LABOR MARKET SIGNALING ANALYSIS USING THE PROBABILITY OF MISCLASSIFICATION AND NEURAL NETWORKS

Carlos Narciso Bouza HERRERA[1]
Josefina Martínez BARBEITO[2]
Pasha G. MITRA[3]

- *ABSTRACT: In this paper the de-codification of labour market signals is studied. The structure of the corresponding signalling game is described. The decisions of the employer are considered as based on a classification of the aspirant. The evaluation of the posterior probability of classifying him correctly provides a decision rule. A naïve estimator based on density function estimator is proposed. Neural Networks models are proposed for coping with the classification using two approaches. Monte Carlo experiments are used for evaluating the behaviour of the proposals.*
- *KEYWORDS: Signalling games; posterior probabilities; feedforward neural networks; classification tree.*

1 Introduction

Signalling games where introduced by Spence (1973) and dealt with the study of labour market. The game consisted of two players: the aspirant to the job and the employer. The aspirant sends a signal using his curriculum. It determines an input in a system and the employer analyses its output. A decision is made: to give him the contract or not. In section 2 this classic game is presented.

We propose to consider some procedures for classifying the signals. In section 3 we consider the classification problem whose behaviour is evaluated by the probability of deciding whether to hire the aspirant is a good decision or not. The probability mass function of accepting that the aspirant will be successful if a prior probability and a posterior probability are estimated. If it is larger than a preassigned threshold probability, the aspirant is accepted. The estimation uses the observed behaviour of the hired aspirants. Delta sequence estimators of the involved density function and the counts are used. In section 4, the signal sent by the curriculum goes through a unidirectional channel to the employer and a Neural Network uses the inputs provided by a training data set for learning. It aims at minimizing the sum of squares of the classification error. Section 5

[1] Facultad de Matemática y Computación, Universidad de La Habana, San Lázaro y L. CP 10400, Habana, Cuba, E-mail: *bouza@matcom.uh.cu*
[2] Facultad de Ciencias Económicas y Empresariales, Universidade A Coruña, CP 15001, A Coruña, Galicia, Espanha.
[3] Department of Computer Sciences, College of Management Sciences and Business Administration, Universidade A Coruña, CP 15001, A Coruña, Galicia, Espanha.

considers that a Classification Tree is defined for evaluating the aspirants. It is combined with a Neural Network, which teaches how to classify the aspirants.

We used a database of 556 Msc. graduates in Business Administration for evaluating the proposed methods that were studied during the period 2000-2003 in West Bengal. The signals were considered to be the final qualification obtained, the grades in the clusters of Hard and Soft sciences courses and the percent of the total credits obtained in each cluster. The results were completed with the status of the graduates at the end of the course in 2003-2004. The signals took values in $[0,100]^5$. Their success of them was measured by the realization of:

a) Signing a contract with an enterprise within 6 months after graduation.

b) Obtaining a Ph.D. fellowship from a university or a research institute.

Signal $\xi = (x_1, \ldots, x_5)$ was represented by the variables

- X_1 = Final Grade
- X_2 = Final Mean Grade in the set of Hard Sciences courses
- X_3 = Final Mean Grade in the set of Soft Sciences courses
- X_4 = Percent of credits due to Hard Sciences courses
- X_5 = Percent of credits due to Soft Sciences courses

They took value in [0,100].

For predicting if a graduate (aspirant) would be successful or not, signal ξ was used. We propose a procedure for predicting the probability of success in section 3. The success probability was modelled using the model provided by non-parametric density function estimation. The behaviour of two well-known estimators is studied. The data provided by the expedients of the 536 students was analysed.

This database was also used for evaluating the behaviour of the assignment of the graduates when a Neural Network with Classification Tree was used. Success was considered having achieved if the performance, after 3 years in the job, was evaluated as satisfactory or if the advisor of the PhD student considered that he had fulfilled the planned activities. Section 3 is devoted to this study.

The main objective of the Monte Carlo experiments was to evaluate the misclassification probabilities generated by the procedures. Three approaches were considered for generating sub samples from the training set: Simple random sampling, Jacknife and Bootstrap. The samples were selected accordingly. Generally Bootstrap sampling provided the more reliable results. The analysis of the computed misclassification probabilities and accuracy yields a preference for the use of Neural Network methods,

As a result, specific models are proposed for evaluating the signals sent by the curriculum allowing ranking the chances of success of graduate students

2 Signalling as a game

Signalling games where introduced by Spence (1973) and they can be described as follows.

Extensive form of a signaling game

First Movement: The nature chooses a type for player 1.

Second Movement: Player 1 is informed of his type, but player 2 is not informed.

Third Movement: Player 1 sends a signal, which is observed by player 2.

Fourth Movement: Player 2 chooses an action responding to the action of player 2 (signal).

End: The payoffs are produced depending on player 1's type and on the actions of the two players.

