Support Vector Clustering Combined with Spectral Graph Partitioning

JinHyeong Park[1], Xiang Ji[1], Hongyuan Zha[1] and Rangachar Kasturi [2]

[1]Dept. of Computer Science and Engineering, The Pennsylvania State University,
[2]Dept. of Computer Science and Engineering, University of South Florida,
[1]{jhpark,xji,zha}@cse.psu.edu, [2]chair@csee.usf.edu

Abstract

In this paper, we propose a new support vector clustering (SVC) strategy by combining (SVC) with spectral graph partitioning (SGP). SVC has two main steps: support vector computation and cluster labeling using adjacency matrix. Spectral graph partitioning (SGP) method is applied to the adjacency matrix to determine the cluster labels. It is feasible to combine multiple adjacency matrices computed using different parameters. A novel multi-resolution combination method is proposed for cluster labeling using the SGP for the purpose of boosting the clustering performance.

1. Introduction

SVMs are powerful pattern recognition techniques and have been successfully applied to many machine learning tasks such as classification [1] and regression [2]. SVMs have outperformed many other machine learning methods such as artificial neural networks and k-nearest neighbors and attracted a great deal of attention from the machine learning community because of many desirable properties, including good generalization performance, robust noise performance and fast convergence. Although SVMs have been widely recognized as supervised learning techniques, they have recently been adapted for unsupervised learning such as novelty detection [3] and cluster analysis [4, 5]. With the support vector clustering method, data points are first mapped to a high dimensional feature space. Then a hypersphere with a minimum radius R and center $\vec{a}$ is searched to enclose most of the data points in the new feature space. When this hypersphere is mapped back to the original data space, its surface forms a set of closed contours enclosing the data points. These contours correspond to the clustering boundaries which are defined by the support vectors. To assign data points to individual clusters, a straightforward geometric approach was used. Each pair of data points are defined either as adjacent or not depending on whether the line segment connecting the pair of data points goes outside of the enclosing sphere or not. Clusters are then defined as the connected components of the graph induced by the adjacency matrix between pairs of points.

Despite the many advantages the support vector clustering method has, its time complexity is very high. The algorithm consists of two major steps: support vector learning and cluster assignment. Suppose that the number of data points is n and M points are sampled for each line segment during the cluster assignment. The complexity of cluster assignment step is $O(n^2M)$ which is much higher than that of support vector learning. Therefore, reducing the computation of the cluster assignment step is the crucial issue in support vector clustering. Yang et al. [5] has proposed the constructing of a proximity graph to model a data set. This method avoids redundant checks in a complete graph and successfully reduced the computation cost in cluster assignment to a certain extent. However, other problems still exist in the cluster labeling stage. When an adjacency matrix is constructed, and a sampling strategy is employed for checking whether the line segment connecting two data points goes outside the enclosing sphere. It is possible that some data points outside the enclosing sphere may be skipped. As a result, two data points might be denoted as adjacent to each other and given the same class label when the connected component approach [4, 5] is employed.

In this paper, we propose a support vector clustering method using spectral graph partitioning (SGP) for the cluster label assignment We combine several adjacency matrices to produce a new adjacency matrix, and apply SGP to it for cluster label computation. It can be considered that the σ in the Gaussian kernel function plays a role of yielding multi-resolution results for SVC: a small σ produces high resolution (detail) contours while a larger σ produces low resolution (smoothing) contours. When we combine several adjacency matrices, we select them to represent different resolutions, from coarse to fine. The method improves the accuracy of cluster assignment and avoid the merging of clusters in the same spirit of wavelet convolution.

