Computing and Informatics, Vol. 31, 2012, 613–638

A NOVEL SCHEME FOR ACCELERATING SUPPORT VECTOR CLUSTERING

Yuan Ping

Information Security Center
Beijing University of Posts and Telecommunications
West Tucheng Road No. 10, Haidian District, Beijing, 100876, China
&
Department of Computer Science and Technology, Xuchang University,
BaYi Road No. 88, Xuchang, 461000, China
e-mail: pyuan.lhn@gmail.com

Yajian Zhou, Yixian Yang

Information Security Center
Beijing University of Posts and Telecommunications
West Tucheng Road No.10, Haidian District, Beijing, 100876, China

e-mail: {yajian, yxyang}@bupt.edu.cn

Communicated by Ondrej Habala

Abstract. Limited by two time-consuming steps, solving the optimization problem and labeling the data points with cluster labels, the support vector clustering (SVC) based algorithms, perform ineffectively in processing large datasets. This paper presents a novel scheme aimed at solving these two problems and accelerating the SVC. Firstly, an innovative definition of noise data points is proposed which can be applied in the design of noise elimination to reduce the size of a data set as well as to improve its separability without destroying the profile. Secondly, in the cluster labeling, a double centroids (DBC) labeling method, representing each cell of a cluster by the centroids of shape and density, is presented. This method is implemented towards accelerating this procedure and addressing the problem of labeling the original data set with irregular or imbalanced distribution. Compared with the state-of-the-art algorithms, the experimental results show that the

proposed method significantly reduces the computational resources and improves the accuracy. Further analysis and experiments of semi-supervised cluster labeling confirm that the proposed DBC model is suitable for representing cells in clustering.

Keywords: Support vector clustering, noise elimination, centroid, semi-supervised clustering

Mathematics Subject Classification 2010: 62H30, 68T30, 94A17

1 INTRODUCTION

Inspired by the support vector machines (SVM) [1], the support vector clustering (SVC) [2, 3, 4] and its variants [5, 6, 8, 7] are recently emerged algorithms to characterize the support of a high-dimensional distribution. Through a kernel function, these methods map data points from the original space to the high-dimensional feature space and find a sphere with the minimal radius which contains most of the mapped data points. This sphere, when mapped back to the original data space, can be separated into several components, each enclosing a separate cluster of points [6]. For its support to any shape of data sets, the SVC has been successfully applied to solve some difficult and diverse clustering or outlier detection problems.

Many studies have shown that both solving the optimization problem and labeling the data points with cluster labels consume too much time and are, therefore, two major bottlenecks to the SVC's application [5, 6, 7, 9, 10, 11, 12, 13]. When solving the optimization problem, the noise data points not only increase the redundancy of describing a data set but also affect the abilities of classic algorithms in supporting for geometrical shape and membership decision; for instance, they would lead to overfitting [14]. Despite of these side effects, the SVC-based methods [2, 4, 17, 5, 6, 9, 11, 15, 16] treat the noise data points as outliers for lack of a clear definition. These approaches that confuse these two concepts of noise data points and outliers will cause the removal of outliers like noise data points. Yet, according to the principle of the SVC, the outliers, as shown in Figures 1 a) and 1 b), are different from noise data points and affected by the parameters of the convex quadratic minimization problems. Nath et al. [18] proposed a preprocessing method which exploited the geometry based on R*-tree to reduce the size of training data and resulted in drastically reducing the run-time of the cluster algorithm. With several pre-specified parameters, including the number of nearest neighbors k and the average strength of all data points, Wang et al. [19] presented a data preprocessing method which employs a shared nearest neighbor (SNN) based algorithm [20] and the concept of unit vectors to eliminate insignificant data points from the data set; but different pre-specified parameters could lead to extremely distinguished clustering results.

On the other hand, the complete graph-based (CG) strategy [2] and proximity graph-based (PG) strategy [21] are two popular cluster labeling algorithms. Although it is relatively easy to be implemented, the CG will suffer the drawback of becoming highly intensive (usually $O(N^2)$) as the number of the data points or free support vectors increases. In contrast with the CG, the PG can significantly reduce the cluster labeling time (usually $O(N)$ or $O(N \log N)$). Unfortunately, it fails frequently in the accuracy of cluster labeling and becomes highly complex while handling the high dimensional data sets. In sight of these problems, based on a topological property of a trained kernel radius function, Lee et al. [5] presented a robust labeling method, called Reduced Complete Graph (R-CG) for the SVC. In the R-CG, the stable equilibrium points (SEPs), each one representing a small number of disjoint groups decomposed from a given data set, are always found after a series iterative calculations on the trained kernel radius function. Although R-CG is one of the fastest algorithms with low time complexity of $O(N \log N)$, only SEPs are used to find the connected components among data points which would lead to relatively big error. In dealing with clusters of low connectivity as illustrated in Figure 1 b), the R-CG algorithm misclassifies it into two disconnected clusters. Therefore, in spite of representing data within their neighborhoods well, the SEPs are not proper substitutes for support vectors (SVs). As a knowledge base for future classifications or membership decisions, it would cause the accumulation of incorrect results [5].

In order to improve the performance, Lee et al. [6, 8] defined a transition point between two SEPs to check the connectivity of any two neighboring basin cells. Then, in [7], a weighted graph and attractors were constructed to make cluster labeling more robust. However, these methods, consuming too much time in searching the redundant SEPs (about 20-30 times of computation in feature space before achieving the exact SEP for each data point(see Table 3)), are much slower than the R-CG for the vast operations in addition. Anyway, as well as the poor performance on drawing the profile of datasets irrespective of noise data points, the drawback shared by these aforementioned methods [5, 6, 8, 7, 12] is that they can not label data sets with irregular shape or are accompanied by noise data points effectively. Another way of reducing the complexity suggested by [17] is to label the data points using the nearest neighboring labeled data points or the related SEP. Unfortunately, since the data points in clusters are sparse or imbalanced, that simple labeling strategy could not perform well while data points have two or more subequal distances to points in different clusters (see Figure 1 a)).

By adopting the multi-sphere structure, Jung Hsien [9] enriched the labeling algorithm in a form of cluster cell growing method which works in the input space and is able to represent the natural grade of membership in partition. Then it was implemented in handwriting recognition and document categorization by [22]. Practically, they chose another way of iterative constructing centroids and surfaces and searching the nearest neighbors when the cells are growing up. Obviously, similar to the algorithm in [10], they are both time-consuming.

In summary, the critical problems, ineffectively dealing with noise and outliers, inappropriately representing data sets with different shape and imbalanced distribution, are the bottlenecks for SVC based algorithms.

To overcome these problems and simplify clustering operations without losing efficiency and robustness, in this study we propose a novel scheme to accelerate SVC for large-scale datasets by two phases. In the first phase, to reduce the size of training data without affecting the profiles of clusters, we eliminate the noise data points which are clearly defined and differ from the outliers. The second phase starts following the resolution of optimization problem. In parallel, the connected components and stable equilibrium points are found by employing the support vectors. Then, the whole set of support vectors is decomposed into a number of subsets. In each subset, a DBC of shape and density is constructed to represent one cell. For a better support of any shape and distribution, the remaining data points (core points, outliers) could be labeled following the principle of maximum subordinated degree among the cells with a weighted distance in the input space. According to these two phases, the DBC, less than 5 % of the whole data points, represent the data set well and reduce the complexity of membership decision immensely. With a small knowledge-base of labels, the proposed method can support semi-supervised clustering efficiently.

The paper is structured as follows. The principle of the SVC is briefly reviewed in Section 2. Section 3 presents the scheme of two phases of accelerating the SVC, as well as a short suggestions on the approach of the semi-supervised clustering. We show experiments results in Section 4. The conclusions are drawn and the future work is discussed in Section 5.

