A Hierarchical Projection Pursuit Clustering Algorithm

Alexei D. Miasnikov, Jayson E. Rome, Robert M. Haralick
Pattern Recognition Lab
Department of Computer Science
The Graduate Center of The City University of New York
365 Fifth Avenue
New York, New York 10016
{ amiasnikov,jrome,haralick}@gc.cuny.edu

Abstract

We define a cluster to be characterized by regions of high density separated by regions that are sparse. By observing the downward closure property of density, the search for interesting structure in a high dimensional space can be reduced to a search for structure in lower dimensional subspaces. We present a Hierarchical Projection Pursuit Clustering (HPPC) algorithm that repeatedly bi-partitions the dataset based on the discovered properties of interesting 1-dimensional projections. We describe a projection search procedure and a projection pursuit index function based on Cho, Haralick and Yi's improvement of the Kittler and Illingworth optimal threshold technique. The output of the algorithm is a decision tree whose nodes store a projection and threshold and whose leaves represent the clusters (classes). Experiments with various real and synthetic datasets show the effectiveness of the approach.

1. Introduction

A cluster is a set of data points partitioned in such a way that individual data elements within the same cluster are in some sense more similar to elements in the cluster than to elements outside of the cluster. Clustering is an old and well studied problem [13], though most standard techniques assume that the data is drawn from a known parametric distribution or that the data exists in a low dimensional space. There has been a great deal of recent work toward developing efficient and effective algorithms for clustering including [1, 7, 12, 15, 14, 17], and we refer the interested reader to [9, 11] for extensive surveys of recent clustering research.

Given a collection of observations $X = \{x_i\}$, $x_i \in R^d$, $|X| = N$, we would like to find clusters in data sets in which there is no known parametric distribution and in which clusters may take on arbitrary shapes. For these kinds of data sets traditional techniques fail and new methods must be developed.

We assume that clusters are characterized by regions of high density separated by regions that are sparse and observe that any low density region in a subspace of the data corresponds to a low density region in the full space. Thus the search for interesting structure in the high dimensional space can be reduced to a search for structure in lower dimensional subspaces. Based on this downward closure property of density, we develop a Hierarchical Projection Pursuit Clustering (HPPC) algorithm. The algorithm has no required input parameters, other than a sensitivity parameter and produces a compact description of the clusters found in the form of a binary tree. This tree, once clusters have been found, can be used to classify new observations as belonging to particular clusters.

Projection Pursuit [10] or PP, is defined to be the search for interesting (structured) projections. Mathematically, interestingness is a sample estimate of a distance between the distribution of the projected data and a distribution that is known to be uninteresting. Uninteresting distributions are normally taken to be uniform or normal, though some other parametric form may be used, or they may be determined empirically for a specific case. A PP algorithm consists of two components: an index function $I(\alpha)$ that measures the "usefulness" or "interestingness" of projection α and a search algorithm that varies the projection direction so as to find the optimal projections, given the index function $I(\alpha)$ and the data set X.

In section 2 we present the indexing function, search procedure and stopping criteria that comprise the HPPC algorithm. In section 3 we address the issue of cluster validation and describe a method for computing the accuracy of a cluster algorithm. In section 4 we describe various experiments that where performed on a variety of real and synthetic data, while section 5 discusses conclusions and possible future

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE

work.

2. The HPPC Algorithm

HPPC is a hierarchical projective clustering algorithm that repeatedly bi-partitions a dataset by looking for separations in one dimensional subspaces of the data. Subspaces are split using Cho, Haralick and Yi's [4] improvement of Kittler and Illingworth's minimum error thresholding technique [16]. The pseudo-code in figures 1 and 2 describe the algorithm. Each time a split is chosen a node is created in a decision tree, the data is partitioned and the algorithm is repeated on the data in each of the leaves of the tree, until a stopping condition is satisfied. The output of the algorithm is a decision tree whose nodes store the projection and optimal threshold and whose leaves represent the clusters. The tree that HPPC constructs can be used to classify new observations.

