8-1-1992

A Decision Boundary Method for Embedding Selection

W. Hsu
Purdue University School of Electrical Engineering

L. S. Hsu
Purdue University School of Electrical Engineering

M. F. Tenorio
Purdue University School of Electrical Engineering

A Decision Boundary Method for Embedding Selection

W. Hsu
L. S. Hsu
M. F. Tenorio

TR-EE 92-52
December 1992

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907-1285

A Decision Boundary Method for Embedding Selection

W. Hsu L. S. Hsu M. F. Tenorio

Parallel Distributed Structures Laboratory
Department of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

Abetract

Previously, we have presented a method for embedding selection based on cluster analysis. In this paper, we described an embedding selection method based on a feature reduction transformation matrix. This method extracts features that are important for maintaining decision boundaries in the supervised clusters. Experimentally, we demonstrate that our method allow accurate prediction of the Mackay-Glass chaotic time series. Three important properties of the feature reduction transformation are proved in this paper.

Keywords: Feature Reduction, Embedding Selection, Decision Boundary.

1 Introduction

The choice of an embedding scheme is an important step in the modeling and prediction of any chaotic dynamical systems. The modeling and prediction of chaotic systems has attracted much recent attention due to the discovery of the presence of chaos in many interesting phenomenal previously thought to be random. Examples are these systems include the economic systems[1, 2], weather[3] and a number of physiological processes[4].

The two step to chaotic time series prediction are the feature extraction and th e pattern learning steps. When the chaotic time series assumption can be made, the feature extraction step is equivalent to specifying an embedding scheme.

Specifying an embedding scheme is equivalent to identifying the set of features necessary to characterize the system. In physics, this process is sometimes referred to as state-space reconstruction. A large body of theoretical work has been done in this area. A comprehensive summary can be found in [5]. The most cited work among neural network researchers dealing with chaotic dynamical system is perhaps the work by Takens[6]. Takens showed that a chaotic time series $x(t)$ can be predicted T step in the future by using only m number of equally spaced past samples of the chaotic time series itself as follows :

$$x(t+\mathrm{T}) = \mathcal{F}\{x(t), x(t-\mathrm{A}), x(t-2\Delta), \ldots, x(t-(\mathrm{m}-1)\Delta)\} \quad (1)$$

where $\mathcal{F}$ is nonlinear but continuous under the suitable assumptions[6]. Taken's theorem does not, however, provide a way of constructing $\mathcal{F}$. An embedding scheme for a chaotic time series is given by the 3-tuple

$$\Pi = [m, \Delta, T]. \quad (2)$$

Equation 1 says that a chaotic time series $x(t)$ can be predicted T time step in advance using only m past samples of $x(t)$ spaced A distance apart.

Previously, we have presented a method for embedding selection based on cluster analysis[7]. In this paper, we described a decision boundary method and its application to the embedding selection problem. The decision boundary method extracts features that are important in preserving some given decision boundaries. The embedding obtained are a set of features which. are linear

combinations of the original features. This method is different from principal component analysis[8] which extracts features that maximize the fidelity of representation of the training samples. The embedding so obtained are different from the usual embedding because in the usual embedding subset of the original features are selected but in our method, linear combinations of original features are obtained.

The concept of a decision boundary is first suggested to us by Lee and Landgrebe[9]. Lee and Landgrebe[9] *estimote* their decision boundaries by examining the classification of various input data by the fully ***trained*** neural network. The resulting feature set obtained by their procedure can then be used to train another smaller neural network. Our method does not require this duplication of effort because the decision boundary for the network can be determined analytically.

Our method is related to the Fisher's method[10] in the following way. For simplicity, let us assume we are given date points in 2-d Euclidean space. Assume that clusters are created by some algorithms. Fisher method finds a scaling factor that when multiplied to the original data sets, maximizes the inter cluster and minimize the intra-cluster distances. Fisher's method takes into consideration all data points. Using a different approach, our method was based on the concept of a decision boundary. Our method finds scaling factor that preserves the boundary of each cluster by concentrating on "pushing" data points near the boundary to their respective cluster centers. Consequently, data points that are far away from the boundary are moved only slightly or not at all. Thus, the emphasis of this method is on preserving the given boundaries. Since the points on the boundary are those that are more likely to be confused by the predictor, the boundary method represents a more direct method to minimizing the number of data points that are likely to be misclassified.