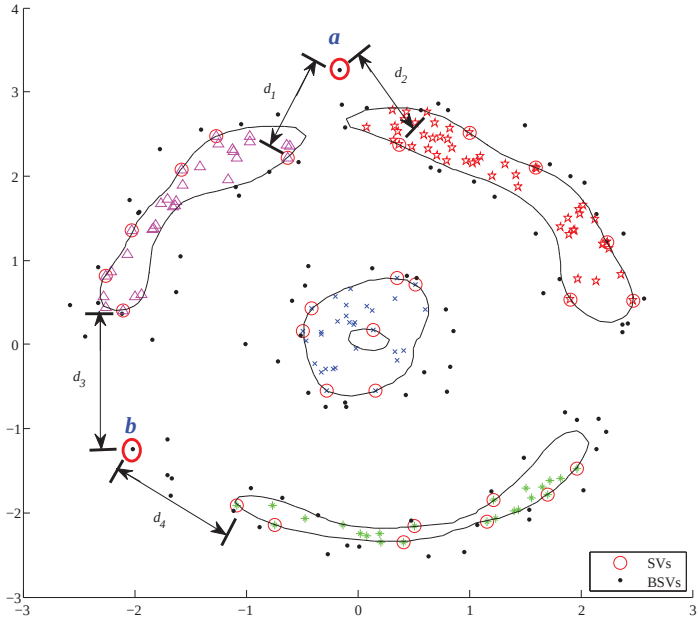
2 SUPPORT VECTOR CLUSTERING

This overview of support vector clustering follows closely the derivation of [2]. Firstly, assume N points $\{x_1, x_2, \ldots, x_N\}, x_i \in \Re^d$ in a data set described in the input space, where d is the dimension of the data space. The procedure of clustering these data points by the original support vector clustering algorithm with CG strategy could be summarized as follows.

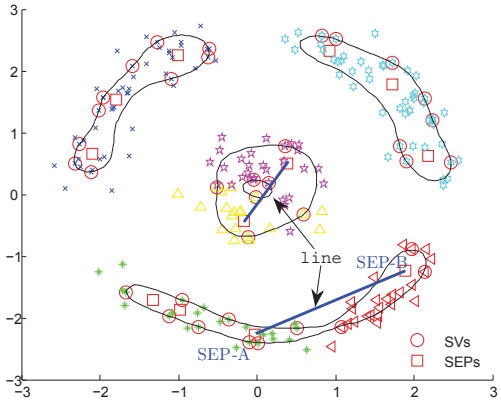
A nonlinear mapping function Φ is used to map the data set into a high-dimensional feature space such that the radius of the hypersphere, denoted by R, enclosing all the data points, is as small as possible. This objective can be cast as a convex quadratic minimization problem:

$$\begin{aligned} &\min_{R,\mu,\xi_j} R^2 + C\sum_j \xi_j \\ &\text{s.t.}\, ||\Phi(x_j) - \alpha||^2 \leq R^2 + \xi_j, \quad \xi_j \geq 0, \forall j \end{aligned} \tag{1}$$

where $||\cdot||$ is the Euclidean distance, α is the center of the hypersphere, ξ_j are the slack variables that loosen the constraints to allow some data points lying outside the hypersphere, C is a hyper-parameter and $C\sum_j \xi_j$ is a penalty term controlling



a)



b)

Fig. 1. Drawbacks of CG and R-CG algorithms. a) CG Algorithm ($q = 2$, $C = 0.08$). The clusters of a 2D data set *ring* (26 new data points are added) [5] assigned by CG algorithm which treats outliers as noise data points with no labels. The subequal distances (d_1, d_2 or d_3, d_4) between unlabeled data points a, b and several clusters may cause the labeling strategy of [17] fail. b) R-CG Algorithm ($q = 2$, $C = 0.1$). Incorrect result occurs to R-CG algorithm while the segmers on the line segment between two SEPs do not satisfy the condition of connected components.

the noise data points[1] in general. The larger the $C(C \in [0,1])$ is, the less the points whose images lie outside the hypersphere are. The solution to the primal optimization problem (1) can be obtained by solving its dual problem [23]:

$$\begin{aligned} \max_{\beta_j} \quad W &= \sum_j \Phi(x_j) \cdot \Phi(x_j)\beta_j - \sum_{i,j} \beta_i \beta_j \Phi(x_i) \cdot \Phi(x_j) \\ &= \sum_j K(x_j, x_j)\beta_j - \sum_{i,j} \beta_i \beta_j K(x_i, x_j) \\ &\quad \text{s.t.} \sum_i \beta_i = 1, \quad 0 \le \beta_i \le C \end{aligned} \tag{2}$$

where the inner product of $\Phi(x_i) \cdot \Phi(x_j)$ can be replaced by a kernel $K(x_i, x_j)$ which usually is the Gaussian kernel:

$$K(x_i, x_j) = \exp(-q||x_i - x_j||^2) \tag{3}$$

where q is the width of the kernel. For any points x in the input space, the distance from its image in the feature space to the center of the hypersphere is given by

$$R^2(x) = ||\Phi(x) - \alpha||^2 = K(x,x) - 2\sum_j \beta_j K(x_j, x) + \sum_{i,j} \beta_i \beta_j K(x_i, x_j). \tag{4}$$

The radius R of the hypersphere can be obtained by

$$R = \{R(x_j) | x_j \text{ is a support vector}\}. \tag{5}$$

In the aforementioned equations, β_j is the Lagrange multiplier. The Karush-Kuhn-Tucker (KKT) conditions, necessary and sufficient for optimality of the dual, lead to a partitioning of the training data points into three categories as given below. The data points with $\beta_j = C$ are the bounded support vectors (BSVs, or outliers), $\beta_j = 0$ are the points lying in the hypershpere called the inners, while all the other data points with $0 < \beta_j < C$ are treated as SVs. However, the BSVs, contributing in solving the optimization problem, should not be treated as the noises.

Upon solving the dual problem, a simple graphical connected-component method is used in cluster labeling. If there are two data points, x_i and x_j, we can check the segmers sampled on the line segment between them by traveling its image in hypersphere. According to Equation (4), x_i and x_j should be labeled by the same cluster index while all the L segmers (usually 10-20 points) are always lying in the hypershpere. Two data points, x_i and x_j, satisfying the above condition are defined as connected components. An adjacency matrix A is defined to identify the connected components of a cluster. We define the components of A, a_{ij} between

[1] Generally, the traditional SVC-based algorithms do not distinguish outliers from noise data points. So we follow this concept in this section.

pairs of x_i and x_j with $y_k(k \in [1, L])$ on the line segment:

$$a_{ij} = \begin{cases} 1, & \forall y_k, R(y_k) \leq R \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

In the matrix A, $a_{ij} = 1$ means that x_i and x_j belong to the same cluster. Otherwise, they are in different clusters.

3 THE PROPOSED METHOD

3.1 Phase I: Eliminating Noise Data Points

In order to eliminate the noise data points, we introduce a case to show the characteristics of such data points. As depicted in Figure 2, there are two clusters denoted by S_A and S_B. Obviously, x_1 is particularly different from others for its location. According to the traditional cluster labeling algorithms [2, 10, 9], after solving the optimization problem of Equation (2), x_1 will be labeled identically as the outliers x_4, x_5. However, the more the data points like x_1 are, the more computations would be consumed. In contrast to the outliers, whose size is controlled by the parameters C and q, the noise data points have some particular features which could be observed in Figure 2 a). Firstly, from the regions of the two dot-line circles regarding x_1 and x_2 as centers respectively and R as the radius, a careful observation would find the distinguished amounts of neighborhoods around x_1 and x_2. Accordingly, the frequency of x_1 appearing in the neighborhoods of the other data points would not match up to x_2 in a special region. Furthermore, the two pairs of the subequal distances, $\{d_{12}, d_{13}\}$ and $\{d_{1A}, d_{1B}\}$, would cause the decision failed, especially for those traditional distance-based labeling algorithms, such as those in [6, 10].

