Constrained Locally Weighted Clustering

Hao Cheng, Kien A. Hua and Khanh Vu
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL, 32816
{haocheng, kienhua, khanh}@eecs.ucf.edu

ABSTRACT

Data clustering is a difficult problem due to the complex and heterogeneous natures of multidimensional data. To improve clustering accuracy, we propose a scheme to capture the local correlation structures: associate each cluster with an independent weighting vector and embed it in the subspace spanned by an adaptive combination of the dimensions. Our clustering algorithm takes advantage of the known pairwise instance-level constraints. The data points in the constraint set are divided into groups through inference; and each group is assigned to the feasible cluster which minimizes the sum of squared distances between all the points in the group and the corresponding centroid. Our theoretical analysis shows that the probability of points being assigned to the correct clusters is much higher by the new algorithm, compared to the conventional methods. This is confirmed by our experimental results, indicating that our design indeed produces clusters which are closer to the ground truth than clusters created by the current state-of-the-art algorithms.

1. INTRODUCTION

A cluster is a set of data points which share similar characteristics to one another compared to those not belonging to the cluster [18]. While the definition is fairly intuitive, it is non trivial at all to partition a multi-dimensional dataset into meaningful clusters. Such a problem has attracted much research attention from various Computer Science disciplines because clustering has many interesting and important applications [19].

In general, data objects are represented as feature vectors in clustering algorithms. Although the feature space is usually complex, it is believed that the intrinsic dimensionality of the data is generally much smaller than the original one [27]. Furthermore, the data are often heterogeneous. That is, different subsets of the data may exhibit different correlations; and in each subset, the correlations may vary along different dimensions [25]. As a result, each feature dimension may not necessarily be uniformly important for different regions of the entire data space. These observations motivate a lot of interest in constructing a new 'meaningful' feature space over a given set of data. Many global dimension reduction techniques such as [13] work on the derivation of new axes in the reduced space, onto which the original data space is projected. Recent studies in manifold learning [37] embed the space onto low-dimensional manifolds in order to discover the intrinsic structure of the entire space, which have shown encouraging results. To directly tackle the heterogeneous issue, adaptive distance metrics have been proposed [14], which define the degree of similarity between data points with regard to their surrounding subspaces. Basically, the focus of the above research is to work out a new salient representation of the data in order to improve the clustering performance.

Although clustering is traditionally an unsupervised learning problem, a recent research trend is to utilize partial information to aid in the unsupervised clustering process. It has been pointed out that the pairwise instance-level constraints are accessible in many clustering practices [29], each of which indicates whether a pair of data points must reside in the same cluster or not. The constraint set is useful in two ways. One way is to learn an appropriate distance metric. The other way is to direct the algorithm to find a more suitable data partitioning by enforcing the constraints and penalizing any violations of them.

In this paper, we propose to improve the accuracy of the clustering process in two aspects:

1. We capture the local structures and associate each cluster with its own local weighting vector. For each cluster, a dimension along which the data values of the cluster exhibit strong correlations receives a large weight; while a small one is assigned to a dimension of large value variations.

2. We integrate the constrained learning into the local weighting scheme. The data points in the constraint set are arranged into disjoint groups, each assigned as a whole to a cluster according to our defined criteria.

Our experimental results as well as the theoretical analysis reveal advantages of the proposed technique.

The remainder of the paper is organized as follows. Section 2 provides a survey of the related works. The locally weighted cluster concept, and the constrained learning are discussed in Section 3 and 4, respectively. The experimental results are reported in Section 5. Finally we conclude the paper in Section 6.

PVLDB '08, August 23-28, 2008, Auckland, New Zealand
Copyright 2008 VLDB Endowment, ACM 978-1-60558-305-1/08/08

2. RELATED WORK

In this section, we will discuss the related research works in different areas, including clustering, dimension reduction, manifold learning, and constrained clustering.

There are different types of clustering algorithms, such as *partitional clustering* and *hierarchical clustering*. An example of partitional clustering is *K-Means* [17, 33], in which a cluster is represented by its centroid. K-Means takes the iterative approach to minimize the sum of distances between data points and their respective nearest centroid. In hierarchical clustering, an agglomerative tree structure on a given dataset is generally created in either a *bottom-up* or *top-down* fashion. In the bottom-up approach, each data point is initially treated as a cluster by itself; and these clusters are merged in subsequent steps according to some specific criteria, such as *Single-Link*, *Complete-Link* or *Ward's method* [21]. A limitation of these methods is that they are sensitive to outliers [35]. A representative of top-down clustering is *Bisection K-Means* [36], which starts with the entire dataset as one big cluster and iteratively picks a cluster and divides it into two parts using K-Means until the desired number of clusters has been reached. Since the clusters produced by this repeated bisection procedure tend to have relatively uniform sizes, this approach generally has a more robust performance compared to the bottom-up clustering algorithms [35]. Recently there are also some proposals on graph theoretic clustering techniques [24, 34]. Generally, they are very computationally intensive [37].

Dimension reduction techniques aim to reduce the dimensionality of the original data space. One well-known technique is *Principal Component Analysis* [13], which minimizes the information loss caused by the reduction. Since it optimizes the mapping based on the global correlations in the dataset, PCA is likely to distort the local correlation structures of individual clusters that might reside in different subspaces. To address this problem, the *Locality Preserving Projection* [37] encodes the local neighborhood information into a similarity matrix and derives a low-dimensional linear manifold embedding as the optimal approximation to this neighborhood structure. Nonetheless, this type of global transformation schemes lacks the flexibility to directly model different shapes of individual clusters. As each cluster generally is compactly embedded in a different subspace, *ProClus* and its generalization [3, 4] seek to directly determine the subspaces for individual clusters. One disadvantage of these methods is that it may not be easy to determine the optimal dimensionality of the reduced space or the subspaces [25]. To overcome these problems, all the feature dimensions are properly weighted in the *Locally Adaptive Clustering* technique [14]. Specifically, the local feature selection is adopted so that different weighted distance metrics are in effect around the neighborhoods of different clusters. LAC and our local weighting scheme share the same motivation and both formulate the clustering problem as an optimization problem. However, as detailed in Section 3, our proposal differs in defining the objective function and the constraints. Moreover, our method does not require any tuning to control the weighting scheme and thus the performance is more stable, while that of LAC is fairly sensitive to its own tunable factor [5].

In constrained clustering, instance-level constraints indicate whether the corresponding pairs of data points belong to the same cluster or not. The constraints are usually used in learning a suitable Mahalanobis distance metric [6, 32] so that the data points marked similar are kept close to each other and the points which are identified dissimilar are dispersed far apart. The constraints are also used to directly guide the cluster assignment process. For a given set of constraints, it is desirable that a clustering algorithm does not violate any of them when producing data partitions. *Constrained K-Means* [30] adopted this idea and strictly enforces all the constraints over the cluster assignments. However, it has been shown that constrained clustering is a hard problem [10] and it is not necessarily a good idea to derive the partitions strictly satisfying every constraint [28]. Instead of enforcing the constraints directly, recent techniques introduced penalties on constraint violations; for example, the proposal in [10] seeks to minimize the *constrained vector quantization error*. The unified method, *MPCK-Means* [8] performs metric learning in every clustering iteration and penalizes the violations of the constraints. This technique also uses seeding to infer the initial centroids from the given constraint set to further improve the clustering performance [7]. In [16], a systematic approach is developed to tune the weights of dimensions to achieve a better clustering quality, which is defined as a weighted combination of the proportion of constraints satisfied in the output and an objective cluster validity index. Other interesting related research include the study of the utility of the constraint set [11, 12], and the modification of the Complete-Link clustering algorithm by exploring the spatial implications from the instance-level constraints [22].

In this paper, we integrate the local distance metric learning with constrained learning: the locally weighting scheme can well discover clusters residing in different subspaces, and our chunklet assignment strategy aggressively utilizes the input constraints to guide the clustering process. The improvement of the clustering accuracy has been observed in our experimental study.

3. LOCALLY WEIGHTED CLUSTERING

Let $\Re^m$ be the m-dimensional data space containing a set of N data points $\vec{x}_i$, whose jth component is x_{ij}. In the K-Means clustering, a cluster is represented by its centroid $\vec{c}_k \in \Re^m$, and a given point is assigned to the closest centroid based on the Euclidean distance or some global Mahalanobis distance. As discussed before, global distance metrics are ineffective to capture the local structures.

Instead, our scheme allows different weighted distance metrics for different clusters. Specifically, besides the centroid $\vec{c}_k$, a cluster is now associated with an adaptive weighting vector $\vec{w}_k$, which is determined based on the points in this cluster. The weights $\vec{w}_k$ are used to re-scale the distance from a data point $\vec{x}$ to the centroid $\vec{c}_k$, i.e.,

$$\mathcal{L}_{2,\vec{w}_k}(\vec{x}, \vec{c}_k) = \sqrt{\sum_{j=1}^{m} w_{kj} |c_{kj} - x_j|^2}.$$

Each data point is placed in its nearest cluster according to the adaptive distance metric. Formally, the membership function ϕ_c, the mapping of a point $\vec{x}$ to one of the K clusters, is

$$\phi_c(\vec{x}) = \arg \min_{1 \le k \le K} \mathcal{L}_{2,\vec{w}_k}(\vec{x}, \vec{c}_k). \quad (1)$$

Accordingly, all the points which belong to the kth cluster

are denoted as,

$$C_k = \{\vec{x} \mid \phi_c(\vec{x}) = k\}.$$

To achieve optimal clustering, the set of centroids and the corresponding clusters' weights together must minimize the sum of squared weighted distances from all the data points to their respective centroid, which is

$$\sum_{i=1}^{N} \mathcal{L}^2_{2,\vec{w}_{\phi_c(\vec{x}_i)}}(\vec{x}_i, \vec{c}_{\phi_c(\vec{x}_i)}), \tag{2}$$

subject to $\forall k \ \prod_{j=1}^{m} w_{kj} = 1$.