2. Support Vector Clustering using Kernel Technique

2.1 Support vector analysis using Kernel

Given a set of data points $X = \{x_i\}, i = 1, 2, .., n$, where $x_i \in \mathcal{R}^d$, a nonlinear mapping Φ transforms the data points to some high dimensional feature space, and we search for the smallest hypersphere that contains most data points. Mathematically, we can express the problem as:

$$\min_{\vec{a},\, R,\, \xi_j} R, \quad \text{subject to } ||\Phi(x_j) - \vec{a}||^2 \leq R^2 + \xi_j \;\; \forall j. \quad (1)$$

We can construct the Lagrangian to solve the problem:

$$\begin{aligned} L = R^2 \quad &- \quad \sum_j (R^2 + \xi_j - ||\Phi(x_j) - \vec{a}||^2)\beta_j \\ &- \quad \sum \xi_j \mu_j + C \sum \xi_j, \end{aligned} \quad (2)$$

where $\beta_j \geq 0$ and $\mu_j \geq 0$ are Lagrange multipliers, C is a constant, and $C \sum \xi_j$ is a penalty term for the outliers. Applying KTT complementarity conditions the problem can be reformed as:

$$\max L = \sum_i \beta_i \Phi(\vec{x_i})^2 - \sum_{i,j} \beta_i \beta_j \Phi(\vec{x_i}) \cdot \Phi(\vec{x_j}),$$

$$\text{subject to } 0 \leq \beta_i \leq C, \; \sum \beta_i = 1, \; i = 1, \ldots, n(3)$$

If $\beta_i = C$, then x_i is a bounded support vector or BSV, which lies outside of the hypersphere and is treated as noise. If $0 < \beta_i < C$, then x_i is a support vector or SV, which lies on the surface of the hypersphere and thus is on cluster boundaries. For x_i with $\beta_i = 0$, it is inside the hypersphere.

Following the SVMs methods, a kernel representation $K(\vec{x_i}, \vec{x_j}) = \Phi(\vec{x_i}) \cdot \Phi(\vec{x_j})$ is adopted and Eq. (3) is rewritten as:

$$\max L = \sum_i \beta_i K(\vec{x_i}, \vec{x_i}) - \sum_{i,j} \beta_i \beta_j K(\vec{x_i}, \vec{x_j}),$$

$$\text{subject to } 0 \leq \beta_i \leq C, \; \sum \beta_i = 1, \; i = 1, \ldots, n(4)$$

The kernel methods do not require an explicit calculation of the feature map Φ but only use the values of the dot products between mapped patterns. For clustering purpose, Gaussian kernels $K_q(\vec{x_i}, \vec{x_j}) = e^{-\frac{||\vec{x_i} - \vec{x_j}||^2}{2\sigma^2}}$ are generally used. The clustering level can be controlled by the width parameter of the Gaussian kernel (σ). When σ decreases, the number of disconnected contours in the data space increase, which leads to an increasing number of cluster. The distance from the center of the hypersphere to the image of a point x in the feature space can be calculated as:

$$R^2(\vec{x}) = K(\vec{x}, \vec{x}) - 2 \sum_j \beta_j K(\vec{x_j}, \vec{x}) + \sum_{i,j} \beta_i \beta_j K(\vec{x_i}, \vec{x_j}) \quad (5)$$

The radius R of the sphere is the distance between the hypersphere center and the support vectors:

$$R = \{R(\vec{x_i}) | \; x_i \text{ is a support vector}\} \, . \quad (6)$$

Cluster boundaries are formed from the set of points in data space $\{\vec{x} | \; R(\vec{x}) = R\}$. Therefore, SVs are on cluster boundaries, BSVs are outside the clusters, and other points are inside the clusters.

2.2 Cluster Labeling

The current cluster labeling method generally employees a straightforward geometric approach to define their assignment to clusters based on the following observation: given a pair of data points in different clusters, any path that connects them must exist from the hypersphere.

