Two Classes Linear Discrimination, a Min-Max Approach

Philippe Castagliola, Bernard Dubuisson

Pattern Recognition Letters 10 (1989) 281–287
North-Holland

November 1989

Two classes linear discrimination A min-max approach

P. CASTAGLIOLA and B. DUBUISSON
Université de Technologie de Compiègne, URA CNRS. 817, B.P. 649, 60206 Compiegne Cedex, France

Received 11 April 1989
Revised 22 June 1989

Abstract: In this paper we propose a new dichotomic linear discrimination algorithm based on a criterion recursively defined from the min and max functions. This criterion allows us, using a gradient algorithm, to find two hyperplanes which split two classes in an optimal way. We give two examples of the possibilities of the method.

Key words: Linear discrimination, min and max function, gradient algorithm.

1. Introduction

The main goal of pattern recognition is to assign a new observed vector into one class out of M known classes. This decision can be made with the help of discriminant surfaces obtained from the knowledge of sample vectors in classes.

Let us suppose that we have to discriminate only between two classes (dichotomy) in $\mathbb{R}^d$. Different possibilities exist; the optimum one is the Bayes discriminator. Associate costs are $\{0,1\}$ (cost of good classification is 0, cost of an error is 1), the optimum decision is associate $\boldsymbol{x}$ to ω_i if:

$$p(\omega_i \mid \boldsymbol{x}) = \max p(\omega_j \mid \boldsymbol{x}), \quad j = 1,2.$$

$p(\omega_i \mid \boldsymbol{x})$ is the a posteriori probability of class ω_i given $\boldsymbol{x}$.

If these probabilities are unknown, they have to be estimated using samples of different classes. This set of samples is called the learning set.

Another possibility is to first decide on the form of the boundary between the two classes. One of the most studied discriminant functions is the linear one. Classes are said to be linearly separable if we can compute the equation of a hyperplane in $\mathbb{R}^d$ separating the two classes.

The first linear discrimination algorithm was developed by Rosenblatt (Rosenblatt, 1957) in the mid 1950s, and was inspired from the neuronal model of Mc Culloch and Pitts (Mc Culloch & Pitts, 1943). Its usual name is the Perceptron. This algorithm is considered as a special case of the reward-and-punishment procedure. Unfortunately, if the classes are not linearly separable, this algorithm does not converge and no indication of this fact is given to the user.

The Ho–Kashyap procedure (Ho & Kashyap, 1965) gives a better solution to this problem. It always converges, indicating that the classes are linearly separable or not; if they are, this algorithm furnishes a hyperplane equation.

New solutions are currently proposed to solve non-linear separability by using the connexionnist approach

(Widrow, Hoff, 1960) (neuron models). Although this non-probabilistic approach has interesting new developments, we are not interested in this approach.

Our goal is to find two parallel hyperplanes $\mathbb{P}_1$ and $\mathbb{P}_2$ separating the two classes in an optimal way. This problem is completely equivalent to find an optimal unit vector $\boldsymbol{u}^*$, orthogonal to $\mathbb{P}_1$ and $\mathbb{P}_2$, and maximising a criterion $d(\boldsymbol{u})$ which represents a signed distance between the classes.

We will define this signed distance from the difference between some min and max functions, and use it to find the optimal solution.

If the signed distance $d(\boldsymbol{u}^*)$, corresponding to the optimal solution is positive, we will conclude that the classes are linearly separable; if it is not, we will conclude that they are not linearly separable.

2. Principle of the method

Let ω_1 and ω_2 be two classes of $\mathbb{R}^d$. We want to discriminate between these two classes.

The learning set Σ is composed of n vectors, $n1$ from ω_1 and $n2$ from ω_2:

$$\omega_1 = \{\boldsymbol{x}_{1,1}, \ldots, \boldsymbol{x}_{1,n1}\} = \{\boldsymbol{x}_{1,i}\}_1^{n1} \quad \text{and} \quad \omega_2 = \{\boldsymbol{x}_{2,1}, \ldots, \boldsymbol{x}_{2,n2}\} = \{\boldsymbol{x}_{2,i}\}_1^{n2}.$$

Let $\boldsymbol{u}$ be a unitary vector orthogonal to $\mathbb{P}_1$ and $\mathbb{P}_2$ (see Figure 1). And let $\hat{x}_{i,k} = \boldsymbol{u}^t \boldsymbol{x}_{i,k}$.

