A New Perspective of Support Vector Clustering with Boundary Patterns

Yuan Ping, Huina Li, Yong Zhang, and Zhili Zhang

Department of Computer Science and Technology
Xuchang University, 461000 Xuchang, China
pyuan.lhn@gmail.com

Abstract. To overcome the pricey computation required by redundant kernel function matrix and poor label performance, in a novel perspective, we present support vector clustering with boundary patterns (BPSVC for abbreviation) for efficiency. For the first phase, the conventional method of estimating the support vector function with the whole data is altered by only essential boundary patterns. Thence, BPSVC only need to solve a much simpler optimization problem. For the second phase of cluster labeling, both convex decomposition and cone cluster labeling method are employed by an ensemble labeling strategies for further improvements on accuracy and efficiency. Both theoretical analysis and experimental results show its superiorities in comparison of the state-of-the-art methods, especially for large-scale data analysis.

Keywords: data analysis, support vector clustering, convex decomposition, boundary pattern, cluster labeling.

1 Introduction

With the advantage of generating cluster boundaries of arbitrary shape, support vector clustering (SVC)[12, 15] has attracted many researchers and been extensively applied to wide variety of domains, e.g., instance-based learning, pattern denoising and medical information processing etc[6, 13, 14].

However, the literatures show that training to estimate a support function and cluster labeling are two major bottlenecks which might degrade its popularity. As a quadratic programming problem, the prior can be solved by many classic algorithms, such as sequential minimal optimization and entropy-based algorithms[16], in approximately $O(N^2)$ kernel evaluations, here N is the number of data points. In addition, data block based methods [1, 17] may be a good choice even though persistent parameter tuning is generally required. Correspondingly, the time complexity of the latter is $O(N^2m)$ with $m \ll N$ which is the sample rate on each edge. Naturally, the studies argue that cluster labeling takes most of the computation time in the entire SVC clustering process. For efficiency, especially on large-scale data, some insightful methods have been designed to replace the complete graph (CG)[15], such as support vector graph (SVG)[15], proximity graph of delaunay (DD)[9], minimum spanning tree (MST),

Z.-H. Zhou, F. Roli, and J. Kittler (Eds.): MCS 2013, LNCS 7872, pp. 224–235, 2013.

k-nearest neighbor (kNN)[10], divide and conquer-based[11, 17], cone cluster labeling (CCL)[8], equilibrium based approaches[18–21], fast support vector clustering (FSVC) [14], double centroids (DBC) labeling [4] and position regularized support vector clustering (PSVC)[2], etc.

However, we find that many of them reach lower time consumption with higher error or improve accuracy at the cost of efficiency. We consider to make improvements in terms of decreasing both the number of points N and the sample rate m. Three works are included in the proposed support vector clustering with boundary patterns (BPSVC) method, i.e., selecting critical points on cluster boundaries, constructing a so-called minimum hypersphere in feature space by the selected points, and integrating our convex decomposition based clustering labeling (CDCL)[3] and CCL to complete labeling under an ensemble labeling strategy. Benchmarks depict the main contributions of this paper including:

- The proposal of constructing hypersphere by only the boundary points at much lower cost of time and space to estimate the support function.
- The ensemble labeling strategy, especially for transferring connectivity checks between all pair-wise points (or SVs in [15], or stable equilibrium points (SEPs) in [18]) into between neighboring convex hulls with significantly reduced sample rate to avoid redundant checks.

2 Preliminaries

2.1 Estimating a Trained Support Function

Following [14, 13], the support function is defined as a positive scalar function $f : \mathbb{R}^n \to \mathbb{R}^+$ where a level set of f estimates a support of a data distribution and which can be decomposed into several disjoint connected components corresponding to different clusters. In support vector domain description (SVDD)[12], estimating a support function is to find the exact SVs by solving the dual problem in Eq.(1) where C is a constant for penalty and $\mathbf{x}_i$ corresponds to coefficient $\beta_i (i = 1, \ldots, N)$ if its $0 < \beta_i < C$ is a support vector.

$$\begin{aligned} &\max_{\beta_j} \sum_j K(\mathbf{x}_j, \mathbf{x}_j)\beta_j - \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{s.t.} \quad \sum_j \beta_j = 1, \quad 0 \le \beta_j \le C, \quad j = 1, \ldots, N \end{aligned} \tag{1}$$

By optimizing Eq.(1) with Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-q||\mathbf{x}_i - \mathbf{x}_j||^2}$, the objective trained support function can be formulated by a squared radial distance of the image of $\mathbf{x}$ from the sphere center $\boldsymbol{\alpha}$ given by

$$f(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}) - 2\sum_j \beta_j K(\mathbf{x}_j, \mathbf{x}) + \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{2}$$

Theoretically, the squared radius R^2 is usually defined by the value of $f(\mathbf{x}_i)$ while $\mathbf{x}_i$ is one of SVs.

