The Piecewise Linear Classifier DIPOL92[*]

Barbara Schulmeister, Fritz Wysotzki

Fraunhofer-Institute for Information and Data Processing
Branch Lab for Process Optimisation
Kurstraße 33, D-10117 Berlin, Germany

Abstract. This paper presents a learning algorithm which constructs an optimised piecewise linear classifier for n-class problems.
In the first step of the algorithm initial positions of the discriminating hyperplanes are determined by linear regression for each pair of classes. To optimise these positions depending on the misclassified patterns an error criterion function is defined. This function is minimised by a gradient descent procedure for each hyperplane separately. As an option in the case of non-convex classes, a clustering procedure decomposing the classes into appropriate subclasses can be applied. The classification of patterns is defined on a symbolic level on the basis of the signs of the discriminating hyperplanes.

1 Introduction

The introduced algorithm can be considered

- as a statistical approach with the special option of clustering classes and carrying out the subtask of classification on a symbolic level or
- as a neural network approach with the special choice of the initial conditions of hidden units (number and weights) and fixed Boolean functions in the second layer.

These two ways of interpretation of the algorithm emphasize that there are many relations between statistical and neural network algorithms. In statistical and neural network algorithms often a lot of parameters have to be determined to obtain an optimal solution of the classification problem. Especially in neural network design the choice of these parameters remains open.
The use of the introduced algorithm requires only the number of clusters of the classes to be fixed. With the help of clustering it is possible to gain an insight into the position and structure of the classes.
The algorithm is implemented as the program DIPOL92 (DIscrimination and POst Learning). It was part of the Esprit project STATLOG [1]. The aim of the project was to provide a review of different approaches to classification, compare their performance on a wide range of datasets and draw conclusions on their applicability to real-world problems, e.g., technical and medical diagnosis, image

[*] This work was supported by the German Ministry of Research and Technology within the project WISCON

recognition and credit datasets. DIPOL92 came off very well in comparison to the other 21 algorithms. Taking the top six places across all datasets it has more occurences than all other algorithms.

2 Pairwise Linear Regression

Suppose that $X \subset R^m$ (m-dimensional Euclidean space) is the set of classified patterns $\mathbf{x} = (x_1, \ldots, x_m)$. Linear regression is used for discrimination of two classes of patterns k_1 and k_2 by defining the dependent variable b in the following manner :

$$\text{if } \mathbf{x} \epsilon k_1, \text{ then } b = +1, \text{ if } \mathbf{x} \epsilon k_2, \text{ then } b = -1$$

Let W be the linear regression function $W : X \rightarrow R$ with $W(\mathbf{x}) = w_0 + w_1 x_1 + \ldots + w_m x_m$. Then a pattern $\mathbf{x}$ is correctly classified if

$$W(\mathbf{x}) > 0 \text{ for } \mathbf{x} \epsilon k_1, \; W(\mathbf{x}) < 0 \text{ for } \mathbf{x} \epsilon k_2.$$

For each pair of classes a discriminating regression function is calculated.

3 Learning Procedure

For all misclassified patterns, the squared distances from the corresponding decision hyperplane multiplied with the costs for these misclassifications are summed up :

Suppose $W = 0$ defines the decision hyperplane between the classes k_1 and k_2. Then let m_1 be the set of all misclassified patterns of class k_1, i.e., $x \epsilon k_1$ and $W(\mathbf{x}) < 0$, let m_2 be the set of all misclassified patterns of class k_2, i.e., $x \epsilon k_2$ and $W(\mathbf{x}) > 0$, and let $cost(k_i, k_j)$ be the costs of misclassification of class k_i into class k_j :

$$F(W) = cost(k_1, k_2) * \sum_{\mathbf{x} \epsilon m_1} \frac{W(\mathbf{x})^2}{||\mathbf{x}||^2} + cost(k_2, k_1) * \sum_{\mathbf{x} \epsilon m_2} \frac{W(\mathbf{x})^2}{||\mathbf{x}||^2}$$

The learning procedure consists of minimising the criterion function by a gradient descent algorithm seperately for each decision surface. Since the gradient $\nabla_W F(W)$ of $F(W)$ with respect to W defines the direction of maximum increase in F, it is used to form an iterative minimisation procedure:

$$W^{(n+1)} = W^{(n)} + \rho_n \nabla_W F(W^{(n)})$$

This represents an accumulated correction to the position of the decision hyperplane, or *learning by epoch* as an alternative to stochastic approximation, or *learning by sample.* The costs are included explicitly in the learning procedure.

