-density dataset	1000	Extreme density variations between spherical clusters; tests stability.
D5	Anisotropic	900	Stretched elliptical clusters; challenges purely Euclidean distance metrics.
D6	Background noise	900	Spherical clusters with heavy random background noise; tests robustness.

Detailed characteristics of the Benchmark datasets are compiled and presented at Table 4.

Table 4. *Summary of the characteristics of standard datasets.*

Dataset Name	Size (n)	Dimensions (d)	Number of clusters (c)
Segment	2.310	19	7
COIL100	7.200	1024	100
USPS	9.298	256	10
Letter	20.000	16	26
Covertype	581.012	54	7

3.2. Experimental setup

3.2.1. Set the parameters

The SS-FGCC hyperparameters are configured as follows: **(1) Convergence:** The algorithm terminates if iterations exceed the maximum limit ($maxiter = 100$) or the membership matrix update falls below the tolerance threshold $\left(\left|F^{(t)} - F^{(t-1)}\right|_F < 10^{-6}\right)$. **(2) Fuzzy parameter (γ):** Fixed at $\gamma = 0.5$ to balance probability boundary sparsity and smoothness. **(3) Kernel bandwidth (σ):** Adaptively tuned per dataset to preserve manifold structures (e.g., $\sigma = 0.15$ for the nested D1 dataset to avoid cross-clustering, and $\sigma = 0.8$ for the widely dispersed D2 dataset). **(4) Anchor Initialization:** Total anchors (m) are extracted via the BKHK algorithm ($m = 512$ for small/medium, $m = 16384$ for large datasets). A stratified 5-10% subset of m is One-Hot encoded (Y_L) and hard-locked iteratively to guide clustering; the impact of this ratio is evaluated in the sensitivity analysis

3.2.2. Experimental methods

To comprehensively evaluate SS-FGCC, we compared it against ten established baselines : **(1) Graph & Spectral Clustering:** RCut [1], NCut [2], LSC [6], and EDCAG [7]; **(2) Fuzzy & Density-based:** FCM [3], DBSCAN [8], FFCAG [9], and its direct unsupervised predecessor, FGCC [4]; **(3) Semi-Supervised:** SSFCM [10]; and **(4) Clustering Ensemble:** CEHM [11]. These baselines were strategically selected to validate SS-FGCC's improvements in computational scalability, noise robustness, and the distinct advantage of embedding semi-supervised anchor constraints within a fuzzy graph-cut framework.

3.2.3. Evaluation index

Clustering performance is quantified through the following three standard metrics: Accuracy (ACC): The proportion of correctly clustered samples after optimal label mapping via the Kuhn-Munkres algorithm, ranging from 0 to 1. Normalized Mutual Information (NMI): An entropy-based metric evaluating the structural similarity between predicted clusters and actual ground-truth labels. Running Time: The total computation time (in seconds) required for convergence, serving as a primary indicator of the algorithm's scalability on large datasets.

3.3. Experimental results and discussion

3.3.1. Experimental results on a simulated dataset

The results of visual clustering on simulated datasets are presented in Figure 2. Semi-monitored anchor points are marked with a red star symbol. As visually demonstrated in Figure 2, SS-FGCC achieved near-perfect ACC (>0.98) and NMI (>0.85) on datasets D1-D5, with processing times under 0.3s. On the D6 dataset with heavy background noise, it still maintained robust performance (ACC = 0.8900, NMI = 0.6581).

3.3.2. Experimental results based on standard data.

The performance of the SS-FGCC algorithm is summarized in Table 5 and Table 6. The results show that the algorithm achieves high accuracy (ACC) on most datasets.