The organization of the paper is as follows. In Section 3, several definitions are made. In Section 4, the feature extraction procedure is described. In Section 5, three important properties of this procedure are proved. In Section 6, experimental results of prediction the benchmark Mackey-Glass chaotic time series are presented.

2 The Supervised Clustering Network

2.1 The Input to the Network

Given a time series

$$x_1, x_2, \ldots, x_i, \ldots \tag{3}$$

The following steps are taken to construct a set of delayed vectors.

1. Determine the region to be explored by assigning the maximum values of T, A and m. Let these be T_{max}, A_{max} and m_{max} respectively. We define $L = m_{max} A_{max} + T_{max}$.
2. Decide on the size of the training set. Let it be n.
3. Form the following delayed vectors:

$$\begin{array}{cccccccccc}
(z_1^1 & z_2^1 & \ldots & z_L^1 & y_1) & = & (x_1 & x_2 & \ldots & x_L & x_{L+1}) \\
(z_1^2 & z_2^2 & \ldots & z_L^2 & y_2) & = & (x_2 & x_3 & \ldots & x_{L+1} & x_{L+2}) \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\
(z_1^p & z_2^p & \ldots & z_L^p & y_p) & = & (x_p & x_{p+1} & \ldots & x_{L+p-1} & x_{L+p}) \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\
(z_1^n & z_2^n & \ldots & z_L^n & y_n) & = & (x_n & x_{n+1} & \ldots & x_{L+n-1} & x_{L+n})
\end{array}$$

4. These delayed vectors $[z^p, y_p]$ will be used as input vectors to the network to be described in the next subsection.

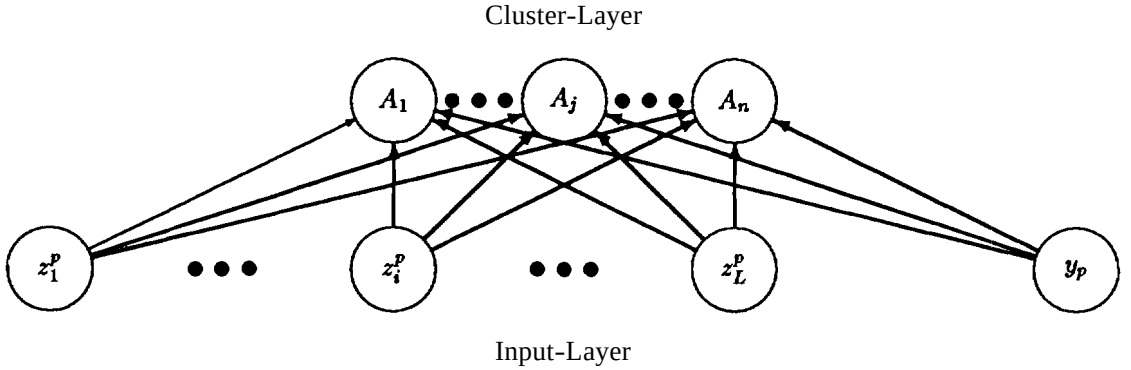

Figure 1: The **SupNet** Architecture

2.2 Network Architecture

The supervised clustering network **SupNet**[11] is a neural network that classifies a given set of n delayed vectors $\mathbf{z}^p$ into N clusters, according to their corresponding values of y_p. Its architecture is shown in Figure 1.

The network consists of 2 layers. The first layer is the input layer. It consists of $L+1$ nodes. The first L nodes represent the components of the delayed vector $\mathbf{z}$. The last node represents the value of $\mathbf{y}$.

The second layer is the cluster layer. Its size is determined dynamically by the learning algorithm described in the next subsection. The number of nodes corresponds to the number of clusters needed to classify the values of y to within a given accuracy ϵ_y.