Now we have to choose a criterion for our problem. This criterion represents a signed distance:

$$d(\boldsymbol{u}) = s(\boldsymbol{u}) - b(\boldsymbol{u}) \tag{1}$$

where

$$s(\boldsymbol{u}) = \min(\hat{x}_{2,1}, \ldots, \hat{x}_{2,n2}) = \min(\hat{x}_{2,i})_1^{n2}, \qquad b(\boldsymbol{u}) = \max(\hat{x}_{1,1}, \ldots, \hat{x}_{1,n1}) = \max(\hat{x}_{1,i})_1^{n1}.$$

Let $\boldsymbol{u}^*$ be the direction which maximises the criterion $d(\boldsymbol{u})$. Two opposite cases may happen:

Case 1: $d(\boldsymbol{u}^*) > 0$. Then ω_1 and ω_2 are linearly separable (Figure 2a). We can easily find two parallel hyperplanes, orthogonal to the direction $\boldsymbol{u}^*$, with the equations:

$$\mathbb{P}_1: \boldsymbol{x}^t\boldsymbol{u}^* - b(\boldsymbol{u}^*) \quad \text{and} \quad \mathbb{P}_2: \boldsymbol{x}^t\boldsymbol{u}^* - s(\boldsymbol{u}^*).$$

If the classes are interchanged, the optimal vector $\boldsymbol{u}^*$ becomes $-\boldsymbol{u}^*$ and the difference $d(\boldsymbol{u}^*)$ is still positive. For example, if we have first $b(\boldsymbol{u}^*) = 2$ and $s(\boldsymbol{u}^*) = 3$ then $d(\boldsymbol{u}^*) = 1 > 0$. After interchanging the classes, $\boldsymbol{u}^*$ becomes $-\boldsymbol{u}^*$, and $b(\boldsymbol{u}^*) = -3$, $s(\boldsymbol{u}^*) = -2$, then $d(\boldsymbol{u}^*) = -2 - (-3) = 1$ is still positive.

Case 2: $d(\boldsymbol{u}^*) < 0$. Then ω_1 and ω_2 are not linearly separable with a zero error probability (Figure 2b). The same considerations can be reformulated about the interchange of ω_1 and ω_2.

Let us first consider the functions min and max used above in the definition of the functions $s(\boldsymbol{u})$ and $b(\boldsymbol{u})$.

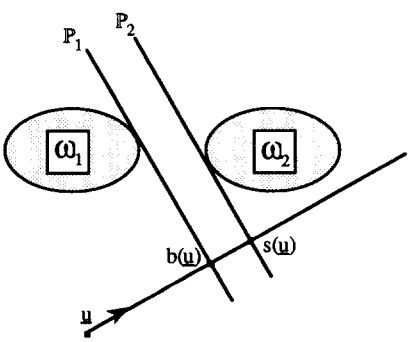

Figure 1.

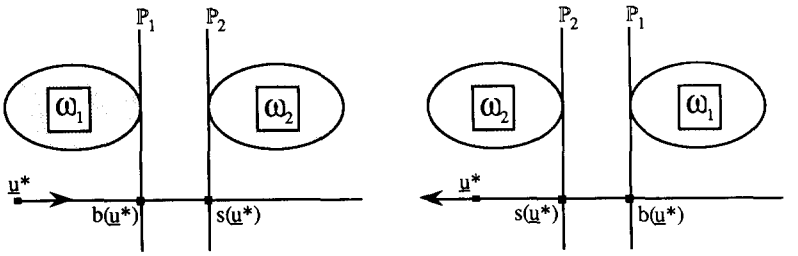

Figure 2a.

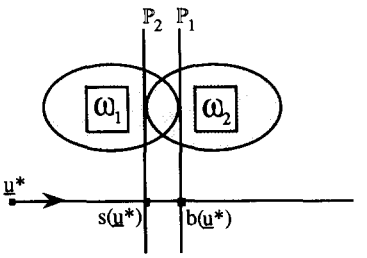

Figure 2b.

