A Group-Based Distance Learning Method for Semisupervised Fuzzy Clustering

Xuyang Jing, Zheng Yan, Senior Member, IEEE, Yinghua Shen, Witold Pedrycz, Life Fellow, IEEE, and Ji Yang

Abstract—Learning a proper distance for clustering from prior knowledge falls into the realm of semisupervised fuzzy clustering. Although most existing learning methods take prior knowledge (e.g., pairwise constraints) into account, they pay little attention to local knowledge of data, which, however, can be utilized to optimize the distance. In this article, we propose a novel distance learning method, which learns from the Group-level information, for semisupervised fuzzing clustering. We first present a new format of constraint information, called Group-level constraints, by elevating the pairwise constraints (must-links and cannot-links) from point level to Group level. The Groups, generated around data points contained in the pairwise constraints, carry not only the local information of data (the relation between close data points) but also more background information under some given limited prior knowledge. Then, we propose a novel method to learn a distance by using the Group-level constraints, namely, Group-based distance learning, in order to optimize the performance of fuzzy clustering. The distance learning process aims to pull must-link Groups as close as possible while pushing cannot-link Groups as far as possible. We formulate the learning process with the weights of constraints by invoking some linear and nonlinear transformations. The linear Group-based distance learning method is realized by means of semidefinite programming, and the nonlinear learning method is realized by using the neural network, which can explicitly provide nonlinear mappings. Experimental results based on both synthetic and real-world datasets show that the proposed methods yield much better performance compared to other distance learning methods using pairwise constraints.

Index Terms—Constraint weight, distance learning, Mahalanobis distance, neural networks, pairwise constraints, semisupervised fuzzy clustering.

I. INTRODUCTION

CLUSTERING is a general methodology and a remarkable algorithmic framework for data analysis and interpretation [1]. It aims to partition data into several clusters such that the data located in the same cluster are logically close to each other while the data in different clusters are highly distinct. Such a method is widely used in data mining [2], [3]; image processing [4], [5]; industrial data analysis [6], [7]; and other areas. Sometimes, we can provide some prior knowledge to guide the clustering process in order to obtain precise results being in rapport with the structure existing in data under analysis. This is the main goal of semisupervised clustering [8]. The prior knowledge mainly comprises [9] pairwise constraints (must-links and cannot-links); class labels; clusters’ position or identity; the size of clusters; proximity knowledge [10], [11]; and partition-level information [12], [13]. Among these forms of prior knowledge, the pairwise constraints are relatively easy to acquire and are mostly applied to clustering because they do not require users to have enough prior knowledge about the dataset. The pairwise constraints provide two types of data relationships: 1) must-link that states that the two data points should be assigned to the same cluster and 2) cannot-link that requests that the two data points should be assigned to different clusters.

Semisupervised fuzzy clustering, which can conveniently provide the description of real-world data and generate more meaningful data partition than hard clustering [1], has two ways to cooperate with pairwise constraints: 1) cost based and 2) distance based. In the first way, the objective function used in clustering is modified by adding two penalty terms that are calculated based on membership degrees [14], [15]. These two terms represent the cost incurred by violating must-links and cannot-links when partitioning the data. In the second way, a distance that satisfies the pairwise constraints is learned [16], [17]. Actually, it aims to learn a transformation that projects constrained links into a new space. In this way,
the data with must-links are as close as possible while the data with cannot-links are as far apart as possible. After that, all data will be transformed into the new space. This way can increase the separability between clusters, particularly, for the datasets that contain overlapping clusters. Learning a suitable distance is widely applied toward many popular data mining algorithms, for example, classification [18]–[20], clustering [21]–[23], dimensionality reduction [24], and kernel learning [25].

Learning a distance from pairwise constraints has attracted much attention and some remarkable methods have been proposed. Klein et al. [26] proposed a learning method that can adaptively adjust the distance according to the proximity matrix calculated from constraints. Xing et al. [22] formulated the distance learning process as a convex optimization problem and learned a global Mahalanobis distance based on pairwise constraints. Although these two methods are very popular, they exhibit worse performance than many recent methods and cannot achieve good clustering accuracy. Bar-Hillel et al. [27] proposed a relevant component analysis (RCA) for learning the Mahalanobis distance by only considering must-links. Hoi et al. [28] devised a discriminative component analysis (DCA) that considers both must-links and cannot-links to improve Bar-Hillel’s method. They learned the transformation matrix by maximizing the total variance of data in cannot-links while minimizing the total variance of data in must-links. Using the similar learning principle, Xiang et al. [29] used the ratio of distances of must-links and cannot-links as the objective function and presented an improved optimization process. However, these kinds of optimization methods involve a lot of computation when dealing with high-dimensional data, which highly decreases the clustering accuracy. Köstinger et al. [30] considered the independent generation processes for commonalities of must-links and cannot-links and designed a learning process based on the likelihood-ratio test. But this method fails to learn a proper covariance matrix when the number of constraints is small so that the learned distance has a bad performance. There are still some outstanding distance learning methods, such as LMNN [18], DMLMJ [20], ANMM [31], DML-eig [32], LDML [25], and class collapse [33], but they are more suitable for classification rather than clustering tasks because the learning process is initialized by finding some nearest neighbors with the same and distinct labels.

Although some current distance learning methods show great advance for optimizing fuzzy clustering, they still suffer from some problems that seriously impact their performance. First, the distance only learns from the provided constraints. In general, a good distance learned from pairwise constraints can distinctly aggregate must-links and separate cannot-links. The more background information we provide, the better the distance performs. As each constraint captures limited information about the data, it may suffer from such problems when being applied to unseen data, for example, overfitting. Second, the learned distance does not retain local information of the data, which may cause abrupt changes in the geometrical structure of data during the process of distance learning. Third, most existing methods neglect the weights of constraints [23]. They assume that all constraints share the same weights. However, the links that are located in dense regions are more informative than others, which would be very helpful for improving distance learning.

For solving the above problems, we are facing a number of challenges. The first one is how to mine sufficient background knowledge using the given limited prior knowledge and, at the same, time ensure the quality of distance learning. Most of the previous studies address this issue by adding more pairwise constraints into the learning process [29], [34]. Undoubtedly, this way needs a lot of human effort and increases the calculation cost of distance learning. Second, the local information of the data needs careful preservation because the local neighborhood relation may be changed during the distance learning process. Third, how to derive the importance degree from the graph of data for measuring the weights of constraints deserves a particular study.

In this article, we propose a new form of constraint information and a novel distance learning method for the purpose of solving the above issues and challenges in distance learning. First, with the purpose of mining more background knowledge and at the same time preserving the local information of data, we propose the notion of Group-level information by elevating the given pairwise constraints from point level to Group level. It contains specific background knowledge under the given limited number of constraints and offers special advantages for learning a proper distance. Then, by using Group-level information, a new distance learning algorithm, called Group-based distance learning, is proposed to enhance the performance of semisupervised fuzzy clustering. Both linear and nonlinear distance learning methods are proposed. In summary, the following major contributions of this article are worth stressing.

1) We propose a novel Group-based distance learning method, which learns from Group-level information, to improve the capabilities of fuzzy clustering. The Group that we originally propose in this article mines more background knowledge for clustering and retains local information around data points. Moreover, the learning method considers the importance degrees of constraints by assigning a weight to each constraint according to its location.

