Rough Core Vector Clustering

CMB Seshikanth Varma, S. Asharaf, and M. Narasimha Murty*

Computer Science and Automation,
Indian Institute of Science, Bangalore-560012
Tel.: +91-80-22932779
mnm@csa.iisc.ernet.in

Abstract. Support Vector Clustering has gained reasonable attention from the researchers in exploratory data analysis due to firm theoretical foundation in statistical learning theory. Hard Partitioning of the data set achieved by support vector clustering may not be acceptable in real world scenarios. Rough Support Vector Clustering is an extension of Support Vector Clustering to attain a soft partitioning of the data set. But the Quadratic Programming Problem involved in Rough Support Vector Clustering makes it computationally expensive to handle large datasets. In this paper, we propose Rough Core Vector Clustering algorithm which is a computationally efficient realization of Rough Support Vector Clustering. Here Rough Support Vector Clustering problem is formulated using an approximate Minimum Enclosing Ball problem and is solved using an approximate Minimum Enclosing Ball finding algorithm. Experiments done with several Large Multi class datasets such as Forest cover type, and other Multi class datasets taken from LIBSVM page shows that the proposed strategy is efficient, finds meaningful soft cluster abstractions which provide a superior generalization performance than the SVM classifier.

1 Introduction

Several domains that employ cluster analysis deal with massive collections of data and hence demands algorithms that are scalable both in terms of time and space. Some of the algorithms that have been proposed in the literature for clustering large data sets are DB-SCAN[1], CURE[2], BIRCH[3], etc. Another major concern in data clustering is whether the clustering scheme should produce a hard partitioning (crisp sets of patterns representing the clusters) of the data set or not. Yet another concern is that the clusters that may exist in any data set may be of arbitrary shapes. We can say that there is an intrinsic softness involved in cluster analysis.

Several Rough Set[4] based algorithms like Rough Kmeans[5], Rough Support Vector Clustering (RSVC)[6] etc have been proposed for achieving soft clustering. RSVC is a natural fusion of Rough sets and Support Vector Clustering[7] paradigm. But the Quadratic Programming(QP) Problem involved in RSVC

* Corresponding author.

A. Ghosh, R.K. De, and S.K. Pal (Eds.): PReMI 2007, LNCS 4815, pp. 304–310, 2007.

makes it computationally expensive(at least quadratic in terms of the number of training points) to handle large datasets. In this paper we propose Rough Core Vector Clustering (RCVC) which is computationally efficient for achieving Rough Support Vector Clustering. The rest of the paper is organized as follows. RSVC is discussed in Section 2. In Section 3 the RCVC technique is introduced. Empirical results are given in Section 4 and Section 5 deals with Conclusions.

2 Rough Support Vector Clustering (RSVC)

RSVC uses a non linear transformation ϕ from data space to some high dimensional feature space and looks for the smallest enclosing rough sphere of inner radius R and outer radius T. Now the primal problem can be stated as

$$\begin{aligned} \min \quad & R^2 + T^2 + \frac{1}{\upsilon m}\sum_{i=1}^{m}\xi_i + \frac{\delta}{\upsilon m}\sum_{i=1}^{m}\xi_i^{'} \\ \text{s.t.} \quad & \|\phi(x_i) - \mu\|^2 \leq R^2 + \xi_i + \xi_i^{'} \\ & 0 \leq \xi_i \leq T^2 - R^2 \qquad \xi_i^{'} \geq 0 \quad \forall i \end{aligned} \tag{1}$$

The Wolfe Dual[8] of this problem can be written as

$$\min \; \sum_{i,j}^{m}\alpha_i\alpha_j K(x_i, x_j) - \sum_{i=1}^{m}\alpha_i K(x_i, x_i) \;\text{ s.t. }\; 0 \leq \alpha_i \leq \frac{\delta}{\upsilon m} \;\forall i, \quad \sum_{i=1}^{m}\alpha_i = 2 \tag{2}$$