2.2 Cluster Assignments

Since SVs locate on the border of clusters in data space, a simple graphical connected-component method can be used for labeling. For any two points, $\mathbf{x}_i$ and $\mathbf{x}_j$, we can check m segmers sampled on the line segment connecting them by traveling its image in the hypersphere. According to Eq.(2), $\mathbf{x}_i$ and $\mathbf{x}_j$ should be labeled the same cluster index while all the m segmers are always lying in the hypershpere, i.e., $f(\mathbf{x}_{\tilde{m}}) \leq R^2$ for $\tilde{m} = 1, \cdots, m$.

3 Support Vector Clustering with Boundary Patterns

Notice that the hypersphere is determined by SVs which are a subset of boundaries. Obviously, either SVs or boundaries are sufficient for constructing the hypersphere. Thus, we prefer a transferred strategy which collects a candidate set of them from cluster boundaries.

3.1 Obtaining Boundary Patterns

To select the most informative points, a border-edge pattern selection method presented by Li and Maguire [22] is preferred in this study. It confirms that, on the cluster boundaries, every point actually has all or most of its nearest neighbors sitting on one side of the tangent plane passing through it. Therefore, boundaries identification is to count the ratio of a point's nearest neighbors on two sides. Following [22], for a given point $\mathbf{x}_i$ with its k nearest neighbors $\mathbf{x}_j (j = 1, 2, \ldots, k)$, we can reformulate the procedure as follows:

- Setting a threshold γ $(0 < \gamma < 1)$ to control the curvature of the aforementioned surface.
- Generating normal vector $\boldsymbol{n}_i = \sum_{j=1}^{k} \boldsymbol{v}_{ij}$, where $\boldsymbol{v}_{ij} = \mathbf{x}_j - \mathbf{x}_i$.
- Calculating $l_i = \frac{1}{k} \sum_{j=1}^{k} g(\boldsymbol{n}_i^T \cdot \boldsymbol{v}_{ij})$, where the function $g(\mathbf{x})$ returns 1 if $\mathbf{x} \geq 0$, otherwise it returns 0.
- Cluster boundary identification. If $l_i \geq 1 - \gamma$, then $\mathbf{x}_i$ is considered as one of the boundary points.

3.2 Constructing Hypersphere for Support Function

Given a data set $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ in $\mathbb{R}^d$, selecting the cluster boundaries will return a subset $\mathcal{Z}_S = \{\mathbf{x}_{s_1}, \mathbf{x}_{s_2}, \ldots, \mathbf{x}_{s_M}\} \subseteq \mathcal{X}$ which contains the most informative points for constructing the hypersphere. That is, each point $\mathbf{x}_{s_i}$ $(i = 1, \ldots, M)$ can be approximately expected as one of SVs.

Estimating Coefficients for Boundaries. Consider the definition of SVs[15], we have

$$f(\mathbf{x}_{s_1}) = f(\mathbf{x}_{s_2}) = \cdots = f(\mathbf{x}_{s_M}) \tag{3}$$

Since $K(\mathbf{x}_{s_i}, \mathbf{x}_{s_i})$ with Gaussian kernel and $\sum_{i,j} \beta_i \beta_j K(\mathbf{x}_{s_i}, \mathbf{x}_{s_j})$ are respectively equal in Eq.(3), it is easy to check that Eq.(3) has the following expression:

$$\begin{cases} \sum_j \beta_j [K(\mathbf{x}_{s_j}, \mathbf{x}_{s_1}) - K(\mathbf{x}_{s_j}, \mathbf{x}_{s_2})] = 0 \\ \sum_j \beta_j [K(\mathbf{x}_{s_j}, \mathbf{x}_{s_1}) - K(\mathbf{x}_{s_j}, \mathbf{x}_{s_3})] = 0 \\ \quad \cdots \\ \sum_j \beta_j [K(\mathbf{x}_{s_j}, \mathbf{x}_{s_1}) - K(\mathbf{x}_{s_j}, \mathbf{x}_{s_M})] = 0 \end{cases} \tag{4}$$

where $j \in [1, M]$ and $\sum_j \beta_j = 1$. Let $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots, \beta_M]^T$, $\mathbf{0} = [0, 0, \cdots, 0]^T$ and $\boldsymbol{Q} = [Q_1, Q_2, \cdots, Q_{M-1}]^T$ where

$$\begin{aligned} Q_j = [1 - K(\mathbf{x}_{s_1}, \mathbf{x}_{s_{j+1}}), & K(\mathbf{x}_{s_2}, \mathbf{x}_{s_1}) - K(\mathbf{x}_{s_2}, \mathbf{x}_{s_{j+1}}), \cdots, \\ & K(\mathbf{x}_{s_M}, \mathbf{x}_{s_1}) - K(\mathbf{x}_{s_M}, \mathbf{x}_{s_{j+1}})] \end{aligned} \tag{5}$$

then the Eq.(4) can be further written as

$$\begin{aligned} & \boldsymbol{Q}\boldsymbol{\beta} = \mathbf{0} \\ \text{s.t.} \quad & \sum_i \beta_i = 1, \quad \beta_i \geq 0 \end{aligned} \tag{6}$$

