Neurocomputing 73 (2010) 2030–2038

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

New method for instance or prototype selection using mutual information in time series prediction

A. Guillen [a,*], L.J. Herrera [a], G. Rubio [a], H. Pomares [a], A. Lendasse [b], I. Rojas [a]

[a] University of Granada, Department of Computer Technology and Architecture, Spain
[b] Helsinki University of Technology, Information and Computer Science Department, Finland

ARTICLE INFO

Available online 7 March 2010

Keywords:
Time series
Regression
Prediction
Mutual information
Prototype
Instance
Selection

ABSTRACT

The problem of selecting the patterns to be learned by any model is usually not considered by the time of designing the concrete model but as a preprocessing step. Information theory provides a robust theoretical framework for performing input variable selection thanks to the concept of mutual information. Recently the computation of the mutual information for regression tasks has been proposed so this paper presents a new application of the concept of mutual information not to select the variables but to decide which prototypes should belong to the training data set in regression problems. The proposed methodology consists in deciding if a prototype should belong to or not to the training set using as criteria the estimation of the mutual information between the variables. The novelty of the approach is to focus in prototype selection for regression problems instead of classification as the majority of the literature deals only with the last one. Other element that distinguishes this work from others is that it is not proposed as an outlier detector but as an algorithm that determines the best subset of input vectors by the time of building a model to approximate it. As the experiment section shows, this new method is able to identify a high percentage of the real data set when it is applied to highly distorted data sets.

1. Introduction

The task of selecting an adequate subset of input vectors that are included in a training set when classifying, approximating or predicting an output is a relevant task that, if accomplished correctly, can provide storage and computational savings and improve the accuracy of the results. This problem can be tackled from different perspectives: outlier detection or instance selection of noise-free data. The concept of outlier was firstly introduced by Grubbs in [1] as: "…an outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs…". This concept was later extended by Barnett and Lewis [2] as: "…An observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data…". Many other notations (novelty detection, anomaly detection, noise detection, deviation detection, exception mining) have been used for this problem [3].

Instance selection of noise-free data aims at determining a subset of the initial training data set in such a way that the accuracy of the model is not decreased but the computational cost and storage requirements are diminished. This approach is usually applied to large databases where the amount of data is large and noise in the instance is not present. These situations are specially important in biomedical applications and data mining where the amount of noise-free data is enormous [4–6].

The work presented in this paper fits in the definition of outlier. This definition does not bound the way in which an outlier differs from the other elements, thus, an outlier could be considered as a different element in the input space (one dimension, a group of dimensions) or the output space. However, in classification problems, since the classes are predefined beforehand, the outlier is in the input vector space. Because the term outlier is mostly used in problems related to classification and the set of input vectors in regression problems has been denoted as instances or prototypes, this paper will use this last notation.

To sum up, the problem tackled in this paper can be formulated as: given a set of input/output pairs (prototypes or instances with continuous output), it is desired to identify the elements that could be erroneous.

The majority of the research in outlier/instance/prototype selection has been focused in classification problems [7,8], although few works aimed at solving problems for continuous output. For example, in [9] is presented a method to select the input vectors when using the k-NN algorithm as model, thus, this methodology does not permit the selection of the input vectors

* Corresponding author.
E-mail address: aguillen@atc.ugr.es (A. Guillen).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2009.11.031

before designing more complex models such as neural networks. In [10], a genetic algorithm is used to select both the feature and the input subsets, however, it is only suitable for linear regression models.

Three main approaches have been used in order to optimize the set of inputs that the training algorithm will use: *incremental*, *decremental* and *batch*. The incremental approach starts from an empty set of input vectors and defines the training set adding input vectors iteratively [11]. The opposite perspective is taken in the *decremental* approach that starts considering all the input vectors available and, following a prefixed criteria, proceeds to remove the non-desired instances [12]. There are other techniques such as the batch method that iterates several times before deleting the instance from the training set, setting a flag on the instances that could be removed in next iterations [13]. Recently, evolutionary algorithms [7], boosting-based algorithms [14], and pruning techniques [15] have also been applied to this problem.

The work developed in this paper is framed within the decremental approach since it considers the whole data set at the beginning. The criteria to remove the input vectors has been taken from the method used to perform feature selection. The problem of feature selection consists in finding an adequate subset of variables in such a way that it is possible to train more accurate models. If the set of input data has redundant or irrelevant data, the training can result in an overfitted model with poor generalization capabilities [16,17]. Furthermore, if the dimensionality is not reduced, some local approximator models suffer from the curse of dimensionality, making it impossible to design accurate models [18].

To tackle the feature selection problem, two main streams have been followed: *filter* and *wrapper* methods. The filter approach consists of a preprocessing of the input data so the model is built afterwards [19]. The wrapper approach attempts to design the model at the same time that performs the variable selection [20]. The concepts of entropy and mutual information (MI) make the Information theory an interesting framework for filtering approaches. The MI will be used in order to decide which input vectors should be included or not in the training data set. This new approach is possible thanks to the possibility of computing the MI for continuous variables. Thus, it will be possible to identify outliers and noise input vectors from the data set, providing as output a subset of input vectors which are adequate to be used to design the regression model.

The rest of the paper is organized as follows: Section 2 presents the formulation of MI and describes the method used to compute it, then, Section 3 introduces the new methodology to select the input vectors. Section 4 includes a variety of experiments to verify the proposed approach and, finally, Section 5 draws the conclusions.

2. Prototype selection based on mutual information

This section firstly describes the concept of mutual information, then, the algorithm to perform the prototype selection is introduced.