Our formulation differs from Locally Adaptive Clustering (LAC) [14]. In LAC, the constraint is the sum of weights to be one, which can lead to a trivial solution: the dimension along which the data exhibit the smallest variation is weighted one and the other dimensions receive zero weights. Thus, a regulation term representing the negative entropy of weights is added to the objective function with a coefficient. Consequently, the clustering objective is a weighted sum of vector quantization error and the regulation term. However, the critical coefficient greatly affects the quality of clustering outputs in practice, and there does not exist a simple and principal way to determine its value in LAC. In our proposal, we use the constraint that the product of the weights of any cluster must be equal to 1. This design is not trapped with the above mentioned trivial solution, and the regulation term is avoided. We do not need any user-specified parameters to control the locally weighting scheme. Note that the Euclidean distance is a special weighted distance measurement with all the weights being 1 and therefore the constraint conditions are satisfied. Our constrained minimization problem can be solved using the *Lagrange Multipliers*. We state major conclusions below:

THEOREM 1. *For the problem defined in Eq. 2, the optimal cluster centroids are, for* $1 \le k \le K, 1 \le j \le m$,

$$c_{kj} = \frac{1}{|C_k|} \sum_{\vec{x} \in C_k} x_j, \tag{3}$$

and the optimal weights are,

$$w_{kj} = \frac{\lambda_k}{\sum_{\vec{x} \in C_k} |x_j - c_{kj}|^2}, \tag{4}$$

in which $\lambda_k = \left(\prod_{j=1}^{m} (\sum_{\vec{x} \in C_k} |x_j - c_{kj}|^2)\right)^{\frac{1}{m}}$.

PROOF. See Appendix A. □

It is highly desired that Eqs. 3 and 4 are the closed-form formulae so that the centroids and weights can be computed fairly efficiently during the clustering iterations. It is also interesting to see that in our scheme, the centroid of a cluster is still the center of all the points in the cluster irrespective of the different weights. As Eq. 4 shows, the local weighting coefficients of a cluster are non-negative and completely determined by all the points it encloses and are not directly affected by other clusters. Specifically, the component w_{kj} is inversely proportional to the variance of the values in the jth dimension of all data points in C_k. If the points in the kth cluster differ greatly in dimension j, the weight w_{kj} is smaller. On the other hand, if the points exhibit a strong correlation in the jth dimension, then a larger weight is assigned to this dimension. In general, the adaptive weights can characterize the shapes of the clusters and are expected to well reflect the heterogeneous natures of different clusters. Our formulation is intuitive and has a stable performance with no tuning.

It is possible that for some cluster k and some dimension j, the value $\sum_{\vec{x} \in C_k} |x_j - c_{kj}|^2$ can be very small and even zero, which can cause troubles in computing the weights of this cluster. To circumvent this problem, we set a threshold in practice and when the value $\sum_{\vec{x} \in C_k} |x_j - c_{kj}|^2$ falls below this threshold, we use the threshold instead in the subsequent computations of w_{kj} (in the experiments of this paper, the threshold is 10^{-6}). On the other hand, if the values $\sum_{\vec{x} \in C_k} |x_j - c_{kj}|^2$ are very large for some dimensions, it is likely that the direct computation of λ_k could result in an overflow. Eq. 4 to compute weights can be rewritten in logarithm to avoid this problem, as below:

$$\log w_{kj} = \frac{1}{m} \sum_{i=1}^{m} \log(\sum_{\vec{x} \in C_k} |x_i - c_{ki}|^2) - \log(\sum_{\vec{x} \in C_k} |x_j - c_{kj}|^2).$$

The adaptiveness of locally weighted clustering can be further extended by considering (the inverse of) the covariance matrix of each individual cluster in computing the Mahalanobis distance, which can describe any arbitrarily oriented ellipsoid centered at the centroid. However, as pointed out in [31], it is not robust when a small number of data points are used to compute the covariance matrix. During the clustering process, some intermediate clusters may only have several points and the estimated ill-conditioned covariance matrix can potentially compromise the clustering accuracy. Therefore, in this paper, we fit the shapes of the clusters to be ellipsoids aligned with the axes for stable performance.

Algorithm 1 Locally Weighted Clustering (LWC)

Require: a dataset of N points $\vec{x}_i \in \Re^m$, the number of clusters K.

Ensure: K cluster centroids $\vec{c}_k$ and weights $\vec{w}_k$.

1: Start with K initial centroids and set all the weights to be 1, i.e., $w_{kj} = 1$ for $1 \le k \le K, 1 \le j \le m$.
2: E-Step: Compute the membership decision $\phi_c(\vec{x}_i)$ for all the N data points according to Eq. 1 and derive K cluster sets C_k.
3: M-Step: For each cluster, recompute the centroid $\vec{c}_k$ with regard to all the points it has, according to Eq. 3 and then update the weights $\vec{w}_k$ according to Eq. 4.
4: Repeat steps 2 and 3 until converge.

Similar to K-Means, we propose an iterative procedure to reach a good partition for a given dataset, as shown in Algorithm 1. In the initial phase, we can use either *Forgy initialization* or *subset furthest first* for the centroid selection [17]. At the beginning, we assume that the shape of each cluster is a sphere and therefore all the weights are set to 1, indicating that the Euclidean distance is used. After the initialization, the whole procedure alternates between cluster assignments (E-step) and the updates of the centroids and the weights for individual clusters (M-step). In the E-step, each point is assigned to the closest cluster based on the local distance metric, and therefore the objective function defined in Eq. 2 for the new assignments surely becomes smaller. In the M-step, the centroids and the weights of the clusters are re-estimated using all the points which now belong to them, and this also certainly reduces the objective function, which

has been proved in Theorem 1. There are a finite number of partitions dividing N points into K sets, and the objective function keeps decreasing from iteration to iteration. Therefore Algorithm 1 guarantees to converge and the converged $\vec{c}_k$ and $\vec{w}_k$ give a local minimum of the objective function (the detailed proof is available in Appendix B). In practice, our algorithm LWC stops if either the data placements are stable or the user-specified maximum number of iterations is reached.

4. CLUSTERING UNDER CONSTRAINTS

Let ϕ_g denote the membership function of data points in the dataset according to the ground truth. Thus, $\phi_g(\vec{x})$ represents the true cluster label for $\vec{x}$. Define the binary relation $\mathbb{R}_g$ for any pair of data points to be either 1 if they both belong to the same cluster or 0 otherwise:

$$\mathbb{R}_g(\vec{x}_i, \vec{x}_j) = \begin{cases} 1, & \text{if } \phi_g(\vec{x}_i) = \phi_g(\vec{x}_j), \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

For a dataset of N points, there are $\frac{(N-1)*N}{2}$ unique pairs of relations in $\mathbb{R}_g$ between different points. As pointed out by Wagstaff et al [29, 30], a small part of the relation $\mathbb{R}_g$ is usually accessible in the clustering practice and they are naturally represented as instance-level constraints. That is, there are a certain number of pairs in the constraint set $\mathcal{C}$ and we know $\mathbb{R}_g(\vec{x}_i, \vec{x}_j)$ for all the pairs in $\mathcal{C}$. If $\mathbb{R}_g(\vec{x}_i, \vec{x}_j) = 1$, these two points must belong to the same cluster and this is called a *Must-Link* constraint. Otherwise, it is a *Cannot-Link* constraint. It is desired to have the clustering outputs satisfying these pairwise instance-level constraints. It has been shown that this partial information is fairly useful to improve the clustering accuracy and the semi-supervised clustering under constraints is a promising research direction. One example is the Constrained K-Means [30], in which each data point is individually placed in its 'closest feasible' cluster in the assignment phase. This motivates us to integrate our locally weighted clustering scheme with the constraints-driven clustering process.

4.1 Chunklet Assignment Basics

Aharon et al. [6] defined a *chunklet* as 'a subset of points that are known to belong to the same although unknown class'. Note that for a given set of pairwise constraints, it is possible to combine them to form chunklets based on the transitive closure of the must-link constraints. For instance, if $\mathbb{R}_g(\vec{x}_1, \vec{x}_2) = 1$ and $\mathbb{R}_g(\vec{x}_2, \vec{x}_3) = 1$, then $\mathbb{R}_g(\vec{x}_1, \vec{x}_3) = 1$ can be inferred and a chunklet can be formed by including these three points: $\Delta = \{\vec{x}_1, \vec{x}_2, \vec{x}_3\}$, whose size is the number of data points in the set, i.e., $s(\Delta) = 3$. The other type of the constraints, cannot-link, defines the relationships among different chunklets. Suppose, besides Δ, there is another chunklet $\Delta' = \{\vec{x}_4, \vec{x}_5\}$. Given that $\mathbb{R}_g(\vec{x}_3, \vec{x}_4) = 0$, then it can be inferred that chunklets Δ and Δ' should not be placed in the same cluster. Consequently, given a set of instance-level constraints, we can derive a set of chunklets and their relationships.