The original work of Spence (1974) was in the labour market. He considered the following example:

Example 1. There are two types of aspirants: skilled (S) and unskilled (W) . The difference between them is that the skilled produces a marginal product M_S and the unskilled a marginal product M_U and $M_S > M_U$. Taking the fraction of skilled workers as P , 1 -P of the aspirants are considered as unskilled.

Two situations may arise:

Situation 1. The worker's skill is observable after an evaluation period. The employer (firm) would offer them a salary: WS = MS to the skilled workers and WU = MU to the unskilled workers.

Situation 2. The firm cannot establish the aspirants' skills. Then, the offer will be the expected, or weighted marginal product: $w = (1 - P)M_U + PM_S$.

The aspirants to a job should send signals for establishing their type . The curriculum is a signal that gives information on the skill of the aspirant through his level of education. Let e_H denote the level of education of the aspirant, H = S, U, where S denotes skilled and U unskilled . Education is obtained at a unitary cost c_H. It is natural to assume that $0 < c_U < c_S$. Let us consider that the total cost of education for unskilled and skilled workers respectively are given by the product $c_H e_H$, for H = S, U. Hence a separating equilibrium is an equilibrium that involves each type of worker making a choice that separates the types.

A student may consider the benefit of acquiring an education that sustains a certain curriculum with a level e* . He would consider that the increase in his wages will be $W_\alpha - W_\zeta$. It is as a function of the increase in the marginal product $M_\alpha - M_\zeta$, evaluated by the firm, provided that his signal establishes his belonging to the α-type instead of a ζ-type. As,

$$W_\alpha - W_\zeta < c_\zeta e^*$$

if you are of the ζ-type it does not pay to look for level e*, but it pays if you are of the α-type because

$$W_\alpha - W_\zeta > c_\alpha e^*.$$

This result follows from the fact that for the set of separating equilibrium, we have that as $e_\zeta = 0$ and $e_\alpha = e^*$:

$$[M_\alpha - M_\zeta\ ]/c_\zeta < e^* < [M_\alpha - M_\zeta\ ]/c_\alpha$$

In practice the aspirant to a job sends a signal that should correspond to his type but the employer evaluates it and fixes the wages considering the probability that the signalling means that giving him the job will be a good decision. Then the employer uses the information provided by the curriculum through a vector

$\xi = [x_1, \ldots, x_p]^T$ and links it with the quality

A = set of aspirants that will be successful in the job to be assigned.

The employer wants to classify the aspirant using the information obtained from signal ξ (vector of features). He tries to identify the signal's source. A variable y is generated by the real state, and he wants to determine a function $\nu(\xi)$ such that an adequate prediction of y can be made. The random nature of the nature (source) establishes the randomness of y. Let us assume that the joint distribution of signals and state of the nature are described by the joint density function $f(\xi,y)$. The employer (firm) aims to determine a function ν such that for a norm $\|\bullet\|$ expectation $E[\|y-\nu(\xi)\|]$ is minimum. A common choice is to select the L2 norm, which generates the optimization problem

$$\nu_0(\xi) = ArgMin\{ E[\|y-\nu(\xi)\|^2\ ]\ \} = \int_{\Re\times\Re^p} [y-\nu(\xi)]^2 f(\xi,y)dyd\xi$$

and $\nu_0(\xi) = E[y|\xi]$. The optimality of this method is given by Gauss-Markov Theorem see, Johnson and Wichern (1998) for example, which relies on a large set of assumptions. Among them are the normality and independence of the residuals $e\text{-}\nu_0(\xi)$. As normality is not acceptable in many cases L2 solutions may not be reliable. Financial data for example do not have a Gaussian distribution, see Müller et.al. (1998) for a discussion of this fact.

We will assume that the firm considers that the signalling provides information for deciding whether the aspirant, as an output of a system, supports the use of the decision rule

a) The aspirant should be classified as successful if y = 1 (The aspirant belongs to A)

b) The aspirant should be classified as unsuccessful if y = 0 (The aspirant does not belong to A)

The received signal ξ permits to evaluate variable y and to rewrite the decision rule as

Y = 1 if ξ supports A

ξy = 0 if ξ supports A^c (A^c is the complement of A)

3 A Bayesian model

Statistical classification is concerned about the assignment of an item to a certain class using the information provided by a set of variables. We will use this framework for dividing a set of aspirant in

