Density-Aware Pairwise Constraint Propagation via Bidirectional Trees

Zixuan Liu[1], Chuan Qin[2], Zhiguo Long[2], and Hua Meng[1]

[1] School of Mathematics, Southwest Jiaotong University, Chengdu 611756, China
[2] School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China

Abstract. Pairwise Constraints are widely used in semi-supervised learning. However, their limited quantity often restricts performance. To address this issue, pairwise constraint propagation (PCP) algorithms have been proposed to extend initial must-link and cannot-link constraints to more samples. In this paper, we propose an algorithm for density-aware pairwise constraint propagation via bidirectional trees to overcome the limitations of existing methods on complex manifold data. The approach constructs a hierarchical topology driven by local density, where each sample is associated with leader and follower points, forming bidirectional propagation trees. A sparse similarity matrix is built using both density differences and Euclidean distances, and a multi-path weighted mechanism computes propagation scores. Finally, an empirically selected optimal threshold is used to binarize the results and produce refined pairwise constraints. Experimental results demonstrate that the proposed method outperforms state-of-the-art PCP algorithms in terms of constraint accuracy and clustering performance.

Keywords: pairwise constraint propagation, constrained spectral clustering, semi-supervised learning

1 Introduction

Among various types of supervised information, class labels are commonly used and very informative. However, due to their high acquisition costs, it is often impractical to obtain large-scale labeled samples in many real-world scenarios. Consequently, researchers have increasingly focused on exploring more practical forms of weak supervision. This paper focuses on a widely used form of weak supervision, i.e., Pairwise Constraints [1] (PCs), which provide auxiliary supervised information by specifying the relationship between two samples and are generally easier and less costly to obtain. Currently, PCs have been widely applied in various machine learning tasks, including classification [2,3], clustering [4,5], feature selection [6], metric learning [7], image retrieval [8,9], and semi-supervised dimensionality reduction [10].

However, in practical applications, these PCs are still limited in quantity, and the performance of using only a few PCs is usually not satisfactory. To

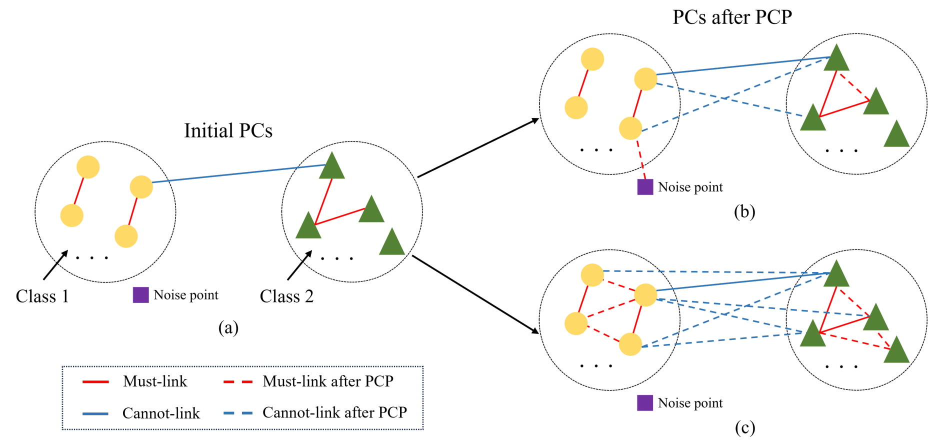

Fig. 1. Illustration of the advantage of the proposed method over existing ones. (a) Initial Pairwise Constraints Graph; (b) Pairwise Constraints Graph after Traditional PCP Method; (c) Pairwise Constraints Graph after Proposed PCP Method.

address this issue, researchers have proposed Pairwise Constraint Propagation (PCP) algorithms, which aim to extend limited initial constraints to a larger set of samples, thereby generating richer supervised information. For example, Wu et al. [11] incorporated non-negative matrix factorization into constraint propagation but considered only PCs (must-link constraints; MLs) that specifies which pairs of samples belong to the same class, neglecting PCs (cannot-link constraints; CLs) for samples that are not in the same class. Lu et al. [12] constructed separate propagation structures for ML and CL constraints. Given that CL constraints typically outnumber ML constraints significantly, Liu et al. [13] proposed a well-posed constrained optimization model to address this imbalance. Furthermore, Fu et al. [14] introduced low-rank matrix recovery and noise removal techniques into PCP, enhancing the robustness of the propagated constraints.

Although existing PCP methods have shown some success, they still struggle with data that exhibit complex manifold structures, such as face images, handwritten digits, and biomedical signals. Traditional global propagation approaches often propagate constraint information to irrelevant regions, causing error propagation and performance degradation. Meanwhile, local structures may be poorly modeled due to insufficient initial constraints, especially in sparsely labeled or noisy areas, as demonstrated in Figure 1.

To tackle these challenges, this paper proposes an algorithm for density-aware pairwise constraint propagation via bidirectional trees (DABT-PCP). The algorithm improves modeling of complex manifold structures by building a hierarchical propagation framework based on local density distribution. The key contributions are summarized as follows:

1. A constraint propagation algorithm that exploits local density distribution and leader-follower hierarchical relationships to better model manifold geometry and capture nonlinear structures;
2. A new propagation mechanism via leader/follower bidirectional trees, reducing global error spread while improving coverage in sparsely constrained regions.

