LAMB MEAT TENDERNESS PREDICTION USING NEURAL NETWORKS AND SENSITIVITY ANALYSIS*

Paulo Cortez
Departamento de Sistemas de Informação
University of Minho
4800-058 Guimarães, PORTUGAL
Email: pcortez@dsi.uminho.pt

Manuel Portelinha, Sandra Rodrigues, Vasco Cadavez, Alfredo Teixeira
Escola Superior Agrária
Bragança Polytechnic Institute
5301-855 Bragança, PORTUGAL
Emails: {mportelinha, srodrigues, vcadavez, teixeira}@ipb.pt

KEYWORDS: Regression, Multilayer Perceptrons, Multiple Regression, Meat Quality, Ensembles, Data Mining.

ABSTRACT

The assessment of quality is a key factor for the meat industry, where the aim is to fulfill the consumer's needs. In particular, *tenderness* is considered the most important characteristic affecting consumer perception of taste. In this paper, a *Neural Network Ensemble*, with feature selection based on a *Sensitivity Analysis* procedure, is proposed to predict lamb meat tenderness. This difficult real-world problem is defined in terms of two regression tasks, by using instrumental measurements and a sensory panel. In both cases, the proposed solution outperformed other neural approaches and the *Multiple Regression* method.

1. Introduction

A top priority factor in the success of meat industry relies on the ability to deliver specialties that satisfy the consumer's taste requirements. Although there are several factors that influence meat quality (e.g. *juiciness* or *appearance*), *tenderness* is considered the most important attribute (Huffman et al., 1997). The ideal method for measuring tenderness should be accurate, fast, automated and noninvasive. In the past, two major approaches have been proposed (Arvanitoyannis and Houwelingen-Koukaliaroglou, 2003): *instrumental* and *sensory* analysis. The former is based in an objective test, such as the *Instron* instrument, which measures the *Warner-Bratzler Shear (WBS)* force and is the most commonly used device. On the other hand, sensory methods are based in subjective information, usually given by a human taste panel. Both approaches are invasive, expensive and time demanding, since they require laboratory work. For instance, the *WBS* values can only be obtained 72 hours after slaughtering, while the preparation and execution of consumer taste panel may take several days.

An alternative is to use cheap and non invasive carcass measurements that can be collected within the first 24 hours after slaughtering (e.g. pH and color). Under this scheme, the classic animal science approach is based on *Multiple Regression* models (Arvanitoyannis and Houwelingen-Koukaliaroglou, 2003), using meat features as independent (or input) variables and the *WBS* or *sensory* measures as the depended (or output) ones. Yet, these linear models will fail when strong nonlinear relationships are present. In such cases, a better option is to use *Neural Networks (NNs)*, due to their nonlinear mapping and noise tolerance capabilities (Haykin, 1999). Indeed, NNs are gaining an attention within the *Data Mining* field, due to their performance in terms of predictive knowledge. Another promising research area is based in the use of *Ensembles*, where several models are combined in some way in order to produce an answer (Dietterich, 2001). This interest arose due to the discovery that ensembles are often more accurate than single models.

In *Data Mining* applications, besides obtaining a high predictive performance, it is often useful to provide explanatory knowledge. In particular, the measure of input importance is relevant within this domain. Since carcass features are often highly correlated, *Principal Component Analysis* has been proposed to reduce the input dimensionality (Arvanitoyannis and Houwelingen-Koukaliaroglou, 2003). However, the principal components are compressed variables and they do not represent a direct meaning for the meat user. A better approach is to use *Sensitivity Analysis* (Kewley et al., 2000), which has outperformed other input selection techniques (e.g. *Forward Selection* and *Genetic algorithms*).

In the past, several studies have used *NNs* to assess meat quality (e.g. beef, pork, poultry or sausages) (Arvanitoyannis and Houwelingen-Koukaliaroglou, 2003). However, regarding tenderness prediction, the literature seems scarce and it is primarily oriented towards beef (Hill et al., 2000). In this work, a *Neural Network Ensemble*, in conjunction with a feature selection procedure based on a *Sensitivity Analysis*, is proposed to predict lamb meat tenderness. This real-world problem will be modeled in the R simulation environment (R Development Core Team, 2004) in terms of two regression tasks, using instrumental and sensory measurements. The proposed strategy will be tested on animal data, collected from the *Trás-os-Montes* region of Portugal, and compared with other *NNs* approaches and a *Multiple Regression*.