A = {successful aspirants}

A^c = {successful aspirants}^{c}

As we consider the classification problem the conditional density of ξ, $f(\xi|A)$ provides the probability information for deciding if y = 1 is a good decision or not. Let $\pi(A) = \pi$ be the probability mass function of accepting that the aspirant will be successful. From a Bayesian point of view, it is a prior probability. The posterior probability, once ξ is observed is:

$$\pi(A|\xi) = f(\xi,A)/f(\xi)$$

Then the joint density is:

$$f(\xi,A) = f(\xi|A)\ \pi(A) = f(\xi|y = 1)\pi = f(\xi|y)\pi$$

and the marginal density is equal to

$$f(\xi|A)\pi(A) + f(\xi|A^c)\ \pi(A^c) = f(\xi|y)\pi + f(\xi|y^c)(1-\pi)$$

As A^c is the complement of A and y^c 1-y. Hence it is clear that

$$E[y|\xi] = \pi(y=1|\xi) = \pi(y|\xi)$$

$$V(y|\xi) = \pi(y|\xi)[1-\pi(y|\xi)]$$

The firm fixes a threshold π_o and if $\pi(y|\xi)>\pi_0$, it accepts the aspirant. As considered previously, ξ is a signal that provides the relevant information on y. The firm selects a data set $D = \{(\xi_i, y_i), i = 1, \ldots, N\}$ which is divided into D_T and D_V, $D_T \cap D_V = \varnothing$. Take the number of entries in D_T as $\#D_T = n$. The data provided by data D_T, are analysed using classification techniques. For a detailed discussion on these multivariate statistical models see Johnson and Wichern (1998). The use of Discriminant Analysis is particularly adequate for establishing the role of the signals. It permits to determine a separating plane $\lambda = \xi_d{}^T W$. A signal with $\xi^T W>\lambda$ classifies the aspirant in the population associated with the possession of A. Therefore the normal equation of λ permits to determine the set of signals $D(T1) = \{\xi_i \in D_T | y_i = 1\}$. An approximate value of prior probability π may be fixed by the firm using its information or estimated using de data

$$\pi^* = \#D(T1)/n=m/n$$

The conditional density function $f(\xi|y = 1)$ can be estimated by using an estimator of it. The theory of nonparametric density function estimation provides a large set of estimators which can be used. A broad class of them is determined by the structure defined by:

$$f^*_m(\xi) = \sum_{j\in D(T1)} K[(\xi-\xi_j)/h(m)]/mh^p(m) \qquad (3.1)$$

where

- $K : \Re \rightarrow \Re$ is a kernel function
- .h(m) is the bandwidth, window or smoothing parameter

An estimator belonging to the class characterized by (3.1) is called a kernel-type estimator of the density function. See Devroye and Gyorfy (1985) for a detailed study of

this class of estimators. After consequently redefining sample $D(T0) = \{\xi_i \in D_T | \ y_i = 0\}$, $|D(T0)| = m' = n-m$ and the bandwidth we obtain as an estimator of $f(\xi|y = 0)$:

$$f^*_{m'}(\xi) = \sum_{j \in D(T0)} K[(\xi - \xi_j)/h(m')]/m'h^p(m') \quad (3.2)$$

Therefore a naive estimator of $\pi(y = 1|\xi)$ is

$$\pi^*(A|\xi) = f^*(\xi,A)/f^*(\xi) \quad (3.3)$$

where:

$$f^*(\xi,A) = f^*(\xi|A)\pi^* = f^*(\xi|y=1)\pi^* = f^*(\xi|y)\pi$$

$$f^*(\xi|A)\pi^*(A) + f^*(\xi|A^c)\ \pi^*(A^c) = f^*(\xi|y)\pi^* + f^*(\xi|y^c)(1-\pi^*).$$

Then, the evaluation of signal ξ_s sent by an aspirant s should be evaluated as follows:

Evaluation of a signal using the posterior estimated probability of success

Step 1. Take the aspirant's signal sent by s $\xi(s)$ and evaluate (3.3)

Step 2. If $\pi^*(y = 1|\xi(s)) > \pi 0$ decide that $s \in A$ else $s \in A^c$.

The kernel definition fixes the characteristics of the density function estimator. Taking V_p as the volume of the p-dimensional unit, we will consider the estimators studied by Lima (1998):

Multivariate Epanechnikov

$$K_E(\xi) = V_p(p+32)(1-\xi^T\xi) \text{ if } \xi^T\xi<1$$

$$K_E(\xi) = 0, \qquad \text{otherwise}$$

Multivariate Gaussian

$$K_\Phi(\xi) = [2\pi]^{-p/2} \exp(-\xi^T\xi/2) \text{ if } \xi \in \Re^p$$

The density function estimators, based on these kernels have been studied from different point of views. See textbooks such as Silverman (1986) and Devroye and Gyorfy (1985). Lima (1998) and Bouza and Sing (2004) derived that they enjoyed consistency properties, ensuring their pointwise convergence to the true value of the density function, under some particular conditions, generalising the results presented by Devroye and Gyorfy (1985).

We analysed the behaviour of the estimations of the posterior probability using the proposed estimators of the density function. The database described in the introduction was used. It was randomly divided into two sets of the same size, $\#D_T = \#D_V$. SAS procedure for discrimination was used for determining λ. See Little et al. (1996) and

Wolfinger (1999) for a detailed discussion on the characteristics of the procedures provided by SAS for dealing with logistic regression fitting.