Since $\boldsymbol{Q}$ is determined by the cluster boundaries, $\boldsymbol{\beta}$ can be found by solving this linear system of equations with inequality constraint. Using $\boldsymbol{\beta}$ and Eq.(2), the required hypersphere with radius R can be constructed. However, the Eq.(6) can hardly be solved directly. Thus an alternative method is constructed by converting Eq.(6) into a quadratic programming problem.

Consider Eq.(6), we get

$$\sum_j (Q_j \boldsymbol{\beta})^2 = \begin{bmatrix} Q_1\boldsymbol{\beta} \\ Q_2\boldsymbol{\beta} \\ \cdots \\ Q_{M-1}\boldsymbol{\beta} \end{bmatrix}^T \times \begin{bmatrix} Q_1\boldsymbol{\beta} \\ Q_2\boldsymbol{\beta} \\ \cdots \\ Q_{M-1}\boldsymbol{\beta} \end{bmatrix} = 0 \tag{7}$$

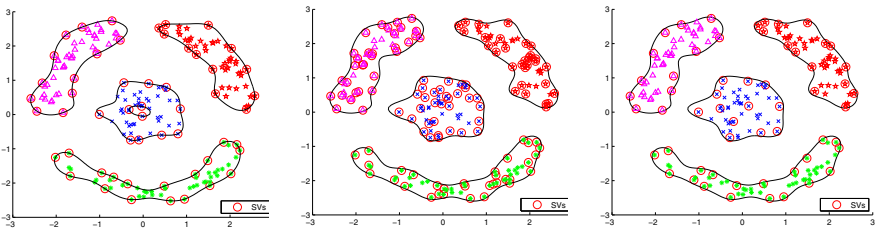
where Q_j $(j = 1, \cdots, M-1)$ is either positive or negative and each element of $Q_j\boldsymbol{\beta}$ or $(Q_j\boldsymbol{\beta})^2$ is 0. Naturally, it can be approximately reformulated by

$$\begin{aligned} & \min \quad \boldsymbol{\beta}^T \boldsymbol{H} \boldsymbol{\beta} \\ \text{s.t.} \quad & \sum_j \beta_j = 1, \beta_j \geq 0, j = 1, \cdots, M \end{aligned} \tag{8}$$

where $\boldsymbol{H} = \boldsymbol{Q}^T\boldsymbol{Q}$ is a Hessian matrix in $\mathbb{R}^{M \times M}$. Note that it is a standard convex quadratic program, its global optimal solution can be obtained effectively and the value of the object function can be guaranteed very close to 0 for $(Q_j\boldsymbol{\beta})^2 \geq 0$. Obviously, the penalty factor C is no longer existing.

Removal of Less Informative Points. For real problem, it is poetical that all the boundary patterns are expected to be SVs. One aspect is that none of the

boundary pattern selection methods can guarantee a hundred percent correct. On the other hand, taking two nearest neighboring patterns to constructing the hypersphere is unnecessary. As depicted by Fig.1, compared with CG, in spite of a correct result achieved by BPSVC (see Fig.1b) which estimates an acceptable support function by solving the convex quadratic program (8), too many points lying on the cluster boundaries are recognized as SVs. Obviously, some of these data points are useless.



(a) CG **(b)** BPSVC (original) **(c)** BPSVC (neat SVs)

Fig. 1. Comparison of cluster boundaries and clustering results on `ring`[14]. (a) CG[15] ($q = 2, C = 1$). (b) BPSVC ($k = 10, \gamma = 0.2, q = 3.125$) before removing any boundary point. (c) BPSVC ($k = 10, \gamma = 0.2, q = 3.125$) after removing boundary points whose corresponding coefficient lower than 10^{-3}.

Our intuitive solution is quite simple: since the importance of a data point $\mathbf{x}_{s_j}$ relates to its corresponding coefficient β_j ($j = 1, \cdots, M$) directly in constructing center $\boldsymbol{\alpha} = \sum_j \beta_j \Phi(\mathbf{x}_j)$, where $\Phi(\cdot)$ is a nonlinear map function. To further obtain a neat data set, those boundary points with coefficients lower than a predefined threshold β_s should be removed for uselessness or little information. Obviously, a smaller β_s allows more redundant data to profile cluster boundaries accurately, but sometimes go along with overfitting; whereas a smoother profile would be generated by a greater β_s at the risk of much more overlapped regions between clusters. Following the principle of SVC, a large number of experiments suggest that an appropriate threshold β_s for removing the less informative points should be $10^{-\lceil \lg N \rceil}$ more or less, e.g., $\beta_s = 10^{-3}$ for `ring` in Fig.1c.

