Variable and Delay Selection Using Neural Networks and Mutual Information for Data-Driven Soft Sensors

Francisco Souza, Pedro Santos, and Rui Araújo
Institute for Systems and Robotics (ISR-UC), and Department of Electrical and Computer Engineering (DEEC-UC), University of Coimbra, Pólo II, PT-3030-290 Coimbra fasouza@isr.uc.pt, rui@isr.uc.pt

Abstract

This paper proposes a new method for input variable and delay selection (IVDS) for Soft Sensors (SS) design. The IVDS algorithm is composed by the following steps: (1) Time delay selection; (2) Identification and exclusion of redundant variables; (3) Best variables subset selection. The IVDS algorithm proposed in this work performs the delay and variable selection through two distinct methods, mutual information (MI) is applied to delay selection and for variable selection a multilayer perceptron (MLP) based approach is performed. It is shown in the case studies that the application of the delay selection before applying the variable selection increases the generalization of the MLP-model. The algorithm uses the relative variance tracking precision (RVTP) criterion and the mean square error (MSE) to evaluate the precision of soft sensor. Simulation results are presented showing the effectiveness of the method.

Keywords: variable selection, soft sensors, neural networks, multilayer perceptrons.

1. Introduction

Data-driven soft sensors (DDSS) are inferential models that use on-line available sensor measures for on-line estimation of variables which cannot be automatically measured at all, or can only be measured at high cost, sporadically, or with high delays (e.g. laboratory analysis). These models are based on measurements which are recorded and provided as historical data, the methods themselves are empirical predictive models like Multi-layer Perceptron (MLP), Support Vector Machines (SVM), etc. They are valuable tools to many industrial applications such as refineries, pulp and paper mills, wastewater treatment systems, just to give few examples [5].

The development of DDSS can be divided into four main stages: (I) Data collection, selection of historical data; (II) Data pre-processing; (III) Model selection, training and validation; (IV) Soft sensor maintenance. In the first stage the training and evaluation data of the model are selected. The usual steps in pre-processing are the handling of missing data, outliers detection and replacement, selection of relevant variables (i.e. feature selection), handling of drifting data and detection of delays between the particular variables [8]. One problem in the pre-processing step is that it requires a large amount of manual work and expert knowledge about the underlying process. The model selection, training and validation phase is one of most important in soft sensors development, requiring the correct selection of the model, so that it can correctly reproduce the target variable. The last step is soft sensor maintenance, where the goal is to maintain a good soft sensor response even in the presence of process variations, or some data change.

The DDSS is built based on empirical observations of the process, where steps (II) and (III) above are essential for a correct development, being directly related to the selection of input variables and respective time lags and to the type of selected prediction model.

Some techniques were proposed for delay selection in soft sensors context. A linear time-delay model optimized by genetic algorithm was proposed by [13] to delay selection, was used to show a good performance to predict the nitrogen oxides $NO_X$ and oxygen $O_2$ estimation in combustion operation in industrial boilers. A method based on radial basis function model (RBF) was used to select best delays using the MSE as cost function [4]. A delay selection method was proposed by [11] using mutual information method for data-driven prediction using non-linear models and was shown that mutual information may give better results when compared with a correlation analysis approach in the case analyzed.

Process engineers are often eager to find the optimal levels of process variables that make the key quality variable as close to its target as possible. Some studies have used techniques based on variance as principal component analysis (PCA) for variable selection [14], these methods are designed for linear models, so they can not be the best choice for non-linear modeling. A method to select best variables is to use PLS method [7, 5], when the model
The method was successfully applied to predict the butane has better results when compared with previous methods. The simulations show that the proposed algorithm performed with industrial process data sets are pretrained. The selection of input variables has two problems, the first one is to select the best subset, the second is to find respective time lags for each variable. Below is given the mathematical definition of the problem.

2. Input Variable and Delay Selection

The selection of input variables can be defined as the optimal subset of the whole set, so that it can correctly predict the output using a certain model. However, when it comes to complex industrial processes there is also the issue of selecting the delay of each variable, in this case, the selection of input variables has two problems, the first one is to select the best subset, the second is to find respective time lags for each variable. Below is given the mathematical definition of the problem.