procedure *[τ, θ, α] = SplitData(dataset X)*
Initialize best threshold $\tau = 0$, best index $\theta = -\infty$, best projection $\alpha = \mathbf{0}$;
Construct S : the set of candidate 1D projections;
FOR $\alpha_i \in S$ **DO**
    $hist_i = histogram(\alpha_i X)$;
    current threshold $\tau_i = FindMinErrorThreshold(hist_i)$;
    if(StoppingConditionSatisfied($hist_i$)) **CONTINUE**;
    current evaluation $\theta_i = EvaluateThreshold(\tau_i, hist_i)$;
    if($\theta_i > \theta$) $\theta = \theta_i, \tau = \tau_i, \alpha = \alpha_i$;
END FOR LOOP

Figure 1. The splitting procedure

procedure *HPPC(dataset X, tree $Dtree$)*

    $[\tau, \theta, \alpha]$ = $SplitData(X)$;
    If($\tau = 0$ *or* $\theta = -\infty$ *or* $\alpha = 0$)
        no such projection exists, **RETURN**;
    Else split the data
        $X_L = \{x_i \in X | \alpha x_i < \tau\}$;
        $X_R = \{x_i \in X | \alpha x_i \geq \tau\}$;
    Add new node (α, τ) *to* $Dtree$;
    HPPC($X_L, Dtree$);
    HPPC($X_R, Dtree$);

Figure 2. The clustering procedure

2.1. The Index Function

Diaconis and Freedman [6] showed that, as a consequence to the central limit theorem, most projections of high dimensional datasets to low dimension will be approximately normally distributed. Based on this observation, Dasgupta [5] suggested that clusters in subspaces could be used to estimate the distribution of the data in the full space.

We consider interesting projections to be those for which there exists a natural partition of the dataset into two components, the projections of which will each be approximately normally distributed, according to the results of Diaconis and Freedman.

Assuming that each component represents separate classes, we use Kittler and Illingworth's [16] method for splitting a mixture of two 1D Gaussian components whose goal is to minimize the miss-classification error.

Let $h(g)$ be the normalized frequency histogram of the various levels of $g = \alpha x_i, i = 1, ..., N$. The histogram is viewed as an estimate of the probability density function of a mixture of two clusters. Let $p(g|i)$, $i = 1, 2$, be the estimated distribution of the i^{th} component's projection, having mean μ_i, standard deviation σ_i and *a priori* probability P_i, so that $p(g) = \sum_{i=1}^{2} P_i p(g|i)$, where $p(g|i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-(g-\mu_i)^2/2\sigma_i^2}$. Given the (μ_i, σ_i, P_i), there exists a threshold τ such that $P_1 p(g|1) > P_2 p(g|2)$ if $g \leq \tau$ and $P_1 p(g|1) < P_2 p(g|2)$ if $g > \tau$.

Threshold τ is the Bayes minimum error threshold. For a given threshold T we can model the two resulting populations by a normal density $h(g|i, T)$ with parameters $(\mu_i(T), \sigma_i(T), P_i(T))$ given by: $P_i(T) = \sum_{g=a}^{b} h(g)$, $\mu_i(T) = \sum_{g=a}^{b} g * h(g)/P_i(T)$ and

$$\sigma_i^2(T) = \frac{\sum_{g=a}^{b} (g - \mu_i(T))^2 * h(g)}{P_i(T)}$$

where $a = 0$ if $i = 1$, $a = T + 1$ if $i = 2$, $b = T$ if $i = 1$ and $b = n$ if $i = 2$. The Kittler and Illingworth criterion function for threshold T is given by

$$J(T) = 1 + 2\left(P_1(T)\log\sigma_1(T) + P_2(T)\log\sigma_2(T)\right)$$
$$-2\left(P_1(T)\log P_1(T) + P_2(T)\log P_2(T)\right).$$

Note that the tails of the distributions have been truncated by the thresholding operation and therefore the models $h(g|i, T), i = 1, 2$ will be biased estimates of the true mixture components. Cho, Haralick and Yi [4] proposed an improvement of Kittler and Illingworth's criterion function that corrects the biased variance estimates.

The index function that evaluates the interestingness of a projection α is given by $I(\alpha) = sep * depth$, where

$$sep(\tau) = \frac{(\mu_1(\tau) - \mu_2(\tau))^2}{\sigma_1^2(\tau) + \sigma_2^2(\tau)}.$$

and $depth = J(T_{max}) - J(\tau)$, with $J(T_{max})$ being the closest local maxima.

