A Kernel Prototype-based Clustering Algorithm

CHI-YUAN YEH¹, SHIE-JUE LEE¹, CHIH-HUNG WU², SHING-H. DOONG³
¹Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung 804, Taiwan
²Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan
³Department of Information Management, ShuTe University, YenChau 824, Taiwan

Abstract: One-class SVM is a kernel-based method which utilizes the kernel trick for data clustering. However, it is only able to detect one cluster of non-convex shape. In this study, we propose a strategy using one-class SVM to calculate the centroid of the sphere for each cluster in feature space. In addition, a mechanism is provided to control the position of the cluster centroid in feature space to work against outliers. We compare our method with other kernel prototype-based clustering algorithms, like KKM and KFCM, on two synthetic data sets and four UCI real data sets, the results indicate that our method outperforms KKM and KFCM.

Key-Words: One-class SVM, Kernel Prototype-based Clustering, Kernel K-means, Kernel Fuzzy C-means

1 Introduction
The prototype-based method is a well-known technique in unsupervised data clustering. However, they are not suitable to discover clusters with non-convex or overlapped shapes. Recently, a number of kernel-based learning methods have been proposed for data clustering. They utilize the kernel trick, doing all calculations in the low-dimensional input space, to perform inner products in the high-dimensional feature space where the data are expected to be more separable.

Schölkopf first integrated kernel-based methods with K-means [11]. The kernel K-means clustering algorithm (KKM) transforms implicitly the input data into a high-dimensional feature space via a nonlinear mapping function, and then performs K-means in the feature space. The results indicate that KKM outperforms conventional K-means due to the nonlinear mapping. Girolami developed an EM algorithm to solve the KKM problem, and recasted the objective function as trace maximization [7]. Dhillon et al. introduced a weighted version of the KKM objective function to improve the quality and speed of KKM [4][5]. Zhang and Rudnicky focused on a different issue; they proposed a large scale KKM clustering scheme to deal with large corpus [16].

Zhang and Chen integrated kernel-based methods with fuzzy C-means. The experimental results indicate that Kernel Fuzzy C-means (KFCM) has better performance in the ring dataset due to the RBF kernel function and is robust to noise and outliers due to the fuzzy membership degrees [13][14][15]. Mizutani and Miyamoto integrated possibilistic approaches with KFCM (KPCM), and used the entropy-based objective function instead of the standard objective function. The experimental results indicate that KPCM outperforms KFCM due to the entropy-based method available for both the probabilistic constraint and possibilistic constraint [9]. Kim et al. indicated that kernel-based methods were about 15% more accurate than conventional methods [8]. Du et al. indicated that kernel-based clustering algorithms can extract arbitrarily shaped clusters but are not robust to noise data, while nonlinear distance methods are robust to noise data but cannot extract arbitrarily shaped clusters [6].

Tax and Duin [12], Müller et al. [10], and Ben-Hur et al. [1] introduced a kernel-based method, called support vector data description (SVDD) or one-class Support Vector Machine (one-class SVM); it computes the smallest sphere in feature space enclosing the image of the input data. This method not only can extract arbitrarily shaped clusters, but also are robust against noises and outliers. However, one-class SVM is only able to detect one cluster. We are interested in applying one-class SVM to solve multiple clusters problem. In this study, we propose a K-means like strategy which uses one-class SVM to calculate the centroid of the sphere for each cluster in feature space in stead of the geometric mean of K-means, and compare the performance of this method with that of KKM and KFCM.

2 Background
2.1 One-Class SVM
One-class SVM is a kernel method based on a support vector description of a data set consisting of positive patterns only. If the data set does not contain outliers, one-class SVM computes the smallest
sphere in feature space enclosing the image of the input data. The objective is to minimize

$$R^2$$

s.t. $$\| \phi(x_i) - a \|^2 \leq R^2, \forall i$$

(1)

where $$\phi: X \rightarrow F$$ is a (possibly) nonlinear mapping from an input space to a feature space, $$a$$ is the center of the sphere in the feature space and $$R$$ is the radius.

