A Fast and Stable Cluster Labeling Method for Support Vector Clustering

Huina Li
Department of Computer Science and Technology, Xuchang University, Xuchang 461000, China
Email: leehuina@126.com

Abstract—Even though support vector clustering (SVC) is able to handle arbitrary cluster shapes effectively, its popularity is frequently degraded by highly intensive time complexity, poor label performance and even instability for efficiency. To overcome such problems, a fast and stable cluster labeling (FSCL) method is proposed. Based on stable equilibrium points, the FSCL first finds an appropriate division of support vectors. With a nonlinear sample sequence strategy presented here, the connected components profiled by support vectors (SVs) can be determined in terms of sampling all stable equilibrium point pairs; and the FSCL prefers a density centroid constructed by one subset of SVs, along with a stable equilibrium point, to represent a component while avoiding local optimization. Finally, the remaining data points can be assigned the label of the nearest components with respect to a weighted distance. Time complexity analysis and comparative experiments suggest that the FSCL improves both the efficiency and clustering quality significantly while guaranteeing stability.

Index Terms—support vector clustering, centroid, stable equilibrium point, unsupervised learning method, support vector machine

I. INTRODUCTION

Clustering, focusing on forming natural grouping of data points that maximize intro-cluster similarity and minimize inter-cluster similarity, has been extensively employed in image processing, pattern recognition, data analysis and instance-based learning, etc. Among the previous studies [1], [2], support vector clustering (SVC) [3]–[7], a boundary-based clustering algorithm inspired by the support vector machines (SVM) [8] whose advantage is to generate cluster boundaries of arbitrary shape, is recently emerged algorithm to characterize the support of a high-dimensional distribution.

Given a dataset $X$ with $N$ points \{$x_1, x_2, \ldots, x_N$\}, two main phases are included by SVC to cluster these data points, i.e., SVC training to estimate a support function by solving a dual problem and cluster labeling to assign each data point to its corresponding cluster. Apparently, both of the two phases can be considered as bottlenecks to the SVC’s application since too much time is frequently required. However, along with some insightful techniques employed by the first phase, e.g., chunking [9], problem transferring [10], [11], noise and core points elimination [12], [13], the cluster labeling phase with time complexity of $O(N^2m)$ takes most of the computation time for the entire SVC process. Here, $m$ is the sample rate for each point pair. Thus, it is crucial to make improvement on efficiency of the labeling step, especially for large-scale problems purpose. To achieve this objective, based on the complete graph (CG) strategy [3], a number of insightful works have been done, i.e., the support vector graph (SVG) [4], proximity graph of delaunay (DD) [14], minimum spanning tree (MST), $k$-nearest neighbor (kNN) [15], [16], divide and conquer-based methods [17], [18], cell growth based method [19], cone cluster labeling (CCL) [20], [21], and equilibrium based approaches [22]–[28].

Despite so many variants have been developed, there are still difficulties to achieve efficiency, accuracy and stability. For instance, even though reduced complete graph (R-CG) [22] reached significantly improvement on efficiency, it suffers from too many local minimums, ineffectively dealing with irregular clusters. Then equilibrium based support vector clustering (E-SVC) [23]–[25] was proposed to make further improvement on accuracy, but the pricey time consumption degrades its application. Furthermore, fast support vector clustering (FSVC) [28], a recent insightful variant of SVC, constructs an amount of small balls using the whole dataset, looks for stable equilibrium vectors (SEVs) for connective analysis from the centers of balls separately. Despite of time consumption reduced significantly, FSVC suffer from instabilities mainly due to the assumption of circular distribution of clusters. As another achieved method, a very fast calculation of adjacency matrix of SVs can be made by CCL. Unfortunately, a strict constraint condition to guarantee the radius of hypersphere to be lower than 1 seriously increases its time complexity of solving dual problem.

