A Support Vector and K-Means Based Hybrid Intelligent Data Clustering Algorithm

Liang SUN†, Nonmember, Shinichi YOSHIDA††, Member, and Yanchun LIANG†††a, Nonmember

SUMMARY Support vector clustering (SVC), a recently developed unsupervised learning algorithm, has been successfully applied to solving many real-life data clustering problems. However, its effectiveness and advantages deteriorate when it is applied to solving complex real-world problems, e.g., those with large proportion of noise data points and with connecting clusters. This paper proposes a support vector and K-Means based hybrid algorithm to improve the performance of SVC. A new SVC training method is developed based on analysis of a Gaussian kernel radius function. An empirical study is conducted to guide better selection of the standard deviation of the Gaussian kernel. In the proposed algorithm, firstly, the outliers which increase problem complexity are identified and removed by training a global SVC. The refined data set is then clustered by a kernel-based K-Means algorithm. Finally, several local SVCs are trained for the clusters and then each removed data point is labeled according to the distance from it to the local SVCs. Since it exploits the advantages of both SVC and K-Means, the proposed algorithm is capable of clustering compact and arbitrary organized data sets and of increasing robustness to outliers and connecting clusters. Experiments are conducted on 2-D data sets generated by mixture models and benchmark data sets taken from the UCI machine learning repository. The cluster error rate is lower than 3.0% for all the selected data sets. The results demonstrate that the proposed algorithm compared favorably with existing SVC algorithms.

key words: data clustering, kernel methods, support vector clustering, K-Means clustering

1. Introduction

Data clustering is a problem of grouping a set of data into clusters so that the data in the same cluster are analogous in properties. It can be found in many fields, such as engineering, computer sciences, life and medical sciences, earth sciences, social sciences, and economics [1]. The support vector clustering (SVC) algorithm, proposed by Ben-Hur et al. [2]–[4], is a recently developed unsupervised learning method inspired by the support vector machine (SVM) [5]. In SVC, data points are mapped from the original space to a higher dimensional feature space by means of a Gaussian kernel. In the feature space, the algorithm seeks the smallest sphere that encloses the images of the data points. When the sphere is mapped back to the original space, it is separated into several contours with each enclosing a cluster of data points.

As a kernel-based algorithm, the SVC has many advantages. It can determine the system’s topological structure without prior knowledge with respect to the system itself, delineate cluster boundaries of arbitrary shapes other than hyper-ellipsoid and hyper-sphere, and deal with outliers by employing a soft margin constant which does not require that the sphere enclose all the data points. Due to these advantages, the SVC has attracted a high level of interest, and many variants have been derived to improve its performance. In greater detail, the method presented in [6] extended the SVC to an adaptive cell growing model, which maps data points to a higher dimension feature space through a desired kernel function. In [7], a spatial chunking algorithm is proposed to speed up the SVC algorithm for large scale data sets. In [8], a cluster labeling method for SVC is developed based on some of the invariant topological properties of a trained kernel radius function. In [9], the authors present a kernel method for clustering inspired by the classical K-Means algorithm in which each cluster is iteratively refined using a one-class support vector machine. In [10], a topological and dynamical characterization of cluster structures described by the support vector clustering is developed. In [11], the authors developed a cluster validity measure with outlier detection for the SVC algorithm.

A review of the above literatures suggests that research on the SVC algorithms has reached a good level of maturation. However, as far as SVC design, there are still some problems requiring further consideration and investigation. We summarize the main problems as follows.

1. The most widely used method for the training of SVC is the sequential minimal optimization (SMO) [12]. Benchmarks reported in [12] show that the time complexity of SMO is O(N^2). This implies that the problem of computational complexity may become intensive as the number of data points increases under real-world problems.

2. The standard deviation of the Gaussian kernel \( \sigma \) plays a crucial role in the clustering results. It controls the shapes of the enclosing contours in the data space. How to configure it suitably so that the SVC is capable of
obtaining a desired cluster result for a given data set is an open problem for SVC designers.

3. The algorithm is sensitive to outliers although a soft margin constant C is employed. If an outlier lies close to the cluster boundary, it will distort the cluster shapes, and make the algorithm fail to obtain the desired results.

4. Since the algorithm is performed by finding connected regions of the data distribution, its performance may deteriorate when the data set has connecting clusters.