2) We design both linear and nonlinear Group-based distance learning (NLGDL) methods. Semidefinite programming is employed to learn the linear transformation by seeking a global positive semidefinite matrix. The nonlinear transformation is obtained by using the neural network, which provides explicit nonlinear mappings and does not have a scalability problem compared to kernel tricks. It highly improves the ability of clustering to deal with nonlinearly separable data.

3) Experimental results on both synthetic and real-world datasets indicate the proposed methods perform better than other state-of-the-art distance learning methods for clustering.

This article is organized as follows. In Section II, we give a brief review on related work, followed by the introduction of preliminaries in Section III. Then, in Section IV, we present the notion of Group-level information, which is originally
proposed in this article. Section V provides the details of linear and nonlinear Group-level distance learning methods, which are one of the first in the literature to learn distance from the view of Group level. The evaluation results of the proposed methods are shown in Section VI. Finally, conclusions are drawn in the last section.

II. RELATED WORK

Many efforts on semisupervised fuzzy C-means clustering (SFCM) have been proposed in the literature. Most of these clustering techniques are learned from pairwise constraints and aim to develop effective models for clustering data. There are two ways for improving SFCM in the presence of pairwise constraints.

The first is the cost-based methods. In such methods, the terms that measure the cost of violating pairwise constraints are directly added to the objective function of the clustering algorithm. This modified objective function guides the clustering process in the direction that the constraints require. Grira et al. [14] used the product of membership degrees of data points in must-links and in cannot-links to measure the violation cost. For the data pair in must-links (cannot-links), there is a small (large) violation cost if they are assigned to the same cluster. Gao and Wu [15] improved Grira’s method by involving the distance between the data point and cluster center into the penalty terms. Maraziotis [35] defined a new score metric called patterns constraints degree to calculate the degree of retained and/or violated for a specific pattern within a certain cluster. He then replaced the membership degree in the objective function by the score metric to guide the clustering process. Abin [36] combined possibilistic C-means with some multiple kernel settings to make the clustering algorithm learn from pairwise constraints. Mei [13] proposed a general form of violation cost for pairwise constraints in SFCM. He then incorporated partition-level information into penalty terms for clustering documents. However, the main problem of cost-based methods is that the penalty terms used to measure the cost of violating pairwise constraints are needed to be carefully designed. Such the problem may lead to generate some unexpected values of membership degree (e.g., negative values), making the cost-based methods lack robustness and decreasing the clustering accuracy. Moreover, the cost-based methods pay less attention to retain the local geometrical structure of data and the importance degrees of constraints, which are especially emphasized in this article.

The second is distance-based methods that replace the Euclidean distance used in the standard fuzzy C-means (FCM) by a new distance learned from pairwise constraints (e.g., Mahalanobis distance). Davis et al. [37] proposed an information theoretic-based distance learning (ITDL) method. They minimized the relative entropy between two multivariate Gaussians derived from the constraints. The learning process was transformed into a particular Bregman optimization problem by using the LogDet Bregman optimization. Qi et al. [38] designed the sparse distance learning (SDL) used in high-dimensional space. They employed two regularizations to learn a compact distance: the first is \( L_1 \)-penalization of the off-diagonal elements of the Mahalanobis matrix and the second one is a log-determinant divergence between the estimated and target distance matrix. Baghshah and Shouraki [34] proposed the generalized form of Xiang’s method [29] by additionally considering the topological structure of data. Liu et al. [39] learned a Mahalanobis matrix by minimizing a convex loss function corresponding to the sum of squared residuals of constraints. In their method, the constraints are represented in the form of the relative distance comparisons, which are especially useful where pairwise constraints are not natural to obtain. They also conducted a sparsity extension to improve the learning method when the data dimension is high and prior knowledge is limited. However, most of them learn the distance from provided limited prior knowledge and ignore the importance degrees of constraints. All these pieces of knowledge can be utilized to further optimize the distance. In addition, some methods lack the ability to deal with nonlinearly separable data since they only provide a linear learning algorithm. Recently, deep clustering, which employs neural networks to learn discriminative features for improving clustering, has attracted much attention. Ren et al. [40] first converted pairwise constraint information into a matrix and then combined the matrix with Kullback–Leibler divergence to form a loss function in order to train the neural network. The learned feature representations are utilized to improve clustering assignments. Li et al. [41] proposed a deep metric learning method by applying a convolutional neural network with a triplet loss function. The method can not only extract discriminative features to enhance the clustering but also have a label propagation strategy to increase the size of training data. However, both of them will face large computational consumption as the number of pairwise constraints increase. The way of mining more background information under the limited prior knowledge to achieve high improvement for clustering is also not considered in these two methods.

In summary, comparing with all current research, our method provides the following desirable properties: 1) mining more helpful information under the given limited prior knowledge for learning a proper distance based on Group-level information; 2) retaining local geometrical structure of data by adopting the local information around the given constraints during learning stage; 3) taking the importance degree of each constraint into consideration; and 4) learning the distance in both linear and nonlinear ways. All these properties could make efforts for improving the quality of distance learning.

III. PRELIMINARIES

In this section, some preliminaries related to our distance learning method are provided. The first is the Hausdorff distance applied to measure the distance between two Groups. The Hausdorff distance is applied here since it has an efficient calculation process and can accurately reflect the spatial property of individual data points by means of maximum–minimum distance calculation. It is widely used to measure dissimilarity between two point sets. The second is the notion of betweenness centrality that is used to calculate the weights of constraints according to the location in their neighborhood graph.
A. Hausdorff Distance

The Hausdorff distance measures the distance between two sets of data [42], [43]. It is an important metric that is commonly used in many applications, for example, face recognition [44] and object locating [45]. Given two sets \( A \) and \( B \), the Hausdorff distance between \( A \) and \( B \) involves the maximum–minimum (max-min) calculation

\[
H(A, B) = \max(h(A, B), h(B, A))
\]  

where \( h(A, B) \) and \( h(B, A) \) are the one-sided Hausdorff distances from \( A \) to \( B \) and \( B \) to \( A \), respectively. They are calculated as

\[
h(A, B) = \max_{a \in A} \min_{b \in B} d(a, b) \quad (2)
\]

\[
h(B, A) = \max_{b \in B} \min_{a \in A} d(a, b). \quad (3)
\]

d is a distance function, such as the Euclidean distance, and \( a \) and \( b \) represent the data points in \( A \) and \( B \), respectively.

\( h(A, B) \) first ranks each point in \( A \) according to its distance to the nearest neighbor in \( B \) and then takes the largest one as the one-sided Hausdorff distance [46]. As illustrated in Fig. 1, where \( \{a_1, a_2\} \in A \) and \( \{b_1, b_2, b_3\} \in B \), we obtain the smallest distance (the red line) from \( a_1 \) (\( a_2 \)) to the points in \( B \), namely, \( d(a_1, b_1) \) (\( d(a_2, b_3) \)). After comparing \( d(a_1, b_1) \) and \( d(a_2, b_3) \), we take the larger one as \( h(A, B) \). The calculation process of \( h(B, A) \) is similar as that of \( h(A, B) \). \( H(A, B) \) is the maximum of \( h(A, B) \) and \( h(B, A) \). It measures the degree of mismatch between two sets by considering the spatial position of each individual point in these two sets.

B. Betweenness Centrality

The betweenness centrality in the graph theory is used to measure the degree of centrality for a given data point based on shortest paths [47]. It plays an important role in analyzing computer networks [48], social networks [49], and other types of data models [50]. For a node \( v \) in a graph, its betweenness centrality is

\[
B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (4)
\]

where \( \sigma_{st} \) is the total number of shortest paths from node \( s \) to node \( t \), and \( \sigma_{st}(v) \) is the number of those paths that pass through \( v \). \( v \) has a high betweenness centrality if it lies on many shortest paths of other pairwise units. That means \( v \) controls the communication of the graph and has a considerable influence within the graph.

