ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 179-184

A FAST BAYESIAN ALGORITHM FOR BOOLEAN FUNCTIONS SYNTHESIS BY MEANS OF PERCEPTRON NETWORKS

Andreu CATALA MALLOFRE
Departament ESAII, Universitat Politècnica de Catalunya, Pau Gargallo, 5
08028 BARCELONA, Catalunya (Spain), andreu@esaii.upc.es
Josep AGUILAR-MARTIN
ETSEIT-UPC, dept ESAII, Colom 11, 08222 TERRASSA, Catalunya (Spain)
LAAS-CNRS, 7 Av. du Colonel Roche, 31077 TOULOUSE
(France),tiger@upc.es
Bernardo MORCEGO SEIX
Departament ESAII, Universitat Politècnica de Catalunya, Pau Gargallo, 5
08028 BARCELONA, Catalunya (Spain)
Nuria PIERA i CARRETE
Departament Matemàtica 2, Universitat Politècnica de Catalunya,
Pau Gargallo, 5; 08028 BARCELONA, Catalunya (Spain)

ABSTRACT: The purpose of this paper is to introduce the mathematical concept of Bayesian-separability applied to logical functions that leads to a direct calculation of the Perceptron weights in order to synthesise Boolean functions avoiding iterations. As far as Bayesian-separability is slightly more restricted than linear separability, a second result is its extension to the concept of m-separability in order to cover most of the linearly separable Boolean functions. Finally, an extension of the present methodology to deal with any Boolean function is given. This procedure gives rise to a Perceptron network structure.

1.Bayesian-separability

Let R^n be the n-dimensional Euclidean space, as usual, and the pair $B=\{0,1\}$ its Boolean basic set; B^n is the set of the vertexes of the unit cube in R^n, and the cardinal of B^n is 2^n.

Let $\underline{x} \in B^n$ be the vector $\{x_1,..,x_n\}$ that represents a vertex of B^n and $\forall i$, $x_i=0$ or $x_i=1$.

Any subset $A \subset B^n$ of cardinal $n_A < 2^n$ defines a partition of B^n and its characteristic function is called a logic Boolean function L_A

$$\forall \underline{x} \in B^n, \quad L_A(\underline{x}) = 1 \text{ if } \underline{x} \in A \text{ and } L_A(\underline{x}) = 0 \text{ if } \underline{x} \notin A, \text{ (or } \underline{x} \in A')$$

Let $\underline{W} \in R^{n+1}$ be any vector $\{w_0,w_1,..,w_n\}$ and $\underline{X} \in R^{n+1}$ the vector $\{1, x_1,..,x_n\}$, then the function $y=\text{sign}[\underline{W}^T\underline{X}]$ is called a Perceptron. Equation $W^T\underline{X} = 0$ gives the Perceptron hyper plane $P_W \subset R^n$ [1].

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 179-184

DEFINITION 1

A logic Boolean function L_A is called **linearly-separable** (L-separable) if

$$\exists\, \underline{W} \in R^{n+1} \text{ such that } \forall \underline{x} \in B^n \quad L_A(\underline{x}) = \mathrm{sign}[\underline{W}^T \underline{X}]$$

From the point of view of the Bayesian probability it is possible to calculate a separation hyper plane P_W following the argument below,

DEFINITION 2

The conditional probability for each component "i" will be,

$$P(x_i=1 / \underline{x} \in A) = \rho_i\,, \quad P(x_i=0 / \underline{x} \in A) = 1-\rho_i\,,$$
$$P(x_i=1 / \underline{x} \in A') = \eta_i\,, \quad P(x_i=0 / \underline{x} \in A') = 1-\eta_i$$

If $\xi_i \in \{0,1\}$, then we could write,

$$P(x_i=\xi_i / \underline{x} \in A) = \rho_i^{\xi_i} (1-\rho_i)^{(1-\xi_i)}$$

If the vector components are independent, the probability for each vector knowing its class membership will be,

$$P(x_1=\xi_1,\dots, x_n=\xi_n / \underline{x} \in A) = \prod_{i=1}^{n} P(x_i=\xi_i / \underline{x} \in A) = \prod_{i=1}^{n} \rho_i^{\xi_i}(1-\rho_i)^{(1-\xi_i)}$$

$$P(x_1=\xi_1,\dots, x_n=\xi_n / \underline{x} \in A') = \prod_{i=1}^{n} P(x_i=\xi_i / \underline{x} \in A') = \prod_{i=1}^{n} \eta_i^{\xi_i}(1-\eta_i)^{(1-\xi_i)}$$