*In Proceedings of the *European Simulation Multiconference - ESM'2005*, Oporto, Portugal, October, 2005, SCS.

2. Materials and Methods

2.1 Lamb Meat Data

This study considered lamb animals with the *Protected Designation of Origin (PDO)* certificate, from the *Trás-os-Montes* northeast region of Portugal. The database was collected from November/2002 until November/2003, with each instance denoting the readings obtained from a slaughtered animal. Since each animal presents considerable costs (around 6 euros per carcass), the dataset is quite small, with a total of 81 examples. Table 1 presents the data attributes. The **HCW** is obtained one hour after slaughter, exfoliation and evisceration. The former two attributes (**Breed** and **Sex**) are also registered at slaughterhouse, while the others are measured in laboratory. Due to their visual nature, the color attributes (**a***, **b***, **dE**, **dL** and **dB***) have a high impact in consumer's perception. In most of the situations, these are the only attributes that the consumer can judge.

Table 1: The Dataset Main Attributes

Attribute	Description	Domain
Breed	Breed type	$\{1,2\}^a$
Sex	Lamb sex	$\{1,2\}^b$
HCW	Hot carcass weight (*kg*)	$[4.1,14.8]$
STF2	Sternal fat thickness	$[6.0,27.8]$
C	Subcutaneous fat depth	$[0.3,5.1]$
pH1	pH 1 hour after slaughtering	$[5.5,6.8]$
pH24	pH 24 hours after slaughtering	$[5.5,5.9]$
a*	Color red index	$[11.5,22.2]$
b*	Color yellow index	$[6.5,12.5]$
dE	Total color difference	$[46.5,60.9]$
dL	Luminosity differential	$[-56,-39]$
dB*	Yellow differential	$[15.3,22.5]$
WBS	Warner-Bratzler Shear force	$[9.5,57.0]$
STP	Sensory Taste Panel	$[0.7,7.1]$

[a] 1 – *Bragançana*, 2 – *Mirandesa*; [b] 1 – *Male*, 2 – *Female*

The **WBS** force is the major index for measuring meat tenderness. It can only be obtained in laboratory, no sooner than 72 hours after slaughter, by using an invasive device called *Instron*. On the other hand, a more elaborated scheme was devised to obtain the sensory values (**STP**). A panel of 12 trained individuals, from the *Bragança Polytechnic Institute*, was selected. Then, meat samples from the *longissinus thoracis* muscle were collected and defrost at 4°C in a refrigerator. Next, each sample was randomly encoded with a 3 digit number, wrapped in an aluminum sheet and heated at 100°C. Then, each panel member was set in an individual compartment, performing a taste proof, under similar conditions, of random selected samples. Between different tastes, mouths were cleaned by using water and by eating small golden apple pieces. Each sample was ranked from 0 (the most tender) to 10 (the most tough). Finally, the **STP** attribute was measured as the average of the grades from the panel. Since the original data contained missing values (2 for the **WBS** and 10 for the **STP**), two new datasets were created by discarding these entries. The first contains 79 rows (for the **WBS** task), while the second has 71 examples (**STP**).

2.2 Learning Models

A regression dataset D is made up of $k \in \{1,...,N\}$ examples, each mapping an input vector $(x_1^k,\ldots,x_I^k)$ to a given target y_k. The error for a given k is: $e_k = y_k - \widehat{y}_k$, where $\widehat{y}_k$ represents the predicted value for k input pattern. The overall regression performance is computed by global metric, namely the *Mean Absolute Error (MAE)*, *Relative Mean Absolute Error (RMAE)*, *Root Mean Squared (RMSE)* and *Relative Root Mean Squared (RRMSE)*, which can be computed as:

$$\begin{aligned} &MAE = 1/N \times \textstyle\sum_{i=1}^{N} |y_i - \widehat{y}_i| \\ &RMAE = 1/N \times MAE / \textstyle\sum_{i=1}^{N} |y_i - \bar{y}_i| \times 100\,(\%) \\ &RMSE = \sqrt{\textstyle\sum_{i=1}^{N} (y_i - \widehat{y}_i)^2 / N} \\ &RRMSE = RMSE / \sqrt{\textstyle\sum_{i=1}^{N} (y_i - \bar{y}_i)^2 / N} \times 100\,(\%) \end{aligned} \quad (1)$$

In all these statistics, lower values result in better predictive models. The *RMAE* and *RRMSE* metrics are scale independent, where a 100% means that the regression method has similar performance as the constant average predictor.

