Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

On Variable Selection of a Nonlinear Non-parametric System with a Limited Data Set: A Stepwise Algorithm

Er-wei Bai * **Kang Li** ** **Wenxiao Zhao** ***

* *Dept. of Electrical and Computer Engineering, University of Iowa, Iowa City, Iowa 52242, er-wei-bai@uiowa.edu*
** *School of Electronics, Electrical Engineering and Computer Science, Queen's University, Belfast, UK*
*** *Academy of Mathematics and Systems Science, China*

Abstract: This paper considers a problem of identification for a high dimensional nonlinear non-parametric system when only a limited data set is available. The algorithms are proposed for this purpose which exploit the relationship between the input variables and the output and further the inter-dependence of input variables so that the importance of the input variables can be established. A key to these algorithms is the non-parametric two stage input selection algorithm.

1. INTRODUCTION

This paper considers identification of a stable scalar discrete nonlinear non-parametric system

$$y(k) = f(x_1(k), x_2(k), ..., x_p(k)) + v(k) \tag{1.1}$$

where $y(k)$ is the system output at k and $v(k)$ is an i.i.d. random noise sequence. The regressor $x(k) = (x_1(k), ..., x_p(k))$ consists of possible contributing variables. The structure of the nonlinear function f is unknown. The system (1.1) represents a large class of nonlinear systems including the well known nonlinear autoregressive moving average models with exogenous inputs (NARX) systems [9]. The goal of nonlinear non-parametric system identification is to estimate the unknown function $f(\cdot)$ based on the available data $\{y(k), x(k)\}_{k=1}^N$.

Since the structure of f is unknown, one approach towards this problem is to approximate the unknown f by some basis functions $\phi_i(x)$'s either linear in the unknown parameters $f(x) = \sum \alpha_i \phi_i(x)$ or nonlinear in the unknown parameters $f(x) = g(\alpha, \phi_1(x), \phi_2(x), ...)$ but for some known and fixed functions g. This approach includes polynomial representation, splines approximation, linearization of f, neural networks and others. The other approach is to estimate $f(\cdot)$ locally. Say if $f(x^0)$ is of interests, the value of $f(x^0)$ is estimated based on the available local data near x^0. Almost all the methods in this class are in some form of weighted local averages. The celebrated kernel and local polynomial estimators [3] as well as the direct weight optimization [2] and the stochastic approximation all belong to this class. An inherent and very serious problem with any local average approach with a high dimensional system is the curse of dimensionality.

For many practical applications, however systems are sparse in the sense that not all variables $x_i(k)$'s, $i = 1, 2, ..., p$ contribute to the output $y(k)$ or contribute little or are dependent. If these variables $x_i(k)$'s that do not contribute or are dependent can be identified and removed, the dimension involved for identification could be much smaller.

The variable selection problem has been extensively investigated in the literature in a linear setting, including some well known methods LASSO, LARS and their variants. Compressive sensing techniques are also applied for this purpose. Despite its importance, there is only scattered efforts for variable selection in a nonlinear non-parametric setting.

There are two approaches to the variable selection problem. One is the hard approach, i.e., to find variables that have absolutely no contribution to f or to the output $y(k)$ and once identified, to remove these variables from $x_1, ..., x_p$. Remaining variables are contributing ones no matter how small their contribution is. The other approach is the soft one. Even contributing, a variable can still be removed if its contribution is insignificant or marginal. Clearly, variables that do not contribute are a part of variables which contribute insignificantly or marginally. For a practical high dimensional system, a soft approach is usually preferred which makes the resultant model simpler. This is important in two senses. First, a parsimony model makes subsequent design and calculation easier. Second, a parsimony model is usually more robust against noise and uncertainty. Of course, a critical step for any soft approach is the definition of what constitutes an insignificant contribution and a way to test it. In this paper, a soft approach is adopted and in addition a reliable way to detect if a variable contributes significantly or not is developed.

Variable selections can also be characterized into two classes. The first one is the top-down approach. It works with the full p dimensional system and tries to determine which variables contribute and if so, contribute by how much. Once determined, the variables that do not contribute or contribute marginally are removed. Asymptotic analytical results are relatively speaking easy to derive for this approach. The problem is that because of working

with a full dimensional system, asymptotic results require a large data set. This is because whether a data length is long enough or not is always relative depending on the dimension of the system. For instance, the contribution of a variable x_i to f may be evaluated by the size of the partial derivative $\frac{\partial f}{\partial x_i}$. In particular, if x_i does not contribute, $\frac{\partial f}{\partial x_i} = 0$. Thus one way to determine the contribution of x_i is to estimate the size of $\frac{\partial f}{\partial x_i}$ by say the local linear estimator or other methods [1, 3, 9, 10]. Though convergent under some technical conditions, the data length needed to achieve the asymptotic convergence is very long because it works with the full dimensional system as a manifestation of the curse of dimensionality. For a problem with a limited data set and a high dimension p, the top-down approach usually does not work well. The other approach is the bottom-up approach. For a given pair of p, n, $(n < p)$, the approach tries to find n variables among all p variables that are the most dominant in terms of their contribution to f. It works upto a n dimensional system. If the best chosen n variables are still not representing f well, let $n = n + 1$ and continue the process. For a practical problem with a limited data set, the bottom-up approach is usually more effective. Though pragmatic and practically useful, theoretical results are not easy to derive compared to the top-down approach. We will follow the bottom-up approach in this paper.