The weights connecting a given cluster node c to the input nodes form the components of the weight vector $\mathbf{W}^c$. The values of these weight vectors are determined by the learning algorithm which will be described in the next subsection.

When input vector $[\mathbf{z}^p, y_p]$ is presented, the activation at node c is defined as

$$A_c = (y_p - W^c_{L+1})^2. \tag{4}$$

2.3 The Learning Algorithm

The learning is done in two stages. During the first stage, the state vector $\mathbf{z}^p$ is taken to be the zero vector and only the value of y_p is presented. We follow the algorithm used in **ClusNet**[12, 13] to determine the $(L+1)$-th component of the weight vectors for all the clusters. The first L components of $\mathbf{W}$ remain at zero.

During the second stage, the n input vectors are presented one at a time. Let us assume that when input vector $[\mathbf{z}^p, y_p]$ is presented, the c-th cluster node has the lowest activation among other cluster nodes. We say that the c-th node is the winning node and the first L components of its weight vector is updated to:

$$W^c_i = \frac{1}{n_c}\{(n_c - 1)(W^c_i + z^p_i)\}, \quad 1 \le i \le L \tag{5}$$

where n_c is the number of vectors belonging to cluster c, after the new vector $\mathbf{z}^p$ has been added.

When all the n input vectors have been presented, the weight vectors $\mathbf{W}$ are all known.

In the above network, the delayed vectors $\mathbf{z}^p$ are clustered according to their corresponding values of y_p. Two delayed vectors, $\mathbf{z}^i$ and $\mathbf{z}^j$, which are separated by a large Euclidean distance may be

grouped together into a single cluster because their corresponding values of y's are "close" to one another. The proposed feature selection algorithm will discriminate within each cluster against the components of any $\mathbf{z}^i$ and $\mathbf{z}^j$ that have very different values and favor those components that have similar values.

3 Preliminary Definitions

3.1 The Decision Boundary Feature Matrix

Consider the boundary between cluster c and cluster d. It is defined by

$$\{\mathbf{z}|h(\mathbf{z}) = 0\}, \tag{6}$$

where

$$h(\mathbf{z}) = (\mathbf{z} - \mathbf{W}^c)^2 - (\mathbf{z} - \mathbf{W}^d)^2. \tag{7}$$

The normal vector to the decision boundary at $\mathbf{z}$ is given by

$$\mathbf{N}(c,d) = \nabla h(\mathbf{z}). \tag{8}$$

Therefore,

$$\begin{aligned} \mathbf{N}(c,d) &= 2[(\mathbf{z} - \mathbf{W}^c) - (\mathbf{z} - \mathbf{W}^d)] & (9)\\ &= 2(\mathbf{W}^d - \mathbf{W}^c) & (10) \end{aligned}$$

Following Lee and Landgrebe[9],

Definition 3.1 The Decision Boundary Feature Matrix $\mathcal{D}$ *is defined by*

$$\mathcal{D} = \sum_{c=1}^{N} \sum_{d=c+1}^{N} \mathbf{N}(c,d)\mathbf{N}(c,d)^T \tag{11}$$

A measure of the ability of each feature to distinguish between vectors belonging to different clusters is given by the trace of $\mathcal{D}$, $tr(\mathcal{D})$. We call this quantity the ***Feature Discriminant.***

Definition 3.2 The Feature Discriminant FD *is defined as*

$$FD = tr(\mathcal{D}) \tag{12}$$

See [14] for a comprehensive review of the terminologys used.