Actually, three critical issues should be taken into account, i.e., redundant initial data points should be avoided for efficiency while the SEPs are to be located, appropriate cluster prototype finding method with suitable sampling strategy are required for accuracy and the labeling algorithm should be independent of the SVC training phase. In addition, stability is a basis. Consider the aforementioned methods, a fast and stable cluster labeling (FSCL) algorithm is proposed in this paper with...
main contributions lie in three aspects:

1) To reduce the account of sampled point pairs, only the SVs are selected to locate the SEVs before finding connected components since they do profile clusters accurately.

2) For efficiency, the work of finding the connected components is done between the SV pairs. More importantly, the average sample rate is significantly reduced according to a novel proposed strategy of disconnection checking first.

3) To assign the remaining data points correctly, a pair of data points, i.e., a SEV and a density centroid, are constructed to achieve a better support of any shape and distribution of a component. Then the remaining data points could be labeled following the principle of maximum subordinated degree among the components with a weighted distance in the input space. Time complexity analysis and comparative experiments with the state-of-the-art methods suggest that the FSCL improves both the efficiency and clustering quality significantly. Moreover, it is stable.

The remainder of this paper is arranged as follows. We formally introduces the framework of the FSCL in Section II. Compared with the traditional ones, Section III details the evaluations with respect to accuracy and efficiency. Finally, the last section draws conclusions for this study with the future work.

II. THE FAST AND STABLE CLUSTER LABELING ALGORITHM

To well describe the proposed FSCL algorithm, three essential phases with novel strategies, i.e., decomposition of SVs by SEVs, connectivity analysis of components and cluster assignments for non-SVs, are adopt to be detailed in this section.

A. Phase I. Partition of SVs

Suppose that the set of SVs are denoted by \( V( V \subseteq \mathcal{X} ) \), from the principle of SVC, the image of any SV, e.g., \( \Phi( v ) ( v \in V ) \), lies on the surface of the feature space sphere and has an unique distance \( R \) to the hypersphere’s center \( \alpha \). Here \( \Phi(\cdot) \) is a nonlinear transformation. That is to say the distance from any data point lies in the profile to \( \alpha \) is always lower than \( R \). Following closely the derivation of [22]–[24], [26], to obtain the solution of the minimal hypersphere approximate covering with an appropriate width \( q \) of the kernel function, a gradient dynamical system which is associated with the trained kernel function \( R^2(x) \) can be constructed as follows.

\[
R^2(x) = ||\Phi(x) - \alpha||^2 \tag{1}
\]

\[
\frac{\partial x}{\partial t} = -\nabla R^2(x) \tag{2}
\]

In Eq.(1), there exists a unique solution (or trajectory) \( x(t) : R \rightarrow R^n \) for each condition \( x(0) = x_0 \) is guaranteed since it is twice differentiable and the norm of \( \nabla R^2(x) \) is bounded. Moreover, a SEV \( \bar{x} \) can be obtained if the equation \( \nabla R^2(x) = 0 \) and all the eigenvalues of its corresponding Jacobian matrix, \( J_f(x) \equiv \nabla^2 R^2(x) \), are positive. Since the cluster boundaries consist of SVs, in order to achieve improvement on efficiency, only SVs are selected to get a unique partition of SVs that satisfies the Theorem 1.

**Theorem 1:** In terms of the gradient dynamic system (2), the dataset of SVs \( V \) can be partitioned into several separate and non-overlapped subsets with respect to SEVs, i.e.,

\[
V = \bigcup_{i=1}^{N_c} V_i \tag{3}
\]

\[
\text{s.t.} \quad \lim_{t \rightarrow \infty} v_i(t) = \overline{x}_{V_i}, \quad V_i \cap V_j = \emptyset, \quad \forall i, j \in [1, N_c], i \neq j
\]

where \( \overline{x}_{V_i} \) is the SEV corresponding to \( V_i \), \( N_c \) and \( N_i \) are the number of subsets and SVs in \( V_i \) respectively.

