Appl Intell
DOI 10.1007/s10489-011-0284-1

ORIGINAL PAPER

A study on the medium-term forecasting using exogenous variable selection of the extra-virgin olive oil with soft computing methods

Antonio J. Rivera · Pedro Pérez-Recuerda · María Dolores Pérez-Godoy · María Jose del Jesús · María Pilar Frías · Manuel Parras

Abstract Time series forecasting is an important task for the business sector. Agents involved in the olive oil sector consider that, for the olive oil price, medium-term predictions are more important than short-term predictions. In collaboration with these agents the forecasting of the price of extra-virgin olive oil six months ahead has been established as the aim of this work. According to expert opinion, the use of exogenous variables and technical indicators can help in this task and must be included in the forecasting process. The amount of variables that can be considered makes necessary the use of feature selection algorithms in order to reduce the number of variables and to increase the interpretability and usefulness of the obtained forecasting system. Thus, in this paper CO^{2}RBFN, a cooperative-competitive algorithm for Radial Basis Function Network design, and other soft computing methods have been applied to the data sets with the whole set of input variables and to the data sets with the selected set of input variables. The experimentation carried out shows that CO^{2}RBFN obtains the best results in medium term forecasting for olive oil prices with the whole and with the selected set of input variables. Moreover, the feature selection methods applied to the data sets highlighted some influential variables which could be considered not only for the prediction but also for the description of the complex process involved in the medium-term forecasting of the olive oil price.

Keywords Forecasting · Olive-oil price · Feature selection · Technical indicators · Cooperative-competitive · Evolutionary algorithms

A.J. Rivera (✉) · P. Pérez-Recuerda · M.D. Pérez-Godoy · M.J. del Jesús
Dept. of Computer Science, University of Jaén, Jaén, Spain
e-mail: arivera@ujaen.es

P. Pérez-Recuerda
e-mail: pedro.perez.recuerda@ujaen.es

M.D. Pérez-Godoy
e-mail: lperez@ujaen.es

M.J. del Jesús
e-mail: mjjesus@ujaen.es

M.P. Frías
Dept. of Statistics and Operation Research, University of Jaén, Jaén, Spain
e-mail: mpfrias@ujaen.es

M. Parras
Dept. of Marketing, University of Jaén, Jaén, Spain
e-mail: mparras@ujaen.es

1 Introduction

Temporal data mining [43, 69] is a growing area concerned with mining sequential data, a kind of data ordered with respect to some index. Time series constitute a popular class of sequential data, where records are indexed by time. Forecasting a time series is a common problem in many domains of science such as economics, electricity, hydrology, etc. The interest in this kind of problems is motivated by different reasons including the need to control a given process, the economic profits obtained or a high availability of such data, among others.

Nowadays, olive oil is an important business sector in an expanding market. According to the International Olive Council,[1] in 2009/2010, 2,888,000 tons of olive oil were produced worldwide and Spain, with 1,395,000 tons produced, was the first producer and exporter.

[1] http://www.internationaloliveoil.org

The Official Market for the negotiation of future contracts for olive oil (MFAO)[2] in Spain is a society whose objective is to forecast prices to balance supply and demand in future periods of time. The aim of this work is to predict these future prices in order to increase the global benefits of the sector.

Time series forecasting problem has been addressed for a long time by statistics/econometrics methods [9]. However, in recent years soft-computing methods have achieved accurate solutions [4, 55, 76], even better than traditional statistics methods, as can be seen in the references. The authors have developed an algorithm for the cooperative-competitive design of Radial Basis Functions Networks, CO^{2}RBFN [62], that has been successfully used in short-term forecasting of time series.

In [62], CO^{2}RBFN is used to predict the price of extra-virgin olive oil one week ahead. From the analysis of the time series, next week price is predicted using the past five weeks, so not exogenous variables have been used. Test data set is composed by the last twenty weeks of 2005 and weeks from the 32nd week of 2000 to the 32th week of 2005 were used for training. The results obtained are compared with typical soft-computing and statistical techniques.

Agents of the olive oil sector consider more important a medium-term prediction than a short-term prediction of the olive oil price, specially for the Official Market for the negotiation of future contacts for olive oil. The objective of the present paper is to forecast the price at source of the extra-virgin olive oil six months ahead. To help in this task the price itself as well as up to 9 exogenous variables (such as price at destination, opening and closing stock, consumer price index, etc.) have been taken into account. With the aim of preprocessing these input variables, technical indicators such as momentums, oscillators, disparities, etc. are used. Due to the combination of technical indicators and exogenous variables a high number of input variables are obtained. Therefore, the application of feature selection algorithms is also analyzed in order to determine the more influential variables in the forecasting of the extra-virgin olive oil price.

Besides CO^{2}RBFN, a diversity of soft-computing methods have been applied to olive oil price data sets, such as a Fuzzy System developed with a GA-P algorithm (FuzzyGAP) [70], a MultiLayer Perceptron Network trained using a Conjugate Gradient learning algorithm (MLPConjGrad) [56], a support vector machine (NU-SVR) [20] and a classical design method for Radial Basis Function Network learning (RBFNLMS) [79].

The rest of the paper is organized as follows: Sect. 2 discusses generalities about time series analysis and forecasting, briefly describes the feature selection field and reviews the Radial Basis Function Network design for forecasting problems. In Sect. 3, CO^{2}RBFN applied to time series forecasting is detailed. The experimental framework is described in Sect. 4. The results obtained for the forecasting methods used are detailed in Sect. 5. In Sect. 6, conclusions and future work are outlined.

[2] http://www.mfao.es

2 Background

In this section the areas of time series analysis and forecasting, feature selection and a brief review of the Radial Basis Function Networks for time series forecasting will be introduced.

2.1 Time series analysis and forecasting

As mentioned, the importance of time series analysis and forecasting [9] has grown in science, engineering and business. A time series is an ordered sequence of values taken by a variable at equally spaced time intervals. Basically, a time series can be represented as a curve that evolves over time. To forecast a time series means to obtain a model that extends the historical values in the future, where the data are not yet available.

Before obtaining a forecasting model, the time series must be analyzed. Time series analysis can be formally defined as a set of statistics that measures structural dependencies among the observed data of the given variable.

Concerning to the financial time series area [73], a fundamental and a technical analysis can be distinguished. Fundamental analysis involves delving into the financial statements by examining related economic and company-specific information; this involves looking at revenue, expenses, assets, liabilities and all the other financial aspects of an organization. On the other hand, technical analysis takes a completely different approach and it does not care the intrinsic values of an organization. Technical analysis [1, 58] (sometimes called chartists) is only interested in the price movements of the market, identifying patterns and using them in order to predict future prices. Examples of indicators used for technical analysis are: momentums, moving averages, oscillators, convergences-divergences, etc.

Soft computing methods are progressively demonstrating their efficiency in the financial world [4, 8, 55, 76]. Different arguments can justify the use of soft computing approaches for mining financial data, such as [55]: large data sets, dealing with an ill-structured problem, better understanding of financial dynamics, etc. Different applications of soft computing methods to financial data can be found, based on neural networks [6, 14, 64, 74, 75] or based on fuzzy systems [5, 44, 45, 82].

In time series forecasting the predicted value is typically estimated from past values (1)

$$\hat{x}_{t+h} = f(x_t, x_{t-1}, \ldots, x_{t-k}) \qquad (1)$$

where $\hat{x}_{t+h}$ is the value to forecast, h is commonly known as the prediction horizon and k is the maximum lag used in the forecasting. Nevertheless a future value is usually influenced by some external (or exogenous) information (as the outside temperature when one tries to estimate the water consumption in a country, or the harvest season when one tries to estimate the price of the product). In this case, the external information can be incorporated into the model, usually in the form of the commonly-known exogenous variables (2).

$$\hat{x}_{t+h} = f(x_t, x_{t-1}, \ldots, e_1, e_2, \ldots) \quad (2)$$

This external information can be summarized by technical indicators [16]. In a real world application, it is difficult to know how much information (in terms of number of variables, or size of input vector) must be used to properly learn the dynamics of a time series. Obviously, the quantity of information increases with the number of variables and may cause different type of problems (as can be seen in next subsection). In these environments, a feasible methodology [23, 32, 47] is to choose the largest possible number of variables to be taken into account (past values of the series and exogenous variables that could influence the series) and then apply feature selection methods in order to transform the initial set of variables into another smaller set of state variables, keeping as much as possible the information contained in the original set.