This is based on the observation that an adjacency matrix between pairs of points is defined as:

$$A_{ij} = \begin{cases} 1 & \text{if, R}(x_i + \lambda(x_j - x_i)) \leq R, , \forall \lambda \in [0, 1] \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

The adjacency matrix A can be constructed based on a complete graph (CG) or a proximity graph[5] such as the Delaunay Diagram(DD), Minimum Spanning Tree(MST) or k-nearest neighbors (k-NN). In the CG-based strategy, matrix A is computed using all the pairs of x_i and x_j, while the proximity graph strategy calculate A_{ij}'s only for pairs of x_i and x_j when they are linked by an edge. The second strategy reduces time complexity significantly compared with the first one. After constructing the matrix A, clusters are defined as the connected components of the graph A.

3. Cluster Labeling using SGP

3.1 Spectral Multi-way Graph Partitioning

We consider the absolute value of the (i, j) element of an adjacency matrix $\mathbf{A}$ as a measure of the similarity of feature points i and j with feature points belonging to the same cluster more similar than those of other points. Our goal is then to partition the feature points into S clusters so that feature points are more similar within each cluster than across different clusters. Let $A = (a_{ij})$ with $a_{ij} = |\mathbf{A}_{ij}|$. For a given partition of the feature points into S groups, we can permute the rows and columns of A so that rows and columns corresponding to the feature points belonging to the same objects are adjacent to each other (i.e., we can re-order the columns and rows of the A matrix accordingly such that $A = \{A_{ij}\}_{i,j=1}^{S}$)

We want to find a partition such that A_{ii} will be large while $A_{ij}, i \neq j$ will be small, and to measure the size of a sub-matrix matrix A_{ij} we use the sum of all its elements and denoted as $\sum(A_{ij})$. Let x_i be a cluster indication vector accordingly partitioned with that of A with all elements equal to zero except those corresponding to rows of A_{ii},

$$x_i = [0 \cdots 0 \;\; 1 \cdots 1 \;\; 0 \cdots 0]^T.$$

Denote $D = diag(D_1, D_2, \cdots, D_S)$ such that $D_i = \sum_{j=1}^{S} A_{ij}$. Since we want to find a partition which will maximize $\sum(A_{ii})$ while minimizing $\sum(A_{ij}), i \neq j$, we seek to minimize the following objective function by finding a set of indicator vectors x_i.

$$\begin{aligned} MCut &= \frac{x_1^T(D-A)x_1}{x_1^T A x_1} + \cdots + \frac{x_S^T(D-A)x_S}{x_S^T A x_S} \\ &= \frac{x_1^T D x_1}{x_1^T A x_1} + \cdots + \frac{x_S^T D x_S}{x_S^T A x_S} - S. \end{aligned}$$

If we define $y_i = D^{1/2}x_i/||D^{1/2}x_i||_2$ and $Y_S = [y_1, \cdots, y_S]$, we have

$$MCut = \frac{1}{y_1^T \hat{A} y_1} + \frac{1}{y_2^T \hat{A} y_2} + \cdots + \frac{1}{y_S^T \hat{A} y_S} - S \quad (8)$$

where $\hat{A} = D^{-1/2}AD^{-1/2}$. It is easy to see that the y_i are orthogonal to each other and normalized to have Euclidean norm one. If we insist that the y_i be constrained to inherit the discrete structure of the indicator vectors x_i, then we will be solving a combinatorial optimization problem which has been proved to be NP-hard even when $S = 2$ [6]. The idea of spectral clustering is to relax these constraints which allows the y_i to be an arbitrary set of orthonormal vectors. In this case, the minimum of Eq. 8 can be shown to be achieved by an orthonormal basis $y_1, \cdots, y_S$ of the subspace spanned by the eigenvectors corresponding to the largest S eigenvalues of $\hat{A}$. Next we discuss how to assign the feature points to each of the clusters based on the eigenvectors and **QR** decomposition.

