In Proceedings of IEEE/IAFE Concerence on Computational Inteligence for Financial Engineering, New York City, 1999.

Modelling Multivariate Data by Neuro-Fuzzy Systems

Jianwei Zhang and Alois Knoll
Faculty of Technology, University of Bielefeld, 33501 Bielefeld, Germany
E-mail: zhang | knoll @ techfak.uni-bielefeld.de

Abstract

This paper proposes an approach for solving multivariate modelling problems with neuro-fuzzy systems. Instead of using selected input variables, statistical indices are extracted to feed the fuzzy controller. The original input space is transformed into an eigenspace. If a sequence of training data are sampled in a local context, a small number of eigenvectors which possess larger eigenvalues provide a good summary of all the original variables. Fuzzy controllers can be trained for mapping the input projection in the eigenspace to the outputs. Implementations with the prediction of time series validate the concept.

1 Multivariate Problems in Modelling

For efficiently modelling complex data, it is desirable that not only the extracted model should approximate the training data as precisely as possible and without overfitting, but also the model be human-understandable, e.g. interpretable with fuzzy linguistic rules. However, it is well-known that general fuzzy rule descriptions of systems with a large number of input variables suffer from the problem of the "curse of dimensionality". In many real applications, it is difficult to identify the correct influential factors and reduce their number to the minimum. A general solution to build fuzzy models is not only interesting from a theoretical point of view, but also practical meaningful since the application of fuzzy systems could then be extended to a wider range of complex modelling/prediction problems.

In the recent literature on neuro-fuzzy modelling and machine learning, some standard benchmark problems with "middle"-dimensional input spaces are frequently discussed and simulation results are presented. As training data set for nonlinear system identification, the Box-Jenkins gas furnace data [2] is often studied and compared. The furnace input is the gas flow rate $x(t)$, the output $y(t)$ is the CO_2 concentration. At least 10 candidate inputs are considered: $x(t-6), x(t-5), \cdots, x(t-1), y(t-1), \cdots, y(t-4)$. If all of them are used, building a fuzzy controller means to solve a 10-input-1-output problem. If each input is defined by 5 linguistic terms, this would result in a fuzzy rule system of about 10 million rules. The modelling and prediction of financial markets are based on much more influential factors which are hard to differentiate since they are inter-related.

2 Existing Solutions

Two main methods examining the multivariate problem are "input selection" and "hierarchy".

Input Selection One implemented approach is "input selection", [4] and [3], which is in principle an experimental method to find the most important input variables among a large number of them. There are two obvious problems with such a procedure. First, all the other less influential inputs are discarded, which means an information loss for the controller. Secondly, the combinational number of lower-dimensional fuzzy controllers for a system with thousands or even millions of inputs is still too large to enumerate and to evaluate.

Hierarchy The solution with hierarchical structure assumes that the input information can be classified into groups, see [5] for an example. Within each group the inputs determine an intermediate variable, they can be decoupled from inputs of other groups. To realise such a grouping, there exists no general automatic approach but heuristics based on fusion of physical sensors.

If a fuzzy controller is only observed at each moment in time, the only input information to the controller comes from the whole set or a selected subset of the input variables. This kind of information is called *horizontal index*. Both "input selection" and hierarchical structure consider only the horizontal indices. They presume that inputs are independent and give no priority or importance of the selected input variables.

3 The System Concept

3.1 The Neuro-Fuzzy Model

Depending on how "local" the measuring data are and, therefore, how similar the observed input patterns appear, a more or less small number of eigenvectors can provide a sufficient summary of the state of all input variables. Our experimental results show under the most diverse conditions that it is very likely that three or four eigenvectors provide all information indices of the original input space necessary for the control and prediction task. Moreover, in the case of very high input dimensions, an effective dimension reduction can be achieved by principal component analysis (PCA). This step is illustrated in the left part of Fig. 1.