The conventional clustering procedures assign data points to clusters in one-by-one fashion. Given a chunklet, we can now consider assigning the points in the chunklet in bulk. Moreover, if we know two chunklets should not be in the same cluster, then their membership decisions are indeed related and we can also consider placing them at the same time. This is the basic idea of our *chunklet assignment* strategy, and how we decide the memberships of the chunklets are explored in detail:

For an isolated chunklet Δ, which does not have any cannot-link constraints with any other chunklets, all points in Δ are assigned to the cluster which minimizes the sum of squared distances between all the points in Δ and the centroid $\vec{c}_i$:

$$\sum_{\vec{x} \in \Delta} \mathcal{L}^2_{2,\vec{w}_i}(\vec{x}, \vec{c}_i). \quad (6)$$

When there are two neighboring chunklets Δ and Δ' and there are cannot-link constraints between them, then they have to belong to different clusters. We assign Δ to cluster i and Δ' to cluster $j, (i \neq j)$, in order to minimize the objective:

$$\sum_{\vec{x} \in \Delta} \mathcal{L}^2_{2,\vec{w}_i}(\vec{x}, \vec{c}_i) + \sum_{\vec{x} \in \Delta'} \mathcal{L}^2_{2,\vec{w}_j}(\vec{x}, \vec{c}_j). \quad (7)$$

In the following, we examine the theoretical background of the above strategies and in the next subsection, we discuss how the theory can be applied in practice.

Consider a simple scenario: there are two clusters C_1 and C_2 in the dataset. For cluster C_i, the data values in the jth dimension follow the normal distribution $N(\mu_{ij}, 1)$, $(1 \leq j \leq m)$, which has the mean value μ_{ij} and the unit variance for simplicity, and values of different dimensions are mutually independent. Ideally, the centroids in the ground truth are $\vec{c}_1 = (\mu_{11}, \ldots, \mu_{1m})$ and $\vec{c}_2 = (\mu_{21}, \ldots, \mu_{2m})$. As the variances are 1 in all the dimensions of both clusters, the Euclidean distance, denoted as $\mathcal{L}_{2,\vec{1}}$, is adopted in the following analysis.

Suppose there is a chunklet Δ, that belongs to cluster i, i.e, $\Delta \subseteq C_i$ $(1 \leq i \leq 2)$. According to Eq. 6, Δ is assigned to cluster j, if for $1 \leq j, p \leq 2$, $j \neq p$,

$$\textstyle\sum_{\vec{x} \in \Delta} \mathcal{L}^2_{2,\vec{1}}(\vec{x}, \vec{c}_j) < \sum_{\vec{x} \in \Delta} \mathcal{L}^2_{2,\vec{1}}(\vec{x}, \vec{c}_p).$$

The probability of this event is denoted as,

$$P_\Delta(j \mid i) = P(\Delta \text{ is assigned to } C_j \mid \Delta \subseteq C_i),$$

which can be computed as below.

THEOREM 2. *For clusters C_1, C_2 and chunklet Δ,*

$$P_\Delta(1 \mid 1) = P_\Delta(2 \mid 2) = P_a(s(\Delta)),$$
$$P_\Delta(2 \mid 1) = P_\Delta(1 \mid 2) = P_a(-s(\Delta)),$$

in which $s(\Delta)$ is the number of data points in the chunklet Δ and $P_a(x)$ is defined as,

$$P_a(x) = \Phi\left(\frac{x}{2\sqrt{|x|}} \sqrt{\sum_{j=1}^{m} (\mu_{1j} - \mu_{2j})^2}\right),$$

and $\Phi(x)$ is the cumulative distribution function of the standard normal distribution $N(0, 1)$, i.e.,

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-\frac{u^2}{2}) du.$$

PROOF. See Appendix C. □

The function $\Phi(x)$ is the cumulative distribution function, that is monotonically increasing with respect to x. Hence, $P_a(x)$ is also a monotonically increasing function. The probability to assign Δ to its true cluster is

$$\textstyle\sum_{i=1}^{2} P(\Delta \subseteq C_i) P_\Delta(i \mid i) = P_a(s(\Delta)).$$

Similarly, we have the mistake probability $P_a(-s(\Delta))$. The chance of correct assignments goes up rapidly with the increase of the size of the chunklet, while that of mistake assignments decreases. In other words, if there are more data points in a chunklet, it is more likely that Δ is assigned to its true cluster using Eq. 6. As there are multiple points in a chunklet and they are independent, the chance that all of them are far away from their true centroid is much smaller than the chance that any of them is far from the centroid. Note that $\sqrt{\sum_{j=1}^{m}(\mu_{1j}-\mu_{2j})^2}$ is exactly the distance of the true centroids, i.e., $\mathcal{L}_{2,\vec{1}}(\vec{c}_1,\vec{c}_2)$. The value $P_a(s(\Delta))$ becomes larger as $\vec{c}_1$ and $\vec{c}_2$ have a greater distance. Therefore, if the two centroids are far away from each other, it is generally easier to distinguish these two clusters and the probability of mistake assignments is much smaller. Theorem 2 reflects this intuition well.

To examine the theoretical advantage of our assignment strategy, we compare the *Average Number of Correct Assignments* (*ANCA*) of some well-known clustering techniques. Specifically, assume each method can find the true centroids in the ground truth and we would like to count on average, how many data points in the chunklet are assigned to their respective true cluster. The conventional K-Means [17] does not utilize any constraints: it determines the membership of each point individually. The probability to assign a point $\vec{x} \in C_i$ correctly is $P_{\{\vec{x}\}}(i \mid i) = P_a(1)$, because a single point itself is a chunklet sized 1. Since the assignments of data points are independent, the occurrence of correct assignments is a binomial process with $n = s(\Delta)$ and $p = P_a(1)$ [20]. Therefore, the ANCA of K-Means is

$$\begin{aligned} & \sum_{i=1}^{2} P(\Delta \subseteq C_i)\Big(\sum_{j=0}^{s(\Delta)} j\binom{s(\Delta)}{j}\Big(P_a(1)\Big)^j\Big(1-P_a(1)\Big)^{s(\Delta)-j}\Big) \\ = \;& s(\Delta)P_a(1). \end{aligned}$$

Another approach, Constrained K-Means [30], decides the cluster assignment for the first point in Δ and all the rest points in Δ are forced to follow this decision and assigned to the same cluster due to the must-link constraints. Therefore, the assignments of the whole chunklet are either completely right or wrong, which solely depend on the decision of the first point. The chance of the first decision being correct is $P_{\{\vec{x}\}}(i \mid i)$. Hence, its ANCA is

$$\begin{aligned} & \textstyle\sum_{i=1}^{2} P(\Delta \subseteq C_i)\Big(s(\Delta) * P_a(1) + 0 * (1 - P_a(1))\Big) \\ = \;& s(\Delta)P_a(1). \end{aligned}$$

Interestingly, in the described scenario, the above two schemes have the same number of correct assignments on average. Unlike these two methods, our chunklet assignment strategy makes a joint decision for all points in Δ at once with the chance of totally correct assignments being $P_a(s(\Delta))$. Consequently our ANCA is

$$\begin{aligned} & \textstyle\sum_{i=1}^{2} P(\Delta \subseteq C_i)\Big(s(\Delta) * P_a(s(\Delta)) + 0 * (1 - P_a(s(\Delta)))\Big) \\ = \;& s(\Delta)P_a(s(\Delta)). \end{aligned}$$

Because $P_a(s(\Delta))$ is far larger than $P_a(1)$, clearly our cluster assignment is superior.

Next, we consider the assignments of two chunklets Δ and Δ' with cannot-link constraints in between, which should not be placed in the same cluster. The ANCA of K-Means is $(s(\Delta) + s(\Delta'))P_a(1)$. For Constrained K-Means, the correctness of the assignments is determined by the first decision of the points in the chunklets and the ANCA is also $(s(\Delta) + s(\Delta'))P_a(1)$. Instead, we use Eq. 7 to decide their memberships. The two chunklets $\Delta \subseteq C_i$ and $\Delta \subseteq C_j$ are placed in two different clusters, C_p and C_q, in order to minimize the aggregated distances ($1 \le i, j, p, q \le 2, i \ne j, p \ne q$). This occurs with a probability,

$$P_{\Delta,\Delta'}(p, q \mid i, j) =$$
$$P(\Delta, \Delta' \text{ are respectively assigned to } C_p, C_q \mid \Delta \subseteq C_i, \Delta' \subseteq C_j),$$

which can be computed according to the below theorem.

THEOREM 3. *For clusters C_1, C_2 and chunklets Δ, Δ',*

$$P_{\Delta,\Delta'}(1,2 \mid 1,2) = P_{\Delta,\Delta'}(2,1 \mid 2,1) = P_a(s(\Delta) + s(\Delta')),$$
$$P_{\Delta,\Delta'}(2,1 \mid 1,2) = P_{\Delta,\Delta'}(1,2 \mid 2,1) = P_a(-s(\Delta) - s(\Delta')).$$

PROOF. See Appendix D. □

Accordingly, the ANCA of our rule in Eq. 7 is the biggest, which is $(s(\Delta) + s(\Delta'))P_a(s(\Delta) + s(\Delta'))$. Intuitively, when we consider the memberships of Δ and Δ' together, the cannot-link constraints actually reduce the search space of all possible assignments and it is much more likely that a joint decision for the two chunklets is correct. In summary, Theorems 2 and 3 indicate that it is better to group points into chunklets and do chunklet assignments with Eqs. 6 and 7. When we consider the memberships of more points collectively (either one chunklet or two neighboring chunklets), it is more likely that we assign them to their true clusters.