2.2 Data preprocessing

Data mining is an integral part of Knowledge Discovery in Databases [51], the overall process of converting raw data into useful information. Data preprocessing, the previous stage to data mining phase, is the stage where data reliability is enhanced by means of tasks such as data fusing, data cleaning, feature selection, etc.

Data sets can have a large number of attributes, which can became a serious obstacle to the efficiency of most of the data mining algorithms [52]. This obstacle is sometimes known as the "curse of dimensionality" [19] and can be addressed with dimensionality reduction or feature selection techniques [34].

There are a variety of benefits in carrying out feature selection. First, irrelevant features, and their associated noise, can be eliminated. Another benefit is that a reduction of the dimensionality can lead to a more understandable or more easily visualized model, because it involves fewer attributes. Finally, the amount of time and memory required by the data mining algorithm is reduced with the decrease of the input features.

Data mining algorithms are computationally intensive. There is a typical trade-off between the error rate obtained by a data mining model and the cost of its obtaining [13]. Besides, on the complexity of the data mining algorithm that

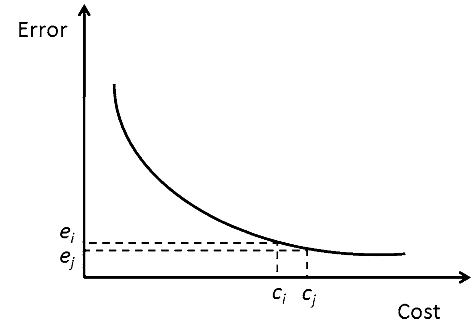

Fig. 1 Cost-error trade-off in the development of Data Mining models

derives the model, this cost depends on the size (number of variables and instances) of the data set. According to [13], the size of the data set is correlated with the number of instances and attributes, and is often used as an estimator to the mining cost. Theoretically, knowing the exact functional relation between the cost and the error (see Fig. 1) the ideal data mining model can be chosen. On some occasions, one might prefer using an inferior model that uses only part of the data and produces an increased error rate.

In the field of dimension reduction [34], a feature selection technique must select the best subset of input features, eliminating irrelevant or redundant attributes. A typical approach to feature selection is to try all possible subsets of features as input and then take the subset that produces the best result. Unfortunately, this approach is impractical because the number of subsets involving n attributes is 2^n. For that reason, different search strategies that control the generation of new sets of features can be used.

There are two standard approaches to feature selection: filter and wrapper methods. In filter methods, features are selected before the data mining algorithm is run, using some approach that is independent of the data mining task. The evaluation is often based on the separability of the classes, correlations, etc. In a wrapper approach the target data mining algorithm is used to obtain the evaluation measure of a given subset of attributes. In this work, the filter approach is used because it operates independently of the learning algorithm without biasing the results. Also, filter approach has proven to be much faster than the wrapper approach and hence can be applied to large data sets containing many features.

2.3 RBFNs for forecasting problems

In this section Radial Basis Function Networks and their design process are described.

2.3.1 Radial Basis Function Networks

Radial Basis Function Networks (RBFNs) are an artificial neural network paradigm [10] with characteristics as simple topological structure and universal approximation ability

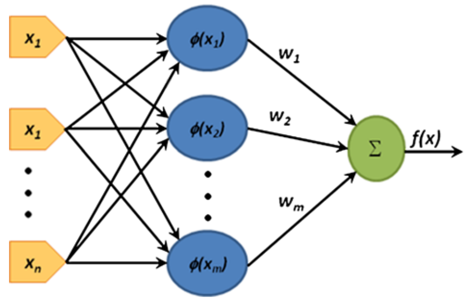

Fig. 2 RBFN Topology for time series forecasting

[60]. They have been successfully used in time series prediction [46, 48, 54, 68, 74, 78, 82].

From a structural point of view, an RBFN is a feed-forward neural network with three layers: an input layer with n nodes, a hidden layer with m neurons or RBFs, and an output layer with one node, in time series forecasting (see Fig. 2).

The m neurons of the hidden layer are activated by a radially-symmetric basis function, $\phi_i : R^n \to R$, which can be defined in several ways, being the Gaussian function the most widely used:

$$\phi_i(\mathbf{x}) = \phi_i\left(e^{-(\|\mathbf{x}-\mathbf{c}_i\|/d_i)^2}\right) \tag{3}$$

where $\mathbf{c}_i \in R^n$ is the centre of basis function ϕ_i, $d_i \in R$ is the width (radius), and $\| \|$ is typically the Euclidean norm on R^n. This expression is the one used in this paper as Radial Basis Function (RBF). The output of a basis function will be high when the input vector and the centre of this basis function are closer, always taking into account the value of the radius. The output node implements the following function:

$$f(\mathbf{x}) = \sum_{i=1}^{m} w_i \phi_i(\mathbf{x}) \tag{4}$$

The weights w_i show the contribution of an RBF to the output node, and therefore the output node implements the weighted sum of RBF outputs (4).

2.3.2 RBFN design processes

The objective of any RBFN design process is to determine centres, widths and the linear output weights connecting the RBFs to the output neuron layer. The most traditional learning procedure has two stages: first, unsupervised learning of centres and widths, and then, supervised learning of output weight. Clustering techniques [61] are normally used to adjust the centres. Regarding to the widths, they may all be given the same value, may reflect the width of the previously calculated clusters (i.e., RBFs), or may be established as the average distance between RBFs, among other possibilities. In order to obtain the weights in the second stage, algorithms such as Least Mean Square (LMS) [79] or Singular Value Decomposition (SVD) [31] can be used. In [35, 49, 74, 82] this methodology has been applied in forecasting tasks.

As well as this typical methodology, different strategies for RBFN design can be found based on deciding which RBFs to aggregate or eliminate, such as, in forecasting problems the algorithms described in [46] and [57]. One disadvantage of this kind of methods is that they could become trapped in local optima.

An important paradigm for the RBFN design which overcomes this limitation is Evolutionary Computation (EC) [7, 29, 40]. EC uses natural evolution and stochastic searching to design optimization algorithms. More precisely, EC maintains a population of individuals, which evolves according to operators as mutation, recombination or selection, and each individual in the population receives a measure of its fitness in the environment.

Reviews of EC applied to RBFN design can be found in [11, 38]. Examples of RBFN design algorithms applied to time series forecasting can be found in [12, 17, 54, 67, 72]. In most of the proposals within this evolutionary paradigm, an individual represents a whole RBFN, and different operators are applied to the entire population to improve individual fitness. Nevertheless EC presents some difficulties for certain learning problems [65], especially when an individual represents a complete solution (i.e. a network) made of independent subcomponents. In these situations, the individuals can have a premature tendency to convergence to a local optima and it is difficult to consider properly the role of the subcomponents in the whole solution.

An alternative to this evolutionary approach is the cooperative-competitive evolutionary or cooperative—coevolutionary strategy [65, 68, 78], which provide a framework within which an individual of the population represents only a part of the solution, evolving in parallel and competing to survive, but at the same time cooperating in order to find a common solution (the complete RBFN). This approach has the advantage of being computationally less complex but it must address two problems: credit assignment, or the fitness allocated to each individual according to its contribution to the final solution, and the mechanism used in order to maintain diversity among individuals of the population.

Different examples of RBFNs applied to Financial analysis can be found [47, 50, 59, 77, 81], for forecast stock market index, exchange-trade fund DIA, stock data and financial time series, respectively with the following main characteristics:

- Among them, different sort of neural networks are used as Gaussian RBFNS [47, 50, 81], Local Linear Radial Basis Function Network [59] and mixtures of different neural networks [77].

```
1. Initialize RBFN
2. Train RBFN
3. Evaluate RBFs
4. Apply operators to RBFs
5. Substitute the eliminated RBFs
6. Select the best RBFs
7. If the stop condition is not
   verified go to step 2
```

Fig. 3 Main steps of CO^2RBFN

– Different approaches for the design and training processes as an immune algorithm [81], the PXtrac algorithm [50] and a particle swarm optimization algorithm [59] are considered.
– The objective for these neural networks are the prediction for future values but in [50] the algorithm is used to recognize irregularities underlying the time series.
– Most of the proposals use only past values of the series but in [47] many exogenous variables are integrated, as our proposal does.