Estimating the Radius of the Hypersphere. Notice that programming (8) is a compromise of programming (1). Therefore, the strict zero can hardly be achieved though we expected it should be. After the removal of useless boundary points, in reality, we get a reduced set $\mathcal{Z}_R = \{\mathbf{x}_{r_1}, \mathbf{x}_{r_2}, \ldots, \mathbf{x}_{r_L}\} \subseteq \mathcal{Z}_S$ with $L \leq M$, which are the exact SVs. However, the relation of their distances to the center of the hypersphere is

$$f(\mathbf{x}_{r_1}) \approx f(\mathbf{x}_{r_2}) \approx \cdots \approx f(\mathbf{x}_{r_L}) \tag{9}$$

Finally, consider the numerical problem in practical[1], we construct the hypersphere whose radius is the maximum distance from $\mathbf{x}_{r_l}$ ($l \in [1, L]$) to its center, i.e., $R^2 = \max_{l=1}^{L} f(\mathbf{x}_{r_l})$.

3.3 Ensemble Labeling Strategy for Efficiency

Following the aforementioned works, a subset of boundary points are selected as exemplars for cluster assignments. Intuitively, three conventional strategies can be employed, i.e., checking the full pairs of SVs like SVG, directly calculating the distance between each pair of SVs to verify if their cones are intersected (like CCL), and traveling the line segments connecting each pair of SEPs converged from the SVs while the solution of finding the minimal hypersphere, $\partial \mathbf{x}/\partial t = -\nabla f(\mathbf{x})$, is considered as a gradient dynamical system (e.g., reduced complete graph (R-CG) [4, 18]). However, as the previously stated, these strategies suffer from some obvious drawbacks. For the first, it is time-consuming while processing large-scale or high dimension data[4, 14]. Although the adjacency matrix of SVs can be calculated very fast, a radius lower than 1 is essential for the second strategy to find connected components. However, due to numerical problem, the radius which should be lower than 1 cannot be guaranteed. Therefore, CCL cannot be employed directly. Finally, as noted in Ref.[20] and detailed by Ref.[4], only SEPs employed by the third strategy to represent data for connectivity checking usually lead to relatively high error on irregular shaped data set. Therefore, to achieve improvements on both efficiency and clustering quality, we prefer a simple but effective ensemble labeling strategies of CDCL and CCL. As depicted in lines 4∼8 of Algorithm 1, it prefers CDCL since the constructed hypersphere's radius is greater than 1; otherwise, CCL is employed.

4 Implementation

In this section, we give description of the proposed BPSVC method as well as some remarks distinguishing from the others. For the given γ and k, line 2 of Algorithm 1 collects cluster boundaries for constructing the objective hypersphere by line 3. Three essential elements of the support function, i.e., the radius of the hypersphere R, the final set of SVs $\mathcal{Z}_R$ with respect to their coefficient $\boldsymbol{\beta}$ are obtained by ConstructHypersphere($\mathcal{Z}_s$, q, β_s). To get a nest set of SVs, the threshold β_s for removing useless boundary points is set in line 1 of the Algorithm 1. Notice that although in Algorithm 1 we start from selecting cluster boundaries by measuring the full data set, the computation is significantly reduced as the calculation is repeated in data space and a rather lower size of data ($M, L \ll N$) remained for both line 3, and ConnectivityAnalysisofConvexHulls($\mathcal{Z}_R, q$) in line 5 or ConnectivityAnalysisofSVs($\mathcal{Z}_R, R$) in line 7. Specifically, taking this tidy data into the rather simple convex quadratic program (8) makes the proposed BPSVC handle large-scale problem efficiently.

Actually, the ensemble labeling strategies is implemented by line 4∼8. Since $R \geq 1$, CDCL is employed to decompose $\mathcal{Z}_R$ into N_c groups for constructing convex hulls (detailed in Ref.[3]). Then the connectivity analysis of clusters can be done between convex hulls. It believes that the far from associate external locations to convex hulls is, the greater probability that the corresponding local region to be sparse distribution with data points is. Practically, it

Algorithm 1. BPSVC($\mathcal{X}, \gamma, k, q$)