The validation of our proposal was made, by considering different values of π_0. The characteristics of the estimators suggested that to 200 replicas is a sufficiently large number of replicas, see Bouza (2001). Then, our Monte Carlo experiment was conducted for computing estimates in each of the M = 200 random replica of D_T and D_V. In each replica (3.3) was calculated and compared with the value of π_D, which is known and computed from the whole data base D. The evaluation was made by computing:

$$\Delta_H = \sum_{t=1}^{200} |\pi^*_t - \pi_D|/200, \; H = Epanechikov, \; Gaussian$$

for each π_0.

The results of the experiment appear in the Table 1. They suggest that the use of Gaussian Kernel has not a differentiated behaviour as a function of π_0 . For Epanechikov Kernel, Δ_H is an increasing function of it.

Table 1 - Values of Δ_H in 200 runs of the Monte Carlo experiment

	π_0.				
	0.50	0.75	0.90	0.95	0.99
Epanechikov	0.0633	0.0650	0.0679	0.0617	0.0666
Gaussian	0.0506	0.0648	0.0724	0.0582	0.0690

D_V was used for validating the proposed procedure by calculating the frequency of misclassification of the inputs. We calculated three measures for evaluating efficiency:

P_{min} = Minimum of the misclassification relative frequency (probability)

P_{Max} = Maximum of the misclassification relative frequency (probability)

P_{200} = Mean of the 200 misclassification relative frequencies.

Table 2 presents the results obtained in the experiments. The behaviour of the Gaussian Kernel seems to be the best because the means of the estimated misclassification probabilities are smaller than those obtained by using Epanechikov's Kernel. Nevertheless, the results associated with the estimates obtained by using Epanechikov Kernel are more accurate because the range of misclassification probabilities, calculated with the validation data set, is considerably smaller than their Gaussian counterparts. Note that Epanechikov's P_{200} is closer to the center of the intervals defined by [P_{min}, P_{Max}] in all the cases. Its behaviour is more stable as a function of π_0 than the results obtained by using the Gaussian Kernel.

Table 2 - Misclassification probabilities computed in the validation set in the 200 runs of the Monte Carlo experiment

	π_0				
	0.50	0.75	0.90	0.95	0.99
Epanechikov					
P_{min}	0.338	0.321	0.328	0.305	0.333
P_{Max}	0.345	0.353	0.369	0.386	0.357
P_{200}	0.340	0.342	0.348	0.350	0.341
Gaussian					
P_{min}	0.016	0.014	0.017	0.010	0.015
P_{Max}	0.380	0.428	0.279	0.395	0.351
P_{200}	0.240	0.243	0.214	0.197	0.174

4 A neural network model

Neural Networks (NN) are efficient for solving identification problems. Their widespread is due to their capability of approximating functions with a large degree of accuracy. See Hornik, Stinchcombe and White. (1991) and Joya et.al (2004) who provided evidence on the capabilities of NN for approximating functions through learning from the data. We modeled the problem by considering a 3-layer NN architecture based on a classification tree. The first layer represents the inputs of information; the second represents the decisions and are associated with each terminal node. The third layer is a disjunctive combination of the decision areas and the decision is made on the most activated neuron. This approach allows the use of a classification tree for simulating the learning. See Breiman et. al . (1984) for a large discussion on Answer Trees and their implementing algorithms. The algorithm to be used in this section is quite simple. It constructs a binary tree recursively by selecting at each node of the tree as the best path. See Allende et. al. (1996) and Tam and Kiang (1992) and Hung et.al. (2002) for applications of learning through NN in regression fitting, finance and cancer risk evaluation.

Using the NN framework we consider that the curriculum sends a signal through an unidirectional channel to the employer. It is modified by a mechanism that we can relate to the synapses in NN. Then the signals are combined and an output is generated. An activation function determines the new state of the neurons taking into account the entry and the previous activation state. The knowledge in the NN is concentrated on the weights (synapses) of the connections among the neurons. When weights change learning is developed. When they do not change (stability), the NN has learned on the problem. Defining a_i as the activation value due to the signal sent by *i*, the error is measured by:

$$SSE = \sum_i [a_i - y_i]^2 \qquad (4.1)$$

Note that the NN determines activation value a_i as the output for the *i*-th observation and looks for the set of weights minimizing SSE.

We consider a feedforward NN where the number of input, hidden and output layers is equal to one. The arcs connect nodes of a lower layer to a higher one and no circuits are present (feedback loops). The NN topology admits direct connections between some input nodes to output nodes (shortcuts arcs). Hence we deal with a non-perceptron feedback NN. It works by gathering the signal coming through the arcs, a node bias is added and the total input is transferred into an output. The function, which makes transfers, is called activation function.