2.1. IVDS Problem Statement

The IVDS problem can be described mathematically as follows. For any set of elements $A = \{a_1, \ldots, a_n\}$, define the $\nu$ operator that transforms $A$ into vector $a = \nu(A) = [a_1, \ldots, a_n]^T$. Only ordered sets will be considered in this paper. Conversely, $A = \nu^{-1}(a)$. A function $G$ receives input from variables belonging to set $U = \{u_1(t - k_1), u_2(t - k_2), \ldots, u_p(t - k_p)\}$,

$$y(t) = G(u),$$

where $u = \nu(U)$, and the set of time delays is defined as $K = \{k_1, k_2, \ldots, k_p\}$. It is assumed that $G$ can be a linear or nonlinear mapping. To estimate $G$, it is assumed that a set $X_D = \{x_1(t - d_1), x_2(t - d_2), \ldots, x_n(t - d_n)\}$ of measurement variables is available, where $D = \{d_1, d_2, \ldots, d_n\}$ is the set of time delays. It is assumed that both the most appropriate $x_i$ variables and corresponding delays $d_i$ can be selected during the IVDS design. Assume that for each variable $x_i(t - d_i)$ there is an optimal delay $d_i^*$. Then define $X_{D^*} = \{x_1(t - d_1^*), \ldots, x_n(t - d_n^*)\}$, where $D^* = \{d_1^*, \ldots, d_n^*\}$ is the set of optimal delays. It is assumed that

$$U \subseteq X_{D^*}.\quad (2)$$

The goal of IVDS is to select the set of optimal delays $D^* = \{d_1^*, \ldots, d_n^*\}$, and a subset of variables

$$X_{D^*} \subseteq X_{D^*},\quad (3)$$

that most adequately represent the information contained in the real input variables from $U$. Hence, an approximation model for $G (1)$ can be written as:

$$\hat{y}(t) = F(x_{D^*}^*; \theta),\quad (4)$$

where $F$ is a functional mapping parameterized by $\theta$, and $x_{D^*}^* = \nu(X_{D^*}^*)$.

It is possible to separate delay selection from variable selection in IVDS. In the approach proposed in this paper, delay selection is performed as the first step using maximum mutual information as criterion to obtain $X_{D^*}$. Then, in a second step, an MLP-model approach is employed to select the best variables, $X_{D^*}$. The definition of mutual information is given below, and it is shown how to select best delay for each variable.
3. Delay Selection

Industrial installations track of their systems by continuously recording the measurements of a large amount of sensors. Industrial competition leads to a need to increase the knowledge of the process for further optimization. For example, consider a process where it would be important to find out how long it takes for material to flow from one point in the process to another. Injecting tracing substances and measuring the time directly might be difficult, expensive, and interfere with the process in a negative way. Instead, it might be possible to measure two related variables, for example temperature, at the two points in the process and find out the time lag between the strongest correlations. The time lag can then give a very good indication of how long it takes for material to travel from one point to the other. Mutual information is a general correlation measure that unlike the correlation coefficient can be generalized to all kinds of probability distributions. Given an appropriate model of the distributions, this measure can potentially detect non-linear dependencies between variables. In this work mutual information is used as the basis to detect the best delay for each variable.

3.1. Discrete Mutual Information

Mutual information (MI) is a measure of dependency between variables, that takes into account the probabilistic distribution of variables. It can be calculated through entropy measurements [3]. The mutual information between two discrete random variables $X$ and $Y$ is given by:

$$I(Y;X) = H(Y) + H(X) - H(X,Y),$$

where $H(X)$ and $H(X,Y)$ are Shannon entropies [3] are given by:

$$H(X) = \sum_{i=1}^{N} -\log[P(x_i)] \cdot P(x_i),$$

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

where $N$ is the number of discrete values (bins) for each variable, and $x_i$ is the event $i$ of variable $X$, for $i = 1, \ldots, N$.

3.2. Delay Selection Procedure

The purpose of the delay selection procedure is to find $d^*_i \in D^*$ for $i = 1, \ldots, n$. The approach proposed here uses MI. For each input variable, $x_i(t - d_i)$, the MI of the input-output pair is computed for a set of admissible input delays. The chosen delay, $d^*_i$, is the one that maximizes the MI (5):

$$d^*_i = \arg\max_{d_i \in \Delta_i} \{I[x_i(t-d_i); y(t)]\}$$

where $\Delta_i$ is the set of admissible values of $d_i$ and $i = 1, \ldots, n$.

This approach is independently used for each input variable to chose the most appropriate time lag for the input variable, without considering the dependency among input variables.

4. MLPB Review

In this section the MLP-based variable selection (MLPB) developed by [1] is reviewed. This section starts with an overview about MLP model. Then, the MLPB algorithm is presented.