In view of the above, this study is conducted with three integral parts: a SVC training method, an empirical study, and a support vector and K-Means based hybrid algorithm. The three integral parts provide solutions to the aforementioned problems. The SVC training method is proposed based on analysis of the Gaussian kernel radius function. The empirical study is conducted by formulating the SVC training procedure as building up weighted kernel density estimator for underlying distribution of the given data set. The proposed hybrid algorithm works by redefining the SVC as “one cluster as one sphere” in the feature space. It is achieved in three steps. In the first step, the outliers are identified and removed by training a global SVC. In the second step, the refined data set is clustered by a kernel-based K-Means algorithm. In the final step, several local SVCs are trained for the clusters, and then the removed data points are labeled according to their distance to the local SVCs. Since the proposed algorithm can integrate the advantages of the conventional SVC and the K-Means, it may overcome the difficulties of conventional SVC for its ability to deal data set with noise and connecting clusters.

The remainder of this paper is organized as follows. Section 2 reviews the basic concepts of the SVC and K-Means algorithm. Section 3 presents the new SVC training method. Section 4 conducts empirical study to guide better selection of the standard deviation of the Gaussian kernel. In Sect. 5, the details of the proposed hybrid algorithm are presented. Section 6 presents the experimental settings and provides simulated results for illustration and comparison, followed by concluding remarks in Sect. 7. 

2. Review of Support Vector Clustering and K-Means Algorithm

2.1 Support Vector Clustering Algorithm

The mathematical formulation of the SVC algorithm is summarized as follows [4]. Let \( \chi \subseteq \mathbb{R}^d \) be a d dimensional data space, \( \{x_1, x_2, \cdots, x_N\} \subseteq \chi \) be a data set. The algorithm uses a nonlinear transformation \( \Phi \) to map the data points from \( \chi \) to a higher dimensional feature space, and then seeks the smallest enclosing sphere in the feature space. The optimization problem can be formulated as follows:

\[
\text{minimize} \quad R^2 + C \sum_{j=1}^{N} \xi_j
\]

subject to \( ||\Phi(x_j) - a||^2 \leq R^2 + \xi_j, \quad j = 1, \cdots, N \) (1)

where \( R \) is the radius of the enclosing sphere, \( \xi_1, \xi_2, \cdots, \xi_N \) are slack variables, \( C \) is the soft-margin constant, ||·|| is the Euclidean norm and \( a \) is the sphere center. The problem can be solved by introducing the Lagrangian function:

\[
L = R^2 - \sum_{j=1}^{N} (R^2 + \xi_j - ||\Phi(x_j) - a||^2)\beta_j + C \sum_{j=1}^{N} \xi_j - \sum_{j=1}^{N} \xi_j \mu_j,
\]

(2)

where \( \beta_j \geq 0 \) and \( \mu_j \geq 0 \) are Lagrange multipliers. The solution to the primal problem described in Eq. (2) can be obtained by solving the dual problem [13]:

\[
\text{maximize} \quad W = \sum_{j} \Phi(x_j)^T \beta_j - \sum_{i,j} \beta_i \beta_j \Phi(x_i) \cdot \Phi(x_j)
\]

subject to \( 0 \leq \beta_j \leq C, \sum_{j} \beta_j = 1, \quad j = 1, \cdots, N \) (3)

The inner dot products \( \Phi(x_i) \cdot \Phi(x_j) \) can be replaced by a Mercer Kernel \( K(x_i, x_j) \). In this paper, the Gaussian kernel is considered:

\[
K(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{\sigma}}
\]

(4)

where \( \sigma \) is the standard deviation of the Gaussian kernel. After optimization, the distance from a given data point \( x \) to the sphere center can be computed by:

\[
f(x) = R^2(x) = ||\Phi(x) - a||^2 = K(x, x) - 2 \sum_{j} \beta_j K(x, x_j) + \sum_{i,j} \beta_i \beta_j K(x_i, x_j).
\]

(5)

Data points can be identified based on the \( \beta \) values. If \( \beta_j = C \), \( x_j \) is identified as a bounded support vector (BSV), else if \( 0 < \beta_j < C \), \( x_j \) is identified as a support vector (SV), else, \( x_j \) is identified as an inner point. When mapped back to the original data space, the BSVs lie outside of the cluster boundaries, SVs lie on the cluster boundaries, and the inner points lie inside of the cluster boundaries.

The above procedure delineates the contours of the data set. The next problem is the cluster assignment of each data point. The most widely used method for cluster labeling is the complete graph based method, which checks the connectivity for each pair of data points. Other methods include proximity graph based method [14], trained kernel radius function topology based method [8], [10], etc.

2.2 K-Means Algorithm

The K-Means algorithm is a squared error based clustering algorithm [15]. Its key steps can be summarized as follows:

1. Initialize a K-partition randomly or based on some prior knowledge.
2. Based on current partition, calculate the cluster prototype matrix \( M = [m_1, m_2, \cdots, m_K] \)

3. Assign each data point \( x_j \) (\( j = 1, \cdots, N \)) to the cluster \( c_w \) with the closest cluster prototype, i.e., \( x_j \in c_w \), if \( \|x_j - m_w\| < \|x_j - m_i\|, \forall i = 1, \cdots, K (i \neq w) \).
4. Repeat Steps 2-3 until the cluster prototype matrix becomes stable.