A commonly used transfer function is the logistic

$$A(\xi) = [1-e^{-\xi}]^{-1}$$

In our case, the NN training poses a nonlinear problem and attaining the global minimum is not guaranteed. Its non-convexity suggests using a multiple starting solution for initializing the NN parameters. The optimal solution of the procedure is the final result with the smallest rank, see Hung et.al. (2002).

Then we proceed as follows for obtaining an appropriate solution:

NN-Procedure

Step 1. Set Q starting solutions for initializing the NN parameters.

Step 2. Compute SSE[q], q = 1, ... ,Q

Step 3. Rank the final solutions and select the output with the smaller rank.

We may consider, as starting solutions, the parameters of a logistic regression adjusted for a sample of the data. The software provided by SAS, Genmod-SAS was used for estimating the regression parameters. Its consistency is expected also when training samples are used, see Hadjicostas (2003). The final result a_i for aspirant i provides an estimation of the posterior probability of the signal i being generated by $y_i = 1$.

We proceeded as in Section 3 and for each sample selected from the training set D_T a NN was trained. A sample fraction $f \in (0,1)$ was fixed. We considered three approaches used:

NN-training approaches

1. The starting points are the regression parameters of the logistic regression obtained from K = 100 samples of size n[j] = nf.
2. The starting points are the regression parameters of the logistic regression obtained from Jacknife samples of size n[j] = n[1-f].
3. The starting points are the regression parameters of the logistic regression obtained from Q = nf Bootstrap samples.

The first approach establishes that simple random sampling is used and Q = 100 independent samples are selected. The data provided by each sample is used for estimating the parameters of a logistic regression

A set of Jackknife samples with common size n[1-f] should be determined, in the second approach, by deleting fn different observations from D_T . $Q = C^n_{n[1-f]}$ regressions are adjusted for obtaining starting points for the learning process.

In the third approach, a simple random sample with replacement is used for selecting Q independent samples of size n. As in the other approaches, logistic regression is used in each Bootstrap sample for fitting the input parameters.

We can evaluate the capacity of NN by computing the standard deviation of the outputs. Defining a_{ij} as the activation value for the consequence of signal i in the sample j, j = 1, ... ,K

$$S[i] = \{\sum_{j=1}^{K} [a_{ij} - \sum_{j=1}^{K} a_{ij}/K]^2 /K-1\}^{-1/2} \qquad (4.2)$$

is the standard deviation of the predictions. We expect that they should be small if NN learns adequately from i.

For training the network, we used Brain Maker v 3.7 for Windows, which provides an efficient implementation of the Classification (Answer) Tree algorithm.

Table 3 presents the statistics of the individual standard deviations computed in the Monte Carlo experiments. Bootstrap provides a more stable set of standard deviations. Its mean, median and mode are very similar in all the cases. Jacknife produces a distribution which is less centered around the mean or the median than the other results. It seems better to generate Bootstrap subsets of data for obtaining the input measures to be used in the learning of the NN.

Table 3 - Statistics of standard deviations [S[i]'s] for the individuals in the Monte Carlo experiments

f	Approach	Statistics				
		Max.	Min.	Mean	Median	Mode
0.1	Simple sampling	53.7	6.2	20.1	23.4	18.6
	Jacknife	71.8	20.0	13.1	18.9	18.2
	Bootstrap	51.5	9.8	27.6	26.8	21.1
0.2	Simple sampling	62.8	6.1	25.1	23.9	20.4
	Jacknife	75.1	18.4	14.9	25.4	23.9
	Bootstrap	51.5	9.4	26.5	26.4	22.0
0.5	Simple sampling	62.4	5.9	20.7	22.8	19.5
	Jacknife	76.0	17.8	18.4	20.4	19.8
	Bootstrap	52.4	9.1	26.7	25.9	25.7

Again the validation set was used for computing in each unit in it if the rule "Accept that $\pi(y|\xi) > \pi_0$" was a correct decision. It relies on the classification of the unit sending

signal ξ. Bishop (1995) points out the abilities of NN for classification. Zhang (2000) brings a large analysis of the literature published on this theme.

Take :

$I[j] = 1\ [0]$ if the decision is correct [otherwise]

We measure it in each $j \in D_V$ and defining

$$P[1] = \sum_{j \in DV} I[j] / \Lambda D_V |$$

$|D_V|$-m = fN and P[1], the estimate of the misclassification probability, was computed in M = 200 random partitions of D for each approach. The results appear in Table 4 The results generated by the use of Bootstrap approach are better than those obtained by simple random sampling and Jacknife inputs. Jacknife appears to be similar to Bootstrap for f = 0.01 and generally provided smaller misclassification probabilities than simple sampling.

Table 4 - Values of the proportion of correct decisions using the adjusted NN in the Monte Carlo experiments.