A *Multiple Regression (MR)* model is defined by the equation (Hastie et al., 2001):

$$\widehat{y} = \beta_0 + \sum_{i=1}^{I} \beta_i x_i \quad (2)$$

where $\{x_1,\ldots,x_I\}$ denotes the set of input variables and $\{\beta_0,\ldots,\beta_I\}$ the set of parameters to be adjusted, usually by applying a least squares algorithm. Due to is additive nature, this model is easy to interpret and has been widely used in regression applications.

Neural Networks (NNs) denote a set of connectionist models inspired in the behavior of the central nervous system of living beings. In particular, the *Multilayer Perceptron* is the most popular neural architecture, where *neurons* are grouped in *layers* and only *forward connections* exist (Haykin, 1999). The *Multilayer Perceptrons* used in this study make use of biases, one hidden layer with H hidden nodes and sigmoid activation functions (Fig. 1). When modeling regression tasks, the usual approach is to adopt one output node with a linear function, since outputs may lie out of the logistic output range ($[0,1]$) (Hastie et al., 2001). Thus, each regression task (**WBS** and **STP**) will be modeled by a different *NN* and the overall model is given by the equation:

$$\widehat{y} = w_{o,0} + \sum_{j=I+1}^{o-1} f\Big(\sum_{i=1}^{I} x_i w_{j,i} + w_{j,0}\Big) w_{o,i} \quad (3)$$

where $w_{i,j}$ denotes the weight of the connection from node j to i (if $j = 0$ then it is a *bias* connection), o denotes the output

node, f the logistic function ($\frac{1}{1+e^{-x}}$), and I the number of input nodes.

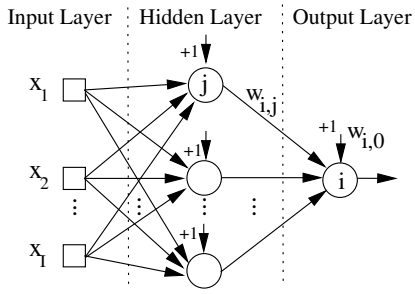

Fig. 1: The *Multilayer Perceptron* Architecture

Supervised learning is achieved by an iterative adjustment of the network connection weights (the *training* algorithm), in order to minimize an error function (typically the sum of squared errors), computed over the *training examples* (or *cases*). Before training, the data needs to be preprocessed. Hence, all attributes were standardized to a zero mean and one standard deviation domain (Hastie et al., 2001)

The performance will be sensitive to the *NN* topology choice: a small network will provide limited learning, while a large one will overfit the data. To solve this hurdle, one solution is to use a large number of hidden nodes (H) and train the *NN* with a *regularization* method (Hastie et al., 2001). In this work, regularization will be performed by a *weight decay* procedure, where a weight penalty term (λ) shrinks the size of the neural weights. Under this scheme, the crucial parameter is the choice of λ.

For a given network, the initial weights will be randomly set within the range $[-0.7,+0.7]$. Next, the training algorithm is applied and stopped when the error slope approaches zero or after a maximum of E epochs. After training, the *Sensitivity Analysis* is performed. It is measured as the variance (V_a) produced in the output ($\widehat{y}$) when the input attribute (a) is moved through its entire range (Kewley et al., 2000):

$$\begin{array}{rcl} V_a & = & \sum_{i=1}^{L}(\widehat{y}_i - \overline{\widehat{y}})/(L-1) \\ R_a & = & V_a / \sum_{j=1}^{I} V_j \times 100\,(\%) \end{array} \quad (4)$$

where I denotes the number of input attributes and R_a the relative importance of the a attribute. The $\widehat{y}_i$ output is obtained by holding all input variables at their average values. The exception is x_a, which varies through its entire range with L levels. In this work, L was set to 2 for the binary attributes and 7 for the continuous inputs.

Since the *NN* cost function is nonconvex (with multiple minima), the quality of the trained network depends on the choice of the starting weights. Thus, R runs will be applied to each neural configuration and the selected *NN* will be the one with the lowest penalized error. This setup will be called *Multiple Neural Network (MNN)*. Another option is to use a *Neural Network Ensemble (NNE)*, consisting of R networks trained with random weights. The final prediction is given as the average of the individual predictions.