4.1. MLP Model

For a multilayer perceptron feed-forward neural network model $M = (V,E,w)$, let $V$ be the set of nodes, which is divided into the subset $V_I$ of input nodes, the subset $V_H$ of hidden nodes and the subset $V_O$ of output nodes. $E \subseteq V \times V$ is the set of connections, $V_H$ is partitioned into collection of sets of nodes $V_i$, where $V_i$ is the set of nodes of hidden layer $i$, $i = 1, \ldots, N_L$, and $N_L$ is the number of layers. For each connection from node $j$ to node $i$, $(i,j) \in E$, it is associated a weight $w_{ij} \in R$, and $X_{ij}$ is input $i$ of neuron $j$, $Y_j$ the output of neuron $j$ and $w_{ij}$ the weight from neuron $i$ to neuron $j$.

4.2. MLPB Algorithm

Let $X = \{x_1, \ldots, x_n\}$ be the set of input variables, and $y$ the output variable, for the variable selection problem. $X$ corresponds to the result $X_{D^*}$ of the delay selection method of Sec. 2. Assume there is available a data set $D = \{(x(t),y(t)) : t = 1, \ldots, N\}$ of measurements of input and output variables for $N$ time instants $t = 1, \ldots, N$, where $x = \nu(X) = [x_1, \ldots, x_n]^T$.

Assume that a MLP neural network, $M_X$, where the inputs are the elements of $X$, and the output is $y$, was trained with the data set of measurements $D$ to approximate $G(1)$. The variable selection procedure can be used independently of the learning method used for training. The elimination criterion used to select the less relevant input variable in the network is given by the minimal sum of inputs criterion for the nodes of the first hidden layer $V_1$ (the layer whose nodes receive inputs from variables in $X$). The MLPB algorithm performs a backward search procedure, using the following minimal weighted sum of input nodes criterion to chose a node from $V_1$:

$$r = \arg\min_{x_i \in X} \left\{\sum_{t=1}^{N} x_{ij}(t) \sum_{i \in V_1} w_{ij}\right\}$$

and pruning-off from $X$ the selected irrelevant input $x_r$, such that the variables remaining are those belonging to $X - x_r$. After the removal of the input node/variable, the remaining weights are adjusted in such a way that the net input of every node of $V_1$ remains unchanged. This corresponds to finding $\delta_{ij}$ such that the following equalities

\[ ... \]

\[ ... \]
The quantities $\delta_{ij}$ are the adjusting factors for the weights $w_{ij}$. Simple algebraic manipulations yield the following linear system of equations:

$$\sum_{i=1}^{N} \sum_{j \in X \setminus \{x_r\}} \delta_{ij} x_j(t) = \sum_{i=1}^{N} w_{ir} x_r(t), \forall i \in V_1. \quad (11)$$

Without loss of generality (nodes and variables can be re-numbered), assume that in (11) $i = 1, \ldots, P$, $j = 1, \ldots, Q$, and $r = Q + 1$. Equations (11) can be represented in vector form as

$$A\delta = b, \quad (12)$$

having an overdetermined solution, where $\delta = [\delta_{11}, \ldots, \delta_{1Q}, \ldots, \delta_{P1}, \ldots, \delta_{PQ}]^T$, $A = [a_{ij}]$ is a matrix where $a_{ij} = 0, \forall i, j$, except $a_{ik} = \sum_{j=1}^{N} x_j(t)$ for $k = (i-1)Q + j$, and $j = 1, \ldots, Q$. $b = [b_i]$ is a vector where $b_i = \sum_{r=1}^{N} w_{ir} x_r(t)$. A way to solve (12) is by means of a standard least-square method. A common way to solve (12) is to use the conjugate-gradient method called CGPCNE [2], providing a good and fast least-squares solutions.

The MLPB algorithm is described as follows:

1. (Initialization) Set $X \leftarrow "$Initial set of $n$ input variables"", ""y as corresponding output""; set $k = n$.
2. (MLP Train) Set $M_X \leftarrow "$A trained MLP neural network having as input $X$ and corresponding output $y"$.
3. (Backward Selection) repeat until $k = 0$
   (a) (Selected Variables) Set $S_k \leftarrow X$ "the subset of selected variables resulting from iteration $k$; it has cardinality $k";
   (b) (Remove Variable) Set $x_r \leftarrow "$Variable to be removed using criterion (9); $X \leftarrow X \setminus \{x_r\};$
   (c) (Update MLP model) Set $M_X \leftarrow "$Updated MLP neural network using $X$, by adjusting the remaining weights according to (10), (12), as follows: $w_{ij}^{(\text{new})} = w_{ij}^{(\text{old})} + \delta_{ij}$";
   (d) Set $k \leftarrow k - 1$.
4. (Selected variables) Set $S^*$ as the best $S_k$ subset ($k = n, \ldots, 1$) by a manual analysis of the RVTP and the MSE on a validation set;
5. Output the set $S^*$ containing the selected variables.