To allow for the possibility of outliers in the data set, and to make the method more robust, distances form an image to the center need not be strictly smaller than $$R$$; but the excessive distances should be penalized. Therefore, we can introduce slack variables $$\xi_i, \xi_i \geq 0, \forall i$$, to account for the excessive distance and the new objective function becomes

$$R^2 + C \sum_{i=1}^{n} \xi_i$$

s.t. $$\| \phi(x_i) - a \|^2 \leq R^2 + \xi_i, \xi_i \geq 0, \forall i$$

(2)

The parameter $$C$$ gives a tradeoff between the radius of sphere and the excessive distances of outliers. With a large $$C$$, penalty on excessive distances becomes substantial, thus few outliers should exist and the sphere is big. On the other hand, when $$C$$ becomes small, many outliers with large excessive distances may exist to take advantages of the small penalty and the radius is generally small. In practice, the size of $$C$$ determines the number of outliers.

In order to solve the constrained optimization problem, a Lagrangian is introduced as follows:

$$L = R^2 + C \sum_{i=1}^{n} \xi_i$$

$$- \sum_{i=1}^{n} \alpha_i \left[ R^2 + \xi_i - \| \phi(x_i) - a \|^2 \right] - \sum_{i=1}^{n} \beta_i \xi_i$$

(3)

where $$\alpha_i \geq 0$$ and $$\beta_i \geq 0$$ are Lagrange multipliers. $$L$$ has to be minimized with respect to $$R$$, $$a$$ and $$\xi$$ given $$\alpha$$ and $$\beta$$, and then maximized with respect to $$\alpha$$ and $$\beta$$.

Setting partial derivatives equal to 0 can get the results as follows:

$$\frac{\partial L}{\partial R} = 2R - 2R \sum_{i=1}^{n} \alpha_i = 0$$

$$\sum_{i=1}^{n} \alpha_i = 1$$

(4)

$$\frac{\partial L}{\partial a} = -2 \sum_{i=1}^{n} \alpha_i \phi(x_i) + 2a \sum_{i=1}^{n} \alpha_i = 0$$

$$a = \sum_{i=1}^{n} \alpha_i \phi(x_i)$$

(5)

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0$$

(6)

The Karush-Kuhn-Tucker (KKT) complementary condition yields:

$$\alpha_i \left[ R^2 + \xi_i - \| \phi(x_i) - a \|^2 \right] = 0$$

(7)

$$\beta_i \xi_i = 0$$

(8)

For a data point $$x_i$$ with $$\xi_i > 0$$, we have $$\beta_i = 0$$ and $$\alpha_i = C$$. Equation (7) implies $$\| \phi(x_i) - a \|^2 > R^2$$. In other words, the image $$\phi(x_i)$$ lies outside the feature space sphere. This is called a bounded support vector (BSV), or is sometimes called an outlier. For a data point $$x_i$$ with $$\xi_i = 0$$, if its $$\beta_i \neq C$$ then $$0 < \alpha_i \leq C$$. Equation (7) implies $$\| \phi(x_i) - a \|^2 \leq R^2$$, so $$\phi(x_i)$$ lies on the surface of the feature space sphere. Such a point will be referred to as a support vector (SV). If $$\beta_i = C$$, then $$\alpha_i = 0$$. Equation (7) implies $$\| \phi(x_i) - a \|^2 < R^2$$, so $$\phi(x_i)$$ lies inside the sphere in the feature space. In addition, the constraint of Equation (4) implies when $$C \geq 1$$ no outliers exist.

Substituting Equations (4), (5) and (6) into Equation (3), we can eliminate the variables $$R$$, $$a$$, and $$\beta_i$$, turning the Lagrangian into the Wolfe dual form

$$W_D = \sum_{i=1}^{n} \alpha_i K(x_i, x_i) - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(x_i, x_j)$$

s.t. $$0 \leq \alpha_i \leq C$$

(9)

For each point $$x$$, the distance of its image in the feature space from the center of the sphere is given by

$$\| \phi(x) - a \|^2 = \phi(x) \cdot \phi(x) - 2 \phi(x) \cdot \phi(x) + \phi(x) \cdot \phi(x)$$

$$= K(x, x) - 2 \sum_{j=1}^{n} \alpha_j K(x, x_j) + \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(x_i, x_j)$$