2.3 Simulation Environment

All experiments were conducted with an *Intel Centrino 1.60 GHz* processor, under the *Linux* operating system. The simulations were programmed in the **R** environment (R Development Core Team, 2004), an open source and high-level matrix programming language that provides a powerful suite of tools for statistical and graphical analysis.

The **R** functions that were used by the written code include: **lm**, **nnet** and **crossval**. The former function is defined in the **R** base distribution (R Development Core Team, 2004) and fits a *Multiple Regression*. The second procedure is available in the *nnet* package (Venables and Ripley, 2002) and trains a multilayer network with the *BFGS* algorithm, from the family of quasi-Newton methods, allowing also the use of *weight decay*. Finally, the last function implements the K-fold estimation procedure and it can be found in the *bootstrap* package (Efron and Tibshirani, 1993). For demonstrative purposes, a small piece of the main **R** code is shown:

```
library(bootstrap) # load this package
library(nnet)
source("code.R") # load the written R code
# read the WBS dataset from a file
d<-read.table("wbs.csv",header=T,sep=';')
# set the input and output variables
Inputs<-d[,1:12] # matrix with the 12 inputs
Output<-d[,13] # vector with the WBS values
Runs<-5 # number of runs
for(i in 1:Runs)
{
 # display current run and time
 print(paste("Run:",i,date()))
 # fit the MR model (uses lm and crossval)
 MR<-lm.ktest(Inputs,Output)
 # get the MAE, RMAE, RMSE, RRMSE errors
 eMR<-errors(MR,Output)
 # fit the MNN model (uses nnet and crossval)
 MNN<-mlp.ktest(Inputs,Output)
 eMMN<-errors(MNN,Output)
 # fit the NNE model (uses nnet and crossval)
 NNE<-mlpens.ktest(Inputs,Output)
 eNNE<-errors(NNE,Output)
}
```

3. Results

After preliminary experiments, the maximum number of training epochs was set to $E = 10$, the number of hidden nodes was set to $H = 24$ and the number of runs/ensemble networks was set to $R = 5$. The most important parameter (λ) is tuned by applying a coarse grid-search. The first grid level searches all discrete values within the range $\{0.00, 0.01, \ldots, 0.20\}$ and the configuration with the lowest prediction error (λ_1) is selected. Then, the second level proceeds with a fine tune within the range $\lambda_2 \in \{\lambda_1 - 0.005, \ldots, \lambda_1 - 0.001, \lambda_1 + 0.001, \ldots, \lambda_1 + 0.004\} \wedge \lambda_2 \geq 0$. Therefore, the number of searches is equal to $21+9 = 30$ (or $21+5 = 26$ if $\lambda_1 = 0$).

To estimate the *NN* prediction accuracy for the grid-search, a 10-fold cross-validation (Efron and Tibshirani, 1993) will be adopted, where the training set is divided in 10 subsets of equal size. Sequentially, one different subset is tested (with 10% of the data) and the remaining data used for ajusting the *NN* weights. At the end of 10 trainings, the predictor has been tested on all training data and the final estimate is given by the *RMSE* (Equation 1) computed over the 10 test sets. As an example, Fig. 2 plots the error evolution for a given execution of the two level grid-search (**WBS** task). The figure clearly illustrates that the error curve is nonconvex, thus justifying the use of the grid search. In this case, the highest predictive decay ($RMSE = 6.75$) was found for $\lambda = 0.097$. After obtaining the best decay, the final *NNs* are retrained with the all the data from the training set.

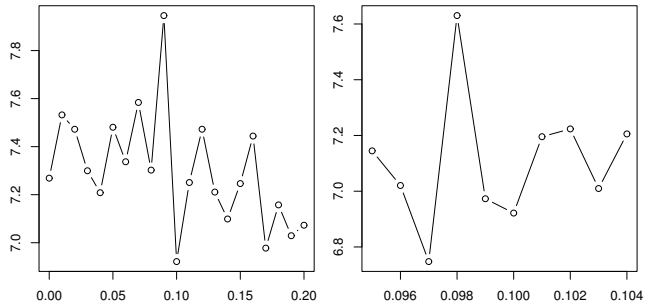

Fig. 2: The Decay (*x*-axis) vs *RMSE* (*y*-axis) Values for the First Level (left) and Second Level (right) Grid Searches

At a higher level, and to compare the different models, 5 runs of a 10-fold cross-validation (computed over all available data) were executed. This means that in each of these 50 experiments, 90% of the data is used for learning and 10% for testing. The results are shown in Table 2, in terms of the average of the test errors obtained over the 50 experiments.

