Review

Applications of information theory, genetic algorithms, and neural models to predict oil flow

Oswaldo Ludwig Jr.^{a,*}, Urbano Nunes^a, Rui Araújo^a, Leizer Schnitman^{b,1},
Herman Augusto Lepikson^{b,1}

^a Institute for Systems and Robotics (ISR) and the Department of Electrical and Computer Engineering, University of Coimbra, P-3030-290 Coimbra, Portugal

^b Polytechnic School of Federal University of Bahia, Salvador, CEP 40000, Brazil

ARTICLE INFO

Article history:

Received 21 October 2008

Received in revised form 3 December 2008

Accepted 11 December 2008

Available online 24 December 2008

PACS:

89.70.-a

84.35.+i

05.45.-a

02.60.-x

Keywords:

Information theory

Genetic algorithm

Neural networks

Non-linear modeling

ABSTRACT

This work introduces a new information-theoretic methodology for choosing variables and their time lags in a prediction setting, particularly when neural networks are used in non-linear modeling. The first contribution of this work is the Cross Entropy Function (XEF) proposed to select input variables and their lags in order to compose the input vector of black-box prediction models. The proposed XEF method is more appropriate than the usually applied Cross Correlation Function (XCF) when the relationship among the input and output signals comes from a non-linear dynamic system. The second contribution is a method that minimizes the Joint Conditional Entropy (JCE) between the input and output variables by means of a Genetic Algorithm (GA). The aim is to take into account the dependence among the input variables when selecting the most appropriate set of inputs for a prediction problem. In short, these methods can be used to assist the selection of input training data that have the necessary information to predict the target data. The proposed methods are applied to a petroleum engineering problem; predicting oil production. Experimental results obtained with a real-world dataset are presented demonstrating the feasibility and effectiveness of the method.

© 2008 Elsevier B.V. All rights reserved.

Contents

1. Introduction	2871
2. Discrete joint conditional entropy	2871
3. Proposed cross entropy function	2872
3.1. Cross correlation function	2872
3.2. Cross entropy function	2872
3.3. Application of XEF to choose input variables and their time lags	2873
4. Considering dependence among the input variables	2873
4.1. Direct approach	2873
4.2. Indirect approach	2874
4.3. Justifying the equivalence between the direct and indirect approaches	2874

* Corresponding author.

E-mail addresses: oludwig@isr.uc.pt (O. Ludwig), urbano@isr.uc.pt (U. Nunes), rui@isr.uc.pt (R. Araújo), leizer@ufba.br (L. Schnitman), herman@ufba.br (H.A. Lepikson).

¹ Tel.: +55 071 3203 9472; fax: +55 071 8881 8008.

5. Experiments	2876
5.1. Case study	2877
5.2. The applied neural predictor model.	2879
5.3. Empirical comparison between XCF and XEF	2879
5.4. Manually minimizing the JCE	2883
5.5. Applying the direct approach to choose the input vector by GA	2884
5.6. Applying the indirect approach to choose the input vector by GA	2884
6. Conclusion	2884
Acknowledgements	2884
References	2884

1. Introduction

The application of Artificial Neural Networks (ANN) has been attracting much research interest. In particular, the application of ANNs in function approximation has been widely studied. However, as in other black-box models, the success of ANN applications depends heavily on the choice of the input variables of the problem. Some previous works have proposed doing feature selection by means of a Mutual Information (MI) criterion. Simon and Verleysen [1] proposed the use of MI in order to select the delay for time series phase space reconstruction, while François et al. [2] combined the MI criterion with a forward feature selection strategy to offer a good trade-off between optimality of the selected feature subset and computation time.

This paper proposes Cross Entropy Function (XEF) analysis for the selection of input variables in regression problems. The proposed approach may not only support the selection of pertinent data [3,4], but also help to determine the lag between signals. The analysis does not assume a linear input–output relation, which contrasts with the method based on Cross Correlation Function (XCF). Therefore, the XEF is more suitable to analyze non-linear relationships between signals. In spite of the effectiveness of the proposed XEF method in input variable selection, the combination of individually good input variables does not necessarily lead to good prediction performance. In other words, “the m best input variables may not be the best m input variables”. In order to consider redundancy among input variables the present work proposes a method that minimizes the Joint Conditional Entropy (JCE) between the input and output variables by means of a Genetic Algorithm (GA). The GA solution considers two methods: the direct and indirect. The direct approach applies the JCE as the fitness function of the GA. As the JCE calculation demands high computational effort, the indirect approach, which is based on a less expensive fitness function, is proposed to decrease the computational cost.