(10)

Three commonly used kernel functions are Polynomial kernel, Sigmoid kernel, and RBF kernel. In this study, the RBF kernel is adopted. The RBF kernel is:

$$K(x_i, x_j) = \exp \left( -\frac{\| x_i - x_j \|^2}{\sigma^2} \right)$$

(11)

2.2 Related work

Since one-class SVM is only able to detect one cluster, mechanisms were introduced to solve multiple clusters problem. Chiang and Hao proposed a multiple-sphere support vector clustering algorithm
based on an online cluster cell growing method [3]. When a pattern belongs to one cluster and lies outside the sphere, it must retrain one-class SVM for this cluster. The disadvantage of this method is that it is very time-consuming. Camara and Verri proposed a clustering method inspired by the conventional K-means algorithm where each cluster is iteratively refined using a one-class SVM [2]. This method is also called Kernel Grower (KG) because a threshold \( \rho \) is introduced to control the number of cluster members that tend to grow after each iteration. Furthermore, they set the parameter \( C = 1 \), therefore no cluster can contain outliers (BSV). Because of this, the KG method lacks the ability to handle possible outliers. In this study, we use support vector data description (SVDD) [12] based on K-means to cluster data. We don’t use the threshold \( \rho \) but modulate the parameter \( C \) to provide a tradeoff between the radius of sphere and the excessive distances resulted from outliers. We set \( C \) initially small to discount the influence of the noises or outliers on the centroid of the sphere in the early stage of the K-means like iteration. When the cluster memberships become more stable in the later stage, we set \( C = 1 \) to disallow outliers for each cluster.

3 The Proposed Method

Since the value of parameter \( C \) provides a tradeoff between the radius of sphere and the excessive distances resulted from outliers, it affects the position of the centroid of the sphere. For example, consider the 48 patterns shown in Fig. 1. Various \( C \) can be tried according to the following formula:

\[
C = \frac{1}{\text{fraction} \times N} \tag{12}
\]

where \( N \) is the number of the patterns. With \( \text{fraction} = 0.9 \), we get a small \( C = 0.023 \) which allows for many outliers and SVs to exist as shown in Table 1. According to Equation (10), we can see that the distance between the image of a pattern from Group 1 (G1) and the centroid is generally smaller than the distance between the image of a pattern from Group 2 (G2) and the centroid. This implies that the position of the centroid of the sphere is close to G1. When \( \text{fraction} = 0.002 \), we get a larger \( C = 10.42 \) which allows a few SVs to exist as shown in Table 1. In this case, the position of the centroid of the sphere is located between G1 and G2.

When the cluster members are randomly drawn from the data set, we may have some patterns belonging to the wrong clusters initially. In this case, a small \( C \) will generally set a tighter control for the centroid of the sphere. If we set \( C = 1 \), the centroid of the sphere will lie between the two clusters, and the assignment of cluster members will be affected. In this study, we set a smaller \( C \) to control the position of the centroid of the sphere at the beginning. After four iterations, we set \( C = 1 \) since the cluster memberships tend to be more stable at that time. The proposed algorithm uses one-class SVM to compute the centroid of the sphere for each cluster instead of the means as used in KKM. Like prototype-based clustering algorithms, we assume that the number of clusters, \( k \), is known in advance. We set \( C < 1 \) to discount the influence of outliers on the centroid of the sphere in the early stage. Each pattern is assigned to the closest centroid by the following equation:

\[
c_i = \{ x_j \in X, \phi(x_j) \in F \mid i = \arg \min_{i=1\ldots k} \| \phi(x_j) - a_i \| \} \tag{13}
\]

where \( X = [x_1, \ldots, x_n]^T \in \mathbb{R}^{d \times n} \) is an unlabeled data set, \( x_j \) represents a \( d \)-dimensional pattern, \( j=1, \ldots, n \), and \( c_i \) are exclusive clusters, \( i=1, \ldots, k \). The proposed algorithm consists of the following steps:

![Fig.1 An example for the analysis of parameter C](image)