Input: the data set $\mathcal{X}$, number of neighbors k
threshold γ and Gaussian kernel width q
Output: clustering labels for all the data points
1 **set** $\beta_s = 1/|\mathcal{X}|$
2 $\mathcal{Z}_s \leftarrow$ SelectClusterBoundaries($\mathcal{X}, \gamma, k$)
3 $\{\mathcal{Z}_R, R, \boldsymbol{\beta}\} \leftarrow$ ConstructHypersphere($\mathcal{Z}_s$, q, β_s)
4 **if** $R \geq 1$ **then**
5 $\quad \boldsymbol{A} \leftarrow$ ConnectivityAnalysisofConvexHulls($\mathcal{Z}_R, q$)
6 **else**
7 $\quad \boldsymbol{A} \leftarrow$ ConnectivityAnalysisofSVs($\mathcal{Z}_R, R$)
8 **end**
9 Labels $\leftarrow$ FindConnComponents($\boldsymbol{A}$)
10 **for each** $\mathbf{x} \in \mathcal{X} \backslash \mathcal{Z}_R$
11 $\quad inx \leftarrow$ find the nearest SV from $\mathbf{x}$
12 $\quad$ Labels[$\mathbf{x}$] $\leftarrow$ Labels[$\boldsymbol{v}_{inx}$]
13 **end**
14 **return** Labels

does reduce the average sample rate m significantly. All of these tasks are completed by function ConnectivityAnalysisofConvexHulls(H) in line 5. After that, the adjacency matrix $\boldsymbol{A}$ is obtained for connectivity analysis by means of any standard algorithm. Otherwise, while R is lower than 1, the invoked function ConnectivityAnalysisofSVs($\mathcal{Z}_R, R$) will check the connectivity among SVs following the CCL method, which could be explained by Ref.[8]. By now, the output of FindConnComponents($\boldsymbol{A}$) in line 9 is an array with size N_c which contains the cluster labels. Finally, the remaining data points are separately assigned with the labels of their nearest SVs.

5 Experiments

5.1 Datasets and Experimental Settings

To demonstrate the performance of the proposed BPSVC, in this section, we conduct comparisons among ten state-of-the-art methods, i.e., CG, DD, k-NN ($k = 4$), MST, R-CG, E-SVC, CCL, FSVC, PSVC and CDCL. The employed data sets (described in Table 1) include: `five-Gaussians`, `twocircles` and `D31` from Refs.[14, 13, 24] and `iris`, `wisconsin`, `zoo`, `movement_libras` and `shuttle` from UCI repository[25]. For fair comparisons, all the simulations are carried out in MATLAB 2011b on system with Intel Dual Core 2.66 GHz and 3GB RAM, and all of the data sets are employed without any preprocessing.

Table 1. Description of the benchmark data sets

Data sets	dims	size	# of classes
twocircles	2	300	2
iris	4	150	3
wisconsin	9	683	2
zoo	16	101	7
movement_libras	90	360	15
five-Gaussians	2	1000	5
D31	2	3100	31
shuttle	9	43500	7

To measure the clustering accuracy, we use adjusted rand index (ARI)[6, 23], rand index (Rand), jaccard coefficient (Jaccard)[5], and normalized mutual information (NMI)[7], which are a widely used similarity measure between two data partitions where both true labels and predicted cluster labels are given.

5.2 Benchmark Results

Table 2 shows the performance achieved by the evaluated algorithms. Notice that the time cost is an average value of ten times of the execution for each data. Rank of each algorithm is given depending on its performance measure followed by corresponding rank (from 1 to 3). In particular, the value of rank 1 for each test item is highlighted by boldface.

Table 2. Benchmark results on data sets with different sizes

Data	Methods	(C, q)	ARI	Rand	Jaccard	NMI	Time(sec.)
twocircles	CG	0.5,0.125	**1.00000**^{a}	**1.00000**^{a}	**1.00000**^{a}	**1.00000**^{a}	10.01
	DD	0.3,0.0638	**1.00000**^{a}	**1.00000**^{a}	**1.00000**^{a}	**1.00000**^{a}	11.66
	kNN	0.32,0.125	0.69679	0.84854	0.69612	0.76529^{c}	2.38
	MST	0.3,0.3252	0.59935	0.79541	0.58953	0.64044	17.56
	R-CG	0.3, 0.1072	0.67695	0.83864	0.67625	0.75411	2.93
	E-SVC	0.2,0.074	0.73547^{c}	0.86785^{c}	0.73486^{c}	0.77801	28.62
	CCL	0.1, 0.0633	0.76193^{b}	0.88581^{b}	0.77423^{b}	0.80800^{b}	9.71
	FSVC	0.1, 50	0.14592	0.57411	0.14552	0.50433	0.75^{b}
	CDCL	0.1,0.1385	**1.00000**^{a}	**1.00000**^{a}	**1.00000**^{a}	**1.00000**^{a}	0.77^{c}
	PSVC	—,0.1385	**1.00000**^{a}	**1.00000**^{a}	**1.00000**^{a}	**1.00000**^{a}	14.78
	BPSVC	—,11.3379	**1.00000**^{a}	**1.00000**^{a}	**1.00000**^{a}	**1.00000**^{a}	**0.22**^{a}
iris	CG	0.46,15.4321	0.61780	0.84886	0.55187	0.67392	1.13
	DD	0.5, 13.8504	0.58334	0.83696	0.51645	0.65589	2.87
	kNN	0.45, 13.8504	0.64143	0.85718	0.57655	0.68600	0.36^{b}
	MST	0.45, 4.0816	0.79457	0.91257	0.74981	0.79331	1.08
	R-CG	0.29,12.5	0.73737	0.89208	0.68095	0.75000	3.07
	E-SVC	0.19,1.3889	0.56812	0.77629	0.59514	0.76117	4.74
	CCL	0.03, 0.7436	0.88579^{b}	0.94953^{b}	0.85776^{b}	0.87052^{b}	0.76
	FSVC	0.33, 2.4691	0.56196	0.77271	0.57842	0.71256	1.07
	CDCL	0.19,9.4518	**0.92218**^{a}	**0.96564**^{a}	**0.90075**^{a}	**0.90112**^{a}	0.67^{c}
	PSVC	—,15.4321	0.61467	0.84904	0.54504	0.68354	1.50
	BPSVC	—, 4.0816	0.85089^{c}	0.93548^{c}	0.8148^{c}	0.82681^{c}	**0.15**^{a}