The \( K \)-Means algorithm is by far the most widely used data clustering algorithm. It is simple in concept, easy in implementation, and has good performance on data sets with compact super sphere distributions.

3. A Novel SVC Training Method

The optimization problem described in Eq. (3) is a quadratic programming (QP) problem which associates with finding the optimal values of Lagrange multipliers \( \beta_1, \beta_2, \cdots, \beta_N \). The sequential minimal optimization (SMO) algorithm is by far the most widely used method to solve this problem. The benchmarks reported in [12] show that the time complexity of SMO is \( O(N^2) \). This implies that its time complexity becomes intensive as the number of data points increases. In this paper, a new SVC training method is proposed. The equality constraint \( \sum \beta_i = 1 \) is eliminated by introducing variables \( a_1, a_2, \cdots, a_N \). Let \( \beta_i = a_i / \sum_k a_k \), then the QP problem described in Eq. (3) can be written as:

\[
\begin{align*}
\text{maximize} & \quad W = \sum_j K(x_j, x_j) \frac{a_j}{\sum_k a_k} \\
\text{subject to} & \quad 0 \leq \frac{a_j}{\sum_k a_k} \leq C, \ j = 1, \cdots, N.
\end{align*}
\] (6)

Equation (6) describes a QP problem which associates with finding the optimal values of variables \( a_1, a_2, \cdots, a_N \). A brief outline of the proposed training method is stated as follows:

1. Set \( a_1, a_2, \cdots, a_N \) values randomly. Initially, each of the variable \( a_i \) satisfies \( 0 \leq a_i / \sum_k a_k \leq C \).
2. Select a variable \( a_i \) randomly.
3. Calculate \( u_i = \sum_{j=1}^N a_j K(x_i, x_j) \), \( v_i = \sum_{j=1}^N a_j K(x_i, x_k) \), \( a_{\text{max}} = \max a_j \), \( \text{cons}_i = \sum_{j=1, j \neq i}^N a_j K(x_i, x_j) \),
4. Calculate \( L = \max(0, -v_i + a_{\text{max}}/C) \), \( H = C v_i / (1 - C) \).
5. Calculate \( a_i' = (u_i - \text{cons}_i) / (u_i - v_i) \).
6. Set the optimized value of variable \( a_i^{\text{new}} \), according to:

\[
\begin{cases}
L, & \text{if } a_i' \leq L, \\
\text{a}_i', & \text{if } L < a_i' < H, \\
H, & \text{if } a_i' \geq H.
\end{cases}
\] (7)

7. Repeat Steps 2-6 until \( a_1, a_2, \cdots, a_N \) turn stable.
8. For each \( \beta_i \), set \( \beta_i = a_i / \sum_j a_j \), return \( \beta_1, \beta_2, \cdots, \beta_N \) as final optimization results.

In the following, we describe the derivation of the above optimization method. Without loss of generality, let the variable to be optimized at each iteration be \( a_i \). Since \( K(x_j, x_j) = e^{-\|x_j - x_i\|^2/(2\sigma^2)} = e^0 \), then \( K(x_j, x_j) = 1 \) is satisfied. Let \( u_i = \sum_{j=1, j \neq i}^N a_j K(x_i, x_j) \), \( v_i = \sum_{j=1, j \neq i}^N a_j \), \( a_{\text{max}} = \max a_j (j \neq i) \), \( \text{cons}_i = \sum_{k=1, k \neq i}^N a_j K(x_j, x_k) \), then Eq. (6) can be written as:

\[
\begin{align*}
\text{maximize } & \quad W = 1 - \frac{a_i^2 + 2a_i u_i + \text{cons}_i}{(a_i + v_i)^2}, \\
\text{subject to} & \quad 0 \leq \frac{a_i}{a_i + v_i} \leq C, \ 0 \leq \frac{a_{\text{max}}}{a_i + v_i} \leq C.
\end{align*}
\] (8)

The extremum of the objection function \( W \) is at:

\[
\frac{\partial W}{\partial a_i} = \frac{2a_i (u_i - v_i) - 2(u_i - \text{cons}_i)}{(a_i + v_i)^3} = 0.
\] (9)

Following Eq. (9), we have:

\[
a_i = \frac{u_i v_i - \text{cons}_i}{u_i - v_i}.
\] (10)

Following the inequality constraint \( 0 \leq \beta_j \leq C \), we have:

\[
0 \leq \frac{a_i}{a_i + v_i} \leq C, \text{ and } 0 \leq \frac{a_j}{a_i + v_i} \leq C \ (j \neq i).
\] (11)

Following Eq. (11), we have:

\[
\frac{a_{\text{max}} - C v_i}{C} \leq a_i \leq \frac{C v_i}{1 - C}.
\] (12)

Considering both Eq. (10) and Eq. (12), we can calculate the new value of variable \( a_i \) according to Eq. (7).