The rest of the paper is organized as follows: Section 2 reviews related PCP work; Section 3 presents the DABT-PCP algorithm and its use in constrained spectral clustering; Section 4 reports experimental results; and Section 5 concludes the paper with future research directions.

2 Related Work

2.1 Pairwise Constraint Propagation

Pairwise Constraints (PCs) provide weak supervision by indicating whether two samples belong to the same class. However, in practice, the number of available PCs is often limited, which constrains the performance of semi-supervised learning methods. To overcome this, PCP methods have been proposed to automatically infer additional constraints, expanding the initial set to cover the entire dataset and improve model performance. This section briefly reviews and discusses several representative PCP approaches.

Firstly, we introduce three typical PCP algorithms: E2CP [15], MC-PCP [16], and SPCP [17]. E2CP, proposed by Lu et al. [15], treats ML/CL as "soft labels" and propagates them using graph Laplacian regularization. It is efficient and suitable for large datasets, iteratively optimizing the similarity matrix to meet constraints. MC-PCP, introduced by Yang et al. [16], models constraint propagation as a low-rank matrix completion problem, optimizing the similarity matrix via nuclear norm minimization. This method is robust to noise and missing data, making it ideal for high-dimensional datasets. SPCP, developed by Fu et al. [17], uses symmetric graph regularization to decompose constraint propagation into binary label propagation subproblems. This approach can be directly applied to multi-class tasks, enhancing its applicability. These methods effectively extend limited PCs to improve semi-supervised learning performance.

Among various PCP methods, some use optimization frameworks. Li et al. [18] modeled the problem as a semidefinite programming (SDP) task, creating a smooth mapping function that places ML-constrained samples close together and CL-constrained samples farther apart. Yang et al. [19] and Fu et al. [14] converted PCP into a matrix completion problem, treating the propagated PCs as partially observed matrices and using matrix analysis for effective algorithms. Jia et al. [20] proposes a novel PCP model via dual adversarial manifold regularization to fully explore the potential of the limited initial, considering the adversarial relationship between ML and CL. With advances in machine learning, deep learning approaches have also been introduced into PCP research [21,22], aiming to improve constraint propagation by leveraging the powerful feature learning capabilities of deep neural networks.

2.2 Constrained Spectral Clustering

Spectral clustering (SC) [23] is widely used in machine learning tasks due to its excellent clustering performance and straightforward implementation. Traditional SC is an unsupervised method that relies solely on the data's similarity matrix for clustering. To overcome the limitations of traditional SC in unsupervised learning, researchers have proposed constrained spectral clustering, which incorporates PCs to adjust the similarity or Laplacian matrix, ensuring that clustering results align with supervisory information. Existing constrained spectral clustering methods can be broadly categorized into two types [24]: low-dimensional embedding learning methods and affinity matrix modification methods.

The first type adjusts the structure of the low-dimensional embedding space (i.e., the eigenvectors of the laplacian matrix) to comply with given PCs. These methods typically integrate constraint information into the optimization objective function to directly learn a low-dimensional representation that satisfies the constraints. For example, Li et al.'s CCSP [25] obtains initial low-dimensional embeddings via spectral decomposition of the Laplacian matrix and then optimizes these embeddings to align with initial PCs. Cucuringu et al. [26] and Jiang et al. [27] reformulate the constrained clustering problem as a generalized spectral method (GSM), solving it mathematically to derive low-dimensional embeddings that satisfy PCs. This approach leads to semi-supervised SC (S3C) [28].

The second type of method modifies the similarity (affinity) matrix W directly, either by adjusting its elements or learning a new matrix that incorporates PCs, before applying standard spectral clustering. For example, graph/kernel learning methods [29][30] construct graphs or kernel matrices to represent data relationships, which are then used as affinity matrices in SC. To better utilize constraint information in guiding the clustering process, researchers have introduced constraint propagation mechanisms to refine the affinity matrix. For example. Fu et al. [14] introduced a PCP model with symmetric constraints, which was later extended using a joint low-rank mode [17] for improved constraint propagation. Jia et al. [31] introduced a unified PCP framework for constrained spectral clustering by simultaneously learning the propagation and affinity matrices. Specifically, the method is formulated as a bounded symmetric graph-regularized low-rank matrix completion problem, enabling more effective integration of PCs into the clustering process.

3 Method

The proposed method in this paper consists of three core stages: construction of density-aware topological structure, generation of bidirectional propagation trees, and the corresponding constraint propagation mechanism. The following sections provide detailed descriptions of these stages. The overall process of the algorithm is shown in Figure 2.