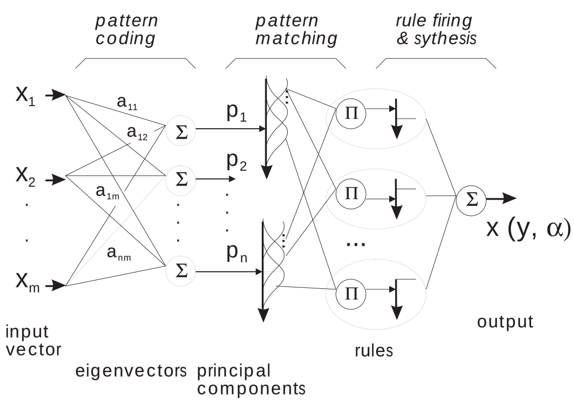


Figure 1: The task based mapping can be interpreted as a neuro-fuzzy model.

Eigenvectors can be partitioned by covering them with linguistic terms (see the right part of Fig. 1). In the following implementations, fuzzy controllers constructed according to the B-spline model are used [7]. This model provides an ideal implementation of CMAC proposed by Albus [1]. We define linguistic terms for input variables with B-spline basis functions and for output variables with singletons. Such a method requires fewer parameters than other set functions such as trapezoid, Gaussian function, etc. The output computation is very simple and the interpolation process is transparent. Through comparative studies, B-spline model generally achieves better approximation capabilities and rapid convergence than the other fuzzy models.

3.2 Dimension Reduction via PCA

Let us assume k sample input vectors $\vec{x^1}, \ldots, \vec{x^k}$ with $\vec{x^i} = (x_1^i, \ldots, x_m^i)$ originating from a pattern-generating process. The PCA can be applied to them as follows:

First the (approximate) mean value $\vec{\mu}$ and the covariance matrix $\mathbf{Q}$ of these vectors are computed according to

$$\vec{\mu} = \frac{1}{k}\sum_{i=1}^{k}\vec{x^i}$$

$$\mathbf{Q} = \frac{1}{k}\sum_{i=1}^{k}(\vec{x^i}-\vec{\mu})(\vec{x^i}-\vec{\mu})^T$$

The eigenvectors and eigenvalues can then be computed by solving

$$\lambda_i \vec{a}_i = \mathbf{Q}\vec{a}_i$$

where λ_i are the m eigenvalues and $\vec{a}_i$ are the m-dimensional eigenvectors of $\mathbf{Q}$. Since $\mathbf{Q}$ is positive definite all eigenvalues are also positive. Extracting the most significant structural information from the set of input vectors $\vec{x^i}$ is equal to isolating the first n ($n < m$) eigenvectors $\vec{a}_i$ with the largest corresponding eigenvalues λ_i. If we now define a transformation matrix

$$\mathbf{A} = (\vec{a}_1 \ldots \vec{a}_n)^T$$

we can reduce the dimension of the $\vec{x^i}$ by

$$\vec{p^i} = \mathbf{A}\cdot\vec{x^i}; \qquad dim(\vec{p^i}) = n$$

The dimension n should be determined depending on the discrimination accuracy needed for further processing steps vs. the computational complexity that can be afforded.

3.3 Off-line and On-line Phases

The working systems implements two phases: off-line training and on-line evaluation. In the off-line phase, a sequence of training input patterns and their corresponding outputs are used. In the on-line phase the input pattern is transformed into the eigenspace and is then processed by the fuzzy controller. The controller output is the system prediction (Fig. 2).

4 Implementations

In the following implementations, linguistic terms of the eigenvectors are defined by B-spline basis function of

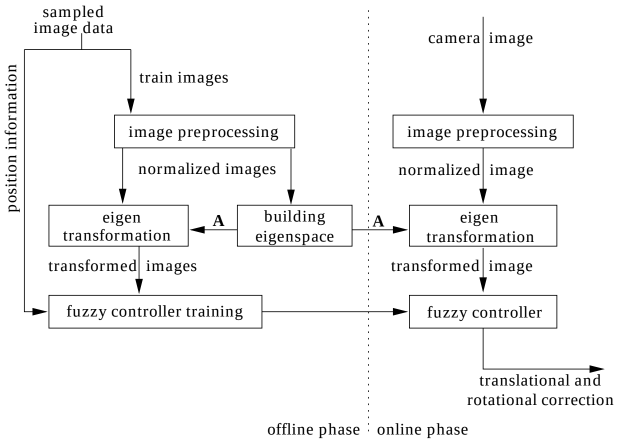


Figure 2: The training and the application of the PCA neuro-fuzzy controller.

order three. The output value of each rule is represented as a fuzzy singleton which is called control vertex in the B-spline fuzzy controller. The control vertices are adaptively determined by minimising the "Root Mean Squared Error" (as in [4]) for the supervised learning.