Table 2: The Lamb Meat Tenderness Regression Results

Task	**Model**	*MAE*	*RMAE*	*RMSE*	*RRMSE*
	MR	9.2	134.7%	11.6	130.4%
WBS	*MNN*	6.2	90.1%	8.1	91.2%
	NNE	5.9	86.6%	7.8	87.3%
	NNESA	**5.5**	**81.4**%	**7.5**	**83.7**%
	MR	1.6	119.3%	2.1	131.7%
STP	*MNN*	1.4	99.9%	1.7	104.1%
	NNE	1.3	92.4%	1.6	96.7%
	NNESA	**1.2**	**84.9**%	**1.5**	**89.5**%

The *Multiple Regression (MR)* results are worst than the trivial average forecast. The differences between the *MR* and the *NN* methods suggest that both tasks present nonlinearity. The *Multiple Neural Network (MNN)* works better than the *MR*, although it is outperformed by the ensemble version (*NNE*). Regarding the computational effort, the *MR* results were obtained after 1 second, while the *MNN* and *NNE* configurations required 1 hour each. Since both neural approaches demand a similar computation, the last setup will be favored due to its best performance.

Table 3 shows the average relative importance (Equation 4) of the most important input variables for the *NNE* method. For the feature analysis, it was decided to select the attributes with a relative importance $\geq 3\%$, which allows an input reduction to around half the inputs. Despite the difference in the percentage values, the selected features are quite similar for both problems. It is also interesting to notice that the **Sex** attribute is the least relevant factor, with a relevance of 0.08% (**WBS**) and 1.48% (**STP**). Apparently, this contrasts with the known knowledge that gender affects tenderness. However, female meat often presents a higher weight and fatness, thus the sex information may be indirectly represented in the **HCW** and **STF2** variables.

Since non relevant inputs may affect the performance, another setup, called *Neural Network Ensemble based on Sensitivity Analysis (NNESA)*, was devised by considering the most important inputs of Table 3. Indeed, the *NNESA* method managed to obtain the best results (Table 2), outperforming the *NNE* approach, specially for the **STP** task. In Table 3, the sensitivity values were also presented for this last method. For the **WBS** task, the red color (**a***) seems to be the most important attribute, followed by the weight (**HCW**) and total color difference (**dE**). Regarding the **STP** problem, the most relevant features are the **Breed**, red color index (**a***) and sternal fat thickness (**STF2**). The differences obtained between the two tasks may be explained by psychological factors. For instance, the **Breed** importance increased from 2.8% (**WBS**) to 36.8% (**STP**). As an example, Fig. 3 shows the scatter plots of the predicted values vs the observed ones for the **WBS** task, where the diagonal line denotes the perfect forecast. The *NNESA* approach clearly presents a better performance, with more predictions along the line than the *MR* method.

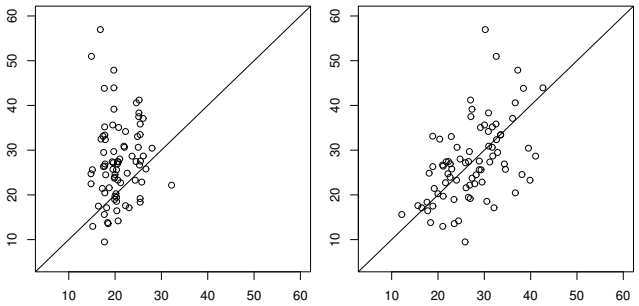

Fig. 3: The Predicted (*x*-axis) vs Observed (*y*-axis) Values for the *MR* (left) and the *NNESA* (right)

Table 3: The Relative Importance of the Input Variables (in percentage)

Task	Model	Attribute							
		Breed	**HCW**	**STF2**	**pH1**	**a***	**dE**	**dL**	**dB***
WBS	*NNE*	4.3	5.8	7.6	–	**50.3**	11.1	5.5	8.5
	NNESA	2.8	21.4	7.7	–	**41.7**	11.7	6.2	8.5
STP	*NNE*	**41.0**	–	5.1	6.6	22.6	7.5	–	3.8
	NNESA	**36.8**	–	20.1	9.3	22.4	9.7	–	1.7