Table 1 Numbers of patterns lying inside, on the surface or outside the sphere for various values of \( C \)

<table>
<thead>
<tr>
<th>Fraction</th>
<th>0.9</th>
<th>0.5</th>
<th>0.2</th>
<th>0.1</th>
<th>0.021</th>
<th>0.002</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>0.023</td>
<td>0.042</td>
<td>0.104</td>
<td>0.208</td>
<td>1</td>
<td>10.42</td>
</tr>
<tr>
<td>inside</td>
<td>G1</td>
<td>4</td>
<td>22</td>
<td>38</td>
<td>40</td>
<td>43</td>
</tr>
<tr>
<td>G2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>surface</td>
<td>G1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>G2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>outside</td>
<td>G1</td>
<td>37</td>
<td>18</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>G2</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Step 1: Given $k$ and $\sigma$, initial cluster members are prepared using the input variable with the largest variance. The data set $X$ is divided into $k$ disjoint subsets $X_Di$ ($X_Di \subset X$) according to the selected input variable, then initial cluster members are drawn randomly from the partition ($c_i \subset X_Di$). Note that $c_1 \cup c_2 \cup \ldots \cup c_k = X$.

Step 2: Train a one-class SVM for each cluster $c_i$.

Step 3: Assign each pattern to its closest cluster according to Equation (13). Note that $c_1 \cup c_2 \cup \ldots \cup c_k = X$.

Step 4: If no cluster changes in Step 3, exit; otherwise, go to Step 2.

The whole process of the proposed algorithm is shown in Fig.2.

The Wisconsin’s breast cancer data set contains some missing values. We have removed 16 patterns with missing values from the data set; therefore the data set considered in the experiments has 683 patterns. The detailed information of the data sets is shown in Table 2. The standard deviation $\sigma$ for the RBF kernel with three kernel prototype-based clustering algorithms in six the data sets are shown in Table 3. The parameters used in KFCM were a termination criterion of $r = 0.001$ and a weighting exponent of $m = 2.0$. The parameters used in one-class SVM were $C < 1$ and $C = 1$. Each clustering method is applied to each data set for 20 times with different initial cluster members.

### Table 2 The properties of six data sets

<table>
<thead>
<tr>
<th>Name</th>
<th>Dim</th>
<th>Size</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ring</td>
<td>2</td>
<td>720</td>
<td>473</td>
<td>247</td>
<td>#</td>
</tr>
<tr>
<td>Delta</td>
<td>2</td>
<td>424</td>
<td>212</td>
<td>212</td>
<td>#</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>9</td>
<td>683</td>
<td>444</td>
<td>239</td>
<td>#</td>
</tr>
<tr>
<td>IRIS</td>
<td>4</td>
<td>150</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Wine</td>
<td>13</td>
<td>178</td>
<td>59</td>
<td>71</td>
<td>48</td>
</tr>
<tr>
<td>Thyroid</td>
<td>4</td>
<td>215</td>
<td>150</td>
<td>35</td>
<td>30</td>
</tr>
</tbody>
</table>

### Table 3 The standard deviation $\sigma$ of the RBF Kernel

<table>
<thead>
<tr>
<th>Name</th>
<th>KKM $\sigma$</th>
<th>KFCM $\sigma$</th>
<th>OUR METHOD $\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ring</td>
<td>0.14</td>
<td>0.14</td>
<td>0.14</td>
</tr>
<tr>
<td>Delta</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>10</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>IRIS</td>
<td>2</td>
<td>4</td>
<td>0.4</td>
</tr>
<tr>
<td>Wine</td>
<td>2</td>
<td>0.57</td>
<td>6</td>
</tr>
<tr>
<td>Thyroid</td>
<td>2</td>
<td>0.34</td>
<td>8</td>
</tr>
</tbody>
</table>

Fig.3 shows the result of applying three kernel prototype-based clustering algorithms on the 2-D Ring Data Set. These methods achieved 100.00% accuracy due to the RBF kernel function. Fig.4, Fig.5 and Fig.6 show the results of applying three kernel prototype-based clustering algorithms on the 2-D Delta Data Set, which also appeared in [2]. Similar to the conventional K-means method, KKM cannot separate this data set well. KFCM outperforms KKM; sometimes KFCM can achieve 100.00% accuracy, but the probability is very low (5%). Note that our method can obtain 100.00% accuracy in all 20 trials.
From Table 4, we can see that all three kernel prototype-based clustering algorithms have the same results in 20 trials on the Wisconsin’s breast cancer data set. However, our method (96.93%) slightly outperforms KKM (96.19%) and KFCM (94.14%). In the best case, our method misclassified 14 patterns in class 1 and 7 patterns in class 2.