Variable selections can be carried out in basically two ways, with or without the involvement of the output values. Define

$$Y = \begin{pmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{pmatrix}, \; X_i = \begin{pmatrix} x_i(1) \\ x_i(2) \\ \vdots \\ x_i(N) \end{pmatrix}, \; i = 1, 2, ..., p$$

The first one is by checking the regressor vector structure $(X_1, X_2, ..., X_p)$. Suppose $X_1, ..., X_p$ are in a subspace or lie on a low dimensional manifold. Then, there exists a known or unknown $q < p$ and some known or unknown functions g and $\bar{f}$ such that

$$(X_{i1}, X_{i2}, ..., X_{iq}) = g(X_{i,q+1}, ..., X_{i,p})$$

that implies

$$Y = f(X_1, ..., X_p) = \bar{f}(X_{i,q+1}, ..., X_{i,p}))$$

resulting in a low dimensional $(p-q)$ system. A special case is that $(X_1, ..., X_p)$ is in a linear subspace. In such a case, by checking the rank of the regressor matrix $(X_1, ..., X_p)$, one can easily determine how many regressor vectors are linear independent, say $(p - q)$. Then, it is trivial to calculate

$$(X_{i1}, X_{i2}, ..., X_{iq}) = (X_{i,q+1}, ..., X_{i,p})A$$

for a $(p - q)$ by N matrix A and this in turn results in

$$Y = f(X_1, ..., X_p) = \bar{f}(X_{i,q+1}, ..., X_{i,p}))$$

Though much more work is involved, the same idea applies when $(X_1, ..., X_p)$ lies on a low dimensional manifold. The idea is reminiscent to the unsupervised training in the artificial intelligence and machine learning literature where most of work, e.g., LLE and others project data to a lower dimensional space based on some features ignoring the output variable y and thus are not ideal for system identification of the system (1.1) where the output error is one of the major concerns. Ignoring the output values and checking the structure of the regressor matrix alone do not always solve the variable selection problem [11]. For example, if $(X_1, ..., X_p)$ are linear independent and in fact do not lie on any low dimensional manifold but f does not depend on $(X_{i1}, X_{i2}, ..., X_{iq})$ which can be removed without affecting f at all. In such a case, the output value Y plays an important role. Another example is that the regressors only approximately lie in a low dimensional subspace or on a low dimensional manifold. How good a linear subspace or low dimensional manifold approximation is needs to take the output values into the consideration.

Though different, the order determination has also been frequently used as a tool for variable selections in the literature, e.g., in a neural network setting [15] or in a NARX setting [9].

2. FORWARD/BACKWARD STEPWISE APPROACH

In this section, we propose a forward/backward stepwise selection. It starts with an 1-dimensional system by picking up the most important variable and then, the dimension increases one by one by picking up the next most important variable until the prescribed performance is met.

2.1 The minimum set of the unfalsified variables and low dimensional neighborhood

Consider the system (1.1). First, our focus is for a given pair of n, p, $(n < p)$, to find the n most important variables $x_{i_j}(k), j = 1, 2, ..., n$ among $x_1(k), x_2(k), ..., x_p(k)$. Our idea is the minimum set of the unfalsified variables. To explain easily, let us say that n variables $(x_{i_1}(k), x_{i_2}(k), ..., x_{i_n}(k))$ really contribute and the rest $p - n$ variables do not. Then the set $(x_{i_1}(k), ..., x_{i_n}(k))$ is the minimum set of the unfalsified variables, i.e., for some g

$$|f(x_1(k), ..., x_p(k)) - g(x_{i_1}(k), ..., x_{i_n}(k))| \approx 0 \qquad (2.2)$$

for all k independent of of the values of $x_j(k)$ for all $j \notin (i_1, i_2, ..., i_n)$. Any set $(x_j(k), ..., x_{\hat{n}+j-1}(k))$ for some $\hat{n} > n$ that contains $(x_{i_1}(k), x_{i_2}(k), ..., x_{i_n}(k))$ also has the property of unfalsified variables but is not the minimum set. On the other hand with $\hat{n} < n$, there does not exist any function g such that

$$|f(x_1(k), ..., x_p(k)) - g(x_{j_1}(k), x_{j_2}(k), ..., x_{j_{\hat{n}}}(k))| \approx 0$$

for all k.

The algorithm starts at one variable, i.e., to find the most important single variable $x_i(k)$ among all $x_1(k), ..., x_p(k)$. To this end, low dimensional neighborhood has to be defined for each $1 \le i \le p$. The 1-dimensional neighborhood of $x_i(k)$ is the collection of $x(j)$, $j = 1, 2, ..., N$ formally defined by

$$\{x(j) \mid (x(k) - x(j))_i = \sqrt{(x_i(k) - x_i(j))^2} \le h\} \qquad (2.3)$$

where k is fixed and $h > 0$ is the bandwidth that will be discussed later. Note that the 1-dimensional neighborhood of $x_i(k)$ is solely defined by $x_i(k)$ and any $x_l(k)$, $l \neq i$, do not play any role at all. Let those of $x(j)$ in the neighborhood of $x_i(k)$ be denoted by $x(k_1^i), x(k_2^i), ..., x(k_{l_i}^i)$ and the corresponding outputs be $y(k_1^i), y(k_2^i), ..., y(k_{l_i}^i)$,

where l_i is the number of $x(j)$ in the neighborhood of $x_i(k)$. Then, the estimated output $\hat{f}_i(x(k))$ at $x(k)$ based on the neighbors $x(k_1^i), x(k_2^i), ..., x(k_{l_i}^i)$ defined by $x_i(k)$ as in (2.3) can be calculated by the kernel estimator

$$\hat{f}_i(x(k)) = \frac{\sum_{j=1}^{l_i} K((\frac{x(k_j^i)-x(k)}{h})_i) y(k_j^i)}{\sum_{j=1}^{l_i} K((\frac{x(k_j^i)-x(k)}{h})_i)}, \ i = 1, 2, ..., p \quad (2.4)$$

where $(\frac{x(k_j^i)-x(k)}{h})_i = \sqrt{(\frac{x_i(k_j^i)-x_i(k)}{h})^2}$ and the subscript i in $\hat{f}_i$ indicates that $\hat{f}_i$ depends on only one component $x_i(k)$. The kernel function $K(\cdot)$ can be any standard multivariate kernel function with the following properties

(1) $K(\cdot) \geq 0$
(2) $K(x) = 0, \ if \ \|x\| > h$
(3) $K(\cdot)$ is symmetric with respect to the origin.
(4) $\int K(x)dx = 1$

and $K_Q(x) = \frac{1}{|Q|} K(Q^{-1}x)$, where $Q = hI$ is a bandwidth matrix that controls the size of the neighborhood by adjusting the bandwidth $h > 0$. $K(x(k) - x(j)) = 0$ if $x(k)$ is not in the h-neighborhood of $x(j)$ or $\|x(k) - x(j)\| > h$. One such example is the the Biweight kernel function [3]

$$K(x) = \begin{cases} c(1 - (x_1^2 + x_2^2 + ... + x_p^2))^2 & \|x\| \leq 1 \\ 0 & \|x\| > 1 \end{cases}$$

where $c > 0$ is a scaling constant so that

$$\int K(x_1, ..., x_p) dx_1 ... dx_p = 1.$$

Then the residual sum of squares (RSS)

$$RSS_i = \frac{1}{N} \sum_{k=1}^{N} (y(k) - \hat{f}_i(x(k)))^2 \quad (2.5)$$

could be used to determine which $x_i(k)$ are the most important variable in terms of the smallest output error. We may say $x_{i_1^*}(k)$ is the most important if

$$i_1^* = arg \min_{1 \leq i \leq p} RSS_i \quad (2.6)$$

where RSS_i is given by (2.5).

To find the second most important variable once i_1^* is obtained as defined in (2.6), choose $i \in (1, 2, ..., p)/i_1^*$, the corresponding $x_i(k)$ and define the neighborhood of $(x_{i_1^*}(k), x_i(k))$ similarly as in (2.3) by

$$\{x(j) \mid (x(k) - x(j))_{i_1^*, i} = \sqrt{(x_{i_1^*}(k) - x_{i_1^*}(j))^2 + (x_i(k) - x_i(j))^2} \leq h, \ i \neq i_1^*\} \quad (2.7)$$

Say $\{x(k_1^{i_1^*,i}), x(k_2^{i_1^*,i}), ..., x(k_{l_{i_1^*,i}}^{i_1^*,i})$ are in the neighborhood. Calculate the estimated output $\hat{f}_{i_1^*,i}(x(k))$ and RSS for 2 variables respectively,

$$\hat{f}_{i_1^*,i}(x(k)) = \frac{\sum_{j=1}^{l_{i_1^*,i}} K((\frac{x(k_j^{i_1^*,i})-x(k)}{h})_{i_1^*,i}) y(k_j^{i_1^*,i})}{\sum_{j=1}^{l_{i_1^*,i}} K((\frac{x(k_j^{i_1^*,i})-x(k)}{h})_{i_1^*,i})} \quad (2.8)$$

$$(\frac{x(k_j^{i_1^*,i}) - x(k)}{h})_{i_1^*,i} = \sqrt{(\frac{x_{i_1^*}(k_j^{i_1^*,i}) - x_{i_1^*}(k)}{h})^2 + (\frac{x_i(k_j^{i_1^*,i}) - x_i(k)}{h})^2}$$

$$RSS_{i_1^*,i} = \frac{1}{N} \sum_{k=1}^{N} (\hat{f}_{i_1^*,i}(x(k)) - y(k))^2 \quad (2.9)$$

Let

$$i_2^* = arg \min_i RSS_{i_1^*,i} \quad (2.10)$$

and the corresponding $x_{i_2^*}(k)$ is the second most important variable. The process continues for 3,4,...,n variables. This finishes the Forward selection and then the backward selection starts which exchanges one variable at a time among the chosen n variables with the remaining $p - n$ variables to achieve the largest improvement. More precisely, let $i_1^*, i_2^*, ..., i_n^*$ and the corresponding variables $x_{i_1^*}(k), x_{i_2^*}(k), ..., x_{i_n^*}(k)$ be chosen at the end of the forward selection. Exchange i_1^* with one of the remaining $i \in (1, 2, ..., p)/(i_1^*, i_2^*, ..., i_n^*)$ if further improvement can be made. Define the neighborhood of $x(k)$ similarly based on $(i, i_2^*, ..., i_n^*)$

$$\{x(j) \mid (x(k) - x(j))_{i, i_2^*, ..., i_n^*} = \sqrt{(x_i(k) - x_i(j))^2 + (x_{i_2^*}(k) - x_{i_2^*}(j))^2 + ... + (x_{i_n^*}(k) - x_{i_n^*}(j))^2} \leq h\} \quad (2.11)$$

Let the neighbors be $\{x(k_j^{i,i_2^*,...,i_n^*}), \ j = 1, ..., l_{i,i_2^*,...,i_n^*}\}$. Now, compute the estimated output and the corresponding RSS respectively,

$$\hat{f}_{i,i_2^*,...,i_n^*}(x(k)) = \frac{\sum_{j=1}^{l_{i,i_2^*,...,i_n^*}} K((\frac{x(k_j^{i,i_2^*,...,i_n^*})-x(k)}{h})_{i,i_2^*,...,i_n^*}) \cdot y(k_j^{i,i_2^*,...,i_n^*})}{\sum_{j=1}^{l_{i,i_2^*,...,i_n^*}} K((\frac{x(k_j^{i,i_2^*,...,i_n^*})-x(k)}{h})_{i,i_2^*,...,i_n^*})} \quad (2.12)$$

$$RSS_{i,i_2^*,...,i_n^*} = \frac{1}{N} \sum_{k=1}^{N} (y(k) - \hat{f}_{i,i_2^*,...,i_n^*}(x(k)))^2 \quad (2.13)$$

Let i^* be

$$i^* = arg \min_{i \in (1,2,...,p)/(i_1^*,i_2^*,...,i_n^*)} RSS_{i,i_2^*,...,i_n^*} \quad (2.14)$$

If $RSS_{i^*,i_2^*,...,i_n^*} < RSS_{i_1^*,i_2^*,...,i_n^*}$, replace $x_{i_1^*}(k)$ by $x_{i^*}(k)$ and i_1^* by i^*. $x_{i^*}(k)$ is the variable that achieves the largest improvement. Further shift $(i^*, i_2^*, ..., i_n^*)$ and the corresponding $(x_{i^*}, x_{i_2^*}, ..., x_{i_n^*})$ to $(i_2^*, ..., i_n^*, i^*)$ and $(..., x_{i_n^*}, x_{i^*})$ respectively and rename $(i_2^*, ..., i_n^*, i^*)$ and $(x_{i_2^*}, ..., x_{i_n^*}, x_{i^*})$ as $(i_1^*, ..., i_n^*)$ and $(x_{i_1^*}, ..., x_{i_n^*})$. If there is no improvement, keep i_1^*. Now, the process is repeated one variable at a time until no improvement can be achieved or some stopping criterion is met. The newly renamed variables $x_{i_1^*}(k), x_{i_2^*}(k), ..., x_{i_n^*}(k)$ are considered to be the most important n variables among p variables $x_1(k), ..., x_p(k)$ around.

2.2 The algorithm

We are now in a position to summarize the nonlinear non-parametric kernel based Forward/backward stepwise approach to variable selection.

The Forward/backward stepwise algorithm for variable selection.

Consider the system (1.1), the data $\{y(k), x(k)\}_{k=1}^{N}$ and an $n < p$.

Step 1: Determine the bandwidth $h > 0$ (will be discussed later).

Step 2: Forward selection.

Step 2.1: Start one variable selection.

(1) Choose one variable $x_i(k)$, $1 \leq i \leq p$, determine the neighborhood as in (2.3), calculate the estimated output $\hat{f}_i(x(k))$ as in (2.4) and RSS as in (2.5).
(2) Determine the optimal i_1^* and $x_{i_1^*}(k)$ as in (2.6).

Step 2.2: Start two variables selection.

(1) Choose one more variable $x_i(k)$ from $(1, 2, ..., p)/i_1^*$, determine the neighborhood as in (2.7), calculate the estimated output $\hat{f}_{i_1^*,i}(x(k))$ and RSS as in (2.8) and (2.9) respectively.
(2) Determine the optimal (i_1^*, i_2^*) and $(x_{i_1^*}(k), x_{i_2^*}(k))$ as in (2.10).

Step 2.3: Continue the process for 3,4,..., n variables.

Step 3: Backward selection.

(1) Exchange $x_{i_1^*}(k)$ with the most significant one from the remaining variables $(1, 2, ..., p)/(i_1^*, ..., i_n^*)$ that contributes more to the error reduction than $x_{i_1^*}(k)$. Determine the neighborhood, calculate the estimated outputs and RSS as in (2.11), (2.12) and (2.13) respectively.
(2) Find i^* from (2.14) and compare the corresponding error with previously selected n variables. If the error is smaller, replace $x_{i_1^*}$ with i^*, shift and rename $(i^*, i_2^*, ..., i_n^*)$ and the corresponding $(x_{i^*}, x_{i_2^*}, ..., x_{i_n^*})$ as described in the algorithm.
(3) Continue the process until all n variables have been checked. If error can not be further reduced, go to Step 4, otherwise go back to Step 3.

Step 4: The finally selected $x_{i_1^*}(k), x_{i_2^*}(k), ..., x_{i_n^*}(k)$ are the most important n variables.

The idea of the algorithm is that RSS (2.13) monotonically decreases when the number of variables chosen is less than the actual number of contributing variables and also increases. When the number of variables chosen is equal or large than the actual number of contributing variables, RSS of (2.13) is flattened and does not decrease. as shown in the following theorem [3].

Theorem 2.1. Consider the system (1.1). Assume that the unknown f is continuously differentiable and the noise is an iid zero mean finite variance random sequence. Assume that $h \to 0$, $h^n N \to \infty$ as $N \to \infty$. Also assume that only $n* \leq p$ variables $x_{i_1^*}(k), ..., x_{i_{n^*}^*}(k)$ contribute in (1.1) and the chosen variable set $(x_{i_1}(k), ..., x_{i_n}(k))$ contains $(x_{i_1^*}(k), ..., x_{i_{n^*}^*}(k))$. Further assume that the regressor $\{x(k)\}$ is geometrically ergodic and α-mixing with some $\{\alpha(k)\}$ satisfying $\alpha(k) \leq c\rho^k$ for some $c > 0$ and $0 < \rho < 1$. Then for each k, $\hat{f}_{i_1^*,i_2^*,...,i_n^*}(x(k)) - f(x(k))$ goes to zero asymptotically in probability as $N \to \infty$.

We would like to comment that the ergodic and mixing condition listed above amounts to some stability conditions of the system (1.1). A sufficient condition is that the system (1.1) has asymptotical fading memory or is exponentially and incrementally input-to-state stable. Interested readers may refer [3] for details.

2.3 Statistical hypothesis test of the dimension

When the number of variables chosen is equal or large than the actual number of contributing variables, (2.13) is flattened. Thus, the corresponding RSS as defined by (2.13) is an indication of how appropriate the dimension n is. By checking the knee is very efficient but is more or less ad hoc. To make the dimension n determination more rigorous, statistical methods can be applied for testing if the chosen n is appropriate. The most well known is the Box-Pierce test [4]. Let $\hat{f}_{i_1^*,...,i_n^*}(x(k))$ be the output estimate at $x(k)$ when the correct variables are selected. Then, the residual $r(k) = y(k) - \hat{f}_{i_1^*,...,i_n^*}(x(k))$ is almost white. Let

$$\gamma(j) = E(r(k) - Er(k))(r(k-j) - Er(k)),$$
$$\rho(j) = \gamma(j)/\gamma(0)$$

denote the lag-j autocovariance and the lag-j correlation coefficient of $r(k)$, where E is the expectation operator. If $r(k)$ is white,

$$\rho(1) = ... = \rho(p-1) = 0.$$

To check, the Box-Pierce test says that for a large N the sampled version of

$$N \sum_{j=1}^{p-1} \rho(j)^2$$

follows a Chi-square distribution with $(p-1)$ degree of freedom if $r(k)$ is white. This provides a framework for statistical hypothesis tests. Set the null hypothesis

$$H_0\text{: the residual } r(k) \text{ is white.} \tag{2.15}$$

Then, the null hypothesis H_0 can be tested based on $N \sum_{j=1}^{p-1} \rho(j)^2$ and the $\chi^2(p-1)$ distribution. If H_0 is accepted, $r(k)$ is considered to be white and the dimension n is accepted. To test the hypothesis, we calculate $N \sum_{j=1}^{p-1} \rho(j)^2$ based on the residuals $r(k)$. Let the threshold d be taken from the $\chi_\alpha^2(p-1)$ distribution with α being the level of significance, i.e., the probability to reject H_0 though H_0 is true. The hypothesis H_0 is accepted if $N \sum_{j=1}^{p-1} \rho(j)^2 \leq d$. If $N \sum_{j=1}^{p-1} \rho(j)^2 > d$, H_0 is rejected and we conclude that the dimension n is not high enough.

There is a problem however. What we really test by the Box-Pierce is not if the residual $r(k)$ is white or not but if $r(i)$ and $r(j)$ are uncorrelated or not. The Box-Pierce test [4] works well for this purpose in linear identification but may not work for nonlinear identification because the residual $r(k)$ may exhibit some nonlinear dependence which is usually the case in nonlinear identification. In such a case, the Box-Pierce test does not work well. In fact, the Box-Pierce test could provide some misleading conclusions [8, 16]. Therefore, a modified Box-Pierce test is needed in the presence of nonlinear dependence of $r(k)$. To this end, along the direction of [8, 16] and our early work, the Box-Pierce test is modified here. Let $r(k)$ be the residual.

Denote the sampled mean, the lag-j autocovariance, the lag-j correlation coefficient by respectively

$$\widehat{\mu} = \frac{1}{N}\sum_{k=1}^{N} r(k),$$

$$\widehat{\gamma}(j) = \frac{1}{N-j}\sum_{k=j+1}^{N}(r(k)-\widehat{\mu})(r(k-j)-\widehat{\mu}),\ \widehat{\rho}(j) = \widehat{\gamma}(j)/\widehat{\gamma}(0)$$

It was shown in [8, 16] that for large N,

$$N(\widehat{\rho}(1),...,\widehat{\rho}(p-1))V^{-1}\begin{pmatrix}\widehat{\rho}(1)\\ \vdots \\ \widehat{\rho}(p-1)\end{pmatrix} \tag{2.16}$$

follows a chi-square distribution with (p-1) degree of freedom when H_0 is true, where

$$V = C/\gamma(0)^2 = \begin{pmatrix} c_{11} & \cdots & c_{1,p-1}\\ \vdots & \ddots & \vdots \\ c_{p-1,1} & \cdots & c_{p-1,p-1}\end{pmatrix}/\gamma(0)^2$$

$$c_{ij} = \sum_{l=-\infty}^{\infty} E(r(k)-\mu)(r(k-i)-\mu)(r(k+l)-\mu)(r(k+l-j)-\mu)$$

$$i,j = 1,...,p-1$$

with μ being the mean value of $r(k)$. The difference is that the identity matrix is used in the Box-Pierce test [4] while in the modified Box-Pierce test, the actual autocovariance matrix V is used. The modified Box-Pierce test is more reliable for large N even the residual $r(k)$ exhibits nonlinear dependence. For our application, however, the actual autocovariance matrix V is unknown and has to be estimated. To this end, let

$$\hat{W}(k) = \begin{pmatrix}(r(k)-\widehat{\mu})(r(k-1)-\widehat{\mu})\\ (r(k)-\widehat{\mu})(r(k-2)-\widehat{\mu})\\ \vdots \\ (r(k)-\widehat{\mu})(r(k-p+1)-\widehat{\mu})\end{pmatrix}$$

and $K(x)$ be the Biweight kernel function as defined before. Now, define the estimate $\widehat{V}$ of V by $\widehat{C}/\widehat{\gamma}(0)^2$ with

$$\begin{aligned}\widehat{C} &= \sum_{j=-l}^{l} K(\frac{j}{l})\frac{1}{N-p+1-|j|}\sum_{k}\hat{W}(k)\hat{W}(k-j)'\\ &= \sum_{j=-l}^{0} K(\frac{j}{l})\frac{1}{N-p+1+j}\sum_{k=n}^{N+j}\hat{W}(k)\hat{W}(k-j)'\\ &+\sum_{j=1}^{l} K(\frac{j}{l})\frac{1}{N-p+1-j}\sum_{k=n+j}^{N}\hat{W}(k)\hat{W}(k-j)'\end{aligned}$$

where l is the bandwidth of the kernel $K(\cdot)$. Note all the variables $\widehat{\mu}$, $\widehat{\rho}(j)$, $\hat{W}(k)$ and $\widehat{\gamma}(j)$ are computable. Now, we show that the modified Box-Pierce test is still valid if the actual autocovariance matrix V is replaced by its estimate as discussed above,

Theorem 2.2. Consider the residual $r(k)$ and the corresponding $\widehat{\mu}$, $\widehat{\gamma}(j)$, $\widehat{\rho}(j)$ and $\widehat{V} = \widehat{C}/\widehat{\gamma}(0)^2$. Then,

$$Q_{p-1} = N(\widehat{\rho}(1),...,\widehat{\rho}(p-1))\widehat{V}^{-1}\begin{pmatrix}\widehat{\rho}(1)\\ \vdots \\ \widehat{\rho}(p-1)\end{pmatrix} \tag{2.17}$$

converges, in distribution as $N \to \infty$, to a chi-square distribution with (p-1) degree of freedom if the residual $r(k)$ is white, provided that

$$l \to \infty,\ l/N \to 0,\ as\ N \to \infty$$

Now, the hypothesis test of the dimension n can be stated as follows.

(1) Set the null hypothesis as in (2.15).
(2) Calculate Q_{p-1} as in (2.17).
(3) Let the threshold d be taken from $\chi^2_\alpha(p-1)$ distribution where α is the level of significance.
(4) The null hypothesis is accepted if $Q_{p-1} \le d$ or rejected if $Q_{p-1} > d$.

3. NUMERICAL SIMULATION

Clearly, the proposed algorithm depends on the bandwidth $h > 0$ in the kernel calculation which is subjective. There exists a large literature on the choice of the bandwidth in kernel estimation [3] which is beyond the scope of this paper. In this paper, we adopt the m-fold cross validation method [3] to make the choice of h automatic from the obtained data set. The exact steps can be found in the reference [3]. In all simulations the bandwidth $h > 0$ is calculated from the 5-fold cross validation and all the estimates are constructed by the Biweight kernels.

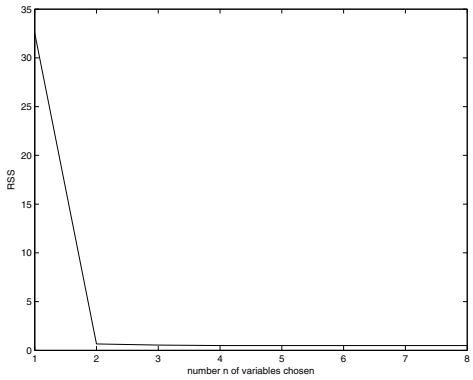

Fig. 1. RSS vs the number n of variables chosen.

Consider a classical example [9]

$$y(k) = 10sin(x_1(k)x_2(k)) + 20(x_3(k)-0.5)^2 + 10x_4(k) + 5x_5(k)$$
$$+x_6(k)x_7(k) + x_7(k)^2 + 5cos(x_6(k)x_8(k)) + exp(-|x_8(k)|) + 0.5\eta(k)$$

$k = 1, 2, ..., 500$, where $\eta(k)$ is i.i.d. Gaussian noise of zero mean and unit variance. It was assumed that $x_3(k), x_5(k)$ are independent and uniformly in $[-1, 1]$ and the rests are dependent variables

$$\begin{aligned}x_4(k) &= x_3(k)\cdot x_5(k) + 0.1\cdot\eta(k)\\ x_1(k) &= x_3(k)^2\cdot x_5(k) + 0.1\cdot\eta(k)\\ x_2(k) &= x_3(k)\cdot x_5(k)^2 + 0.1\cdot\eta(k)\\ x_6(k) &= x_1(k) - x_4(k) + 0.1\cdot\eta(k)\\ x_7(k) &= x_3(k)^3\cdot x_5(k) + 0.1\cdot\eta(k)\\ x_8(k) &= x_2(k)\cdot x_5(k) + 0.1\cdot\eta(k)\end{aligned} \tag{3.18}$$

so the 8-dimensional regressors $x_1(\cdot), ..., x_8(\cdot)$ are not exactly but approximately on an 2-dimensional manifold. Only 500 data points were available for this 8-dimensional system and $h = 0.2$ was chosen by the 5-fold cross validation.

First the residual sum of squares RSS vs the number n of the variables chosen were calculated as shown in Figure 1 by the Forward/Backward selection algorithm. Clearly from the figure, when $n = 2$ the error is small and flatten which suggests that only two variables are dominant or equivalently the regressors are close enough to a 2-dimensional manifold so that an 2-dimensional representation of the original 8-dimensional system is satisfactory in terms of RSS. We then calculated $Q_{p-1} = Q_7 = 8.2273$ of (2.17) for the hypothesis test. Let $\alpha = 0.05$ which implies $\chi^2_{0.05}(7) = 14.07 > 8.2273$. Thus the dimension 2 was also accepted by the hypothesis test. We then applied the Forward/Backward algorithm again by fixing $n = 2$ to find which 2 variables are the dominant ones. In 100 Monte Carlo runs, the variables (x_3, x_5) were picked 100 times. To verify the claim, a fresh validation data $k = 1, ..., 40$

$$x_3(k) = 0.9 * sin(\frac{2\pi k}{20}),\ x_5(k) = 0.9 * cos(\frac{2\pi k}{20}),$$

were generated and other variables $x_i(k)$'s, $i \neq 3, 5$, remained the same relation as in (3.18). Two estimated outputs $\hat{y}(k) = \hat{f}(x(k)),\ k = 1, 2, ..., 40$ were calculated by the training data $N = 500$ as described before based on the Biweight kernels. The first one was based on identification by considering Example 1 directly as an 8-dimensional system. The second one was based on identification of an 2-dimensional nonlinear non-parametric system $y(k) = g(x_3(k), x_5(k))$ for some unknown g as the results of the proposed variable selection algorithms. Figure 2 shows the actual output (solid line) and the estimated outputs by identifying the 8-dimensional Example 1 directly with GoF=0.6940 (dash) and by identifying a low dimension system with GoF=0.9205 (dash-dot) where GoF stands for the goodness-of-fits is defined as

$$GoF = (1 - \sqrt{\frac{\sum(y(k) - \hat{y}(k))^2}{\sum(y(k) - \frac{1}{40}\sum y(k))^2}}) \times 100\%$$

Clearly, by taking advantages of the proposed variable selection algorithms, a good estimate of the system can be obtained even with a very limited data length $N = 500$ for an 8-dimensional nonlinear system.

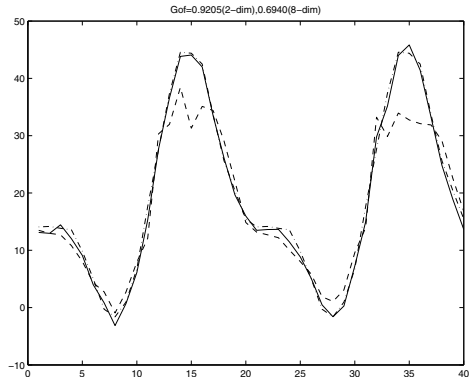

Fig. 2. Outputs and GoFs of different identification results.

REFERENCES

[1] Bai, E.W., K. Li, W.Zhao and W. Xu (2013) "Kernel based approaches to local nonlinear non-parametric variable selection", Technical Report, University of Iowa, submitted for publication

[2] Bai, E.W. and Y. Liu, (2007) "Recursive direct weight optimization in nonlinear system identification: a minimal probability approach", *IEEE Trans on Automatic Control*, **52**, pp.1218-1231

[3] Bai, E.W. (2010) "Non-Parametric Nonlinear System Identification: An Asymptotic Minimum Mean Squared Error Estimator", *IEEE Trans on Automatic Control*, **55**, pp.1615-1626

[4] Box, G.E.P. and D. Pierce, (1970), "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models", *J of the American Statistical Association*, **65**, pp. 1509-1526.

[5] Hong, X, Mitchell, S. Chen, C. Harris, K. Li and G.W Irwin (2008), "Model selection approaches for nonlinear system identification:a review", *Int. J of System Science*, **39**, pp.925-949

[6] Li, K. and J. Peng (2007), "Neuro input selection-a fast model based approach", *Neurocomputing*, **70**, pp.762-769.

[7] Li, K, J. Peng and EW Bai (2006), "A two-stage algorithm for identification of nonlinear dynamic systems", *Automatica*, **42**, pp.1187-1196

[8] Lobato, I.N., J. Nankervis and N. Savin, (2002), "Testing for zero autocorrelation in the presence of statistical dependence", *Econometric Theory*, **18**, pp. 730-743

[9] Mao, K and S.A. Billings (2006), "Variable selection in nonlinear system modeling", *Mechanical Systems and Signal processing*, **13**, pp.351-366

[10] Mosci, R., L. Rosasco, M. Santoro, A. Verri and S. Silvia (2012) "Is There Sparsity Beyond Additive Models?", *Proceedings of the SYSID*, 2012, Brussels

[11] Ohlsson, H, J. Roll, T. Glad and L. Ljung (2007) "Using manifold learning for nonlinear system identification", *Proc of the 7th IFAC Symposium on Nonlinear Control Systems*

[12] Peduzzi, P (1980) "A stepwise variable selection procedure for nonlinear regression methods", *Biometrics* **36** pp.510-516

[13] Pillonetto, G., M Quang and A. Chiuso (2011), "A New Kernel-Based Approach for Nonlinear System Identification,", *IEEE Trans on Automatic Control*, **56**, no.12, pp.2825-2840

[14] Soderstrom, T and P. Stoica (1989), System Identification, Prentice Hall, New York

[15] Su, S. and F. Yang (2002) "On the dynamical modeling with neural fuzzy networks", *IEEE Trans. Neural Netw.*, **13**, pp. 1548-1553

[16] Velasco, C. and I. Lobato (2004) "A simple and general test for white noise", *Proc. Economet. Soc. Latin Amer. Meetings*, Santiago, Chile, pp.112113