4.2 Constrained Clustering

For a given set of pairwise constraints, our Constrained Locally Weighted Clustering (CLWC) first builds the chunklets and then the chunklet graph. Initially, each point in the constraint set is a chunklet of size 1. For every must-link constraint, we merge the chunklets containing the two points of the constraint. This procedure continues until all must-link constraints have been processed. Next, we construct the chunklet graph by representing each chunklet as a vertex. For each cannot-link constraint, an edge is added between the two vertices whose chunklets enclose any one of the points in the constraint. Eventually, an edge in the resulting graph indicates that the chunklets of the vertices connected by the edge (neighbor chunklets in the graph) should belong to different clusters. The generated graph, denoted as G_c, is used to guide the cluster assignment step and this has implicit impacts on the updates of the new centroids and the weights during iterations.

In each E-step, the memberships of all data points are re-examined. For the points not participating in any constraints, they are assigned to their closest clusters as usual. The main difference is that the chunklet assignment strategy is applied for the points of all the chunklets in G_c. At the start of the E-step, all chunklets are unassigned (to any cluster). CLWC picks either one or two chunklets at a time and decides their memberships until all the chunklets are assigned. As there are usually a number of chunklets in G_c, two questions need to be answered: which chunklets should be first chosen from G_c for consideration of the memberships and which clusters they should be assigned to.

According to Theorems 2 and 3, the probability of correct assignments of the two neighboring nodes Δ and Δ' is proportional to the number of data points in they two, i.e., $s(\Delta) + s(\Delta')$. This suggests that we should pick the biggest chunklets first. To make decisions for chunklet Δ, it is best to combine its assignment with that of its largest unassigned

neighbor Δ' if available. Only if Δ does not have any neighbors or all its neighbors have already been assigned, is the membership of this chunklet considered singly. Specifically, let $N_u(\Delta)$ denote the set of the immediate neighbor chunklets of Δ in G_c which have not yet been assigned. Define the score for each unassigned chunklet,

$$score(\Delta) = \begin{cases} s(\Delta) & \text{if } N_u(\Delta) = \emptyset, \\ s(\Delta) + \max(\{s(\Delta') \mid \Delta' \in N_u(\Delta)\}) & \text{otherwise.} \end{cases}$$

The max function is used in the score computation so that a chunklet and its largest unassigned neighbor (if available) can be decided jointly, corresponding to a smallest probability of mis-assignments. The score of Δ is the maximum number of data points that can be considered for the memberships along with Δ. Hence, if chunklet Δ of the biggest score (draws are broken randomly) has undetermined neighbors, it and its largest unassigned neighbor are selected. Otherwise only Δ is chosen for the determination of its membership at this time. As chunklets are assigned to clusters in the descending order of their sizes, the assignment decisions are generally correct and more reliable.

Next, we consider the question of how to make the assignment decision. When a single chunklet Δ is in consideration, some of its neighbors may already be assigned to some clusters and therefore these clusters are blocked from accepting Δ due to the possible violations of the cannot-link constraints between Δ and its neighbors. This effectively limits the search space for the assignment of Δ. Among all the remaining feasible clusters, we pick the one which has the minimum sum of squared distance between the centroid and all the points in the chunklet. If such a cluster cannot be found, a conflict is encountered: no matter which cluster the chunklet is assigned to, some constraints are surely going to be broken. As to find cluster assignments to enforce all the constraints (specifically the cannot-links) is an NP-Hard problem [10], CLWC deals with this situation by tolerating some violations and assigning Δ to its closest cluster without considering the cannot-link constraints between itself and its assigned neighbor chunklets. As observed in our experimental study, violations are indeed a rare exception.

A similar process is designed to make a joint decision for chunklet Δ and its neighbor Δ'. First we find cluster candidates for Δ and Δ' respectively. Among all the feasible choices (without putting both of them in the same cluster and violating the constraints with their already assigned neighbor chunklets), we select the one that minimizes the objective in Eq. 7. If we fail to find a feasible assignment, this indicates that any assignments of the two chunklets will cause conflicts with some of their assigned neighbor chunklets. In this case, we ignore the decisions of all the assigned chunklets, and put Δ and Δ' in the clusters which minimize the objective defined in Eq. 7. Again, constraint violations are surely incurred, however, they rarely happen in practice.

The time complexity of our chunklet assignment algorithm is competitive to that of the K-Means. The cost of each iteration of K-Means is $O(|X|Km)$ [15] in which $|X|$ is the size of the dataset, K is the number of clusters and m is the dimensionality. In an efficient implementation of CLWC, at the start of each iteration, the distances between each chunklet and each cluster are computed first, which are used to decide the membership of each chunklet in the subsequent process of the iteration. The worst case time complexity of the assignment procedure is still $O(|X|Km)$. In addition, our algorithm takes fewer iterations to converge compared with K-Means, as observed in the experiments.

5. EXPERIMENTAL RESULTS

5.1 Methods and Datasets

We evaluated the clustering performance of our proposals, LWC and CLWC and compared them with other state-of-the-art techniques. All the methods are listed below.

1. K-Means [17]: K-Means using the default Euclidean distance metric.
2. Bisection K-Means [36]: repeatedly partition the dataset into two parts using K-Means.
3. PCAC [13]: K-Means over the reduced space generated by Principal Component Analysis (PCA).
4. LPC [37]: K-Means over the reduced space generated by Locality Preserving Projection (LPP).
5. LAC [14]: Locally Adaptive Clustering.
6. LWC: The proposed Locally Weighted Clustering.
7. COP-KMeans [30]: Constrained K-Means.
8. MPCK-Means [8]: involves both metric learning and constraints satisfaction.
9. CLWC: The proposed Constrained Locally Weighted Clustering.

We implemented those methods except for LPP and MPCK-Means, which we obtained from the authors' web sites [1, 2]. Techniques 1 through 6 are unsupervised learning ones, while the last 3 utilize instance-level constraints to guide the cluster assignment process as well as learning the distance metric. Since the optimal number of clusters K for each dataset is already known, we used them in our experiments. In the case that additional tuning parameters were needed, we used the default parameters and followed the authors' recommendations. When they were not available, we manually tuned and reported only the best performance. Extensive experiments were carried out over the datasets in Table 1. Most datasets were downloaded from the UCI repository [23], among which the Digits and Letter datasets were sampled by respectively extracting characters 3, 8, 9 and A, B, as in [5, 8]. The Protein dataset was used in [32].

dataset	N	m	K
Soybean Small	47	35	4
Protein	116	20	6
Iris Plant	150	4	3
Wine Recognition	178	13	3
Heart Stat Log	270	13	2
Ionosphere	351	34	2
Balance Scale	625	4	3
Wisconsin Breast Cancer	683	9	2
Digits (3,8,9)	1008	16	3
Letter (A,B)	1555	16	2

Table 1: Datasets used in the experiments

5.2 Evaluation Metrics

We used two common metrics to evaluate the qualities of clustering outputs of different methods. The first metric is the *Rand Index* [26]. Given the membership $\phi_c(\vec{x})$ for each point $\vec{x}$ by a clustering algorithm, a pairwise relation $\mathbb{R}_c$ is defined for each pair of points, similar to Eq. 5. Then the Rand index is the percentage of pairs in the relations $\mathbb{R}_g$ and $\mathbb{R}_c$, which agree with each other, i.e,

$$Rand(\phi_g, \phi_c) = \frac{\sum_{i=1}^{N}\sum_{j=(i+1)}^{N} \mathbf{1}(\mathbb{R}_g(\vec{x}_i, \vec{x}_j) - \mathbb{R}_c(\vec{x}_i, \vec{x}_j))}{\frac{(N-1)N}{2}},$$

in which $\mathbf{1}(x)$ is the indicator function, equal to 1 if $x = 0$, and 0 otherwise. The second metric is the *Normalized Mutual Information* [5, 37], which measures the consistency of the clustering output compared to the ground truth. It reaches the maximum value of 1 only if ϕ_c perfectly matches ϕ_g and the minimal zero if the assignments of ϕ_c and ϕ_g are independent. Formally,

$$NMI(\phi_g, \phi_c) = \frac{\sum_{i=1}^{K}\sum_{j=1}^{K} p_{g,c}(i,j) \log \frac{p_{g,c}(i,j)}{p_g(i)p_c(j)}}{\min(\sum_{i=1}^{K} p_g(i) \log \frac{1}{p_g(i)}, \sum_{j=1}^{K} p_c(j) \log \frac{1}{p_c(j)})},$$

where $p_g(i)$ is the percentage of points in Cluster i according to the ground truth, i.e. $p_g(i) = \frac{\sum_{k=1}^{N} \mathbf{1}(\phi_g(\vec{x}_k)-i)}{N}$. Similarly, $p_c(j) = \frac{\sum_{k=1}^{N} \mathbf{1}(\phi_c(\vec{x}_k)-j)}{N}$ and $p_{g,c}(i,j)$ is the percentage of points that belong to Cluster i in ϕ_g and also Cluster j in ϕ_c, i.e. $p_{g,c}(i,j) = \frac{\sum_{k=1}^{N} \mathbf{1}(\phi_g(\vec{x}_k)-i)\mathbf{1}(\phi_c(\vec{x}_k)-j)}{N}$.