IV. Definition of Group-Level Information

In this section, we introduce the notion of Group-level information and discuss the generation process of Groups. First, let us recall the distance learning with the pairwise constraints. In that, the pairwise constraints we provide in advance are employed to learn an expected distance, which has the ability to decrease the distance among must-links and increasing the distance among cannot-links. After the learning stage, the distance between data points will be measured by this new distance. For a dataset of a given size, the more prior knowledge we offer, the better the learnt distance performs on this dataset. Although increasing the number of pairwise constraints is a feasible way (which incurs more human efforts to identify those data pairs), we are not sure how many pairwise constraints will satisfy the learning goal and the calculation consumption will increase accordingly. However, one can form this question in another way: find a solution that mines more background knowledge under the given limited number of constraints instead of increasing the number of constraints.

In this article, we consider the local information of data points (the relation between close data points) contained in pairwise constraints in order to mine more background knowledge under the given limited constraints. We propose Group-level information to realize this goal. Compared to point-level information that contains the information of each data point, the Group-level information carries the information of each Group, which is generated around each data point. The notion of Group is defined as follows.

Definition 1: The Group of a data point is a set that includes the data points whose distance to it is smaller than a threshold.

For illustration, the dotted circle in Fig. 2 explicitly shows the boundary of Group of \( x \in \mathbb{R}^d \), where \( d \) is the feature dimensionality. During the clustering process, \( x \) may be assigned to the cluster that its neighbors to. The closer they are, the more possible they are to be in the same cluster. Such a principle demonstrates that the nearest neighbors of \( x \) can be integrated with it to generate a compact representation, namely, the proposed Group, which captures local information and contains more data points that are not provided in the prior knowledge. Therefore, the data points contained in pairwise constraints could be represented by their Groups. The point-level information is elevated to the Group-level information by adopting the neighborhood information of each data point. Through using Group-level information, more background knowledge is mined under the limited number of provided constraints (more pairs of constraints are generated accordingly), which could be very helpful for improving the learning process of distance.
As illustrated in Definition 1, the value of the threshold, denoted as $\epsilon$, could be essential to construct the Group of $x$. It is a distance that used to select group members from nearest neighbors of $x$. For calculating $\epsilon$, we first search $k$ nearest neighbors of $x$. The search process will be stopped if $k$ neighbors are found or a data point that has a cannot-link relationship with $x$ is found. The number of searched neighbors of $x$ is denoted as $\hat{k}$, where $\hat{k} \in [1, k]$. $\epsilon$ is calculated as the average distance between $x$ and its $\hat{k}$ nearest neighbors

$$
\epsilon = \frac{\sum_{i=1}^{\hat{k}} d(x, n_i)}{\hat{k}}
$$

where $n_i$ is the $i$th neighbor of $x$, and it is the group member of $x$ if $d(x, n_i) < \epsilon$. The value of $\epsilon$ decides the size of Group. The larger the $\epsilon$ is, the bigger the Group may be. Actually, from formula (5), we can see that $\epsilon$ is determined by the number of nearest neighbors we searched. More searched neighbors will increase the value of $\epsilon$ and further make the Group become larger. A suitable size of Group is needed when learning the ideal distance from pairwise constraints. The impact of the number of nearest neighbors on the performance of the proposed distance learning method will be discussed in the experiment part.

V. GROUP-BASED DISTANCE LEARNING

In this section, we provide the details of Group-based distance learning methods. We first discuss the learning target of our methods in the presence of pairwise constraints on Group level. Then, we propose the linear Group-based distance learning (LGDL) method that aims to learn a global Mahalanobis matrix with the help of semidefinite programming. After that, we introduce the NLGDL method that is realized by employing the neural network. Furthermore, we discuss the optimization algorithms of the proposed learning methods.

A. Learning Target

Suppose $(x_i, x_j) \in$ ML and $(x_m, x_n) \in$ CNL, where ML and CNL represent must-links and cannot-links, respectively, their Groups are constructed according to the rule discussed in Section IV, which are denoted as $\tilde{x}_i$, $\tilde{x}_j$, $\tilde{x}_m$, and $\tilde{x}_n$. In the learning stage, each data point contained in constraints is replaced by its Group. Therefore, $\tilde{x}_i$ and $\tilde{x}_j$ ($\tilde{x}_m$ and $\tilde{x}_n$) holds a must-link (a cannot-link) relationship since the data points in them must be linked (cannot be linked), namely, $(\tilde{x}_i, \tilde{x}_j) \in$ ML ($\tilde{x}_m, \tilde{x}_n) \in$ CNL). With the purpose of pulling must-links close and pushing cannot-links far away, the target of Group-based distance learning is to minimize the distance between $\tilde{x}_i$ and $\tilde{x}_j$, while maximizing the distance between $\tilde{x}_m$ and $\tilde{x}_n$. Fig. 3 shows the ideal learning processes and results for the Group-level constraints $(\tilde{x}_i, \tilde{x}_j)$ and $(\tilde{x}_m, \tilde{x}_n)$.

B. Linear Group-Based Distance Learning

In the linear case, a linear transformation $L(\in \mathbb{R}^{d \times h})$ is learnt based on Group-level pairwise constraints. The transformed data are represented as $x_i^* = L^T x_i$, $x_j^* \in \mathbb{R}^h$, $i \in 1, 2, \ldots$. For two data points $x_i$ and $x_j$, their squared distance with regard to $L$ is represented as $d_i^2 = (x_i - x_j)^T LL^T (x_i - x_j)$. The formula below describes the loss function of the linear learning method. It directly reduces the Hausdorff distance between the Groups of must-links and uses a margin parameterized by a constant value to increase the Hausdorff distance between the Groups of cannot-links

$$
J_L = \frac{1}{|ML|} \sum_{(x_i, x_j) \in ML} w_{ij} * H_L(\tilde{x}_i, \tilde{x}_j)
$$

$$
+ \frac{1}{|CNL|} \sum_{(x_i, x_j) \in CNL} w_{ij} * [\lambda - H_L(\tilde{x}_i, \tilde{x}_j)]_+
$$

In formula (6):

1) the first item aims to pull the Groups of must-links as close as possible and the second item aims to push the Groups of cannot-links as far as possible using a margin;

2) $[z]_+ = \max(z, 0)$ is the hinge loss function. It monitors the inequality in the second item. If cannot-links have been pushed far away or they have a safe distance large than $\lambda$, the inequality does not hold. That is to say, under that condition, the second item makes no contributions to the loss function;

3) $|ML|$ and $|CNL|$ are the sizes of must-links and cannot-links. They make the must-links and cannot-links have the same weight to the overall loss function;

4) $\tilde{x}_i$ and $\tilde{x}_j$ are the Groups of $x_i$ and $x_j$. The group members of $x_i$ and $x_j$ will be not changed during the learning process;

5) $H_L(\tilde{x}_i, \tilde{x}_j)$ is the Hausdorff distance between $\tilde{x}_i$ and $\tilde{x}_j$ with regard to $L$. The distance between two data points that come from $\tilde{x}_i$ and $\tilde{x}_j$ is calculated as $d = (x_{r_i} - x_{p_j})^T LL^T (x_{r_i} - x_{p_j})$, where $x_{r_i} \in \tilde{x}_i$ and $x_{p_j} \in \tilde{x}_j$;

6) $w_{ij}$ is the constraint weight, which measures the importance of a constraint according to the location of data points contained in the constraints. The data points located in a density area have a higher importance degree than the data points located in a low density area. For the data pair $(x_i, x_j)$, the betweenness centrality $B(x_i)$ and $B(x_j)$ are calculated based on (4) from their $g$-neighborhood graph with an extra edge (with a weight $10^{-5}$) that connects $x_i$ and $x_j$ [23]. We define $w_{ij}$ as the sum of betweenness centralities of $x_i$ and $x_j$, namely,

$w_{ij} = B(x_i) + B(x_j)$.