2.1. Mutual information

Given a single-output multiple input function approximation or classification problem, with input variables $X = [x_1, x_2, \ldots, x_d]$ and output variable $Y = y$, the main goal of a modeling problem is to reduce the uncertainty on the dependent variable Y. According to the formulation of Shannon, and in the continuous case, the uncertainty on Y is given by its entropy defined as

$$H(Y) = -\int \mu_Y(y)\log \mu_Y(y)\, dy, \tag{1}$$

considering that the marginal density function $\mu_Y(y)$ can be defined using the joint probability density function $\mu_{X,Y}$ of X and Y as

$$\mu_Y(y) = \int \mu_{X,Y}(x,y)\, dx. \tag{2}$$

Given that we know X, the resulting uncertainty of Y conditioned to known X is given by the conditional entropy, defined by

$$H(Y,X) = -\int \mu_X(x) \int \mu_Y(y|X=x)\log \mu_Y(y|X=x)\, dy\, dx. \tag{3}$$

The joint uncertainty on the $[X,Y]$ pair is given by the joint entropy, defined by

$$H(X,Y) = -\int \mu_{X,Y}(x,y)\log \mu_{X,Y}(x,y)\, dx\, dy. \tag{4}$$

The mutual information (also called cross-entropy) between X and Y can be defined as the amount of information that the group of variables X provide about Y, and can be expressed as $I(X,Y) = H(Y) - H(Y|X)$. In other words, the mutual information $I(X,Y)$ is the decrease of the uncertainty on Y once we know X. Due to the mutual information and entropy properties, the mutual information can also be defined as

$$I(X,Y) = H(X) + H(Y) - H(X|Y), \tag{5}$$

leading to

$$I(X,Y) = \int \mu_{X,Y}(x,y)\log \frac{\mu_{X,Y}(x,y)}{\mu_X(x)\mu_Y(y)}\, dx\, dy. \tag{6}$$

Thus, only the estimate of the joint PDF between X and Y is needed to estimate the mutual information between two groups of variables.

Estimating the joint probability distribution can be performed using a number of techniques. As mentioned already, histograms and kernel density estimators have been used for this purpose [21]. Although there exists a variety of algorithms to calculate the mutual information between variables, this paper uses the approach presented in [22] which is based on the k-nearest neighbors.

2.2. Estimating the mutual information using the k-nearest neighbors

There is extensive literature about estimators based on the k-nearest neighbors for the entropy, but it has been only recently extended to the MI [23].

We define the space $Z = \{X,Y\}$ and we will use the maximum norm for any pair of points $z = (x,y)$ and $z' = (x',y')$,

$$\|z - z'\| = \max\{\|x - x'\|, \|y - y'\|\}, \tag{7}$$

although any other norm could be used. Denote by $\varepsilon(i)$ the distance from a point z_i to its k-th nearest neighbor and by $\varepsilon_x(i)$ and $\varepsilon_y(i)$ the distances between the same points projected into the X and Y subspaces. Obviously $\varepsilon(i) = \max\{\varepsilon_x(i), \varepsilon_y(i)\}$.

We will count the number $n_x(i)$ of points x_j whose distance from x_i is strictly less than $\varepsilon(i)$, and similarly for y instead of x. The estimate for MI is then (see [23] for a proof of the convergence of this estimator)

$$\hat{I}_1(X,Y) = \psi(k) - \frac{1}{N}\sum_{i=1}^{N}[\psi(n_x(i)+1) + \psi(n_y(i)+1)] + \psi(N), \tag{8}$$

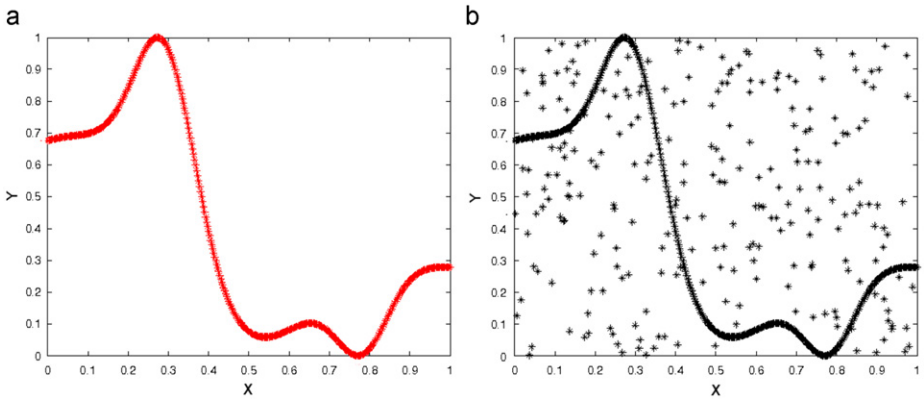


Fig. 1. (a) Original target function and (b) distorted data set.

where ψ is the digamma function given by

$$\psi(t)=\frac{\Gamma'(t)}{\Gamma(t)}=\frac{d}{dt}\ln\Gamma(t) \tag{9}$$

with

$$\Gamma(t)=\int_0^\infty u^{t-1}e^{-u}\,du. \tag{10}$$

Function ψ satisfies the recursion equation $\psi(x+1)=\psi(t)+1/x$ and $\psi(1)=C$ where $C=-0.5772156\ldots$ is the Euler–Mascheroni constant.

Another alternative is to replace $n_x(i)$ and $n_y(i)$ by the number of points with $\|x_i-x_j\|\le\varepsilon_x(i)/2$ and $\|y_i-y_j\|\le\varepsilon_y(i)/2$. The estimate for MI is then

$$\hat{I}_2(X,Y)=\psi(k)-\frac{1}{k}-\frac{1}{N}\sum_{i=1}^{N}[\psi(n_x(i))+\psi(n_y(i))]+\psi(N). \tag{11}$$

In this paper this second estimator is used, which is the one implemented in [24]. See [23] for an extended explanation.

As can be noted, this MI estimator has a dependency on the value chosen for k (k-th nearest neighbor). As it is recommended in [25] for a tradeoff between variance and bias, in the examples, a mid-range value for k ($k=6$) will be used.