Apart from the noise data point, following the concept of unit vectors, Wang and Chiang [19] exploited a preprocessing algorithm to eliminate the core points for reducing the size of dataset. However, when confronted with a scenario illustrated in Figure 2 b), this algorithm is not recommended to eliminate the core points for it would destroy the distributional characteristic of a dataset. Two core points, denoted by x_A and x_B in Figure 2 b), are lying in clusters S_A and S_B, respectively. x_4 and x_5 are the outliers of the cluster S_B and the inners of cluster $S_I(\neq S_B)$. In the former case, x_4 and x_5, as outliers, are labeled by the index of S_B, while d_i is greater than their distance to S_B. With the core elimination algorithm, S_A will be split into S_{A1} and S_{A2} with the removal of x_A and x_B. In order to maintain the same cluster label of S_{A1} and S_{A2}, an appropriate kernel width q will be found and that would usually cause the border of S_B to be enlarged to $S_B^{'}$. Then, x_4 and x_5 will be the inners of $S_B^{'}$. If there is another cluster near the S_B, the algorithm would impose or misguide the decision of cluster labeling. The reason is that the value of d_i is large enough to impose the parameter setting of q on finding the bifurcations of clusters. The compact relationship between the initial value of q and distances among data

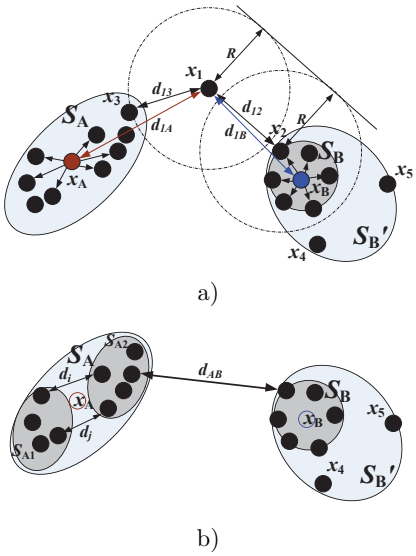


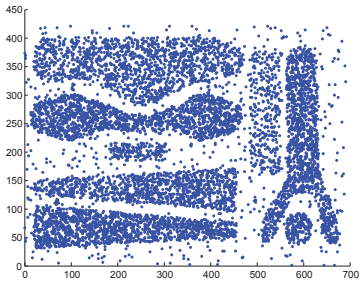
Fig. 2. Analysis of Noise Data Points and Core Data Points. a) Noise Analysis. S_A and S_B are two clusters covering an elliptical region and a circular area with dotted line, respectively. x_A and x_B are the core of S_A and S_B, respectively. x_1 is a data point outside the two clusters. x_4 and x_5 are outliers specified by SVC with width q_1 of Gaussian kernel function in Equation (3). b) Core Analysis. After the elimination of the core points by algorithm in [19], x_A and x_B denoted by the circle with the dot-line are removed. However, without the data point x_A, S_A has been divided into two clusters denoted by S_{A1}, S_{A2}. $d_i(= d_j)$ is the minimum distance between S_{A1} and S_{A2}

points had been discussed in [2, 17]. Therefore, a suitable core points elimination algorithm, not in the scope of this article, is expected to be able to maintain the profile of a data set and perform well without negatively affecting the clustering.

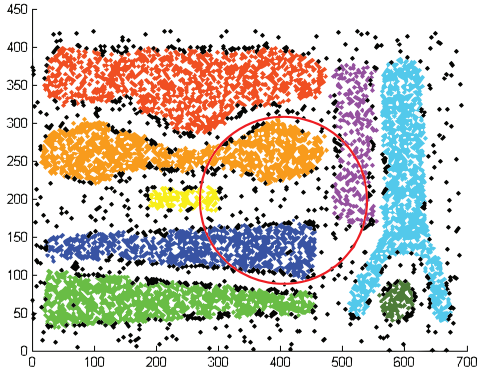
Throughout the aforementioned analysis, the proposed method of noise elimination algorithm should distinguish noise data points from outliers and provide with a self-adapting strategy to deal with any dataset with sparse or imbalanced distribution. Before introducing the noise elimination algorithm, we first give a novel definition of noise data points for the proposed algorithm.

Definition 1 (Noise Data Points)**.** Noise data points are those sparse distribution of data points, which are seldom surrounded by similar data points in a certain or local region, and have a relatively balanced membership degree to their adjacent clusters.

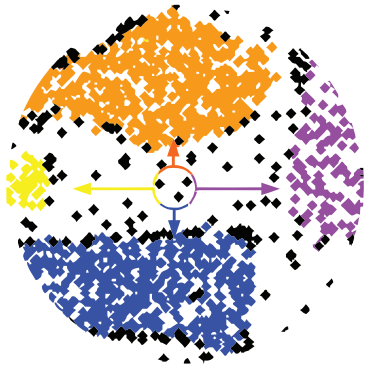
In order to eliminate these data points, different from using SNN algorithm to obtain similarity matrix whose components are defined as the similarity measure


450
400
350
300
250
200
150
100
50
0
0
100
200
300
400
500
600
700

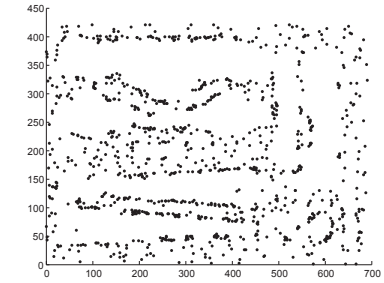
a)


450
400
350
300
250
200
150
100
50
0
0
100
200
300
400
500
600
700

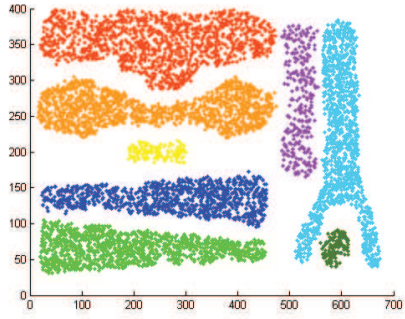
b)



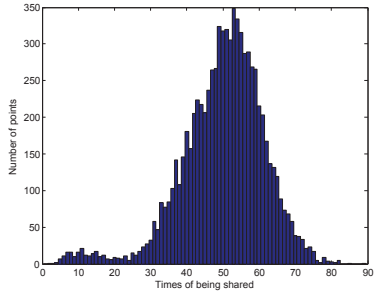
c)



d)



e)



f)

Fig. 3. Definition of noise data point. a) Origin t8.8k. The origin data set t8.8k is a 8 000*2 matrix. b) Clusters with noises. Processed by the proposed algorithm, $K = 50, \lambda = 0$, data points are shown in the form of clusters with different colors and noise data points denoted by black points. c) Balanced memberships. Noise data points perform a relatively balanced membership degree to their four adjacent clusters. d) Sparse distribution of the noise data points. e) Data without noise points. The remaining data points after the removal of noise data points. f) Histogram of being shared. The histogram shows the distribution of the frequency of data points being shared by others.

between neighboring points [20, 24, 19] in local region, we extend the similarity measure to the entire data sets in another way called similarity level of being shared (SLS). Specifically, if X_i and X_j are two data points in dataset X, the strength of their similarity level is defined as

$$sim\_level(X_i, X_j) = \frac{\min\{KNN_i(X), KNN_j(X))\}}{\max\{KNN_i(X), KNN_j(X))\}} \tag{7}$$

where $KNN_i(X) = \sum_{\substack{r\in[1,N]\\ r\neq i}} KNN(X_r)$ is the notation of point X_i occurring in X_r's k nearest neighbor lists denoted by $KNN(X_r)$, N is the number of the data points in X. Obviously, the similarity level reflects the similarity degree of any two data points being shared by others.

According to the definition, the proposed noise elimination algorithm could be expressed by the following steps:

Step 1: Initialization. Set K and get the lists of K nearest neighbors for each point.