**Proof:** Proved by Ref. [22], [28], [29], all the data points can be grouped by different convergence objects which are the local minimum positions reached from the points with respect to the dynamic system. Furthermore, the local minimum position is unique for each point, so the decomposed data sets are non-overlapping.

B. Phase II. Connectivity Analysis of Components

In geometry, the region whose boundary is built up by \( V_i \) is considered as a component (denoted by \( C_i \)) for further connectivity analysis. Since these SEVs locate inside components with a local minimum distance to the center \( \alpha \) of hypersphere while being mapped into feature space, naturally, the nearer a data point is away from a SEV, the less impossible it locates outside a component. Thus, for efficiency, a disconnection checking first strategy is presented in this section.
SEVs (i.e., $S_1, S_2, S_3, S_4$) have been found, the connection status of their corresponding components can be checked by sampling line segments connecting each SEV pair, e.g., $S_1S_2$, $S_1S_7$, etc. Generally, the conventional methods [3], [4], [13], [19], [22]–[25], [27], [28] prefer a linear sample sequence from one side to the other, for instance, either $S_1S_2'$ or $S_1S_2''$. Apparently, redundant segments can not be avoided, e.g., segments on line segments $S_1X$ and $S_2B$. Therefore, a nonlinear sample sequence generated by a disconnection checking first strategy, shown in Fig.1(b), is proposed in a simple but rather more effective way. Consider the line segment connecting two SEVs (i.e., $S_1S_2$), it is split into $m$ segments, for instance, $m = 1$ while $m − 1$ points are expected to be checked. Among these sample points, the center of the line segment (i.e., 5) is selected as initial point and to be check first; then the distance from each sample point to the initial point is increase gradually. In the worst case, the full sample sequence is $\{5, 6, 4, 7, 3, 8, 2, 9, 1\}$. If two components are strongly disconnected or not connected directly like that of corresponding to $S_1S_2$, only one sample point might be sufficient for giving the final decision correctly.

C. Phase III. Cluster assignments for non-SVs

Generally, the remaining data points would be labeled by their nearest neighboring SVs or SEVs. However, since the locations of SVs are determined by the parameters [4], i.e., kernel width $q$ and penalty factor $C$, and the SEVs usually are at the center of their corresponding components’ profiles, they are not sufficient for judging a data point with relatively balanced membership grades to multiple neighboring components. So, in addition to SEV, inspired by Ref. [30], a kind of density centroid is defined for representing a component.

Definition 1 (density centroid): The density centroid (denoted by DC) is a logic data point defined by formula (4) for a component:

$$DC(C_i) = \frac{1}{N_i} \sum_{i=1}^{N_i} v_i, \quad \forall v_i \in V_i$$ (4)

where $N_i$ is the number of SVs in subset $V_i$ w.r.t. component with SEV $S_i$.

Actually, the density centroid get close to the highest density position of a component but usually not overlap with the SEV. The greater the gap of the distance between them, the much more imbalanced distribution for the component is. Consider unequal contributions from them, a simple linear programming problem (Eq.(5)) is constructed and solved by a series of SVs to quantify the weight of each points.

$$\min \frac{1}{N_i} \sum_{i=1}^{N_i} (W_{S_i} \parallel x_i - S_i \parallel^2 + W_{DC(C_i)} \parallel x_i - DC(C_i) \parallel^2)$$ s.t. $W_{S_i} + W_{DC(C_i)} = 1$ $W_{S_i} \geq \frac{1}{2}$ $W_{DC(C_i)} \geq 0$ (5)

In Eq.(5), $W_{S_i}$ and $W_{DC(C_i)}$ are the weight of contributions from SEV $S_i$ and density centroid $DC(C_i)$ in $C_i$, respectively. Notice that $W_{S_i} \geq \frac{1}{2}$ is set to emphasize the contribution of SEVs. Based on the extracted weights, a number of weighted norm distances from each remaining data point $x$ to component $C_i(i \in [1, N_c])$ could be achieved; and then $x$ would be assigned with the label of its nearest component with the minimum weighted norm distance detailed by Eq.(6). In order to get the two euclidean distances between $x$ and any component, the computation taken in the input space is recommended for efficiency.