It can be observed that for RSVC, $\delta > 1$ and it reduces to the original SVC formulation for $\delta = 1$. It may also be observed that the images of points with s $\alpha_{\mathrm{i}} = 0$ lie in lower approximation (hard points), $0 < \alpha_{\mathrm{i}} < \frac{1}{\upsilon m}$ form the Hard Support Vectors, (Support Vectors which mark the boundary of lower approximation), $\alpha_{\mathrm{i}} = \frac{1}{\upsilon m}$ lie in the boundary region (patterns that may be shared by more than one cluster, a soft point). $\frac{1}{\upsilon m} < \alpha_{\mathrm{i}} < \frac{\delta}{\upsilon m}$ form the Soft Support Vectors (Support Vectors which mark the boundary of upper approximation) and $\alpha_{\mathrm{i}} = \frac{\delta}{\upsilon m}$ lie outside the sphere (Bounded Support Vectors or outlier points). From $\sum_{i=1}^{m}\alpha_i = 2$ and $0 \leq \alpha_i \leq \frac{1}{\upsilon m}$ we can see that the number of BSVs $n_{bsv} < 2(\frac{\upsilon m}{\delta})$ for $\delta = 1$ $n_{bsv} < 2(\upsilon m) = \upsilon' m$ This corresponds to all the patterns x_i with $\| \phi(x_i) - \mu \|^2 > R^2$. Since $\delta > 1$ for RSVC, we can say that $\frac{\upsilon'}{\delta}$ is the fraction of points permitted to lie outside T and υ' is the fraction of points permitted to lie outside R. Hence υ and δ together give us control over the width of boundary region and the number of BSVs. The algorithm to find clusters in RSVC is given in Algorithm 1.

3 Rough Core Vector Clustering

Consider a situation where $k(x, x) = \kappa,$ a constant. This is true for kernels like Gaussian given by $k(x_i, x_j) = e^{-g\|x_i - x_j\|^2}$, where $\| \cdot \|$ represents the L_2 norm

```
Data: Input Data
Result: Cluster Labels for the Input Data
  – Find adjacency matrix M as $M[i,j] = \begin{cases} 1 \text{ if } G(y) \leq R & \forall y \in [x_i, x_j] \\ 0 \text{ otherwise} \end{cases}$
  – Find connected components for the graph represented by M.
    This gives the Lower Approximation of each cluster.
  – Now find the Boundary Regions as $x_i \in L\_A(C_i)$ and
    pattern $x_k \notin L\_A(C_j)$ for any cluster j,
    if $G(y) \leq T \ \forall y \in [x_i, x_k]$ then $x_k \in B\_R(C_i)$
```

Algorithm 1: Algorithm to find clusters

and g is a user given parameter. The dot product kernel like polynomial kernel given by $k(x_i, x_j) = (< x_i, x_j > +1)^{\lambda}$ with normalized inputs x_i and x_j also satisfy the above condition(λ is a non-negative integer).Now the dual of the RSVC problem given in equation (2) reduces to the form

$$\min \sum_{i,j}^{m} \alpha_i \alpha_j K(x_i, x_j) \;\; s.t. \;\; 0 \leq \alpha_i \leq \frac{\delta}{vm} \;\; \forall i \quad \sum_{i=1}^{m} \alpha_i = 2 \tag{3}$$

The above equation can be solved by an adapted version of the Minimum Enclosing Ball(MEB) finding algorithm used in CVM[9]. Let us define a Soft Minimum Enclosing Ball(SMEB) as an MEB of the data allowing $\frac{v'}{\delta}$ fraction of the points to lie outside T(see the discussion in Section 2). To solve the above problem we use an approximate SMEB algorithm given in next Section.

3.1 Approximate Soft MEB Finding Algorithm

The traditional algorithms for finding MEBs are not efficient for $d > 30$ and hence as in CVM[9], the RCVC technique adopts a variant of the faster approximation algorithm introduced by Badou and Clarkson[10]. It returns a solution within a multiplicative factor of $(1 + \epsilon)$ to the optimal value, where ϵ is a small positive number.

The $(1 + \epsilon)$ approximation of the SMEB problem is obtained by solving the problem on a subset of the data set called *Core Set*. Let $B_S(a, T)$ be the exact SMEB with center a and outer radius T for the data set S and $B_Q(\tilde{a}, \tilde{T})$ be the SMEB with center $\tilde{a}$ and radius $\tilde{T}$ found by solving the SMEB problem on a subset of S called Core Set(Q). Given an $\epsilon > 0$, a SMEB $B_Q(\tilde{a}, (1 + \epsilon)\tilde{T})$ is an $(1 + \epsilon)$-approximation of $SMEB(S) = B_S(a, T)$ if $B_Q(\tilde{a}, (1 + \epsilon)\tilde{T})$ forms the $SMEB(S)$ and $\tilde{T} \leq T$.