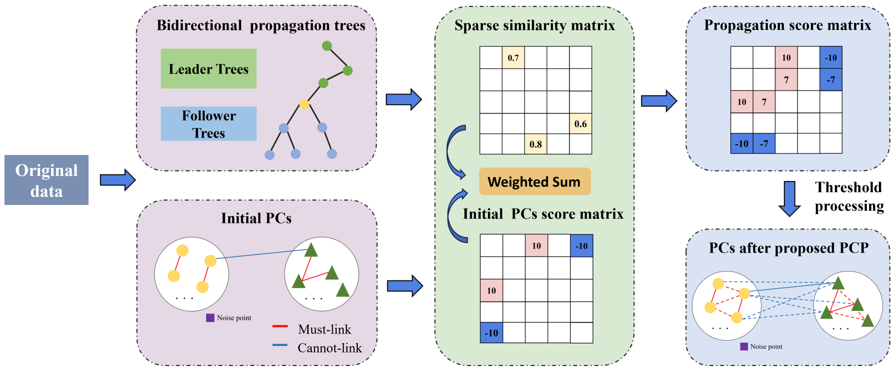

Fig. 2. The overall process of the proposed algorithm.

Let the input data matrix be $X = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^{d\times n}$, where each sample $x_i \in \mathbb{R}^{d\times 1}$ is a d-dimensional vector. The (sparse) similarity matrix is represented as $S \in \mathbb{R}^{n\times n}$,where the element S_{ij} denotes the similarity between samples x_i and x_j. The initial pairwise constraint score matrix is denoted by F, in which F_{ij} represents the initial constraint score between x_i and x_j. The propagated pairwise constraint score matrix is denoted by F'.

3.1 Construction of Density-Aware Topological Structure

The core task of this stage is to determine the "Leader Point" and "Follower Point" for each sample point. Specifically, for any sample point x_i, its leader point is defined as the nearest point with a higher density. The local density of each data point is calculated based on the average distance to its neighbors; a larger average distance indicates a more dispersed neighborhood, implying a lower local density.

Definition 1 (Local Density). *For a sample $x_i \in \mathbb{R}^d$, its* local density *ρ_i is defined as:*

$$\rho_i = exp(-(\frac{1}{k}\sum_{x_j\in\mathcal{N}_k(x_i)} d(x_i, x_j)^2)) \tag{1}$$

where $N_k(x_i)$ denotes the set of k-nearest neighbors of x_i,and $d(x_i, x_j)^2$ represents the squared Euclidean distance between x_i and x_j.

Definition 2 (Leader Point and Follower Point). *Let x_j be the data point that satisfies $\rho_j > \rho_i$ and is closest to x_i, Then x_j is referred to as the* leader point *of x_i, and x_i is the* follower point *of x_j, This relationship is formally defined as:*

$$leader(x_i) = \{x_j | arg \min_{x_j:\rho_j>\rho_i} (d(x_i, x_j))\} \tag{2}$$

Algorithm 1 Generate Leader Trees

Input: Data set $X = \{x_1, x_2, ..., x_n\}$
Output: Leader tree set $L = \{L_1, L_2, ..., L_n\}$
Initialization: For each data point x_i, create empty sets $L_i = \emptyset$.
for each data point x_i **do**
    $x_{\text{current}} \leftarrow \text{leader}(x_i)$
    if $x_{\text{current}} \in N_k(x_i)$ **then**
        $L_i \leftarrow L_i \cup \{x_{\text{current}}\}$
        $x_{\text{current}} \leftarrow \text{leader}(x_{\text{current}})$
return L

Algorithm 2 Generate Follower Trees

Input: Data set $X = \{x_1, x_2, ..., x_n\}$
Output: Follower tree set $U = \{U_1, U_2, ..., U_n\}$
for each $x_i \in X$ **do**
    Initialize empty queue Q and create empty sets $U_i = \emptyset$.
    for each x_j where $\text{leader}(x_j) = x_i$ and $x_j \in X$ **do**
        if $x_j \in N_k(x_i)$ **then**
            $U_i \leftarrow U_i \cup \{x_j\}$, $Q.\text{Enqueue}(x_j)$
    while Q is not empty **do**
        $x_k \leftarrow Q.\text{Dequeue}()$
        for each x_m where $\text{leader}(x_m) = x_k$ and $x_m \in X$ **do**
            if $x_m \in N_k(x_i)$ and $x_m \notin U_i$ **then**
                $U_i \leftarrow U_i \cup \{x_m\}$, $Q.\text{Enqueue}(x_m)$
return U

3.2 Generation of Bidirectional Trees

Unlike existing works, we construct bidirectional trees for constraint propagation, instead of just one directional of propagation. Bidirectional trees consist of two types of propagation paths for each sample point based on the density structure constructed in the previous section, i.e., *Leader Trees* and *Follower Trees*.

For each sample point x_i, its leader tree L_i is built by iteratively adding leader points of the added points in L_i that are among the k nearest neighbors, until no more such points can be found. Algorithm 1 outlines the detailed steps.

The follower tree of each sample point x_i is constructed similarly but in the reverse direction. That is, for each sample point x_i, its follower tree U_i is built by iteratively adding followers of the added points in U_i that are among the k-nearest neighbors of x_i, until no more such points can be found. This approach ensures constraint propagation in both directions and avoids erroneous diffusion associated with global propagation. Algorithm 2 gives the steps for the construction of follower trees.