3.2 Clustering Labeling using QR Decomposition

Here we follow the approach proposed in [7]. Denote $\hat{Y} = [\hat{y}_1, \cdots, \hat{y}_S]^T$ as the optimal solution of Eq 8. The vectors $\hat{y}_i$ can be used for cluster assignment because $\hat{y}_i \approx D^{1/2}\hat{x}_i/||D^{1/2}\hat{x}_i||_2$, where $\hat{x}_i$ is the cluster indicator vector of $i-th$ cluster. Ideally, if A is partitioned perfectly into S clusters, then, the columns in $\hat{X} = [\hat{x}_1, \cdots, \hat{x}_S]^T$ of the $i-th$ cluster are the same, one for the $i-th$ row and zeros for the others. Two columns of different clusters are orthogonal each other. This property is approximately inherited by $\hat{Y}$: two columns from two different clusters are orthogonal to each other, and those from one cluster are the same. We now pick a column of $\hat{Y}$ which has the largest norm and say it belongs to cluster i. We have orthogonalized the rest of the columns of $\hat{Y}$ against this column. We assign the columns to cluster i whose residual is small. We then perform this process S times. As discussed in [7], it is exactly the same procedure for **QR** decomposition with column pivoting applied to $\hat{Y}$. In particular, we compute **QR** decomposition of Y^T with column pivoting

$$Y^T E = \hat{Q}R = \hat{Q}[R_{11}, R_{12}]$$

where $\hat{Q}$ is a $S \times S$ orthogonal matrix, R_{11} is a $S \times S$ upper triangular matrix, and E is a permutation matrix. Then we compute a matrix $\hat{R}$ as

$$\hat{R} = R_{11}^{-1}[R_{11}, R_{12}]E^T = [I_S, R_{11}^{-1}R_{12}]E^T, \quad (9)$$

The matrix $\hat{R} \in R^{S \times N}$ can be considered as giving a level of confidence to a point that is to be assigned to each cluster. Notice that the columns correspond to the feature points and the rows correspond to the clusters. The cluster membership of each feature point is determined by the row index of the largest element in absolute value of the corresponding column of $\hat{R}$.

3.3 Multi-Resolution Combination (MRC)

It can be considered that the σ in the Gaussian kernel function plays a role of yielding multi-resolution results for SVC: a small σ produces high resolution (detail) contours while a larger σ produces low resolution (smoothing) contours. We propose a multi-resolution combination method using SGP to boost clustering performance in the same spirit of wavelet convolution . Several adjacency matrices are computed using different σ's which cover high resolution to low resolution of the feature space contours. We simply make a combinatorial adjacency matrix by linear combination of the adjacency matrices. Finally, the matrix is applied to SGP algorithm to assign cluster labels.

4. Experiment Results

We performed experiments using a 2D synthetic data set (150 points, 5 clusters). Fig. 1 illustrates comparison of the results between the complete graph (CG) approach ($1st$ row) and the K-NN approach (2^{nd} row), and the comparison results among connected component (CC) labeling scheme ((a)&(d)), SGP labeling scheme ((b)&(e)), and MRC labeling scheme ((e)&(f)).

In Fig. 1, each black dot of the connected component based labeling approach represents an independent cluster.