Formally, a subset $Q \subseteq S$ is a core set of S if an expansion by a factor $(1+\epsilon)$ of its SMEB forms the $SMEB(S)$. The approximate MEB finding algorithm uses a simple iterative scheme: At the t^{th} iteration, the current estimate $B_Q(\tilde{a}_t, \tilde{T}_t)$ is expanded incrementally by including that data point in S that is farthest from the center $\tilde{a}$ and falls outside the $(1 + \epsilon)$-ball $B_Q(\tilde{a}_t, (1 + \epsilon)\tilde{T}_t)$. To speed up the process, CVM uses a probabilistic method. Here a random sample S' having 59 points is taken from the points that fall outside the $(1+\epsilon)$-ball $B_Q(\tilde{a}_t, (1+\epsilon)\tilde{T_t})$. Then the point in S' that is farthest from the center $\tilde{a}_t$ is taken as the approximate farthest point from S. The iterative strategy to include the farthest point in the MEB is repeated until only $\frac{v'}{\delta}$ fraction of the points to lie outside T. The set of all such points that got added forms the core set of the data set.

3.2 The Rough Core Vector Clustering Technique

The cluster labeling procedure explained in RSVC can now be used to find the soft clusters in Q. Now the question is how can we cluster the rest of the points in the data set S. For that RCVC technique employs the Multi class SVM algorithm discussed below.

A Multi class SVM(MSVM). MSVM is formulated here as an SVM[11] with vector output. This idea comes from a simple reinterpretation of the normal vector of the separating hyperplane. This vector can be viewed as a projection operator of the feature vectors into a one dimensional subspace. An extension of the range of this projection into multi-dimensional subspace gives the solution for vector labeled training of SVM.

Let the training data set be $S = \{(x_i, y_i)\}_{i=1}^{m}$ where $x_i \in R^d, \quad y_i \in R^T$ for some integers $d, T > 0$. i.e. we have m training points whose labels are vector valued. For a given training task having T classes, these label vectors are chosen out of the finite set of vectors $\{y_1, y_2, ...y_T\}$. The primal is given as

$$\min_{W,\rho,\xi_i} \quad \frac{1}{2}trace(W^T W) - \rho + \frac{1}{\nu_2 m}\sum_{i=1}^{m}\xi_i$$

$$\text{s.t.} \quad y_i^T(W\phi(x_i)) \geq \rho - \xi_i \tag{4}$$

$$\xi_i \geq 0 \quad \forall i \tag{5}$$

The corresponding Wolfe Dual[8] is

$$\min_{\alpha_i} \quad \sum_{i,j=1}^{m}\alpha_i\alpha_j(\frac{1}{2} < y_i, y_j > k(x_i, x_j)) \text{ s.t. } \sum_{i=1}^{m}\alpha_i = 1 \quad 0 \leq \alpha_i \leq \frac{1}{\nu_2 m} \quad \forall i \tag{6}$$

The decision function predicting one of the labels from $1...T$ is

$$\arg\max_{t=1...T} < y_t, (W\phi(x_j)) >= \arg\max_{t=1...T}\left(\sum_{i=1}^{m}(\alpha_i < y_i, y_t > (k(x_i, x_j)))\right) \tag{7}$$

Let $y_i(t)$ denote the t^{th} element of the label vector y_i corresponding to the pattern x_i. One of the convenient ways is to choose the label vectors is as

$$y_i(t) = \begin{cases} \sqrt{\dfrac{(T-1)}{T}} & \text{if item } i \text{ belongs to category } t \\[2ex] \sqrt{\dfrac{1}{T(T-1)}} & \text{otherwise} \end{cases}$$

It may be observed that this kind of an assignment is suitable for any $T > 2$. Now it may be observed that the MSVM formulation given by equation (6) will become an MEB problem with the modified kernel $\tilde{k}(z_i, z_j) =< y_i, y_j > k(x_i, x_j)$ given $k(x, x) = \kappa$. So we can use the approximate SMEB finding algorithm discussed above to train the MSVM.

3.3 Cluster Labeling in RCVC

Once the RSVC clustering is done for the core set, we train the MSVM(used as classifier) on the coreset using the cluster labels. Now any point $x_i \in (S - Q)$ or the test data is clustered as follows. If it is a hard point, take the output cluster label predicted by MSVM (*Hard decision*) else if it is a soft point or outlier point, we take a soft labeling procedure where the point is allowed to belong to more than one cluster. Here the number of allowed overlaps *limit* is taken as a user defined parameter and the point is allowed to belong to any of the top ranked *limit* number of clusters found by the MSVM(*Soft decision*).