Next, we construct a sparse similarity matrix S. For each data point x_i, we calculate its similarity to every point in its leader tree set L_i and follower tree set U_i. The similarity calculation formula is:

$$S_{ij} = \exp\left(-\frac{|\rho_i - \rho_j|}{\max(\rho)}\right) \cdot \exp\left(-\frac{d_{ij}^2}{\sigma_1^2}\right), x_j \in L_i \cup U_i \tag{3}$$

where ρ_i and ρ_j are the local densities of points x_i and x_j, respectively, and d_{ij} is the distance between points x_i and x_j. This similarity measure not only considers the distance between data points but also incorporates the factor of local density difference. This comprehensive consideration allows for a more accurate reflection of the actual similarity between data points.

3.3 Constraint Propagation Mechanism

With the bidirectional trees constructed, we can now proceed to the constraint propagation mechanism. We firstly define an initial pairwise constraint score matrix F based on the initial must-link $\mathcal{M}$ and cannot-link $\mathcal{C}$ constraints:

$$F_{ij} = \begin{cases} 10, & \text{if } (x_i, x_j) \in \mathcal{M}, \\ -10, & \text{if } (x_i, x_j) \in \mathcal{C}, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

For each data point x_i, all points that have an initial constraint relationship with x_i form a set $P(x_i) = \{x_{i,i_1}, x_{i,i_2}, \ldots, x_{i,i_p}\}$. That is, if $x_j \in P(x_i)$ then $F_{ij} \neq 0$, where p is the number of points satisfying this condition.

Subsequently, we calculate the constraint propagation scores with the help of the previously obtained similarity matrix S_{ij} in Eq (3). For a given data point x_j, take any point x_i from $P(x_j)$, and traverse all points x_k in its leader tree set L_i and follower tree set U_i. We use the sparse similarity as a weight value to compute the propagated information F'_{kj}. The calculation formula is:

$$F'_{kj} = S_{ik} \cdot F_{ij} \cdot \exp(-\frac{d_{kj}^2}{\sigma_2^2}), x_k \in L_i \cup U_i \tag{5}$$

where S_{ik} represents the similarity between points x_i and x_k, which measures the strength of constraint propagation between two points. The constraint information F_{ij} between x_i and x_j is propagated through x_i as a relay point to x_k; S_{ik} acts as the weight value during the propagation process, resulting in the constraint propagation score F'_{kj} between x_k and x_j. The term $\exp(-\frac{d_{kj}^2}{\sigma_2^2})$ further adjusts the propagation score based on the distance between x_k and x_j, ensuring that the propagation is appropriately attenuated with increasing distance.

Given that the constraint propagation path between x_k and x_j may involve multiple intermediate points, we comprehensively consider the contributions of multiple propagation paths to determine the final score. The formula is as follows:

$$F'_{kj} = (\frac{1}{p}\sum_{i=j_1}^{jp} S_{ik} \cdot F_{ij}) \cdot \exp(-\frac{d_{kj}^2}{\sigma_2^2})(x_k \in L_i \cup U_i) \tag{6}$$

Algorithm 3 DABT-PCP

Input: Data set $X = \{x_1, x_2, ..., x_n\}$
Output: Constraint matrices M, C
Procedure:
1: Construct leader-follower relationships via Eq (1) & Eq (2)
2: Generate leader trees and follower trees using Algorithm 1 & Algorithm 2
3: Build similarity matrix S according to Eq (3)
4: Initialize constraint score matrix F via Eq (4)
5: Compute propagated score matrix F' via Eq (6)
6: Obtain propagated constraints M, C through Eq (7) & Eq (8)
7: **return** M, C

We further requires that $F'_{kj} = F_{kj}$ if $(x_k, x_j) \in \mathcal{M} \cup \mathcal{C}$.

Depending on the positive or negative values of the elements, F' is divided into a must-link constraint score matrix F'^{+} and a cannot-link constraint score matrix F'^{-}. Subsequently, by setting a threshold τ for binarization of the scores, we obtain the propagated pairwise constraint information. The must-link constraint matrix M and cannot-link constraint matrix C are constructed by the following expressions respectively:

$$M_{ij} = \mathbb{I}\left(F'^{+}_{ij} > \tau\right) \tag{7}$$

$$C_{ij} = \mathbb{I}\left(F'^{-}_{ij} < \tau\right) \tag{8}$$

where $\mathbb{I}$ denotes the indicator function. Algorithm 3 gives the steps for the whole process.

3.4 Application in Constrained Spectral Clustering

This paper applies the proposed pairwise constraint propagation method to constrained spectral clustering [32]. Compared to normal spectral clustering, constrained spectral clustering further considers pairwise constraints to adjust the similarity or Laplacian matrix and requires that the clustering results align with supervised information.