Approach	m	f		
		0.1	0.2	0.5
Simple sampling	55	0.35	0.48	0.41
Jacknife		0.26	0.34	0.28
Bootstrap		0.28	0.22	0.19
Simple sampling	110	0.21	0.29	0.25
Jacknife		0.20	0.22	0.34
Bootstrap		0.19	0.15	0.18
Simple sampling	278	0.30	0.31	0.31
Jacknife		0.22	0.35	0.32
Bootstrap		0.21	0.22	0.17

5 Prior classes and a neural network solution

In this section we consider that the aspirant sends a signal and the employer analyses it for deciding the job for which he is capable. The employer has previously defined a set of C classes identifying the possible jobs. One of them identifies not to employ the aspirant. This is a classification problem. Classification Trees is a computer intensive method with widespread popularity. It can be used effectively for solving this classification problem. This approach is denominated in the literature as Classification and Regression Trees (CART), see Breiman et. al. (1984) and Johnson and Wichern (1996). It uses clustering procedures in different phases providing a binary tree. It hierarchically describes the space of the obtainable information provided by the signals. The firm determines a set of questions and they are posed sequentially. A stop-splitting rule allows determining the path to be followed. Each terminal node determines a class (job). The employer can consider not only the signals provided by the curriculum, but

some subjective ones; however we will consider that all the answers are quantitative variables. The decision rules consider the information provided by signal $\xi = (x_1, ...,x_i ,..,x_p)$. Without losing in generality, we will consider that x_i is evaluated before x_j if i<j. By analysing x_1, the first split is made and the first node is divided into two nodes; one defines a left branch (yes) and the other the right one (no). Then another variable is analysed and the splitting ends when a terminal node is attained. In our case, an aspirant is classified in the class corresponding to the terminal node where his signals conduct him. Then the final result is:

$$Y(\xi) = t, \text{ if the terminal node is } t.$$

As in section 3, we are interested in the prior probabilities. Now we have C classes and we denote the prior by $\pi(A_t)$. If we classify an aspirant in A_t when he really belongs to A_{t^*} the firm incurs in a cost $C[A_t|A_{t^*}]$, the misclassification cost . The employer determines a decision rule ρ and considers the cost associated to it by considering the behavior of the risk:

$$R(\rho) = \sum_{t=1}^{C} \pi(A_t) \sum_{t\neq t^*} C[A_t|A_{t^*}) \, Prob\{\rho(\xi) = t|Y(\xi) = t^*]) = \sum_{t=1}^{C} \pi(A_t) \sum_{t\neq t^*} C[A_t|A_{t^*}) \, f(\rho(\xi) = t|t^*)$$

Following the Bayesian approach, see Celeux and Lechevalier (1982), we should look for the Bayes rule

$$\rho_B = Arg\ Min\{R(\rho)\}$$

Using the same reasoning as in section 3 the joint density is:

$$f(\rho(\xi) = t, A_{t^*}) = f(\rho(\xi) = t|A_{t^*}) \, \pi(A_{t^*}) = f(\rho(\xi) = t|y = t^*)\pi_{t^*} = f(\rho(\xi) = t|t^*)\pi_{t^*}$$

The conditional density function may be estimated by using an adequate estimator .

We consider that the Classification Tree (CT) is determined using the decision rules:

- $\rho 1$ = if X1 < 80 and X2 < 50 then the aspirant is refused.
- $\rho 2$ = if X1 < 80 and X2 $\geq$ 50 then Job1 is proposed to the aspirant.
- $\rho 3$ = if X1 $\geq$ 80 and X4 < 50 and X3 $\geq$ 80 then Job1 is proposed to the aspirant.
- $\rho 4$ = if X1 $\geq$ 80 and X4 $\geq$ 50 and X3 $\geq$ 80 then Job2 is proposed to the aspirant
- $\rho 5$ = if X1 $\geq$ 90 and X4 < 50 and X2 $\geq$ 90 then Job3 is proposed to the aspirant

The objective of the firm is to learn from a set of aspirants trying to minimize the counterpart of (4.1)

$$SSE = \sum_{i=1}^{n} \sum_{t=1}^{C} [a_i(t)-y_i(t)]^2 = \sum_{i=1}^{n} \sum_{t=1}^{C} e_i(t)^2 \qquad (5.1)$$

Where n is the size of the learning data set D_T , C the number of classes (jobs) and $a_i(t)$ the calculated output .

Following the proposals of Sethi (1990) a NN's architecture can be conceived using information from the decision tree. It uses $e_i(t)$ for assigning an input to a prior class. The signal is considered strong if $e_i(t)>0$ and weak otherwise. The weight assigned to the neuron is in correspondence with the strangeness of the signal. The neurons of the first

hidden layer are non-terminal nodes of the decision tree. Bennet (1992) provides an insight of the relation between NN and CT and their algorithmisation. The output of this layer provides a codification of the variables space and activates the neurons of the second layer, thus identifying the terminal nodes of the decision tree. Note that each observation (signal of an aspirant) goes through the decision tree in the usual way and NN acts by learning how to classify it. The number of connections made for arriving to a terminal node is large as a function of the distance between the terminal node and the root. The weight is an increasing function of the number of connections. The third layer represents the prior classes. It receives inputs from the second hidden layer (terminal nodes) and classifies the observations.