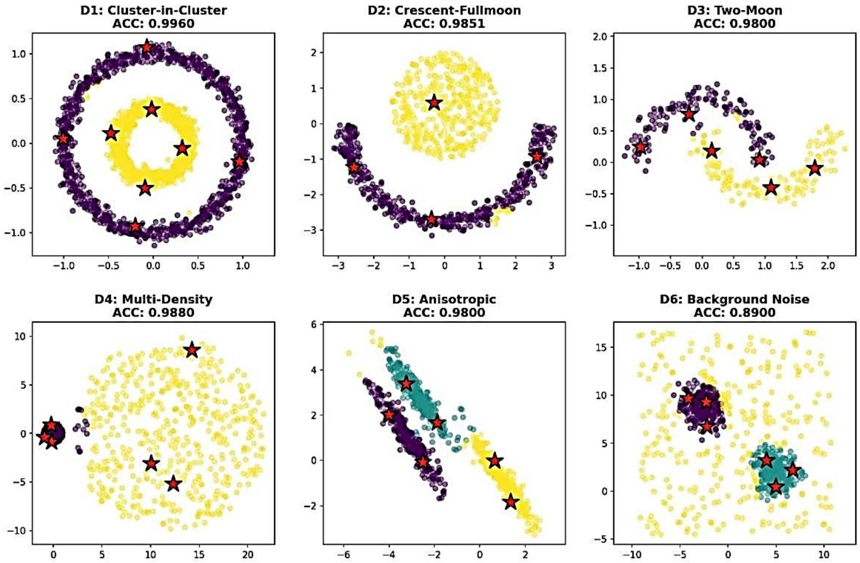

Figure 2. *Visualizing SS-FGCC clustering results on simulated data.*

Table 5. *Comparing the ACC and NMI indices of SS-FGCC with other clustering algorithms.*

Algorithm	Segment	COIL100	USPS	Letter	Covertype
ACC					
RCut	0.3814	0.3488	0.5164	0.0697	OM
NCut	0.3948	0.4891	0.6636	0.1280	OM
FCM	0.6057	0.5012	0.6638	0.2552	0.2521
LSC	0.1883	0.3143	0.1704	0.1300	0.4878
DBSCAN	0.4208	0.1000	0.2291	0.1724	OM
SSFCM	**0.8351**	0.7806	0.8136	0.5554	OM
FFCAG	0.5552	0.4128	0.7456	0.1845	0.4399
CEHM	0.6557	0.4673	0.7057	0.2634	0.2896
EDCAG	0.2582	0.4390	0.2407	0.2019	0.1802
FGCC	0.7467	0.5959	0.7458	0.2978	0.4994
SS-FGCC	0.8205	**0.8276**	**0.8304**	**0.6349**	**0.5261**
NMI					
RCut	0.3713	0.5258	0.5303	0.0314	OM
NCut	0.3813	0.7105	0.6026	0.1248	OM
FCM	0.5782	0.7459	0.6034	0.3432	0.0620
LSC	0.0164	0.6365	0.0359	0.1503	0.0620
DBSCAN	0.4874	0.0000	0.1151	0.2844	OM
SSFCM	0.7410	0.8631	0.6755	0.5050	OM
FFCAG	0.4827	0.6418	0.7498	0.2225	0.0313
CEHM	0.6170	0.7250	0.7087	0.3659	0.1148
EDCAG	0.0895	0.6852	0.1270	0.2409	0.0035
FGCC	0.6947	0.7964	**0.7620**	0.4057	0.0871
SS-FGCC	**0.7085**	**0.9074**	0.6823	**0.5696**	**0.2557**

Table 6. *Running Time (seconds) of SS-FGCC with 10 different clustering algorithms.*

Algorithm	Segment	COIL100	USPS	Letter	Covertype
RCut	0.3541	12.6691	8.0346	66.3431	OM
NCut	0.4858	13.7435	8.9235	58.5404	OM
FCM	8.899	88738.4166	153.1335	1512.4324	5425.9390
LSC	0.9298	148.3063	3.7301	3.0656	58.5603
DBSCAN	0.1765	166.3921	5.0402	1.9717	OM
SSFCM	2.3491	100.2492	3.4734	**1.1026**	OM
FFCAG	2.9662	4.9767	3.3465	7.5806	2755.3006
CEHM	0.6661	88.3377	15.0175	19.2642	421.8937
EDCAG	0.1304	**1.9133**	**1.9939**	3.7947	1079.8925
FGCC	**0.0569**	3.8662	2.5071	7.0627	258.6514
SS-FGCC	0.788260	27.745786	2.9413	8.7037	**30.5952**