2.3. Prototype selection using mutual information

The idea that motivates this paper is: since the MI is able to let us know how much information from the output can be retrieved using the different variables starting from a set of input vectors (prototypes), it would be possible that if a significant prototype is removed from the set of input vectors, the amount of MI that could be retrieved would be decreased. On the other hand, if an insignificant prototype is deleted from the original set, the amount of MI should not be decreased. These two sentences are correct, however, there are situations where they might not be completely true. For example, if there are outliers, they will probably provide a significant amount of MI but they should not be considered. On the other hand, if the output of the system remains constant, the amount of information will not fluctuate if similar prototypes are removed.

Thus, in order to make an objective evaluation of how relevant an input vector is, it is necessary to consider the loss of MI with respect to its neighbors in such a way that, if the loss of MI is similar to the prototypes near $\mathbf{x}_i$, this input vector must be included in the filtered data set. The algorithm proposed to calculate the reduced set of prototypes is described in Algorithm 1

Table 1
Parameters for the function f_1.

RBF centers	Radii	Weights
0.484	0.237	−0.387
0.151	0.530	0.284
0.781	0.091	−0.208
0.100	0.405	0.103
0.294	0.104	0.464

Algorithm 1. MI prototype selection.

1. Calculate the K nearest neighbors in the input space of $\mathbf{x}_i=(x_{i\,1},x_{i\,2},\ldots,x_{id})$ for $i=1\ldots n$ ($nn\,[\mathbf{x}_i,j]$ for $j=1\ldots K$)
2. **for** i=1…n
   Calculate the mutual information value $I(X,Y)_i$ when removing $\mathbf{x}_i$ from X
   end
3. Normalize $I(X,Y)_i$ in [0,1]
4. **for** i=1…n
   $Cdiff=0$
   for $j=1\ldots K$
   $diff=I(X,Y)_i-I(X,Y)_{nn[\mathbf{x}_i,j]}$
   if $diff>\alpha$
   $Cdiff=Cdiff+1$
   end
   if $Cdiff<K$
   Select prototype
   else
   Discard prototype
   end
   end

where d is the number of dimensions, n the number of input vectors, α is a predefined threshold that determines how different the MI should be with respect to the neighbors and K is the number of neighbors to be considered in the comparisons. The value of α must be set manually and determines the sensitivity or the specificity of the algorithm. As the experiments have shown, a value of 0.05 could be a good compromise between those two objectives.

When calculating how much of MI was lost, two approaches could be taken: (1) to calculate the MI between the complete set of variables and the output or (2) to compute the MI between each variable and the output. With the MI estimator used in the

experiments, no difference between those two approaches could be seen, however, other implementations should be analyzed.

Table 2
Confusion matrix for the one dimensional synthetic function.

Predicted	Actual		Total
	Positive	Negative	
Positive	391	55	446
Negative	9	195	
Total	400	250	

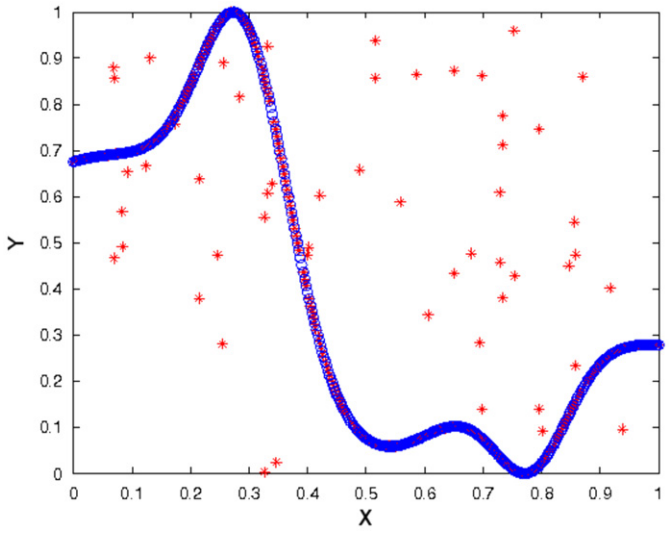


Fig. 2. Filtered data (stars) and original data (circles).

Table 3
Approximation errors (normalized root mean squared error, NRMSE) obtained when training the networks using the different data sets.

Data set	Train	Test
Original	0.027	0.027
Distorted	0.773	0.432
Selected	0.451	0.188

Table 4
Sensitivity (true positive rate, TPR) and false positive rate (FPR) of the algorithm using several values of α and K for function f_1.

α	$K=1$	$K=3$	$K=5$
Sensitivity			
0.005	0.8603	0.9751	1.0
0.015	0.8953	0.9850	1.0
0.025	0.9377	0.9900	1.0
0.035	0.9576	0.9975	1.0
0.045	0.9651	0.9975	1.0
0.055	0.9751	1.0	1.0
0.065	0.9800	1.0	1.0
0.075	0.9850	1.0	1.0
0.085	0.9850	1.0	1.0
0.095	0.9875	1.0	1.0
FPR			
0.005	0.1520	0.4480	0.6160
0.015	0.1560	0.4800	0.6640
0.025	0.1600	0.5000	0.6880
0.035	0.1760	0.5320	0.7240
0.045	0.1880	0.5680	0.7600
0.055	0.1960	0.5920	0.7880
0.065	0.2040	0.6160	0.8040
0.075	0.2120	0.6400	0.8240
0.085	0.2160	0.6680	0.8520
0.095	0.2200	0.6800	0.8560