3. Definition and computation of MIN and MAX

3.1. *The MIN function*

Let $E = \{a_1,\ldots,a_n\}$, be a set of n real values. The min function is defined by the following:

$$\min\{a_1,\ldots,a_n\} = a_i \quad \text{if } a_i < a_j \quad \text{for } j = 1,\ldots,n, \quad j \neq i.$$

If the cardinality of E is one, it is obvious that $\min(a_1) = a_1$. Let us choose the cardinality of E equal to 2. It is also obvious that:

$$\min(a_1,a_2) = \frac{1}{2}(a_1 + a_2 - |a_1 - a_2|).$$

So recursively we can write when card $E = n$:

$$\min(a_1,\ldots,a_n) = \frac{1}{2}(\min(a_1,\ldots,a_{n-1}) + a_n - |\min(a_1,\ldots,a_{n-1}) - a_n|).$$

Using the notation $\min(a_1,\ldots,a_n) = \min(a_i)_1^n$:

$$\min(a_i)_1^1 = a_1, \qquad \min(a_i)_1^n = \frac{1}{2}(\min(a_i)_1^{n-1} + a_n - |(\min(a_i)_1^{n-1} - a_n|. \tag{2}$$

3.2. *The MAX function*

The same considerations may be used for the max function.

$$\max(a_1) = a_1, \qquad \max(a_1,a_2) = \frac{1}{2}(a_1 + a_2 + |a_1 - a_2|),$$

$$\max(a_i)_1^1 = a_1, \qquad \max(a_i)_1^n = \frac{1}{2}(\max(a_i)_1^{n-1} + a_n + |(\max(a_i)_1^{n-1} - a_n|). \tag{3}$$

4. Computation of the optimal vector u^*

4.1. A gradient algorithm

We search for a unit vector u^* that maximizes $d(u)$. This optimisation can be solved using a fixed step gradient:

$$u_{n+1} = u_n + p \cdot \frac{\partial}{\partial u} d(u), \qquad p \text{ being the step to be fixed.}$$

If we use this method directly, the vector u_n tends towards the right direction, but unfortunately its norm tends towards $+\infty$ (unconstrained solution).

We used the unitary norm constraint: $\|u^*\| = 1$. So, we must normalize the vector u after each gradient algorithm iteration. Thus, the modified gradient algorithm is:

$$v_n = u_n + p \cdot \frac{\partial}{\partial u} d(u), \qquad u_{n+1} = \frac{v_n}{\|v_n\|}.$$

4.2. Gradient of $s(u)$, $b(u)$ and $d(u)$

The functions $s(u)$ and $b(u)$ are recursively defined, so are their gradient functions.

Remark. Let $f(x)$ be a function from $\mathbb{R}^n$ to $\mathbb{R}$, derivable from $\mathbb{R}^n$. Theoretically, $(\partial/\partial x)|f(x)|$ is not defined on $f(x) = 0$. So, we will fix by definition:

$$\frac{\partial}{\partial x}|f(x)| = 0 \quad \text{when } f(x) = 0,$$

which is completely equivalent to:

$$\frac{\partial}{\partial x}|f(x)| = \operatorname{sign}(f(x))\frac{\partial}{\partial x} f(x)$$

with

$$\operatorname{sign}(f(x)) = -1 \quad \text{if } f(x) < 0, \quad \operatorname{sign}(f(x)) = 0 \quad \text{if } f(x) = 0, \quad \operatorname{sign}(f(x)) = 1 \quad \text{if } f(x) > 0.$$

We have

$$\frac{\partial}{\partial u} s(u) = \frac{\partial}{\partial u} \min(\hat{x}_{2,i})_1^{n2}$$

and from (2)

$$= \frac{1}{2}\left(\frac{\partial}{\partial u} \min(\hat{x}_{2,i})_1^{n2-1} + x_{2,n2} - \operatorname{sign}(\min(\hat{x}_{2,i})_1^{n2-1} - x_{2,n2})\left(\frac{\partial}{\partial u} \min(\hat{x}_{2,i})_1^{n2-1} - x_{2,n2}\right)\right).$$

So

$$= \frac{1}{2}\left(\frac{\partial}{\partial u}\min(\hat{x}_{2,i})_1^{n2-1}(1 - \operatorname{sign}(\min(\hat{x}_{2,i})_1^{n2-1} - x_{2,n2}) + x_{2,n2}(1 + \operatorname{sign}(\min(\hat{x}_{2,i})_1^{n2-1} - x_{2,n2})\right) \qquad \text{with } \frac{\partial}{\partial u} \min(\hat{x}_{2,i})_1^1 = x_{2,1}. \tag{4}$$

Likewise, we can write:

$$\frac{\partial}{\partial \boldsymbol{u}} b(\boldsymbol{u}) = \frac{\partial}{\partial \boldsymbol{u}} \max(\hat{x}_{1,i})_1^{n2}$$

and from (3)

$$= \frac{1}{2}\left(\frac{\partial}{\partial \boldsymbol{u}} \max(\hat{x}_{1,i})_1^{n1-1} + \boldsymbol{x}_{1,n1} + \operatorname{sign}(\max(\hat{x}_{1,i})_1^{n1-1} - \boldsymbol{x}_{1,n1})\left(\frac{\partial}{\partial \boldsymbol{u}} \max(\hat{x}_{1,i})_1^{n1-1} - \boldsymbol{x}_{1,n1}\right)\right).$$