A drawback of the MLPB algorithm is that it fails to remove redundant variables. This can decrease the performance of the resulting model after variable selection and model learning. Moreover, when the dimension of the input space is large the intermediate MLP models often fail to represent the dynamics of the system, decreasing the performance of the MLPB variable selection algorithm.

5. Proposed Algorithms

This section presents the new algorithms proposed in this paper. The first is the MLP-based variable selection by minimal redundancy (MLPBR) algorithm (Sec. 5.1). The second is the delay selection using mutual information and MLP-based variable selection (DMLPB) algorithm (Sec. 5.2). Finally, the third is the delay selection using mutual information and MLP-based variable selection by minimal redundancy (DMLPB) algorithm (Sec. 5.3).

5.1. MLPBR Algorithm

The MLPB has some disadvantages, as discussed above. The MLPBR algorithm solves the redundancy problem, excluding redundant variables. Moreover, this reduces the dimensionality of the input space, thus increasing the algorithm performance. The criterion used to detect redundant variables is the pearson coefficient, given by:

$$\rho(x, y) = \frac{\sum_i [(x(t) - \bar{x})(y(t) - \bar{y})]}{\sqrt{\sum_i (x(t) - \bar{x})^2} \sqrt{\sum_i (y(t) - \bar{y})^2}}, \quad (13)$$

where $x(t)$ and $y(t)$ are the values of variables $x$ and $y$ at time instant $i$, i.e. are the $i$-th samples of these variables. 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 $\rho = 1$ corresponds to a positive perfect correlation among the two variables, $\rho = -1$ corresponds to a perfect negative correlation among the two variables (i.e. if one increases, the other decreases), and $\rho = 0$ means that the two variables are linearly independent.

The algorithm for removal of redundant variables works as follows. For every pair of input variables $(x_i, x_j)$, such that $\rho(x_i, x_j) > K$, the variable of the pair that has the lowest influence on the output, $y$, is removed. The degrees of influence on the output are measured by the pearson coefficients $\rho(x_i, y)$, and $\rho(x_j, y)$, respectively. $K$ is a free parameter. A typical adequate value for $K$ is 0.8. The MLPBR algorithm is described as follows:

1. (Initialization) Set $X \leftarrow "$Initial set of $n$ input variables"", ""y as corresponding output"".
2. (Detect Redundant Variables) Set $R \leftarrow "$Set of selected redundant variables to be excluded"; set $X \leftarrow X \setminus R$. 

4
3. (MLPB Algorithm) Set \( X \) and \( y \) as input of MLPB algorithm (Sec. 4.2).

4. (Output) Set \( S \) as output of MLPB algorithm (Sec. 4.2).

This algorithm is suitable for applications of variable selection, when it do not needs to know the delays for each variable.

### 5.2. DMLPB Algorithm

To improve the output prediction, the DMLPB algorithm performs a delay selection before the variable selection algorithm. The DMLPB algorithm is described as follows:

1. (Initialization) Set \( X_D \) ← “Initial set of \( n \) variables where \( D \) is the set delays to be determined, one for each variable of \( X_D \)”; and “set \( y \) as the corresponding output”;

2. (Delay Selection) Set \( X_{D*} \) ← “The best delay for each variable in \( X_D \) using (8), where \( D^* \) is the set of best delays, one for each variable of \( X_{D*} \)”;

3. (MLPB Algorithm) Set \( X_{D*} \) and \( y \) as input of MLPB algorithm (Sec. 4.2).

4. (Output) Set \( S_{D*} \) as output of MLPB algorithm (Sec. 4.2).

### 5.3. DMLPBR Algorithm

This final algorithm is a combination of the algorithms described before, it can easily defined as following:

1. (Initialization) Set \( X_D \) ← “Initial set of \( n \) variables where \( D \) is the set delays to be determined, one for each variable of \( X_D \)”; and “set \( y \) as the corresponding output”;

2. (Delay Selection) Set \( X_{D*} \) ← “The set of input variables with the best delay for each variable in \( X_D \) being selected using (8), where \( D^* \) is the set of best delays, one for each variable of \( X_{D*} \)”;

3. (Detect Redundant Variables) Set \( R_{D*} \) ← “Set of selected redundant variables to be excluded”; set \( X_{D*} \) ← \( X_{D*} \setminus R_{D*} \).

4. (MLPB Algorithm) Set \( X_{D*} \) and \( y \) as the input of MLPB algorithm (Sec. 4.2).