3.2 The Feature Extraction Transformation

We now diagonalize the matrix $\mathcal{D}$ so that:

$$\mathbf{DV} = \mathbf{\Lambda V} \tag{13}$$

where A is the diagonal matrix whose diagonal matrix elements are the eigenvalues

$$\lambda_1, \lambda_2, \ldots, \lambda_m \tag{14}$$

and V is the matrix formed by the corresponding eigenvectors

$$\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_m \tag{15}$$

In matrix notations, the eigenvalue matrix Λ and eigenvector matrix **D** can be written as follows.

$$\Lambda = \left(\begin{array}{c|c} \lambda_1 \;\; \ddots \;\; \lambda_r & \\ \hline & \lambda_{r+1} \;\; \ddots \;\; \lambda_m \end{array}\right) \tag{16}$$

$$\mathbf{D} = [\mathbf{V}_1 \ldots \mathbf{V}_r \mathbf{V}_{r+1} \ldots \mathbf{V}_m] \tag{17}$$

We shall assume that the λs are in descending order so that

$$\lambda_1 > \lambda_2 > \ldots > \lambda_r \geq 0 \tag{18}$$

$$\lambda_{r+1}, \lambda_{r+2}, \ldots, \lambda_m \approx 0 \tag{19}$$

We now define a $r \times m$ matrix U

$$\mathbf{U} = [\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_r] \tag{20}$$

and a $(m - r) \times m$ matrix

$$\mathbf{U}' = [\mathbf{V}_{r+1}, \mathbf{V}_{r+2}, \ldots, \mathbf{V}_m]. \tag{21}$$

We shall call the U matrix the ***Feature Reduction Matrix***. This matrix transforms a m-dimensional ***Full Feature Vector*** **z** into a vector in an r-dimensional space

$$\mathbf{r} = \mathbf{U}^T \mathbf{z} \tag{22}$$

The new vectors **r** are called the ***Reduced Vectors.***

3.3 The Euclidean Matrix

The training set of full feature Vectors **z** are clustered with respect to their values of y. The quality of prediction depends on the total Euclidean distances of the vectors to their respective centroids. The smaller this quantity, the better the vectors are clustered about their centroids and the better prediction can be obtained. This quantity is computed from the Euclidean Matrix.

Definition 3.3 ***The*** Euclidean Matrix ***E is defined as***

$$\mathbf{E} = \sum_{c=1}^{N} \sum_{i=1}^{N_c} (\mathbf{z}^{c,i} - \mathbf{W}^c)(\mathbf{z}^{c,i} - \mathbf{W}^c)^T \tag{23}$$

It is clear that the trace of this matrix gives the total Euclidean distances of each training vector from its respective centroids.

Definition 3.4 ***The*** Total Euclidean Distance TE ***is defined as***

$$TE = tr(\mathbf{E}) \tag{24}$$

4 Three Important Properties of the Feature Reduction Procedure

The feature reduction procedure described above has the following three nice properties.

1. It reduces the length of the state vectors from ***m*** to ***r***.
2. It preserves the ***Feature Discriminant*** FD.
3. It reduces the ***Total Euclidean Distances*** TE.

4.1 Reduction in the Dimensions of the Feature Vectors

The ability of the feature extraction procedure to reduce the size of the state vector depends on the rank of the matrix D. Since $\mathcal{D}$ is the sum of the direct products of vectors, we shall discuss the property of such matrices in the next subsection.

4.1.1 Rank of Sum of Direct Products

Definition 4.1 *(Sum of Direct Products) The Sum of Direct Products of a set of vectors* $\{ \mathbf{T}^j, j = 1, r \}$ *is defined as*

$$\mathbf{S} = \sum_{j=1}^{r} \mathbf{T}^j (\mathbf{T}^j)^T \tag{25}$$

Lemma 4.1 *The rank of the Sum of Direct Products of* r *vectors* $\leq r$.

Proof The rank of each of the direct product $\mathbf{T}^j(\mathbf{T}^j)^T$ is 0 or 1. The rank of the sum of r such matrices $\leq$ r.

Lemma 4.2 *If the* T *vectors in Lemma 4.1 are constructed by taking linear combiirations of s independent vectors* $\{ \mathrm{V}^i, i = 1, s \}$, *then the rank of* $\mathbf{S} \leq s$.