Step 2: Calculating the average degree of points' being shared by others. Here, we denote this parameter by α:

$$\alpha = \overline{freq(X)} = \frac{1}{N}\sum_{i=1}^{N} KNN_i(X). \tag{8}$$

Step 3: Removal of noise data points. With an adjustment factor λ, the threshold δ of low-similar data points is defined as

$$\delta = \lambda\alpha \tag{9}$$

where λ is between 0 and 1, default value is:

$$\lambda = \begin{cases} 1, & \text{if } KNN_i(X) \geq \alpha \\ 1 - \frac{\{\frac{1}{N-1}\sum_{i=1}^{N}[KNN_i(X)-\alpha]^2\}^{\frac{1}{2}}}{\alpha}, & \text{if } KNN_i(X) \leq \alpha. \end{cases} \tag{10}$$

For a specific point X_i, if the ratio of SLS to the average degree is lower than λ, $KNN_i(X) \leq \lambda\alpha$, X_i will be considered as noise and removed from the data set in the preprocessing stage. Actually, to achieve an complete judgement for the whole data set, in the proposed scheme, these data points will be collected into a candidate set and will be assigned labels by the following algorithm of Phase II.

By employing t8.8k, one of the chameleon datasets [25], Figure 3 shows the principle of the proposed noise elimination. The original data set which is plotted in Figure 3 a) contains 8 000 data points. Since K is 50, the proposed algorithm checks 50 nearest neighbors for each data point to find out the degree of being shared by others (see Figure 3 f)). While λ is set to 0, the noise data points are denoted by black

points in Figure 3 c). To verify the proposed definition of noise data points, a close insight into the annular region of Figure 3 b) is depicted in Figure 3 c). Obviously, the noise data points perform a relatively balanced membership degree to their adjacent clusters. The sparsity of noise data points can be proved by Figure 3 d). After the removal of noise data points, the remaining data points shown in Figure 3 e) would keep the profile of the data set. Moreover, the profile of each cluster is becoming much clearer and smoother such that the separability among clusters is enhanced without compromising the labeling result.

3.2 Phase II: Labeling the Data Points in Double Centroids Mode

Considering the kernel function of Equation (4), there exists a unique solution (or trajectory) $x(\cdot) : \Re \to \Re^n$ for each initial condition $x(0) = x_0$ is guaranteed since it is twice differentiable and the norm of $\nabla R^2(x)$ is bounded. Lee et al. [5] called the state vector $\overline{x}$ satisfying the equation $\nabla R^2(\overline{x}) = 0$ equilibrium point (EP) and a SEP if all the eigenvalues of its corresponding Jacobian matrix, $J_R(\overline{x}) \equiv \nabla^2 R^2(\overline{x})$, are positive.

Instead of CG strategy, SEPs, generated by iterative searching for local minimums from all the data points in the data set, are used to find the connected components. For the characteristic of convergence, all the data points could be assigned the same clustering labels as the SEP which would converge to [5, 8]. It is an excellent characteristic that would be improved and employed in the proposed procedure of constructing double centroids.

3.2.1 Constructing Double Centroids

Out of the need of computational savings, the bottlenecks, too many redundant iterative computations, low connectivity of clusters and imbalanced distribution of data sets, should be resolved appropriately. Therefore, the proposed cluster labeling algorithm uses only support vectors to check the connected components and find the SEPs which are treated as shape centroid.

From Equation (5), the distance from any data point in a sphere[2] to the center α of the hypersphere is lower than R. This means we can find at least one data point, in each sphere, which has the shortest distance to the center of the hypersphere. Therefore, we can extract the exact data points from a data set by any local optimization algorithm. However, for some particular distributions, one cluster with one sphere may not support all the data set patterns appropriately. Lee [9] thus proposed a cell growing model to support one cluster with multi-sphere. However, in high dimension feature space, the sphere is expected to be a regular one to get an effective performance when calculating the norm of membership degree. Therefore, it is not good at dealing with data set of strange shape and distribution. In

[2] In this paper, we use "cluster" to represent one class of data points, "sphere" defined in the input space following the reference [9] to denote a number of connected components in a cluster, "cell/basin cell" following reference [8] as a part of a sphere.

order to improve the performance, as well as the deviation of one cluster to multiple sphere, the proposed method decomposes each sphere into multiple cells.

Definition 2 (Shape Centroid)). The shape centroid (SC), the same as SEP in [6], is a logic data point lying in a basin cell defined in a fresh way:

$$SC(C_{ij}) = \{s | s \in \Re^n, \lim_{\substack{x_i \in C_{ij}, C_{ij} \in S_j, \\ t \to \infty}} x_i(t) = s\} \tag{11}$$

where C_{ij} is the i^{th} basin cell unit divided from the sphere S_j.

Lemma 1. Deduced from the principle of convex set and gradient line affecting the steps to each local minimum, the boundary of the basin cell, denoted by $BD(C_{ij})$, can construct any data point lying in C_{ij} though a convex combination.

Proof. Suppose there are two data points, x_a and x_b, on the boundary of a basin cell $BD(C_{ij})$. Taking the result of support vector clustering, the penalty parameter $C = 1$ to avoid the outliers[3], x_a and x_b is in the same sphere of a cluster. This means any data point x_i in the basin cell has a distance R_i to the center in hypersphere. Therefore, there always exists a $\lambda \in [0, 1]$ setting up the following formula.

$$\exists \lambda \in [0, 1] \quad \rightarrow \quad R_i \leq R(\lambda x_a + (1 - \lambda) x_b) \tag{12}$$

On the contrary, the data points on the boundary of a sphere would not satisfy that equation because of concave boundary (see Figure 4), so that we can always get a number of data points $\{x_1, x_2, \cdots, x_n\}$ from the boundary of a basin cell $BD(C_{ij})$ to construct a convex set $CS(C_{ij})$ which can compose all the data points $X_{bc_{ij}}$ lying in it. Furthermore, a sphere could be divided into one or more cells each consisting of a number of vertexes and the corresponding convex combinations.

$$X_{bc_{ij}} = \sum_{i=1}^{n} \alpha_i x_i, \quad \sum \alpha_i = 1 \tag{13}$$

□

Lemma 2. For a strongly separable data set, the following requirement should be met:

1. The number of clusters is finite and much less than the data set size;
2. Without considering the overlapped cases, each cluster should have a profile consisting of a number of data points to represent itself (not affected by the cells);
3. To draw the profile of clusters, the number of employed data points should be relatively stable and independent upon the data set size.

[3] The parameter of $C = 1$ is not a necessary condition.

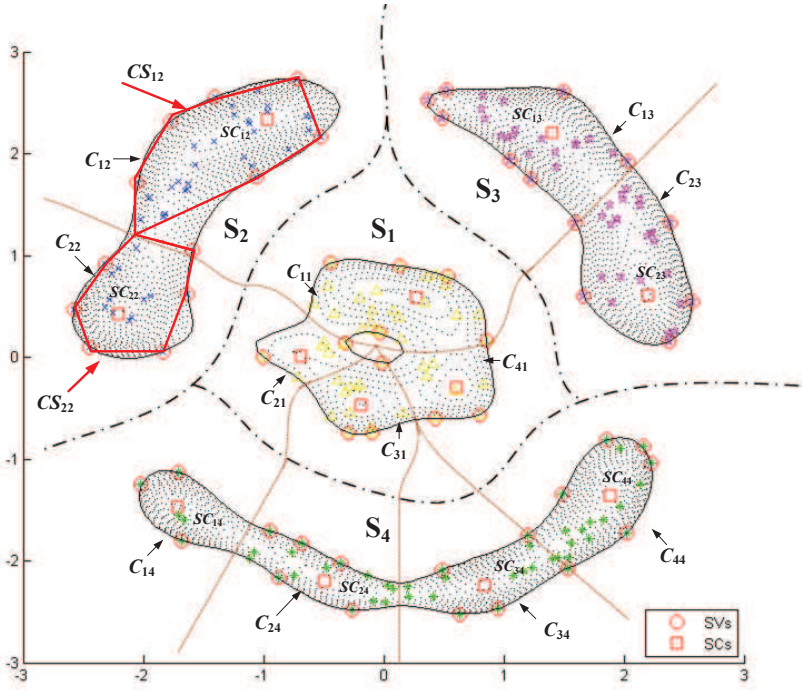


Fig. 4. Convex analysis of basin cell. In sphere S_2, there are two shape centroids $\{SC_{12}, SC_{22}\}$ lying in each owner basin cell $\{C_{12}, C_{22}\}$. With a number of support vectors, we can construct two boundaries of convex sets $\{CS_{12}, CS_{22}\}$ to enclose all the data points lying in the two basin cells. The distance, taking $dist_{12}$ and $dist_{13}$ for example, between the shape centroid and density centroid of a cell is suggested to reflect the degree of imbalanced distribution. The larger the distance for a cell, the more the imbalanced distribution of data points.