The algorithm of the back-propagation method of multilayer perception permits to train this NN as the neurons of the layer that are not connected to each other but with the following layer. It tries to minimize (5.1) by minimizing the last layer using a gradient-type adaptive algorithm.

We use, in our experiments, the NN-procedure described in section 4 as well as the proposed approaches for partitioning the training sample.

The statistics of the accuracy, measured by (4.2), of the Monte Carlo results using CT (Classification Tree) and NN with CT are given in Table 5.

Table 5 - Statistics on the individual standard deviations in the Monte Carlo experiments of CT and a NN with CT

f			Statistics				
			Max.	Min.	Mean	Median	Mode
0.1	Classification Tree	Simple sampling	40.4	15.1	20.8	17.6	18.6
		Jacknife	43.4	17.1	12.2	20.7	14.1
		Bootstrap	32.7	16.8	12.3	21.7	15.7
	NN with CT	Simple sampling	36.3	13.5	19.5	19.2	17.6
		Jacknife	36.5	11.6	12.3	181.4	14.7
		Bootstrap	29.4	7.3	10.1	11.1	11.3
0.2	Classification Tree	Simple sampling	40.9	13.1	20.7	23.2	19.3
		Jacknife	43.7	11.1	13.3	17.1	18.5
		Bootstrap	33.4	16.0	16.8	24.2	20.7
	NN with CT	Simple sampling	37.1	7.1	22.0	18.7	13.7
		Jacknife	36.9	8.2	13.2	18.9	22.7
		Bootstrap	36.3	7.1	10.1	14.9	11.9
0.5	Classification Tree	Simple sampling	45.0	11.7	21.1	18.7	19.4
		Jacknife	44.7	10.4	17.7	19.6	19.2
		Bootstrap	41.8	13.9	15.7	24.9	22.8
	NN with CT	Simple sampling	44.4	6.5	20.4	21.9	19.6
		Jacknife	39.6	12.0	16.7	19.4	18.4
		Bootstrap	37.3	7.8	11.8	15.4	13.7

An analysis of the results suggests that NN with CT provides standard deviations that are more variable than those obtained in the previous section. Bootstrap approach provides the most accurate results in the outputs in both methods.

The probability of misclassification was calculated under the previously developed approaches for generating different NN data sets for training. The results, when the CT technique is used directly and when NN with CT classifies the signals, are given in Table 5.

Table 6 - Values of the proportion of correct decisions using the CT and NN with CT in the Monte Carlo experiments.

m			f		
			0.1	0.2	0.5
55	Classification tree	Simple sampling	11.4	12.7	27.7
		Jacknife	21.7	22.4	20.3
		Bootstrap	18.3	17.8	12.9
	NN with CT	Simple sampling	12.8	11.4	19.5
		Jacknife	18.4	18.2	16.4
		Bootstrap	15.3	16.8	15.7
110	Classification tree	Simple sampling	11.4	12.7	27.7
		Jacknife	21.7	22.4	20.3
		Bootstrap	18.3	17.8	12.9
	NN with CT	Simple sampling	12.8	11.4	19.5
		Jacknife	18.4	18.2	16.4
		Bootstrap	15.3	16.8	15.7
278	Classification tree	Simple sampling	18.2	12.2	19.0
		Jacknife	10.5	10.0	18.2
		Bootstrap	17.0	13.9	18.0
	NN with CT	Simple sampling	13.0	10.6	11.0
		Jacknife	6.4	6.6	15.9
		Bootstrap	14.8	15.1	15.0

The use of the NN with CT diminishes the probability of misclassification as noted in Table 6.

6 Conclusions

The procedures proposed in this paper permit to use signals for classifying aspirants to a job. A database permitted to evaluate the behaviour of the proposed evaluation methods for deciding if an aspirant would be successful in job to which he/she aspires.

In section 3, the use of a Bayesian point of view allows the development of a classifying procedure and to evaluate its behaviour by estimating the probability of misclassifying the aspirants.

An NN procedure allows learning how to classify the aspirants. Its behaviour is fairly better than the use of the classification procedure described in section 3.

If the firm establishes a set of decisions, based on the scores of the variables, a CT can be described and used for classifying the aspirants. It can be combined with an NN. The procedure is called NN with CT and works better than the use of CT directly.

The learning obtained by the use of NN provides an increase in the effectiveness of the classification of aspirants.

A ranking of the graduate students in the Data Base was made.

Acknowledgements

This paper was supported by project Tarifación Marítima of Universidad A Coruña and by the assignment of an Associated Professorship to one of the authors provided by the Third World Academy of Sciences. This version has been improved thanks to the suggestions of an anonymous referee.