Proof Let **L** be the length of the T vectors, we can write:

$$\mathbf{S} = \begin{pmatrix} \sum_{j=1}^{r} T_1^j (\mathbf{T}^j)^T \\ \sum_{j=1}^{r} T_2^j (\mathbf{T}^j)^T \\ \cdots \\ \sum_{j=1}^{r} T_L^j (\mathbf{T}^j)^T \end{pmatrix}$$

Since the row vectors are constructed from s independent vectors, it is not possible to have more than s independent rows. Therefore

$$rank(\mathbf{S}) \leq s \tag{26}$$

From Lemma 4.1 and Lemma 4.2 it is easy to infer the following theorem.

Theorem 4.1 *The rank of matriz S is less than or equal to* r *or* s *which ever is the lesser, i.e.,*

$$rank(\mathbf{M}) \leq \min(r, s) \tag{27}$$

Since $\mathcal{D}$ is the sum of the direct product of $\frac{N(N-1)}{2}$ vectors and only N − 1 of these are linearly independent, we conclude that

$$rank(\mathcal{D}) \leq \mathrm{N} - 1. \tag{28}$$

Therefore, the size of the *Reduced Vectors* $\leq N - 1$. In a typical prediction task, the value of m is around 100 and the value of N is around 10.

4.2 The Preservation of the Feature Discriminant

We recall that the *Feature Reduction Matrix U* transforms a m-dimensional *Full Feature Vector z* into a *Reduced Vector x* in a r-dimensional space by

$$\mathbf{r} = \mathbf{U}^T \mathbf{z}. \tag{29}$$

In the original space, the *Feature Discriminant* is

$$FD = tr(\mathcal{D}) = \sum_{i=1}^{m} \lambda_i = \sum_{i=1}^{r} \lambda_i \tag{30}$$

because $\lambda_{r+1}, \ldots, \lambda \approx 0$. In the transformed space, the *Feature Discriminant* is

$$FD' = \sum_{i=1} \lambda_i. \tag{31}$$

Therefore the *Feature Discriminant* is preserved by this transformation.

4.3 Reduction in the Total Euclidean Distances

In the original space, we defined the Euclidean Matrix E *as*

$$\mathbf{E} = \sum_{c=1}^{N}\sum_{i=1}^{N_c}(\mathbf{z}^{c,i} - \mathbf{W}^c)(\mathbf{z}^{c,i} - \mathbf{W}^c)^T \tag{32}$$

Under the transformation of the matrix $[\mathbf{U}|\mathbf{U}']$, we write

$$\mathbf{E}'' = [\mathbf{U}^T|\mathbf{U}'^T]\mathbf{E}[\mathbf{U}|\mathbf{U}'] \tag{33}$$

$$= \begin{pmatrix} \mathbf{U}^T\mathbf{E}\mathbf{U} & (\mathbf{U}')^T\mathbf{E}\mathbf{U} \\ \mathbf{U}^T\mathbf{E}(\mathbf{U}')^T & (\mathbf{U}')^T\mathbf{E}(\mathbf{U}')^T \end{pmatrix} \tag{34}$$

It is obvious that

$$T\,E = tr(\mathbf{E}) \geq tr(\mathbf{U}^T\mathbf{E}\mathbf{U}) \tag{35}$$

When the ***Reduced Feature Vector*** $\mathbf{r}$ is used, we define the corresponding Euclidean matrix $\mathbf{E}'$ as

$$\mathbf{E}' = \sum_{c=1}^{N}\sum_{i=1}^{N_c}(\mathbf{r}^{c,i} - \mathbf{W}'^c)(\mathbf{r}^{c,i} - \mathbf{W}'^c)^T \tag{36}$$

where we have assume that in the transformed space each state vector remains assigned to the same centroid it was assigned to in the original space so that

$$\mathbf{r}^{c,i} = \mathbf{U}^T\mathbf{z}^{c,i} \tag{37}$$

and the centroid in the new space for cluster c can be written as

$$\mathbf{W}'^c = \frac{1}{N_c}\sum_{i=1}^{N_c}\mathbf{r}^{c,i} \tag{38}$$

$$= \frac{1}{N_c}\sum_{i=1}^{N_c}\mathbf{U}^T\mathbf{z}^{c,i} \tag{39}$$

$$= \mathbf{U}^T\mathbf{W}^c \tag{40}$$

Therefore

$$E' = \mathbf{U}^T\mathbf{E}\mathbf{U} \tag{41}$$

and the total Euclidean distance in the transformed space TE' is reduced ***as*** follows.

$$TE' = tr(\mathbf{E}') \leq T\,E \tag{42}$$

5 The Feature Reduction Procedure

The proposed feature reduction procedure consists of the following steps.