3 CO^2RBFN for time series forecasting

CO^2RBFN [63], is an hybrid evolutionary cooperative-competitive algorithm for the design of RBFNs. As mentioned, in this algorithm each individual of the population corresponds, using a real representation, to an RBF and the entire population is responsible for the final solution. The individuals cooperate towards a definitive solution, but they must also compete for survival. In this environment, in which the solution depends on the behavior of many components, the fitness of each individual is known as credit assignment. In order to measure the credit assignment of an individual, three factors have been proposed: the RBF contribution to the network output, the error in the basis function radius and the degree of overlapping among RBFs.

The application of the operators is determined by a Fuzzy Rule-Based System. The inputs of this system are the three parameters used for credit assignment and the outputs are the operators' application probability.

The main steps of CO^2RBFN, explained in the following subsections, are shown in Fig. 3 in pseudocode.

RBFN initialization. To define the initial network, a specified number m of neurons (i.e. the size of population) is randomly allocated among the different patterns of the training set. To do so, each RBF centre, $\mathbf{c}_i$, is randomly established to a pattern of the training set. The RBF widths, d_i, will be set to half the average distance between the centres. Finally, the RBF weights, w_{ij}, are set to zero.

RBFN training. The Least Mean Square algorithm [79] has been used to calculate the RBF weights. This technique exploits the local information that can be obtained from the behaviour of the RBFs.

RBF evaluation. A credit assignment mechanism is required in order to evaluate the role of each RBF ϕ_i in the cooperative-competitive environment. For an RBF, three parameters, a_i, e_i, o_i are defined:

– The contribution, a_i, of the RBF ϕ_i, $i = 1 \dots m$, is determined by considering the weight, w_i, and the number of patterns of the training set inside its width, pi_i. An RBF with a low weight and few patterns inside its width will have a low contribution:

$$a_i = \begin{cases} |w_i| & \text{if } pi_i > q \\ |w_i| * (pi_i/q) & \text{otherwise} \end{cases} \tag{5}$$

where q is the average of the pi_i values minus the standard deviation of the pi_i values.
– The error measure, e_i, for each RBF ϕ_i, is obtained by calculating the Mean Absolute Percentage Error (MAPE) inside its width:

$$e_i = \frac{\sum_{\forall p_i} \left| \frac{f(p_i) - y(p_i)}{f(p_i)} \right|}{npi_i} \tag{6}$$

where $f(p_i)$ is the output of the model (4) for the point p_i, inside the width of RBF ϕ_i, $y(p_i)$ is the real output at the same point, and npi_i is the number of points inside the width of RBF ϕ_i.
– The overlapping of the RBF ϕ_i and the other RBFs is quantified by using the parameter o_i. This parameter is computed by taking into account the fitness sharing methodology [30], whose aim is to maintain the diversity in the population. This factor is expressed as:

$$o_i = \sum_{j=1}^{m} o_{ij} \tag{7}$$

$$o_{ij} = \begin{cases} (1 - \|\phi_i - \phi_j\|/d_i) & \text{if } \|\phi_i - \phi_j\| < d_i \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where o_{ij} measures the overlapping of the RBF ϕ_i and ϕ_j $j = 1 \dots m$.

Applying operators to RBFs. In CO^2RBFN four operators have been defined in order to be applied to the RBFs:

– Operator Remove: eliminates an RBF.
– Operator Random Mutation: modifies the centre and width of an RBF in a percentage below 50% of the old width.
– Operator Biased Mutation: modifies the width and all coordinates of the centre using local information of the RBF environment. The technique used follows the recommendations in [28] that are similar to those used by the LMS algorithm. The error of the patterns within the radius of

Table 1 Fuzzy rule base representing expert knowledge in the design of RBFNs

Antecedents				Consequents			
	va	ve	vo	p_{remove}	p_{rm}	p_{bm}	p_{null}
R1	L			M-H	M-H	L	L
R2	M			M-L	M-H	M-L	M-L
R3	H			L	M-H	M-H	M-H
R4		L		L	M-H	M-H	M-H
R5		M		M-L	M-H	M-L	M-L
R6		H		M-H	M-H	L	L
R7			L	L	M-H	M-H	M-H
R8			M	M-L	M-H	M-L	M-L
R9			H	M-H	M-H	L	L

the RBF, ϕ_i, are calculated. For each coordinate of the radius and the center the values Δd_i and Δc_{ij} (see (9) and (10)) are respectively calculated. The new coordinates and the new radius are obtained by changing (increasing or decreasing) its old values to a random number (between 5% and 20% of its old width), depending on the sign of the value calculated.

$$\Delta d_i = \sum_k e(\overrightarrow{p_k}) \cdot w_i \tag{9}$$

where $e(\overrightarrow{p_k})$ is the error for the pattern $\overrightarrow{p_k}$.

$$\Delta c_{ij} = sign(c_{ij} - p_{kj}) \cdot e(\overrightarrow{p_k}) \cdot w_i \tag{10}$$

– Operator Null: in this case all the parameters of the RBF are maintained.

The operators are applied to the whole population of RBFs. The probability of choosing an operator is determined by means of a Mandani-type fuzzy rule based system [53] which represents expert knowledge about the operator application in order to obtain a simple and accurate RBFN. The inputs of this system are the parameters a_i, e_i and o_i used to define the credit assignment of the RBF ϕ_i. These inputs are considered as the linguistic variables va_i, ve_i and vo_i. The outputs, p_{remove}, p_{rm}, p_{bm} and p_{null}, represent the probability of applying Remove, Random Mutation, Biased Mutation and Null operators, respectively.

Table 1 shows the rule base used to relate the described antecedents and consequents. The rule base provides a set of simple guidelines from heuristics that represent expert knowledge to be used in the design of RBFNs and that have been successfully used in past research [62, 63]. For example, in Table 1 where each row represents a rule, the interpretation of the first rule is: if the contribution of an RBF is Low Then the probability of applying the operator Remove is Medium-High, the probability of applying the operator Random Mutation is Medium-High, the probability of applying the operator Biased Mutation is Low and the probability of applying the operator null is Low.

Introduction of new RBFs. In this step, the eliminated RBFs are substituted by new RBFs. The new RBF is located in the centre of the area with maximum error or in a randomly chosen pattern with a probability of 0.5 respectively.

The width of the new RBF will be set to the average of the RBFs in the population plus half of the minimum distance to the nearest RBF. Its weights are set to zero.

Replacement strategy. The replacement scheme determines which new RBFs (obtained before the mutation) will be included in the new population. To do so, the role of the mutated RBF in the net is compared with the original one to determine the RBF with the best behaviour in order to include it in the population.

4 Experimental framework

In collaboration with Poolred,[3] an initiative of the Foundation for the Promotion and Development of the Olive and Olive Oil, located in Jaén (Spain), the time series of the monthly extra-virgin olive oil price per ton at source in Spain has been obtained (see Fig. 4). Concretely, the time series contains data from the 1st month of 2002 to the 12th month of 2009.

4.1 Exogenous variables and technical indicators

The chosen exogenous variables or stock indexes, that contributes to predict the extra-virgin olive oil, are shown in Table 2. As can be seen the source of these variables/indexes are: the cited Poolred, the Agency for the Olive Oil[4] in Spain, the National Institute of Statistic of Spain[5] and the Ministry of Industry, Tourist and Trade.[6] All these variables/indexes are monthly.

With the aim of extracting additional information of the above data, a set of technical indicators, frequently referenced in the specialized bibliography [4], have been used and are shown in Table 3. In this table, i_t is the value of the index at time t, H_{t-k} and L_{t-k} are the highest and lowest values respectively, during a period of time k, and H_n and L_n are the highest and lowest values respectively from the beginning of the time series.