Proof. Lemma 2 lists three sufficient conditions for a data set to be strongly separable that could be proved by contradiction.

Obviously, the first condition is established. If there is no profile to be used to represent each cluster, the data set is indistinguishable. According to Lemma 1, any data point, lying in a cluster, could be constructed by a convex combination of vertices from the profile. Assuming there is a data point, x_{new}, there is no convex combination of vertex set $CS(C_{ij})$ which consists of $x_{i1}, x_{i2}, \ldots, x_{ik}$ in data set $\Re^d$. If the label of x_{new} is the same as $x_{iq}(q \in [1, k])$, it must be a vertex in $CS(C_{ij})$. While updating the $BD(C_{ij})$, the original data points would lie in a new profile which can enclose x_{new}. In contrary, x_{new} should be subsumed in another cluster or removed as a noise data point. Therefore, the second condition is established. Furthermore, if the profile of a cluster is not relatively stable to reflect the characteristics of the

objective data set but alters with the increase of data size, either the features of the data set are sampled incorrectly or the data set has too sparse and imbalanced distribution to be distinguished. One of those separable data sets would have a stable profile constructed by support vectors (see columns 2–3 of Table 1). The distinct profiles would be maintained as the size of data set increases. □

Definition 3 (Density Centroid)**.** The density centroid (DC) is a logic data point defined by Equation (14) for a cell:

$$DC(C_{ij}) = \frac{1}{N_{ij}} \sum_{i=1}^{N_{ij}} x_i, \quad \forall x_i \in C_{ij} \tag{14}$$

where N_{ij} is the number of data points in the basin cell C_{ij}.

With a new data point to join the basin cell, the density centroid will get close to the highest density position of the basin cell dynamically. Following the probabilistic theory, in this study, only the support vectors are used to construct the density centroid. Certainly, the density centroid would never locate out of its basin cell.

3.2.2 Membership Decision

Generally, the two centroids of shape and density would not overlap in data set with imbalanced distribution. The greater the gap of the distance between the two centroids in a cell, the much more imbalanced the distribution is (see Figure 4). In consideration of the different contributions from the two centroids, a simple linear programming problem is constructed and solved by a series of support vectors to quantify the weight of each centroid. Then, a normalized contribution of the two centroids, without noise points especially, which is defined in Equation (15) would express both distribution and shape for a data set.

$$\begin{aligned} \min \quad & \frac{1}{N_{ij}} \sum_{i=1}^{N_{ij}} \left(W_{SC_{ij}} \parallel x_i - SC_{ij} \parallel^2 + W_{DC_{ij}} \parallel x_i - DC_{ij} \parallel^2 \right)^{\frac{1}{2}} \\ & \text{s.t.}\, W_{SC_{ij}} + W_{DC_{ij}} = 1 \\ & W_{SC_{ij}} \geq \frac{1}{2}, \quad W_{DC_{ij}} \geq 0 \end{aligned} \tag{15}$$

where $W_{SC_{ij}}$ and $W_{DC_{ij}}$ are defined to measure the contributions of shape centroid and density centroid in a basin cell C_{ij}, respectively. The constrained condition of $W_{SC_{ij}} \geq \frac{1}{2}$ is used to emphasize the contribution of shape centroid. Based on the distance measurement, the optimization problem could be solved either in the input space or feature space. In the feature space, the distance can be calculated by

$$\begin{aligned} \parallel \Phi(x_i) - \Phi(x_j) \parallel^2 &= K(x_i, x_i) - 2K(x_i, x_j) + K(x_j, x_j) \\ &= 2 - 2 \cdot exp(-q \parallel x_i - x_j \parallel^2) \end{aligned} \tag{16}$$

where x_j denotes one of the centroids SC_{ij} and DC_{ij}. However, for computational savings, the computation of the distance in the input space is recommended.

After solving the linear programming problem, the membership decision following the principle of the maximum subordinated degree could be simply expressed as follows:

Step 1: Picking up a data point x from the candidate dataset or the input data set excluding noise points and support vectors.

Step 2: Calculating the weighted norm distance $dist(x, C_{ij})$ between x and the cells C_{ij}

$$dist(x, C_{ij}) = \left(W_{SC_{ij}} \parallel x - SC_{C_{ij}} \parallel^2 + W_{DC_{ij}} \parallel x - DC_{C_{ij}} \parallel^2\right)^{\frac{1}{2}}. \quad (17)$$

Step 3: Labeling x with the label of its nearest cell with minimum weighted norm distance.

$$label(x) = label(\arg\min_{C_{ij}}\{dist(x, C_{ij})\}). \quad (18)$$

3.2.3 Suggestions on Semi-Supervised Clustering for Future Approaches

As mentioned above, the double centroids mode is suitable for standing for cells and then to represent clusters at the minimum cost. Practically, for a better approach to represent a dataset, we investigate this mode in semi-supervised clustering for some special purposes:

- Support vector clustering is poor in handling several overlapped clusters for the kernel function. If we have got a number of labeled data points which is close to the exact size of centroids, the proposed method would save much more time at the minimum error rate by computing in the input space. The comparisons are listed in Table 3.
- Usually, a data set should be decomposed into a number of spheres, such as *iris* [26]. Although the traditional support vector clustering based algorithms could recognize the spheres, they cannot establish the relationship among these spheres.

For those reasons, the semi-supervised clustering algorithms are the subject of attention of researchers and are surveyed in [27]. In the semi-supervised clustering mode, the proposed method is recommended to be employed as the following steps:

Step 1: Choosing a threshold value γ for merging spheres.

Step 2: Utilizing the proposed method to cluster the data set without considering labels such that labeling would only be done among the support vectors.

Step 3: A number of cells then can be labeled by the known labels or merged into one. In addition, the maximum weighted norm distances (denoted by $r_{C_{ij}}$ for cell C_{ij}) from labeled data points to their cells should be computed by Equation (17) and referred to in the following decisions.

Step 4: Finally, the remaining data points will be labeled in sequence. If the distance from a labeled cell C_{ij} to an unlabeled cell is lower than $r_{C_{ij}} + \gamma$, the two cells are suggested to be merged into one.

4 EXPERIMENTS

4.1 Experimental Program

In order to evaluate the effectiveness of the proposed method, denoted by DBC or DBC with noise elimination (DBC-NE, the DBC labeling algorithm following noise elimination), we conducted a series of experiments on 7 data sets. The target of this study is to achieve fair contrast evaluations with the state-of-the-art labeling methods. Therefore, all the data sets would be preprocessed by the proposed noise elimination (NE) algorithm. The DBC-NE will be evaluated with a subset which is extracted from the data set and have unclear profile (see Table 1).

t4.8k, t5.8k, t7.10k and *t8.8k* are *Chameleon* data sets obtained from [25] and the *ring* data set is from [5]. *iris, wine and wisconsin* are widely used clustering or classification data sets from [26].

The proposed method is compared with five different unsupervised SVC methods: Complete Graph (CG) [2], Delaunay Diagram (DD) [28], Minimum Spanning Tree (MST), K-Nearest Neighbors (K-NN) [21], and Reduced Complete Graph (R-CG) [5] algorithms.