A review of some entropy concepts is presented in Section 2. Section 3 describes the proposed XEF tool and how it can be used in order to select the input variables. Section 4 proposes a new method that extends the XEF approach by complementing it in order to deal with significant dependence among the input variables. A set of experiments demonstrating the feasibility and effectiveness of the new approach is presented in Section 5. Conclusions are finally drawn in Section 6.

2. Discrete joint conditional entropy

This section presents a brief overview of some concepts of information theory [5,6], that will be useful to describe the proposed methods. In particular we describe JCE which enables the evaluation of the amount of information that lacks to determine the target output variable. The method proposed in this paper considers discrete variables. Therefore, each continuous variable is discretized by sub-dividing its range into a finite set of intervals. For each interval, a value is considered to represent it; the mean value is used. In this paper, without losing generality, for all variables, N discrete possible values (bins) are considered.

The conditional entropy of the scalar discrete random variable Y , assuming the event $X = x_k$, is given by

$$H(Y|x_k) = \sum_{i=1}^N -\log(P(y_i|x_k)) \cdot P(y_i|x_k) \quad (1)$$

where $P(y_i|x_k)$ is the probability that $Y = y_i$ assuming event $X = x_k$. The conditional entropy of Y , considering the knowledge of X , is given by

$$H(Y|X) = \sum_{k=1}^N H(Y|x_k) \cdot P(x_k) \quad (2)$$

which defines the uncertainty about Y when all the trials of X are known. A suitable calculation form is

$$H(Y|X) = H(X, Y) - H(X) \quad (3)$$

where $H(X, Y)$ is the joint entropy:

$$H(X, Y) = \sum_{i=1}^N \sum_{k=1}^N -\log(P(x_i, y_k)) \cdot P(x_i, y_k) \quad (4)$$

The concept of joint entropy (4) can be extended to the following general formulation (5), where n input variables $X_1, \dots, X_n$, and N discrete possible values (bins) for each variable are assumed:

$$H(Y, X_1, \dots, X_n) = \sum_{i=1}^N \sum_{i_1=1}^N \dots \sum_{i_n=1}^N -\log(P(y_i, x_{1,i_1}, \dots, x_{n,i_n})) \cdot P(y_i, x_{1,i_1}, \dots, x_{n,i_n}) \quad (5)$$

where x_{j,i_j} is event i_j of variable X_j , for $i_j = 1, \dots, N$, $j = 1, \dots, n$.

Due to the exponential explosion of combinations, considering the joint probability over 1000 examples, a 64 bit processor with 2 GHz, in Matlab environment, requires almost 15 min to compute (5) when six random variables are present, considering 15 discrete levels ($N = 15$). Here, a crucial problem is the curse of dimensionality associated with having 15^6 bins in the histogram to compute meaningful probabilities and estimate entropy.

This work defines

$$H(Y|(X_1, \dots, X_n)) = H(X_1, \dots, X_n, Y) - H(X_1, \dots, X_n) \quad (6)$$

as the JCE, which defines the amount of expected information that lacks to determine a target variable Y by the use of a set of input variables $X_1, X_2, \dots, X_n$.

Eq. (6) can be applied to an ANN training data set. If the JCE is zero, it means that the input training dataset contains all the information about the target dataset and the designer can be certain that this training dataset is appropriate.

3. Proposed cross entropy function

This section proposes the XEF analysis method. The goal of the method is to allow the selection of the time lag to compose a dynamic model. The XEF tool results from an analogy with the XCF. Thus, it is important to recall the XCF.