Given an initial similarity matrix W, to improve constrained spectral clustering with pairwise constraints, inspired by literature [12], we update the initial similarity matrix W used in spectral clustering by incorporating the constraint information:

$$W^*_{ij} = \begin{cases} 1 - (1 - s_{ij} + d_{ij})(1 - w_{ij}) & s_{ij} - d_{ij} \geq 0 \\ (1 + s_{ij} - d_{ij})w_{ij} & s_{ij} - d_{ij} < 0 \end{cases} \tag{9}$$

where W^*_{ij} represents the $(i, j) - th$ element of the improved weight matrix $W^* \in \mathbb{R}^{n \times n}$. From the above formula, we can draw the following conclusions:

When $s_{ij} - d_{ij} > 0$, it suggests a must-link constraint between x_i and x_j, increasing w_{ij} so they are more likely to be clustered together. If $s_{ij} - d_{ij} < 0$,

Table 1. Details of the datasets.

Dataset	Samples(n)	Dimensions(d)	Classes(c)
Libras	360	90	15
UMIST	575	1024	20
MNIST	500	784	10
USPS	400	256	10
Yale	165	1024	15
ORL	400	5600	40

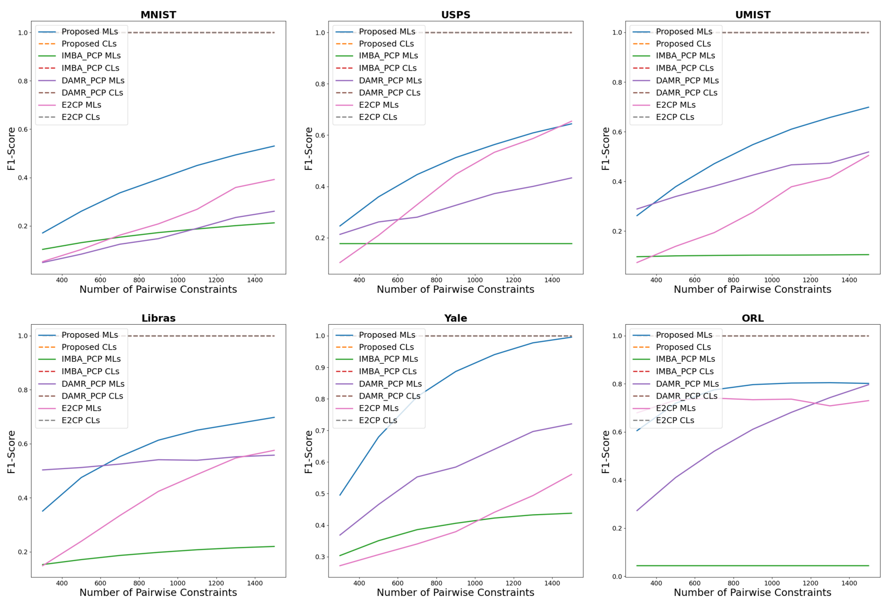

Fig. 3. Comparisons of the propagation capabilities of different PCP methods. Here, MLs and CLs refer to the must-link and cannot-link constraints, respectively.

it indicates a cannot-link constraint, decreasing w_{ij} so they are likely to be in different clusters. If $s_{ij} - d_{ij} = 0$, there is no clear constraint, and w_{ij} remains unchanged.

4 Experiments

In this section, we first evaluate the propagation performance of the proposed PCP model and then apply it to constrained spectral clustering. Experiments are conducted on six benchmark datasets of varying scales, as summarized in Table 1.

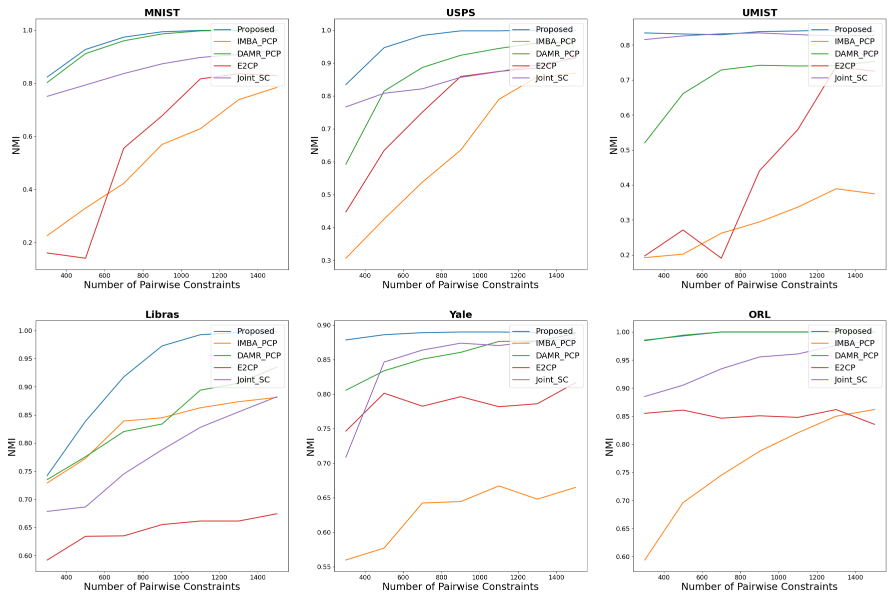

Fig. 4. Comparisons of the clustering performance by different semi-supervised clustering methods in terms of NMI.