**Remarks.**

1. Both the proposed algorithm and the SMO decompose the QP problem into subproblems. The theorem reported in [16] indicates that the QP problem can be broken into a series of subproblems. Thus, both algorithms can guarantee to converge to the global optimum of the QP problem.
2. Both the proposed algorithm and the SMO can optimize the variables analytically, instead of using numerical QP optimization steps.
3. The SMO performs optimization by repeatedly executing two steps: 1) select two variables by heuristic; 2) optimize the two variables analytically, instead of using numerical QP optimization steps.

4. Empirical Study

4.1 SVC Training and Weighted Kernel Density Estimator

Let \( x_1, x_2, \cdots, x_N \) be a data set taken from an univariate distribution with unknown density \( p \), the weighted kernel density estimator [17] approximates \( p \) by function \( \hat{p} \):
Let

\[ \hat{\beta}(x) = \sum_{j=1}^{N} w_j \varphi(x, x_j), \quad (13) \]

where \( \varphi(x, x_j) \) is the window function, \( 0 \leq w_j \leq 1 \) is the weight of the \( j \)-th kernel, and \( w_1, w_2, \ldots, w_N \) satisfy \( \sum w_j = 1 \).

Without loss of generality, we consider the Gaussian kernel window function:

\[ \varphi(x_i, x_j) = \frac{1}{\sqrt{2\pi\sigma^2}} K(x_i, x_j). \quad (14) \]

The trained kernel support function of SVC described in Eq. (5) can be defined by the squared radial distance from a data point \( x \) to the sphere center. Given a data point \( x \), if \( x \) is a BSV, then:

\[ f(x) = K(x, x) - 2 \sum_j \beta_j K(x, x_j) + \sum_{i,j} \beta_i \beta_j K(x_i, x_j) > R^2. \quad (15) \]

Since \( K(x, x) = e^{\frac{-|x-x|^2}{2\sigma^2}} = e^0 = 1 \) is satisfied, then:

\[ \sum_j \beta_j K(x, x_j) < \frac{1}{2} \left( 1 + \sum_{i,j} \beta_i \beta_j K(x_i, x_j) - R^2 \right). \quad (16) \]

Similarly, if \( x \) lies on the boundary of the sphere, then:

\[ \sum_j \beta_j K(x, x_j) = \frac{1}{2} \left( 1 + \sum_{i,j} \beta_i \beta_j K(x_i, x_j) - R^2 \right), \quad (17) \]

and if \( x \) lies inside of the sphere, then:

\[ \sum_j \beta_j K(x, x_j) > \frac{1}{2} \left( 1 + \sum_{i,j} \beta_i \beta_j K(x_i, x_j) - R^2 \right). \quad (18) \]

Combining Eqs. (13) and (14) with the term \( \sum_j \beta_j K(x, x_j) \), we have:

\[ \sum_j \beta_j K(x, x_j) = \sqrt{2\pi\sigma^2} \cdot \hat{\beta}(x). \quad (19) \]

Let \( 1/2(1 + \sum_{i,j} \beta_i \beta_j K(x_i, x_j) - R^2) = \sqrt{2\pi\sigma^2} \cdot d \), we have the following relations:

1. If \( x \) is a BSV, then \( \hat{\beta}(x) < d \).
2. If \( x \) is a SV, then \( \hat{\beta}(x) = d \).
3. If \( x \) is an inner point, then \( \hat{\beta}(x) > d \).

The above derivation can be explained intuitively as follows. The trained kernel support function of SVC describes a weighted kernel density estimator for the underlying data set. The objective of the SVC training procedure is to find the optimal weights \( \beta_1, \beta_2, \ldots, \beta_N \) such that the lower bound of the density function values of the given data points is maximized.