Making use of the maximum likelihood theorem, we can employ the statistical mean, ρ_i for A, and η_i for A', for each component [2],

$$\text{for } i=1\dots n, \quad \rho_i = \frac{1}{n_A} \sum_{\underline{x} \in A} x_i, \quad \text{and} \quad \eta_i = \frac{1}{2^n - n_A} \sum_{\underline{x} \notin A} x_i$$

Now if we are interested in the ownership probability of each vector $\underline{x}$ to class A $P(\underline{x} \in A / x_1=\xi_1,\dots,x_n=\xi_n)$, and to class A' $P(\underline{x} \in A' / x_1=\xi_1,\dots,x_n=\xi_n)$. We could apply Bayes theorem on both classes, equalising the resulting expressions, and after a few algebraic manipulations we will obtain the separation hyper plane between the members of class A and the members of class A', as done in [3],

$$\sum_{i=1\dots n} \xi_i \,\mathrm{Log}\frac{\rho_i(1-\eta_i)}{\eta_i(1-\rho_i)} + \mathrm{Log}\frac{n_A}{2^n - n_A} + \sum_{i=1\dots n} \mathrm{Log}\frac{1-\rho_i}{1-\eta_i} = 0 \tag{1.1}$$

With this expression we retrieve the Perceptron structure using as weights,

$$w_i = \mathrm{Log}\frac{\rho_i(1-\eta_i)}{\eta_i(1-\rho_i)} \quad \text{and} \quad w_o = \mathrm{Log}\frac{n_A}{2^n - n_A} + \sum_{i=1\dots n} \mathrm{Log}\frac{1-\rho_i}{1-\eta_i} \tag{1.2}$$

then a logic Boolean function L_A is called **Bayesian-separable,** (B-separable) [4] if $\forall \underline{x} \in B^n, L_A(\underline{x}) = \mathrm{sign}[\underline{W}^T \underline{X}]$.

It is not difficult to see that for $n \le 3$ any L-separable Boolean function is B-separable [4]. One can understand the high interest of B-separability as, instead of using iterative learning algorithms, the mere calculation of statistical means gives the weights of the Perceptron, thus highly simplifying the synthesis of these functions.

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 179-184

Unfortunately for n>3 there are L-separable logic Boolean functions not B-separable, which means that the hyper plane obtained by (1.2) is unable to correctly separate the classes A and A' obtaining some misclassified vertexes. As an example the following L-separable function shows one misclassified vertex, $\underline{x}_1$, let us see it,

	x_1	x_2	x_3	x_4	L_A		x_1	x_2	x_3	x_4	L_A
$\underline{x}_1$	0	0	0	0	0	$\underline{x}_9$	1	0	0	0	0
$\underline{x}_2$	0	0	0	1	1	$\underline{x}_{10}$	1	0	0	1	0
$\underline{x}_3$	0	0	1	0	1	$\underline{x}_{11}$	1	0	1	0	1
$\underline{x}_4$	0	0	1	1	1	$\underline{x}_{12}$	1	0	1	1	1
$\underline{x}_5$	0	1	0	0	1	$\underline{x}_{13}$	1	1	0	0	0
$\underline{x}_6$	0	1	0	1	1	$\underline{x}_{14}$	1	1	0	1	0
$\underline{x}_7$	0	1	1	0	1	$\underline{x}_{15}$	1	1	1	0	1
$\underline{x}_8$	0	1	1	1	1	$\underline{x}_{16}$	1	1	1	1	1

Table 1

Using this function table we can easily obtain easily the membership probability for each class,

$$P(\underline{x}_1 \in A / \underline{x}_1) = ((7/11)*(3/11)*(5^2/11^2)*(11/16))/0.748 = 0.518$$

$$P(\underline{x}_1 \in A' / \underline{x}_1) = ((1/5)*1*(3^2/5^2)*5)/0.748 = 0.482$$

Therefore $P(\underline{x}_1 \in A / \underline{x}_1) > P(\underline{x}_1 \in A' / \underline{x}_1)$ and the Bayesian hyper plane will misclassify this vertex. The preceding example allows us to introduce the concept of erroneous vertexes.

DEFINITION 3

Given a Boolean function $L_A(\underline{x})$ a vertex or vector $\underline{x}$ is said to be erroneous or misclassified by B-separable if,

i) $L_A(\underline{x}) = 0$ and $P(\underline{x} \in A / \underline{x}) > P(\underline{x} \in A' / \underline{x})$;

ii) $L_A(\underline{x}) = 1$ and $P(\underline{X} \in A / \underline{x}) < P(\underline{x} \in A' / \underline{x})$

A more powerful concept, based on 'm' repetitions of erroneous vertexes called m-separability [4] is presented in the next section.