4.1 Prediction with Box-Jenkins Data

296 data in form of $(x(t), y(t))$ are first transformed into the form $((x(t-6), x(t-5), \ldots, x(t), y(t-1), \ldots, y(t-4)), y(t))$. The computed eigenvectors are shown in Table 3(a), the eigenvalues of each eigenvectors are depicted in Fig. 3(b). The projection of the data into the eigenspaces constructed by the first two and first three eigenvectors can be found in Fig. 4(a) and (b).

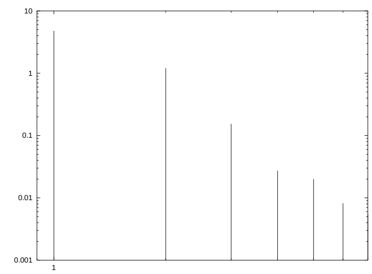
The first, second, third and fourth eigenvector are defined by 10, 7, 7 and 5 linguistic terms respectively. The RMS (also MS) training errors achieved with the first three eigenvectors are listed in Tab. 1.

	RMS (MS) Error		
epochs	1	2	3
100	0.73 (0.533)	0.22 (0.048)	0.25 (0.063)
1000	0.71 (0.504)	0.19 (0.036)	0.20 (0.04)
10000	0.71 (0.504)	0.19 (0.036)	0.17 (0.029)

Table 1: RMS (MS) training error by using 1, 2, and 3 first eigenvectors if all 296 data are used for training.

4.2 Comparative Results

In [6], Table 2 was summarised as the comparison results for solving the Box-Jenkins gas furnace data. By comparing Tab. 1 and Tab. 2, it can be seen that as expected, the training error achieved with the first two eigenvectors is less than that achieved with all the above models,



(a) Eigenvalues

	p_1	p_2	p_3	p_4	p_5
$x(t-6)$	-0.158	0.074	0.239	0.354	0.602
$x(t-5)$	-0.149	0.204	0.233	-0.015	0.291
$x(t-4)$	-0.134	0.310	0.081	-0.382	0.115
$x(t-3)$	-0.117	0.372	-0.164	-0.419	0.288
$x(t-2)$	-0.099	0.386	-0.413	-0.044	0.228
$x(t-1)$	-0.084	0.361	-0.582	0.531	-0.222
$y(t-1)$	0.477	-0.334	-0.295	0.136	0.523
$y(t-2)$	0.487	-0.015	-0.236	-0.263	0.141
$y(t-3)$	0.480	0.278	0.053	-0.225	-0.248
$y(t-4)$	0.457	0.501	0.448	0.362	-0.017

(b) Eigenvectors

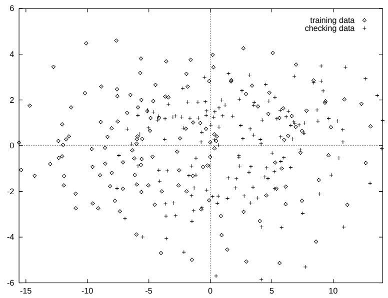
Figure 3: The important eigenvectors and eigenvalues of the Box-Jenkins data.

and with the first three or four eigenvectors, the error can be still reduced significantly. The RMS error is also less than that achieved by the ANFIS model with input selection in [4]. Fig. 5 shows the result achieved by using four eigenvectors.

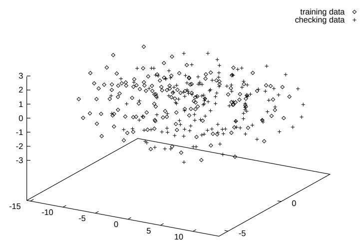
Model	Input	Rule No.	MS Error
Tong's	$x(t-1), y(t-4)$	19	0.469
Pedrycz's	$x(t-1), y(t-4)$	81	0.320
Xu/Lu's	$x(t-1), y(t-3)$	25	0.328
Chiu's TSK 2	$x(t-1), y(t-3)$	3	0.146
Chiu's TSK 3	$x(t-1), x(t-3),$ $y(t-3)$	3	0.072
[6] GA-fuzzy	$x(t-1), y(t-4)$	25	0.257

Table 2: Comparison of different models derived using the Box and Jenkins gas furnace data, excerpted from [6].