Optimization: Gradient descent is the straightforward optimization method to minimize the loss function (6).
However, this method always obtains local optimization results and the initial value of \( L \) highly impacts the results because of the loss function is not convex with regard to \( L \). In order to obtain global optimal results, we formulate the optimization process of the loss function (6) into semidefinite programming. In such an approach, we aim to learn a Mahalanobis matrix according to the provided prior knowledge. The Mahalanobis matrix is a positive semidefinite matrix, which is defined as \( M = LL^T \). With this matrix, the squared distance between two data points is \( D_M(x_i, x_j) = (x_i - x_j)^T M (x_i - x_j) \).

The loss function (6) can be rewritten as

\[
J_M = \frac{1}{|ML|} \sum_{(x, x') \in ML} w_{ij} \ast H_M(\tilde{x}_i, \tilde{x}_j) + \frac{1}{|CNL|} \sum_{(x, x') \in CNL} w_{ij} \ast [\lambda - H_M(\tilde{x}_i, \tilde{x}_j)]_+.
\]  

(7)

The partial derivative of \( J_M \) with regard to \( M \) is

\[
\nabla_{M} J_M = \frac{1}{|ML|} \sum_{(x, x') \in ML} w_{ij} \ast H'_M(\tilde{x}_i, \tilde{x}_j) - \frac{1}{|CNL|} \sum_{(x, x') \in CNL} w_{ij} \ast dl \ast H'_M(\tilde{x}_i, \tilde{x}_j).
\]  

(8)

In (8), \( dl = [\lambda - H_M(\tilde{x}_i, \tilde{x}_j)]_+ \) is the derivative of hinge loss function and \( |0|_+ = 0 \). If the data pair in cannot-links do not violate the constraint, namely, \( \lambda - H_M(\tilde{x}_i, \tilde{x}_j) < 0 \), they have no contributions to update the matrix.

The Hausdorff distance between \( \tilde{x}_i \) and \( \tilde{x}_j \) is

\[
H_M(\tilde{x}_i, \tilde{x}_j) = \max \left( \min_{x_m \in x_i, x_j \in x_i} D_M(x_m, x_j), \min_{x_m \in x_j, x_j \in x_i} D_M(x_m, x_j) \right).
\]  

(9)

Although the Hausdorff distance is used to measure the distance between two \( Groups \), it still indicates the distance between two data points that come from different \( Groups \), as illustrated in Fig. 1. During the calculation process of the Hausdorff distance between \( \tilde{x}_i \) and \( \tilde{x}_j \), we can easily find these two points that measure the distance of \( \tilde{x}_i \) and \( \tilde{x}_j \) in order to simplify (9). We denote them as \( x_a \) and \( x_j \), where \( x_a \in \tilde{x}_i \) and \( x_j \in \tilde{x}_j \). Therefore, (9) is simplified as \( H_M(\tilde{x}_i, \tilde{x}_j) = D_M(x_a, x_j) \).

Next, we calculate the derivative of \( D_M(x_a, x_j) \) with regard to matrix \( M \) by

\[
\nabla_{M} D_M(x_a, x_j) = (x_a - x_j)(x_a - x_j)^T.
\]  

(10)

Finally, (8) can be expressed as

\[
\nabla_{M} J_M = \frac{1}{|ML|} \sum_{(x, x') \in ML} w_{ij} \ast \nabla_{M} D_M(x_a, x_j) - \frac{1}{|CNL|} \sum_{(x, x') \in CNL} w_{ij} \ast dl \ast \nabla_{M} D_M(x_a, x_j).
\]  

(11)

Algorithm 1 Optimization Process of LGDL

\begin{itemize}
  \item **Input:**
  \begin{itemize}
    \item Must-links, cannot-links, \( \lambda, \mu, \delta \), maximum iteration time;
  \end{itemize}

  \item **Output:**
  \begin{itemize}
    \item The Mahalanobis matrix \( M \);
  \end{itemize}

  \begin{enumerate}
  \item Initialize \( M \) as an identity matrix;
  \item Generate \( Groups \) for the data in must-links and cannot-links;
  \item Calculate constraint weight for each constraint;
  \item Repeat:
  \begin{enumerate}
  \item Find data points that measure the distance between \( Groups \) of each constraint with regard to \( M \);
  \item Compute the results based on formula (10);
  \item Calculate results of derivative for \( M \) based on formula (11);
  \item Update matrix \( M \) as in (12);
  \item Ensure the semi-definite characteristic of matrix \( M \) by formula (13);
  \end{enumerate}
  \item Until: reach the termination condition;
  \item Return \( M \)
  \end{enumerate}

\end{itemize}

At the \( t \)-th iteration time, \( M_t \) is updated by

\[
M_t = M_{t-1} - \mu \frac{\partial J_M}{\partial M_{t-1}}
\]  

(12)

where \( \mu \) is the learning rate. In order to speed the convergence rate and ensure the direction of optimization process, at each iteration, we increase \( \mu \) by a factor of 1.1 if the loss function decreases and decrease \( \mu \) by a factor of 0.5 if the loss function increases. The optimization process will terminate under the condition \( |J_M - J_{M_{t-1}}| < \delta \) or the number of iteration reaches the maximum value.

Notably, \( M_t \) should be a semidefinite matrix during the optimization process [18]. To satisfy this condition, at the end of iteration, we need to check the characteristic of the semidefinite of \( M_t \). We conduct an eigen decomposition of \( M_t \) as \( M_t = E \Delta E^T \), where \( E \) is the orthonormal matrix of eigenvectors and \( \Delta \) is the diagonal matrix of the corresponding eigenvalues. We decompose \( \Delta \) into the positive and negative matrix, namely, \( \Delta = \Delta^+ + \Delta^- \), where \( \Delta^+ \) and \( \Delta^- \) have the positive and negative eigenvalues, respectively. The semidefinite matrix that considered as the initial matrix for the next iteration is given by

\[
M_t^+ = E \Delta^+ E^T.
\]  

(13)

Equation (13) is used to ensure the positive semidefiniteness of \( M_t \) in the learning stage. After projecting \( M_t \) into the cone of positive semidefinite matrices, the optimization process of the linear method still converges [18].

The optimization process of the linear learning method is described in Algorithm 1. After obtaining the optimal Mahalanobis matrix \( M_t \), fuzzy clustering is performed by employing the new distance function \( D_M(x_a, x_j) \).

C. Nonlinear Group-Based Distance Learning

To realize the goal of dealing with nonlinearly separable data, we propose NLGDL. Usually, kernel tricks are
employed to achieve this ability. They project data into a high-
dimensional space and distance learning is conducted in that
space. However, they only provide implicitly nonlinear mapp-
ings and sometimes they face the problem of scalability and
overfitting [51]. In order to learn a more flexible one that can
fit the data in a better way, we employ the neural network to
explicitly learn the nonlinear transformation.