4.2 The Selection of Parameter \( \sigma \)

The derivations in Sect. 4.1 indicate that the SVC training can be formulated as the procedure of building up weighted kernel density estimator for underlying distribution of the given data set. Figure 1 shows an illustrative example of the effect of parameter \( \sigma \) on the density estimator. Where Fig. 1 (a) is the original data set, Fig. 1 (b) is the estimator with \( \sigma = 0.2 \), and Fig. 1 (c) is the estimator with \( \sigma = 0.1 \). It can be seen from Fig. 1 that the parameter \( \sigma \) has effect on the resulting estimator \( \hat{\beta}(x) \). Moreover, the SVC also specifies the contours that enclose the peaks of the probability distribution. As \( \sigma \) decreases, the estimator processes more peaks, and the SVC delineates more clusters. Thus the selecting of parameter \( \sigma \) is equivalent to selecting a proper bandwidth parameter for the density estimator. The density estimator simulated the true density so that the mean square error (MSE) is minimized. Let the true density be \( p(x) \) and the estimated density be \( \hat{\beta}(x) \), then the MSE is defined by:

\[ MSE(\hat{\beta}(x)) = E[(\hat{\beta}(x) - p(x))^2] \]

\[ = (E[\hat{\beta}(x) - p(x)])^2 + Var(\hat{\beta}(x)), \quad (20) \]

where \( E[\cdot] \) is the mathematical expectation, \( E[\hat{\beta}(x) - p(x)] \) represents the squared bias of the estimator, and \( \hat{\beta}(x) \) represents the variance of the estimator. Generally, a large bandwidth will reduce the variance and increase the bias. On the other hand, a small bandwidth will reduce the bias and increase the variance. In [19], the author proposed to...
compute the bandwidth of the Gaussian kernel according to some practical heuristic. That is:

$$\sigma = \left( \frac{4\hat{\sigma}^4}{3N} \right)^{-\frac{1}{4}} \approx 1.06\hat{\sigma}N^{-\frac{1}{5}},$$

(21)

where $N$ is the number of data points, and $\hat{\sigma}$ is the standard deviation of the samples, which is defined by:

$$\hat{\sigma} = \sqrt{\frac{1}{N} \sum_{j=1}^{N} ||x_j||^2 - \left( \frac{1}{N} \sum_{j=1}^{N} x_j \right)^2}.$$  

(22)

In view of this, we select the parameter $\sigma$ according to Eq. (21) as a compromise between the bias and the variance.

5. The Proposed Hybrid Intelligent Algorithm

The SVC algorithm as reviewed in Sect. 2 has many advantages over other algorithms for its ability to delineate cluster boundaries of arbitrary shape and to deal with outliers by the soft margin constant. However, there are two major difficulties encountered when it is applied to solving real-life data. Firstly, the algorithm is sensitive to outliers although the soft margin constant is employed. The outliers will distort the cluster boundaries, and make the algorithm fail to obtain the desired clustering results. Figure 2 shows an illustrative example of this problem. In Fig. 2 (a), no outliers exist in the data set. It can be seen that the SVC can delineate the cluster boundaries correctly. However, in Fig. 2 (b), when the data set contains outliers, the cluster boundaries are distorted. This makes the SVC algorithm delineate undesired cluster boundaries. Secondly, the SVC algorithm is performed by identifying clusters by finding connected components among data points, thus it may have difficulty in dealing with data sets with connecting clusters. Figure 3 shows an illustrative example of this problem. The simulation is conducted on the last two dimensions of IRIS data set. Figure 3 (a) shows the original data set. As can be seen from Fig. 3 (a), cluster 1 is linearly separable from clusters 2 and 3, and clusters 2 and 3 are connected. In the simulation, we firstly remove the outliers by the data set refining method, which will be described in the rest of this section. Following this, we run the SVC algorithm on the refined data set. Figure 3 (b) shows the clustering results. As can be seen from Fig. 3 (b), the SVC algorithm can not distinguish clusters 2 and 3.

To overcome the above difficulties, we propose a support vector and K-Means based hybrid intelligent algorithm (HIA). The key steps of the HIA are as follows:

1. Data set refining: identify and remove the outliers by building a global SVC.
3. Local SVC modeling: build local SVCs for each of the clusters.
4. Relabeling: label each removed data point according to the distance from it to the local SVCs.

![Fig. 2](image)

(a) SVC clustering on data set without outliers  
(b) SVC clustering on data set with outliers

Illustrative example for the effect of outliers on the cluster boundaries.

![Fig. 3](image)

(a) Original data set  
(b) Results of the standard SVC algorithm  
(c) Results of the kernel-based K-Means algorithm

Results of the kernel-based K-Means algorithm and the standard SVC algorithm on data set with connecting clusters.
We now describe and explain the detailed procedure for the above steps.