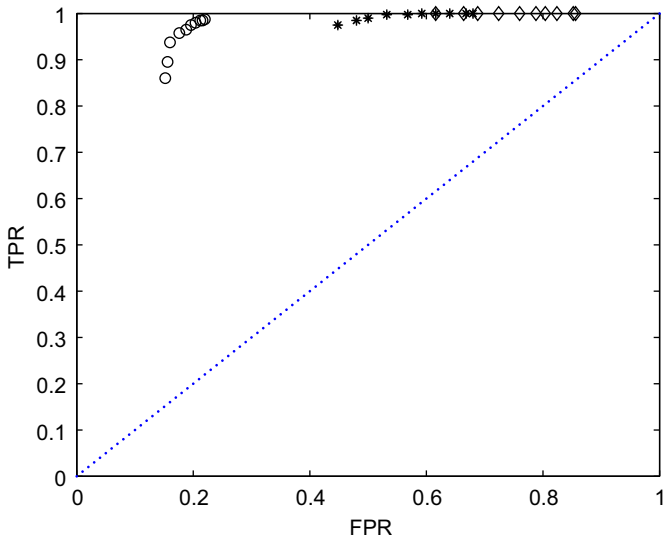


Fig. 4. ROC curve for the function f_1. Circles were computed using $K=1$, stars with $K=2$ and diamonds with $K=3$.

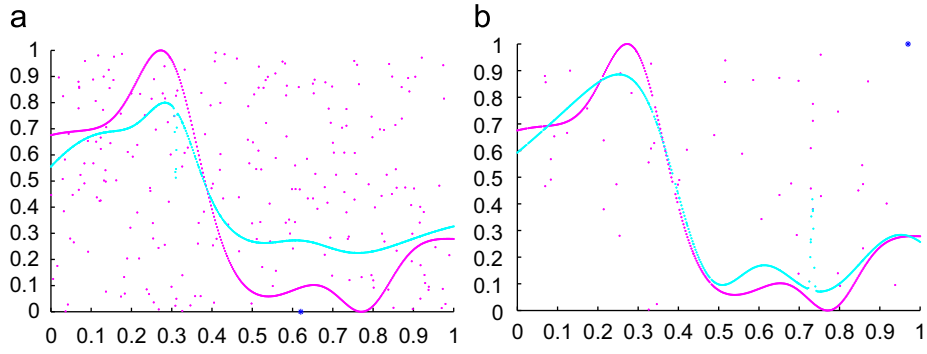


Fig. 3. Approximation made by the RBFNN trained with the data before prototype selection (a) and after the selection (b).

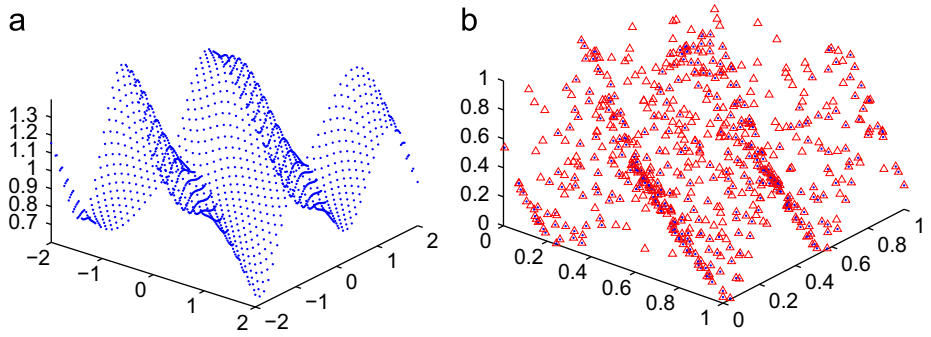


Fig. 5. (a) Original target function and (b) distorted data set (diamonds) with the original training data (dots).

Table 5
Confusion matrix for the synthetic function f_4.

Predicted	Actual		Total
	Positive	Negative	
Positive	302	106	408
Negative	98	94	
Total	400	200	

Table 6
Sensitivity (true positive rate, TPR) and false positive rate (FPR) of the algorithm using several values of α and K for function f_4.

α	$K=1$	$K=3$	$K=5$
Sensitivity			
0.005	0.611	0.844	0.917
0.015	0.652	0.877	0.930
0.025	0.675	0.890	0.932
0.035	0.702	0.905	0.940
0.045	0.715	0.915	0.950
0.055	0.747	0.927	0.962
0.065	0.777	0.937	0.972
0.075	0.792	0.945	0.977
0.085	0.807	0.947	0.982
0.095	0.827	0.957	0.985
FPR			
0.005	0.400	0.590	0.690
0.015	0.425	0.625	0.720
0.025	0.445	0.645	0.730
0.035	0.465	0.670	0.750
0.045	0.490	0.695	0.775
0.055	0.520	0.715	0.800
0.065	0.540	0.735	0.810
0.075	0.555	0.750	0.825
0.085	0.575	0.770	0.850
0.095	0.605	0.810	0.870

3. Experiments

This section presents the results of applying the new algorithm to highly distorted data sets, some artificially generated and some taken from real life problems.

3.1. Experiment 1: one dimensional synthetic function

The data set was generated synthetically so it was possible to know exactly which input vectors were the originals and which the noisy ones. The target was a one dimensional function (Fig. 1(a)) that was generated using a Gaussian radial basis function neural network (RBFNN) with randomly chosen parameters, which are shown in Table 1.

Table 7
Approximation errors (normalized root mean squared error) obtained after training the networks (ICFA and OVI algorithm) using the different data sets.

Time series	Error (original)	Error (outliers)	Error (filtered)
ICFA			
Logistic	0.0001	0.136	0.027
Henon	0.009	0.158	0.044
Electric	0.281	0.707	0.416
MCglass	0.031	0.177	0.138
OVI			
Logistic	0.021	0.143	0.032
Henon	0.037	0.152	0.042
Electric	0.342	0.773	0.449
MCglass	0.024	0.175	0.142

The test data set is the original data without noise.

The original data set consists of 400 equally distributed prototypes and their corresponding output. This data set was modified adding a set of 250×2 (X, Y) random values in [0,1] from a uniform distribution, obtaining a new data set of 650 prototypes of dimension 1 with one output. This data set is represented in Fig. 1(b).