To analyze the time complexity of the proposed method, let m (approximately 20) be the average number of iterations for each data point to locate its corresponding local minimum (shape centroid) via the steepest decent process [5], N be the number of data points in the data set, l be the sample rate (20 is used here) between any two support vectors. The time complexity of the noise elimination procedure is $O(N^2)$. After the removal of noise data points, the remaining data points which keep the profile of the original data set could accelerate the process of solving the optimization problem in Equation (2). Although the number of local minima normally depends on the size of the width parameter of a Gaussian kernel function [5], a clear and separable profile achieved by the proposed noise elimination algorithm simplifies the setting of parameters. Consistent with the principle of SVC and the analysis of Section 3.2, the number of support vectors, denoted by n_{SV} and ranged from 3 % to 15 % of total size (see column 1 of Table 1), is relatively stable in representing the profile of a data set. Actually, the number of shape centroids, in column 2 of Table 1, is even more stable than the size of support vectors. The greater the amount of data points, the more obvious the stable size of support vectors is. Then the time complexity is $O(l \cdot n_{SV}^2 + mn_{SV} + (N - n_{SV})n_{SV})$ which could approximately be about $O(l \cdot n_{SV}^2)$ for large-scale data sets (because of $mn_{SV} < l \cdot n_{SV}^2$ and $(N - n_{SV}) \leq l \cdot n_{SV}$).

This observation implies that the total time complexity of the proposed method desponds on the size of support vectors, which is solely depending on the shape of the distribution of a data set.

4.2 Experimental Results

For each method, the performance is affected by different parameters. In order to make a fair comparison with the other methods, the parameters of each method, q and C, are set very close but different to achieve their best performances.

t7.10k, *t4.8k*, *t5.8k*, and *t8.8k* are large-scale data sets of *Chameleon* which are used in the evaluation of noise elimination with different K. As depicted in Figure 5 a), as well as $K \geq 30$ for *t7.10k*, the curves become smooth while K is greater than 25 for *t4.8k*, *t5.8k*, and *t8.8k*. Numerous experiments show that the number of remaining data points ranges from 0.3 % to 0.5 % with different K. Therefore, with the removal of noise data points, the number of remaining data points is relatively stable as K increases. Similarly, the smooth curves of time cost shown in Figure 5 b) suggest that the time delay is not so much sensitive to K.

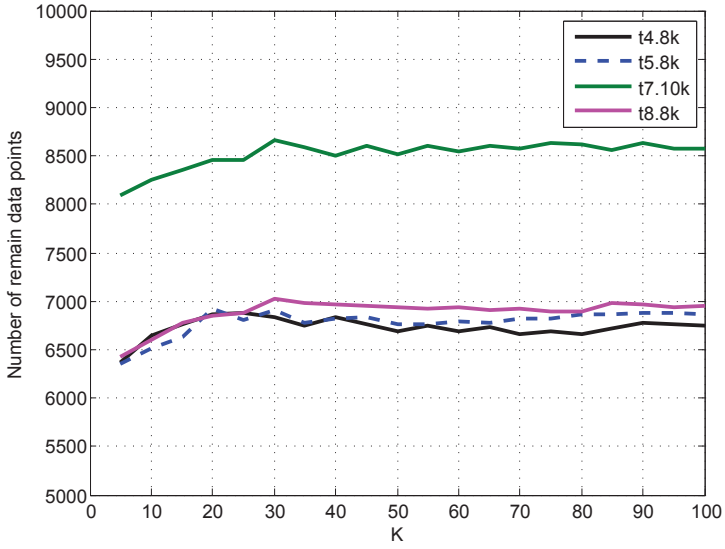
In order to compare the performance on the time cost of cluster labeling for the six methods, we conduct ten evaluations on *t4.8k* (2D-data set) without noise data points that could be observed from Table 1, Table 2 and Figure 6. With an addition of 500 samples for every round, the number of experimented data points ranges from 500 to 5 000. The proposed methods, both DBC and DBC-NE, are evaluated because the randomly extracted data points might have unclear profile. The criteria for comparisons are the cluster labeling error rate and the average CPU time of cluster labeling. In Table 2, the labeling error rate is the percentage of the mislabeled data with respect to the cluster labels determined by the trained kernel radius function. The experimental results and the complexity analysis demonstrate that the proposed method is substantially fast and sufficiently accurate in comparison with the conventional cluster labeling algorithms.

Data size	# of SVs	# of SCs	CG	DD	MST	4-NN	R-CG	DBC	DBC-NE
500	72	23	21.2	4.8	64.1	4.1	4.9	2.2	1.9
1 000	77	23	156.0	37.0	1 734.7	33.0	10.8	2.9	2.9
1 500	75	23	1 298.3	131.0	8 753.9	96.2	18.5	3.5	3.9
2 000	87	26	–	295.4	–	234.7	26.7	4.2	5.5
2 500	88	32	–	700.0	–	523.3	40.1	7.6	7.3
3 000	135	33	–	1 152.2	–	908.5	52.0	11.8	11.8
3 500	133	36	–	1 951.1	–	1 292.9	64.9	17.9	15.4
4 000	135	38	–	2 059.6	–	3 042.2	87.3	20.0	17.0
4 500	141	41	–	3 346.1	–	4 457.7	103.4	28.7	19.8
5 000	164	41	–	5 069.3	–	6 092.9	139.0	42.9	25.9

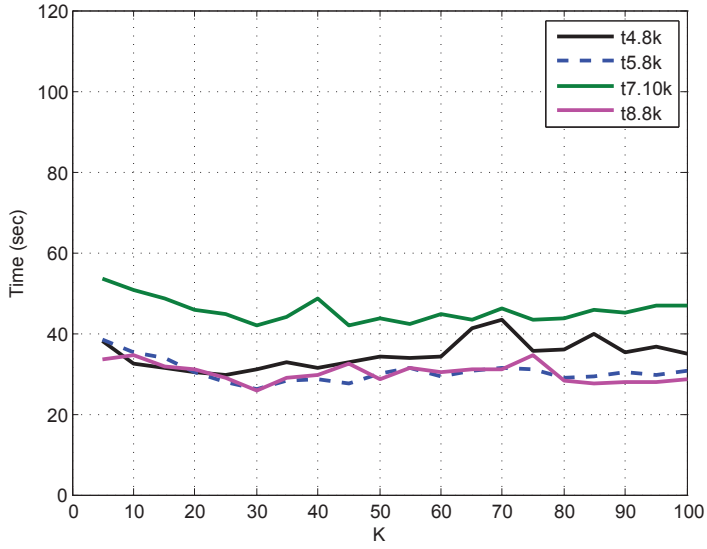
Note: "–" means not available.

Table 1. Cluster labeling time for t4.8k with different data sizes N

In addition to the noise elimination and the stable size of support vectors, another factor of speeding up the labeling procedure, shown in Table 3, is less time-consuming for labeling the inner data points and bound support vectors in the input



a)



b)

Fig. 5. Performance of different K in dealing with 4 data sets of chameleon. a) Data remained. shows the number of remaining clean data points after noise elimination. b) Time cost. shows the run-time for different K.

Data size	CG	DD	MST	4-NN	R-CG	DBC	DBC-NE
500	0	10	37	18.6	0	0	0
1 000	0	13.2	23	20	0.6	0	0
1 500	0	13.8	42.86	11.8	9.67	0	0
2 000	–	15	–	24.7	1.0	0	0
2 500	–	11.3	–	18.8	0.8	0	0
3 000	–	9.7	–	16.6	1.9	0	0
3 500	–	8.1	–	16.3	0.8	0	0
4 000	–	22.2	–	16.1	1.6	0	0
4 500	–	19.7	–	5.6	1.7	0	0
5 000	–	17.9	–	2.6	1.3	0	0

Note: "–" means not available.

Table 2. Cluster labeling error rate (%) for t4.8k with different sizes N (corresponding with Table 1)

space than these computations in the feature space. The ratio of CPU time required by labeling one point in the feature space and in the input space, denoted by $\frac{T_{R-CG}}{T_{DBC}}$, is consistently greater than 100. In consistency with the analysis of time complexity, the proposed method is mainly affected by the support vectors while the R-CG is affected by the size of dataset (see columns 2–3 of Table 3).