1. Data set refining: the main objective of this step is to identify and remove the outliers. A global SVC is trained to perform this task. In the global SVC, the BSVs are considered as outliers. The constant $C$ plays a crucial role in the identifying results. It can be seen from Eq. (3) that $0 \leq \beta_j \leq C$ and $\sum_j \beta_j = 1$. The lower bound of $C$ is $1/N$, since the constraint $\sum_j \beta_j = 1$ cannot be satisfied if $C < 1/N$. The upper bound of $C$ is 1, since $\sum_j \beta_j = 1$, the constraint $\beta_j \leq C$ has no effect if $C > 1$. We propose to select the constant $C$ value by training SVC iteratively: starting with $C = 1/N$, and increasing it by $\delta$, to count the number of outliers. The final $C$ value is selected when the number of outliers exceeds a pre-specified threshold $n$.

2. Clustering: the SVC algorithm may have difficulty clustering data sets with connecting clusters. One natural way to solve this problem is to use the K-Means algorithm, since the K-Means algorithm can work well for compact clusters. However, the K-Means algorithms that use geometric representation are often limited to hyper-ellipsoids. In view of the above, we propose a novel kernel-based K-Means algorithm. The proposed algorithm works as follows. Firstly, the algorithm maps the data points from the original space to a higher dimensional feature space by a nonlinear transformation $\Phi$. Secondly, in the feature space, the K-Means algorithm is used to cluster the mapped data points. The key steps of the algorithm are as follows:

1. Initialize a K-partition randomly.
2. For each partition, train a one class SVC, i.e., enlarge the SVC by redefining “one cluster as one sphere” in the feature space.
3. Assign each data point $x_j$ ($j = 1, \ldots, N$) to the sphere with the closest sphere center. The distance from the data point to the sphere center is computed by Eq. (5).
4. Repeat Steps 2–3 until the partition becomes stable.

In order to verify the validity of the clustering results, we use separation as the criteria for clustering evaluation. In our study, we run the kernel-based K-Means for a predefined number of times, and the cluster results that yield the biggest separability function value are adopted.

The kernel-based K-Means is performed by finding the prototypes (centers) of the clusters, while the standard SVC is performed by finding connected components among data points. Thus the kernel-based K-Means may have the advantage over the standard SVC for its ability to deal with connecting clusters. As a comparison, Fig. 4(c) shows the clustering results of the kernel-based K-Means algorithm on the refined IRIS data set. As can be seen from Fig. 4(c), the kernel-based algorithm can distinguish clusters 2 and 3, even though they are connected.

3. Local SVC modeling: in this step, we build local SVCs for the clusters obtained by the K-Means. Each of the cluster is described by a one class local SVC. In the local SVC, the parameter $\sigma$ is determined by the method presented in Sect. 4.2.

4. Relabeling: The main objective of this step is to label the removed data points. Let set $S_{vi}$ contains the support vectors of cluster $c_i$. Given a removed data point $x_i$, we define the distance from it to cluster $c_i$ as follows:

$$d(x_i, c_i) = \min_{x_j \in S_{vi}} ||x_i - x_j||^2.$$  

The data point $x_i$ is then assigned to the cluster with the minimum distance.

6. Experimental Studies

6.1 Experimental Setup

We have conducted three sets of experiments to test the performance of the proposed algorithm. The experiments are performed on a PC with Pentium IV 3.0 GHz Processor and 4 GB memory. The first set of experiments is conducted to test the time complexity of the proposed SVC training method. In the experiments, 100 data sets are simulated. They are simulated as follows: the dimension of the data
space is set to 4. In the data space, the data set is generated based on mixture models with size ranging from 50 to 1000. For the experiments, the parameter $\sigma$ is set by the method described in Sect. 4.2, and the soft margin constant $C$ is set to 1. The CPU time is used to evaluate the performance of the method. The SMO algorithm is used for comparison.

The second set of experiments is conducted to illustrate the execution process of the proposed HIA. In the experiments, two 2D data sets, namely 2D-N160 and 2D-N450, are generated. The 2D-160 is generated based on the same mixture model as used in [4], and 2D-450 is generated based on the same mixture model as used in [9]. In order to add complexity to the problem, noise data points are also included in the data set. Parameter settings are crucial for the performance of the HIA. Throughout the experiments, the parameter $\sigma$ in the kernel function is selected by the method described in Sect. 4.2. In the data set refining step, the soft margin constant $C$ is selected by training SVC repeatedly, and the parameters $\delta$ and $n$ are taken as follows: $\delta = 1/N$, $n = 0.1N$, where $N$ is the total number of data points in the data set. In the clustering step, we run the kernel-based K-Means algorithm for five times, and the cluster results that yield the biggest separability function value are adopted. In the local SVC modeling step, since the outliers are removed in Step 1, the constant $C$ is set to 1. In our experiments, we set the parameters as above and find that they are proper for all the cases.