4.1 Experimental Setup

To comprehensively evaluate the performance of the model, we randomly select initial PCs from real data based on label information. The number of constraint pairs ranges from 300 to 1500. To reduce the bias introduced by random sampling, each experimental scenario is repeated 20 times, with different initial PCs randomly selected each time, and the final results are averaged.

Besides that, we construct the Laplacian matrix L using the k-nearest neighbor graph method, where the value of k is set to $\lfloor \log_2 n \rfloor + 1$ [33]. Additionally, we use the RBF kernel, with its variance σ set as the average distance from each sample to its k nearest neighbors. For fair comparison, all methods involving graph matrices are configured using the aforementioned approach. The parameters for all comparative methods are determined through exhaustive search to ensure optimal settings.

4.2 Evaluation of Constraint Propagation Performance

To validate the advantages of the proposed model in constraint propagation capabilities, we compared it with several state-of-the-art PCP algorithms, including E2CP [15], IMBA-PCP [13], and DAMR-PCP [20].

Table 2. NMI values of different constrained clustering models.

Dataset	Number of PCs	IMBA-PCP	DAMR-PCP	E2CP	Joint-SC	Proposed
MNIST	300	0.22	**0.83**	0.7	0.75	0.82
	900	0.57	0.98	0.97	0.87	**0.99**
USPS	300	0.36	0.59	0.45	0.77	**0.84**
	900	0.59	0.9	0.84	0.84	**0.94**
UMIST	300	0.28	0.38	0.66	0.87	**0.96**
	900	0.38	0.86	0.89	0.91	**0.99**
Libras	300	0.73	0.14	0.59	0.68	**0.74**
	900	0.84	0.16	0.65	0.79	**0.97**
Yale	300	0.74	0.78	0.59	0.73	**0.99**
	900	0.89	0.99	0.70	**1**	**1**
ORL	300	0.59	**0.98**	0.70	0.88	**0.98**
	900	0.79	**1**	0.73	0.95	**1**

The adjusted similarity matrix is converted into binary constraints by applying a threshold, and these constraints are compared with the ground-truth constraints, the optimal threshold is determined through exhaustive search.

The commonly used F1 score is selected as the performance metric here, and we separate the F1 scores for must-link constraints and cannot-link constraints to avoid the imbalance between the two types of constraints. Figure 3 shows the performance of different PCP methods. Across all datasets, all methods achieve the highest F1 score on cannot-link constraints. This is because cannot-link constraints dominate the constraint set and are easier to identify during propagation. In contrast, must-link propagation is more challenging.

In terms of F1 scores on must-link constraints, the proposed method consistently outperforms existing methods, demonstrating not only stronger propagation capability but also a better ability to handle the imbalance between must-link and cannot-link constraints, leading to more accurate must-link propagation. Moreover, as the number of initial constraints increases, the F1 scores of most methods improve. This upward trend occurs because additional supervision provides more reliable anchor points for propagation, thereby reducing uncertainty and improving the precision of inferred constraints.

In conclusion, while preserving strong performance on cannot-link constraints, the proposed method significantly improves must-link propagation, highlighting its effectiveness in leveraging limited supervision under weakly supervised settings.

4.3 Evaluation of Constrained Clustering Performance

To comprehensively evaluate the performance in constrained clustering, we compare it not only with the three PCP methods mentioned earlier but also with another state-of-the-art constrained clustering algorithms, i.e., Joint-SC [31]. All PCP algorithms are integrated into the spectral clustering as described in

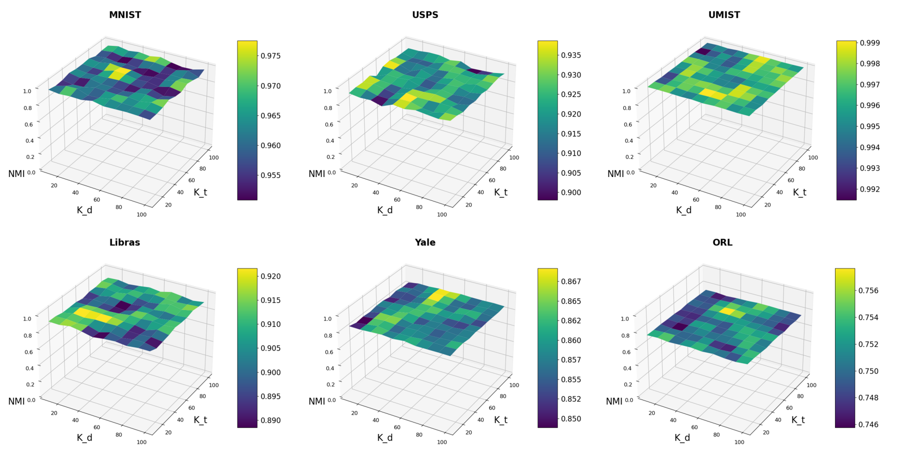

Fig. 5. Influence of neighborhood number on NMI value of constrained spectrum clustering.

Section 3.4, and the commonly used Normalized Mutual Information (NMI) is employed as the performance metric.