3.1. Cross correlation function

The XCF is based on the Pearson Correlation Coefficient, r . This coefficient measures the degree of correlation among two random variables, based on the quality of a linear adjustment of the data. It takes values between -1 and 1 , where $r = 1$ refers to a positive perfect correlation among the two variables, $r = -1$ refers to a perfect negative correlation among the two variables (i.e. if one increases, the other decreases) and $r = 0$ means that the two variables do not linearly depend one from the other. However, when $r = 0$, a non-linear dependence can exist between the variables. The Pearson Coefficient of Correlation is calculated over N samples, according to

$$r = \frac{\sum_{i=0}^{N-1} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N-1} (x_i - \bar{x})^2} \sqrt{\sum_{i=0}^{N-1} (y_i - \bar{y})^2}} \quad (7)$$

where x_i and y_i are the measured values of random variables x and y , and $\bar{x}$, $\bar{y}$ are the mean values of x and y , respectively. Considering two temporal series, the XCF is computed through successive lags in one of the two series. For each lag, the value of the Pearson Correlation Coefficient (7) is computed.

3.2. Cross entropy function

The XCF is based on a linear adjustment between variables. Due to this, the method encounters some problems when applied to non-linear systems. In order to avoid these problems, this paper proposes a suitable analysis that is more appropriated for non-linear dynamic relationships.

Let us first define the normalized mutual information, R , between two signals, X and Y :

$$R = \frac{H(Y) - H(Y|X)}{H(Y)} \quad (8)$$

Parameter R is restricted to the $[0, 1]$ interval and represents the amount of information about the target variable that is revealed by the input variable. Notice that if X contains all the necessary information to predict Y , then $H(Y|X) = 0$ and $R = 1$. On the other hand, if X does not contain information about Y , then $H(Y|X) = H(Y)$ and $R = 0$.

The XEF is a discrete function that maps possible time lags between two temporal series into the corresponding normalized mutual information R . The XEF is computed through successive lags in one of the two temporal series, similarly to the XCF computation. For each lag, the value of R (8) is computed. Then, the time lag to implement the prediction is chosen as the one for which R attains the maximum value. For each input variable, a time lag is chosen using this XEF analysis. This approach is used to independently choose the most appropriate time lag for each input variable, without considering the dependence among them.

Fig. 1. Selection of variables and its delay in order to compose the regression vector of a neural predictor model.

Notice that, there are other meanings for the isolated term cross-entropy. For example, cross-entropy has been used for an energy function in the calculation of the back-propagation error on binary classification tasks [7]. The present work does not have relationship with the context of [7].

3.3. Application of XEF to choose input variables and their time lags

Computing lags in normalized mutual information is not a novel technique in causality analysis [3]. However, the present work proposes XEF to choose input variables and their time lags in a prediction setting. If the extreme point of XEF happens with positive lag, the current sample of the output series has relationship with a delayed sample of the input series, due to the system dynamics. If the extreme point of XEF happens with null lag, then the model seems to be static. However, when the extreme point of XEF happens to have negative lag, it means that the output in the current time will affect the future of the input variable (it shows that past output values are affected by future input data, and characterizes a non-causal system, which will not be considered here). Fig. 1 illustrates this application of XEF to predict oil flow based on the oil well data. If the XEF analysis suggests a negative lag for an input variable x , such variable has an inverse causal relationship with the output variable. In this case, variable x is not suitable to compose the input vector.

4. Considering dependence among the input variables

This section proposes a new method that extends the XEF approach by complementing it in order to deal with significant dependence among the input variables. Instead of estimating input–output lags independently for each input variable, the lags of input variables are jointly chosen by minimizing the JCE between the input vector and output. Considering n input variables and a time range of possible lags with d days, an algorithm that performs this minimization by exhaustive search must explore n^d possible solutions, calculating (6) n^d times. The computational effort of an exhaustive search algorithm makes such approach unfeasible for real applications. To overcome this problem, this work applies a GA to jointly estimate the lags between the various input variables and the output.

4.1. Direct approach

In the first approach the JCE (6) is directly used as the fitness function of the GA and the objective of the optimization problem is to obtain the input signals $X_1, \dots, X_n$ that attain the minimum JCE, according to

$$\min_{X_1 \dots X_n} H(Y|X_1, \dots, X_n) \quad (9)$$

More specifically, the goal is to find the optimal lags for the input variables. In the GA, the chromosomes are composed of the lag of all the input variables. The input vector, composed of input variables with lags suggested by the XEF analysis, is sowed as a first individual in the initial population, with the objective of accelerating the convergence of the algorithm. Then, the other individuals of the initial population are randomly generated, with uniform distribution and standard deviation δ , around the seeded individual.