The architecture of the neural network used in our learning
method is shown in Fig. 4. It is comprised of two identi-
cal feedforward networks that share the same weights. The
symmetrical structure used here is to guarantee the learning
results will not be impacted by the input order and ensure
learning mappings is the same. Each network has \( l \) layers and \( q_n \) neurons. The output of the \( n \)th layer is
\[
x_n = \varphi \left( w_n x_{n-1} + b_n \right)
\]  
where \( w_n \in \mathbb{R}^{q_n \times q_{n-1}} \) and \( b_n \in \mathbb{R}^{q_{n-1}} \) are the weight matrix and bias vector of the \( n \)th layer, \( x_{n-1} \) is the output of last layer, \( \varphi \) is the activation function [e.g., sigmoid, rectified linear unit
(ReLU), and tanh].

For the first layer, we set the inputs of these two networks
as the data points that measure the Hausdorff distance between
\( \mathbf{x}_1 \) and \( \mathbf{x}_2 \), denoted as \( x_1 \in \mathbb{R}^b \) and \( x_2 \in \mathbb{R}^b \). Accordingly, the
outputs of the neural network are represented as \( x_1' \in \mathbb{R}^b \) and \( x_2' \in \mathbb{R}^b \). The loss function of the neural network is given as
\[
J = \frac{1}{\text{ML}} \sum_{(x_i, x_j) \in \text{ML}} w_{ij} \ast d^2(x_i', x_j')
+ \frac{1}{\text{CNL}} \sum_{(x_i, x_i) \in \text{CNL}} w_{ii} \ast \left[ \lambda - d^2(x_i', x_i') \right].
\]  

Each item has the same meaning as in (6). The weight updates of the neural network are calculated based on the
loss gradient by using a backpropagation scheme with an
adaptive moment estimation (Adam) optimizer [52]. In a for-
dward pass, the data points are continuously fed into the
network and the loss function is updated accordingly. In a
backward pass, for example, the weight update of \( w_n \) is
given by \( \Delta = -\eta \frac{\partial J}{\partial w_n} \), where \( \eta \) is a learning rate. The optimization process is terminated if \( |J_i - J_{i-1}| < \delta \) or the
number of iterations has reached a preset threshold.

The optimization process of the nonlinear learning method
is described in Algorithm 2. After the learning stage, the data
can be fed into one of the two identical networks. Then, fuzzy
clustering is performed over the transformed data.

VI. EXPERIMENTS AND PERFORMANCE EVALUATION

In this section, we conduct a series of experiments on
synthetic, UCI, and handwritten datasets to evaluate the
performance of the proposed linear and nonlinear methods
by comparing with other state-of-the-art distance learning
methods. All methods were implemented with the Python pro-
gramming language by calling the metric-learn library [53] to
conduct a comparative analysis and the Pytorch library [54]
to implement nonlinear learning method. First, we give the
details of experimental settings. Next, we test the impact of
parameters on our proposed learning methods. The experimen-
tal results along with essential discussions on the performance
of the proposed methods are provided in the last. We have
released the source code of our methods at GitHub [55].

A. Experimental Settings

1) Evaluation Criteria: In our experiments, several eval-
cuation criteria are employed to measure the performance of
fuzzy clustering under different distances.

1) Adjusted Rand Index (ARI) [56]: It evaluates the
agreement between the true partition of data \( C = \{c_1, c_2, \ldots, c_l\} \) and the experimental partition of data
\( \Omega = \{w_1, w_2, \ldots, w_q\} \) obtained from the evaluated
clustering algorithms. Let \( N_i \) be the number of data
points that appear in cluster \( i \) in \( C \) and in cluster \( j \) in \( \Omega \).
ARI is calculated by
\[
\text{ARI}(\Omega, C) = \frac{R - E(R)}{\text{Max}(R) - E(R)}
\]  
where \( R = \sum_{i} \left( \frac{N_i}{2} \right), E(R) \) is the excepted value of \( R, \)
and \( \text{Max}(R) \) is maximum value of \( R \).

2) Purity (PUR) [57]: It calculates the average accuracy of
correctly assigning the dominating class in each cluster.
These are the prominent methods that learn the distance with the following distance learning methods: {3, 6, 9}, {1, 2, 3, 4}, and {4, 6, 8, 10}. Images from digit 1 to 10, including {1, 7}, {5, 8}, {2, 4, 7}, {3, 6, 9}, {1, 2, 3, 4}, and {4, 6, 8, 10}.

We generate six subsets by randomly selecting 500 images. Each image can be represented as a 256-D feature vector. We contain 9298 handwritten digit images with the size 16×16 pixels. These datasets are challenging for some distance learning methods because they are nonlinearly separable.

Figure 9. All data points of the same class are denoted by the same color and style. These datasets have been widely used as benchmarks in machine learning. All data features are normalized into [-1, 1] using the maximum–minimum normalization method in order to avoid the influence caused by the scale of each feature for the distance learning. The numbers of must-links and cannot-links are equal when selecting the constraints from the labeled data in the datasets.

The first type is some synthetic datasets in R², includes 3-class data, double-moons data, and double-circles data, as shown in Fig. 9. All data points of the same class are denoted by the same color and style. These datasets are challenging for some distance learning methods because they are nonlinearly separable.

The second type is some publicly available datasets obtained from the UCI machine-learning repository [59]. Table I shows the information about the number of data points (#data points), the number of data features (#features), and the number of classes (#classes) of these datasets.

The third type is about the USPS handwritten dataset [60]. It contains 9298 handwritten digit images with the size 16×16 pixels. Each image can be represented as a 256-D feature vector. We generate six subsets by randomly selecting 500 images from digit 1 to 10, including {1, 7}, {5, 8}, {2, 4, 7}, {3, 6, 9}, {1, 2, 3, 4}, and {4, 6, 8, 10}.

3) Comparative Analysis: We compare our linear and nonlinear methods with the following distance learning methods. These are the prominent methods that learn the distance from pairwise constraints. All these methods use the same constraints that are randomly selected from the data. The experimental results are reported in the context of FCM clustering. We give the average results for each method over 20 runs in order to minimize the impact that brings by a random selection of constraints. We use the ReLU function as the activation function of the nonlinear method throughout the experiments.

1) FCM without distance learning.
2) FCM with the ITDL [37].
3) FCM with the Mahalanobis distance learning from side information (MDL) [29].
4) FCM with SDL [38].
5) FCM with least squared-residual distance learning (LSDL) [39].
6) FCM with the distance learning from constraints (DLCs) [23].
7) FCM with the proposed LGDL.
8) FCM with the proposed NLGDL.

### Experimental Results on Parameter Configuration

In this section, we investigate the impact of some important parameters on the performance of the proposed methods over six datasets with different size and dimensionality. The number of pairs of constraints is set to 30 for all datasets. In this article, we focus on studying the performance of Group-based distance learning on traditional FCM. We set the dimensionality of transformed data in the linear method and the output dimensionality of the neural network in the nonlinear method as the dimensionality of the original (input) data, namely, \( d = h \).

We provide the experimental results with regard to ARI, NMI, and PUR of FCM with LGDL and FCM with NLGDL.