The proposed method was applied using the value 0.05 for the threshold α and 1 for K, obtaining a filtered data set of size 446. The results of the classification are shown in Table 2. The algorithm discriminated the 78.2% of the incorrect prototypes and identified correctly the 97.7% of the original prototypes. Fig. 2 depicts the results, where the circles represent the original prototypes and the stars represent the prototypes selected from the distorted data set. If a star is included in a circle, it means that the original prototype was chosen correctly.

To evaluate the utility and effectiveness of the proposed approach, several RBFNNs were designed using the three different data sets: original, distorted and filtered. Each network was trained using these three data sets and the test set was generated using the original function and it consisted of 1000 noise-free homogeneously distributed points. The methodology to design the RBFNN was: first, initialize the centers with the algorithm ICFA (Improved Clustering for Function Approximation) proposed in [26], then apply k-NN to get a first value for the radii and then, apply a local search to make a fine tuning of these parameters. As it was expected, thanks to the prototype selection, the approximation errors (Table 3) that can be obtained are much smaller than if no prototype selection was made. Fig. 3 shows the approximations of the original function by the RBFNNs generated using the distorted data and using the data after the prototype

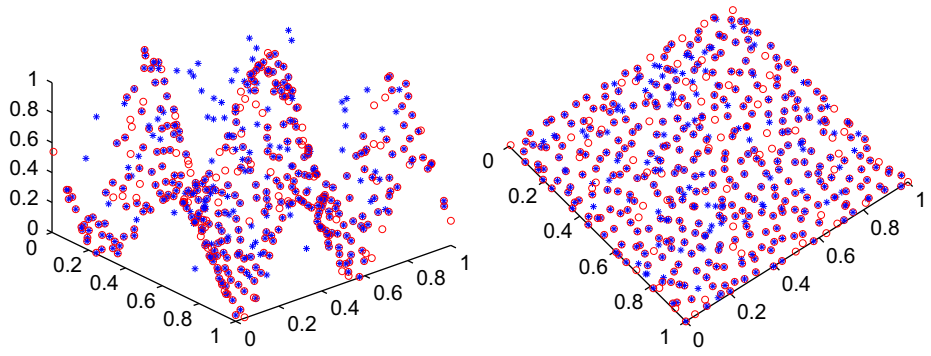


Fig. 6. Filtered data (stars) and original data (circles).

selection. The error is measured using the normalized root mean squared error (NRMSE) which is a common measure to evaluate the quality of the approximations [27].

Regarding the effect of the K and α parameters, Table 4 shows the sensitivity (true positive rate, $TPR = TP/(TP+FN)$) and the false positive rate (FPR) (specificity $= 1-FPR$) of the algorithm when modifying these parameters. These values are represented as a ROC curve in Fig. 4. As the results show, the smaller the K and α are, the more restrictive the algorithm becomes, allowing to identify a significant number of noisy points although discarding many original points. Nonetheless, the behavior of the algorithm shows to be quite adequate, performing a good prototype selection.

3.2. Experiment 2: two dimensional synthetic function

Secondly, a two dimensional synthetic function f_4 (in order to keep the original name) represented in Fig. 5(a) and defined in Eq. (12) was used. The function f_4 was presented in [28] and it has been used as a benchmark in [29–31] due to its high variable output. First, 400 input vectors were generated using f_4, then, 200 input vectors were generated using random values in [0,1] from a uniform distribution, creating the remaining complete data set as it is depicted in Fig. 5(b):

$$f_4(x_1,x_2) = 1.9[1.35+e^{x_1}\sin(13(x_1-0.6)^2)e^{-x_2}\sin(7x_2)], \quad x_1,x_2 \in [0,1]. \tag{12}$$

Table 5 shows the confusion matrix of an execution for function f_4. The results are not as impressive as in the previous one but the function presents a high variability in the output and the sampling is not very high. Furthermore, the amount of noisy points introduced is really high so the results could be considered satisfactory. Table 6 shows the sensitivity of FPR of the algorithm and Fig. 7 depicts these data.

3.3. Experiment 3: time series prediction

In order to evaluate the proposed approach in time series prediction, new experiments have been developed. The data sets used are

(1) *Logistic*: The Logistic map is a demographic model that has been popularized by [32] as an example of simple nonlinear system that exhibits complex, chaotic behavior. It is drawn from Eq. (13).

$$y(t) = 4y_{t-1}(1-y_{t-1}). \tag{13}$$

(2) *Hénon*: The Hénon map is one of the most studied dynamical systems. The canonical Hénon map takes points in the plane following [33]:

$$x_{n+1} = y_n + 1 - 1.4x_n^2,$$

$$y_{n+1} = 0.3x_n. \tag{14}$$

(3) *Electric*: Daily electric load data in California[1] (the original load data are sampled every hour).

(4) *Mackey–Glass*: The Mackey–Glass time series is approximated from the differential equation (15) [34]. It is a widely used benchmark for generalization abilities of time series prediction methods. The series is continuous and it is obtained by integrating Eq. (15) with a numerical integration method such as the fourth order Runge–Kutta method. The data for this time series were obtained from the `mgdata.dat` file included in the *Fuzzy Logic Toolbox*[2] from the Matlab software.

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-17)} - 0.1x(t). \tag{15}$$

A new approach to test the methodology must be taken. The reason is because in time series prediction the definition of the set of input vectors X is made using the output variable Y and the regressors. Thus, adding random input vectors with random outputs, as it was done in the previous experiments, would not respect the nature of the problem. Hence, the data sets were distorted in the following way:

(1) adding noise to the original normalized output: for each point, a random value was added or subtracted;
(2) adding outliers: pure random values were introduced in random positions within the series.