The GA applied in this work operates on the decimal basis. For details on the GA algorithm, in particular the decimal basis GA, see [8]. The chosen crossover genetic operator was the BLX- α [8], defined by

$$g_c^n(k+1) = \text{round}(\alpha \cdot g_1^n(k) + (1 - \alpha)g_2^n(k)) \quad (10)$$

where k is the generation; $\alpha \in [0,1]$ with uniform distribution; g_c^n is the gene n of the child chromosome c , and g_1^n and g_2^n are genes corresponding to position n of parent chromosome vectors. In short, the chromosome of the m^{th} individual is given by the vector $[g_m^1, g_m^2, \dots, g_m^n]^T$ where gene g_m^n codifies the lag of input variable n . Since lags are natural numbers, a $\text{round}(\cdot)$ operator that returns the nearest integer, is used in (10). As the crossover operation (10) is a linear interpolation, its usage pre-

serves the initial limits of the search space. This is a useful property because it satisfies an inequality constraint: the minimum delay of all the contribution variables must be D days, where D is the length of the prediction horizon. Notice that, if the aim is to predict the oil flow D days ahead, it is not possible to adopt lags shorter than D days for the input variables. The whole initial population is generated with lags equal to or greater than D days. Thus, by using the crossover operator (10) this minimum time lag constraint is preserved throughout the generations.

In order to use (6) to evaluate individual m , the algorithm composes a set of samples of each input signal X_n delayed by g_m^n days, remembering that the gene g_m^n is the lag, suggested by individual m , for input variable n .

After fitness evaluation, the algorithm organizes the individuals by their fitness values, ranking the indexes of the individuals into a vector, from the best to the worst. Then, the crossover operator is applied in order to create a new generation. Each individual has a probability of reproduction that is given by its fitness value. More adapted individuals (i.e. the individuals in higher positions of the ranking) have more probability of participating in the crossover operation.

A selection process based on a uniform random variable is not adequate for our purposes. The random variable used to select individuals to participate in the crossover must have a probability density that is greater for small values than for large values in order to favor the more adapted individuals. To do this, the following random variable σ' is proposed:

$$\sigma' = \frac{e^{a\sigma} - 1}{e^a - 1}, \quad (11)$$

where a is a positive constant, and $\sigma \in [0,1]$ is a random variable with uniform distribution. This approach makes it possible to increase or decrease the selective pressure. Parameter a can be increased to increase selective pressure. The increase in selective pressure leads to faster convergence; however, for large selective pressure the algorithm may drive the population to a local minimum. Individuals are selected for crossover by its rank i , according to the discrete random variable

$$i = \text{round}(T\sigma'), \quad (12)$$

where T is the total number of individuals.

The algorithm applies the elitism operator [8] only on the best individual which is replicated in the next generation.

4.2. Indirect approach

In the indirect approach the GA is applied with the same parameters and criteria used in the direct approach. However, the fitness function is changed to avoid the computational cost of (6). The applied fitness function is based on the principle of Max-Relevance and Min-Redundancy [9]. According to this principle it is possible to solve the optimization problem (9) by jointly solving the following two problems:

$$\max_{X_1 \dots X_n} V \quad (13)$$

$$\min_{X_1 \dots X_n} P \quad (14)$$

where

$$V = \frac{1}{n} \sum_{i=1}^n I(X_i; Y), \quad (15)$$

$$P = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n I(X_i; X_j) \quad (16)$$

and, for any random variables U and Z , $I(U;Z)$ is the mutual information between U and Z . The application of the Max-Relevance principle corresponds to searching the input variables and their lags in order to perform maximization (13), i.e. to increase the mean of all mutual information values, $I(X_i; Y)$, ($i = 1, \dots, n$), where X_i is the individual input i , and Y is the output. The application of the Min-Redundancy principle corresponds to searching the input variables and their lags in order to perform minimization (14), i.e. to decrease the mean of all the mutual information values $I(X_i; X_j)$, ($i = 1, \dots, n$), where X_i and X_j are individual inputs. The minimum value defined in (14) is attained when the inputs are mutually exclusive.

The indirect approach applies the GA, described in Section 4.1, to maximize the fitness function $\Phi = V - P$, i.e. the following maximization is performed:

$$\max_{X_1 \dots X_n} \Phi \quad (17)$$

The maximization of Φ is a convenient approach to simultaneously maximize V (13) and minimize P (14).