Figure 4 presents the clustering performance across six datasets. Table 2 records the NMI values of various constraint spectral clustering methods when the number of constraints is 300 and 900. The results show that our method achieves the highest NMI scores on nearly all datasets. Notably, on the Libras dataset, our model demonstrates a significant advantage over competing methods. This indicates that the bidirectional constraint propagation tree indeed improves the quality of constraint propagation, enabling more effective utilization of initial constraints to guide the clustering process.

It is worth noting that both the quality and quantity of initial PCs have a substantial impact on performance. Therefore, as shown in the figure, the NMI scores of all methods tend to increase with more initial constraints. Particularly on the UMIST, ORL, and Yale datasets, our model achieves high NMI even with limited initial supervision. Moreover, our method generally yields the largest performance gain in terms of NMI, further confirming the reliability and effectiveness of the PCs generated by the proposed model.

4.4 Parameter Analysis

We conducted a sensitivity analysis of the key parameters, K_d and K_t. In our proposed model, K_d is the number of nearest neighbors considered for local density calculation (i.e., the k value in Eq. 1), while K_t is the number of nearest neighbors considered for constructing leader trees and follower trees (i.e., the k value in Algorithm 1 and Algorithm 2).

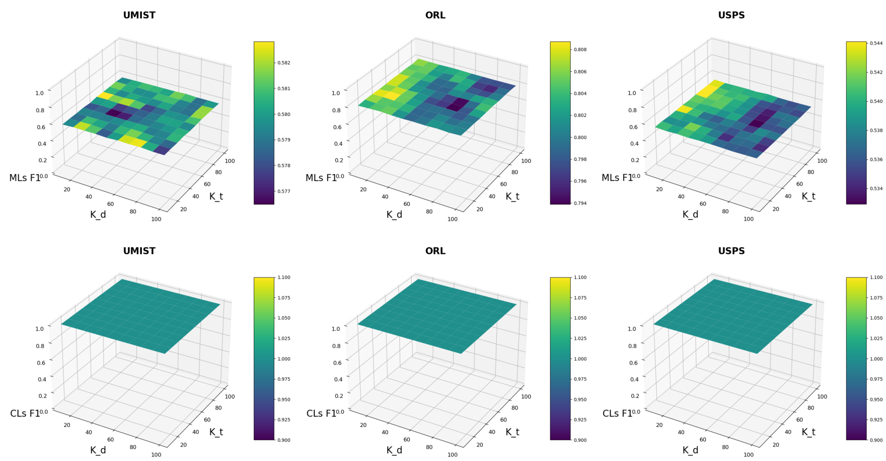

Fig. 6. Influence of neighborhood number on F1 value of propagation performance.

Figure 5 shows the results of NMI values across all datasets containing 600 initial PCs. Figure 6 presents the F1 scores of cannot-link and must-link constraints under the same conditions. It reveals that our model achieves optimal or near-optimal F1 and NMI values over a broad range of K_d and K_t settings. This indicates high robustness and adaptability of our model in both constraint propagation performance and clustering effectiveness.

5 Conclusion

This paper proposes a Density-Aware Pairwise Constraints Propagation via Bidirectional Trees (DABT-PCP) algorithm to better utilize limited initial pairwise constraints. The algorithm includes three main components: building a density-aware topology, generating bidirectional propagation trees, and applying an adaptive propagation mechanism. These components together improve performance on complex data compared to traditional methods. The key contribution is the integration of density-aware structures into constraint propagation, offering a new solution for semi-supervised learning on high-dimensional data. This approach effectively captures data structure and improves constraint utilization. Experiments show that DABT-PCP outperforms existing PCP methods in both constraint propagation and clustering accuracy. Future work will explore its use in tasks such as metric learning and constraint-based deep learning to further evaluate its potential.