The above defined metrics were used to evaluate the accuracy of the clustering algorithms in addition to the number of violated constraints for the semi-supervised ones. We will report the number of iterations our proposals take to converge compared to the efficient techniques.

5.3 Unsupervised Clustering Accuracy

Each of the six unsupervised clustering methods was run 100 times with different initializations over all the datasets. For LPC and PCAC, we tested them with all the possible reduced dimensionalities and recorded their best performances. Similarly we tried different h's for LAC. The averaged Rand index and NMI are summarized in Table 2. The methods that performed the best on different datasets with regard to a particular metric are highlighted (boldface).

In general, the two evaluation metrics are quite consistent. Although no single method can outperform all the others for all the datasets, the proposed LWC is effective in many cases. According to the Rand index (or NMI), the LWC has the best performances in 4 (5 for NMI) datasets. For the other datasets, it is within 3.9% (respectively 8.8%) compared to the best method except in the sampled hand digits dataset. In addition to good overall performance, LWC does not require any parameter tuning. Thus, our method is an advanced unsupervised method for the real-world clustering problems.

5.4 Semi-Supervised Clustering Accuracy

To generate constraints, we adopted the methodology in [29, 30]: for each constraint, two data points were randomly picked from the dataset and if both were in the same cluster in the ground truth, a must-link constraint between them was generated. Otherwise it was a cannot-link constraint. In each dataset, totally 1000 sets of constraints of different sizes were created (every 50 sets were of the same size), typically ranging from 50 to 1000 constraints (25 to 500 for the Soybean dataset). The semi-supervised methods, COP-KMeans, MPCK-Means and the proposed CLWC were tested over all constraint sets, whose average performances are reported in Table 3 and Figures 1(a) to 1(f). Since COP-KMeans strictly enforces all the constraints, for many datasets, it failed to produce any feasible clustering partitions (with different initializations) when given more than 100 constraints. We therefore only report its performances in experiments with a small number of constraints.

As shown, CLWC generally produces much better clusters compared to the other two methods: the accuracy curves of CLWC are almost always higher than those of MPCK-Means for the datasets. As the number of the constraints becomes larger, indicating more partial information is used to guide the clustering process, the accuracy of both CLWC and MPCK-Means improves consistently. Note that the performance curves of MPCK-Means may drop when given a small number of constraints, and the performances under constraints may be even a little worse than those without constraints for several datasets, for example, the performance degradation in the wine dataset under around 300 constraints. This is consistent with observations in [8], which is due to the fact that its metric learning may become biased when there is not enough information to train the metric parameters. It is interesting to observe that CLWC does not suffer this problem, having a much smoother performance with additional constraints; there are rarely noticeable 'dips' in the performance of CLWC.

Although our constrained clustering algorithm does not guarantee the satisfaction of all constraints, only a small number of constraints were observed broken by our method in the experiments. The average numbers of violated constraints for the datasets are shown in Figures 2(a) - 2(c): they grow slowly as the number of pairwise constraints increases and are much smaller compared to those of MPCK-Means.

5.5 Clustering Efficiency

We summarize the average number of iterations the clustering algorithms took to reach convergence in Table 4. The 4th and 7th columns are the numbers of the constraints in use. Compared with K-Means, which is an efficient algorithm [18], the LWC algorithm converges fairly quickly and it took a comparable number of iterations to generate the clusters. The CLWC algorithm generally took even fewer iterations to converge than K-Means and MPCK-Means, and the more constraints were given, the faster CLWC completed the data partition. Therefore, our proposals are also quite efficient.

6. CONCLUSIONS

In this paper, we proposed to use local weighting vectors in order to capture the heterogeneous structures of data clusters in the feature space. Each set of weights defines the subspace spanned by the corresponding cluster. We integrated the constrained learning into our locally weighted clustering algorithm. A set of chunklets are built upon constraints, whose points are assigned to clusters collectively. Theoretical analysis and experiments have confirmed the superiority of our new proposals.

Currently, we are investigating the proposed technique for

dataset	K-Means		Bisection		PCAC		LPC		LAC		LWC	
	Rand	NMI	Rand	NMI	Rand	NMI	Rand	NMI	Rand	NMI	Rand	NMI
Soybean	0.847	0.788	0.852	0.767	0.851	0.798	0.856	**0.833**	0.853	0.809	**0.857**	0.810
Protein	0.774	0.392	0.783	0.405	**0.787**	0.400	0.765	0.335	0.778	**0.408**	0.762	0.372
Iris	0.853	0.648	0.858	0.695	0.898	0.800	**0.924**	**0.832**	0.861	0.743	0.899	0.823
Wine	0.709	0.436	0.727	0.396	0.710	0.436	0.710	0.440	0.861	0.696	**0.884**	**0.741**
Heart	0.516	0.019	0.516	0.019	0.516	0.019	0.524	0.031	0.504	0.048	**0.617**	**0.181**
Ionosphere	0.589	0.137	0.589	0.137	**0.590**	0.137	0.574	0.107	0.589	**0.138**	0.566	0.126
Balance	0.582	0.114	0.576	0.109	0.586	0.121	0.586	0.124	**0.589**	0.128	**0.589**	**0.129**
Breast	0.925	0.752	0.925	0.755	0.926	0.755	0.924	0.753	0.891	0.686	**0.927**	**0.757**
Digits(3,8,9)	0.842	0.777	**0.908**	**0.801**	0.861	0.786	0.851	0.800	0.657	0.372	0.789	0.701
Letter(A,B)	0.777	0.474	0.775	0.473	0.777	0.474	**0.890**	0.731	0.816	0.556	0.889	**0.734**

Table 2: Accuracy of unsupervised clustering algorithms.

dataset	constraint	COP-KMeans		MPCK-Means		CLWC		constraint	MPCK-Means		CLWC	
	count	Rand	NMI	Rand	NMI	Rand	NMI	count	Rand	NMI	Rand	NMI
Soybean	25	0.848	0.734	**0.936**	**0.881**	0.873	0.790	75	**0.935**	**0.871**	**0.935**	0.862
Protein	50	0.778	0.382	**0.795**	**0.431**	0.773	0.390	200	**0.828**	**0.509**	0.824	0.501
Iris	50	0.861	0.723	0.912	0.804	**0.937**	**0.856**	100	0.943	0.854	**0.977**	**0.930**
Wine	50	0.714	0.388	0.919	0.793	**0.924**	**0.821**	100	0.889	0.707	**0.958**	**0.888**
Heart	100	0.525	0.020	0.586	0.126	**0.802**	**0.500**	300	0.799	0.495	**0.967**	**0.881**
Ionosphere	100	0.556	0.085	0.592	0.148	**0.594**	**0.216**	300	0.691	0.294	**0.937**	**0.791**
Balance	100	**0.604**	**0.160**	0.593	0.134	0.598	0.147	300	0.687	0.311	**0.699**	**0.339**
Breast	100	0.904	0.702	0.908	0.713	**0.934**	**0.781**	300	0.893	0.673	**0.967**	**0.874**
Digits(3,8,9)	150	**0.877**	**0.730**	0.773	0.651	0.790	0.658	500	0.833	0.671	**0.929**	**0.832**
Letter(A, B)	200	0.848	0.606	0.854	0.626	**0.900**	**0.740**	500	0.850	0.606	**0.931**	**0.802**

Table 3: Accuracy of semi-supervised clustering algorithms.

different application domains. In particular, we have implemented a content-based image retrieval system [9]. We are also studying other approaches such as nonnegative matrix factorization and random walk techniques for constrained locally weighted clustering.

7. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their comments on our manuscript, which help improve the quality of the paper.

8. REFERENCES

[1] http://www.cs.utexas.edu/users/ml/risc/code/.

[2] http://www.ews.uiuc.edu/~dengcai2/data/data.html.

[3] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. In *SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 61–72, 1999.

[4] C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. *SIGMOD Rec.*, 29(2):70–81, 2000.

[5] M. Al-Razgan and C. Domeniconi. Weighted clustering ensembles. In *SDM '06: Proceedings of the Fourth SIAM International Conference on Data Mining*, 2006.

[6] A. Bar-Hillel, N. S. Tomer Hertz, and D. Weinshall. Learning distance functions using equivalence relations. In *ICML '03: Proceedings of the Twentith International Conference on Machine Learning*, 2003.

[7] S. Basu, A. Banerjee, and R. J. Mooney. Semi-supervised clustering by seeding. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 27–34, 2002.

[8] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 11, 2004.