1) Impact of \( l \) and \( q \): The number of hidden layers \( l \) and the number of neurons in each hidden layer \( q \) determine the depth and width of the neural network employed in NLGDL. Both parameters affect the effectiveness of the neural network. Figs. 5 and 6 show the evaluation results of the impact of \( l \) and \( q \), respectively. As illustrated by the results, appropriately increasing the value of \( l \) and \( q \) can improve the performance of NLGDL because ARI, NMI, and PUR become higher at first. However, continuously increasing \( l \) and \( q \) degrades the performance.

| Dataset | #data points | #features | #classes |
|---------|-------------|-----------|----------|
| Iris    | 150         | 4         | 3        |
| Banknote| 1372        | 4         | 2        |
| Seeds   | 210         | 7         | 3        |
| Glass   | 214         | 9         | 6        |
| Magic   | 19020       | 10        | 2        |
| Wine    | 178         | 13        | 3        |
| Segment | 2310        | 18        | 7        |
| Waveform| 5000        | 21        | 3        |
| Breast  | 569         | 30        | 2        |

**Table I Information on UCI Datasets**

\[
\text{PUR}(\Omega, C) = \frac{1}{N} \sum_{j=1}^{k} \max_{i} M_{ij} \tag{17}
\]

where \( M_{ij} = |w_i \cap c_j| \) is the number of common data points of \( \Omega \) and \( C \), and \( N \) is the number of data points.

3) **Normalized Mutual Information (NMI)** [58]: It computes the dependence between the experimental partition of data and true partition of data under the independence assumption. NMI is computed as

\[
\text{NMI}(\Omega, C) = \frac{I(\Omega, C)}{\sqrt{H(\Omega)H(C)}} \tag{18}
\]

where \( I(\Omega, C) \) is the mutual information and \( H(\Omega) \) and \( H(C) \) are the entropies of \( \Omega \) and \( C \), respectively.

All criteria assume the value lying in the range of \([0, 1]\), with 1 denoting perfect clustering (in terms of the assumed performance index).

2) Datasets: Three kinds of datasets are used in our experiments. Such datasets have been widely used as benchmarks in machine learning. All data features are normalized into \([-1, 1]\) using the maximum–minimum normalization method in order to avoid the influence caused by the scale of each feature for the distance learning. The numbers of must-links and cannot-links are equal when selecting the constraints from the labeled data in the datasets.

The first type is some synthetic datasets in \( \mathbb{R}^2 \), includes 3-class data, double-moons data, and double-circles data, as shown in Fig. 9. All data points of the same class are denoted by the same color and style. These datasets are challenging for some distance learning methods because they are nonlinearly separable.

The second type is some publicly available datasets obtained from the UCI machine-learning repository [59]. Table I shows the information about the number of data points (#data points), the number of data features (#features), and the number of classes (#classes) of these datasets.

The third type is about the USPS handwritten dataset [60]. It contains 9298 handwritten digit images with the size 16×16 pixels. Each image can be represented as a 256-D feature vector. We generate six subsets by randomly selecting 500 images from digit 1 to 10, including {1, 7}, {5, 8}, {2, 4, 7}, {3, 6, 9}, {1, 2, 3, 4}, and {4, 6, 8, 10}.

Fig. 5. Impact of \( l \) on the nonlinear method. (a) Iris. (b) Banknote. (c) Wine. (d) Breast cancer. (e) USPS 1, 7. (f) Moon.
performance of NLGDL since such an increase makes the neural network learn some noise. Considering the experimental results and the effectiveness of the neural network, we select \( l = 2 \) and \( q = 200 \) as their default values.

2) Impact of \( k \): \( k \) is the number of searched nearest neighbors during the generation process of the Group. It determines the size of the generated Group and also influences the purity of each Group. Fig. 7 shows the impact of \( k \) on the performance of LGDL and NLGDL. The experimental results (ARI, NMI, and PUR) are better when setting \( k \) to some small values than the results under large values of \( k \). With increasing the value of \( k \), some data points that belong to different classes may be included in the Group. Such a situation will reduce the purity of Group and further impact the learning process of distance. Taking the generalization and accuracy into account, we set the default value of \( k \) to 3 in our experiments.

3) Impact of \( g \): \( g \) is the number of neighbors when generating the neighborhood graph for calculating the importance degree of constraints. Fig. 8 demonstrates the experimental results (ARI, NMI, and PUR) with different values of \( g \). From the results, we can see that both small and large values of \( g \) do not result in a good transformation of data. When \( g \) is set to a large value, many nodes in transition regions and boundaries will be included in the neighborhood graph. Such information is useless for calculating the importance degree of constraints because the degree is calculated based on the popularity of the nodes. The constraint has a large importance degree if it contains some popular nodes that locate in the density area. In contrast, small values of \( g \) make it difficult to appropriately model the neighborhood graph. According to the experimental results, we select \( g = 5 \) as its default value in our experiments.

C. Experiments on Synthetic Data

Fig. 9 demonstrates the explicit transformation results of LGDL and NLGDL on three synthetic datasets. The number of constraints is set to 20 for all datasets. As illustrated by the results, NLGDL produces a good transformation of the data onto a line and perfectly separates the clusters for all the synthetic datasets. Based on the learning target, NLGDL makes the data points in the same class more compact and pushes the data points in different classes farther away. All these expected targets are clearly visible in Fig. 9. The linear
method only changes the scale of the data and does not do any transformations. We do not report the transformation results of other distance learning methods because they have similar results as being generated by the linear method.

D. Experiments on UCI Datasets

For the experiments on UCI datasets, we compare our work with others under different numbers of constraints. We set the number of constraints from 20 to 100 and randomly select them from the labeled data.

Fig. 10 displays the experimental results about ARI, NMI, and PUR of each method under different number of constraints. According to the results, we can see that the proposed LGDL (the blue line) and NLGDL (the red line) methods perform well on most datasets, especially on (a), (b), (e), and (f). Sometimes, our methods compete with ITDL with regard to ARI and NMI. However, comparing with ITDL, LGDL, and NLGDL are more stable and achieve high ARI, NMI, and PUR at the same time for all datasets. For MDL, its performance is getting worse with the increase of data dimensionality. SDL makes a poor performance on these datasets since it is tailored for only high-dimensional data, which makes it lack generality. LSDL also has some dissatisfied results because it needs more background knowledge to learn the relative comparisons. DLC has a relatively stable performance over the datasets. The results also demonstrate that LGDL and NLGDL have some growth in ARI, NMI, and PUR by increasing the number of constraints. All these comparisons illustrate that our proposed methods can make full use of constraint information and mine more background knowledge to improve the performance of FCM.

For obtaining an accurate comparison, we conduct a paired $t$-test with significance level 0.05 to evaluate the experimental results shown in Fig. 10 under 40 constraints. Such a method has been employed in many works [23], [38]. Table II provides the comparison results of ARI, NMI, and PUR. In the table, the symbol “$\sim$” indicates that the experimental results of two compared methods are not significantly different under the given confidence level and the symbol “$<$” denotes that the result of the latter method is significantly higher than the result of the former one. From the comparison results of the