4.3. Justifying the equivalence between the direct and indirect approaches

Considering that the direct approach of Section 4.1 consists of minimizing the JCE, we will show that the indirect approach also minimizes the JCE. Let us first define the discrete mutual information for multiple scalar variables [9]:

$$\begin{aligned}
J(X_1, \dots, X_n) &= \sum_{k=1}^n H(X_k) - H(X_1, \dots, X_n) \\
&= - \sum_{j=1}^n \sum_{i=1}^N P(x_{j,i}) \log P(x_{j,i}) + \sum_{i_1=1}^N \dots \sum_{i_n=1}^N P(x_{1,i_1}, \dots, x_{n,i_n}) \log P(x_{1,i_1}, \dots, x_{n,i_n}) \\
&= \sum_{i_1=1}^N \dots \sum_{i_n=1}^N P(x_{1,i_1}, \dots, x_{n,i_n}) (-\log(P(x_{1,i_1}) \dots P(x_{n,i_n})) + \log P(x_{1,i_1}, \dots, x_{n,i_n})) \\
&= \sum_{i_1=1}^N \dots \sum_{i_n=1}^N P(x_{1,i_1}, \dots, x_{n,i_n}) \log \frac{P(x_{1,i_1}, \dots, x_{n,i_n})}{P(x_{1,i_1}) \dots P(x_{n,i_n})}
\end{aligned} \tag{18}$$

where $P(x_{n,i_n})$ is the probability of event $X_n = x_{n,i_n}$. Isolating the joint entropy $H(X_1, \dots, X_n)$ in the first line of (18) and substituting it in (6) we can alternatively express the JCE as:

$$\begin{aligned}
H(Y|X_1, \dots, X_n) &= H(X_1, \dots, X_n, Y) - H(X_1, \dots, X_n) \\
&= \left(\sum_{k=1}^n H(X_k) + H(Y) - J(X_1, \dots, X_n, Y) \right) - \left(\sum_{k=1}^n H(X_k) - J(X_1, \dots, X_n) \right) \\
&= H(Y) - J(X_1, \dots, X_n, Y) + J(X_1, \dots, X_n)
\end{aligned} \tag{19}$$

According to the last line of (19), to minimize the JCE it is sufficient to choose input variables and their lags maximizing $J(X_1, \dots, X_n, Y)$ and minimizing $J(X_1, \dots, X_n)$. Since the minimum value of $P(x_{1,i_1}, \dots, x_{n,i_n}) / (P(x_{1,i_1}) \dots P(x_{n,i_n}))$ is attained when all the input variables are independent of each other, i.e. when

$$P(x_{1,i_1}, \dots, x_{n,i_n}) = P(x_{1,i_1}) \dots P(x_{n,i_n}), \tag{20}$$

then, from the last line of (18), it results that performing the minimization associated to the principle of Min-Redundancy (14) is a sufficient condition to minimize $J(X_1, \dots, X_n)$. Similarly, since the maximum value of $P(x_{1,i_1}, \dots, x_{n,i_n}, y_i) / (P(x_{1,i_1}) \dots P(x_{n,i_n})P(y_i))$, subject to (20), is attained by maximizing the dependency between input variables and the output Y , then, it results that performing the maximization associated to the principle of Max-Relevancy (13), is a sufficient condition to maximize $J(X_1, \dots, X_n, Y)$. Thus using the indirect approach, i.e. performing optimizations (13) and (14) is a sufficient condition to minimize the JCE.

Fig. 2. Samples of Percentage of Water (BSW) in points and the result after the filter in line.

5. Experiments

This section reports a set of experiments performed with real data. Different input vectors are composed, each of which is based on one of the methods that were described in the previous sections. These vectors are applied to a neural predictor model in order to evaluate their performance over a real dataset. A comparison between the input selection methods is performed.

Fig. 3. Samples of BSW over one year.

Fig. 4. Samples of RGO over one year.

5.1. Case study

The aim of this set of experiments is to predict the oil flow 30 days ahead (i.e. a delay of $D = 30$ is considered) in order to enable adjustment of storage infrastructure. As described above, the prediction is based on a neural predictive model and the problem is the choice of the best set of input variables to compose the input vector for this model. In these experiments, all the available production test data are analyzed, more specifically, the percentage of gas (RGO), the percentage of water (BSW), the dynamic oil level (DOL), and the static oil level (SOL), in order to predict the oil flow (OF).