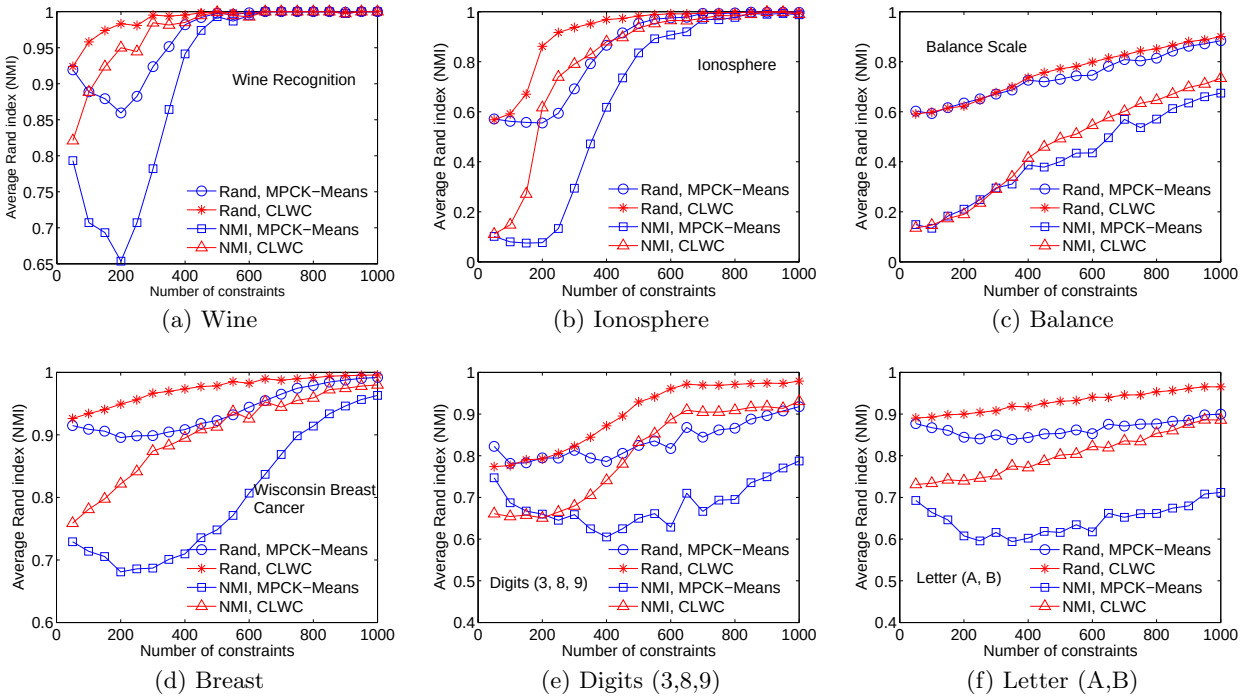
[9] H. Cheng, K. A. Hua, and K. Vu. Leveraging user query log: Toward improving image data clustering. In *CIVR '08: ACM International Conference on Image and Video Retrieval*, 2008.

[10] I. Davidson and S. S. Ravi. Clustering with constraints: Feasibility issues and the k-means algorithm. In *SDM '05: Proceedings of the Fourth SIAM International Conference on Data Mining*, 2005.

[11] I. Davidson and S. S. Ravi. Identifying and generating easy sets of constraints for clustering. In *AAAI '06: Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference*, 2006.

[12] I. Davidson, K. Wagstaff, and S. Basu. Measuring constraint-set utility for partitional clustering algorithms. In *Proceeding of ECML/PKDD*, 2006.

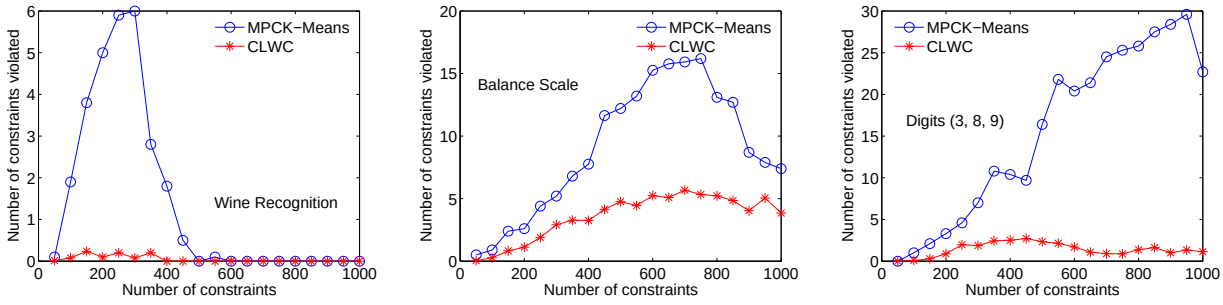
[13] C. Ding and X. He. K-means clustering via principal component analysis. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 29, New York, NY, USA, 2004. ACM Press.



(a) Wine (b) Ionosphere (c) Balance

(d) Breast (e) Digits (3,8,9) (f) Letter (A,B)

Figure 1: Accuracy of semi-supervised clustering algorithms.



(a) Wine (b) Balance (c) Digits (3,8,9)

Figure 2: Constraints violation of semi-supervised clustering algorithms.

[14] C. Domeniconi, D. Papadopoulos, D. Gunopulos, and S. Ma. Subspace clustering of high dimensional data. In *SDM '04: Proceedings of the Fourth SIAM International Conference on Data Mining*, 2004.

[15] C. Elkan. Using the triangle inequality to accelerate k-means. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003)*, pages 147–153, 2003.

[16] M. Halkidi, D. Gunopulos, M. Vazirgiannis, N. Kumar, and C. Domeniconi. A clustering framework based on subjective and objective validity criteria. *ACM Trans. Knowl. Discov. Data*, 1(4):1–25, 2008.

[17] G. Hamerly and C. Elkan. Alternatives to the k-means algorithm that find better clusterings. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 600–607, New York, NY, USA, 2002. ACM Press.

[18] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[19] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[20] E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

[21] S. D. Kamvar, D. Klein, and C. D. Manning. Interpreting and extending classical agglomerative clustering algorithms using a model-based approach. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 283–290, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[22] D. Klein, S. D. Kamvar, and C. D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *ICML '02: Proceedings of the Nineteenth*

dataset	K-Means	LWC	constraint count	MPCK-Means	CLWC	constraint count	MPCK-Means	CLWC
Soybean Small	3.60	2.80	25	5.00	2.89	75	3.60	2.73
Protein	6.21	3.95	50	6.40	4.78	200	5.15	4.34
Iris Plant	6.75	7.25	50	5.40	4.69	100	4.95	4.07
Wine	8.30	8.10	50	8.80	6.49	100	7.90	5.39
Heart Stat Log	9.20	10.30	100	8.60	9.33	300	5.00	2.71
Ionosphere	6.00	7.05	100	6.20	5.63	300	5.50	3.74
Balance Scale	12.60	6.25	100	13.00	8.72	300	12.20	12.39
Breast Cancer	4.20	5.00	100	5.70	4.51	300	4.70	3.15
Digits (3,8,9)	9.80	10.55	150	7.48	11.90	500	7.39	13.37
Letter (A,B)	10.95	10.20	200	8.79	7.84	500	6.50	5.45

Table 4: Average number of iterations.

International Conference on Machine Learning, pages 307–314, 2002.

[23] D. Newman, S. Hettich, C. Blake, and C. Merz. UCI repository of machine learning databases, 1998.

[24] A. Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS '02: Advances in Neural Information Processing Systems*, 2002.

[25] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explor. Newsl.*, 6(1):90–105, 2004.

[26] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:622–626, 1971.

[27] K. Vu, K. A. Hua, H. Cheng, and S.-D. Lang. A non-linear dimensionality-reduction technique for fast similarity search in large databases. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 527–538, New York, NY, USA, 2006. ACM.

[28] K. Wagstaff, S. Basu, and I. Davidson. When is constrained clustering beneficial, and why?. In *AAAI '06: Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference*, 2006.

[29] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1103–1110, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[30] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[31] T. Wang, Y. Rui, and S.-M. Hu. Optimal adaptive learning for image retrieval. In *CVPR '01: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1140–1147, 2001.

[32] E. Xing, A. Y. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS '03: Advances in Neural Information Processing Systems*, 2003.

[33] H. Xiong, J. Wu, and J. Chen. K-means clustering versus validation measures: a data distribution perspective. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 779–784, 2006.

[34] D. S. Yeung and X. Z. Wang. Improving performance of similarity-based clustering by feature weight learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):556–561, 2002.

[35] Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. Technical Report CS 01–40, Department of Computer Science, University of Minnesota, 2001.

[36] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524, New York, NY, USA, 2002. ACM Press.

[37] X. Zheng, D. Cai, X. He, W.-Y. Ma, and X. Lin. Locality preserving clustering for image database. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 885–891, New York, NY, USA, 2004. ACM Press.

