Finding Clusters using Support Vector Classifiers

Kees Jong, Elena Marchiori, Aad van der Vaart
Department of Mathematics and Computer Science
Free University Amsterdam
The Netherlands

Abstract. This paper shows how clustering can be performed by using support vector classifiers and model selection. We introduce a heuristic method for non-parametric clustering that uses support vector classifiers for finding support vectors describing portions of clusters and uses a model selection criterion for joining these portions. Clustering is viewed as a two-class classification problem and a soft-margin support vector classifier is used for separating clusters from other points suitably sampled in the data space. The method is tested on five real life datasets, including microarray gene expression data and array-CGH data.

1 Introduction

Clustering is the problem of finding structure in the data by identifying groups of objects that are more similar to each other than to objects in other groups. The notion of similarity is domain dependent, so clustering is an ill-defined problem for which many heuristic methods based on different approaches are introduced (see e.g. [3] for a recent overview of clustering methods). Support vector machine (SVM) is a powerful technique for classification and regression [10]. In this paper we propose a method for clustering that uses a support vector classifier for finding support vectors which represent portions of clusters. The method searches for support vectors close to clusters cores. Each support vector is used to construct a (portion of a) cluster. Then a Bayesian information criterion is used for merging nearby (portions of) clusters.

We view clustering as a two-classes classification problem, where class A contains data elements while class B contains other points suitably generated. Initially, class A consists of all the data, and class B consists of uniform points randomly generated in the minimum hyper-rectangle containing the data. A soft-margin support vector classifier is trained for separating A and B, and the non-bounded support vectors SV_A of class A are considered as candidate representatives of (portions of) clusters. Next, SV_A is removed from A, B is set to SV_A, and a SVM classifier is trained to separate the resulting new classes A and B. The idea is that at each iteration the non-bounded support vectors of the actual class A get closer to the cores of some clusters. At each iteration the

same SVM parameters are used, with small soft-margin constant and kernel width, resulting in few non-bounded support vectors. A heuristic criterion is used for deciding when to stop the iterative process. The non-bounded support vectors selected at the end of the iterative process are used for building clusters by assigning each data point to the closest support vector. Finally, near clusters are merged using a model selection criterion.

We conduct experiments on real life datasets to test the effectiveness of this clustering method. We focus mainly on real life datasets from biological experiments, in particular microarray gene expression data and array-CGH data. The results of the experiments are rather satisfactory and indicate that support vector classifiers can be used in combination with a model selection criterion for clustering.

Related Work

The use of SVM for clustering has been recently proposed by Ben-Hur et al. [2]. However, they view clustering as one-class classification problem, where the data is distinguished from the rest of the feature space [6, 7, 9, 10] by finding a sphere with minimal volume enclosing the data. The boundary of the sphere in the input space forms a set of closed contours which are interpreted as clusters boundaries, where each closed boundary forms a cluster. The authors introduce a heuristic algorithm for finding these clusters, based on the observation that two points belong to the same cluster if all the points on the line segment connecting them lie in or on the sphere in the feature space.

A recent method that incorporates model selection as decision test in a clustering algorithm has been introduced by Pelleg and Moore [5]. The authors propose a non-parametric clustering algorithm called X-means, where a clustering is incrementally constructed by splitting clusters and using BIC as criterion for accepting the splittings. The algorithm is an extension of K-means with efficient estimation of the number of clusters.

2 The Clustering Algorithm

The clustering method employs a soft-margin SVM classifier with Gaussian kernel

$$K(x, y) = e^{-\sigma||x-y||^2}$$

with width parameter σ. In the input space the points for which the SVM decision function is equal to 1 describe boundaries enclosing the data. Support vectors lay on or outside these boundaries, where non-bounded support vectors[1] lay on these boundaries. The SVM parameters C and σ affect the form and smoothness of the boundaries. Small values of C and σ induce smooth boundaries with few non-bounded support vectors and more bounded ones. Non-bounded support vectors can be viewed as representatives of portions of clusters. Clusters can be obtained by joining these portions using a model

[1] a support vector is non-bounded if its Lagrange multiplier is smaller than C

selection criterion based on the Bayesian information criterion (BIC) [8]. According to this criterion, the most probable model is the one that maximizes the log-likelihood of the model given the observed data minus a term which takes into account model dimension.