Therefore, the data sets were highly distorted. In order to evaluate the utility of the proposed approach, it is not possible to find out which input vectors are pure outliers because, depending on the regressors, an outlier can be included in several input vectors. Then, to be able to evaluate the quality of the proposed data filtering method, the following steps are followed:

(1) Generate a model that predicts the original "clean" time series.
(2) Generate a model that predicts the distorted time series.
(3) Generate a model that predicts the filtered time series.

A priori, it could seem trivial to guess that the model approximating the filtered data will perform better than the model approximating the distorted data set as the outliers have

[1] Available at http://www.ucei.berkeley.edu/CSEM/datamine/ISOdata/
[2] See http://www.mathworks.com/products/fuzzylogic/?BB=1

been removed. So, in order to analyze the quality of the models, the original time series were given to them as test data. The results are shown in Table 7 where a radial basis function neural network was designed using the deterministic methodology proposed in [35] with five Gaussian neurons. The value of the two parameters α and K were chosen after performing several runs. Each time series was distorted by adding 50 random outliers. Fig. 6 shows the results provided so, figures for the first two time series represent the input vectors and the output since the dimension of the input vectors is one and two. The other two figures represent just the value of the outputs for the sake of clarity. As these figures show, for the four time series the results are quite satisfactory since the number of outliers identified is high, this is reflected in smaller approximation errors when the original output is given to the trained models. For the case of the first two time series, the approximation errors are very close to the ones obtained with the original output. For the other two, due to the high dimensionality of the problem, the approximation errors differ a bit more although it is quite clear that, after the filtering, the approximation obtained is better. In order to perform a fair comparison for the model, other algorithm (output-value based initializer, OVI) presented in [36] was used to train the model. The approximation errors obtained with this algorithm are higher with the original data although the networks generated are able to generalize increasing the performance when the filtered data set was given as input (Figs. 7 and 8).

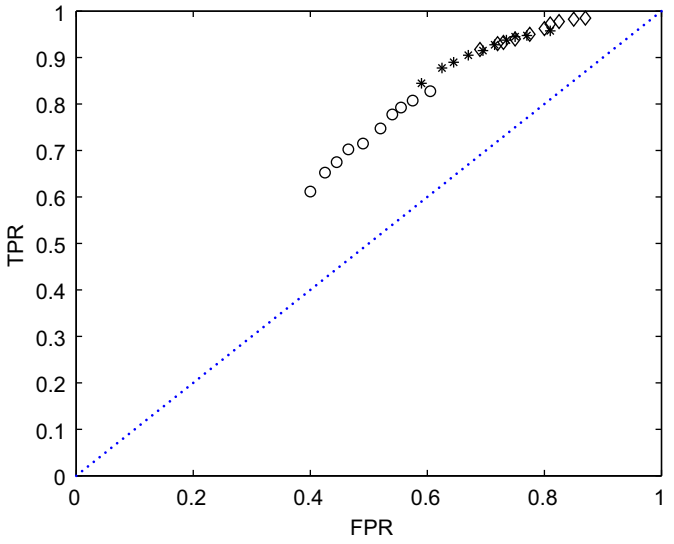


Fig. 7. ROC curve for the function f_4. Circles were computed using $K=1$, stars with $K=2$ and diamonds with $K=3$.

4. Conclusions and further work

This paper has presented a possible approach to solve the problem of selecting adequate data samples before using any model to approximate a function. This new method is based on the concept of mutual information which was used before for feature selection. The main difference between the already existing approaches and the proposed one is that is oriented to data sets with a continuous output value instead of a predefined set of labels and with the global perspective that the MI provides of the complete data set. As the experiments have shown, the method seems quite effective selecting the correct prototypes with a high accuracy. Although the algorithm performance is quite robust, further work could consider a comparison among other different ways of calculating the mutual information and the parallelization of the algorithm.