1. The m dimensional ***Full Feature Vector*** are used to construct the ***Decision Boundary Feature Matrix*** $\mathcal{D}$.
2. $\mathcal{D}$ is then diagonalized to obtain the ***Feature Reduction Matrix*** U.
3. The Full Feature Vectors $\mathbf{z}$ are transformed to r dimensional Reduced Vectors x by

$$\mathbf{r} = \mathbf{U}^T\mathbf{z} \tag{43}$$

4. The ***Reduced Vectors*** r are then used for prediction tasks.

6 Predicting the Mackey-Glass Time Series

The following discussion is based on the example of the prediction of a univariate chaotic time series. In this case, the full feature set consists of delayed samples of the time series. However, the methodology can be generalized to general prediction problem.

For the sake of discussion, our discussion is based on the prediction of time series and the features in our full feature set are made up of delayed samples of the time series. The prediction accuracy will be reported in **nrmse** or normalized root mean square values defined as follows. If the true Y values of the prediction set is

$$\mathbf{Y}^j \quad j = t : n \tag{44}$$

and let

$$\mathbf{L} = [\mathbf{Y}^j \quad j = t : n] \tag{45}$$

and the predicted Y is

$$\hat{\mathbf{Y}}^j \quad j = t : n \tag{46}$$

and let

$$\mathbf{L}' = [\hat{\mathbf{Y}}^j, j = t : n] \tag{47}$$

then the **nrmse** of the predicted $\boldsymbol{Y}$ with respect to the true Y is

$$nrmse(\mathbf{L}, \mathbf{L}') = \frac{\sqrt{mean((\mathbf{L} - \mathbf{L}')^2)}}{\sigma(\mathbf{L})} \tag{48}$$

where $\sigma(\mathbf{L})$ denotes the standard deviation of the vector L. The mean operation in Equation 48 makes the measure independent of the length of vector L. The normalization of the quantity in Equation 48 removes the dependence on the dynamic range of the data. From Equation 48, if the mean of L is used as the prediction for L, i.e., $\mathbf{L'} = \mathbf{mean(L)}$, then,

$$\mathbf{nrmse}(\mathbf{L}', \mathbf{L}) = 1.0. \tag{49}$$

In **ClusNet**, state vectors are grouped into clusters such that patterns within the same cluster can be predicted similarly. An important subproblem of the prediction task is to classify these patterns to the correct prediction cluster. In this section, we apply the algorithms described above in the prediction of the benchmark Mackey-Glass chaotic time series[4]. We choose a prediction lead time of T = 85 for our demonstration. **SupNet** results in N = 7 clusters. The number of nonzero eigenvalues for $\mathcal{D}$ is 6. As shown in Table 1, this procedure results in improved prediction performance on the Mackey-Glass benchmark. The prediction is done using ClusNet. The entry refers to the conventional feature vector for this problem consisting of 4 features.

In Table 2, we see a reduction in the TE in the transformed feature space compared to that computed in the original space. The reduction results in a tighter clustering of the vectors around their respective centers and thus reduces the probability of predicting a vector to belong to a different cluster and thus improve prediction accuracy.

In a separate experiment, a 32 feature subset of the original feature set (of length 100) is chosen. Using this feature extraction procedure, results are tabulated in in Table 3. This experiment suggests that this feature extraction procedure is more effective if some preprocessing to remove the less useful features prior to applying this feature reduction algorithm.