Fig. 5. Samples of DOL over one year.

Fig. 6. Samples of SOL over one year.

The signals in the dataset have a high level of noise. To overcome this, the signals were pre-processed with an algorithm that first subtracts the mean value of the time series and then applies a digital low-pass 3rd order Butterworth FIR filter, and finally sums the series mean value in order to recover the DC level. The FIR filter uses an impulsive response composed of 4000 samples. For each variable, the cut-off frequency was chosen according to the corresponding signal and noise characteristics. An example of the procedure described above is illustrated in Fig. 2.

Fig. 7. Samples of OF over one year.

Fig. 8. XCF between oil flow (OF) and percentage of gas (RGO).

The dataset is composed of sample data corresponding to one year of daily samples of all the physical parameters of the oil well in order to generate the XEF. Figs. 3–7 illustrate these parameters in the time window used in this work. The signals are represented after noise filtering.

5.2. The applied neural predictor model

The approximation model used to investigate the input vector quality is a Multilayer Perceptron (MLP) neural network [10,11]. A MLP with one sigmoidal hidden layer and linear output layer is enough to guarantee a universal approximation capability [12]. A MLP with architecture $n \times 8 \times 1$, where n is the number of input variables, was applied. Sigmoidal activation functions were used in the hidden layer, and linear activation functions were used in the output layer. The training process uses the Nguyen–Widrow algorithm [13] for weights initialization, and the Levenberg–Marquardt training method with Mean Squared Error (MSE) as performance function, and minimum gradient threshold, $T = 1e-10$, as stop criterion. The MLP model used in this work is described by

$$\begin{aligned} y_h &= \varphi(W_1 x + b_1) \\ \hat{y} &= W_2 y_h + b_2 \end{aligned} \quad (21)$$

where x is the input vector, W_1 and b_1 are the synaptic weights matrix of the hidden layer and the bias vector of the hidden layer, respectively, y_h is the output vector of the hidden layer, $\varphi(\cdot)$ is the sigmoid function, W_2 and b_2 are the synaptic weights matrix of the output layer and bias of the output layer, respectively, and $\hat{y}$ is the ANN output value (i.e. predicted sample).

5.3. Empirical comparison between XCF and XEF

This sub-section presents the results of two experiments to compare the performance of the XCF and XEF methods to select input variables. The first experiment uses the XCF while the second experiment is similar but uses XEF. These methods are used to choose the suitable lag of the input variables which compose the input vector for the above described neural model used to predict the oil flow some days ahead. The experiments were repeated 100 times in order to compute both the average and the standard deviation of the model MSE. These parameters are computed over a validation dataset composed of 90 samples of all the variables. Figs. 8–11 illustrate the XCF of all the available variables regarding the oil flow.

In XCF analysis, according to the suggestion of choosing the time lag that minimizes XCF, the following variable lags were chosen to compose the input vector to the neural model: $[RGO(i-10), BSW(i-20), DOL(i), SOL(i)]^T$.

The target output is the current sample of oil flow $OF(i)$. Notice that, this is not a predictive model, because the current samples of DOL and SOL are used in the estimate of the current sample of OF. This model presents average MSE equal to

Fig. 9. XCF between oil flow (OF) and percentage of water (BSW).

2.59e−5 and standard deviation of 4.36e−6. However, this model is not suitable since the goal is to predict the oil production 30 days ahead.

Fig. 10. XCF between Oil Flow (OF) and Dynamic Level (DOL).

Fig. 11. XCF between oil flow (OF) and static level (SOL).

The second set of experiments uses the proposed XEF to perform an analysis for selecting the input vector for the same ANN. The probabilities required to compute the XEF (Sections 2,3) are obtained from the time series, by means of normalized histograms. Figs. 12–15 illustrate the XEF of all the available variables related to the oil flow.

Fig. 12. XEF between oil flow (OF) and percentage of gas (RGO).

Fig. 13. XEF between oil flow (OF) and percentage of water (BSW).