As we have managed nine exogenous variables, besides the source price of the extra-virgin olive oil itself and six technical indicators, besides the absolute or raw value of each variables, the first experiments, taking into account all

[3] http://www.oliva.net/poolred/
[4] http://aao.mapa.es
[5] http://www.ine.es
[6] http://www.mityc.es

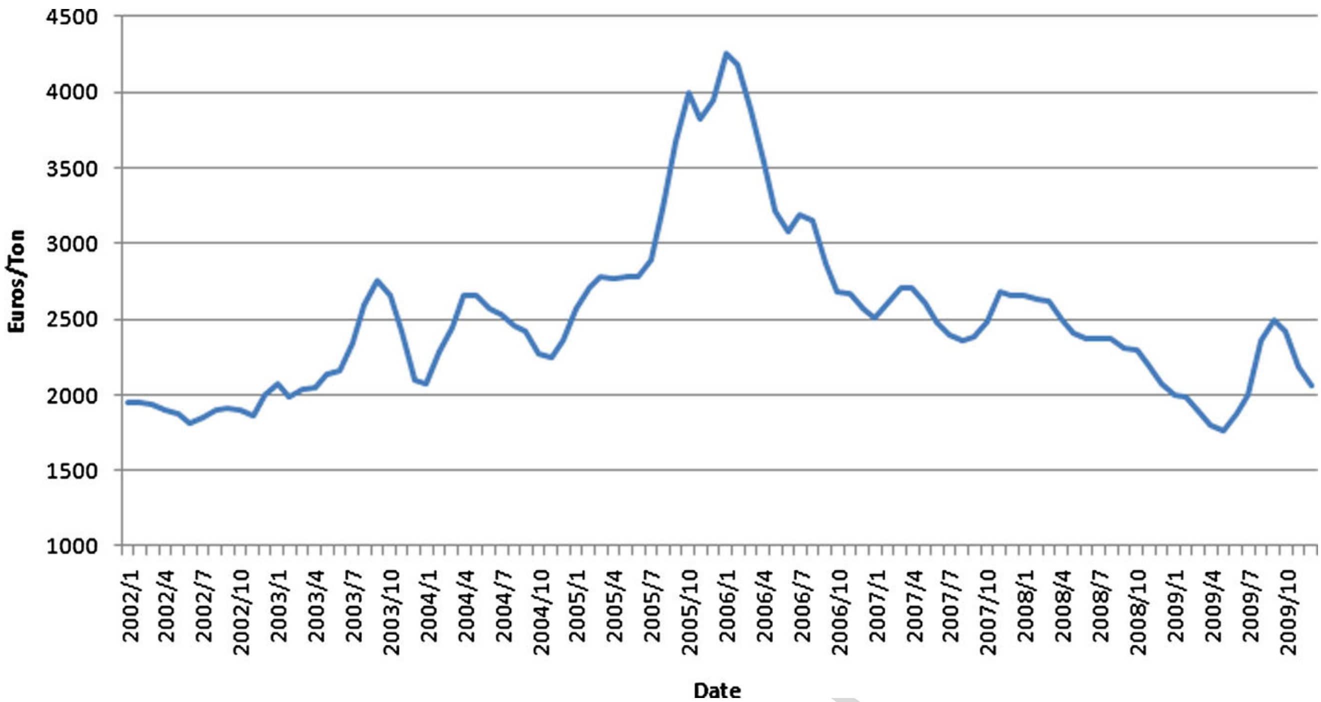

Fig. 4 Time series of the extra-virgin olive oil price

Table 2 Exogenous variables used to forecast the olive oil price

Variable	Description	Source
TgtPrice	Target Price of the extra-virgin olive oil	Poolred
OpStock	Opening stocks of olive oil	Agency for the Olive Oil
ClStock	Closing stocks of olive oil	Agency for the Olive Oil
InMarket	Trades in Internal Market of olive oil	Agency for the Olive Oil
Imports	Imports of olive oil	Agency for the Olive Oil
Exports	Exports of olive oil	Agency for the Olive Oil
ConMK	Consumption of olive oil in millions of kilos	Ministry of Industry, Tourist and Trade
GenCPI	General Consumer Price Index	National Institute
FoodCPI	Food Consumer Price Index	National Institute of Statistics

the combinations, are composed by seventy input variables. All the variables and technical indicators managed can be seen in Table 10. Also, data are normalized in the interval [0, 1].

4.2 Feature selection algorithms

In order to carry out the feature selection [34], the Weka data mining software [36] is used.

Table 3 Technical indicators and their formulas

Feature name	Description	Formula
Momentum 1	Measures the change of an index over a time span of one moth	$i_t - i_{t-1}$
Momentum 3	Measures the change of an index over a time span of three moths	$i_t - i_{t-3}$
Momentum 6	Measures the change of an index over a time span of six moths	$i_t - i_{t-6}$
Stochastic %k	Measures the last value of the index relative to its price range over a given time period. $k = 6$ in our case.	$\frac{i_t - L_{t-k}}{H_{t-k} - L_{t-k}} \times 100$
Williams %R	Larry William's %R. A momentum indicator that measures overbought/oversold levels	$\frac{H_n - i_t}{H_n - L_n} \times 100$
Disparity6	6-day disparity. Means the distance of current price and the moving average of 6 days	$\frac{i_t}{MA_6} \times 100$

As mentioned a filter approach has been chosen because it operates independently of the learning algorithm without biasing the results, is much faster than the wrapper approach and hence can be applied to large data sets containing many features.

In Weka for feature selection tasks the feature evaluator and a search method, that defines the set of attributes, can be chosen independently. In this work, as feature evaluator the CfsSubsetEval [37] method has been chosen. CfsSubsetEval evaluates the worth of a feature subset by calculating feature-class and feature-feature correlations. Feature subsets with high correlation with the class and low intercorrelations among the features, are preferred. These two objectives are combined in one in [37] by means a measure of symmetrical uncertainty [66] based on information theory [34]. For a wider explanation please refer to Appendix A.

With the objective of determining the best attribute subset, the following search methods have been chosen:

– BestFirst [33]: Searches the space of attribute subsets by greedy hillclimbing augmented with a backtracking facility. Best first starts with the empty set of attributes and search forward (by considering all possible single attribute additions and deletions at a given point).
– GeneticSearch [29]: Performs a search using a simple genetic algorithm.
– GreedyStepwise [33]: Performs greedy forward search through the space of attribute subsets. It starts with no attributes. Stops when the addition/deletion of any remaining attributes results in a decrease in evaluation.
– LinearForwardSelection [33]: As an extension of BestFirst, it takes a restricted number of k attributes into account. Fixed-set selects a fixed number k of attributes, whereas k is increased in each step when fixed-width is selected. The search uses either the initial ordering to select the top k attributes, or performs a ranking.
– ScatterSearch [22]: An scatter search through the space of attribute subsets is performed. It start with a population of many significants and diverse subsets and stops when the result is higher than a given threshold or there is not more improvement.
– SubsetSizeForwardSelection [33]: It is an extension of LinearForwardSelection. The search performs an interior cross-validation (with a number of specified folds). A linear forward selection is performed on each fold to determine the optimal subset-size (using the given subset size evaluator). Finally, a linear forward selection up to the optimal subset-size is performed on the whole data.

CfsSubEval has been run as evaluator method for all the search methods obtaining six feature selection methods.

4.3 Algorithms for comparison results

In order to compare we have chosen the following soft computing methods:

– FuzzyGAP method [70]. A GA-P method [42] uses an evolutionary computation method, hybridation between a genetic algorithms and a genetic programming, optimized to perform symbolic regressions. Each element comprises a chain of parameters and a tree which describes a function, depending on these parameters. The new members of the population are generated by means of crossover and mutation. In the GA-P algorithm both operations are performed independently over the tree and the parameter chain. In the FuzzyGAP algorithm, the fuzzy sets of the fuzzy model are codified on the terminal nodes of the tree and fuzzy arithmetic operators are used to evaluate the tree.
– MLPConjGrad [56]. MLPConjGrad uses the conjugate gradient algorithm to adjust weight values of a multilayer perceptron [39]. Compared to gradient descent, the conjugate gradient algorithm takes a more direct path to the optimal set of weight values. Usually, the conjugate gradient is significantly faster and more robust than the gradient descent. The Conjugate gradient also does not require the user to specify learning rate and momentum parameters.
– RBFNLMS. Builds an RBFN with a pre-specified number of RBFs. By means of the k-means clustering algorithm [18] it chooses an equal number of points from the training set to be the centres of the neurons. Finally, it establishes a single radius for all the neurons as half the average distance between the set of centres. Once the centres and radio of the network have been fixed, the set of weights is analytically computed using the LMS algorithm [79].
– NU-SVR. The SVM (Support Vector Machine) model uses the sequential minimal optimization training algorithm and treats a given problem in terms of solving a quadratic optimization problem. The NU-SVR, also called v-SVM, for regression problems is an extension of the traditional SVM and it aims to build a loss function [20].