So

$$= \frac{1}{2}\left(\frac{\partial}{\partial \boldsymbol{u}} \max(\hat{x}_{1,i})_1^{n1-1}(1 + \operatorname{sign}(\max(\hat{x}_{1,i})_1^{n1-1} - \boldsymbol{x}_{1,n1}) + \boldsymbol{x}_{1,n1}(1 - \operatorname{sign}(\max(\hat{x}_{1,i})_1^{n1-1} - \boldsymbol{x}_{1,n1}))\right) \quad \text{with } \frac{\partial}{\partial \boldsymbol{u}} \max(\hat{x}_{1,i})_1^1 = \boldsymbol{x}_{1,1}. \tag{5}$$

Finally, using (1):

$$\frac{\partial}{\partial \boldsymbol{u}} d(\boldsymbol{u}) = \frac{\partial}{\partial \boldsymbol{u}} s(\boldsymbol{u}) - \frac{\partial}{\partial \boldsymbol{u}} b(\boldsymbol{u}). \tag{6}$$

5. Algorithm implementation

We give a simplified skeleton of the min-max algorithm, using (4), (5), (6). We must fix the gradient algorithm step, the parameter ε which controls the calculation accuracy, and the initial unit vector.

Do
  min1 $= \hat{x}_{2,1}$; **grad**$s = \boldsymbol{x}_{2,1}$
  For $i = 2$ **to** $n2$
    min2 $= \hat{x}_{2,i}$
    grad$s = \frac{1}{2}((1 - \operatorname{sign}(\text{min1} - \text{min2}))$ **grad**$s + (1 + \operatorname{sign}(\text{min1} - \text{min2}))\boldsymbol{x}_{2,i})$
    min1 = min(min1,min2)
  End_For
  max1 $= \hat{x}_{1,1}$; **grad**$b = \boldsymbol{x}_{1,1}$
  For $i = 2$ **to** $n1$
    max2 $= \hat{x}_{1,i}$
    grad$b = \frac{1}{2}((1 + \operatorname{sign}(\text{max1} - \text{max2}))$ **grad**$b + (1 - \operatorname{sign}(\text{max1} - \text{max2}))\boldsymbol{x}_{1,i})$
    max1 = max(max1,max2)
  End_For
  grad$d =$ **grad**$s -$ **grad**b
  $\boldsymbol{u} = \boldsymbol{u} + \text{step} \cdot$ **grad**d; $\boldsymbol{u} = \boldsymbol{u}/\|\boldsymbol{u}\|$
  diff1 = min1 − max1
While (diff1 − diff2) $> \varepsilon$
If (diff1 < 0) **then write** "ω_1 and ω_2 are not linearly separable" **End**
Else
  Write "ω_1 and ω_2 are linearly separable"
  Write "Hyperplane equations:" $\boldsymbol{u}^t\boldsymbol{x} -$ min1, $\boldsymbol{u}^t\boldsymbol{x} -$ max1
End_Else

6. Experimental results

The following are two examples of the use of this algorithm. In both configurations, each class contains 100 vectors coming from two normal populations of $\mathbb{R}^2$, where the variance-covariance matrices are set to the identity matrix.

Configuration 1 (Figure 3): linearly separable classes (there exists a hyperplane separating the two classes with a zero estimated error probability).

$$\omega_1\text{: mean} = \begin{pmatrix}0\\0\end{pmatrix}, \qquad \omega_2\text{: mean} = \begin{pmatrix}5\\5\end{pmatrix}.$$

parameters: step $= 10^{-1}$, $\varepsilon = 10^{-8}$, total iteration number $= 52$.

This algorithm converges in an efficient way, concluding that the classes are actually linearly separable (Difference $= +1.674 > 0$). This maximum difference gives a good indication of the absolute gap between the two classes.

