KYBERNETIKA — VOLUME 34 (1998), NUMBER 4, PAGES 375–380

DETECTING A DATA SET STRUCTURE THROUGH THE USE OF NONLINEAR PROJECTIONS SEARCH AND OPTIMIZATION

VICTOR L. BRAILOVSKY AND MICHAEL HAR-EVEN

Detecting a cluster structure is considered. This means solving either the problem of discovering a natural decomposition of data points into groups (clusters) or the problem of detecting clouds of data points of a specific form. In this paper both these problems are considered. To discover a cluster structure of a specific arrangement or a cloud of data of a specific form a class of nonlinear projections is introduced. Fitness functions that estimate to what extent a given subset of data points (in the form of the corresponding projection) represents a good solution for the first or the second problem are presented. To find a good solution one uses a search and optimization procedure in the form of Evolutionary Programming. The problems of cluster validity and robustness of algorithms are considered. Examples of applications are discussed.

1. INTRODUCTION

The main approach to Exploratory Data Analysis is connected with Projection Pursuit (PP). In many cases this search for "structure in data" is reduced to finding clusters or cluster–like structures. Traditional understanding of what a cluster means is based on the description presented in Everitt (cf. [3]): "A cluster is an aggregation of points in the test space such that the distance between any two points in the cluster is less than the distance between any point in the cluster and any point not in it". The basic approaches to Cluster Analysis are discussed in [3] and [5].

Over the years there has been a tendency to widen this rather vague definition of clusters. This tendency is most clearly expressed in conceptual clustering, with the basic premise that "objects should be arranged in classes that represent simple concepts and are useful from the viewpoint of the goal of clustering. These objects in the same cluster do not necessarily have to be similar in some mathematically defined sense, but must as a group represent the same concept" (cf. [8]).

Thus, according to the latter definition straight lines, spherical and ellipsoidal surfaces, etc may be considered as concepts, and clouds of data points elongated around these lines and surfaces represent clusters. As a result of such an approach there is a bridge between the problem of clustering and the problem of form recognition for 2D images which is a central problem in Image Analysis and Computer

Vision [1].

Nevertheless, despite some robust properties of the methods, developed in this field, it is very problematic to apply them to the problem of interest: to discover and to identify clouds of data points elongated around a certain line or surface.

In this paper we introduce a class of nonlinear projections that are used to detect cluster structures of a given form. On the basis of these projections one performs a PP procedure that should culminate either in discovering a cluster of a given form or in establishing that no clusters of that kind exist in a given data set.

2. NONLINEAR PROJECTIONS

2–1. Consider a d–dimensional space of objects (data points) R_d with coordinates $x_1, x_2, \dots, x_d$.

Let a surface (a curve) exist that can be described by the equation:

$$F(x) = \sum_{i,j=1}^{d} a_{ij} x_i x_j + \sum_{i=1}^{d} a_i x_i + a_0 = 0. \tag{2.1}$$

If there is a structure that may be represented as a cloud of data points elongated around the surface than the majority of such data points complies with the inequality $-\varepsilon \leq F(x) \leq \varepsilon$. Here the value of the threshold ε is chosen according to the "width" of the cloud.

If one searches for two clusters that may be separated by a surface of form (2.1) then the following classification rule applies:

If $F(x) > t$ then data point x belongs to cluster 1; otherwise to cluster 2.

Consider now the transformation of the coordinates and transition to a new space with dimensionality $m = 2d + d(d-1)/2$

$$\begin{gathered} z_1 = x_1, \dots, z_d = x_d;\ z_{d+1} = x_1^2, \dots, z_{2d} = x_d^2; \\ z_{2d+1} = x_1 x_2, \dots, z_{2d+d(d-1)/2} = x_{d-1} x_d. \end{gathered} \tag{2.2}$$

In this new coordinate system the left-hand-side of (2.1) takes the form of the inner product of the transformed data point with coordinates $z_1, z_2, \dots, z_m$ with a parameter vector that may be easily calculated from eqs (2.1), (2.2). The inner products (projections) that correspond with a cloud of data points of parametric form (2.1), are concentrated within a 2ε interval (see above). As for the cluster structure described above, it may be found in the one dimensional distribution of the projections of data points.

Naturally, such a transformation may be performed for polynomials of any order (not only for the order two). Note that such a transformation was used in algorithm for optimal margin classifier [9].

2–2. Consider a number of special cases.

1. Let (2.1) represent a hyperplane, i.e. all values $a_{ij} = 0$. In this case there is no need to make a transition to a new space; we should consider the usual projection on a vector which is orthogonal to the hyperplane and search for the optimal one.

The dimensionality of the space of search is $d-1$. In the case of cluster analysis the considered situation means the search for linearly separable clusters. This case will be considered in Section 3.