The third set of experiments is conducted to test the performance of the HIA on benchmarks. In the experiments, three data sets taken from the UCI machine learning repository are selected. The three data sets include: the IRIS data set, the Wisconsin’s breast cancer data set, and the Spam data set. Their key properties are as follows.

1. The IRIS data set: The IRIS data set contains three classes, where each class contains 50 instances and refers to one type of IRIS plant. Each instance is composed of 4 measurements of an IRIS flower. One class is linearly separable from the other two. The remaining two classes have significant overlap and are not linearly separable from each other.

2. The Wisconsin’s breast cancer data set: the Wisconsin’s breast cancer data set contains 699 cases of diagnostic samples. After the removal of the 16 samples with missing values, the data set consists of 683 diagnostic samples. The diagnostic samples are partitioned into two classes, benign and malignant tumors. Each sample is composed of 9 measurements of the clinical attributes. The benign tumors take about 65.5% of the data set, while the malignant tumors take about 34.5% of the data set.

3. The Spam data set: The Spam data set is a 57 dimension data set formed by 4601 instances. Each instance represents a spam email or a non-spam email. The spam emails came from a postmaster and individuals who had filed spam, and the non-spam ones came from filed work and personal emails.

The parameter settings are the same as those in the second set of experiments. The CPU time and the cluster labeling error rate are used to evaluate the performance of the HIA. Here, the cluster labeling error rate is the percentage of the miss-labeled data points with respect to the total data set. For the purpose of comparison, we select the following algorithms which have been applied to all or some of the selected benchmarks: 1) the support vector clustering algorithm (SVC) [4], the kernel method (Kernel) as described in [9], the SVC described topological and dynamical characterization of cluster structures (TDSVC) [10], and the SVC with cluster validity (CVSVC) [11]. All of the above algorithms are kernel methods. They are able to separate a set of complex and nonlinear data points by transforming them to a higher dimensional feature space. These algorithms have been shown to be successful and can find good enough solutions. Hence, a comparison with these algorithms will demonstrate the performance of the proposed HIA, and will show if it is better or worse than other algorithms. Since the selected benchmarks are labeled data, the conventional SVM is also included for comparison. In the experiment, the SVM tool available in [20] is used. For each data set, 40% percent of the data points are used for training, 20% percent of the data points are used for validating, i.e., selecting optimal values for the parameters of SVM, and 40% percent of the data points are used for testing.

6.2 Simulated Results

6.2.1 Results on Data Sets with Different Size

Figure 4 shows the CPU time comparison of the proposed SVC training method and the SMO, where the abscissa is the size of data set and the vertical axis is the SVC training time. It can be seen from Fig. 4 that the two algorithms perform almost the same when the size of data set is smaller than 200. However, when the data set is larger than 200, the proposed algorithm performs much better than the SMO. The results demonstrate that the proposed SVC training method is fast compared to the SMO.

6.2.2 Results on Generated 2D Data Sets Based on Mixture Models

1) Results on 2D-N160 data set: The 2D-N160 data set comprises 160 data points in a 2D space and consists of four clusters with noise. Figure 5 shows an illustrative example for the execution process of the HIA on the 2D-160 data set. In Fig. 5 (a), the outliers are identified by a global SVC. The data points marked with circle are considered as outliers, and the data points marked with plus are considered as support vectors or inner points. After the removal of the identified outliers, the kernel-based K-Means algorithm operates on the refined data set. Figure 5 (b) shows the kernel-based K-Means clustering results, the data points in the four clusters are marked with cross, star, circle, and plus, respectively. Following this, four local SVCs are built for each
of the clusters. Figure 5 (c) shows the cluster contours delineated by the local SVCs. In Fig. 5 (d), the removed data points are labeled according to the distance from them to the local SVCs, resulting in labeling the whole data set.

2) Results on 2D-450 data set: The 2D-450 data set comprises 450 data points in a 2D space and consists of two clusters with noise. The two clusters are not linearly separable from each other. The experiments conducted on 2D-160 are repeated on 2D-450. Figure 6 shows an illustrative example for the execution process of the HIA on the 2D-450.
The results on the 2D-160 data set and the 2D-450 data set demonstrate that the proposed HIA can perform well when the data set contains nonlinearly separable classes and are polluted by noise.

6.2.3 Results on Data Sets Taken from the UCI Machine Learning Repository

Table 1 summarizes the results of the experiments on data sets taken from the UCI machine learning repository. The contents of the table include the name of each data set (Data Set), the scale of the data set (Size), the dimension of the data set (Dim), the CPU time used for the proposed algorithm (Time), the cluster labeling error rate (Error%) of the HIA, and the results reported in literatures using other kernel-based algorithms [4], [9]–[11] and the results of SVM. For the algorithm compared, not all results of selected benchmarks are reported in literature, so for those benchmarks without reported results, their corresponding comparison values are marked with ‘-’. For the SVM, the CPU time includes the training time and the testing time, and the cluster labeling error rate is the percentage of the miss-labeled testing data points with respect to the total testing data points.