3.3.3. Comments and discussion

Accuracy (ACC): SS-FGCC demonstrates superior performance (Figure 3), notably on the Letter dataset (0.6349) over RCut/NCut. The semi-supervised mechanism boosts COIL100's ACC to 0.8276 (vs. FGCC's 0.5959). On the massive Covertype dataset, despite trailing FGCC (0.4007 vs. 0.4994), SS-FGCC surpasses FCM (0.2521) and completely eliminates the Out-of-Memory (OM) errors typical of traditional graph cuts.

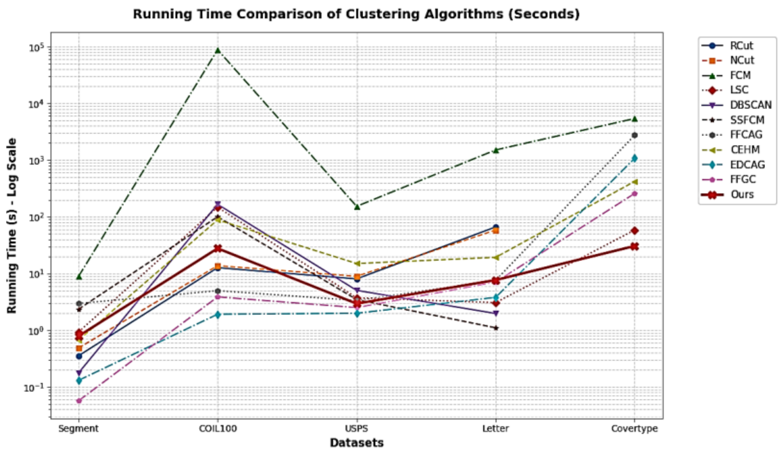

Figure 3. *The execution time (in seconds) of clustering algorithms is expressed on a base-10 logarithm scale.*

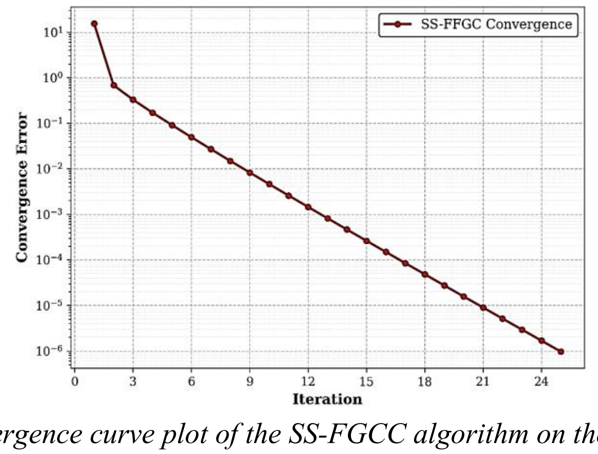

Figure 4. *Convergence curve plot of the SS-FGCC algorithm on the letter dataset.*

Data Structure Preservation (NMI): SS-FGCC achieved the highest NMI on 4 out of 5 datasets, peaking at 0.9074 on COIL100. On Covertype, it nearly doubled FGCC's NMI (0.2557 vs. 0.0871), highlighting its robust boundary definition despite noise. **Running Time**: FGCC exhibits excellent scalability (Figure 4), processing Covertype in 30.6s (8x faster than FGCC's 258.7s). The minor time increase on smaller sets like COIL100 (27.7s vs. 3.8s) is a necessary trade-off for the substantial leap in clustering quality.