4 Clustering of Classes

To handle also problems with non–convex, especially non simply–connected class regions, it is suggested to perform a clustering procedure before the linear regression is carried out. To solve the clustering problem a standard minimum–squared–error algorithm is used [4].
From some initial partition of a class k of N patterns into I clusters k_i $(i = 1, \ldots, I)$ with l_i patterns and with mean vectors $\mathbf{s}_i$

$$\mathbf{s}_i = \frac{1}{l_i} \sum_{\mathbf{X} \epsilon k_i} \mathbf{x}$$

the criterion function

$$J = \sum_{i=1}^{I} \sum_{\mathbf{X} \epsilon k_i} ||\mathbf{x} - \mathbf{s}_i||^2$$

is calculated. If no reasonable initial partition is known a general approach to an initial partition is the following :

$$\mathbf{x}_j \ \epsilon \ k_i, \ (j = 1, \ldots, N, i = 1. \ldots, I), \ if \quad i \ = j \ (mod \ I) + 1$$

Patterns are moved from one cluster to another if such a move will improve the value of the criterion function J. The mean vectors are updated after each pattern move. Like hill–climbing algorithms in general, these approaches guarantee local but not global optimisation. Different initial partitions and the order in which the training patterns are selected can lead to different solutions.
In case of clustering, the number of two–class problems increases correspondingly.
It is to be noted that by combining the clustering algorithm with the regression technique the number and initial positions of discriminating hyperplanes are a priori fixed (i.e., before learning) in a reasonable manner, even in case that some classes have multimodal distributions (i.e., consist of several subclasses). Thus, a well known bottleneck of neural nets can at least be partly avoided.

5 Classification Procedure

When discriminating hyperplanes were computed then any pattern $\mathbf{x}$ (member of the training set or not) can be classified, i.e., the class can be predicted.
For the pairwise discrimination of c classes $C = c(c-1)/2$ hyperplanes W^i are calculated (in case of clustering the number c is changed into $c + c_{clust}$).
The following C–dimensional vector $\mathbf{V}_k$ is defined for each class k : if the function W^i discriminates the classes k_1 and k_2, then the i-th component $V_{k,i}$ is equal to 1, if $k = k_1$, equal to -1, if $k = k_2$, and equal to 0 in all other cases. These vectors contain the coded information about the convex class or subclass regions depending on the C functions W^i.
On the basis of the discriminant functions the vector function sw is defined for each pattern $\mathbf{x}$

$$sw : X \rightarrow \{1, 0, -1\}^C$$

with the components

$$(sw(\mathbf{x}))_i = sign(W^i(\mathbf{x})).$$

The vector $sw(\mathbf{x})$ contains the coded information about the position of the pattern $\mathbf{x}$ depending on the C functions W^i.
For each class k the function (G is the set of integers)

$$S_k : X \rightarrow G$$

is defined as the scalar product S_k of the two coded vectors $\mathbf{V}_k$ and $sw(\mathbf{x})$

$$S_k(\mathbf{x}) = \sum_{i=1}^{C} V_{k,i} * (sw(\mathbf{x}))_i.$$

A pattern $\mathbf{x}$ is uniquely classified by the discriminating hyperplanes W^i into the class k if

$$S_k(\mathbf{x}) = c - 1,$$

i.e., with respect to the $c - 1$ hyperplanes, which discriminate the class k from the other $c - 1$ classes, the pattern $\mathbf{x}$ is placed in the halfspace belonging to class k (the coded vectors $\mathbf{V}_k$ and $sw(\mathbf{x})$ have the same value (+1 or -1) for all components i with $V_{k,i} \neq 0$). For all other classes j, $j \neq k$, $S_j(\mathbf{x}) < c - 1$ is valid, because at least with respect to the hyperplane, which discriminates class j from class k the pattern $\mathbf{x}$ is placed in the halfspace of class k (the coded vectors $\mathbf{V}_k$ and $sw(\mathbf{x})$ do not have the same value (+1 or -1) for all components i with $V_{k,i} \neq 0$).
A pattern $\mathbf{x}$ is not uniquely classified if

$$\max_j S_j(\mathbf{x}) < c - 1.$$

In this case the pattern is classified on the basis of the minimum of the distance to the class regions.

References

1. Michie,D., Spiegelhalter,D., Taylor,C. (Eds.): Machine Learning, Neural and Statistical Classification. Results of the Esprit project STATLOG (to appear)
2. Meyer-Brötz, G., and Schürmann, J.: Methoden der automatischen Zeichenerkennung. Akademie-Verlag, Berlin (1970)
3. Unger, S., Wysotzki, F.: Lernfähige Klassifizierungssysteme. Akademie-Verlag, Berlin (1981)
4. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley (1973)