2. M-separability

For a given non-B-separable logic Boolean function, the direct application of B-separability yields a Perceptron such that for the vertexes $\underline{x}^* \in E$ it gives an erroneous answer, $\forall \underline{x}^* \in E \; L_A(\underline{x}^*) \neq \text{sign}[\underline{W}^T \underline{X}^*]$

The subset $E \subset B^n$ is split in $E = E_+ \cup E_-$, where E_+ corresponds to the erroneous vertexes such that $L_A(\underline{x}^*) = 1$ and E_- to those where : $L_A(\underline{x}^*) = 0$. The cardinals of E_+ and E_- are respectively n_+ and n_- . A new Perceptron $\underline{W}_m$ can be obtained by modification of the statistical means calculation as follows,

$$\rho_i = \frac{1}{n_A + mn_+}\left(mn_+ \sum_{\underline{x} \in E_+} x^*_i + \sum_{\underline{x} \in A} x_i\right), \quad \eta_i = \frac{1}{2^n - n_A + mn_-}\left(mn_- \sum_{\underline{x} \in E_-} x^*_i + \sum_{\underline{x} \notin A} x_i\right) \quad (2.1)$$

It can be seen from the above expressions that the new statistical means include the repetition of erroneous vectors m times. The main idea is to move the hyper plane direction precisely to recover the erroneous vertexes by giving them more and more importance in means calculations.

DEFINITION 4

With the same formulae as before for the components of the weight vector $\underline{W}_m$, a logic Boolean function L_A is called **m-Bayesian-separable** (m-separable), if,

$$\forall \underline{x} \in B^n \quad L_A(\underline{x}) = \mathrm{sign}[\underline{W}_m^T \underline{X}] \quad (2.2)$$

Applying m-separability to the example of table 1, with one erroneous vertex and trying with one repetition gives the following results for the Bayesian probability,

$$P_r(\underline{X}_1 \in A' / \underline{X}_1) = ((2/6)*1*(4^2/6^2)*6)/1.283) = 0.693$$

$$P_r(\underline{X}_1 \in A / \underline{X}_1) = ((7/11)*(3/11)*(5^2/11^2)*11)/1.283) = 0.307$$

Therefore $P(\underline{X}_1 \in A' / \underline{X}_1) > P(\underline{X}_1 \in A / \underline{X}_1)$ and the new Bayesian hyper plane will correctly classify this vertex, using only one repetition, m =1. Then L_A is called 1-separable.

As a remark of the importance of the m-separability concept we found that for n = 4 any L-separable logic Boolean function is m-separable. Moreover it must be noticed that there are 65536 logic Boolean functions in B^4 and 1882 are L-separable [7], among them 922 are B-separable, 832 are 1-separable and 128 are 2-separable, so the concept of m-separability enables the realisation of 960 functions [4].

Unfortunately the present methodology does not cover all the L-separable functions for n ≥5, even though for n=5 is possible to synthesise 75% of all L-separable functions with m=4 as maximum number of repetitions [4]. As the dimension of the Boolean functions increases, it is more difficult to estimate the percentage of synthesised functions, and obviously the maximum number of repetitions also increases.

One important point in the search of m-separable functions of n components, is to have a relation between the maximum number of repetitions m and the dimension of the Boolean function n.

There is a L-separable class of Boolean functions that has only one erroneous vector but it needs the maximum number of repetitions at least for n=4 and n=5. The geometric structure of all this functions is a n-1 hyper cube plus one element in one class (A or A'), or a n-1 hyper cube minus one element in the other one. Considering the members of this class as the worst m-separable functions, which means that they need the maximum number of repetitions for each dimension to be synthesised, it is possible to obtain the following relation between m and n, [4]

$$\left(\frac{m+1}{2^{n-1}+1+m}\right)\left(\frac{2^{n-2}+1+m}{2^{n-1}+1+m}\right)^{n-2} = \left(\frac{2^{n-2}-1}{2^{n-1}-1}\right)^{n-2} \quad (2.3)$$

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 179-184

3. Decomposition in B-separable functions

We shall exploit the above results for the realisation of logic Boolean functions, notwithstanding if they are or they are not linearly separable [5].

DEFINITION 5

Let L_A be chosen as reference logic Boolean function, and let us define a subset $A^* \subset B^n$, and its logic Boolean function L_{A*}, then let us define the logic Boolean function L_E by, $L_E(\underline{x}) = 0$ if $L_A(\underline{x}) = L_{A*}(\underline{x})$ and $L_E(\underline{x}) = 1$ if $L_A(\underline{x}) \neq L_{A*}(\underline{x})$.
Obviously $L_A(\underline{x}) = L_{A*}(\underline{x})$ XOR $L_E(\underline{x})$, where, as usual, XOR means the Boolean exclusive disjunction.
The formula stated for B-separable applied to a non linearly separable logic Boolean function L_A yields a Perceptron $\underline{W}_{A*}$, and its logic Boolean function is $L_{A*}(\underline{x}) = \mathrm{sign}[\underline{W}_{A*}{}^T \underline{X}]$, that differs from L_A.
The subset of erroneous vertexes is $E = A \cup A^* - A \cap A^*$; if the logic Boolean function L_E is B-separable, then the function that recognises the errors is realised by another Perceptron $L_E(\underline{x}) = \mathrm{sign}[\underline{W}_E{}^T \underline{X}]$ and then,

$$L_A(\underline{x}) = (\mathrm{sign}[\underline{W}_{A*}{}^T \underline{X}]) \mathrm{XOR} (\mathrm{sign}[\underline{W}_E{}^T \underline{X}])$$