3.3.4. Sensitivy analysis

To evaluate the stability and parameter dependence of the SS-FGCC algorithm, a sensitivity analysis was conducted on the Letter dataset. The investigation primarily focused on two core hyperparameters: the fuzzy regularization coefficient γ and the ratio of semi-supervised labeled anchors. **Fuzzy Regularization (γ):** Weak regularization ($\gamma = 10^{-3}$) causes extreme sparsity (ACC ≈ 0.40). ACC plateaus at ≈ 0.59 for $\gamma \geq 0.05$, validating our empirical baseline of $\gamma = 0.5$. **Labeled Anchor Ratio:** A 10% ratio optimally balances accuracy (ACC > 0.61, NMI ≈ 0.545) and manual annotation costs. Ratios $> 10\%$ restrict fuzzy propagation and risk boundary overfitting. **Complexity & Scalability:** Hard-labeling adds slight overhead on medium datasets, but anchor bipartite graphs completely bypass $O(n^3)$ spectral decomposition. This reduces computational and storage costs to $O(n \cdot m \cdot d)$ and $O(n \cdot m)$, respectively. Since $m, d \ll n$, practical complexity approaches $O(n)$. Consequently, SS-FGCC processes Covertype 8x faster than FGCC (30.6s vs. 258.7s) and completely eradicates Out-of-Memory (OOM) errors.

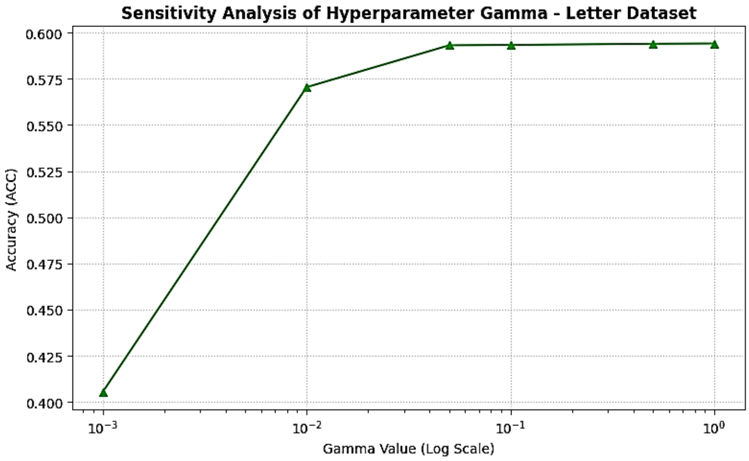

Figure 5. *Sensitivy analysis of gamma - letter dataset.*

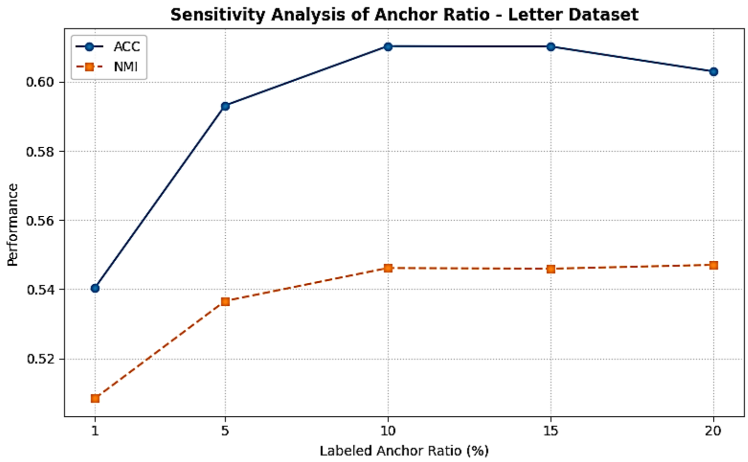

Figure 6. *Sensitivity analysis of anchor ratio - letter dataset.*

4. CONCLUSIONS

This paper presented the Semi-Supervised Fast Graph-Cutting Fuzzy Clustering (SS-FGCC) algorithm, which incorporates hard semi-supervised constraints into an anchor-based graph-cut framework. The proposed method directly solves the graph-cut problem without relaxation or post-processing while preserving linear computational and storage complexity. Experimental results demonstrate that SS-FGCC achieves superior clustering performance and robustness by effectively exploiting prior label information and reducing the bias of unsupervised optimization. Owing to its scalability and efficiency, SS-FGCC provides a promising solution for large-scale clustering tasks. Future research will investigate automated hyperparameter optimization and extensions to multi-view and unstructured graph data.