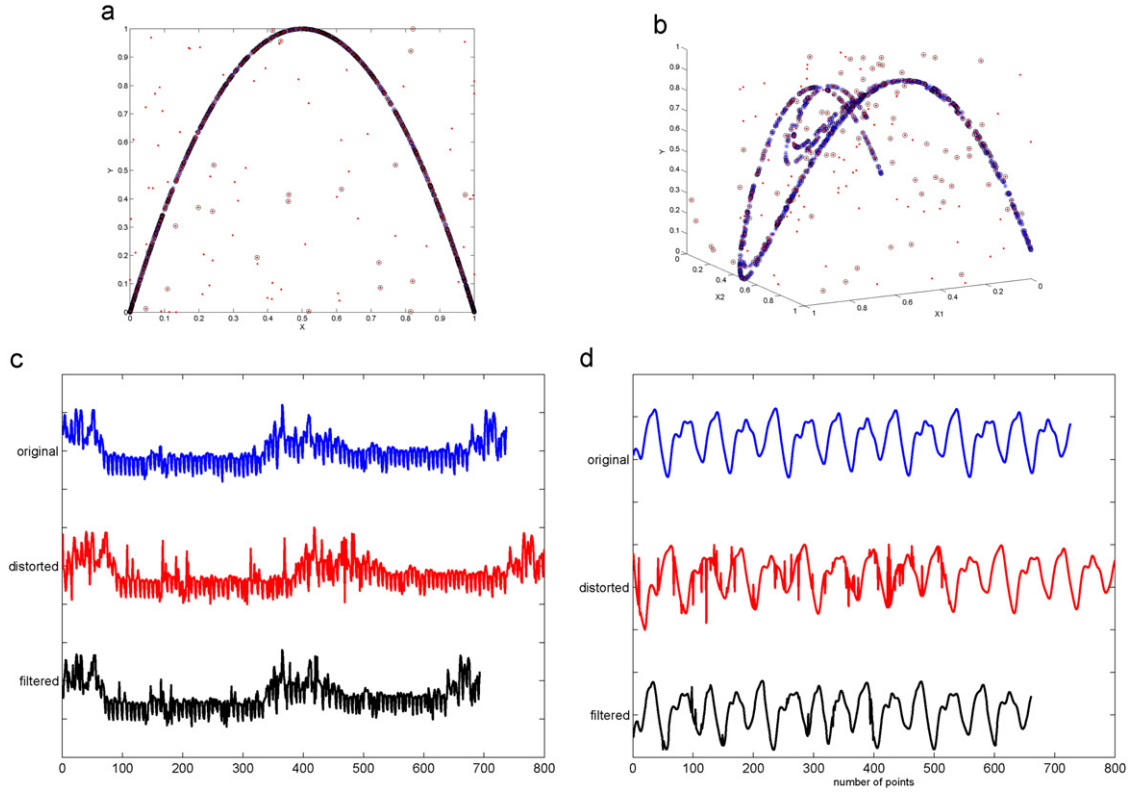


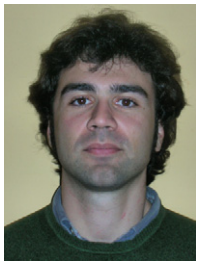
Fig. 8. Original input vectors and the outputs for the original data set (stars), outliers (dots) and filtered data set (circles) for time series one (a) and two (b). Plotted output for time series three and four representing the original output (top), the distorted output (middle) the filtered output (bottom).

Acknowledgements

This work has been partially supported by the Spanish CICYT Project TIN2007-60587 and Junta Andalucia Project P07-TIC-02768.

References

[1] F.E. Grubbs, Procedures for detecting outlying observations in samples. Technometrics 11 (1969) 1–21.
[2] V. Barnett, T. Lewis, Outliers in Statistical Data, Wiley Series in Probability & Statistics, Wiley, April 1994.
[3] V. Hodge, J. Austin, A survey of outlier detection methodologies, Artificial Intelligence Review 22 (2) (2004) 85–126.
[4] J.R. Cano, F. Herrera, M. Lozano, S. García, Making cn2-sd subgroup discovery algorithm scalable to large size data sets using instance selection, Expert Systems with Applications 35 (4) (2008) 1949–1965.
[5] A. Coulet, M. Smail-Tabbone, P. Benlian, A. Napoli, M.-D. Devignes, Ontology-guided data preparation for discovering genotype–phenotype relationships, BMC Bioinformatics 9 (Suppl. 4) (2008) S3.
[6] H. Knublauch, R.W. Fergerson, N.F. Noy, M.A. Musen, The protégé owl plugin: an open development environment for semantic web applications, in: The Semantic Web—ISWC 2004, 2004, pp. 229–243.
[7] H. Ishibuchi, T. Nakashima, M. Nii, Learning of neural networks with Ga-based instance selection, in: IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th, vol. 4, 25–28 July 2001, pp. 2102–2107.
[8] E. Pekalska, R.P.W. Duin, P. Paclík, Prototype selection for dissimilarity-based classifiers, Pattern Recognition 39 (2) (2006) 189–208 (Part Special Issue: Complexity Reduction).
[9] J. Zhang, Y. Yim, J. Yang, Intelligent selection of instances for prediction functions in lazy learning algorithms, Artificial Intelligence Review 11 (1997) 175–191.
[10] J. Tolvi, Genetic algorithms for outlier detection and variable selection in linear regression models, Soft Computing 8 (8) (2004) 527–533.
[11] D.W. Aha, Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms, International Journal of Man-Machine Studies 36 (2) (1992) 267–287.
[12] D.R. Wilson, T. Martinez, Reduction techniques for instance based learning algorithms, Machine Learning 38 (3) (2000) 257–286.
[13] I. Tomek, An experiment with edited nearest neighbor rule, IEEE Transactions on Systems, Man and Cybernetics 6 (1976) 448–452.
[14] M. Sebban, R. Nock, S. Lallich, Stopping criterion for boosting-based data reduction techniques: from binary to multiclass problems, Journal of Machine Learning Research 3 (2002) 863–865.
[15] V.B. Zubek, T.G. Dietterich, Pruning improves heuristic search for cost-sensitive learning, in: Proceedings of the International Conference on Machine Learning, 2002, pp. 27–34.
[16] E. Liitiäinen, F. Corona, A. Lendasse, Non-parametric residual variance estimation in supervised learning, in: IWANN 2007, Lecture Notes in Computer Science, Springer-Verlag, June 20–22, p. 63.
[17] E. Eirola, E. Liitiäinen, A. Lendasse, F. Corona, M. Verleysen, Using the delta test for variable selection, in: European Symposium on Artificial Neural Networks, Bruges, Belgium, April 2008, pp. 25–30.
[18] L.J. Herrera, H. Pomares, I. Rojas, A. Guillén, A. Prieto, O. Valenzuela, Recursive prediction for long term time series forecasting using advanced models, Neurocomputing 70 (16–18) (2007) 2870–2880.
[19] A. Guillén, A. Sorjamaa, Y. Miche, A. Lendasse, I. Rojas, Efficient parallel feature selection for steganography problems, in: J. Cabestany, F. Sandoval, A. Prieto, J.M. Corchado (Eds.), IWANN (1), Lecture Notes in Computer Science, vol. 5517, Springer, 2009, pp. 1224–1231.
[20] A. Guillén, H. Pomares, J. González, I. Rojas, O. Valenzuela, B. Prieto, Parallel multi-objective memetic RBFNNs design and feature selection for function approximation problems, Neurocomputing 72 (16-18) (2009) 3541–3555.
[21] B.V. Bonnlander, A.S. Weigend, Selecting input variables using mutual information and nonparametric density estimation, in: Proceedings of the ISANN, pp. 42–50.
[22] L.J. Herrera, H. Pomares, I. Rojas, M. Verleysen, A. Guillén, Effective Input Variable Selection for Function Approximation, in: Lecture Notes in Computer Science, vol. 4131, 2006, pp. 41–50.
[23] A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information, Physics Review June (2004) 066138.
[24] 〈http://www.klab.caltech.edu/~kraskov/MILCA/〉.
[25] H. Stögbauer, A. Kraskov, S.A. Astakhov, P. Grassberger, Least dependent component analysis based on mutual information. Physics Review December (2004) 066123.
[26] A. Guillén, J. González, I. Rojas, H. Pomares, L.J. Herrera, O. Valenzuela, A. Prieto, Using fuzzy logic to improve a clustering technique for function approximation, Neurocomputing 70 (16–18) (2007) 2853–2860.
[27] A. Guillén, H. Pomares, I. Rojas, J. González, L.J. Herrera, F. Rojas, O. Valenzuela, Studying possibility in a clustering algorithm for rbfnn design for function approximation, Neural Computing and Applications 17 (1) (2008) 75–89.
[28] V. Cherkassky, H. Lari-Najafi, Constrained topological mapping for nonparametric regression analysis, Neural Networks 4 (1) (1991) 27–40.
[29] H. Pomares, I. Rojas, J. Ortega, J. González, A. Prieto, A systematic approach to a self-generating fuzzy rule-table for function approximation, IEEE Transactions on Systems, Man, and Cybernetics-Part B 30 (3) (2000) 431–447.
[30] J.H. Friedman, Projection pursuit regression, Journal of the American Statistical Association 76 (1981) 817–823.
[31] J.H. Friedman, Multivariate adaptive regression splines (with discussion), Annals of Statistics 19 (1991) 1–141.
[32] R.M. May, Simple mathematical models with very complicated dynamics, Nature 261 (1976) 459–467.
[33] M. Hénon, A two-dimensional mapping with a strange attractor, Communications in Mathematical Physics 50 (1976) 69–77.
[34] M.C. Mackey, L. Glass, Oscillation and chaos in physiological control systems, Science 197 (4300) (1977) 287–289.
[35] A. Guillén, J. González, I. Rojas, H. Pomares, L.J. Herrera, O. Valenzuela, A. Prieto, Using fuzzy logic to improve a clustering technique for function approximation, Neurocomputing 70 (16–18) (2007) 2853–2860.
[36] A. Guillén, I. Rojas, J. González, H. Pomares, L.J. Herrera, O. Valenzuela, F. Rojas, Output value-based initialization for radial basis function neural networks, Neural Processing Letters 25 (3) (2007) 209–225.