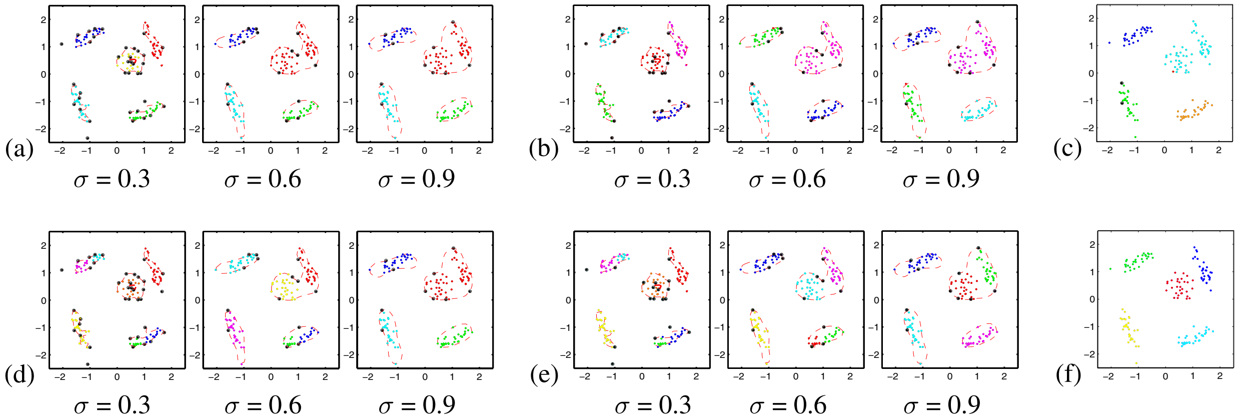

Figure 1. SVC Clustering Results. Each color represents a cluster but black dots represent cluster fragments. (a)∼(c): complete graph based SVC. (d)∼(f): K-NN based SVC. (a)&(d): Connected Component based approach. (b)&(e): Spectral Graph Partitioning based approach. (c)&(f): Multi-Resolution Combination approach (σ=0.3,0.6 and 0.9).

The black dots of the SGP based labeling denote points whose corresponding column values of the matrix $\hat{R}$ (Eq 9) are close to zero. As explained in Sec 3.2, column values of $\hat{R}$ represent the levels of confidence of the points to be assigned to each cluster. If a point has almost zero level of confidence for all clusters, the point is not assigned to any cluster and considered as a cluster fragment. These points are generated if the number of graph fragments in an adjacent is much greater than that of the given cluster number.

As depicted in Fig 1, the CC approach and K-NN approach yield very similar results. As far as the labeling method is concerned, the SGP based method (single resolution) performs very similar compared to the CC-based labeling. However, when we compare (d)-σ=0.9 (CC) and (e)-σ=0.9(SGP), the SGP approach (single resolution information is used) can divide the points into 5 clusters while the CC approach merges the right upper two clusters. The MRC method, which combines three adjacency matrices computed using σ=0.3, 0.6 and 0.9, yields better results than the other two methods. Especially, MRC labeling with the K-NN approach successfully divides all the points into 5 clusters without cluster fragments.

5. Conclusion

We have proposed a new MRC cluster labeling method, which bases on SVC along with spectral graph clustering. The method allows us to combine several adjacency matrices which cover low resolution to high resolution of SVC contours. It is shown in the experimental results that the proposed MRC cluster labeling method yields better performance than the CC approach used in [4, 5]. The CC approach is very sensitive to SVC parameters such as σ in Gaussian function and C in Eq. 1. The proposed MRC method is less sensitive to the parameters than CC approach by producing better performance. We belive that is because the method reflects the same spirit of wavelet convolution.

References

[1] B. Scholkopf, Ch. Burges, and A. Smola, editors. *Advances in Kernel Methods: Support Vector Learning*. The MIT Press, Cambridge, London, 1999.

[2] A. Smola and B. Scholkopf. A tutorial on support vector regession. *Statistics and Computing*, 2000

[3] B. Scholkopf, R. C. Williamson, A. J. Smola, J. Shawe-Talyor, and J.C. Platt. Support vector method for novelty detection. in *Advances in Neural Information Processing Systems 12:* Proceedings of the 1999 conference. Sara A. Solla, Todd K. Leen and Klaus-Robert Muller eds., pp. 582-588. MIT Press, 2000.

[4] A. Ben-Hur, D. Horn, H. T. Siegelmann, V. Vapnik. Support vector clustering. *Journal of Machine Learning Research* 2:125-137, 2001.

[5] J. Yang, V. Estivill-Castro, and S. K. Chalup. Support vector clustering through proximity graph modelling. *Special Session on Support Vector machines: 9th International Conference on Neural Information Processing ICONIP 2002*. November 18-22, 2002, Singapore.

[6] J. Shi and J. Malik. Normalized cut and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[7] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering. In *Neural Information Processing Systems 14*, pages 1057–1064, 2002.