$$\text{label}(x) = \text{label}(\arg \min_{S_i} (W_{S_i} \parallel x - S_i \parallel^2 + W_{DC(C_i)} \parallel x - DC(C_i) \parallel^2))$$ (6)

D. Implementation and Time Complexity

Algorithm 1 shows the FSCL method. In line 1, for the given $q$ and $C$, it collects the SVs by solving dual problem. Then $V$ is split while their corresponding SEVs (denoted by $S_{SEVs}$) are obtained according to LocateSEVsbySVs($S_{SEVs}$) detailed in section II-A. In line 3, the adjacency matrix $A$ is got by CheckConninSEVs($S_{SEVs}$) and line 4 returns the labeled clusters. After that, lines 5~11 label the remaining data points following the way described in section II-C.

Algorithm 1. FSCL($X, q, C$)

| Input: | dataset $X$, Gaussian kernel width $q$ and the penalty term $C$ |
| Output: | labels for all the data points |
| 1 | collect $V$ for $q$ by solving dual problem |
| 2 | $S_{SEVs} \leftarrow$ LocateSEVsbySVs($V$) |
| 3 | $A \leftarrow$ CheckConninSEVs($S_{SEVs}$) |
| 4 | $\text{Labels} \leftarrow$ FindConnComponents($A, V$) |
| 5 | collect DCs by Eq.(4) for each component $C_i$ |
| 6 | calculate $W_{S_i}$ and $W_{DC(C_i)}$ for $C_i$ by Eq.(5) |
| 7 | for each $x \in X \setminus V$ |
| 8 | $\text{inx} \leftarrow$ find the nearest component by Eq.(6) |
| 9 | $\text{Labels}[^x] \leftarrow$ Labels[$C_{\text{inx}}$] |
| 10 | end |
| 11 | return Labels |

To analyze the time complexity of the proposed method, let $N$ be the number of data points in a dataset, $N_{SV}$ be the number of SVs, $l$ be the average number of iterations for each data point to locate its corresponding local minimum via the steepest decent process [22], $N_{SEV}$ is the number of SEVs and $m$ be the sample rate. Apparently, the time cost by the three phases are $O(lN_{SV})$, $mN_{SEV}^2$ and $O((N - N_{SV})N_{SEV})$ respectively. Therefore, the time complexity of FSCL is $O(lN_{SV} + lN_{SEV})$. In comparison of FSCL, the time complexities of the state-of-the-art algorithms are also listed in Table I where $f(N)$ is a function of $N$ for the DD algorithm, $k$ is the number of nearest neighbors ($k = 4$ is preferred) used by the $k$NN, $N_k$ is the number of small balls extracted from the dataset and $\gamma$ ranges from $\frac{1}{2}$ to 1.
TABLE I.: Time complexity.

<table>
<thead>
<tr>
<th>Index</th>
<th>Method</th>
<th>Time complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CG</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>2</td>
<td>DD</td>
<td>$O(n \log n + mf(n))$</td>
</tr>
<tr>
<td>3</td>
<td>kNN</td>
<td>$O(n \log n + mkN)$</td>
</tr>
<tr>
<td>4</td>
<td>MST</td>
<td>$O(n \log n + mN)$</td>
</tr>
<tr>
<td>5</td>
<td>R-CG</td>
<td>$O(n + mN_{SEV})$</td>
</tr>
<tr>
<td>6</td>
<td>E-SVC</td>
<td>$O(n + mN_{SEV} + 2N_{SEV})$</td>
</tr>
<tr>
<td>7</td>
<td>CCL</td>
<td>$O(N_{SEV})$</td>
</tr>
<tr>
<td>8</td>
<td>FSVC</td>
<td>$O(N_{SEV} + \gamma N^2)$</td>
</tr>
<tr>
<td>9</td>
<td>FSCL</td>
<td>$O(N_{SEV} + N N_{SEV})$</td>
</tr>
</tbody>
</table>