In this way, the dimension of the local perceptual space is reduced to a manipulable size of a subspace. If the eigenvalue of each selected eigenvector, noted as p_j ($j = 1, \ldots, n$), is covered with B-spline basis functions, noted as $X^j_{i_j,k_j}$, the rule can be written in the form:



(a) 2D projection



(b) 3D projection

Figure 4: Distribution of the data in the 2D and 3D eigenspaces.

IF (p_1 IS $X^1_{i_1,k_1}$) and ... and (p_n IS $X^n_{i_n,k_n}$)
(INPUT IS $PATTERN_{i_1 i_2 \ldots i_n}$)
THEN ($output$ is $Y_{i_1 i_2 \ldots i_n}$)

Each rule corresponds to a supporting point for the interpolation in the eigenspace, see Fig. 6. Our experiment showed that with a few eigenvectors, a correction of the robot hand can be attained.

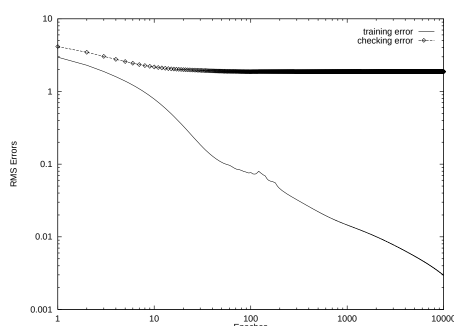


Figure 5: RMS errors of modelling the Box-Jenkins data with four eigenvectors. 148 data are used for training and 148 data for checking.

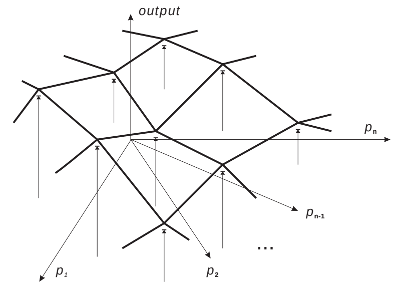


Figure 6: The application phase is an interpolation process if the control vertices have been learned.

5 Discussion

The main advantage of the proposed approach to "input selection" is that less information is lost after the dimension reduction for problems with correlated input training data. Even if the training data are not correlated at all, the projection in the eigenspace provides also information which input variables have larger variance. These variables can be good candidates for the inputs to be selected. Therefore, no "trial-comparison-select" procedure is necessary.

To generally deal with the high-dimensional input space, the solution based on the low-dimensional fuzzy controllers would need the partition of the complete high-dimensional input data set into clusters, within which the data are correlated to a large degree. Such a partition would be intrinsically fuzzy, since there are no crisp boundary between two continuous "situations". A "behaviour arbiter" coordinates multiple simultaneously active local controllers to achieve a high-level task and can be realised with a set of meta-rules like: "IF $Situation\_Evaluation$ IS for_C_i THEN Apply Controller C_i."

References

[1] J. S. Albus. A new approach to manipulator control: The Cerebellar Model Articulation Contorller (CMAC). *Transactions of ASME, Journal of Dynamic Systems Measurement and Control*, 97:220–227, 1975.

[2] G. E. P. Box and G. M. Jenkins. *Time series analysis*. Holden Day, San Francisco, 1970.

[3] S. L. Chiu. Selecting input variables for fuzzy models. *Journal of Intelligent and Fuzzy Systems*, 4:243–256, 1996.

[4] J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Prentice Hall, 1997.

[5] V. Lacrose and A. Tilti. Fusion and hierarchy can help fuzzy logic controller designers. In *IEEE International Conference on Fuzzy Systems, Barcelona*, 1997.

[6] S. Lotvonen, S. Kivikunnas, and E. Juuso. Tuning of a fuzzy system with genetic algorithms and linguistic equations. In *Proceedings of Fourth European Congress on Intelligent Techniques and Soft Computing, Aachen*, 1997.

[7] J. Zhang and A. Knoll. Constructing fuzzy controllers with B-spline models - principles and applications. *International Journal of Intelligent Systems*, 13(2/3):257–285, Feb./Mar. 1998.