4. Conclusions

In this work, a *Neural Network Ensemble based on Sensitivity Analysis (NNESA)* algorithm is proposed, aiming at the prediction of lamb meat tenderness. This real-world problem was addressed by two distinct regression tasks by using instrumental and sensory measurements. In both cases, the *NNESA* outperformed other *Neural Network* approaches, as well as a *Multiple Regression*. Furthermore, the final neural solution is much simpler, requiring only half the number of inputs (7/6 instead of 12). In addiction, the proposed method is noninvasive, much cheaper than the *WBS* or *STP* procedures, and can be computed just 24 hours after slaughters. This opens the room for the development of automatic tools for decision support (Turban et al., 2004). One drawback may be the obtained accuracy, which is still high when compared with the simple constant average predictor. Nevertheless, it should be stressed that the tested datasets are very small. Furthermore, as argued by Díez et al. (2004) , modeling sensory preferences is a very difficult regression task. To our knowledge, this is the first time lamb meat tenderness is approached by neural regression models and further exploratory research needs to be performed.

Another relevant issue regards the high importance of the **Breed** attribute in the *STP* task, which seems to contradict the animal science theory. The obtained results were discussed with the experts, which discovered that the *Mirandesa* lambs were considered less stringy and more odor intense. This behavior may be due to animal stress during slaughter, although further research needs to be addressed towards this issue. In future work, it is also intended to enrich the datasets by gathering more meat samples. Moreover, other nonlinear techniques (e.g. *Support Vector Machines*) will also be explored (Hastie et al., 2001).

References

Arvanitoyannis, I. and Houwelingen-Koukaliaroglou, M. 2003. Implementation of Chemometrics for Quality Control and Authentication of Meat and Meat Products. *Critical Reviews in Food Science and Nutrition*, 43(2):173–218.

Dietterich, T. 2001. Ensemble methods in machine learning. In Kittler, J. and Roli, F., editors, *Multiple Classifier Systems, LNCS 1857*, 1–15. Springer.

Diéz, J., Bayón, G., Quevedo, J., Coz, J., Luaces, O., Alonso, J., and Bahamonde, A. 2004. Discovering relevancies in very difficult regression problems: applications to sensory data analysis. In *Proc. of ECAI*, 993–994.

Efron, B. and Tibshirani, R. 1993. *An Introduction to the Bootstrap*. Chapman & Hall, USA.

Hastie, T., Tibshirani, R., and Friedman, J. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, NY, USA.

Haykin, S. 1999. *Neural Networks - A Compreensive Foundation*. Prentice-Hall, New Jersey, 2nd edition.

Hill, B., Jones, S., Robertson, W., and Major, I. 2000. Neural Network Modeling of Carcass Measurements to Predict Beef Tenderness. *Canadian Journal of Animal Science*, (80):311–318.

Huffman, K., Miller, M., Hoover, L., Wu, C., Brittin, H., and Ramsey, C. 1997. Effect of beef tenderness on consumer satisfaction with steaks consumed in the home and restaurant. *Journal of Animal Sc.*, 74(1):91–97.

Kewley, R., Embrechts, M., and Breneman, C. 2000. Data Strip Mining for the Virtual Design of Pharmaceuticals with Neural Networks. *IEEE Transactions on Neural Networks*, 11(3):668–679.

R Development Core Team 2004. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, http://www.R-project.org.

Turban, E., Aronson, J., and Liang, T. 2004. *Decision Support Systems and Intelligent Systems*. Prentice-Hall.

Venables, W. and Ripley, B. 2002. *Modern Applied Statistics with S*. Springer, New York, USA, fourth edition.

Author Biography

PAULO CORTEZ was born in Braga, Portugal and went to the University of Minho where he obtained an Engineering Degree (1995), a M.Sc. (1998) and a Ph.D (2002) in Computer Science. Currently, he is a Lecturer at the Department of Information Systems. He is co-author of more than forty publications in int. conferences, journals and book chapters. Research interests: Data Mining and Machine Learning; Neural and Evolutionary Computation; and Forecasting. Web-page: http://www.dsi.uminho.pt/˜pcortez