2.2. Searching for Projections

As the dimensionality of the data increases an efficient projection selection becomes crucial. We use a stochastic search method which combines a genetic approach with Simultaneous Perturbation Stochastic Approximation (SPSA) [18] to optimize the search procedure. The objective of the stochastic search procedure is to find a projection α which maximizes the index function I. Due to page limitations, we have to omit the details of the procedure in this report.

2.3. Stopping Conditions

Due the recursive nature of HPPC, defining a robust stopping condition is crucial. Given Apriori knowledge of the characteristics of uninteresting clusters, one can construct a distribution f_I of the values of the criterion function. These clusters may be determined empirically or may be chosen from some parametric form.

For a given set of observations C we use f_I to test the Null Hypothesis H_0, that C is a cluster, against the alternative hypothesis H_1, that C can be partitioned into at least two clusters. Let I_ϕ be the value of the index function such that

$$\int_{-\infty}^{I_\phi} f_I dI = 1 - \phi.$$

Where ϕ is the level of significance of the test. We reject H_0 if $I > I(\phi)$ and do not reject otherwise.

The value of ϕ, which is the probability that a truly cohesive cluster will be split by the algorithm, can be viewed as a sensitivity parameter and can be tuned for a particular application.

Lacking prior knowledge of what constitutes a cohesive cluster, datasets used to train the Null distribution should be those which are difficult to evaluate. For example, a very sparse sample of a cohesive cluster is easy to split mistakenly.

2.4. Extensions to Non-Linearly Separable Data and Large Datasets

It is important to note that while the HPPC algorithm is only able to locate separations in linearly separable data, the methodology can be extended to the case of non-linearly separable data by appropriate pre-processing of the data with various non-linear mappings.

The problem of evaluating datasets with a large number of observations can be ameliorated by subset sampling. The full dataset can then be clustered using the tree constructed from the sample. We apply these techniques in section 4.1.

3. Cluster Validation and Evaluation

We use datasets with known ground truth labeling to evaluate the algorithm. Given the ground-truth labels and the labels determined by HPPC we form a contingency table by counting the number of times the algorithm assigns a label j to a cluster point when the actual label is i. We form all possible mappings from the smaller of the actual and assigned label sets onto the larger of these two sets and build a confusion matrix for each mapping. Since the diagonal elements of the confusion matrix represent the number of times that an instance of class i was correctly identified as belonging to class i, we can therefore derive a criterion function that returns a value for each mapping based on the number of correct classifications. The optimal mapping is the one associated with the confusion matrix whose sum along the diagonal (the trace) is maximal.

4. Experiments

To verify the effectiveness of our method we ran experiments on various real and synthetic datasets. The algorithm was implemented in the C++ programming language. First we utilize common and well known 2D datasets for which clusters can be visually verified. Then we evaluate the algorithm on synthetic and real data and compare the results against standard k-means and the EM algorithm with unconstrained Gaussian mixture models. The true number of clusters was specified as a parameter for both k-means and EM algorithms. Initial centers for k-means were chosen randomly and initial parameters of EM algorithm were estimated from the runs of k-means procedure.

4.1. Results For 2D Data Sets

We considered two examples that are considered difficult to cluster: D1, first used in CURE [12] and a set of concentric circles D2, both shown in figure 3. For D1 we used a uniform sampling procedure to reduce the dataset size from 100,000 points to 2,000 points. HPPC successfully finds correct clusters defined in [12].

Concentric circles are considered to be the degenerate case for projection based methods. For this dataset we map the data from 2 to 5 dimensions by introducing the quadratic terms: $(x, y) \rightarrow (x, y, x^2, xy, y^2)$.

As can be seen in figure 3, the quadratic mapping allows HPPC to easily find the clusters, which is impossible using regular linear projections.

4.2. Experiments with Real and Synthetic Data Sets

We used a synthetic data generation procedure proposed in [2] to construct mixtures of multivariate normal distri-

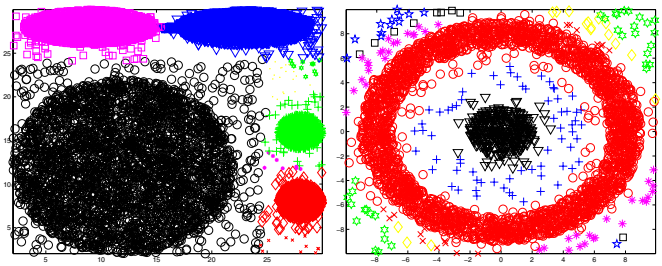

Figure 3. Clustering of 2D datasets.