It also gives two hyperplane equations (here they are straight lines) that give two linear boundaries for the classes ω_1 and ω_2:

$$\mathbb{P}_1\text{: } 0.711\,x_1 + 0.703\,x_2 - 4.373 = 0, \qquad \mathbb{P}_2\text{: } 0.711\,x_1 + 0.703\,x_2 - 2.699 = 0.$$

Configuration 2 (Figure 4): nonlinearly separable classes.

$$\omega_1\text{: mean} = \begin{pmatrix}0\\0\end{pmatrix}, \qquad \omega_2\text{: mean} = \begin{pmatrix}3\\3\end{pmatrix}.$$

parameters: step $= 10^{-3}$, $\varepsilon = 10^{-8}$, total iteration number $= 1062$.

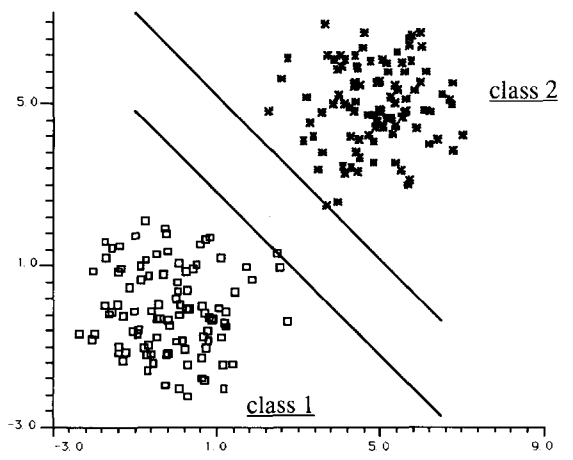

Figure 3. Direction = 7.1118283E-01, 7.0300710E-01. Direction = 7.1119996E-01, 7.0298977E-01. Result after 52 iterations: Min = 4.3736852E + 00, Max = 2.6990202E + 00, Difference = 1.6746650E + 00, Direction = 7.1119996E-01, 7.0298977E-01.

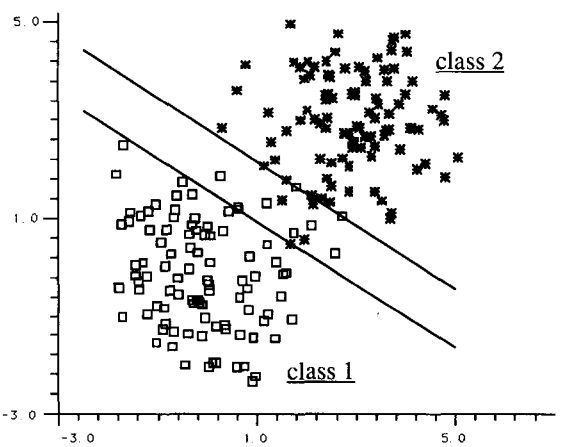

Figure 4. Direction = 5.4219291E-01, 8.4025404E-01. Direction = 5.4171130E-01, 8.4056461E-01. Result after 1062 iterations: Min = 1.3271238E + 00, Max = 2.3541315E + 00, Difference = − 1.0270076E + 00, Direction = 5.4171130E-01, 8.4056461E-01.

We point out that it was necessary to use a small step value (10^{-3}). From that beginning, the algorithm converges, and then it oscillates around the solution. This approximate convergence allows us to conclude that the two classes are not linearly separable (Difference $= -1.027 < 0$). This maximum difference gives a good indication of the classes overlapping.

7. Conclusions

This algorithm based on the new criterion $d(\boldsymbol{u})$, allows us to know, if the two classes are linearly separable or not. In the two cases, it gives a notion about an 'absolute distance' between the classes (gap or overlapping) and if they are linearly separable. It also gives two hyperplane equations which can be used in pattern recognition algorithms.

We noticed that the algorithm behaviour depends on the classes overlapping size: it would converge rapidly if the classes do not overlap.

In practice, we have to set the step value: if it is too big, the algorithm will converge first, and then will oscillate around the solution (with insufficient accuracy); if too small it will converge slowly. This heuristic step setting is the main default of this algorithm.

References

Rosenblatt, F. (1957). The Perceptron: a perceiving and recognizing automaton. Project PARA, Cornell Aeronaut. Lab. Rep. VG-1196-G-4.

Ho, Y.C. and Kashyap, R.L. (1965). An algorithm for linear inequalities and its applications. *IEEE Trans. Electronic Computers* 14 (5), 683–688.

Mc Culloch, W.S. and Pitts, W. (1943). A logical caluclus of the ideas immanent in nervous activity. *Bull. Math. Biophysics* 5, 115–133.

Widrow, B. and Hoff, M.E. (1960). Adaptative switching circuits. 1960 *IRE WESCON Convention record*. New York, 96–104.