5. (Output) Set \( S_{D*} \) as output of MLPB algorithm (Sec. 4.2).

### 6. Evaluation Criterion

The most common indicator of quality models is the root mean square error (MSE), but the MSE does not measure the tracking precision. [10] proposed the use of relative variance tracking precision, RVTP, for soft sensors applications:

\[
RVTP = 1 - \frac{\sigma_{error}^2}{\sigma_{measurement}^2}, \quad (14)
\]

where \( \sigma_{error}^2 \) is the variance of the prediction error (difference between the model prediction and the measurement value), and \( \sigma_{measurement}^2 \) is the output measurement variance. RVTP (14) indicates the tracking precision between output and the model when the output changes. It is a measure of how precisely the SS output remains with enough precision when the value of the output changes. When RVTP is less than zero, the precision of SS is very low. The closer RVTP approaches 1, the more accurately the SS tracks the real process [10].

### 7. Experimental Results

#### 7.1. Experiment I

The first case of study consists in the prediction of butane (\( C_4 \)) concentration in the bottom flow of a debutanizer column. There are seven candidate input variables forming vector \( V = [u_1(t), u_2(t), u_3(t), u_4(t), u_5(t), u_6(t), u_7(t)] \) and an output variable \( y(t) \). This case study was introduced by [5] and an associated data set is available for download in the book website. To apply the IVDS algorithm, the whole data set of 2394 sample was divided into a training data set of 1481 points, and a validation data set of 729 points.

Table 1 presents the results of the variable selection methods. The lists of variables represent the order by which the variables were selected by each algorithm. For example, in the debutanizer results of Table 1, the second subset selected by the DMLPB algorithm is composed by \( u_5(t - 7), u_6(t) \), and the fourth best subset found by the DMLPBR algorithm is composed by \( u_5(t - 7), u_7(t), u_3(t - 5), u_2(t) \).

The four algorithms have been applied to select the best variables to predict \( C_4 \) concentration. Figs. 1 and 2 plot the values of MSE and RVTP criteria versus the
The second case study concerns the estimation of the total nitrogen at the effluent of a wastewater treatment plant (WWTP). The WWTP study was conducted using the Benchmark Simulation Model No. 2 (BSM2) [6]. BSM2 is a platform-independent WWTP simulation environment defining a plant layout, a process model, influent data, test procedures and evaluation criteria. The benchmark is evaluated for two years with acquisition data for the variables being available with a 15min sampling interval. There are 25 input variables in the data set which are candidates for the variables and delay selection problem. Table 2, and Figures 5 and 4, present the results of the delay and variables selection procedures. The results indicate that, with the DMLPBR algorithm, it is possible to reliably estimate \( T_N \) with only eight variables, \( X_{D^*} - X_{R^*} = \{u_{24}(t - 25), u_{25}(t - 22), u_7(t - 17), u_21(t), u_4(t - 35), u_{14}(t - 17), u_{22}(t - 36), u_6(t - 10)\} \). From Figs. 4 and 5 is possible to conclude that with the DMLPBR algorithm the error remains approximately constant with different numbers selected variables, but the
RVTP decreases when the number of selected variables decreases.

Fig. 6 presents the $T_N$ estimate using the model developed with the delays and variables selected by the DMLPBR algorithm. Again, a very reasonable prediction accuracy is attained using the set of variables selected by the proposed DMLPBR algorithm.

8. Conclusion

In this paper, three algorithms were presented for input delay and variable selection, two covering the field of input delay and variable selection (DMLPB and DMLPBR) and one for input variable selection (MLPBR), without perform delay selection, that is an improvement of the algorithm proposed by [1]. It was shown trough experimental results, that when the most adequate delay for each variable is selected, and redundant variables are removed, there is an important improvement in prediction accuracy. The results show the DMLPBR to be the most suitable algorithm for variable and delay selection applications. It was observed that with DMLPBR input variable and delay selection is performed in such a way that the number of required input variables can be reduced, while at the same time the output prediction accuracy is maintained. For applications involving only variable selection (but not delay selection) the MLPBR algorithms can be used. In this paper, an estimation case study concerning this later problem was not presented, but it is possible to conclude, from the graphs in the experimental results, that MLPBR is superior to the MLPB algorithm.

The methods proposed here shows to be suitable for Soft Sensor applications when the input variables for prediction is unknown and when a MLP model can accurately predict the output variable. The proposed methods have shown to be fast and reliable for the two experiments presented in this paper.

Acknowledgement

This work was supported by Mais Centro Operacional Program, financed by European Regional Development Fund (ERDF), and Agência de Inovação (AdI) under Project SinCACI/3120/2009;
References