Table 1. Details of the data sets(x/y) x and y denote Train and Test sizes respectively, NC denote Number of Classes, and Parameter values g, δ, ν, ϵ are used for RCVC and ν_2 used for MSVM and the description of these parameters are same as explained in section 2 and 3

Dataset(x/y)/parameter	dim	NC	g	δ	ν	ϵ	ν_2
combined(78823/19705)	100	3	9.70375	10.922222	0.09	0.0922	0.06
acoustic(78823/19705)	100	3	177.70375	11.07	0.09	0.0922	0.06
seismic(78823/19705)	100	3	177	7.956	0.125	0.0922	0.061
shuttle(43500/14500)	9	7	777	5.6	0.125	0.0922	0.06
forest(526681/56994)	54	7	77.70375	10.666667	0.09	0.0922	0.06

4 Experimental Results

Experiments are done with five real world data sets viz; the shuttle, acoustic, seismic, combined from the LIBSVM[12] page available at *"http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/multiclass"* and Forest covertype data set from the UCI KDD archive available at *"http://kdd.ics.uci.edu/databases/covertype/covertype.html"*. Ten percent of this forest data set is sampled uniformly keeping the class distribution and is taken as the test data set. The rest is taken as the train data set. For all the data sets, a comparison with the one-against-one

Table 2. Results on **One Vs One SVM** and **RCVC**. The abbreviations used are - #SV : No. of Support Vectors, NC : No. of clusters, CA : Classification Accuracy on test data in percentage. Train and Test time are given in seconds.

Algo./Data		**combined**	**acoustic**	**seismic**	**shuttle**	**forest**
	#SV	34875	52593	45802	301	339586
1Vs1 SVM	Train Time	3897	5183	3370	170	67328
	Test time	750	686	565	7	6321
	CA(%)	**81.91**	**66.9728**	**72.5704**	**99.924**	**72.3409**
RCVC	coreset size	142	140	126	161	145
	#SV	142	140	126	159	145
	NC	69	105	54	91	43
	train time	131	141	81	38	370
	test time	22	27	13	8	25
	CA(%)	**92.971327**	**91.727988**	**87.561533**	**99.965517**	**88.08822**

SVM(1Vs1 SVM)[12] is done. In all the experiments we used Gaussian kernel function. All the experiments were done on a Intel Xeon(TM) 3.06GHz machine with 2GB RAM. The details of the data sets along with the parameters used in the experiments and the results obtained are given in Table 1 and 2 respectively.

5 Conclusion

RCVC is computationally efficient than RSVC. Soft clustering scheme used in RCVC enables us to identify ambiguous regions (having overlap between clusters of heterogeneous nature) in the data set. RCVC allows soft clusters of any arbitrary shape and the extent of softness can be varied by controlling the width of boundary region and number of outliers done by changing the user defined parameters. So, this method is scalable and involves less computation when compared to RSVC.

References

1. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, pp. 226–231 (1996)
2. Guha, Rastogi, Shim: CURE: An efficient clustering algorithm for large databases. SIGMODREC: ACM SIGMOD Record 27 (1998)
3. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an efficient data clustering method for very large databases. In: Proceedings of ACM-SIGMOD International Conference of Management of Data, pp. 103–114 (1996)
4. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
5. Lingras, P., West, C.: Interval set clustering of web users with rough K -means. J. Intell. Inf. Syst 23, 5–16 (2004)

6. Asharaf, S., Shevade, S.K., Murty, N.M.: Rough support vector clustering (2005)
7. Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V.: Support vector clustering. Journal of Machine Learning Research 2, 125–137 (2001)
8. Fletcher: Practical Methods of Optimization. Wiley, Chichester (1987)
9. Tsang, I.W., Kwok, J.T., Cheung, P.M.: Core vector machines: Fast SVM training on very large data sets. Journal of Machine Learning Research 6, 363–392 (2005)
10. Badoiu, M., Clarkson, K.L.: Smaller core-sets for balls. In: SODA, pp. 801–802 (2003)
11. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Knowledge Discovery and Data Mining 2, 121–167 (1998)
12. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. Online (2001)