If the logic Boolean function L_E is not m-separable, it will be labeled L_{E1}, then the same construction may be applied. If after k iterations the error function L_{Ek} is B-separable, the realisation of L_A takes the form:

$$L_A(\underline{x}) = (\mathrm{sign}[\underline{W}_{A*}{}^T \underline{X}]) \mathrm{XOR} (\mathrm{sign}[\underline{W}_{E1}{}^T \underline{X}]) \mathrm{XOR} \ldots (\mathrm{sign}[\underline{W}_{Ek}{}^T \underline{X}])$$

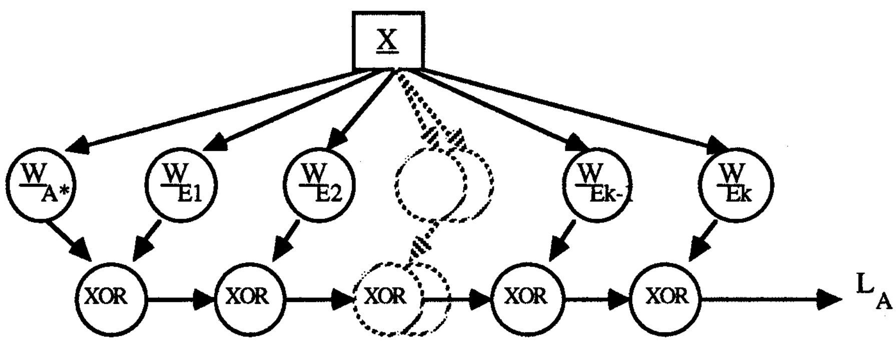

Figure 1

As we can see in figure 1, the realisation of any logic function either linearly separable or not consists in a concatenation of functions, synthesised by a Perceptron and each one correcting the errors of the preceding function.

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 179-184

4 - Conclusion and open questions

The automatic search for a structure in neural networks is a real problem [6]. In the first part of the paper we have found a direct, simple and fast way to synthesise linearly separable functions. The results show that with B-separability and his extension m-separability, we cover a broad range of linearly separable functions of any dimension that can be implemented uniquely by making statistical means calculations.

The last result, as shown in figure 1, is very appealing and implies that any Boolean function, either linearly separable or not, can be realised by k+1 Perceptrons in parallel and a sequence of XOR functions, with the possibility of giving rise to an architecture for circuit integration. Nevertheless for being able to exploit that result it is necessary that no closed loops appear in that decomposition. i.e. the next obtained error function, e_i, has the same output vector that a preceding one e_j, where $i>j$.

Further research examines if absence of loops is a sufficient condition for convergence of the method, as well as to avoid cycles with partial repetitions of the error set. Using this last methodology we are able to synthesise all linearly and non linearly separable functions for n=5.

Another question that has been examined elsewhere [4] is to find a different extension to the B-separability concept that covers all L-separable functions, and a partial solution has been proposed by the use of fractional repetition, replacing m by a rational number.

References

1. F. Rosenblatt,Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain. Psychol. Review, 65, 386-402 (1958)
2. N.Piera, J.Aguilar, Controlling selectivity in non-standard pattern recognition algorithms. IEEE Transactions on Systems, Man and Cybernetics, 21, 71-82, (1991)
3. R.O. Duda, P.E. Hart. Pattern Classification and Scene Analysis. Wiley-Interscience (1973)
4. Andreu Català, Aproximació Connexionista de la separació òptima de classes en sentit Bayesià, Doctoral Thesis.Universitat Politècnica de Catalunya, Departament ESAII, Barcelona (1993)
5. M.L. Dertouzos, Threshold Logic: A synthesis Approach. Research Monograph 32. The MIT Press, Cambridge, Mass. (1965)
6. H.E. Ayestaran, R.W. Prager, The Logical Gates Growing Network. Cambridge University Engng. Dpt. Tech. Re., (1993)
7. S. Muroga, I. Toda, Lower Bound of the Number of Treshold Functions. I.E.E.E. Transactions on Electronic Computers, 805-806, October (1966)