Table 4 Parameters used for CO^2RBFN

Parameter	Value
Generations of the main loop	200
Number of RBF's	10

The implementation of the rest of the data mining methods has been obtained from KEEL [2, 3]. The main parameters used are set to the values indicated by the authors. The parameters used for CO^2RBFN are shown in Table 4.

4.4 Statistical tests for evaluation

In order to asses the results by statistical analysis, data is partitioned as is shown in the Table 5.

To estimate prediction capacity, the Mean Absolute Percentage Error, MAPE, is calculated.

$$\mathrm{MAPE} = \sum_{i}^{z} \left(\left| \left(f(x_i) - y(x_i) \right) / f(x_i) \right| \right) / z, \tag{11}$$

where $f(x_i)$ is the predicted output of the model, $y(x_i)$ is the desired output and z is the number of patterns in the data set.

In order to statistically support the analysis of the results, we will use hypothesis testing techniques [24, 71]. Specifically, we will apply non-parametric tests due to the fact that the initial conditions that guarantee the reliability of the parametric tests may not be satisfied, causing the statistical analysis to lose credibility with these parametric tests [15].

For multiple comparisons we use the Iman-Davenport test [71] to detect statistical differences among a group of results. We will use the Wilcoxon signed-rank test [80] as a non-parametric statistical procedure for performing pairwise comparisons between two algorithms.

Furthermore, we consider the average ranking of the algorithms in order to show graphically how good a method is with respect to its partners. This ranking is obtained by assigning a position to each algorithm depending on its performance for each data set (year), Table 5. The algorithm which achieves the best accuracy on a specific data set will have the first ranking (value 1); then, the algorithm with the second best accuracy is assigned rank 2, and so forth. This task is carried out for all data sets and finally an average ranking is computed as the mean value of all rankings.

For these tests, it is very interesting to compute the p-value associated to each comparison. The p-value goes from 0 to 1 and represents the lowest level of significance of a hypothesis that results in a rejection. In this manner, we can know whether two algorithms are significantly different and how different they are.

These tests are suggested in the studies presented in [15, 24–26], where their use in the field of Machine Learning is highly recommended. For a wider description on the use of these tests, please refer to Appendix B.

Table 5 Data sets

Data set	Training years							Test years
Test2006	2002	2003	2004	2005				2006
Test2007	2002	2003	2004	2005	2006			2007
Test2008	2002	2003	2004	2005	2006	2007		2008
Test2009	2002	2003	2004	2005	2006	2007	2008	2009

5 Results

In order to achieve our objective, to forecast the extra-virgin olive oil price at source (SrcPrice) six months ahead, the absolute value of this SrcPrice and nine exogenous variables Table 2, have been chosen and extra input variables have been obtained processing each exogenous variable with six technical indicators Table 3. To these initial data sets, with seventy input variables, CO^2RBFN and other soft computing methods have been applied.

With the aim of decreasing the number of input variables and increasing the interpretability of the problem, different feature selection algorithms have been applied. So, new data sets have been built with the previously selected variables. Finally, soft computing methods have been applied to these data sets and the results are analyzed.

5.1 Results obtained with the whole set of input variables

First, CO^2RBFN and the rest of soft computing methods are applied to the data sets composed by all the input variables. The results obtained average and standard deviation for 10 repetitions according to MAPE, are shown in Table 6. As can be observed, CO^2RBFN has the lowest average error, followed by RBFNLMS (the other RBFN design method). CO^2RBFN has also the lowest average standard deviation that implies a good robustness.

Next, we use hypothesis testing techniques to provide statistical support to the analysis of the results. From [27] we are aware that with the number of data sets managed in this paper, statistical techniques can not applied over the best circumstances, but we have included it because they may provide more additional information than the error means analysis. Table 7 shows the average ranking computed for all approaches according to the MAPE error, where we can observe that CO^2RBFN has obtained the lowest value in the ranking and therefore it is the best algorithm.

The Iman and Davenport test obtains a p-value of 0.0625, which implies that there are significant differences (with a 93.75% of level of confidence) among the results of the different algorithms and thus we should apply a post-hoc test to detect which methods are outperformed by CO^2RBFN, since it is the best ranked method.

Table 6 MAPE test with the whole set of input variables

Year	CO^2RBFN	FuzzyGap	MLPConjGrad	NUSVR	RBFNLMS
Test2006	0.2366 ± 0.0418	0.2085 ± 0.1411	0.4209 ± 0.3329	0.3603 ± 0.3105	0.1953 ± 0.1090
Test2007	0.0630 ± 0.0209	0.2179 ± 0.0990	0.3901 ± 0.2338	0.2294 ± 0.2193	0.1284 ± 0.0949
Test2008	0.0999 ± 0.0194	0.1752 ± 0.0932	0.1524 ± 0.1391	0.1006 ± 0.1115	0.1156 ± 0.0944
Test2009	0.1998 ± 0.0255	0.2245 ± 0.1470	0.2747 ± 0.2254	0.1863 ± 0.0977	0.2539 ± 0.1689
Mean	$\mathbf{0.1498 \pm 0.0269}$	0.2065 ± 0.1201	0.3095 ± 0.2328	0.2191 ± 0.1847	0.1733 ± 0.1168

Table 7 Ranking of the algorithms for the whole set of input variables. The lower the value, the better

Algorithm	Ranking
CO^2RBFN	**1.75**
FuzzyGap	3.25
MLPConjGrad	4.75
NUSVR	2.75
RBFNLMS	2.5

Table 8 Wilcoxon test table for algorithms and all the input variables

R^+ CO^2RBFN	R^-		p-value
8	FuzzyGap	2	0.273
10	MLPConjGrad	0	0.068
8	NUSVR	2	0.273
8	RBFNLMS	2	0.273

Table 9 Bonferroni/Holm test table for algorithms and all the input variables

i	algorithm	$p_{Bonferroni}$	p_{Holm}
1	MLPConjGrad	0.0292	0.0292
2	FuzzyGap	0.7188	0.5391
3	NUSVR	1.4844	0.7422
4	RBFNLMS	2.0093	0.7422

Then, the Wilcoxon test, Table 8, is applied in order to detect significant differences between the behaviour of pairs of algorithms (pairwise test). The null hypothesis or the limit to establish significant differences (p-value) is shown in the table. As can be observed, significant differences with a high level of confidence are obtained with the MLPConjGrad algorithm.

Finally the results of applying Bonferroni and Holm tests, examples of multiple comparison test, are shown Table 9. As can be observed, from the p-values (or the limit to establish significant differences) of Bonferroni and Holm tests, significant differences with a high level of confidence are obtained with the MLPConjGrad algorithm.

5.2 Results of the feature selection algorithms

Feature selection methods, mentioned in 4.2, have been applied to the four data sets of Table 5. The results of applying feature selection methods are shown in Table 10. In this table, the first row shows the different indexes, the first column contains the different technical indicators (where Absolute means a raw variable when no technical indicator is applied) and each cell represents the number of times that an input variable (defined by the combination of row/column) is chosen by any feature selection algorithm in any data set (year). For example, the cell (row = 2/column = 2) shows that the input variable Absolute/SrcPrice is chosen 12 times by different feature selection algorithms and data sets, but no feature selection algorithm has chosen the input variable Momentum1/SrcPrice for any year.

Thus, we can identify important exogenous variables that often are selected regardless of the technical indicator used to preprocess it. These variables, that can be said that influence the price of the extra-virgin olive oil price six months ahead, are (sorted by the number of times that they have been selected): SrcPrice, FoodCPI, GenCPI, TgetPrice and ConMK. We can conclude that the SrcPrice is the most important variable to take into account in order to predict the future price of extra-virgin olive oil. There is a second group, that have been selected in a similar number of times, to predict the extra-virgin olive oil price: FoodCPI, GenCPI an TgetCPI. The last variable to highlight is ConMK that have been selected moderately. The rest of the variables are punctually selected.

In order to build the new data sets according to the results of the feature selection algorithms, the selected input variables are those that have been chosen at least one time for any feature selection algorithm in any year. This selection aims to minimize the amount of information loss.

5.3 Results obtained with the selected set of input variables

Finally, CO^2RBFN and the rest of soft computing methods are applied to data sets composed only by the selected set of input variables. The results obtained, average and standard deviation for 10 repetitions according to the MAPE error, are shown in Table 11. Also in this case, the CO^2RBFN approach achieves the better result in test (in average and standard deviation) among all the algorithms compared in this study.