REFERENCES

[1]. Y.-C. Wei and C.-K. Cheng, "*Towards efficient hierarchical designs by ratio cut partitioning*", IEEE International Conference on Computer-Aided Design. Digest of Technical Papers, pp. 298–301, (1989).
[2]. J. Shi and J. Malik, "*Normalized cuts and image segmentation*", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, pp. 888–905, (2000).
[3]. J. C. Bezdek, "*Pattern recognition with fuzzy objective function algorithms*", Springer Science & Business Media, (2013).
[4]. Q. Qiang, B. Zhang, C. J. Zhang, Y. Hua and F. Nie, "*Fast fuzzy graph cut for clustering*", IEEE Transactions on Fuzzy Systems, (2025).
[5]. "*UCI Machine Learning Repository*", https://archive.ics.uci.edu/.
[6]. X. Chen and D. Cai, "*Large scale spectral clustering with landmark-based representation*", Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 25, No. 1, pp. 313–318, (2011).
[7]. J. Wang, Z. Ma, F. Nie and X. Li, "*Efficient discrete clustering with anchor graph*", IEEE Transactions on Neural Networks and Learning Systems, Vol. 35, No. 10, pp. 15012–15020, (2024).
[8]. M. Ester, H.-P. Kriegel and X. Xu, "*A density-based algorithm for discovering clusters in large spatial databases with noise*".
[9]. F. Nie, C. Liu, R. Wang, Z. Wang and X. Li, "*Fast fuzzy clustering based on anchor graph*", IEEE Transactions on Fuzzy Systems, Vol. 30, No. 7, pp. 2375–2387, (2021).
[10]. W. Pedrycz and J. Waletzky, "*Fuzzy clustering with partial supervision*", IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 27, No. 5, pp. 787–795, (1997).
[11]. F. Li et al., "*k-HyperEdge medoids for clustering ensemble*", Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 39, No. 17, pp. 18271–18278, (2025).

TÓM TẮT

Một hướng tiếp cận mới đối với bài toán phân cụm sử dụng phương pháp cắt đồ thị mờ bán giám sát

Các thuật toán phân cụm truyền thống dựa trên đặc trưng thường không thể nắm bắt một cách hiệu quả cấu trúc hình học của dữ liệu. Nhằm khắc phục hạn chế này, bài báo đề xuất một thuật toán phân cụm cắt đồ thị mờ bán giám sát (Semi-Supervised Fuzzy Graph Cut Clustering – SS-FGCC), trong đó tích hợp các nhãn của điểm neo vào quá trình phân cụm mờ và khai thác cấu trúc đồ thị nhằm nâng cao chất lượng phân cụm, đồng thời giảm độ phức tạp tính toán xuống thời gian tuyến tính. Các thực nghiệm được tiến hành trên 06 bộ dữ liệu tổng hợp và 05 bộ dữ liệu UCI cho thấy SS-FGCC đạt hiệu quả vượt trội so với các phương pháp RCut, NCut, FCM, FFCAG, FGCC và CEHM, với độ chính xác đạt tới 82,76%. Hơn nữa, phương pháp được đề xuất duy trì được tính ổn định khi xử lý các bộ dữ liệu quy mô lớn, trong khi các phương pháp cắt đồ thị truyền thống thường gặp phải những hạn chế về bộ nhớ. Do đó, SS-FGCC là một giải pháp phân cụm hiệu quả, có khả năng mở rộng và phù hợp với việc xử lý các bộ dữ liệu quy mô lớn

Từ khoá: Phân cụm; Học bán giám sát; Đồ thị; Đồ thị neo.