Using the XEF analysis method proposed in Section 3.2, the following set of delayed variables was chosen to compose the input vector of the neural model: $[RGO(i - 45), BSW(i - 27), DOL(i - 33), SOL(i - 8)]^T$. The target output is the current sample of oil flow $OF(i)$. This results in a predictive model which can estimate a sample of oil 8 days ahead. The model suggested by

Fig. 14. XEF between oil flow (OF) and dynamic level (DOL).

Fig. 15. XEF between oil flow (OF) and static level (SOL).

Fig. 16. Flowchart of the third experiment.

the XEF analysis reveals an average MSE equal to $1.65e-5$ and a standard deviation of $2.24e-6$. The higher performance of the model suggested by the XEF analysis can be explained by the application of the JCE.

According to (6), the JCE $H(OF(i)|RGO(i-10), BSW(i-20), DOL(i), SOL(i))$ has the value of 0.1952. On the other hand, the value of the JCE $H(OF(i)|RGO(i-45), BSW(i-27), DOL(i-33), SOL(i-8))$ is 0.1613. These experiments illustrate that the input vector, selected by the XEF analysis, explains the oil flow better, because the JCE is smaller than in the model suggested by the XCF. However, being able to predict the oil flow 8 days ahead is not enough for our purpose of predicting 30 days ahead.

5.4. Manually minimizing the JCE

The approach employed in the experiment presented in Section 5.3 assumes that the input variables are independent. However, as the input variables often present dependency, the results may be improved using the methods proposed in Section 4. The experiments presented in this section and in Sections 5.5, 5.6 evaluate the performance of the methods proposed in Section 4. Thus, in the third experiment, considering the input vector suggested by XEF analysis, the JCE is manually minimized in order to improve model performance. The flowchart in Fig. 16 explains the methodology applied in this experiment.

According to the methodology shown in Fig. 16, acceptable performance is reached when the dynamics of oil flow is considered by the inclusion of the sample $OF(i-30)$ in the input vector. The problem now becomes auto-regressive and the new input vector is the following:

$$u = [RGO(i-45), BSW(i-30), DOL(i-33), SOL(i-41), OF(i-30)]^T. \quad (22)$$

Using this input vector the JCE is 0.1377. Besides the inclusion of the sample $OF(i-30)$, the samples $BSW(i-27)$ and $SOL(i-8)$ were replaced with $BSW(i-30)$ and $SOL(i-41)$ in order to predict the oil flow 30 days ahead. Moreover, we can see in Fig. 13 that at the lag 30 the XEF between BSW and OF has the acceptable value of $R = 0.69$. Similarly Fig. 15 illustrates that the XEF between SOL and OF has the acceptable value of $R = 0.745$ at the lag 41. This proposed model can predict the oil flow with average MSE equal to $1.07e-5$ and standard deviation of $2.14e-6$. These results consider 100 experiments with a validation dataset. Notice that, the engineer can accept the suggestion of XEF as a reference. However, it is possible to include other variables or to change their lags. This would be useful if it led to a better value of JCE.

Table 1
Experiments results.

	Mean MSE	Standard deviation	JCE	Delay of prediction model (days)	Expended time (s)
XCF Analysis	2.59e–5	4.36e–6	0.1952	0	–
XEF Analysis	1.65e–5	2.24e–6	0.1613	8	–
XEF + manual min. of JCE	1.07e–5	2.14e–6	0.1377	30	–
GA-direct approach	9.89e–6	3.23e–6	0.1128	30	1.21e4
GA-indirect approach	1.01e–5	2.75e–6	0.1263	30	2.74e2

5.5. Applying the direct approach to choose the input vector by GA

The fourth experiment applies the proposed direct approach of GA in order to select the best input vector for the neural model, as explained in Section 4.1. The initial population is composed of 40 individuals randomly generated around the seeded individual which has the chromosome $C_{seeded} = [45, 30, 33, 41, 30]^T$, according to the lags suggested by (22), i.e. the manual minimization of JCE performed in the third experiment. A uniform distribution with standard deviation of 10 days was adopted for the initial population. The whole initial population is generated with lags equal to or greater than 30 days.