APPENDIX

A. PROOF OF THEOREM 1

To solve the optimization problem, we use *Lagrange Multipliers*. Define:

$$\begin{aligned} F &= \sum_{i=1}^{N} \mathcal{L}_2^2(\vec{x}_i, \vec{c}_{\phi_c(\vec{x}_i)}) - \sum_{k=1}^{K} \lambda_k (\prod_{j=1}^{m} w_{kj} - 1) \\ &= \sum_{k=1}^{K} \sum_{\vec{x} \in C_k} \sum_{j=1}^{m} w_{kj} |x_j - c_{kj}|^2 - \sum_{k=1}^{K} \lambda_k (\prod_{j=1}^{m} w_{kj} - 1). \end{aligned}$$

For all $1 \le k \le K, 1 \le j \le m$, let

$$\frac{\partial F}{\partial c_{kj}} = \sum_{\vec{x} \in C_k} 2 w_{kj} (c_{kj} - x_j) = 0.$$

As $w_{kj} \neq 0$, then we get,

$$c_{kj} = \frac{1}{|C_k|} \sum_{\vec{x} \in C_k} x_j.$$

Similarly, let

$$\frac{\partial F}{\partial w_{kj}} = \sum_{\vec{x}\in C_k} |x_j - c_{kj}|^2 - \lambda_k \prod_{j'=1, j'\neq j}^{m} w_{kj'}$$
$$= \sum_{\vec{x}\in C_k} |x_j - c_{kj}|^2 - \frac{\lambda_k}{w_{kj}} = 0.$$

Then we have,

$$w_{kj} = \frac{\lambda_k}{\sum_{\vec{x}\in C_k} |x_j - c_{kj}|^2}.$$

As for $1 \leq k \leq K$, $\prod_{j=1}^{m} w_{kj} = 1$, we have

$$\lambda_k = \Big(\prod_{j=1}^{m} (\sum_{\vec{x}\in C_k} |x_j - c_{kj}|^2)\Big)^{\frac{1}{m}}.$$

The second order partial derivatives of F are computed as:

$$\begin{bmatrix} \frac{\partial^2 F}{\partial c_{kj}^2} & \frac{\partial^2 F}{\partial c_{kj}\partial w_{kj}} \\ \frac{\partial^2 F}{\partial w_{kj}\partial c_{kj}} & \frac{\partial^2 F}{\partial w_{kj}^2} \end{bmatrix} = \begin{bmatrix} 2\sum_{\vec{x}\in C_k} w_{kj} & \sum_{\vec{x}\in C_k} 2(c_{kj} - x_j) \\ \sum_{\vec{x}\in C_k} 2(c_{kj} - x_j) & \frac{\lambda_k}{w_{kj}^2} \end{bmatrix}.$$

Its determinant is positive at the derived optimal weights and centroids, and therefore, they represent a minimum.

B. PROOF OF CONVERGENCE OF LOCALLY WEIGHTED CLUSTERING

COROLLARY 1. *The Locally Weighted Clustering Algorithm (Algorithm 1) converges to a local minimum of the objective function defined in Eq. 2.*

PROOF. The objective function f is defined for the given assignments ϕ and centroids $\vec{c}$ and weights $\vec{w}$:

$$f(\phi, \vec{c}, \vec{w}) = \sum_{i=1}^{N} \mathcal{L}^2_{2,\vec{w}_{\phi(\vec{x}_i)}}(\vec{x}_i, \vec{c}_{\phi(\vec{x}_i)}).$$

Algorithm 1 starts from an initial assignment and runs from iteration to iteration. Each iteration consists of two steps: to determine cluster assignments (E-step, Line 2 in Algorithm 1) and to compute centroids and weights for individual clusters (M-step, Line 3).

Formally, let $\vec{c}^i$, $\vec{w}^i$ and ϕ_c^i respectively denote the centroids, weights, assignments derived in the ith iteration. $\vec{c}^0$ and $\vec{w}^0$ are the initial configuration, while in the algorithm ϕ_c^0 is not initialized and can be any assignment. In ϕ_c^i, each point $\vec{x}_i$ is assigned to its closest cluster according to weights and centroids in the last iteration, $\vec{c}^{i-1}$ and $\vec{w}^{i-1}$. Therefore, each E-step reduces the objective value, i.e.,

$$f(\phi_c^i, \vec{c}^{i-1}, \vec{w}^{i-1}) \leq f(\phi_c^{i-1}, \vec{c}^{i-1}, \vec{w}^{i-1}).$$

In each M-step, for the given ϕ_c^i, the optimal $\vec{c}^i$ and $\vec{w}^i$ are computed using Eqs. 3 and 4 (as in Appendix A). Hence, each M-step reduces the objective value, i.e.,

$$f(\phi_c^i, \vec{c}^i, \vec{w}^i) \leq f(\phi_c^i, \vec{c}^{i-1}, \vec{w}^{i-1}).$$

Overall, we have $f(\phi_c^i, \vec{c}^i, \vec{w}^i)$ no greater than $f(\phi_c^{i-1}, \vec{c}^{i-1}, \vec{w}^{i-1})$. It is guaranteed that Algorithm 1 reduces the objective value in iterations.

The clustering problem is to group N points into K disjoint sets and there are only a finite number of data partitions. For a given ϕ_c, the minimal objective value is determined for the corresponding optimal centroids and weights. Therefore, the objective value for a given assignment is lower-bounded. The objective value in Algorithm 1 decreases gradually until the value reaches a fixed point. This fixed point is a local minimal of $f(\phi, \vec{c}, \vec{w})$. □

C. PROOF ON ONE CHUNKLET

There are K clusters, $C_1, C_2, \ldots, C_K$. For cluster C_i, the data values in the jth dimension follow the normal distribution $N(\mu_{ij}, 1)$.

For a chunklet Δ that belongs to cluster s in the ground truth, $(\Delta \subseteq C_s)$, the conditional probability that the sum of distances from points in Δ to cluster i is smaller than that to cluster p, $(i \neq p)$, is denoted as,

$$P_{d,1}(i, p \mid s) = P\Big(\sum_{\vec{x}\in\Delta} \mathcal{L}^2_{2,\vec{1}}(\vec{x}, \vec{c}_i) < \sum_{\vec{x}\in\Delta} \mathcal{L}^2_{2,\vec{1}}(\vec{x}, \vec{c}_p) \mid \Delta \subseteq C_s\Big).$$

THEOREM 4. *For $1 \leq i, p, s \leq K$, $i \neq p$ we have*

$$P_{d,1}(i, p \mid s) = \Phi\Big(\frac{-s(\Delta)\sum_{r=1}^{m}(\mu_{pr} - \mu_{ir})\mu_{sr} + \frac{1}{2}s(\Delta)\sum_{r=1}^{m}(\mu_{pr}^2 - \mu_{ir}^2)}{\sqrt{s(\Delta)\sum_{r=1}^{m}(\mu_{pr} - \mu_{ir})^2}}\Big).$$

PROOF. We can rewrite the left hand side (LHS) as below,

$$LHS = P\Big(\sum_{\vec{x}\in\Delta}\sum_{r=1}^{m}((x_r - \mu_{ir})^2 - (x_r - \mu_{pr})^2) < 0 \mid \Delta \subseteq C_s\Big)$$
$$= P\Big(\sum_{\vec{x}\in\Delta}\sum_{r=1}^{m}(\mu_{pr} - \mu_{ir})x_r < \frac{1}{2}s(\Delta)\sum_{r=1}^{m}(\mu_{pr}^2 - \mu_{ir}^2) \mid \Delta \subseteq C_s\Big).$$

As x_r follows $N(\mu_{sr}, 1)$, denoted as $x_r \sim N(\mu_{sr}, 1)$, then

$$(\mu_{pr} - \mu_{ir})x_r \sim N((\mu_{pr} - \mu_{ir})\mu_{sr}, (\mu_{pr} - \mu_{ir})^2).$$

Define $Y = \sum_{\vec{x}\in\Delta}\sum_{r=1}^{m}(\mu_{pr} - \mu_{ir})x_r$, following a normal distribution,

$$N(s(\Delta)\sum_{r=1}^{m}(\mu_{pr} - \mu_{ir})\mu_{sr}, s(\Delta)\sum_{r=1}^{m}(\mu_{pr} - \mu_{ir})^2).$$

We can normalize Y into a random variable of the standard normal distribution, $Y_N \sim N(0, 1)$, i.e.,

$$Y_N = \frac{Y - s(\Delta)\sum_{r=1}^{m}(\mu_{pr} - \mu_{ir})\mu_{sr}}{\sqrt{s(\Delta)\sum_{r=1}^{m}(\mu_{pr} - \mu_{ir})^2}}.$$

Therefore, we have,

$$LHS = P\Big(Y_N < \frac{-s(\Delta)\sum_r(\mu_{pr} - \mu_{ir})\mu_{sr} + \frac{1}{2}s(\Delta)\sum_r(\mu_{pr}^2 - \mu_{ir}^2)}{\sqrt{s(\Delta)\sum_{r=1}^{m}(\mu_{pr} - \mu_{ir})^2}}\Big).$$

As $Y_N \sim N(0, 1)$, the above equation can be further rewritten using the cumulative distribution function Φ of $N(0, 1)$. □

According to the definition of $P_{d,1}(i, p \mid s)$, for $1 \leq i, p, s \leq K$, we have

$$P_{d,1}(i, p \mid s) = 1 - P_{d,1}(p, i \mid s).$$

As the probability distribution function of $N(0, 1)$ is symmetric with regard to the $x = 0$, there is a special property of its cumulative function $\Phi(x)$, that is,

$$\Phi(x) + \Phi(-x) = 1.$$

Therefore, we have,

$$P_{d,1}(i, p \mid s) = \Phi(A),$$
$$P_{d,1}(p, i \mid s) = \Phi(-A),$$

in which

$$A = \frac{-s(\Delta)\sum_{r=1}^{m}(\mu_{pr}-\mu_{ir})\mu_{sr} + \frac{1}{2}s(\Delta)\sum_{r=1}^{m}(\mu_{pr}^2-\mu_{ir}^2)}{\sqrt{s(\Delta)\sum_{r=1}^{m}(\mu_{pr}-\mu_{ir})^2}}.$$

DISCUSSIONS:

According to Theorem 4, the event that the sum of distances of the points in $\Delta \subseteq C_s$ to its true cluster C_s is smaller than that to some cluster C_p, occurs with the probability,

$$P_{d,1}(s,p \mid s) = \Phi\Big(\frac{\sqrt{s(\Delta)}}{2}\sqrt{\sum_{r=1}^{m}(\mu_{pr}-\mu_{sr})^2}\Big).$$