2. Let (2.1) represent a hypersphere with center $(x_{10}, x_{20}, \ldots, x_{d0})$. In this case eq. (2.1) may be written in the form $\sum_{i=1}^{d}(x_i - x_{i0})^2 = r^2$. Clearly, in the considered case the value of the projection is equivalent to the Euclidian distance between a data point with coordinates x_i, and center x_{i0}, $i = 1, 2, \ldots, d$. So, one should calculate Euclidean Distances between data points and the center of the hypersphere and search for the optimal location of the center. The dimension of the search space is equal to d.

3. Let (2.1) represent a hyperellipsoid with center x_{i0} and let, for simplicity, its semiaxes a_i, $i = 1, 2, \ldots, d$ be oriented along the coordinate axes. In this case eq. (2.1) may be written in the form $\sum_{i=1}^{d}\frac{(x_i - x_{i0})^2}{a_i^2} = 1$. So, in the case of hyperellipsoid one should calculate Mahalanobis Distances between data points and its center taking into account its orientation and search for its optimal location, eccentricity and orientation. It is easy to estimate the dimensionality of the search space: $\dim = 2d + d(d-1)/2 - 1$.

3. LINEARLY SEPARABLE CASE

3–1. In this section we consider an algorithm which detects linearly separable clusters. The algorithm is of hierarchical type and is built on a tree–like principle. As a result, the cluster structure detected by it, has the form of a binary tree; the dendrogram is cut automatically when a statistically significant number of clusters is found. The first version of this algorithm is described in [4]. The algorithm consists of the following steps.

- Perform a projection pursuit using the clustering–indicating function as an index until a requested statistical bound is achieved (first level of validation). Repeat this process several times to obtain a list of clustering candidates.
- Use a statistical score to choose the best clustering candidate (second level of validation). If the highest score passes a validation bound, the data are split into two clusters according to the best candidate. Otherwise, it indicates the absence of (additional) clustering. Continue the same procedure with each of these two clusters.

A second level of validation is needed to ensure the stability of the algorithm and to improve its performance. In the next two paragraphs we present some details of these steps.

3–2. For the clustering–indicating function, we have chosen an enhanced version of the Fisher criterion [4]. If a set of n one–dimensional observations $x_1, x_2, \ldots, x_n$ is partitioned into two parts by a threshold t with n_1 observations in one part and n_2 in the other, one can estimate the average values in each part m_1 and m_2 and the variances σ_1^2 and σ_2^2 for the distributions on the left and right parts respectively. The clustering–indicating function of this partition is

$$S = \frac{n_1 n_2 (m_1 - m_2)^2}{(n_1 + n_2)(\sigma_1^2 + \sigma_2^2)}. \tag{3.1}$$

Changing the threshold t one may try all n possible partitions of the sample set and obtain the maximum value of (3.1) $\max S$.

While working with clustering data against a noisy background (like objects on a 2D noisy image) we need a more robust clustering–indicating function than (3.1). In these circumstances, in the numerator of (3.1), we use the square difference of percentiles, instead of the square difference of the mean values (e. g. the value of 90 % percentile for the left part and, symmetrically, the value of 10 % percentile for the right part); in the denominator of (3.1), we use the respective median values of square deviations of the data points on the left and right parts from the corresponding medians, instead of variances.

To establish a statistical significance threshold for the value $\max S$ one can generate n samples from the uniform (or Gaussian) distribution (on a segment) and find $\max S$ for the samples. By repeating this procedure many times, we can estimate the set of values B_α such that for the random sample set $\mathrm{Prob}(\max S < B_\alpha) = \alpha$, where α is a preestablished significance level. The values B_α depend on n and may be found as preprocessing.

Projection pursuit was performed with the help of a stochastic optimization procedure called simulating annealing which is described in [4]. In later stages of research we moved on to another procedure of stochastic optimization called evolutionary programming [7].

3–3. The list of clustering candidates obtained as a result of a number of activations of the searching phase is analysed with the help of a bootstrapping procedure [6]. This procedure serves two purposes: the final selection of the best clustering candidate and (if the candidate is statistically significant) determination of the optimal split point of the clustering. The latter problem is connected to the fact that, generally speaking, the threshold t that corresponds to $\max S$ may not represent the optimal split point.

We repeat the following procedure p times:

1. Randomly choose a subsample of m points from n input data points.

2. Compute $\max S$ and the corresponding separation point t on the subsample.

If $\max S$ rejects the null hypothesis, i.e. if $\max S > B_{\alpha^*}$, then increase the content of the accumulator cell *hits* by one.

3. Choose the final separation point t_0 according to the best value $\max S$.

As discussed in [4], if enough subsamples are generated, there is a high probability of reaching a situation in which the final separation point t_0 indicates a good clustering. Note that the threshold t that corresponds to the maximum of the robust version of (3.1) (see above) gives a much better representation of the optimal split point.

As mentioned above, another advantage of this approach is the ability to validate the partition suggested by maximizing S. The global clustering structure should correspond to a large number of hits, hence, $\frac{\text{hits}}{p}$ could be used for internal validation (in addition to null hypothesis testing).