				HPPC		EM		k-means	
Set	d	k	N	avg	max	avg	max	avg	max
D_2	8	2	600	1.0	1.0	1.0	1.0	.99	.99
D_4	8	4	1200	.99	1.0	.80	.81	.51	.61
D_8	8	8	2400	.99	.99	.64	.87	.35	.47
D_i	4	3	150	.94	.99	.87	.96	.70	.94
D_c	5	4	200	.90	.93	.52	.64	.35	.40
D_r	19	7	2310	.69	.76	.59	.68	.46	.59

Table 1. Data description and accuracy.

butions D_2, D_4, D_8 for use in our experiments. In addition, we ran HPPC on the following real datasets chosen from the public domain: Fisher's iris data D_i [3, 8] a commonly utilized benchmark for Pattern Recognition tasks; Australian crab data D_c used in [2]; an image recognition set D_r from UCI Repository of Machine Learning Databases [3].

Since all of the examined algorithms have stochastic components, experiments were performed multiple times. The descriptions of the datasets and summary results from 50 trials of HPPC, EM and k-means algorithms are presented in Table 1, where d is the dimensionality of the data, k is the number of true classes and N is the number of points.

We can see that HPPC performs very well for the synthetic datasets. The average accuracy for sets D_i and D_c is 94% and 90% respectively. There were not more than 9 misclassified observations for the Iris dataset on average with some trials having only 1 misclassification. Accuracy for the set D_r, which has significantly higher dimensionality, is about 70%. The error is due to the overspliting of the sparse clusters during the tree construction. In fact, if we stop the tree expansion when the number of leaves in the tree reaches the true number of clusters, the accuracy increases up to 98%.

5. Conclusions

We have shown that our algorithm can perform very well with a variety of real and synthetic data. The algorithm requires no external input parameters other than the intuitively understandable sensitivity and produces a compact description of the clusters in the form of a binary decision tree which can be efficiently used for classification purposes. For large datasets, sampling can be used to reduce the size of the dataset. Datasets that are not linearly separable can be mapped to higher dimensional spaces in which they are linearly separable. Experimental results show the effectiveness of the technique, particularly in the case of linearly separable data.

References

[1] C. Aggarwal and P. Yu. Redefining Clustering for High-Dimensional Applications. *IEEE Trans. on KDE*, 14(2), 2002.

[2] R. J. Bolton and W. J. Krzanowski. Projection pursuit clustering for exploratory data analysis. *Journal of Computational and Graphical Statistics*, 12(1), 2003.

[3] C. L. Blake and C. J. Merz. UCI Repository of Machine Learning Databases. Technical report, Univ. of California Irvine, Dept. of Information and Computer Sciences, 1998.

[4] S. Cho, R. M. Haralick, and S. Yi. Improvement of Kittler and Illingworth's Minimum Error Thresholding. *Pattern Recognition*, 22(5), 1989.

[5] S. Dasgupta. Experiments with random projection. In *Proc. 16th Conf. on Uncertainty in Artificial Intelligence*, 2000.

[6] P. Diaconis and D. Freedman. Asymptotics of graphical projection pursuit. *Annals of Statistics*, 12, 1984.

[7] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining*, 1996.

[8] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, 1936.

[9] C. Fraley. Algorithms for Model-Based Gaussian Hierarchical Clustering. *Siam Journal on Scientific Compupting*, 20, 1998.

[10] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C-23(9), 1974.

[11] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, 2000.

[12] S. Guha, R. Rastogi, and K. Shim. CURE: an efficient clustering algorithm for large databases. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, 1998.

[13] J. Hartigan. *Clustering Algorithms*. Wiley, 1975.

[14] A. Hinneburg and D. A. Keim. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In *Knowledge Discovery and Data Mining*, 1998.

[15] A. B. Hur, D. Horn, H. T. Siegelmann, and V. Vapnik. Support Vector Clustering. *Journal of Machine Learning Research*, 2, 2001.

[16] J. Kittler and J. Illingworth. Minimum Error Thresholding. *Pattern Recognition*, 19(1), 1986.

[17] R. Ng and J. Han. Clarans: A method for clustering objects for spatial data mining. *IEEE Trans. on KDE*, 14(5), 2002.

[18] J. Spall. Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Trans. on Automatic Control*, 45:1839–1853, 2000.