In chunklet assignment, in case of two clusters C_1 and C_2, the chance to place Δ correctly is $P_{d,1}(1,2 \mid 1)$ (or $P_{d,1}(2,1 \mid 2)$). This is the conclusion in Theorem 2. In case of more than two clusters, Δ is assigned to cluster i if cluster i is the one closest to the points in the chunklet, i.e., for all $1 \le p \le K$, and $p \neq i$,

$$\textstyle\sum_{\vec{x}\in\Delta}\mathcal{L}^2_{2,\vec{1}}(\vec{x},\vec{c}_i) < \sum_{\vec{x}\in\Delta}\mathcal{L}^2_{2,\vec{1}}(\vec{x},\vec{c}_p).$$

Each of these events, $\sum_{\vec{x}\in\Delta}\mathcal{L}^2_{2,\vec{1}}(\vec{x},\vec{c}_i) < \sum_{\vec{x}\in\Delta}\mathcal{L}^2_{2,\vec{1}}(\vec{x},\vec{c}_p)$, is not necessarily independent. Consider there are 3 clusters in 2-dimensional space, $\vec{c}_1 = \langle 1,0\rangle$, $\vec{c}_2 = \langle 2,0\rangle$, $\vec{c}_3 = \langle 3,0\rangle$, it is true that, for any point $\vec{x}$, if it is closer to c_1 than c_2, then $\vec{x}$ is also closer to c_1 than c_3. Thus, for this example,

$$\begin{aligned} & P\Big(\sum_{\vec{x}\in\Delta}\mathcal{L}^2_{2,\vec{1}}(\vec{x},\vec{c}_1) < \sum_{\vec{x}\in\Delta}\mathcal{L}^2_{2,\vec{1}}(\vec{x},\vec{c}_2)\bigcap \\ & \quad \sum_{\vec{x}\in\Delta}\mathcal{L}^2_{2,\vec{1}}(\vec{x},\vec{c}_1) < \sum_{\vec{x}\in\Delta}\mathcal{L}^2_{2,\vec{1}}(\vec{x},\vec{c}_3) \mid \Delta\subseteq C_1\Big) \\ = \;& P_{d,1}(1,2 \mid 1) \neq P_{d,1}(1,2 \mid 1)P_{d,1}(1,3 \mid 1). \end{aligned}$$

Although the probability to assign Δ correctly is not expressed in a closed form for more than two clusters, generally this probability is related to $P_{d,1}(s,p \mid s)$. The more data points the chunklet Δ has, the larger the positive value $\frac{\sqrt{s(\Delta)}}{2}\sqrt{\sum_{r=1}^{m}(\mu_{pr}-\mu_{sr})^2}$ is. Hence, the corresponding probability $P_{d,1}(s,p \mid s)$ is larger, and $P_{d,1}(p,s \mid s)$ is smaller. It is more likely that the points in Δ are close to the cluster they belong to, as a group. Consequently, the probability to decide the membership of Δ correctly becomes larger with the increase of the size of the chunklet, $s(\Delta)$.

D. PROOF ON TWO CHUNKLETS

For chunklets $\Delta \subseteq C_s$ and $\Delta' \subseteq C_t$, $(s \neq t, i \neq p \cap j \neq q)$, denote

$$\begin{aligned} & P_{d,2}(i,j,p,q \mid s,t) \\ & = P\Big(\textstyle\sum_{\vec{x}\in\Delta}\mathcal{L}^2_{2,\vec{1}}(\vec{x},\vec{c}_i) + \sum_{\vec{x}\in\Delta'}\mathcal{L}^2_{2,\vec{1}}(\vec{x},\vec{c}_j) < \\ & \quad \textstyle\sum_{\vec{x}\in\Delta}\mathcal{L}^2_{2,\vec{1}}(\vec{x},\vec{c}_p) + \sum_{\vec{x}\in\Delta'}\mathcal{L}^2_{2,\vec{1}}(\vec{x},\vec{c}_q) \mid \Delta\subseteq C_s, \Delta'\subseteq C_t\Big). \end{aligned}$$

THEOREM 5. *For $1 \le i,j,p,q,s,t \le K$, $s \neq t, i \neq p \cap j \neq q$, we have*

$$P_{d,2}(i,j,p,q \mid s,t) = \Phi\Big(\frac{A}{B}\Big),$$

in which

$$\begin{aligned} A &= -s(\Delta)\textstyle\sum_{r=1}^{m}(\mu_{pr}-\mu_{ir})\mu_{sr} - s(\Delta')\sum_{r=1}^{m}(\mu_{qr}-\mu_{jr})\mu_{tr} \\ &\quad + \frac{1}{2}s(\Delta)\textstyle\sum_{r=1}^{m}(\mu_{pr}^2-\mu_{ir}^2) + \frac{1}{2}s(\Delta')\sum_{r=1}^{m}(\mu_{qr}^2-\mu_{jr}^2), \\ B &= \sqrt{s(\Delta)\textstyle\sum_{r=1}^{m}(\mu_{pr}-\mu_{ir})^2 + s(\Delta')\sum_{r=1}^{m}(\mu_{qr}-\mu_{jr})^2}. \end{aligned}$$

PROOF. The left hand side (LHS) can be rewritten as,

$$\begin{aligned} & LHS = P\Big(\sum_{\vec{x}\in\Delta}\sum_{r=1}^{m}(\mu_{pr}-\mu_{ir})x_r + \sum_{\vec{x}\in\Delta'}\sum_{r=1}^{m}(\mu_{qr}-\mu_{jr})x_r < \\ & \frac{1}{2}\Big(\sum_{\vec{x}\in\Delta}\sum_{r=1}^{m}(\mu_{pr}^2-\mu_{ir}^2) + \sum_{\vec{x}\in\Delta'}\sum_{r=1}^{m}(\mu_{qr}^2-\mu_{jr}^2)\Big) \mid \Delta\subseteq C_s, \Delta'\subseteq C_t\Big). \end{aligned}$$

Define $Y = \sum_{\vec{x}\in\Delta}\sum_r(\mu_{pr}-\mu_{ir})x_r + \sum_{\vec{x}\in\Delta'}\sum_r(\mu_{qr}-\mu_{jr})x_r$, that follows a normal distribution with the mean

$$s(\Delta)\textstyle\sum_{r=1}^{m}(\mu_{pr}-\mu_{ir})\mu_{sr} + s(\Delta')\sum_{r=1}^{m}(\mu_{qr}-\mu_{jr})\mu_{tr},$$

and the variance

$$s(\Delta)\textstyle\sum_{r=1}^{m}(\mu_{pr}-\mu_{ir})^2 + s(\Delta')\sum_{r=1}^{m}(\mu_{qr}-\mu_{jr})^2.$$

Y can be normalized, $Y_N \sim N(0,1)$, and we can derive the result of this theorem in the similar process in Theorem 4. □

According to the definition, for $1 \le i,j,p,q,s,t \le K$, we also have,

$$P_{d,2}(i,j,p,q \mid s,t) = 1 - P_{d,2}(p,q,i,j \mid s,t).$$

DISCUSSIONS:

According to Theorem 5, we have,

$$\begin{aligned} & P_{d,2}(s,t,p,q \mid s,t) \\ = \;& \Phi\Big(\frac{1}{2}\sqrt{s(\Delta)\sum_{r=1}^{m}(\mu_{pr}-\mu_{sr})^2 + s(\Delta')\sum_{r=1}^{m}(\mu_{qr}-\mu_{tr})^2}\Big). \end{aligned}$$

Without prior knowledge, if the distances among cluster centroids are the same, (i.e., for i,j, $\mathcal{L}_{2,\vec{1}}(\vec{c}_i,\vec{c}_j)$ is some constant), then

$$P_{d,2}(s,t,p,q \mid s,t) = \Phi\Big(\frac{\sqrt{s(\Delta)+s(\Delta')}}{2}\sqrt{\sum_{r=1}^{m}(\vec{c}_{sr}-\vec{c}_{tr})^2}\Big).$$

For two clusters C_1 and C_2, the probability to determine the memberships of Δ and Δ' with no mistakes is related to $P_{d,2}(1,2,2,1 \mid 1,2)$ and $P_{d,2}(2,1,1,2 \mid 2,1)$. This is the conclusion in Theorem 3. Similar to the analysis of one chunklet in the previous section, in case of more than two clusters, the probability to assign Δ and Δ' correctly is not necessarily equal to

$$\prod_{p=1,q=1,p\neq s\cap q\neq t}^{K} P_{d,2}(s,t,p,q \mid s,t).$$

In general, if the sizes of the two chunklets $s(\Delta)+s(\Delta')$ are bigger, the value $(s(\Delta)+s(\Delta'))\sum_r(\mu_{sr}-\mu_{tr})^2$ is larger, so is $s(\Delta)\sum_r(\mu_{pr}-\mu_{sr})^2 + s(\Delta')\sum_r(\mu_{qr}-\mu_{tr})^2$. Therefore, the probability is larger that Δ and Δ' are closer to their true clusters rather than any other clusters, and $P_{d,2}(p,q,s,t \mid s,t)$ is smaller. Hence, the probability to decide the membership of the two chunklets correctly is generally larger with more data points in Δ and Δ'.