TABLE II.: Data Description.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>dataset description</th>
<th>dims</th>
<th>size</th>
<th># of classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>sunflowers</td>
<td></td>
<td>2</td>
<td>200</td>
<td>9</td>
</tr>
<tr>
<td>orange</td>
<td></td>
<td>2</td>
<td>140</td>
<td>9</td>
</tr>
<tr>
<td>twocircles</td>
<td></td>
<td>2</td>
<td>300</td>
<td>2</td>
</tr>
<tr>
<td>five-Gaussians</td>
<td></td>
<td>2</td>
<td>1000</td>
<td>5</td>
</tr>
<tr>
<td>iris</td>
<td></td>
<td>4</td>
<td>150</td>
<td>3</td>
</tr>
<tr>
<td>wisconsin</td>
<td></td>
<td>9</td>
<td>683</td>
<td>2</td>
</tr>
<tr>
<td>wine</td>
<td></td>
<td>13</td>
<td>178</td>
<td>3</td>
</tr>
<tr>
<td>zoo</td>
<td></td>
<td>16</td>
<td>101</td>
<td>7</td>
</tr>
<tr>
<td>movement_libras</td>
<td></td>
<td>90</td>
<td>360</td>
<td>15</td>
</tr>
<tr>
<td>P2PTraffic</td>
<td></td>
<td>4</td>
<td>9206</td>
<td>4</td>
</tr>
</tbody>
</table>

III. EXPERIMENTS AND ANALYSIS

A. Data Corpora

To demonstrate the effectiveness and performance of the proposed method, the comparative evaluations are taken on various datasets: five-Gaussians, and twocircles are widely used in the literatures [28], [29], [31], [32], iris, wisconsin, wine, zoo and movement_libras are from UCI repository [33], and P2PTraffic from Ref. [34]. All the datasets are described in Table II.

B. Benchmark methodology

Two series of simulations are conducted with Core dual 2.66 GHz and 3GB memory size machine. To evaluate both the efficiency and accuracy, we conduct the first series of experiments on the ten datasets and employ two phases of time consumption along with adjusted rand index (ARI, denoted by $R_{adj}$) [1] [35] which is widely used similarity measure between two data partitions where both true labels and predicted cluster labels are given. The second series of experiments is to verify if the proposed disconnection checking first strategy contributes to efficiency.

C. Comparisons of Benchmark Datasets

In Table III, we include SVC training time, cluster labeling time, the number of SEVs, the number of clusters and $R_{adj}$ achieved by the evaluated algorithms separately. Rank of each item highlighted by boldface with superscript is given depending on its performance followed by corresponding rank (from 1 to 3).

In terms of ARI measure, except for wisconsin and zoo, FSCL achieves the best performance. More than this, it always reaches the first three rank in each evaluation

<table>
<thead>
<tr>
<th>Data Method</th>
<th>$N_{SEV}$ Training(s)</th>
<th>Labeling(s)</th>
<th>$R_{adj}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG</td>
<td>2.6949</td>
<td>86.3893</td>
<td>17(5)</td>
</tr>
<tr>
<td>DD</td>
<td>3.0631</td>
<td>22.5692</td>
<td>27(5)</td>
</tr>
<tr>
<td>kNN</td>
<td>4.4565</td>
<td>71.1178</td>
<td>23(5)</td>
</tr>
<tr>
<td>MST</td>
<td>1.8702</td>
<td>379.6909</td>
<td>5(5)</td>
</tr>
<tr>
<td>R-CG</td>
<td>3.0924</td>
<td>10.8926</td>
<td>17(5)</td>
</tr>
<tr>
<td>E-SVC</td>
<td>3.0924</td>
<td>1079.6013</td>
<td>18(5)</td>
</tr>
<tr>
<td>CCL</td>
<td>300.4796</td>
<td>413.6639</td>
<td>19(5)</td>
</tr>
<tr>
<td>FSVC</td>
<td>3.1415</td>
<td>1.0443</td>
<td>17(5)</td>
</tr>
<tr>
<td>FSCL</td>
<td>1.6934</td>
<td>6.0467</td>
<td>17(5)</td>
</tr>
</tbody>
</table>