Here a model is a clustering $Cl = \{Cl_1, \ldots, Cl_k\}$ of X. We assume that each cluster Cl_i follows a Gaussian distribution and use a scoring function obtained from the formula $-2\text{log-likelihood}(X) + \lambda N$ where N is the number of free parameters of the model and λ is a penalty weight. (In our experiments we use $\lambda = 5$.)

$$score(Cl) = \sum_i k_i \log(k_i) - nm \log(\hat{\sigma})/2 - (n-k)/2 + 5km$$

where k_i is the cardinality of Cl_i, n the cardinality of X, m the input dimension, and $\hat{\sigma} = \sum_{i=1}^{n}(x_i - \hat{\mu}(i))^2/(n-k)$, with $\hat{\mu}(i)$ the mean vector of the cluster containing x_i. So the smaller the score the better the model. We will compare clusterings using *score* for deciding whether to join two clusters and also to choose a clustering amongst those generated by running the clustering algorithm a number of times with different initializations of B (see Step 1 below).

A high level description of the algorithm is given below, where := denotes variable assignment, the variables A and B denote the two classes to be separated, R the set of support vectors selected as representatives, and Cl the final clustering. The algorithm is called FJC (Find and Join Clusters).

```
FJC CLUSTERING ALGORITHM
Input. X (the dataset of size n).
Output. Cl (A clustering of X).
1. Initialization.
   A := X,
   B := {n randomly generated uniform points
       in the hyper-rectangle containing X},
   R := {},
   C := C_0,
   sigma := sigma_0 (in our experiments C=2, sigma=0.1).
2. Find representatives of clusters portions.
Repeat the following two steps for n_iter
    (in our implementation n_iter=4).
   2.a Apply SVM(C_0,sigma_0) to separate A and B,
         Cl_A := {non-bounded support vectors of A},
         If (Cl_A contains more elements than R)
            then R := Cl_A.
  2.b SV_A := {the support vectors of class A},
       A := A minus SV_A,
       B := SV_A,
3. Build clusters portions.
Let R = {r_1,...,r_k} (obtained from Step 2).
```

```
Cl := {Cl_1,...,Cl_k}  with
Cl_i = {x in X closer to r_i than to any other r_j}.
4. Join clusters portions.
Repeat the following statement until Cl does not change.
   for each  Cl_i in Cl:
      c_i := mean vector of Cl_i,
      Find Cl_j containing a point closest to c_i,
      Cl_union := (Cl-{Cl_i,Cl_j}) union {Cl_i,Cl_j},
      If  (score(CL_union) < score(Cl))
            then Cl := Cl_union.
```

In Step 1, different sets B of uniform points can be generated depending on the random seed used, which may yield to different clusterings. Thus FJC is executed a number of times and a preferred clustering is selected using *score*.

Two heuristic criteria are used in FJC: the way in which R is chosen in Step 2 and the strategy used for joining clusters portions in Step 4.

In Step 2 the biggest of the Cl_A's generated in the iterative process is selected as set of representatives of (portions of) clusters. This is a rather conservative criterion based on the intuition that a higher number of non-bounded support vectors may indicate a better separation of the classes. At each iteration the number of elements of class A decreases, so it would seem more intuitive to use the ratio (cardinality of Cl_A)/(cardinality of A). However, experiments on artificial datasets indicate that by using this latter criterion (representatives of) smaller clusters may be lost.

In Step 4 the algorithm tries to join portions of clusters. We use a simple heuristic where pairs of clusters are joined if the *score* of the resulting clustering decreases. Pairs of clusters are selected as follows: one cluster is chosen (in textual order) from the actual clustering, its mean vector is computed, and then the cluster closest to the mean vector is chosen. This heuristic has the advantage of being fast and yielding satisfactory results.

The algorithm is implemented in Matlab and uses the Matlab implementation by Gavin C. Cawley[2] of Platt SMO algorithm.