HERRERA, C. N. B.; BARBEITO, J. M.; MITRA, P. G. Análise de indicadores do mercado de trabalho através da probabilidade de má classificação em redes neurais. *Rev. Mat. Est*., São Paulo, v.24, n.3, p.61-76, 2006.

- RESUMO: Este trabalho tem por objetivo investigar a decodificação dos indicadores do mercado de trabalho. Para esse fim, descreve-se a estrutura do jogo de sinais correspondente. As decisões do empregador são consideradas com base em uma classificação do aspirante. A avaliação da probabilidade posteriori de classificá-lo corretamente fornece uma regra de decisão. Um estimador naïve baseado no estimador da função de densidade é proposto. Modelos de Redes Neurais são propostos para lidar com a classificação, utilizando duas abordagens. Experimentos Monte Carlo são empregados para avaliar o comportamento das propostas.
- PALAVRAS-CHAVE: Jogos de sinais; probabilidade posteriori; redes neurais de alimentação; árvore de classificação.

References

ALLENDE, S. M. et al. A parametric programming algorithm for learning in artificial neural networks through linear regression curve fitting. *Invest. Operac.*, Habana, v.17, p.55-60, 1996.

BENNET, K. *Decision tree construction via linear programming*. In: MIDWEST ARTIFICIAL INTELLIGENCE AND COGNITIVE SC. CONFERENCE, 4., 1992, California. *Prociddings …*, California, 1997. p.97-101, 1992.

BISHOP, C. M. *Neural networks for pattern recognition*. Oxford: Oxford University Press, 1995.

BREIMAN, L. et al. *Classification and regression trees*. Belmont: Wadsworth, 1984.

BOUZA C. N. Investigation of Burn-in-time problems with unknown failure time distribution . *J. Stat. Managem.*, New Delhi, v. 37, p. 1-7, 2001.

BOUZA, C. N.; SING, L. A note on the estimation of quantiles using record breaking. *Invest. Operac.*, Habana, v.25, p.238-242*,* 2004.

CELEUX, G.; LECHEVALLIER, Y. Non parametric decision tree by Bayesian approach. *Compstat*, Voorburg, v. 82, p.161-166, 1982

DEVROYE, L.; GYORFY, L. *Non parametric density estimation. The L1 view*. New York: Wiley, 1985.

HADJICOSTAS, P. Consistency of logistic regression coefficient estimates calculated from a training sample. *Stat. Probab. Lett.*, Amsterdam, v.62, p.293-303, 2003.

HILERA, J. R.; MARTÍNEZ, V. J. *Redes neuronales artificiales. Fundamentos, modelos y aplicaciones.* Madrid: Ra-Ma, 1995.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. *Neural Networks*. New York, v.2, p.359-366, 1991.

HUNG, M. S.; SHANKAR, M.; HU, M. Y. Estimating breast cancer risks using neural networks. *J. Oper. Res. Soc.*, Baltimore, v.53, p.222-231, 2002.

JOHNSON R. A.; WICHERN, D. W. *Applied multivariate statistical analysis*. New Jersey: Prentice Hall, 1998.

JOYA CAPARRÓS, G. et al. *Optimización inteligente*: técnicas de inteligencia computacional para optimización. Málaga: Instanet, 2004.

LIMA GONZÁLEZ , S. *Estimación de la densidad a través de una axiomática,* 1998. 95f. Thesis, Universidad de La Habana, Habana, 1988.

LITTLE, R. C. et al. *SAS system for mixed models*. Cary: SAS Institute, 1996.

MÜLLER, U. A.; DACAROGNA, M. M.; PICTET, O. V. *Heavy tails in high frequency financial data.*. In: ADLER, R. J.; FELDMAN, R. E.; TAQQU, M. S. *A practical guide to heavy tails: statistical techniques and applications.* Basel: Birkhäuser, 1998. p.55-78

SETHI, I. K. *Entropy nets:* from decision trees to neural networks. *Proc. IEEE*, New York, v.78, p.605-613, 1990.

SILVERMAN, B. W. *Density estimation for statistics and data analysis*. New York: Chapman and Hall, 1986.

SPENCE, M. A. Job market signaling, *Q. J. Econ.,* v87, p.355-374, 1973.

TAM, K. Y.; KIANG, M. Y. Managerial applications of neural networks: the case of bank failure predictions. *Manag. Sci.* Providence, v.38, p.926-947, 1992.

WOLFINGER, R. D. *Fitting nonlinear models with the new NL mixed procedure*. In: ANNUAL SAS USERS GROUP INTERNATIONAL CONFERENCE, 14., Cary. *Proceedings…* Cary: SAS Institute, 1999. p.1666-1675.

ZHANG, G.P. Neural network for classification: a survey. IEEE. *Trans. Syst. Manag. Cybernetic.*, v.30, p.451-462, 2000.

Recebido em 18.06.2005.

Aprovado após revisão em 24.06.2006.