Table 2. *(Continued)*

Data	Methods	(C, q)	ARI	Rand	Jaccard	NMI	Time(sec.)
wisconsin	CG	0.4, 0.3472	0.77930	0.88971	0.80951	0.69747	66.45
	DD	—	—	—	—	—	—
	kNN	0.4, 0.3472	0.76243	0.88110	0.79463	0.68049	31.65
	MST	0.4, 0.3472	0.66311	0.82986	0.70612	0.61597	85.72
	R-CG	0.1, 0.0868	0.80345	0.90241	0.83473	0.66434	22.25
	E-SVC	0.2,0.1134	0.13441	0.59395	0.54846	0.14341	443.23
	CCL	0.1,0.005	0.90763^{b}	0.94861^{b}	0.90957^{b}	0.81521^{b}	124.35
	FSVC	0.1,1.3889	0.66871	0.83192	0.70242	0.45672	13.06^{c}
	CDCL	0.105,0.0595	0.86850^{c}	0.93482^{c}	0.88747^{c}	0.77555^{c}	$\mathbf{1.31}^{a}$
	PSVC	—,2.8345	0.2574	0.63714	0.52731	0.22633	192.95
	BPSVC	—,4.8828	$\mathbf{0.91712}^{a}$	$\mathbf{0.95882}^{a}$	$\mathbf{0.92658}^{a}$	$\mathbf{0.80295}^{a}$	4.41^{b}
zoo	CG	0.49, 0.4287	0.93421^{c}	0.97663^{c}	0.90367^{c}	0.90763^{c}	0.62
	DD	—	—	—	—	—	—
	kNN	0.49, 0.4287	0.93421^{c}	0.97663^{c}	0.90367^{c}	0.90763^{c}	$\mathbf{0.26}^{a}$
	MST	0.49, 0.4287	0.93421^{c}	0.97663^{c}	0.90367^{c}	0.90763^{c}	0.38^{c}
	R-CG	0.27, 0.3916	$\mathbf{0.95702}^{a}$	$\mathbf{0.98455}^{a}$	$\mathbf{0.93633}^{a}$	$\mathbf{0.92036}^{a}$	3.82
	E-SVC	0.27, 0.3916	$\mathbf{0.95702}^{a}$	$\mathbf{0.98455}^{a}$	$\mathbf{0.93633}^{a}$	$\mathbf{0.92036}^{a}$	19.85
	CCL	0.1,2.5826	0.83426	0.89861	0.79016	0.84893	1.03
	FSVC	0.1,0.2551	0.84625	0.94416	0.79033	0.85331	2.67
	CDCL	0.39,0.5	0.94691^{b}	0.98079^{b}	0.92215^{b}	0.90934^{b}	2.83
	PSVC	—,0.4058	0.7441	0.91723	0.65878	0.85325	0.73
	BPSVC	—,50	0.93421^{c}	0.97663^{c}	0.90367^{c}	0.90763^{c}	0.30^{b}
movement_libras	CG	0.5,5.5556	0.24218	0.93013	0.15922	0.70155	15.14
	DD	—	—	—	—	—	—
	kNN	0.5, 3.8580	0.26661^{c}	0.91360^{c}	0.18532^{c}	0.66459^{c}	7.26^{b}
	MST	0.5, 3.8580	0.24872	0.91102	0.17385	0.65660	41.07
	R-CG	0.5, 5.5556	0.23559	0.93375	0.15194	0.70258	252.89
	E-SVC	—	—	—	—	—	—
	CCL	0.5, 5.5556	0.08987	0.93873	0.04556	0.70874	26.91
	FSVC	0.3,0.4132	0.14205	0.93861	0.04478	0.70352	226.09
	CDCL	0.32,4.8828	0.33195^{b}	0.92098^{b}	0.23010^{b}	0.68084^{b}	78.57
	PSVC	—,4.3253	0.25407	0.91882	0.17412	0.67012	12.18^{b}
	BPSVC	—, 2.9744	$\mathbf{0.37034}^{a}$	$\mathbf{0.92103}^{a}$	$\mathbf{0.25995}^{a}$	$\mathbf{0.68956}^{a}$	$\mathbf{4.20}^{a}$
five-Gaussians	CG	0.15,22.8269	0.47118	0.83982	0.39755	0.63436	89.01
	DD	0.2,19.5313	0.61487	0.89230	0.51115	0.69568	25.63
	kNN	0.21,15.4321	0.26661	0.91360	0.18532	0.66459	7.26^{c}
	MST	0.14,0.5	0.67807	0.90912	0.57554	0.72340	52.84
	R-CG	0.14,17.3010	0.86934^{c}	0.95987^{c}	0.80832^{c}	0.85852^{c}	12.90
	E-SVC	0.14,17.3010	0.85854	0.95707	0.79335	0.84993	775.69
	CCL	0.1,0.005	0.00211	0.31746	0.19109	0.091794	650.39
	FSVC	0.1,50	0.71373	0.91753	0.59742	0.77843	2.39^{b}
	CDCL	0.14,17.3010	$\mathbf{0.88074}^{a}$	$\mathbf{0.96344}^{a}$	$\mathbf{0.82348}^{a}$	$\mathbf{0.86874}^{a}$	7.67
	PSVC	—,29.5858	0.00124	0.30446	0.19180	0.13823	750.21
	BPSVC	—,2.0406	0.87043^{b}	0.96020^{b}	0.80979^{b}	0.86845^{b}	$\mathbf{1.19}^{a}$
D31	CG	0.15, 0.5	0.23448	0.87570	0.15893	0.63159	3594.75
	DD	0.5, 2.9744	0.54946	0.95681	0.39769	0.8488	500.37
	kNN	0.5, 2.9744	0.72032	0.98334	0.57344	0.88154	242.70
	MST	—	—	—	—	—	—
	R-CG	0.1, 5.2029	0.86532^{c}	0.99160^{c}	0.76938^{c}	0.88985^{c}	53.27
	E-SVC	—	—	—	—	—	
	CCL	—	—	—	—	—	—