Data size	# of SVs	# of SCs	T_{R-CG} (sec.)	T_{DBC} (sec.)	$\frac{T_{R-CG}}{T_{DBC}}$
200	36	12	0.008294	0.000057	145.5
500	72	23	0.010069	0.000063	159.8
2 500	88	32	0.012641	0.000069	183.2
3 000	135	33	0.014629	0.000072	203.2
5 000	164	41	0.015235	0.000073	208.7
200	179	99	0.015650	0.000080	195.6
500	472	261	0.015303	0.000110	139.1

Table 3. Run-time for labeling one data point by R-CG [5] vs. the DBC

The final interest of our experiments is to show the performance of "DBC" on three high-dimensional data sets. To do this, we set the comparison only with R-CG [5] for too much time consumptions required by the others. The data sets, *iris* and *wine*, are 150 4-dimension and 178 13-dimension, respectively. The *wisconsin* consists of 683 9-dimensional feature values from two classes after removing the 16 samples with missing feature values. For the purpose of evaluating performance on large-scale datasets, we extend the three data sets by a convex combination within each class proportionally. The experimental results are detailed in Table 4 where the column of "proportion" is the proportion of the number of different class. Although the three data sets are not completely linearly separable, these evaluations are conducted without any additional preprocesses. From columns 6 to 9 of Table 4, the evidence that the relationship of one to many exists between class and

Data sets	data set description				# of SCs		# of Spheres		Time cost(sec.)		Error rate (%)	
	dims	size	proportion	classes	R-CG	DBC	R-CG	DBC	R-CG	DBC	R-CG	DBC
iris	3	100	1-1-0	2	2	2	2	2	2.5	***0.35***	0	0
		150	1-1-1	3	56	60	34	33	4.0	5.9	6	***2***
		2 700	1-1-2	3	40	39	7	7	90.6	***49.7***	1.73	1.73
		5 000	1-1-0	2	2	3	2	2	115.9	***2.0***	0	0
		18 750 000*	1-1-0	2	–	2	–	2	–	***889.1***	–	0
wisconsin	9	683	458-225	2	59	***11***	59	***8***	28.2	***6.5***	3.95	***3.22***
		4 000	1-1	2	444	***19***	266	2	456.4	***49.3***	0.03	***0.07***
		237 473*	3.4-1	2	–	***11***	–	***8***	–	***17.8***	–	***0***
wine	13	178	59-61-48	3	65	***3***	55	3	9.5	***1.3***	0.56	3.37
		3 000	1-1-1	3	965	***425***	684	418	716.9	***239.5***	11.13	***9.57***

Note: "*" denotes semi-supervised clustering, "–" means not available.
The size of labeled data points are 20 and 100 for *iris* and *wisconsin* , respectively.

Table 4. Cluster labeling time (sec.) and error rate (%) for high-dimensional data sets with different sizes

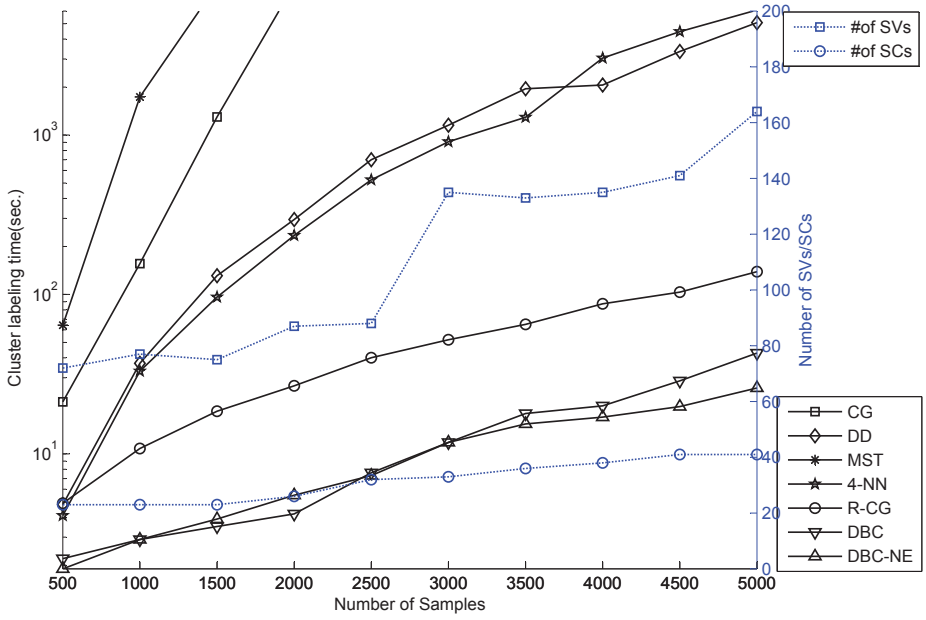


Fig. 6. Comparison of run-time among various SVC cluster labeling methods (left coordinate system) and changes of the number of SVs and SCs along with the size of samples increasing (right coordinate system)

spheres, sphere and cells can be found. The bold italic values highlight the following characteristics of the proposed method:

- The speed of the proposed method outperforms the R-CG algorithm [5] in deal with large-scale datasets. A fine adjustment of hyper-parameter C, lessening the profile [2], could make this better.
- Since only the support vectors are employed to search the SCs, the proposed method avoids too many local minimums effectively. In contrast, the great number of SCs found by R-CG (see column 6 of Table 4), especially for *wisconsin* and *wine*, slows the procedure of cluster labeling down seriously.
- By employing the semi-supervised clustering method proposed in Section 3.2.3, experiments on the expanded data set of *iris* and *wisconsin* show that the proposed double centroids mode is suitable for replacing the cells in clustering.
- To get a comparable error rate, the proposed algorithm shows significant advantages in both computational savings and describing the profile for large-scale datasets.

In summary, experimental results on the benchmark data sets confirm that the proposed method indeed accelerates the support vector clustering for large-scale datasets. Especially, for the data sets with stable profiles, the proposed method would outperform the traditional clustering algorithms significantly. Meanwhile, any cell of clusters can be well represented by shape centroid and density centroid.

5 CONCLUSION

In this study, a novel scheme for accelerating support vector clustering for handling large-scale datasets is proposed. The proposed scheme consists of two phases: a preprocessing of noise elimination and a double centroids labeling algorithm.

In the first phase, our studies discover the differences between the noise data points and the outliers in both the sparse distribution and the grade of different memberships. In view of these characteristics, a novel noise elimination algorithm is proposed to enhance the separability of a data set and dramatically reduce its size. With the removal of noise data points, a much clearer and smoother profile for each cluster can be obtained. Actually, one cluster may have several disconnected cells of imbalanced distribution. Therefore, in the second phase, we define a shape centroid and construct a density centroid to represent each cell of clusters. According to the analysis of convex hulls, we find that the distance between the shape centroid and the density centroid benefits the reflection of the degree of imbalanced distribution of a data set. Thus a double centroid labeling algorithm is presented which uses support vectors to locate the shape centroid, construct the density centroid and find connected components. In consideration of the different contributions of the two centroids, the proposed algorithm measures the difference by two weights calculated by solving a linear programming problem. Through a weighted distance measured in the input space, from data point to each cell, the remaining data points are assigned by the indexes of their nearest cells respectively. Compared with the state-of-the-art algorithms, both theoretical analysis and experimental results show that the proposed scheme is robust and efficient in dealing with large-scale datasets, especially for the data sets with strongly separable characteristic. Furthermore, the double centroids model is suitable to represent cell of clusters for the purpose of clustering for much more large-scale datasets in semi-supervised mode.

Lack of an accurate mechanism to disclose the structure or profile of any data set, an approach of the proposed method which can control the size of points (independent with support vectors) for drawing the profile besides tuning the hyper-parameter C remains to be investigated.