7 Summary

In this paper, a decision boundary method is applied to the problem of selecting important features (embedding) for the prediction of chaotic time series. The reduced feature sets are linear combinations of the original features. We demonstrate the effectiveness of this method by applying it to the **Mackey-Glass** chaotic time series.

Table **1:** Experimental Results **I** with the Feature Extraction Algorithm. The Reduced Feature Vector of length **6** is obtained from the transformation of the Original Feature Vector of length **100.** The Prediction Performance are reported in ***nmse*** on a **500** point prediction set. The Prediction Algorithm used is ***ClusNet*** with two different number of clusters as shown on line **3** and **4** of the Table.

Feature Description	Length of Feature Vector	Num Clusters	Prediction
Standard	4	**65**	**0.2000**
Full Feature Vector **f**	**100**	**78**	**0.2340**
Reduced Feature Vector **r**	**6**	**61**	**0.1876**
Reduced Feature Vector **r**	**6**	**89**	**0.1720**

Table **2:** The Reduction in the TE measure from the Original Feature Space to the Reduced Feature Space. This quantity is defined in Equation **5.37.** A smaller TE measure means the clusters are more separated.

space	TE ($\times 10^3$)
Original Feature Space	1.5952
Reduced Feature Space	1.2925

Table **3:** Experimental Results **II** with the Feature Extraction Algorithm. The Full Feature Vector in this Table is a subset of the Full Feature Vector in Table **5.1.** The Reduced Feature Vector of length **6** is obtained from the transformation of the Full Feature Vector of length **32.** The Prediction Performance are reported in ***nmse*** on a **500** point prediction set. The Prediction Algorithm used is ***ClusNet.***

Feature Description	Length of Feature Vector	Num Clusters	Prediction
Standard	**4**	**65**	**0.2000**
Full Feature Vector **f**	**32**		**0.1925**
Reduced Feature Vector **r**	**6**		**0.1438**

References

[1] W. A. Brock and C. L. Sayers. Is the business cycle characterized by deterministic chaos? Journal of Monetary Economics, 22:71–90, 1988.

[2] E. E. Peters. Chaos and Order in the capital markets. Wiley, 1991.

[3] J. Gleick. Chaos–making a new science. Viking, 1987.

[4] M. C. Mackey and L. Glass. Oscillation and chaos in physiological control systems. Science, 197:287, 1977.

[5] M. Casdagli, D. D. Jardins, S. Eubank, J. D. Farmer, J. Gibson, N. Hunter, and J. Theiler. Nonlinear modeling of chaotic time series : theory and applications. Technical Report LA-UR-91-1637, Los Alomos National Lab, 1991.

[6] F. Takens. Detecting strange attractors in turbulence. In D. A. Rand and L. S. Young, editors, Dynamical systems and turbulence, page 366. Spinger, 1981.

[7] W. Hsu, L. S. Hsu, and M. F. Tenorio. An embedding selection algorithm for the prediction of chaotic dynamical systems. Technical Report TR-EE-92-xx, Purdue University, 1992.

[8] S. Watanabe. Karhunen-loeve expansion and factor analysis – theoretical results and applications. *4th* Proc. Conf. Info. Theory, 1965.

[9] C. Lee and D. A. Landgrebe. Feature selection for neural networks using Parzen density estimator. In Proceedings of IGARSS, pages 839–841, 1992.

[10] R. O. Duda and P. E. Hart. Pattern classification and scene analysis. John Wiley and sons, 1973.

[11] W. Hsu, L. S. Hsu, and M. F. Tenorio. Choosing embedding schemes via supervised learning. In Proceedings of *ICNN'93* (Submitted). IEEE, 1993.

[12] W. Hsu, L. S. Hsu, and M. F. Tenorio. A neural network architecture for prediction. Technical Report TR-EE-92-38, Purdue University, 1992.

[13] W. Hsu, L. S. Hsu, and M. F. Tenorio. A ClusNet architectuer for prediction. In Proceedings of *ICNN'93* (Submitted). IEEE, 1993.

[14] Strang. Linear Algebra and its Applications. Academic Press, 1985.