3–4. The performance of the algorithm was tested on artificial and real data. Artificial clustering data were created by performing a version of the Neyman-Scott process [6]: m clusters are randomly created in a d-dimensional unit hypercube. The algorithm was tested for the values of $d = 2 - 64$. For $m = 3$ the algorithm worked well for the whole range of d. The algorithm also demonstrated very good result for two known clustering problems: Iris plant data and wine recognition data [4].

4. GENERAL CASE

4–1. In the previous section we described the clustering–indicating function (3.1) and its robust version. This function may be used as a fitness function for any kind of cluster types described in Section 2. In this paper we also consider clusters in the form of clouds of data points elongated around some lines or surfaces. For detecting such clusters we should introduce fitness function of another type.

4–2. As seen from Section 2, a high concentration of projected data points within a small interval on the axis of projection may be an indication of the presence of a cloud of data points around a surface (a line) of a given parametric form. We introduce a fitness function that detects such highly populated intervals. Consider the interval of possible projection values and divide this interval into a number (e. g. 50) of small subintervals. We want to find a set of neighboring subintervals such that (1) the density of the projected data points is significantly higher than the average density; (2) the number of data points projected into the set is high enough. Choose the following function:

$$f_{ik} = \frac{(\sum_{j=i}^{i+k-1} n_j)^{1.5}}{k}; \quad f = \max_{i,k}[f_{ik}]. \tag{4.1}$$

The numerator here is equal to the number of data points projected in k consecutive subintervals beginning with the number i to the power 1.5.

If the value f is significant then we may have discovered a surface (a curve) of the given parametric form. One can easily see that the fitness function f introduced by (4.1) is robust with respect to the scattering of data points around the ideal curve (surface) as well as with respect to general level of noise in the image and the image clutter constituted by its structure.

4–3. The search for the maximum of the fitness functions (3.1), (4.1) is performed with the help of a stochastic optimization algorithm known as evolutionary programming [7]. We use the simplest version of this algorithm with a population of $p = 20$ candidate vectors in the search space (see Section 2) and with a probabilistic selection of candidates for the next generation. The number of generations that should be produced to reach an adequate solution depends on the kind of parametric form (less for lines, more for ellipses) and on the complexity of the whole data space. In our experiments $25-100$ generations were produced.

After the optimal projection is fixed the algorithm continues to work as follows. If one detects a clustering structure with the help of the fitness function (3.1), the algorithm works according to tree-like structure as described in Section 3. If one detects a cloud of data points of a given geometric form with the help of the fitness function (4.1) the following operations are performed. Find the number of selected consecutive subintervals k and their location i. Single out the set of data points that were projected into these subintervals. Now decide whether this set of data points corresponds with the given geometric primitive or whether it is an accidental gathering of data points that does not represent a primitive of a given form. To make such a decision analyze the selected set of data points according to the following criteria: the number of data points in the set should be higher than a threshold; the arrangement of the data points should be sufficiently uniform, etc. If the decision is positive, remove this set of data points from the image and continue to search the other geometric primitives.

4–4. Several experiments and applications of the cluster algorithm for linearly separable case are described in [4] and in Section 3. We also applied the suggested technique to a number of problems where there were either clusters that could not be linearly separated (but could be separated with the help of a circle or an ellipse), or there were clusters which were a "fuzzy" version of rings, circles or ellipses. In all the cases all these clusters were found. Figures representing problems of such kind are presented in [2].

(Received December 18, 1997.)

REFERENCES

[1] D. H. Ballard and C. H. Brown: Computer Vision. Prentice–Hall, Engl. Cliffs, NJ 1982.

[2] V. L. Brailovsky and M. Har-even: Detecting a data set structure through the use of nonlinear projections and stochastic optimization. In: Proc. 1st IARP TC1 Workshop on Statistical Techniques in Pattern Recognition, Prague 1997, pp. 7–12.

[3] B. S. Everitt: Cluster Analysis. Wiley, New York 1974.

[4] M. Har–even and V. L. Brailovsky: Probabilistic validation approach for clustering. Pattern Recognition Lett. *16* (1995), 1189–1196.

[5] A. K. Jain and R. C. Dubes: Algorithms for Clustering Data. Prentice Hall, NJ 1988.

[6] A. K. Jain and J. V. Moreau: Bootstrap technique in cluster analysis. Pattern Recognition *20* (1987), 547–568.

[7] V. W. Porto, D. B. Fogel and L. J. Fogel: Alternative neural network training methods. IEEE Expert *10* (1995), 3, 16–22.

[8] S. C. Shapiro (ed.): Encyclopedia of Artificial Intelligence, Vol. 1, 'Clustering'. Wiley, New York 1990, pp. 103–111.

[9] V. N. Vapnik: The Nature of Statistical Learning Theory. Springer Verlag, Berlin 1995.

Prof. Victor L. Brailovsky and M. Sc. Michael Har-even, Department of Computer Science – Tel Aviv University, Tel Aviv, Ramat Aviv, 69978. Israel.
e-mail: brail@math.tau.ac.il