---

**TABLE II**

| Datasets | Criteria | Paired $t$-test |
|----------|----------|----------------|
| Iris     | ARI      | $S < LS \sim Eu < IT \sim M < D \sim LG \sim NLG$ |
|          | NMI      | $LS < S < Eu < D < IT \sim NLG \sim M < LG$ |
|          | PUR      | $S < LS < Eu < IT < D \sim LG \sim NLG$ |
| Banknote | ARI      | $Eu < S < IT < LS \sim M \sim D \sim LG \sim NLG$ |
|          | NMI      | $Eu < S < IT < D < LS \sim LG \sim M \sim NLG$ |
|          | PUR      | $Eu < S < D < IT < M \sim LS \sim LG < NLG$ |
| Seeds    | ARI      | $D < S < M \sim NLG \sim LS \sim IT < Eu \sim LG$ |
|          | NMI      | $NLG \sim M < S \sim LS \sim IT < Eu < LG < D$ |
|          | PUR      | $S < LS < NLG < D < M \sim IT < Eu < LG < LG$ |
| Glass    | ARI      | $S < LS < M \sim D < Eu \sim LG \sim NLG$ |
|          | NMI      | $LS < S < D < M \sim Eu \sim LG \sim NLG < IT$ |
|          | PUR      | $S \sim LS < M \sim IT \sim D \sim NLG < Eu < NLG$ |
| Magic    | ARI      | $Eu \sim M < S < IT < D < LS \sim LG < NLG$ |
|          | NMI      | $M \sim C < LS \sim D < Eu < IT < LG \sim NLG$ |
|          | PUR      | $LS < M \sim D < S < IT < Eu \sim LG \sim NLG$ |
| Wine     | ARI      | $M \sim C < LS \sim D < Eu < IT \sim LG \sim NLG$ |
|          | NMI      | $M \sim S < Eu < D \sim LG < NLG < D < IT \sim LG < NLG$ |
|          | PUR      | $LS < M \sim S < Eu \sim D < IT \sim LG \sim NLG$ |
| Segment  | ARI      | $M \sim LG < D \sim S < Eu \sim LG \sim NLG \sim IT < NLG$ |
|          | NMI      | $S \sim M \sim Eu < LS \sim NLG < C \sim IT \sim LG < NLG$ |
|          | PUR      | $S \sim LS < M \sim Eu \sim LG \sim NLG < D \sim IT < LG < NLG$ |
| Waveform | ARI      | $M \sim LG < D \sim S < Eu \sim IT \sim LG < NLG$ |
|          | NMI      | $M \sim LS \sim S < D \sim Eu \sim IT \sim LG < NLG$ |
|          | PUR      | $M \sim LS \sim Eu \sim D \sim IT < LG < NLG < D \sim IT < LG < NLG$ |
| Breast   | ARI      | $S < LS < M \sim IT < D \sim LG < NLG$ |
|          | NMI      | $S < IT \sim Eu \sim LG < M \sim D \sim LG < NLG$ |
|          | PUR      | $LS < S < Eu \sim M \sim IT < D \sim LG < NLG$ |
| Eu: Euclidean; IT: ITDL; M: MDL; S: SDL; LS: LSDL; D: DLC; LG: LGDL; NLG: NLGDL.
paired $t$-test, we can see that LGDL and NLGDL outperform other methods with a 95% confidence level in most of the tested datasets.

### E. Experiments on USPS Datasets

Fig. 11 shows the mean and variance of the experimental results about ARI, NMI, and PUR on USPS subsets under the specified number of 40 constraints. From the results, we observe that the proposed NLGDL consistently obtains better results than other methods for all the subsets. The powerful learning ability of the neural network makes NLGDL always achieve high ARI, NMI, and PUR with low variance. For example, on the subsets \{3, 6, 9\} and \{1, 2, 3, 4\}, the NLGDL makes a remarkable improvement than the Euclidean distance and also other distance learning methods. ITDL competes with LGDL on some subsets but the latter performs more stable. DLC performs worse than LGDL and NLGDL. SDL and LSDL obtain better results than MDL and compete with each other on these subsets. MDL yields worse results than the Euclidean distance on some subsets because the high dimensionality of the data makes its optimization process become inefficient.

### F. Further Discussions

1) **Effectiveness of Constraint Weights:** The constraint weights measure the importance of a constraint according to the location in their neighborhood graph. Assigning different weights to each constraint can make full usage of the information that the constraint brings. Fig. 12 shows the experimental results of the proposed methods that consider the constraint weights and the methods without considering the constraint weights.

### TABLE III

| Datasets | IT | M | S | LS | D | LG | NLG |
|----------|----|---|---|----|---|----|-----|
| Iris     | 0.37| 4.46| 0.09| 0.08| 2.72| 1.47| 8.21 |
| Banknote | 0.22| 3.12| 0.09| 0.02| 3.71| 1.16| 0.02 |
| Wine     | 0.35| 68.36| 0.11| 0.05| 2.98| 2.94| 16.32 |
| Segment  | 0.47| 80.98| 0.1| 0.32| 4.32| 2.89| 18.13 |
| Waveform | 0.49| 114.13| 0.09| 0.15| 4.17| 4.76| 21.19 |
| Breast   | 0.88| 122.45| 0.1| 0.16| 6.88| 4.44| 20.32 |
| \{1,7\}  | 3.7 | 2421.5| 2.31| 3.49| 12.77| 7.12| 28.31 |
| \{2,4,7\}| 3.55| 2537.1| 3.37| 3.52| 14.79| 7.33| 29.13 |
| \{4,6,8,10\}| 3.9 | 2549.4| 3.51| 3.79| 17.89| 7.44| 29.98 |

IT: ITDL; M: MDL; S: SDL; LS: LSDL; D: DLC; LG: LGDL; NLG: NLGDL.
ITDL, especially the NLGDL obtains a remarkable accuracy improvement. DLC is a little slower than LGDL over most of the tested datasets. The most time-consuming part of the proposed methods and also in DLC is the construction of the neighborhood graph for calculating the importance degree of constraints. As illustrated in the above section, the constraint weights indeed improve the performance of FCM. That is to say, we sacrifice the running time for obtaining more accurate learning ability. But, more importantly, the NLGDL performs well on both linearly and nonlinearly separable data, which is more suitable for practical application. Further running time improvements can be achieved by accelerating the construction process of the neighborhood graph and optimizing the learning process.

3) Computational Complexity: We discuss the computational complexity of our proposed methods. Both linear and nonlinear algorithms are initialized by finding the nearest neighbors to construct Groups. The time complexity of this process is \( O(kNd) \), where \( k \) is the number of nearest neighbors, \( N \) is the size of the dataset, and \( d \) is the feature dimensionality. The time complexity for generating the \( g \)-neighborhood graph is \( O(qNd) \). These two processes only need to be run one time in the distance learning procedure. Next, we analyze the time complexity of LGDL in each iteration. First, we need to determine the Hausdorff distance between two Groups, which runs in \( O(k^2d) \) time. Then, we compute (11) with time complexity of \( O(cqd^2) \), where \( c \) is the number of constraints. At the end of each iteration, we ensure the semidefinite characteristic of a matrix with the time complexity of \( O(d^3) \).

Summarizing, the time complexity of LGDL at each iteration is \( O(k^2d + cd^2 + d^3) \). In NLGDL, the time complexity of the employed neural network is \( O(dq + q(l-1)) \), where \( q \) is the number of neurons in each hidden layer and \( l \) is the number of hidden layers. Therefore, the overall time complexity of NLGDL is \( O(k^2d + dq + q(l-1)) \).

VII. CONCLUSION

In this article, we proposed a novel distance learning method for improving the performance of fuzzy clustering. We elevated the pairwise constraints information to Group level, which contains specific background knowledge and retains a local geometrical structure of the data. These features can highly improve the distance learning process. We designed both linear and nonlinear learning methods based on the Group-level information by considering the importance degrees of constraints. The linear method utilizes semidefinite programming to seek a global transformation matrix. In addition, the neural network was employed here to learn the nonlinear transformation, which provides explicit mappings for both linearly and nonlinearly separable data. The experimental results on synthetic, UCI, and USPS datasets demonstrated that the developed linear and nonlinear methods outperform other distance learning methods that learned from pairwise constraints for clustering. In the future, we are going to improve the computational efficiency of the proposed methods by seeking a new way to compute the importance degree of constraints and refine the optimization process.