The results obtained, but not better, are similar to the results with the whole set of input variables. In any case the objectives of simplifying the problem and identifying for the sector the input variables that influence the future price of the olive oil have been achieved.

Table 12 shows the average ranking computed for all approaches according to the MAPE error. As can be observed in this ranking, CO^2RBFN has obtained the lowest value in the ranking and therefore it is the best algorithm.

For the Iman and Davenport test, a p-value of 0.0222 is obtained, which implies that there are significant differences (with a 97.78% of level of confidence) among the results. With this level of confidence, the Wilcoxon test is also applied in order to detect which methods are outperformed by CO^2RBFN, since it is the best ranked method.

In Table 13 the null hypothesis or the limit to establish significant differences for the Wilcoxon test are shown. Also

Table 10 Results of applying feature selection methods

	SrcPrice	TgetPrice	OpStock	ClStock	InMarket	ConMK	Imports	Exports	GenCPI	FoodCPI	Total
Absolute	12	6	0	0	0	0	0	2	0	0	20
Momentum1	0	0	0	0	0	0	0	0	5	4	9
Momentum3	11	4	0	0	0	5	0	0	0	0	20
Momentum6	23	4	0	0	0	12	0	0	12	12	63
Stochastic %k	0	12	5	1	0	0	0	0	0	1	19
Williams %R	14	6	0	0	0	4	0	2	18	10	54
Disparity6	1	0	0	0	0	0	2	0	0	13	16
Total	61	32	5	1	0	21	2	4	35	40	–

Table 11 MAPE test with the selected set of input variables

Year	CO^{2}RBFN	FuzzyGap	MLPConjGrad	NUSVR	RBFNLMS
Test2006	0.2454 ± 0.0483	0.2419 ± 0.1579	0.4893 ± 0.3916	0.3853 ± 0.2499	0.1434 ± 0.1101
Test2007	0.0711 ± 0.0312	0.2466 ± 0.1317	1.0056 ± 0.4911	0.4040 ± 0.1359	0.1875 ± 0.0453
Test2008	0.0963 ± 0.0138	0.2381 ± 0.1061	0.5144 ± 0.3724	0.2273 ± 0.1618	0.1319 ± 0.1053
Test2009	0.2090 ± 0.0287	0.1422 ± 0.0976	0.3309 ± 0.2046	0.1902 ± 0.1578	0.2326 ± 0.1604
Mean	$\mathbf{0.1555 \pm 0.0305}$	0.2172 ± 0.1233	0.5850 ± 0.3649	0.3017 ± 0.1764	0.1739 ± 0.1053

Table 12 Ranking of the algorithms for the selected set of input variables

Algorithm	Ranking
CO^{2}RBFN	**2**
FuzzyGap	2.5
MLPConjGrad	5
NUSVR	3.25
RBFNLMS	2.25

Table 13 Wilcoxon test table for algorithms and the selected set of input variables

R^+ CO^{2}RBFN	R^-		p-value
3.5	FuzzyGap	7	0.465
2.5	MLPConjGrad	10	0.068
3	NUSVR	9	0.144
2.33	RBFNLMS	7	0.465

in this case, significant differences with a high level of confidence can be established respect to MLPConjGrad algorithm.

Next, Bonferroni and Holm tests, have been applied and the limit to establish significant differences (p-values) are shown in Table 14. As with the whole set of input variables significant differences with a high level of confidence are obtained with the MLPConjGrad algorithm.

Table 14 Bonferroni/Holm test table for algorithms and the selected set of input variables

i	algorithm	$p_{Bonferroni}$	p_{Holm}
1	MLPConjGrad	0.0292	0.0292
2	NUSVR	1.0542	0.7907
3	FuzzyGap	2.6189	1.3094
4	RBFNLMS	3.2923	1.3094

As conclusions, CO^{2}RBFN has achieved the best results in average and standard deviation for the experimentations carried out. Also, it has obtained the first position in the rankings calculated and the limit to establish significant differences regarding the other soft computing methods have been calculated. RBFNLMS algorithm has also obtained good results, obtaining the second position in the rankings after CO^{2}RBFN. Besides, only methods based on RBFNs have maintained the error in the predictions for the data sets composed by selected variables with respect to the data sets composed by all the input variables. The rest of the methods has obtained worst results. These facts validate the use of RBFNs in forecasting problems.

6 Conclusions

Time series forecasting is an active research area and the interest in its results has increased specially for science, engineering and business. Olive oil has become an important business international sector where Spain is the first producer and exporter. The agents involved in this sector are interested in the use of forecasting methods for different tasks. This is especially important in the Official Market for the negotiation of futures contracts for olive oil (MFAO). In this line, and in collaboration with the Foundation for the Promotion and Development of the Olive and Olive Oil (Poolred) the problem of predicting the extra-virgin olive oil price six months ahead has been identified as an interesting task.

Soft computing methods, and particularly RBFNs, have demonstrated their efficiency in the resolution of forecasting problems. For this reason authors propose CO^2RBFN, a hybrid evolutionary cooperative-competitive algorithm for RBFN design in order to solve the given problem. In this proposal, a key point is the identification of the role (credit assignment) of each basis function in the whole network. In order to evaluate this value for a given RBF, three factors are defined and used: the RBF contribution to the network's output, a_i; the error in the basis function radius, e_i; and the degree of overlapping among RBFs, o_i. In order to drive the cooperative-competitive process four operators are used: Remove, Random Mutation, Biased Mutation (based on clustering) and Null. The application of these operators is determined by a fuzzy rule-based system which represents expert knowledge of the RBFN design. The inputs of this system are the three parameters used for credit assignment.

Different exogenous variables and technical indicators have been used, and CO^2RBFN and other soft computing methods have been applied to the initial data sets. The results obtained show that CO^2RBFN is the best method in measures as the average, the standard deviation and the ranking of the individual results per year. Besides, significant differences with a high level of confidence can be established in the results with respect to methods as MLPConjGrad.

In order to reduce the number of input variables and to increase the knowledge about the problem, different feature selection algorithms have been applied. From these results we can conclude that variables as price at source, price at destination, CPI general, food CPI and consumption influence the future price of the extra-virgin olive oil.

Finally, new data sets have been built with the previously selected variables and soft computing methods have been applied. Also for this case, CO^2RBFN is the best method in measures as the average, the standard deviation and the ranking of the individual results per year. Besides, significant differences with a high level of confidence can be established in the results with respect to methods as MLPConjGrad.

As future work, wrapper mechanisms of feature selection will be introduced in CO^2RBFN. In this way, we can observe the sets of selected variables obtained and the efficiency of the new proposal.

Acknowledgements Supported by the Spanish Ministry of Science and Technology under the Project TIN2008-06681-C06-02, FEDER founds, the Andalusian Research Plan TIC-3928 and the Project of the University of Jaén UJA-08-16-30.

Appendix A: CFS: Correlation based feature selection

CFS [37] is a feature selection method that evaluates the worth of a feature subset by calculating feature-class and feature-feature correlations. Feature subsets with high correlation with the class and low intercorrelations among the features, are preferred. When features are continuous, they are transformed to categorical features using the supervised discretisation method of [21].

In order to calculate the association between the features a measure based on Information Theory [34] is used. If X and Y are discrete random variables, (12) and (13) give the entropy of Y before and after observing X.

$$H(Y) = -\sum_{y \epsilon Y} p(y) \log_2 p(y) \tag{12}$$

$$H(Y|X) = -\sum_{x \epsilon X} p(x) \sum_{y \epsilon Y} p(y|x) \log_2 p(y|x) \tag{13}$$

The amount by which the entropy of Y decreases reflects the additional information about Y provided by X and is called the information gain. Information gain is given by:

$$\begin{aligned} gain &= H(Y) - H(Y|X) \\ &= H(X) - H(X|Y) \\ &= H(Y) + H(X) - H(X,Y) \end{aligned} \tag{14}$$

As information gain is biased in favour of attributes with more values, Symmetrical Uncertainty [66] is used to compensate this bias and to normalize its value to the range [0,1]:

$$symmetrical\ uncertainty = 2.0 \times \left[\frac{gain}{H(Y) + H(X)} \right] \tag{15}$$

Thus, CFS calculates feature-class and feature-feature correlations using symmetrical uncertainty for each feature subset.