Table 2. *(Continued)*

Data	Methods	(C, q)	ARI	Rand	Jaccard	NMI	Time(sec.)
	FSVC	1,200000000	0.56109	0.96430	0.40702	0.82011	**4.18**[a]
	CDCL	0.1,5.5556	**0.90199**[a]	**0.99420**[a]	**0.82643**[a]	**0.94355**[a]	19.79[c]
	PSVC	—,12.5	0.45178	0.94391	0.31265	0.80943	7041.01
	BPSVC	—,1.3889	0.87670[b]	0.99224[b]	0.78685[b]	0.90216[b]	6.72[b]
shuttle	CG	—	—	—	—	—	—
	DD	—	—	—	—	—	—
	kNN	—	—	—	—	—	—
	MST	—	—	—	—	—	—
	R-CG	—	—	—	—	—	—
	E-SVC	unknown	0.59[15]	—	—	—	—
	CCL	—	—	—	—	—	—
	FSVC	unknown	0.58[15]	—	—	—	—
	CDCL	—	—	—	—	—	—
	PSVC	—	—	—	—	—	—
	BPSVC	—,0.0078	**0.68574**[a]	**0.86416**[a]	**0.82084**[a]	**0.62654**[a]	**878.19**[a]

Note: [a]Rank 1, [b]Rank 2, [c]Rank 3; "—" means not available.

In terms of accuracy, it is apparent that BPSVC is better for most of data sets (namely `twocicles`, `wisconsin`, `movement_libras` and `shuttle`). Furthermore, it achieves first three ranks consistently on the other data sets. With the help of hypersphere construction and convex decomposition, BPSVC reaches global optimal solutions consistently. For time consumption, BPSVC employs much fewer points to work out the support function, while the others keep solving the same quadratic programming problem with different parameters to achieve their best performance. Thus BPSVC finishes the clustering works fastest on five out of eight data sets. Its advantage is obvious on relative large-scale data, e.g., `shuttle`. Due to memory limitation, we cannot afford the requirement of kernel matrix from FSVC[14], thus a direct citation of experiment result is given.

6 Concluding Remarks

This paper develops a support vector clustering with boundary method namely BPSVC from a new perspective. It gives an optimal solution for these known problems, i.e., a requirement of huge memory for kernel matrix, too many redundant point pairs and a great sample rate.

Even though BPSVC gives consistent results for various cases, how to shrink the cluster boundaries to leave a number of outliers while obtains high quality profiles for clusters in input space, might be an open issue for further improvements on both efficiency and accuracy. And how to redefine the coefficient β_j for the remaining patterns after removing unless data needs to be further investigated as well.