The GA runs with selective pressure $a = 5$. After 40 generations the population converged for the best individual $C_{best} = [50, 33, 32, 39, 30]^T$, which has fitness value (i.e. JCE) equal to 0.1128. The expended time was 1.21e4 seconds in Matlab environment, running on a PC–64 bits with 2 GHz. Using the lags proposed by the chromosome C_{best} the neural model can predict the oil flow with average MSE equal to 9.89e–6 and standard deviation of 3.23e–6, over 100 experiments, repeated automatically, with the validation dataset.

5.6. Applying the indirect approach to choose the input vector by GA

The fifth experiment applies the indirect approach of GA optimization method, as explained in Section 4.2. The initial population is generated in the same way described in Section 5.5, including the same seeded chromosome C_{seeded} .

As in the previous experiment, the GA runs with selective pressure $a = 5$. The population converges to the best individual $C_{best} = [52, 30, 31, 37, 37]^T$. The expended time was 2.74e2 seconds. The input vector with lags proposed by C_{best} has JCE equal to 0.1263. Using this input vector the neural model can predict the oil flow with average MSE equal to 1.01e–5 and standard deviation of 2.75e–6, over 100 experiments.

Table 1 summarizes the results of the experiments presented in Section 5. The neural network MSE decreases when the JCE decreases. The JCE reaches the smallest value with the application of GA and the direct method.

6. Conclusion

Compared to the method that uses XCF, the prediction performance improves when XEF is used, because the approach that uses XEF considers the non-linear dynamic relationship between variables. Further improvements are attained when XEF is used in conjunction with JCE analysis. Finally, the best performance is attained when the method integrates XEF and the direct approach of GA to consider also dependence among input variables. This method is the main contribution of this work. In order to avoid the computational cost of JCE calculation, an indirect GA approach was proposed. The indirect approach slightly decreases the GA performance. However, it is 44 times faster than the direct approach, in our case study. Considering the computational effort applied to calculate the fitness function during the GA application, it is useful to reduce the number of GA generations. Therefore, the authors are working on a new GA model, inspired in the approach suggested by [14].

Acknowledgements

This work was supported by Petrobras, Institute for Systems and Robotics (ISR-UC), and Portuguese Foundation for Science and Technology (FCT) under Project PMITS06.

References

- [1] Simon G, Verleysen M. High-dimensional delay selection for regression models with mutual information and distance-to-diagonal criteria. *Neurocomputing* 2007;70(7–9):1265–75.
- [2] François D, Rossi F, Wertz V, Verleysen M. Resampling methods for parameter-free and robust feature selection with mutual information. *Neurocomputing* 2007;70(7–9):1276–88.
- [3] Yang H, Moody J. Feature selection based on joint mutual information. In: *Advances in Intelligent Data Analysis (AIDA)*. Computational intelligence methods and applications (CIMA), international computer science conventions, Rochester, New York; 1999.
- [4] Torkkola K. Feature extraction by non-parametric mutual information maximization. *J Mach Learn Res* 2003;3:1415–38.
- [5] Cover T, Thomas J. *Elements of information theory*. New York: Wiley; 1991.
- [6] Papoulis A. *Probability, random variables, and stochastic processes*. 3rd ed. New York, NY, USA: McGraw-Hill Inc.; 1991. ISBN: 0 07-048477-5.
- [7] Rubinstein RY, Kroese DP. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*. New York: Springer-Verlag; 2004.

- [8] Eshelman LJ, Shaffer JD. Real-coded genetic algorithms and interval schemata, foundations of genetic algorithms II, CA; 1992. p. 187–202.
- [9] Peng H, Long F, Ding C. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans Pattern Anal Machine Intell* 2005;1226–38.
- [10] Chen S, Billings S. Neural networks for nonlinear dynamic system modeling and identification. *Int J Control* 1992;56(2):319–46.
- [11] Mahmouda MA, Ben-Nakhi AE. Neural networks analysis of free laminar convection heat transfer in a partitioned enclosure. *Commun Nonlinear Sci Numer Simulat* 2007;12(7):1265–76.
- [12] Hornik K, Stinchcombe M, White H. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks* 1999:551–60.
- [13] Nguyen D, e Widrow B. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptatives weights. CA: Stanford University; 1990.
- [14] Chen J, Yang X. Real optimal parameter estimation for Muskingum model based on gray-encoded accelerating genetic algorithm. *Commun Nonlinear Sci Numer Simulat* 2007;12:849–58.