Appendix B: On the use of non-parametric tests based on rankings

In this paper, we have made use of statistical techniques for the analysis of neural network methods, since they are a necessity in order to provide a correct empirical study [15, 24, 26]. Specifically, we have employed non-parametric tests, due to the fact that the initial conditions that guarantee the reliability of the parametric tests may not be satisfied, causing the statistical analysis to lose credibility [15].

« APIN 10489 layout: Large v.1.3.2 file: apin284.tex (petras) class: spr-twocol-v1.2 v.2011/02/19 Prn:2011/03/01; 16:05 p. 13/15»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: basic»

In this appendix we describe the procedures to performs pair and multiple comparisons. Specifically, for multiple comparison we have used an Iman-Davenport, Bonferroni and Holm test to detect statistical differences. We have employed a Wilcoxon signed-rank test as a non-parametric statistical procedure to perform pairwise comparisons between two algorithms. Furthermore, any interested reader can find additional information and the software for applying the statistical tests on the website http://sci2s.ugr.es/sicidm/.

B.1 Multiple comparisons: Iman-Davenport, Bonferroni and Holm tests

In order to perform a multiple comparison, it is necessary to check whether all the results obtained by the algorithms present any inequality. In the case of finding inequality then we can know, by using a post-hoc test, which algorithms partners' average results are dissimilar. Next, we describe the non-parametric tests used.

– The Iman and Davenport test [71] is a non-parametric test, derived from the Friedman test [71]:

$$F_F = \frac{(N_{ds} - 1)\chi_F^2}{N_{ds}(K-1) - \chi_F^2}$$

which is distributed according to the F-distribution with $k-1$ and $(k-1)(N_{ds}-1)$ degrees of freedom. Statistical tables for critical values can be found in [71, 83].

– The Bonferroni-Dunn's test [83]: it is a multiple comparison procedure which can work with a control algorithm and compares it with the remaining methods. The performance of two algorithms is significantly different if the corresponding average of rankings is at least as great as its critical difference (CD).

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

The value of q_α is the critical value for a multiple non-parametric comparison with a control.

– The Holm test [41]: it is a multiple comparison procedure which can work with a control algorithm and compares it with the remaining methods. The test statistics for comparing the ith and jth method using this procedure is:

$$z = (R_i - R_j)/\sqrt{\frac{k(k+1)}{6N_{ds}}}$$

The z value is used to find the corresponding probability from the table of normal distribution, which is then compared with an appropriate level of confidence α.

A Holm test is a step-up procedure that sequentially tests the hypotheses ordered by their significance. We will denote the ordered p-values by $p_1, p_2, \dots,$ so that $p_1 \le p_2 \le \dots \le p_{k-1}$. The Holm test compares each p_i with $\alpha/(k-i)$, starting from the most significant p value. If p_1 is below $\alpha/(k-1)$, the corresponding hypothesis is rejected and we can compare p_2 with $\alpha/(k-2)$. If the second hypothesis is rejected, the test proceeds with the third, and so on. As soon as a certain null hypothesis cannot be rejected, all the remain hypotheses are retained as well.

B.2 Pairwise comparisons: Wilcoxon signed-ranks test

This is the analogue of the paired t-test in non-parametrical statistical procedures; therefore, it is a pairwise test that aims to detect significant differences between the behaviour of two algorithms.

Let d_i be the difference between the performance scores of the two classifiers on i-th out of N_{ds} data sets. The differences are ranked according to their absolute values; average ranks are assigned in the case of ties. Let R^+ be the sum of ranks for the data sets on which the second algorithm outperformed the first, and R^- the sum of ranks for the opposite. Ranks of $d_i = 0$ are split evenly among the sums; if there is an odd number of them, one is ignored:

$$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \qquad (16)$$

$$R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \qquad (17)$$

Let T be the smallest of the sums, $T = min(R^+, R^-)$. If T is less than or equal to the value of the distribution of Wilcoxon for N_{ds} degrees of freedom (Table B.12 in [83]), the null hypothesis of equality of means is rejected.

References

1. Achelis S (2000) Technical analysis from A to Z, 2nd edn. McGraw-Hill, New York
2. Alcalá-Fdez J, Fernández A, García S, Del Jesus M, Ventura S, Garrell J, Otero J, Romero C, Bacardit J, Herrera F (2009) Keel: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput 13(3):307–318
3. Alcalá-Fdez J, Luengo J, Derrac J, García S, Sánchez L, Herrera F (2011) Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J Multiple-Valued Logic Soft Comput (in press)
4. Atsalakis G, Valavanis K (2009) Surveying stock market forecasting techniques—part ii: Soft computing methods. Expert Syst Appl 36(3, Part 2):5932–5941
5. Azadeh A, Saberi M, Ghaderi S, Gitiforouz A, Ebrahimipour V (2008) Improved estimation of electricity demand function by integration of fuzzy system and data mining approach. Energy Convers Manag 49(8):2165–2177
6. Aznarte J, Benítez J, Lugilde D, de Linares C, de la Guardia C, Sánchez F (2007) Forecasting airborne pollen concentration time series with neural and neuro-fuzzy models. Expert Syst Appl 32(4):1218–1225

Journal ID: 10489, Article ID: 284, Date: 2011-03-02, Proof No: 1, UNCORRECTED PROOF

« APIN 10489 layout: Large v.1.3.2 file: apin284.tex (petras) class: spr-twocol-v1.2 v.2011/02/19 Prn:2011/03/01; 16:05 p. 14/15»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: basic»