REFERENCES

[1] W. Pedrycz, Knowledge-Based Clustering: From Data to Information Granules. New York, NY, USA: Wiley, 2005.
[2] X. B. Zhu, W. Pedrycz, and H. C. Xiu, “Granular representation of data: A design of families of \( \varepsilon \)-information granules,” IEEE Trans. Fuzzy Syst., vol. 26, no. 4, pp. 2107–2119, Aug. 2018.
[3] H. Y. E. Y., Cui, W. Pedrycz, and Z. W. Li, “Enhancements of rule-based models through refinements of fuzzy \( c \)-means,” Knowl. Based Syst., vol. 170, pp. 43–60, Apr. 2019.
[4] C. Wang, W. Pedrycz, J. Yang, M. Zhou, and Z. Li, “Wavelet frame-based fuzzy \( c \)-means clustering for segmenting images on graphs,” IEEE Trans. Cybern., vol. 50, no. 9, pp. 3938–3949, Sep. 2020.
[5] J. Yang and C. Wang, “A developed fuzzy \( c \)-means algorithm for mesh segmentation,” in Proc. 11th Int. Congr. Image Signal Process. Biom. Eng. Informat., Beijing, China, 2018, pp. 1–5.
[6] J. Hu et al., “A multilevel prediction model of carbon efficiency based on the differential evolution algorithm for the iron ore sintering process,” IEEE Trans. Ind. Electron., vol. 65, no. 11, pp. 8778–8787, Nov. 2018.
[7] J. Hu, M. Wu, X. Chen, S. Du, W. Cao, and J. She, “Hybrid modeling and online optimization strategy for improving carbon efficiency in iron ore sintering process,” Inf. Sci., vol. 483, pp. 232–246, May 2019.
[8] V. Grossi, A. Romei, and F. Turini, “Survey on using constraints in data mining,” Data Min. Knowl. Discov., vol. 31, no. 2, pp. 424–464, 2016.
[9] D. Dinler and M. K. Tural, “A survey of constrained clustering,” in Unsupervised Learning Algorithms, M. Celebi and K. Aydin, Eds. Cham, Switzerland: Springer, pp. 207–235, 2016.
[10] W. Pedrycz, V. Loia, and S. Senatore, “P-FCM: A proximity-Based fuzzy clustering,” Fuzzy Sets Syst., vol. 148, no. 1, pp. 21–41, 2004.
[11] W. Pedrycz, “Proximity-based clustering: A search for structural consistency in data with semantic blocks of features,” IEEE Trans. Fuzzy Syst., vol. 21, no. 5, pp. 978–982, Oct. 2013.
[12] H. F. Liu, Z. Q. Tao, and Y. Fu, “Partition level constrained clustering,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 10, pp. 2469–2483, Oct. 2018.
[13] J. P. Mei, “Semisupervised fuzzy clustering with partition information of subsets,” IEEE Trans. Fuzzy Syst., vol. 27, no. 9, pp. 1726–1737, Sep. 2019.
[14] N. Grira, M. Crucianu, and N. Boujemaa, “Active semi-supervised fuzzy clustering,” Pattern Recognit., vol. 41, no. 5, pp. 1834–1844, 2008.
[15] C.-F. Gao and X.-J. Wu, “A new semi-supervised clustering algorithm with pairwise constraints by competitive agglomeration,” Appl. Soft Comput., vol. 11, no. 8, pp. 5281–5291, 2011.
[16] B. Kulis, “Metric learning: A survey,” Found. Trends Mach. Learn., vol. 5, no. 4, pp. 287–364, 2013.
[17] P. Mouatasis, M. J. Leng, and I. A. Kakadiaris, “An overview and empirical comparison of distance metric learning methods,” IEEE Trans. Cybern., vol. 47, no. 3, pp. 612–625, Mar. 2017.
[18] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” J. Mach. Learn. Res., vol. 10, pp. 207–244, Jun. 2009.
[19] B. Nguyen, C. Morell, and B. De Baets, “Scalable large-margin distance metric learning using stochastic gradient descent,” IEEE Trans. Cybern., vol. 50, no. 3, pp. 1072–1083, Mar. 2020.
[20] B. Nguyen, C. Morell, and B. De Baets, “Supervised distance metric learning through maximization of the Jeffrey divergence,” Pattern Recognit., vol. 64, pp. 215–225, Apr. 2017.
[21] D. T. Lai, J. M. Garibaldi, and J. Reps, “Investigating distance metric learning in semi-supervised fuzzy \( c \)-means clustering,” in Proc. IEEE Int. Conf. Fuzzy Syst., Beijing, China, 2014, pp. 1817–1824.
[22] E. P. Xing, A. Ng, M. I. Jordan, and S. Russell, “Distance metric learning with application to clustering with side-information,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2003, pp. 505–512.
[23] A. A. Abin, M. A. Bashiri, and H. Beigy, “Learning a metric when consistency in data with semantic blocks of features,” J. Mach. Learn. Res., vol. 21, no. 5, pp. 978–982, Oct. 2013.
[24] B. Wang and J. M. Sun, “Survey on distance metric learning and dimensionality reduction in data mining,” Data Min. Knowl. Discov., vol. 29, no. 2, pp. 534–564, 2014.
[25] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon, “Linear and kernel learning using a linear transformation,” J. Mach. Learn. Res., vol. 13, no. 1, pp. 519–547, 2012.
[26] D. Klein, S. D. Kamvar, and C. Manning, “From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering,” in Proc. 19th Int. Conf. Mach. Learn., 2002, pp. 307–314.
Witold Pedrycz (Life Fellow, IEEE) received the M.Sc., Ph.D., and D.Sc. degrees in automatic control and computer science from the Silesian University of Technology, Gliwice, Poland, in 1977, 1980, and 1984, respectively.

He is a Professor and the Canada Research Chair of Computational Intelligence with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also with the Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland. He has authored 15 research monographs covering various aspects of computational intelligence, data mining, and software engineering. His current research interests include computational intelligence, fuzzy modeling, and granular computing, knowledge discovery and data mining, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and software engineering. He has published numerous papers in the above areas.

Prof. Pedrycz received a prestigious Norbert Wiener Award from the IEEE Systems, Man, and Cybernetics Society, in 2007, the IEEE Canada Computer Engineering Medal, the Cajastur Prize for soft computing from the European Center for Soft Computing, the Killam Prize, and the Fuzzy Pioneer Award from the IEEE Computational Intelligence Society. He is currently a member of a number of editorial boards of other international journals. He is a member of numerous program committees of the IEEE conferences in the area of fuzzy sets and neurocomputing. He is a Foreign Member of the Polish Academy of Sciences. He is extensively involved in editorial activities. He currently serves on the Advisory Board for the IEEE TRANSACTIONS ON FUZZY SYSTEMS. He is the Editor-in-Chief of Information Sciences, WIREs Data Mining and Knowledge Discovery (Wiley), and International Journal of Granular Computing (Springer). He was a Fellow of the Royal Society of Canada, Ottawa, ON, Canada, in 2012.

Ji Yang received the B.Sc. (Hons.) and M.Sc. degrees in computing science from the University of Alberta, Edmonton, AB, Canada, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree with the Vision and Learning Lab, Department of Electrical and Computer Engineering.

His research interests are mainly on computer vision, human behavior analysis, and machine learning.