From the table, it can be seen that the proposed algorithm yields a significant improvement in terms of cluster labeling error rate with respect to the listed results, except for the results obtained by TDSVC algorithm. Compared with TDSVC, it can be seen that both algorithms can obtain the same results for the IRIS data set, and the TDSVC algorithm has a better performance for the Spam data set. Concerning about the time complexity, the TDSVC follows the key steps of the standard SVC, i.e., solving the quadratic programming of SVC, and then labeling each data point. Generally, cluster labeling task is more computationally intensive than the SVC training task, and may become highly intensive as the scale of the data set increase. On the other hand, in the proposed algorithm, the kernel-based K-Means algorithm yields labeled data points for most data points, and each removed data point is labeled according to the distance from it to the clusters. By using this strategy, the cluster labeling step as used in the standard SVC is unneeded. Thus, compared with TDSVC, the proposed algorithm has the potential to obtain a faster computing time with a moderate labeling error rate. The results shown in Table 1 validate the above analysis although it cannot provide a completely fair comparison since the CPU and memory specifications are different. In summary, the above results indicate the successful incorporation of the SVC algorithm and the K-Means.

7. Conclusion

The SVC algorithm has been successfully applied to solving many real-life data clustering problems. In this paper, we aimed at proposing a new kernel-based clustering algorithm to improve the performance of SVC. A SVC training method was proposed based on theoretical analysis of the Gaussian kernel radius function. An empirical study was conducted to guide better selection of the standard deviation of the Gaussian kernel. A new data clustering algorithm, i.e., the support vector and K-Means based hybrid intelligent algorithm, was developed by integrating the merits of both SVC and K-Means algorithm. The effectiveness of the proposed algorithm was validated by three sets of experiments on generated data sets with different sizes, generated 2D data sets, and data sets taken from the UCI machine learning repository. The results demonstrated that the proposed hybrid intelligent algorithm compared favorably with existing kernel-based clustering algorithms. In the future, we will extend this research to study the theoretical properties of the algorithm (in particular, convergence analysis of the hybrid clustering algorithm) and to apply the algorithm to the solving of real-life problems such as image processing and computer vision.

Acknowledgements

The third author is grateful to the support of the NSFC (10872077, 61073075) and the “863” Project (2009AA02Z307) of China.

References


**Liang Sun** received the B.E. degree in computer science and technology from Xidian University, Xi’an, China, and the M.S. degree in computer application technology from Jilin University, Changchun, China, in 2003 and 2006, respectively. During 2006–2009, as a D.E. candidate, he was at College of Computer Science and Technology, Jilin University China. He is now a D.E. candidate at Kochi University of Technology (KUT), Japan, as an international student of cooperation between KUT and Jilin University. His main research interests lie in computational intelligence and machine learning methods.

**Shinichi Yoshida** is an assistant professor in School of Information, Kochi University of Technology. He received his B.E. from Chuo University in 1996. He received Dr. Eng. from Tokyo Institute of Technology in 2001. From 2001 to 2004, he was an assistant professor of Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology. From 2004 to 2007, he was an assistant professor of Information Science Research Center, Aoyama Gakuin University. Since 2007, he has been an assistant professor of Department of Information Systems Engineering. Since 2009, he has been an assistant professor of School of Information, Kochi University of Technology. His interest covers multimedia information retrieval based on softcomputing and computational intelligence. He is a member of IEEE, IPSJ, and SOFT.

**Yanchun Liang** received the PhD degree in applied mathematics in 1997 from Jilin University, Changchun, China, where he is currently a professor in the College of Computer Science and Technology. He was a visiting scholar at Manchester University of United Kingdom from 1994 to 1997, a visiting professor at the National University of Singapore from 2000 to 2001, a guest professor at the Institute of High Performance Computing of Singapore from 2002 to 2004, and a guest professor at Trento University, Italy, from 2006 to 2008, and 2010. He has published more than 300 papers. His research was featured in the Bioinformatics, IEEE Trans. on SMC–Part A, IEEE Trans. on Knowledge and Data Engineering, IEEE Trans. on Geoscience and Remote Sensing, Journal of Micromechanics and Microengineering, Physical Review E, Smart Materials and Structures, Applied Artificial Intelligence, etc. He was the recipient of several grants from NSFC, EU, etc. His research interests include computational intelligence, machine learning methods, text mining, MEMS modeling, and bioinformatics.