Alberto Guillen was born in 1979. He received the M.A.Sc. degree in Computer Science in 2002 and the Ph.D. in 2007 from the University of Granada, Spain. He is currently an associate lecturer within the Department of Computer Technology and Architecture in the University of Granada. His current areas of research interest are in the fields of clustering, function approximation, classification, feature selection, within the context of parallel programming.

Luis Javier Herrera was born in 1978. He received the M.Sc. degree in Computer Engineering in 1995 from the University of Granada, Spain. He is currently an associate lecturer within the Department of Computer Architecture and Technology in the University of Granada. His current areas of research interest are in the fields of function approximation, self-organizing fuzzy interpretable models and time series prediction.

Gines Rubio was born in Bretigny sur Orge, France, in 1979. He received his M.S. in Computer Engineering from the University of Granada, Spain, in 2002. Presently, he is working toward the Ph.D. degree at the University of Granada in the Department of Computer Architecture and Computer Technology. His research interests are application of kernel methods for regression, time series prediction, hybrid systems, genetic programming and high performance computing platforms.

Hector Pomares was born in 1972. He received the M.A.Sc. degree in Electronic Engineering in 1995, the M.Sc. degree in Physics in 1997, and his Ph.D. degree in 2000 from the University of Granada, Spain. He is currently a Fellowship holder within the Department of Computer Architecture and Technology in the University of Granada. His current areas of research interest are in the fields of function approximation and on-line control using adaptive and self-organizing fuzzy systems.

Amaury Lendasse was born in 1972 in Tournai, Belgium. He received M.S. degree in Mechanical Engineering from the Université catholique de Louvain (Belgium) in 1996, M.S. in control in 1997 and Ph.D. in 2003 from the same university. In 2003, he has been a post-doctoral researcher in the Computational Neurodynamics Lab at the University of Memphis. Since 2004, he is a senior researcher and a docent in the Adaptive Informatics Research Centre in the Aalto University School of Science and Technology (former Helsinki University of Technology) in Finland. He has created and is leading the Environmental and Industrial Machine Learning Group (former Time Series Prediction and Chemoinformatics Group). He is chairman of the annual ESTSP conference (European Symposium on Time Series Prediction) and member of the editorial board and program committee of several journals and conferences in machine learning. He is the author or the coauthor of around 140 scientific papers in international journals, books or communications to conferences with reviewing committee. His research includes time series prediction, chemometrics, variable selection, noise variance estimation, determination of missing values in temporal databases, nonlinear approximation in financial problems, functional neural networks and classification.

Ignacio Rojas received the B.Sc. degree in Electronic Physics in 1992 and the Ph.D. degree in Intelligent Systems with honors in 1996, both from the University of Granada, Spain. He has been visiting professor at the University of Dortmund (Germany) at the Department of Electrical Engineering (1993–1995, 2001), as a visiting researcher with the BISC Group of Prof. L. Zadeh, University of California, Berkeley (1998) and as visiting professor at the University of Genova (Italy), Department of Computer Science (2003).

He is currently an Associate Professor with the Department of Computer Architecture and Computer Technology, University of Granada, member of the IEEE Computational Intelligence Society and Secretary of the IEEE Spanish Regional Interest Group of the Neural Networks Council.

His research interests include hybrid system, hardware–software implementation, combination of intelligent system for adaptive control, self-organizing neuro-fuzzy systems, neural networks, time series forecasting, data mining and architectures for complex optimization problems.