Acknowledgements. This work is supported by the grant from the National Natural Science Foundation of China under Grant No. 60972077, partially under

Grant No. 70921061, the Foundation of He'nan Educational Committee under Grant No.13A413750, 13A413747, the Natural Science Foundation of He'nan Province of China under Grant No. 122102210543, and Xuchang Municipal Natural Science Foundation under Grant No. 5018.

References

1. Wang, C.-D., Lai, J.-H., Huang, D., Zheng, W.-S.: SVStream: A Support Vector Based Algorithm for Clustering Data Streams. IEEE Transactions on Knowledge and Data Engineering, 1–14 (2011) (in press), doi:10.1109/TKDE.2011.263
2. Wang, C.-D., Lai, J.-H.: Position Regularized Support Vector Domain Description. Pattern Recognition 46(3), 875–884 (2013)
3. Ping, Y., Tian, Y.J., Zhou, Y.J., Yang, Y.X.: Convex Decomposition based Cluster Labeling Method for Support Vector Clustering. Journal of Computer Science and Technology 27(2), 428–442 (2012)
4. Ping, Y., Zhou, Y.J., Yang, Y.X.: A Novel Scheme for Acclereating Support Vector Clustering. Computing and Infomatics 31(6), 613–638 (2012)
5. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison Wesley (2006)
6. Xu, R., Wunsch, D.C.: Clustering. A John Wiley&Sons (2008)
7. Strehl, A., Ghosh, J.: Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions. Journal of Machine Learning Research (3), 583–617 (2002)
8. Lee, S.-H., Daniels, K.M.: Cone Cluster Labeling for Support Vector Clustering. In: Proceedings of 6th SIAM Conference on Data Mining, pp. 484–488. SIAM, Bethesda (2006)
9. Yang, J.H., Estivill-Castro, V., Chalup, S.K.: Support Vector Clustering Through Proximity Graph Modelling. In: Proceedings of 9th International Conference on Neural Information Processing (ICONIP 2002), pp. 898–903. Orchid Country Club, Singapore (2002)
10. Estivill-Castro, V., Lee, I., Murray, A.T.: Criteria on Proximity Graphs for Boundary Extraction and Spatial Clustering. In: Cheung, D., Williams, G.J., Li, Q. (eds.) PAKDD 2001. LNCS (LNAI), vol. 2035, pp. 348–357. Springer, Heidelberg (2001)
11. Puma-Villanueva, W.J., Bezerra, G.B., Lima, C.A.M., Zuben, F.J.V.: Improving Support Vector Clustering with Ensembles. In: Proceedings of International Joint Conference on Neural Networks, Montreal, Quebec, Canada, pp. 13–15 (2005)
12. Tax, D.M.J., Duin, P.R.W.: Support Vector Domain Description. Pattern Recognition Letters 20(11-13), 1191–1199 (1999)
13. Jung, K.-H., Kim, N., Lee, J.: Dynamic Pattern Denoising Method using Multibasin System with Kernels. Pattern Recognition 44(8), 1698–1707 (2011)
14. Jung, K.-H., Lee, D., Lee, J.: Fast Support-based Clustering Method for Large-scale Problems. Pattern Recognition 43(5), 1975–1983 (2010)
15. Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V.N.: Support Vector Clustering. Journal of Machine Learning Research (2), 125–137 (2001)
16. Guo, C.H., Li, F.: An Improved Algorithm for Support Vector Clustering based on Maximum Entropy Principle and Kernel Matrix. Expert Systems with Applications 38(7), 8138–8143 (2011)
17. Ling, P., Zhou, C.-G., Zhou, X.: Improved Support Vector Clustering. Engineering Applications of Artificial Intelligence 23(4), 552–559 (2010)

18. Lee, J., Lee, D.: An Improved Cluster Labeling Method for Support Vector Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(3), 461–464 (2005)
19. Lee, J., Lee, D.: Dynamic Characterization of Cluster Structures for Robust and Inductive Support Vector Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(11), 1869–1874 (2006)
20. Lee, J., Lee, D.: Equilibrium-based Support Vector Machine for Semisupervised Classification. IEEE Transactions on Neural Networks 18(2), 578–583 (2007)
21. Lee, J., Jung, K.-H., Lee, D.: Constructing Sparse Kernel Machines Using Attractors. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(4), 721–729 (2009)
22. Li, Y.H., Maguire, L.: Selecting Critical Patterns Based on Local Geometrical and Statistical Information. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(6), 1189–1201 (2011)
23. Hubert, P.A.L.: Compariring partitions. Journal of Classification 2, 193–218 (1985)
24. Veenman, C.J., Reinders, M.J.T., Backer, E.: A Maximum Variance Cluster Algorithm 24(9), 1273–1280 (2002)
25. Frank, A., Asuncion, A.: UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml`