Proceedings of Engineering & Technology (PET)
pp. 43-48

2nd International Conference on Automation,Control, Engineering and Computer Science (ACECS-2015)
Proceedings of Engineering & Technology (PET)

Nonlinear Principal Component Analysis Based Input Trainning Network

IT-NET Toolbox for Use with MATLAB

Wafa Bougheloum, Messaoud Bouakkaz, Tarek Ait Izem, Mohamed Faouzi Harkat, Messaoud Djeghaba
Faculty of Engineering Sciences P.O.Box 12, 23000
Badji-Mokhtar, Annaba University
Annaba, Algeria
bougheloumwafa@yahoo.com, messaoud.bouakkaz@univ-annaba.dz, ait_izem@hotmail.com, mohamed.harkat@univ-annaba.dz, djeghaba@univ-annaba.org

***Abstract*—Principal component analysis (PCA) is a standard statistical technique frequently employed in the analysis of large highly correlated data-set. PCA is a linear technique, which limits its use for real processes, considering the highly nonlinear systems frequently encountered in the industry. Several attempts to extend linear PCA to cover nonlinear data sets have been made, mainly using neural networks, as the five layers neural network approach or the input training network (IT-NET). In this paper, we present a new toolbox for an input training network based nonlinear principal component analysis with Matlab, named INTR toolbox.**

Keywords—Principal component analysis; Nonlinear Principal component analysis ; Input Trainning Neural Network

I. INTRODUCTION

Even if the idea of principal component analysis appeared about 100 years ago, principal component analysis research and applications are still very hot subjects. The Principal component analysis (PCA) is a multidimensional technique which analyses a picture of data in which the observations are described by several dependent variables to inter-correlation quantitative. Its objective is to extract the important information of the table, to represent it as a set of new orthogonal variables called principal component, and to display the pattern of similarity of the observations and of the variables as points in maps.

The PCA is unable to find a compact representation describing nonlinear relationships between different variables, against the nonlinear PCA treats each type of relations between linear and non-linear variables, and several extensions of the PCA in the nonlinear case have been developed. Hastie and Stuetzle [1] introduced the first approach, based on principal curves, which is a nonparametric generalization of PCA in the nonlinear case. However, it does not allow for a representation model. To overcome this problem, an auto-associative neural network was used; Kramer [2] proposed a principal component analysis based on an artificial neural network (ANN) with five layers to extract linear and non- linear relationships between variables. Dong and MacAvoy [3] have developed a method combining principal curves and neural networks in three layers. Tan and Mavrovouniotis [4] proposed a method based on the concept of learning inputs of neural network with three layers (IT network).

In this paper, we propose a development of a Toolbox for an Input Training Network- Based Nonlinear Principal Component Analysis, named INTR Toolbox. This toolbox serves as an excellent educational resource by offering a user friendly environment. A brief comparison is also given between the five layers neural network NLPCA approach and the IT-NET approach.

II. PRINCIPAL COMPONENT ANALYSIS

The principal component analysis is a statistical technique mainly used for data compression and information retrieval. The philosophy behind this approach is to reduce the dimensionality of the original data by forming a new set of latent variables which are linear combination of the original data without loss of essential information.

Let X represents matrix of data which includes the n measurements of the m variables in a system's normal operation. PCA determines the optimum transformation of the data matrix X:

$$T = XP \qquad and \qquad X = TP^{\mathrm{T}} \tag{1}$$

Where $T \in \Re^{n \times m}$ is the principal component matrix, and the matrix $P \in \Re^{m \times m}$ contains the principal vectors which are the eigenvectors associated with the eigenvalues λ_i of the covariance matrix (or correlation matrix) Σ of X:

ISSN: 2356-5608

$$\Sigma = \mathrm{P}\Lambda\mathrm{P}^T \quad (2)$$

Where Λ is a diagonal matrix that contains in its diagonal the eigenvalues of Σ sorted in decreasing order. Once the number of components ($l < m$) determined [5], the eigenvectors matrix P can be partitioned in the form:

$$P = (\hat{P}\tilde{P}) \quad (3)$$

Note that the smallest eigenvalues indicate the existence of linear or quasi-linear relations between components of the data.

The transformation matrix $\hat{P} \in \Re^{m \times l}$ is generated by choosing l eigenvectors or columns of P corresponding to l principal eigenvalues. Matrix $\hat{P}$ transforms the space of the measured variables into the reduced dimension space.

$$\hat{T} = X\hat{P} \quad (4)$$

$$\hat{X} = X\hat{P}\hat{P}^T \quad (5)$$

$$\hat{C}^{(l)} = \hat{P}\hat{P}^T \quad (6)$$

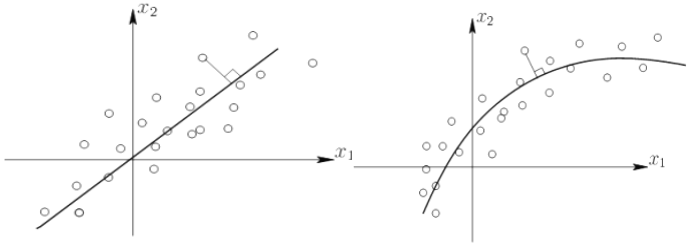
The residual space is spanned by the matrix $\tilde{P}$ generated by choosing the last (m-l) eigenvectors or columns of P.

$$\tilde{X} = X - \hat{X} = X(I - \hat{C}^{(l)}) \quad (7)$$

The determination of the PCA model can be resumed by an eigendecomposition of the covariance matrix Σ and the determination of the number (l) of components to be retained.

III. NON LINEAR COMPONENT ANALYSIS

Nonlinear principal component analysis (NLPCA) is a novel technique for multivariate data analysis, similar to the well-known method of principal component analysis. NLPCA, like PCA, is used to identify and remove correlations among problem variables as an aid to dimensionality reduction, visualization, and exploratory data analysis. While PCA identifies only linear correlations between variables, NLPCA uncovers both linear and nonlinear correlations, without restriction on the character of the nonlinearities present in the data. Fig.1 represents linear and nonlinear PCA model respectively.



(A) (B)

Figure1. Linear PCA (A) and Nonlinear PCA(B)

To better understand the problem of NLPCA and relate it with the linear model, we consider Fig.2 which shows the general principle of PCA model, whether the linear or the nonlinear case. The global model is composed of two sub-models: a sub-model for data compression, which stands for dimension reduction from m to l dimensions space, and a decompression sub-model which performs the inverse process, that is, the estimation of the data.

As in the linear case the two sub-models are characterized by the orthogonal matrix of the eigenvectors of the correlation matrix of data: $\hat{P}$ and the overall model is given by the projection matrix $\hat{C}_l$ defined by $\hat{C}_l = \hat{P}\hat{P}^T$.

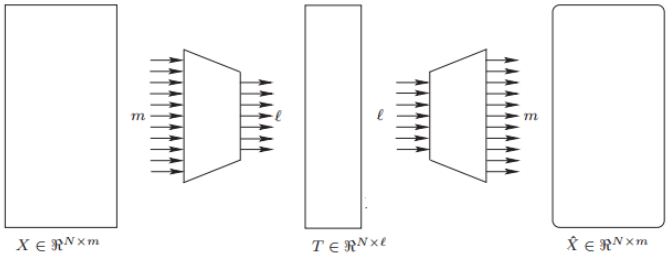


Figure.2 Basic principle of the PCA model

In the nonlinear case, the goal is to find two non-linear functions F and G. The projection of data in the principal space, i.e. nonlinear principal components T, are obtained by the compression mode via the function G. In this case, we can write:

$$t = G(x) \quad (8)$$

Where x and t are the lines of X and T respectively.

The decompression model provides an estimate $\hat{x}$ of x from nonlinear components t:

$$\hat{x} = F(t) \quad (9)$$

Thus, the data matrix X containing m variables can be expressed as a function of the first nonlinear components l as:

$$X = \hat{X} + E = F(T) + E \quad (10)$$

Where $T = [T_1, ..., T_l]$ is the matrix of nonlinear principal component $T=G(X)$, and E is the matrix of residuals. The problem is to identify the functions of nonlinear projections G and F. Therefore the following cost function is minimized:

$$\min \sum_{k=1}^{N} \left\| x(k) - \hat{x}(k) \right\|^2 = \min \sum_{k=1}^{N} \left\| x(k) - F\left(G\left(x(k)\right)\right) \right\|^2 \quad (11)$$

Where x (k) is the row of X and $\hat{x}$ (k) is the estimation given by the NLPCA model.

IV. INPUT TRAINING NEURAL NETWORK

Input training neural network approach was first introduced by Tan and Mavrovouniotis [4], it consist of analyzing nonlinear principal components based on the concept of input training network (IT-NET), the architecture of the later is illustrated in the fig.3. The network has three layers with one hidden layer, where the output layer is composed of *m* neurons corresponding to the number of variables of the data *x*, and the input layer contains *t* neurons corresponding to the number of nonlinear principal component.

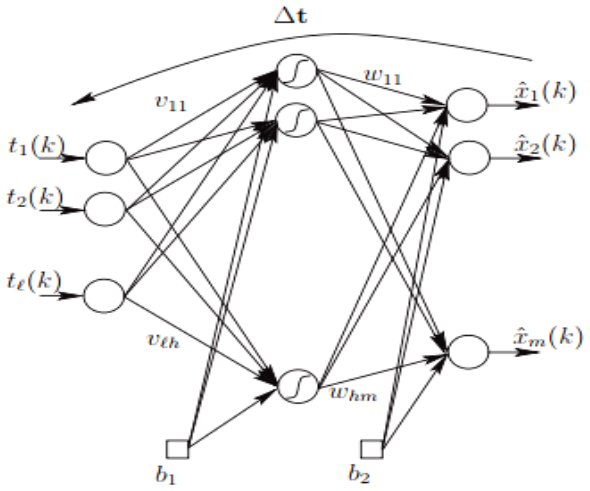


Figure.3 Network with learning inputs IT

Nonlinear principal component analysis (NLPCA) based on input training neural network can effectively extract the nonlinear principal components (PCs) from process variables, but the number of PCs cannot be decided by training network, and the order cannot be distinguished. To remedy to these defects, a hierarchical input training neural network is proposed and a nonlinear PCA based on this type of network is presented, this results in finding ordered nonlinear PCs and determining their number according to the prediction error of the data process based on PCs.

V. INPUT TRAINING NETWORK TOOLBOX

Before introducing our toolbox for the IT-NET based nonlinear principal component analysis, a brief preview of previous works in the same matter is to be considered. A nonlinear PCA toolbox for MATLAB was developed by Matthias Scholz [6] where the Nonlinear PCA is achieved by using a neural network with an auto-associative architecture. Such auto-associative neural network is a multi-layer perceptron that performs an identity mapping, meaning that the output of the network is required to be identical to the input. However, in the middle of the network is a layer that works as a bottleneck in which a reduction of the dimension of the data is enforced. This bottleneck-layer provides the desired component values as shown in Fig.4.

Instead of driving an auto-associative network of five layers, the training input method uses only a part of this network composed of three layers (decompression subnet). Such subnet is interesting and can be done by extending the back-propagation algorithm, since the error function is defined. Another difference, between the input training network and an ordinary multi-layer network, is that the inputs of IT-NET are not known because they represent the principal components sought. Therefore, in the learning phase must be adjusted not only the internal parameters of the network but also the input values by minimizing the network output error.

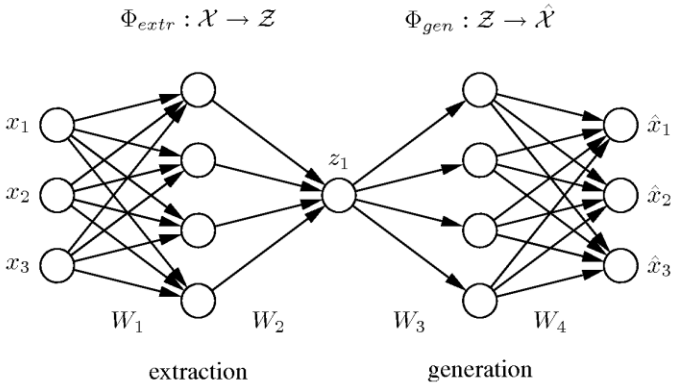


Figure.4 Auto-associative neural network

The purpose of this paper is to present our developed IT-NET toolbox for use with MATLAB. This toolbox serves as an excellent educational resource for student laboratories and engineers, where they can easily modify the existing functions and develop new functions to answer their own needs. A graphic user interface is also included and presented to further simplify the use of the toolbox.

A. *Description of Some Functions*

Several functions performing different tasks are included in the IT-NET toolbox. A brief preview of some functions is given by the following:

1. Function "mainitnet":

```
function [u,xi,W] = mainitnet(xdata)

% function [u,xi,Z] = mainitnet(xdata)
% Calculates 1 NLPCA (or PCA) mode IT-Net
% If there are convergence problems, one could adjust some of the
% parameters marked by '>>>>' (maxiterations, niterchk, overallexpand
% & options) in messaitnet.m directly, though this is usu. not necessary.
getparam
%%%% scales xdata  if xscaling >=0.
if xscaling >=0; %((
[xdata,xmean,xstd] = nondimen(xdata',xscaling);
xdata = xdata';
end %))
```

As its name states, it is the main function of the IT-NET toolbox which calculates the NLPCA (or PCA) model using the input training network structure. Thus, providing the user with

the principal components and the estimation of the introduced data matrix. Note that this function operates using several other functions included in the toolbox.

2. Function "getparam":

```
function getparam
% function getparam
% Parameters for the IT-NET model

global  iprint isave linear nensemble testfrac segmentlength ...
  overfit_tol earlystop_tol xscaling penalty maxiter initRand ...
  initwt_radius options  n l m nbottle iter Jscale xmean xstd...
  ntrain xtrain utrain xitrain ntest xtest utest xitest MSEx ...
  ens_accept ens_MSEx ens_W ens_utrain ens_xitrain ens_utest ens_xitest

%-----------------------------------------------------------------
% Parameters are classified into 3 categories, '%**' means user normally
%  has to change the value to fit the problem. '%++' means user may want
% to change the value, and '%-- means user rarely has to change the value.
```

This function contains several parameters of the Input training network structure, which are necessary for the main function to run. Parameters can be modified from within the script or via the user interface, as will be further explained.

3. Functions "dimen"&"nondimen":

```
function [x] = dimen(xnondimen,xmean,xstd,index)
% function [x] = dimen(xnondimen,xmean,xstd,index)
% After using
% function [xnondimen,xmean,xstd] = nondimen(x,index)
% to standardize a variable x (column vector)(or a matrix of column
% vectors), by removing the mean (xmean) and dividing by the standard

function [xnondimen,xmean,xstd] = nondimen(x,index,xmean0,xstd0)
% function [xnondimen,xmean,xstd] = nondimen(x,index,xmean0,xstd0)
% standardize a variable x (column vector), (or a matrix of column
% vectors), by removing the mean (xmean) and dividing by the standard
% deviation (xstd), to yield the nondimen. variables (xnondimen).
```

Data standardization is a necessary step for the determination of a PCA model, whether in linear or nonlinear case. The two functions presented here are provided for this purpose.

4. Functions "costfnus"&"costfnut":

```
function [J] = costfnut(W);
% function [J] = costfn(W);
% cost function

global  iprint isave linear nensemble testfrac segmentlength ...
  overfit_tol earlystop_tol xscaling penalty maxiter initRand ...
  initwt_radius options  n l m nbottle iter Jscale xmean xstd...
  ntrain xtrain utrain xitrain ntest xtest utest xitest MSEx ...
  ens_accept ens_MSEx ens_W ens_utrain ens_xitrain ens_utest ens_xitest
```

These are cost function, included in the toolbox, and are mainly used in the training and testing of the input training network

B. IT-NET User Interface

The graphical interface has been developed under MATLAB. Depending on the user's knowledge about MATLAB, either the m-files or the graphical interface can be used. To initialize the graphical interface, the current directory of MATLAB should be changed to the directory where the graphical interface is installed. Typing the command 'INTR' in the MATLAB command window executes the graphical interface. When the graphical interface starts, the information about the location of the required m-files is added automatically into the MATLAB paths. Executing the graphical interface file results in displaying the interface window on the screen of the computer. The main screen of our toolbox is presented in Fig. 5.

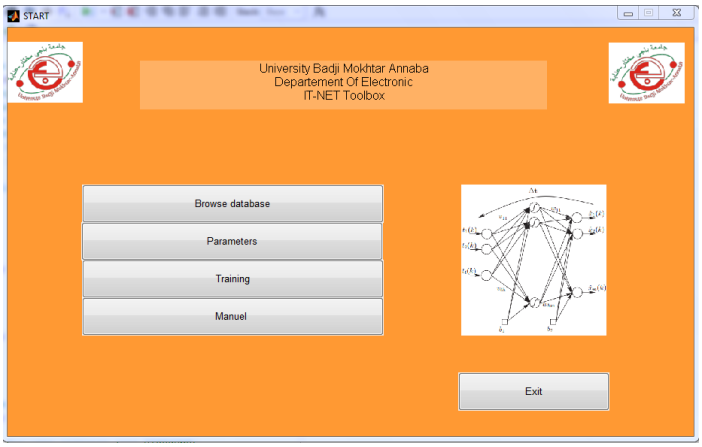


Figure.5 IT-NET User Interface's Main screen

The graphical interface is composed of four panels: "Browse database", "Parameters", "Training" and "Manual". In the following, the different parts of the user interface are explained.

1. Browsing database:

The modeling procedure using the IT-Net toolbox requires a valid set of data. To load the data into the graphical interface, select "Browse database" button, chose the location of the data, and pick a file of *.mat type as shown in Fig.6. We should precise that the graphical interface can only handle MATLAB files, i.e., files with 'mat' extension, and variables such as vectors and matrices, being double arrays.

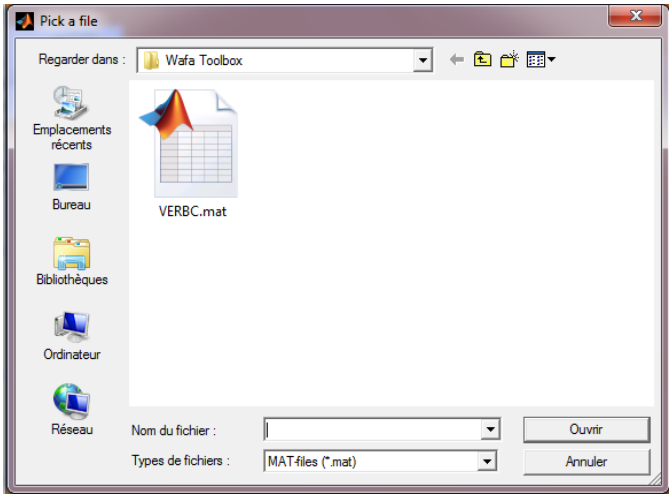


Figure.6 Browsing the database

2. Introducing Parameters:

After loading the database, the next step is to introduce the required parameters by selecting "parameters" button, another graphical interface appears as it is presented in Fig.7, where we can introduce and save the desired parameters.

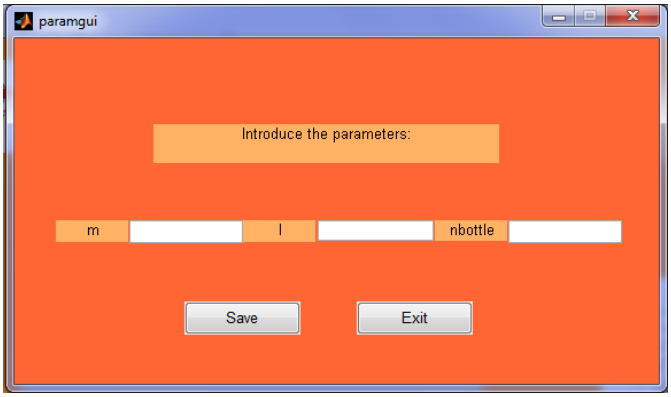


Figure.7 Introducing the parameters

3. Training:

After loading the database and introducing needed parameters, we select 'training' button, to run the main IT-NET function. Thus, resulting in the principal components and the estimation given by input training network. Several plots are obtained which tend to illustrate the effectiveness of the input training network in extracting the important information of nonlinear data compared to other neural network based approaches. An example is given by Fig.8, which represents a dual plot of two estimations; using auto-associative neural network based NLPCA, and IT-Net Based NLPCA.

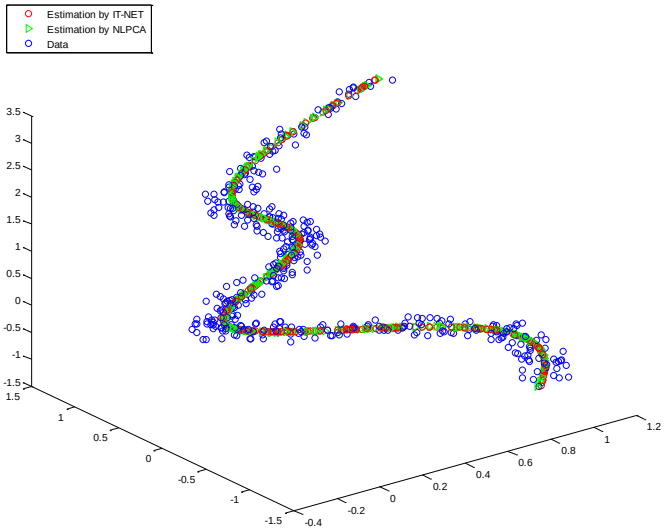


Figure.8 Data estimation by IT-NET and auto-associative ANN approaches

4. Manual

Finally, a manual is included in order to help the user understand all the toolbox functions for further simplicity of use.

VI. Conclusion

Principal component analysis is often used for its simplicity and its ability to capture the linear relationships between process variables and has different application in several fields of engineering. However, this method shows limitations to treat industrial data which generally are nonlinear. Several nonlinear principal component analysis methods based on neural networks have been proposed. Namely, the five layers auto-associative neural network PCA, and the Input training based PCA.

In this paper, we introduced our input training network based NLPCA toolbox offering different functions and abilities for an accurate modeling using IT-NET PCA. It is also brought graphically to facilitate the use of (M-File) functions and to provide a user friendly environment. This work is a part of a diagnosis framework, which is to be further completed with several fault detection and isolation tools for nonlinear data.

References

[1] Hastie T . and Stuetzle W. (1989). Principal curves.Journal of the American Statistical Association, vol.84, pp. 502-516.

[2] Kramer M. A. (1991). Nonlinear principal component analysis using auto-associative neural networks.AIChE Journal, vol. 37, pp. 233-243

[3] Dong D. and McAvoy T . J. (1996). Nonlinear principalcomponent analysis - based on principal curves andneural networks. Computers and Chemical Engineering, vol. 20, pp. 65-78.

[4] Tan S, Mavrovouniotis M. L," Reduction data dimentionality through optimizing neural network inputs,". AIChE Journal, vol. 41, N° 6, 1995 pp. 1471-148

[5] Harkat M. F., Mourot G., Ragot J., "Différentes méthodes de localisation de défauts basées sur les derniéres composantes principales ",Conférence Internationale Francophone d'Automatique CIFA,Nantes-France,2002

[6] Matthias Scholz, Martin Fraunholz, and Joachim Selbig. In Principal Manifolds for Data Visualization and Dimension Reduction, edited by Alexander N. Gorban, Balázs Kégl, Donald C. Wunsch, and Andrei Zinovyev. Volume 58 of LNCSE, pages 44-67. Springer Berlin Heidelberg, 2007