References

1. González-Almagro, G., Peralta, D., De Poorter, E., et al.: Semi-supervised constrained clustering: An in-depth overview, ranked taxonomy and future research directions. Artif. Intell. Rev. 58, 157 (2025).
2. Cai, W., Gao, M., Jiang, Y., et al.: Hierarchical domain adaptation projective dictionary pair learning model for EEG classification in IoMT systems. In: IEEE Trans. Comput. Soc. Syst. 10(4), 1559?1567 (2022).
3. Wu, Z., Lu, Y., Chen, X., et al.: UC-OWOD: Unknown-classified open world object detection. In: Eur. Conf. Comput. Vis. LNCS, vol. 13657, pp. 193–210. Springer, Cham (2022).
4. Chen, R., Tang, Y., Zhang, W., et al.: Deep multi-view semi-supervised clustering with sample pairwise constraints. In: Neurocomputing 500, 832–845 (2022).
5. Cai, J., Hao, J., Yang, H., et al.: A review on semi-supervised clustering. In: Inf. Sci. 632, 164–200 (2023).
6. Lai, J., Chen, H., Li, T., et al.: Adaptive graph learning for semi-supervised feature selection with redundancy minimization. In: Inf. Sci. 609, 465–488 (2022).
7. Yang, Z., Lei, Y., Wang, P., et al.: Simple stochastic and online gradient descent algorithms for pairwise learning. In: Adv. Neural Inf. Process. Syst. 34, 20160–20171 (2021).
8. Xie, Y., Lin, Y., Cai, W., et al.: D3still: Decoupled differential distillation for asymmetric image retrieval. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit, 17181–17190 (2024).
9. Ge, C., Wang, J., Qi, Q., et al.: Scene-level sketch-based image retrieval with minimal pairwise supervision. In: Proc. AAAI Conf. Artif. Intell, 37(1), 650–657 (2023).
10. Zhou, M.C., Cui, M., Xu, D., et al.: Evolutionary optimization methods for high-dimensional expensive problems: A survey. IEEE/CAA J. Autom. Sinica, 11(5), 1092–1105 (2024).
11. Wu, W., Jia, Y., Kwong, S., et al.: Pairwise constraint propagation-induced symmetric nonnegative matrix factorization. In: IEEE Trans. Neural Netw. Learn. Syst. 29(12), 6348–6361 (2018).
12. Lu, Z., Ip, H.H.S.: Constrained spectral clustering via exhaustive and efficient constraint propagation. In: Eur. Conf. Comput. Vis. LNCS, vol. 6316, pp. 1–14. Springer, Berlin, Heidelberg (2010).
13. Liu, H., Jia, Y., Hou, J., et al.: Imbalance-aware pairwise constraint propagation. In: Proc. ACM Int. Conf. Multimedia, pp. 1605–1613 (2019).
14. Fu, Z.: Pairwise constraint propagation via low-rank matrix recovery. In: Comput. Vis. Media, 1, 211–220 (2015).
15. Lu, Z., Peng, Y.: Exhaustive and efficient constraint propagation: A graph-based learning approach and its applications. Int. J. Comput. Vis. 103(3), 306–325 (2013).
16. Yang, Z., Hu, Y., Liu, H., et al.: Matrix completion for cross-view pairwise constraint propagation. In: Proc. ACM Int. Conf. Multimedia, pp. 897–900 (2014).
17. Fu, Z., Lu, Z., Ip, H., et al.: Symmetric graph regularized constraint propagation. In: Proc. AAAI Conf. Artif. Intell, 25(1), 350–355 (2011).
18. Li, Z., Liu, J., Tang, X.: Pairwise constraint propagation by semidefinite programming for semi-supervised classification. In: Proc. Int. Conf. Mach. Learn, pp. 576–583 (2008).
19. Yang, Z., Hu, Y., Liu, H., et al.: Matrix completion for cross-view pairwise constraint propagation. In: Proc. ACM Int. Conf. Multimedia, pp. 897–900 (2014).

20. Jia, Y., Liu, H., Hou, J., et al.: Pairwise constraint propagation with dual adversarial manifold regularization. IEEE Trans. Neural Netw. Learn. Syst. 31(12), 5575–5587 (2020).
21. Tiezzi, M., Marra, G., Melacci, S., et al.: Deep constraint-based propagation in graph neural networks. IEEE Trans. Pattern Anal. Mach. Intell. 44(2), 727–739 (2021).
22. Marra, G., Tiezzi, M., Melacci, S., et al.: Local propagation in constraint-based neural networks. In: 2020 Int. Joint Conf. Neural Netw. (IJCNN), pp. 1–8. IEEE (2020).
23. Von Luxburg, U.: A tutorial on spectral clustering. In: Stat. Comput. 17, 395–416 (2007).
24. Wang, X., Qian, B., Davidson, I.: On constrained spectral clustering and its applications. Data Min. Knowl. Discov. 28, 1–30 (2014).
25. Li, Z., Liu, J.: Constrained clustering by spectral kernel learning. In: 2009 IEEE 12th Int. Conf. Comput. Vis, pp. 421–427. IEEE (2009).
26. Cucuringu, M., Koutis, I., Chawla, S., et al.: Simple and scalable constrained clustering: A generalized spectral method. In: Artif. Intell. Stat. PMLR, Proc. Mach. Learn. Res., vol. 51, pp. 445–454 (2016).
27. Jiang, C., Xie, H., Bai, Z.: Robust and efficient computation of eigenvectors in a generalized spectral method for constrained clustering. In: Artif. Intell. Stat. PMLR, Proc. Mach. Learn. Res., vol. 54, pp. 757–766 (2017).
28. Jia, Y., Kwong, S., Hou, J.: Semi-supervised spectral clustering with structured sparsity regularization. IEEE Signal Process. Lett. 25(3), 403–407 (2018).
29. Li, C.G., You, C., Vidal, R.: Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework. IEEE Trans. Image Process. 26(6), 2988–3001 (2017).
30. Jia, Y., Liu, H., Hou, J., et al.: Clustering-aware graph construction: A joint learning perspective. IEEE Trans. Signal Inf. Process. Netw. 6, 357–370 (2020).
31. Jia, Y., Wu, W., Wang, R., et al.: Joint optimization for pairwise constraint propagation. Trans. Neural Netw. Learn. Syst. 32(7), 3168–3180 (2020).
32. Wang, X., Qian, B., Davidson, I.: On constrained spectral clustering and its applications. Data Min. Knowl. Discov. 28, 1–30 (2014).
33. Von Luxburg, U.: A tutorial on spectral clustering. Stat. Comput. 17, 395–416 (2007).