Acknowledgement

The authors would like to thank the handling editor and anonymous reviewers for helping to greatly improve the paper. This work is supported by the National Natural Science Foundation of China under Grant No. 60972077, the National High-Tech Research and Development Plan of China under Grant No. 2009AA01Z430, the

National S & T Major Program under Grant No. 2009ZX03004-003-03 and Xuchang Science and Technology Development Funds under Grant No. 1101030 and 1101059.

REFERENCES

[1] Burges, C. J.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, Vol. 2, 1998, No. 2, pp. 121–167.

[2] Ben-Hur, A.—Horn, D.—Siegelmann, H. T.—Vapnik, V. N.: Support Vector Clustering. Journal of Machine Learning Research, 2001, No. 2, pp. 125-137.

[3] Scholkopf, B.—Platt, J.—Shawe-Taylor, J.—Smola, A. J.—Williamson, R. C.: Estimating the Support of a High-Dimensional Distribution. Neural Computation, Vol. 13, 2001, No. 7, pp. 1443–1472.

[4] Tax, D. M. J.—Duin, P. R. W.: Support Vector Domain Description. Pattern Recognition Letters, Vol. 11-13, 1999, No. 20, pp. 1191–1199.

[5] Lee, J.—Lee, D.: An Improved Cluster Labeling Method for Support Vector Clustering. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 27, 2005, No. 3, pp. 461–464.

[6] Lee, J.—Lee, D.: Dynamic Characterization of Cluster Structures for Robust and Inductive Support Vector Clustering. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 28, 2006, No. 11, pp. 1869–1874.

[7] Lee, J.–Jung, K. H.—Lee, D.: Constructing Sparse Kernel Machines Using Attractors. IEEE Trans.Pattern Analysis and Machine Intelligence, Vol. 20, 2009, No. 4, pp. 721–729.

[8] Lee, D.—Lee, J.: Equilibrium-Based Support Vector Machine for Semisupervised Classification. IEEE Trans.Neural Networks, Vol. 18, 2007, No. 2, pp. 578–583.

[9] Chiang, J. H.—Hao, P. Y.: A New Kernel-Based Fuzzy Clustering Approach: Support Vector Clustering with Cell Growing. IEEE Tran. Fuzzy Systems, Vol. 11, 2003, No. 4, pp. 518–527.

[10] Hsieh, T. W.—Taur, J. S.—Tao, C. W.—Kung, S. Y.: A Kernel-Based Core Growing Clustering Method. International Journal of Intelligent Systems, Vol. 24, 2009, No. 4, pp. 441–458.

[11] Xu, R.—Wunsch, D. C.: Clustering. John Wiley & Sons 2008.

[12] Ling, P.—Zhou, C. G.—Zhou, X.: Improved Support Vector Clustering. Engineering Applications of Artificial Intelligence, Vol. 23, 2010, No. 4, pp. 552–559.

[13] Yeh, C.-Y.—Huang, C.-W.—Lee, S. J.: Multi-Kernel Support Vector Clustering for Multi-Class Classification. International Journal of Innovative Computing Information and Control, Vol. 6, 2010, No. 5, pp. 2245–2262.

[14] Yang, X.-W.—Zhang, C. Q.—Lu, J.—Ma, J.: A Kernel Fuzzy c-Means Clustering-Based Fuzzy Support Vector Machine Algorithm for Classification Problems With Outliers or Noises. IEEE Transactions on Fuzzy Systems, Vol. 19, 2011, No. 1, pp. 105–115.

[15] Lee, C.-H.—Yang, H.-C.: Construction of Supervised and Unsupervised Learning Systems for Multilingual Text Categorization. Expert Systems with Applications, Vol. 2, Part 1, 2009, No. 36, pp. 2400–2410.

[16] Chicco, G.—Ilie, I.-S.: Support Vector Clustering of Electrical Load Pattern Data. IEEE Trans. Power Systems, Vol. 24, 2009, No. 3, pp. 1619–1628.

[17] Lee, S. H.—Daniels, H. M.: Gaussian Kernel Width Generator for Support Vector Clustering. In Advances in Bioinformatics and Its Applications, November 2004, pp. 151–162.

[18] Nath, J. S.—Shevade, S. K.: An Efficient Clustering Scheme Using Support Vector Methods. Pattern Recognition, Vol. 39, 2006, No. 8, pp. 1473–1480.

[19] Wang, J.-S.—Chiang, J-C.: An Efficient Data Preprocessing Procedure for Support Vector Clustering. Journal of Universal Computer Science, Vol. 15, 2009, No. 4, pp. 705–721.

[20] Jarvis, R. A.—Patrick, E. A.: Clustering Using a Similarity Measure Based on Shared Nearest Neighbors. IEEE Trans. Computers, Vol. 11, 1973, No. C-22, pp. 1025–1034.

[21] Yang, J.—Estivill-Castro, V.—Chalup, S. K.: Support Vector Clustering Through Proximity Graph Modeling. Proc. Ninth Int'l Conf. Neural Information Processing (ICONIP '02), 2002, pp. 898–903.

[22] Hao, P.-Y.—Chiang, J.-H.—Tu, Y.-K.: Hierarchically SVM Classification Based on Support Vector Clustering Method and Its Application to Document Categorization. Expert Systems with Applications, Vol. 33, 2007, No. 3, pp. 627–635.

[23] Vapnik, V. N.: The Nature of Statistical Learning Theory. Springer-Verlag 1995.

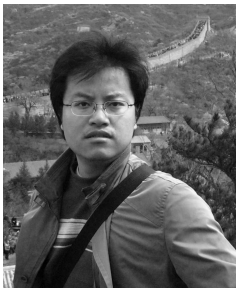
[24] Ertoz, L.—Steinbach, M.—Kumar, V.: Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data. Proc. of SIAM Int. Conf. on Data Mining, 2003, pp. 1–10.

[25] Karypis, G.—Han E.-H.—Kumar, V.: Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling. 1999.

[26] Asuncion, A.—Newman, D. J.: UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences. Available on: `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 2007.

[27] Zhu, X.-J.—Goldberg, A. B.: Introduction to Semi-Supervised Learning. Morgan and Claypool Publishers 2009.

[28] Jenssen, R.—Erdogmus, D.—Principe, J.—Eltoft, T.: The Laplacian PDF Distance: A Cost Function for Clustering in a Kernel Feature Space. In Advances in Neural Information Processing Systems 17, MIT Press 2005, pp. 625–632.



Yuan Ping received his M. Sc. degree in mathematics from the Henan University, Henan in 2008, and his B. Sc. degree in electronics and information engineering from Southwest University, Chongqing in 2003. He is currently a Ph. D. candidate at Beijing University of Posts and Telecommunications, Beijing, China. His current research interests include clustering, text categorization, security protocol and operating system security.

Yajian Zhou received his Ph. D. degree in communications engineering in 2003 from Xidian University at Xi'an, his M. Sc. degree in 1996 and his B. Sc. degree in 1993, both in the department of Material Science and Engineering, from Beijing University of Aeronautics and Astronautics at Beijing. He is currently a lecturer at School of Computer Science and Technology of Beijing University of Posts and Telecommunications. His main research interests include mobile communications, security of wireless networks, security of databases, cryptography theory and its application, etc. He is now the project leader of two projects from the National Natural Science Foundation of China and the National High Technology Research and Development Program of China, respectively. Meanwhile, he is in charge of a project from Beijing Municipal Natural Science Foundation. He has published about 20 technical papers and two books.

Yixian Yang received his Ph. D. degree in signal and information processing from Beijing University of Posts and Telecommunications in 1988. He is now the Director of National Engineering Laboratory for Disaster Backup and Recovery, the Director of Key Laboratory of network and information attack and defence technology of MOE, the Director of Information Security Center, and the Deputy Director of State Key Laboratory of Networking and Switching Technology. Meanwhile, he is a Changjiang Scholars Distinguished Professor and the Editor-in-Chief of Journal on Communications and The Journal of China University of Posts and Telecommunications. His current research interests include network and information security theory and applications, network-based computer application technology, coding theory and technology.