7. Bäck T, Hammel U, Schwefel H (1997) Evolutionary computation: comments on the history and current state. IEEE Trans Evol Comput 1(1):3–17
8. Bao D (2008) A generalized model for financial time series representation and prediction. Appl Intell 29(1):1–11
9. Box G, Jenkins G, Reinsel G (2008) Time series analysis: forecasting and control, 4th edn. Wiley, New York
10. Broomhead D, Lowe D (1988) Multivariable functional interpolation and adaptive networks. Complex Syst 2:321–355
11. Buchtala O, Klimek M, Sick B (2005) Evolutionary optimization of radial basis function classifiers for data mining applications. IEEE Trans Syst Man Cybern, Part B, Cybern 35(5):928–947
12. Chen C, Wu Y, Luk B (1999) Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks. IEEE Trans Neural Netw 10(5):1239–1243
13. Chizi B, Maimon O (2005) Dimension reduction and feature selection. In: The data mining and knowledge discovery handbook, pp 93–111. Springer, Berlin
14. Co H, Boosarawongse R (2007) Forecasting Thailand's rice export: statistical techniques vs. artificial neural networks. Comput Ind Eng 53(4):610–627
15. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
16. Dourra H, Siy P (2002) Investment using technical analysis and fuzzy logic. Fuzzy Sets Syst 127(2):221–240
17. Du H, Zhang N (2008) Time series prediction using evolving radial basis function networks with new encoding scheme. Neurocomputing 71:1388–1400
18. Duda R, Hart P (1973) Pattern classification and scene analysis. Wiley, New York
19. Elder J, Pregibon D (1996) A statistical perspective on knowledge discovery in databases. In: Advances in knowledge discovery and data mining. AAAI Press/MIT Press, Menlo Park/Cambridge
20. Fan R, Chen P, Lin C (2005) Working set selection using the second order information for training svm. J Mach Learn Res 6:1889–1918
21. Fayyad U, Irani K (1993) Multi-interval discretisation of continuous valued attributes for classification learning. In: Proceedings of the thirteenth international joint conference on artificial intelligence. Morgan Kaufmann, San Mateo
22. García F, García M, Melián B, Moreno J, Marcos J (2006) Solving feature subset selection problem by a parallel scatter search. Eur J Oper Res 169:477–489
23. García R, Benítez J, Saíz G (2008) Feature selection for time series forecasting: a case study. In: International conference on hybrid intelligent systems, pp 555–560
24. García S, Fernández A, Luengo J, Herrera F (2009) A study of statistical techniques and performance measures for genetics–based machine learning: accuracy and interpretability. Soft Comput 13(10):959–977
25. Garcíia S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inf Sci 180:2044–2064
26. García S, Herrera F (2008) An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. J Mach Learn Res 9:2677–2694
27. García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric test for analyzing the evolutionary algorithms' behaviour: a case study on the cec'2005 special session on real parameter optimization. J Heuristics 15(6):617–644
28. Ghost J, Deuser L, Beck S (1992) A neural network based hybrid system for detection characterization and classification of short-duration oceanic signals. IEEE J Ocean Eng 17(4):351–363
29. Goldberg D (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading
30. Goldberg D, Richardson J (1987) Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the second international conference on genetic algorithms, pp 41–49
31. Golub G, Van Loan C (1996) Matrix computations, 3rd edn. Johns Hopkins University Press, Baltimore
32. Guillén A, Rubio G, Toda I, Rivera A, Pomares H, Rojas I (2010) Applying multiobjective rbfnns optimization and feature selection to a mineral reduction problem. Expert Syst Appl 37(6):4050–4057
33. Gütlein M, Frank E, Hall M, Karwath A (1999) Large scale attribute selection using wrappers. In: Proceedings of the IEEE symposium on computational intelligence and data mining
34. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
35. Gwo-Fong L, Lu-Hsien C (2005) Time series forecasting by combining the radial basis function network and the self-organizing map. Hydrol Process 19:1925–1937
36. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten I (2009) The Weka data mining software: an update. SIGKDD Explorations 11(1)
37. Hall M, Smith L (1999) Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper. In: Twelfth international FLAIRS conference. AAAI Press/MIT Press, Menlo Park/Cambridge
38. Harpham C, Dawson C, Brown M (2004) A review of genetic algorithms applied to training radial basis function networks. Neural Comput Appl 13:193–201
39. Haykin S (1998) Neural networks: a comprehensive foundation, 2nd edn. Prentice Hall, New York
40. Holland J (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
41. Holm S (1979) A simple sequentially rejective multiple test procedure. Scand J Stat 6:65–70
42. Howard L, D'Angelo D (1995) The ga-p: a genetic algorithm and genetic programming hybrid. IEEE Intell Syst 10(3):11–15
43. Hsu W, Lee ML, Wang J (2007) Temporal and spatio-temporal data mining. IGI Publishing, Hershey
44. Jang J (1993) Anfis: adaptative-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 23(3):665–685
45. Khashei M, Reza Hejazi S, Bijari M (2008) A new hybrid artificial neural networks and fuzzy regression model for time series forecasting. Fuzzy Sets Syst 159(7):769–786
46. Lee C, Chiang Y, Shih C, Tsai C (2009) Noisy time series prediction using m-estimator based robust radial basis function neural networks with growing and pruning techniques. Expert Syst Appl 36(3):4717–4724
47. Lendasse A, de Bodt E, Wertz V, Verleysen M (2000) Non-linear financial time series forecasting—application to the bel 20 stock market index. Eur J Econ Soc Syst 14(1):81–91
48. Liang G, Xu W, He Y, Zhao Y (2008) Study and application of pso-rbfnn model to nonlinear time series forecasting for geotechnical engineering. Yantu Lixue (Rock Soil Mech) 29(4):995–1000
49. Lin Z, Zhang D, Gao L, Kong Z (2008) Using an adaptive self-tuning approach to forecast power loads. Neurocomputing 71(4–6):559–563
50. Liu J, Kwong R (2007) Automatic extraction and identification of chart patterns towards financial forecast. Appl Soft Comput 4:1197–1208
51. Maimon OLR (2005) The data mining and knowledge discovery handbook. Springer, Berlin
52. Maimon O, Last M (2000) Knowledge discovery and data mining—the info-fuzzy network (ifn) methodology. Kluwer Academic, Dordrecht
53. Mandani E, Assilian S (1975) An experiment in linguistic synthesis with a fuzzy logic controller. Int J Man-Mach Stud 7(1):1–13

Journal ID: 10489, Article ID: 284, Date: 2011-03-02, Proof No: 1, UNCORRECTED PROOF

54. Meng K, Dong Z, Wong K (2009) Self-adaptive radial basis function neural network for short-term electricity price forecasting. IET Gener Transm Distrib 4:325–335
55. Mochón A, Quintana D, Sáez Y, Isasi P (2008) Soft computing techniques applied to finance. Appl Intell 29(2):111–115
56. Moller F (1990) A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw 6:525–533
57. Moshiri S, Cameron N, Scuse D (1999) Static dynamic, and hybrid neural networks in forecasting inflation. Comput Econ 14(3):219–235
58. Murphy J (1999) Technical analysis of the financial markets: a comprehensive guide to trading methods and applications. New York Institute of Finance, New York
59. Nekoukar V, Beheshti MH (2009) A local linear radial basis function neural network for financial time-series forecasting. Appl Intell 33(3):352–356
60. Park J, Sandberg I (1993) Universal approximation and radial basis function network. Neural Comput 5(2):305–316
61. Pedrycz W (1998) Conditional fuzzy clustering in the design of radial basis function neural networks. IEEE Trans Neural Netw 9(4):601–612
62. Pérez-Godoy M, Pérez P, Rivera A, del Jesus M, Carmona C, Frías M, Parras M (2010) CO^2RBFN for short-term forecasting of the extra virgin olive oil price in the Spanish market. Int J Hybrid Intell Syst 7(1):75–87
63. Pérez-Godoy M, Rivera A, del Jesus M, Berlanga F (2010) CO^2RBFN: an evolutionary cooperative-competitive RBFN design algorithm for classification problems. Soft Comput 14(9):953–971
64. Pino R, Parreno J, Gomez A, Priore P (2008) Forecasting next-day price of electricity in the Spanish energy market using artificial neural networks. Eng Appl Artif Intell 21(1):53–62
65. Potter M, De Jong K (2000) Cooperative coevolution: an architecture for evolving coadapted subcomponents. Evol Comput 8(1):1–29
66. Press W, Flannery B, Teukolski S, Vetterling W (1988) Numerical recipes in C. Cambridge University Press, Cambridge
67. Rivas V, Merelo J, Castillo P, Arenas M, Castellano J (2004) Evolving rbf neural networks for time-series forecasting with evrbf. Inf Sci 165:207–220
68. Rivera A, Rojas I, Ortega J, del Jesus M (2007) A new hybrid methodology for cooperative-coevolutionary optimization of radial basis function networks. Soft Comput 11(7):655–668
69. Roddick J, Spiliopoulou M (1999) A bibliography of temporal spatial and spatio-temporal data mining research. ACM SIGKDD Explor Newsl 1(1):34–38
70. Sánchez L, Couso I (2000) Fuzzy random variables-based modeling with ga-p algorithms. In: Bouchon B, Yager RR, Zadeh L (Eds.) Information, uncertainty and fusion, pp 245–256
71. Sheskin D (2006) Handbook of parametric and nonparametric statistical procedures, 2nd edn. Chapman & Hall/CRC Press, London/Boca Raton
72. Sheta A, De Jong K (2001) Time-series forecasting using ga-tuned radial basis functions. Inf Sci 133:221–228
73. Tsay RS (2010) Analysis of financial time series, 3rd edn (2010)
74. Ture M, Kurt I (2006) Comparison of four different time series methods to forecast hepatitis A virus infection. Expert Syst Appl 31(1):41–46
75. Vanstone B, Finnie G (2009) An empirical methodology for developing stockmarket trading systems using artificial neural networks. Expert Syst Appl 36(3, Part 2), 6668–6680
76. Vanstone B, Hahn T (2010) Designing stock market trading systems: with and without soft computing. Harriman House, Petersfield
77. Versace M, Bhatt R, Hinds O, Shiffer M (2004) Predicting the exchange traded fund dia with a combination of genetic algorithms and neural networks. Expert Syst Appl 27(3):417–425
78. Whitehead B, Choate T (1996) Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction. IEEE Trans Neural Netw 7(4):869–880
79. Widrow B, Lehr M (1990) 30 years of adaptive neural networks: perceptron, madaline and backpropagation. Proc IEEE 78(9):1415–1442
80. Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics 1:80–83
81. Xiong Q, Yong S, Shi W, Chen J, Liang Y (2005) The research of forecasting model based on rbf neural network. In: Proceedings of the 2005 international conference on neural networks and brain, pp 1032–1035
82. Yun Z, Quan Z, Caixin S, Shaolan L, Yuming L, Yang S (2008) Rbf neural network and anfis-based short-term load forecasting approach in real-time price environment. IEEE Trans Power Syst 23(3):853–858
83. Zar J (1999) Biostatistical analysis. Prentice Hall, Upper Saddle River