TABLE III.: Benchmark results on benchmark datasets.

Note: “—” means not available.
whereas performance of the others are unstable. In most cases, \(k\)NN, CCL and FSVC perform well, but the cluster labeling time increases dramatically as the dimensionality and size of a data set increases. Especially for FSVC, due to the drawback of preprocess which is similar to \(k\)-means [36], [37], it suffers from instabilities. Another time-consuming algorithm is E-SVC, which fails in handling the 90-dimensional dataset movementLibras. Notice that DD can hardly clusters such high-dimensional data since it can not construct the graph; therefore, it is only recommended to deal with low-dimensional data. For time consumption, SVC training and labeling are considered separately. In the view of methodology, CCL requires strict constraints to guarantee \(R \leq 1\) so that it frequently consumes much more time in SVC training; whereas the others cost similarly. In labeling phase, FSCL is consistently included in the first three ranks in most cases. Intuitively, an effective algorithm is expected to achieve a high accuracy with a corresponding number of clusters close to the exact number of classes. In table III, the \(N_R\) denotes the number of clusters got by these algorithms while the \(N_C\) is the exact number of classes summarized in table II. The highlighted \(N_R\) by boldface which is closest to \(N_C\) also suggests that FSCL is suitable for exploring the distribution structure of a data set.

To verify if the proposed sampling strategy reduces the sample rate \(m\) required by finding the connected components, we illustrate the average sample rate required by the FSCL and CG like algorithms respectively in Fig.2. Obviously, the significantly reduced average sample rates proves the effectiveness of the proposed strategy.

IV. Conclusion

In this paper, a novel fast and stable cluster labeling algorithm, namely FSCL is proposed to improve efficiency and accuracy while keeps stability. Differing from the traditional cluster labeling algorithms, we consider to improve the accuracy as well as to reduce the time complexity by decreasing both the number of sampled point pairs and the sample rate. We find that the SVs are sufficient to profile the cluster boundaries. Such that taking all of the data points to locate the SEVs is completely unnecessary and time consuming. Furthermore, while a reduced set of SEVs integrated from the SVs are obtained, the cost for finding connected components is significantly reduced. Meanwhile, in this study, we also confirm that the linear sample sequence generated by tradition algorithms would bring a quantity of redundant segments to connectivity analysis. Therefore a disconnection checking first strategy is presented to achieve further improvement on efficiency. Even though a cluster might have multiple components to support irregular shapes, one component with one SEV is unable to handling imbalanced distributed data points well. Consider this problem, the density centroid is constructed by each subset of SVs with respect to a component; then two points used as prototype for a component in logical contribute to improvements on accuracy remarkably. Since no instable factor is brought in, the proposed FSCL algorithm is suggested to be fast and stable in comparison of the state-of-the-art algorithms.

Since the number of SVs would be one of the major factors related to efficiency, a more efficient strategy which can control the size of SVs while avoid unnecessary analysis with all of the data points is required to be further integrated.

ACKNOWLEDGMENT

The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper.

REFERENCES


Huina Li received her B.S. degree in electronics and information engineering from Central China Normal University, China in June 2003, and his M.S. degree in mathematics from the Henan University, Henan in June 2010. She is currently working at Xuchang University, China. His current research interest includes machine learning, image processing.