The Sorting Methods of Support Vector Clustering Based on Boundary Extraction and Category Utility

Weigao Chen, Xin Jia, Weigang Zhu and Xiaojing Tang

Equipment Academy, Beijing, China

Abstract. According to the problems of low accuracy and high computational complexity in the classification of unknown radar signals, a method of unsupervised Support Vector Clustering (SVC) based on boundary extraction and Category Utility (CU) of unknown radar signals is studied. By analyzing the principle of SVC, only the boundary data of data sets contribute to the support vector extracted. Thus firstly, for reducing the data set, at the same time reducing the computational complexity, the algorithm is designed to extract the boundary data through local normal vector. Then using CU select the optimal parameters. At last distinguish different categories and get the sorting results by Cone Cluster Labelling (CCL) and Depth-First Search (DFS). Through comparing the simulation results, the proposed method which is based on boundary extraction and CU is proved to have turned out quite good time effectiveness, which not only improves the accuracy of classification, but also reduces the computational complexity greatly.

1 Introduction

With the wide application of electronic information technology in the military field, it is significant to improve the whole operational efficiency of the electronic warfare equipment, and also presents a great challenge to each core components of the electronic countermeasure. As radar countermeasure’s reconnaissance front, radar signal sorting bears the brunt of this challenge. The purpose of radar signal sorting is to extract the pulse sequence belonging to the same radar from the disorderly interleaved pulse stream. It is the premise of signal analysis and processing [1]. For the classification of unknown radar signals, prior knowledge is very scarce, so with the help of unsupervised clustering in data mining, the inner characteristic and distribution rule of data set can be obtained without prior knowledge. SVC algorithm uses the kernel function nonlinear mapping method to make the features which is not obvious in the data space enlarged, extracted and cleared in the high dimensional feature space to benefit the clustering. Nature of SVC is a convex quadratic optimization problem, the results have a strong global optimization and generalization ability, in addition, the algorithm can also adapt to the clusters of arbitrary boundary shape, so it is very suitable for unknown disorderly radar signal sorting. However, under the condition of high density and large data quantity, the calculation of optimization and cluster label process are huge, which greatly inhibits the performance of SVC. So in order to solve the problem, we propose a support vector clustering algorithm (BECU SVC) based on boundary extraction and CU. The algorithm firstly uses local normal vector to extract the boundary of de-noised data, which reduces the data set [2]; secondly, extracts the support vectors data; then under the guidance of CU [3], using improved CCL [4] to cluster label; at last accomplishes the category classification of surplus data by DFS.

2 Boundary Extraction Based on Local Normal Vector

Because of the influence of false alarm detection, clutter, jamming, etc. the radar signals transmitted to signal processor carry a lot of noise data, so the noise reduction processing is the precondition of radar signal sorting. Because SVC algorithm needs to select the appropriate data from the data set to construct the support vector function, and these ‘suitable data’ are only distributed in the contour region of the data set, and the remaining data inside the contour have no contribution to SVC, so these data are redundant. Therefore, in order to abandon redundant data and reduce the computation of SVC, it’s necessary to extract the de-noised data set. Consider a universal two-dimensional projection data distribution, which contains three typical boundary data points, as shown in Figure 1.

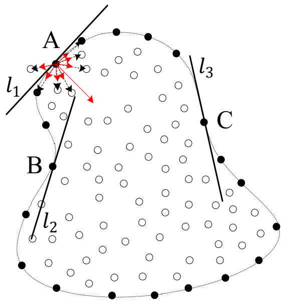

Figure 1. Two-dimensional projection of the data set

The solid dots in figure represent the contour data points, hollow dots are internal or external data points. A, B, C respectively represent three typical boundary distribution points, A point is located in the bulge of boundary, the hyper tangent plane l_1 divides internal data points into the same side; B, C points locate in the sunken boundary, the most points are divided into one side by the hyper tangent plane l_2, l_3, and the left points locate on the other side. In Figure 1, for an example of A point, the local normal vector and the closest 7 points (Euclidean Distance) of A are indicated. Observe all the vectors, we can find for the points of convex (including high-dimensional hyper convex) profile data set, if the angle $\theta < \frac{\pi}{2}$ between local normal vectors of these profile points and unit vectors of other points, these points belong to the same cluster.

In general, we use the k nearest neighbors (kNN) method to calculate the local normal vector of $\boldsymbol{x}_i\,(i = 1,2,\cdots,n)$. At first Euclidean Distance is used to find the nearest k points $\boldsymbol{x}_j$ ($j = 1,2,\cdots,k$) of $\boldsymbol{x}_i$ in the high dimensional data space, calculate the unit vector $\boldsymbol{v}_{ij}$

$$\boldsymbol{v}_{ij} = \frac{x_j - x_i}{|x_j - x_i|},\ j = 1,2,\cdots,k \qquad (1)$$

As shown in Fig.2, the approximate local normal vector can be expressed as sum of k unit vectors, i.e.:

$$\boldsymbol{n}_i = \sum_{j=1}^{k} \boldsymbol{v}_{ij} \qquad (2)$$

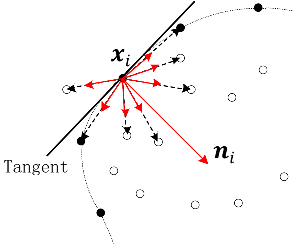

Figure 2. The schematic of local normal vector generated

Transformed the local normal vector $\boldsymbol{n}_i$ into the local unit normal vector $\boldsymbol{n}_i^T$, calculated the angle θ_{ij} between $\boldsymbol{n}_i^T$ and each unit vector $\boldsymbol{v}_{ij}$:

$$\theta_{ij} = \cos^{-1}(\boldsymbol{n}_i^T \cdot \boldsymbol{v}_{ij}), \quad 0 \le \theta_{ij} \le \pi \qquad (3)$$

Point $\mathrm{x_j}$ locate in the inner side of the hyper tangent plane, which side has a large amount of data, if the angle $\theta_{\mathrm{ij}} < \frac{\pi}{2}$; point $\mathrm{x_j}$ locate on the hyper tangent plane, if the angle $\theta_{\mathrm{ij}} = \frac{\pi}{2}$; and if the angle $\theta_{\mathrm{ij}} > \frac{\pi}{2}$, $\mathrm{x_j}$ locate in the outside of the hyper tangent plane. All or most of the values of the data points $\mathrm{x_i}$ on the boundary are satisfy $\theta_{\mathrm{ij}} \le \frac{\pi}{2}$. Thus, the following method can be used to determine whether point $\mathrm{x_i}$ is a boundary data point:

$$p_i = \frac{1}{k}\sum_{j=1}^{k} l_j\,, l_j = \begin{cases} 1, & if\ \ \theta_{ij} \le \frac{\pi}{2}; \\ 0, & \text{else.} \end{cases} \qquad (4)$$

For the boundary point $\boldsymbol{x}_i$ in Fig.2, the $p_i = 1$ if without the effect of interference data which locate in the outside. In Fig.1, because points B and C locate in sunken boundary region, there are few data points on the outside of the hyper tangent plane, thus for the Eq.4, $p_i = 1 - \gamma$ (in general the value of γ is very small). By the above analysis, it can be set that the corresponding point $\boldsymbol{x}_i$ is the boundary data of data set, if $p_i \ge 1 - \gamma$. Through detecting the de-noised data using above method, we can get a simple boundary data set to serve the follow-up SVC progress.

3 Cluster Labelling Method of CCL Based on CU

With the technology theory of Support Vector Machine, SVC can solve the problem of sort learning under the condition of no supervision. The SVC algorithm mainly includes two stages of optimization and cluster label. In the optimization phase, data are mapped to a high dimensional space by kernel function, and then seek the minimum hypersphere which obtains the all data set in the high dimensional space, and then can get support vector and the minimum hypersphere. In the cluster label phase, Complete Graph (CG) [5] method is used to determine the label of support vector data by detecting the points between two support vectors whether locate in the minimum hypersphere.

For given dataset $\{x_1, x_2, \cdots, x_n\}$, $x_i \in R^d$, a total of n samples of d dimensional, a minimum hypersphere of radius is R which covers all data points can be found in the feature space, we can get a soft-margin target function of SVC:

$$\begin{aligned} &\min\ \ R^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{s.t.}\ \ \|\phi(x_i) - a\|^2 \le R^2 + \xi_i,\quad \xi_i \ge 0 \end{aligned} \qquad (5)$$

Where constant C is the penalty factor, which balances the relationship between radius and slack; $\xi = (\xi_1, \xi_2, \cdots, \xi_n)^T$ is the slack which permits few samples exist outside of the hypersphere; which ϕ is the mapping from data space to feature space; $a = (a_1, a_2, \cdots, a_d)^T$ is the center of the hypersphere; $\|\cdot\|$ refers to the Euclidean norm. Convert Eq. (5) to Lagrangian function and introduce Gaussian kernel function $K(x,y) = \phi(x)^T \cdot \phi(y) = \exp(-q\|x - y\|^2)$, which q is the width of the kernel function. Then Eq. (5) can be simplified to:

$$\begin{aligned} &\min\ \ \sum_{i,j=1}^{n}\beta_i\,\beta_j K(x_i, x_j) \\ &\text{s.t.}\ \ \sum_{i=1}^{n}\beta_i = 1,\quad 0 \le \beta_i \le C \end{aligned} \qquad (6)$$

Which $\beta_i, \beta_j \ge 0$ is the Lagrangian Multiplier. At the same time, the distance between the projection of sample data in feature space and the center of hypersphere can be obtained:

$$\begin{aligned} R^2(x) &= \|\phi(x_i) - a\|^2 \\ &= 1 - 2\sum_{j=1}^{n}\beta_i\,K(x_i, x) + \sum_{i,j=1}^{n}\beta_i\,\beta_j K(x_i, x_j) \end{aligned} \qquad (7)$$

Because of Eq. (6) need to meet the condition of Karush-Kuhn-Tucker, thus: 1) When $\beta_i = 0$, the corresponding $\phi(x_i)$ locate in the minimum hypersphere in feature space, and x_i locate in the cluster in the data space; 2) When $\beta_i = C$, the corresponding $\phi(x_i)$ locate outside of minimum hypersphere in feature space, and x_i locate outside of cluster in the data space, these points are defined as the bounded Support Vectors (bSVs); 3) When $0 < \beta_i < C$, the point $\phi(x_i)$ in the feature space locate on the surface of minimum hypersphere, the corresponding x_i locate on the edge of the cluster, these points are defined as the Support Vectors (SVs).

Through the above analysis, we can find that Eq. (7) is just related to the SVs and the bSVs, which is called the Function Support. We can get the radius of minimal hypersphere $R(v_i) = r$ by substitution some SVs v_i into Eq. (7).After the SVs are obtained, the incidence matrix A is established by CG cluster label method. Using incidence matrix traversal of all support vectors, label the support vectors interconnected as the same cluster. Finally, complete the remaining data category based on the cluster label of support vectors.

Because of random sampling test and construction of the correlation matrix, CG cluster labelling algorithm has a greater randomness and computation. Thus in order to avoid randomness and improve the efficiency of cluster labelling, Sei-Hyung Lee proposed the CCL cluster labelling algorithm, with the help of support vector structure-distribution relationship in high dimensional feature space, it constructs high dimensional cone for each support vector, and these support vector cones approximately covering the minimum hypersphere. Transformed to the data space, the contour of data set can be covered by hypersphere accordingly, and then label support vectors by judging whether each hyper sphere intersects or not.

3.1 CCL cluster calibration principle

Set V is the produced support vector data set after optimization phase, $\mathrm{v_i}, \mathrm{v_j} \in \mathrm{V}$， $1 \le \mathrm{i}, \mathrm{j} \le \mathrm{N_{sv}}$, due to the introduction of the Gaussian kernel function, $\mathrm{K}(\mathrm{v_i}, \mathrm{v_i}) = 1$， $\|\phi(\mathrm{v_i})\| = 1$， projecting all data to the surface of unit hypersphere B whose origin is O . The minimal hypersphere S center is b, then:

$$\|b\| \le \|\beta_1 \phi(v_1)\| + \cdots + \|\beta_n \phi(v_n)\| = 1 \qquad (8)$$

Eq.(8) shows that the minimal hypersphere S center b locates in the unit hypersphere's surface or inside. Because the support vectors are distributed on the surface of minimal hypersphere, and all the data are projected onto the surface of unit hypersphere, so the support vectors locate on the dotted circle with the minimal hypersphere S and the unit hypersphere B, and the data within the cluster locate on the region P dotted with S and B. As shown in Fig.3.

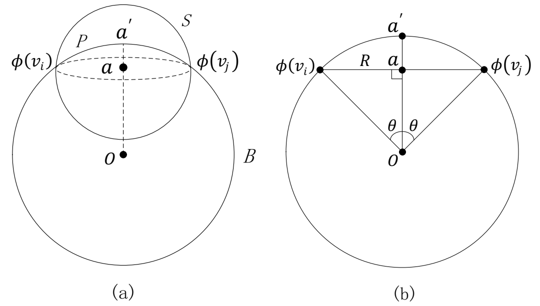

Figure 3. The space distribution of support vectors (a) three-dimensional; (b) two-dimensional cross section.

Where a is center of the support vector dotted circle, a′ is the intersection point of Oa's extension line and the unit ball B, Fig.3 (b) is the cross section of Fig.3 (a). It can be seen from the above analysis, for any support vector $\phi(\mathrm{v_i})$, the angle between $\overrightarrow{\mathrm{O}\phi(\mathrm{v_\imath})}$ and $\overrightarrow{\mathrm{Oa}}$ is θ, i.e. $\angle\phi(\mathrm{v_i})\mathrm{Oa} = \theta$. Set $\varepsilon_{\mathrm{v_i}}$ is the cone formed by $\overrightarrow{\mathrm{Oa}}$ rotate along $\overrightarrow{\phi(\mathrm{v_\imath})}$, and a′ is the intersection point for all the cones, which can be understood as the union of all cones covering cap region P approximately. Correspondingly in the data space, $\varepsilon_{\mathrm{v_i}}$ corresponding to the hypersphere $\rho_{\mathrm{v_i}}$ which center is $\mathrm{v_i}$, radius is $\|\mathrm{v_i} - \mathrm{g_i}\|(\mathrm{g_i} \in \mathrm{R^d}$， $\angle\phi(\mathrm{v_i})\mathrm{O}\phi(\mathrm{g_i}) = \theta)$, $\cup_\mathrm{i}(\rho_{\mathrm{v_i}})$ cover the contour of the data set P′ approximately, and $\phi(\mathrm{P'}) \subseteq \mathrm{P}$.

The CCL algorithm label the data that is not larger than the angle θ as the same type in feature space, transformed into data space, we can label by judging the value of radius $\|v_i - g_i\|$. Because$\|\phi(v_i)\| = \|\phi(g_i)\| = \|a'\| = 1$， thus $cos\,\theta = \angle\phi(v_i)O\phi(g_i) = \angle\phi(v_i)Oa' = \langle\phi(v_i) \cdot a'\rangle = \langle\phi(v_i) \cdot \phi(g_i)\rangle = K(v_i, g_i)$, and then:

$$\|v_i - g_i\| = \sqrt{-\frac{\ln(\cos\theta)}{q}} \qquad (9)$$

For any $v_i \in \mathrm{V}$, $\angle\phi(v_i)Oa' = \theta$, the radius of hypersphere in data space are equal correspondingly, so let $\|v_i - g_i\| = Z$. b and a are coincide, and a is minimal hypersphere's center, the radius of the hyper sphere is R. Due to $\|\phi(v_i)\| = 1$, from Fig.3 (b), it can be seen that $cos\,\theta = \|a\|$, $\|a\|^2 = 1 - R^2$, $\|a\| = \sqrt{1 - R^2}$, introduce to Eq. (9), and then:

$$\|v_i - g_i\| = Z = \sqrt{-\frac{\ln(\sqrt{1-R^2})}{q}} \qquad (10)$$

Next we can determine two support vector $\mathrm{v_i}$ and $\mathrm{v_j}$ whether or not for the same category by judging the two hypersphere $\rho_{\mathrm{v_i}}$, $\rho_{\mathrm{v_j}}$ which are formatted by $\mathrm{v_i}$ 、 $\mathrm{v_j}$ whether intersect or not in data space, i.e. if $\|\mathrm{v_i} - \mathrm{v_j}\| \le 2\mathrm{Z}$, labelling the support vector $\mathrm{v_i}$ and $\mathrm{v_j}$ to the same category. After getting the value of Z, the Depth-First Search (DFS) can be used to complete the labelling of all the SVs, then, according to the Euclidean distance, the remaining data are classified into the nearest SVs' label.

3.2 Parameter optimization based on CU

From Eq. (10), it can be seen that the value of q will directly determine the value of Z, thereby affecting the number of generated clusters. If $q \to 0$, $Z \to \infty$, all hypersphere ρ_{v_i} are formed by SVs intersect each other, and every ρ_{v_i} cover the entire data set, thus all the data can be labelled to the same category. If $q \to \infty$, $Z \to 0$, the entire data set is covered by $\bigcup_i(\rho_{v_i})$, and all the hyperspheres are disjoint, so as to generate categories, whose number is the same as the data samples. In summary, how to select the kernel function width is very important to the whole clustering results.

To solve this problem, this paper proposes a method of parameter optimization based on CU, which guides the optimization of the kernel function width q by measuring the overall cluster quality with calculating the CU of clustering result. The CU is defined as:

$$\mathrm{CU} = \frac{\sum_l \Pr[C_l] \sum_i \sum_j \left(\Pr[a_i = v_{ij} | C_l]^2 - \Pr[a_i = v_{ij}]^2 \right)}{k} \tag{11}$$

Where C_l ($l = 1,2,\cdots,k$) is the l th category of clustering result, a_i is the i th attribute, $v_{ij} = v_{i1}, v_{i2}, \cdots, v_{ij}$ is the attribute value. The upper formula can be understood as the probability $\Pr[a_i = v_{ij} | C_l]$ of a_i, whose attribute’s value is v_{ij} for a certain category C_l is a good estimate relative to the probability $\Pr[a_i = v_{ij}]$ of a_i, whose attribute’s value is v_{ij} for all categories. Extend Eq. (14), make it suit the radar parameters:

$$\mathrm{CU} = \frac{1}{k} \sum_l \Pr[C_l] \frac{1}{2\sqrt{\pi}} \sum_i \left(\frac{1}{\sigma_{il}} - \frac{1}{\sigma_i} \right) \tag{12}$$

Where σ_i is the standard deviation of a_i attribute, and σ_{il} is the standard deviation of a_i attribute in the l category. Because the essence of clustering is to estimate the value of each category attribute accurately, thus the CU can better reflect the clustering results.

Through analyzing the experimental results, we can get that with the increase of q, the number of support vectors are increased, and the clustering contour is shrinking gradually, while the number of bSVs is controlled by penalty factor C. For parameter optimization, firstly set$C = 1$, $q_1 = \frac{max\|v_i - v_j\|^2 + min\|v_i - v_j\|^2}{2max\|v_i - v_j\|^2 \cdot min\|v_i - v_j\|^2}$ in the range of $\left[q_{min} = \frac{1}{max\|v_i - v_j\|^2}, q_{max} = \frac{1}{min\|v_i - v_j\|^2} \right]$, calculate the CU of clustering results; and then let q_2 is the middle point of $[q_{min}, q_1]$, and q_3 is the middle point of $[q_1, q_{max}]$, calculate CU_2 and CU_3; if there are certain categories with one point, decrease the value of C, getting the CU by taking middle point method; at last compare these gotten CUs and select the q^* corresponding to CU_{max} as the appropriate kernel function width.

4 Improved Signal Sorting Process

The proposed support vector clustering method based on boundary extraction and CU is divided into three parts: boundary extraction, support vector extraction and parameter optimization based on CU, the process is as shown in the figure 4:

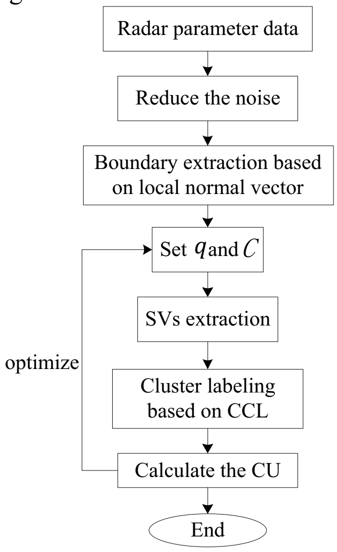

Figure 4. The process of support vector clustering based on boundary and CU.

Step 1: the parameter data extracted from radar signal buffer, after the noise reduction processing, boundary information is extracted by local normal vector;

Step 2: parameter setting, the SVs data is extracted by SVC;

Step 3: first label the SVs through CCL method, and then use DFS classifying the remaining samples; then calculate the CU of clustering results and guidance parameters optimization until get the last clustering results.

5 Simulation Test Analysis

To verify the effectiveness of support vector clustering method based on boundary extraction and CU (BECU-SVC), four groups of radar pulse data with different characteristic parameters are being simulated, totally the number of samples are 7617. Select RF, DOA and PW as the characteristic parameters of radar signal sorting, the parameter information of each group is shown in Table 1. At the same time, for simulating the real environment of reconnaissance, 5% additional jitter error are mixed to each parameters.

The original data are normalized, and the local normal vector is used to extract the contour of data set. In order to achieve the purpose of reducing the data samples, and try not to lose the contour information. Set the parameters $k = 30$, $\gamma = 0.04$ of the Sect.2 to extract the boundary data through multiple simulation experiments. The data distributions of the boundary extraction fore-and-aft are shown in Fig. 5.

Table 1. The Feature Information of Samples.

Signal type	DOA/(°)	PW/(μs)	RF/(MHz)		Number of pulses
			Type	Range	
Pulse sequence 1	42-50	1.2-1.4	Frequency agility	2080-2250	1824
Pulse sequence 2	55-78	1.0-1.2	Frequency agility	2750-2850	1853

Pulse sequence 3	72-80	1.2-1.25	Frequency hopping	2150-2350	2049
Pulse sequence 4	56-64	1.3-1.45	Frequency agility	2650-2810	1891

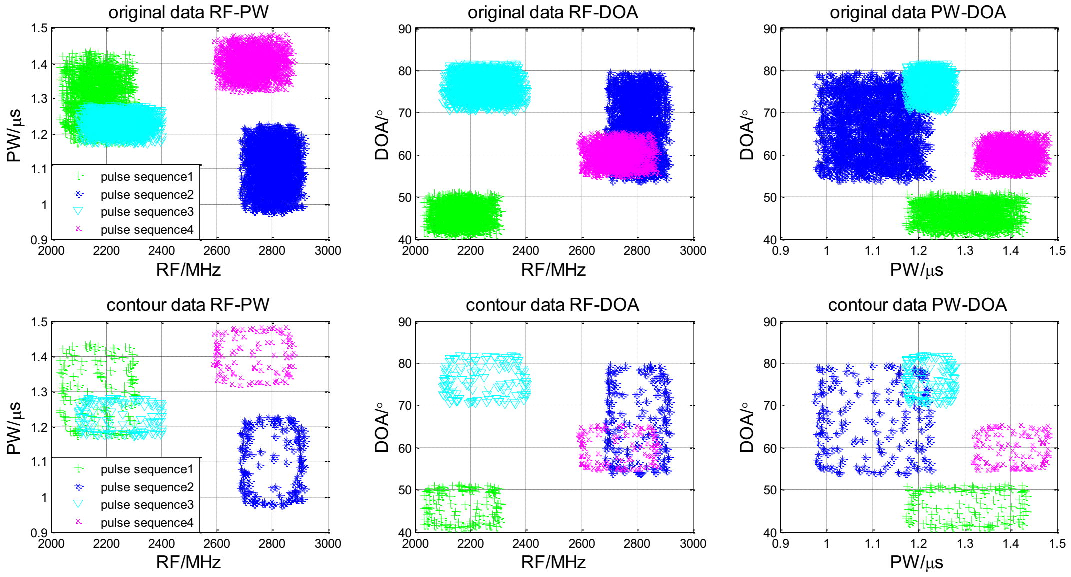

Figure 5. Comparison of data distribution before and after boundary extraction.

From the data distribution in Fig. 5, it is clear that the contour of each pulse sequence is clear after the boundary extraction. The samples for SVC are reduced from 7617 to 503 by using these contour data, which reduces 93.4% approximately, and reduce the computational cost greatly.

As mixed pulse sequences in Table 1 for the sample data, and process these samples by CG, CCL and BECU-SVC respectively. Under the guidance of CU, set the optimal parameter, and the results obtained by 100 Carlo Monte are shown in Table 2.

Table 2. Sorting Result Statistics.

	CG	**CCL**	**BECU-SVC**
Total pulses	7617	7617	7617
Missing pulses	34	0	0
Error sorting pulses	197	93	59
Correct sorting pulses	7420	7524	7558
CU	2.838	3.092	3.262
Sorting accuracy (%)	97.4	98.8	99.2
Time-consuming (s)	326.762	63.672	8.723

By comparing the results of Table 2, we can find the results of CG algorithm have certain randomness, the results with more missing and error pulses, at the same time the incidence matrix needs great computation, thus the time-consuming of CG is the longest. CCL algorithm only needs to calculate the value of Z, through judging the relationship between SVs and Z to complete labelling in data space after extracting the SVs samples, thus avoiding the randomness and the construction of large incidence matrix, but because there is no reduction of the sample data, the calculation is still very large in solving the quadratic optimization problem. In this paper, the BECU-SVC method uses the boundary extraction to reduce the sample data which construct kernel matrix and participate in the quadratic optimization programming. At the same time, improved CCL method is used to reduce the computational complexity of cluster labelling, thus it can improve the efficiency of sorting significantly, and can get better accuracy, the time-consuming of BECU-SVC method is only 2.67% of the traditional CG algorithm.

6 Conclusion

With the rapid development of information technology, the production of new technology system for radar is promoted, and it has brought more complex and changeable signal patterns. Therefore in the future battlefield, we will be threatening by emerging in endlessly unknown radar signals. In order to meet the challenge, aiming at the problem of high time-consuming for unknown radar signal sorting with less prior knowledge, a new method of support vector clustering based on boundary extraction and CU for radar signal sorting is studied. Through comparing the simulation results, it is proved that the proposed BECU-SVC algorithm can not only improve the accuracy of signal sorting, but also greatly reduce the computational complexity, and has good timeliness. However, in the process of experiment, there is still room for parameter optimization of q, which will be the focus of the next step.

References

1 He Ming-Hao, *Radar countermeasure signal processing,* (2010) 6-9.

2 Yuan Ping, YunFengChang, Yajian Zhou, et al. Knowl Inf Syst, **43**, 2 (2015) 281-310.

3 Ian H. Witten, Eibe Frank and Mark A. Hall. *Data Mining: Pratical Machine Learning Tools and Techniques, Third Edition* (2014) 203-205.

4 Sei-Hyung Lee, Karen M.Daniels. Pattern Anal Applic, **15**, 3 (2012) 327-344.

5 Ben-Hur A, Horn D，Siegelmann H. T., et al. JMLR, (2012) 125-137.