3 Experiments

We consider five datasets from biological experiments. The first two datasets, available at the UCI ML repository, have been often used as benchmarks for classification algorithms: the popular Fisher `iris` dataset with 150 points, 4 attributes, 3 classes, 50 points per class; the Wisconsin breast cancer `wdbc` with 569 points, 30 attributes, 2 classes, 357 benign and 212 malignant. The `colon` dataset [1] consists of microarray measurements of 2000 genes over 44 colon cancer and 22 non cancer tissues. Here we cluster tissues, so we have 62 points, and select the 200 genes having highest variance over the tissues. The `leukemia` dataset [4] consists of microarray measurements of 12582 genes over

[2]available at http://www.sys.uea.ac.uk/ gcc/svm/toolbox/

dataset	data	classes	clusters	nbsv	error	Jaccard	robustness
cgh	55	2	4	28	1	0.95	0.69
colon	62	2	2	54	12	0.54	0.61
wdbc	569	2	7	7	22	0.35	0.74
leukemia	72	3	3	36	3	0.91	0.77
iris	150	3	3	11	11	0.78	0.75

72 tissues of three Leukemia types, ALL (24), MLL (20), and AML (37). We select the 50 genes having highest variance over the tissues. Finally the `cgh` dataset consists of array Comparative Genomic Hybridization measurements of 1354 clones of the genome over 43 stomac cancer and 13 ovarian cancer tissues. This dataset was provided by our collegues at the Free University Medical Center of Amsterdam. We select the 150 clones having highest variance over the tissues.

We run FJC on each dataset 10 times and selected the best clustering according to *score*. The results of the experiments are given in the table below, where the columns contain: the dataset name, the number of data, the number of classes (the 'true' clusters), followed by the results of FJC, that is, the number of clusters, the number of non-bounded support vectors, the number of points misclassified, the Jaccard score and the robustness. The Jaccard score is $p_{11}/(p_{11}+p_{10}+p_{01})$ where p_{11} is the number of pairs of points belonging to the same cluster in both the 'true' and FJC clustering, p_{10} (respectively p_{01}) is the number of pairs that belong to the same cluster in the FJC (respectively 'true') clustering but not in the 'true' (respectively FJC) one.

The robustness gives and indication of FJC consistence in assigning pairs of points to the same or different clusters over the runs. For each pair of points x_i, x_j we compute the fraction ns_{ij} of times they occur in the same cluster over the 10 runs, and then $a_{ij} = 1 - 2\text{minimum}(ns_{ij}, 1 - ns_{ij})$. For a pair of point, a_{ij} provides a measure of the confidence of the majority vote decision of assigning x_i and x_j to the same or different clusters. The confidence is 1 when x_i, x_j are either always or never in the same cluster. Then the robustness is the average confidence over all pairs of points.

The results indicate that FJC is a robust clustering method, capable of identifying 'true' structure in all the five datasets. The misclassification errors of FJC are comparable to those found by state-of-the-art classification methods. In particular, on the `cgh` and `leukemia` dataset FJC is able to identify the classes almost perfectly.

In order to show the benefit of using the support vector classifier in FJC, we consider the algorithm obtained from FJC by removing the SVM part. This amounts to start FJC from Step 4 with Cl consisting of one data point per cluster. The resulting algorithm yields the following results: 3 clusters, error 4 for `cgh`; 2 clusters, error 15 for `colon`; 15 clusters, error 25 for `wdbc`; 2 clusters, error 26 for `leukemia`; and 4 clusters, error 26 for `iris`.

4 Conclusion

This paper showed that non-parametric clustering can be performed by using suppport vector classifiers and model selection. We conclude with some issues we intend to address in future work. Alternative ways to initialize the class B of points in Step 1 of FJC. A possibility is finding the sphere with minimum volume in the feature space enclosing the data [9], and consider the resulting support vectors as class B and the rest of the data as class A. The model selection criterion we use for joining clusters assumes a Gaussian distribution. Other decision criteria that do not rely on this assumption should be devised. Finally, in FJC a pre-processing is applied to deal with high dimensional datasets that selects features with high variance. Different methods for input dimension reduction should be investigated.

References

[1] U. Alon and et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon cancer tissues probed by oligonucleotide arrays. *Cell Biology*, 96:6745–6750, 1999.

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

[3] P. Berkhin. Survey of clustering data mining techniques, 2002. http://www.accrue.com/products/researchpapers.html.

[4] T.R. Golub et al. Molecular classification of cancer: Class discovery and class prediction by gene expression. *Science*, 286:531–537, 1999.

[5] Dan Pelleg and Andrew Moore. X-means: Extending K-means with efficient estimation of the number of clusters. In *Proc. 17th International Conf. on Machine Learning*, pages 727–734. Morgan Kaufmann, San Francisco, CA, 2000.

[6] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1472, 2001.

[7] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. *Advances in Neural Information Processing Systems*, 12:582–588, 2000.

[8] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[9] D. M. J. Tax and R. P. W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20(1113):1191–1199, 1999.

[10] V.N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.