The Fisher IRIS data is a well-known data collection that contains 150 patterns belonging to three different classes. One of the classes is linearly separable from the other two, which are not linearly separable. From Table 4, we can see that the non-linearly separable clusters can be better solved by kernel-based methods; conventional K-means just archived 89.0% of accuracy [2]. In addition, our method outperforms KFCM and KKM for this data set. In the best case, our method can totally identify class 1 and achieve 100.0% of accuracy for this class, misclassify 1 pattern in class 2, and misclassify 3 patterns in class 3. KKM is affected by initial centers of the clusters, and in the worst case it just obtains 58.0% of accuracy, though most trials can achieve 96.0% of accuracy. The results indicate that the performance of KKM clustering depends on the initial guess of the centers of the clusters.

From Table 4, we can also see that our method outperforms the other two methods for the Wine data set. KKM outperforms KFCM on the Wine recognition data set. In the best case, our method can separate class 1 and class 3 completely and misclassify 8 patterns in class 2. The best case of KFCM can achieve 96.63% of accuracy, but for most trials, the accuracies are between 65.17% and 80.0%. In addition, KKM and KFCM can identify class 1 completely, but can not separate class 2 and class 3 in most trials.

For the Thyroid Disease data set, our method outperforms the other two methods, and KKM outperforms KFCM. KKM and KFCM can separate class 1 completely, but cannot separate class 2 and class 3 in most trials. In the best case, our method can separate class 2 completely, misclassify 2 patterns in class 1 and 4 patterns in class 3.

Table 4 Accuracy for four real data sets from 20 trials

<table>
<thead>
<tr>
<th>Name</th>
<th>Wisconsin</th>
<th>IRIS</th>
<th>Wine</th>
<th>Thyroid</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>96.19</td>
<td>58.00</td>
<td>60.11</td>
<td>86.05</td>
</tr>
<tr>
<td>K</td>
<td>96.19</td>
<td>96.00</td>
<td>95.51</td>
<td>88.84</td>
</tr>
<tr>
<td>M</td>
<td>96.19</td>
<td>94.10</td>
<td>91.77</td>
<td>88.70</td>
</tr>
<tr>
<td>K</td>
<td>94.14</td>
<td>94.67</td>
<td>65.17</td>
<td>80.00</td>
</tr>
<tr>
<td>C</td>
<td>94.14</td>
<td>94.67</td>
<td>96.63</td>
<td>81.40</td>
</tr>
<tr>
<td>M</td>
<td>94.14</td>
<td>94.67</td>
<td>82.16</td>
<td>81.11</td>
</tr>
</tbody>
</table>
5 Conclusion

We have described an algorithm of using one-class SVM to detect multiple clusters of non-convex shape. Experimental results allow us to make the following remarks regarding the unsupervised clustering problem:

1. Kernel prototype-based clustering algorithms outperform conventional prototype-based clustering algorithms because they can better solve non-linearly separable clusters.
2. Our method outperforms other kernel prototype-based clustering algorithms in all cases. Our idea is to set a smaller C to control the position of the centroid of the sphere. This has resulted in a positive effect on the correct assignment of the cluster members.
3. Regarding the time performance, our method is slower than the other two methods because one-class SVM has to be solved frequently. KFCM can be substantially faster than the other two methods because it does not have to update cluster members in each iteration.
4. Regarding the initialization of the algorithms, KKM and KFCM strongly depend on the guess of the initial values. In this study, we use a heuristic method to solve this problem, and this issue is not so important for the proposed method.

References:


