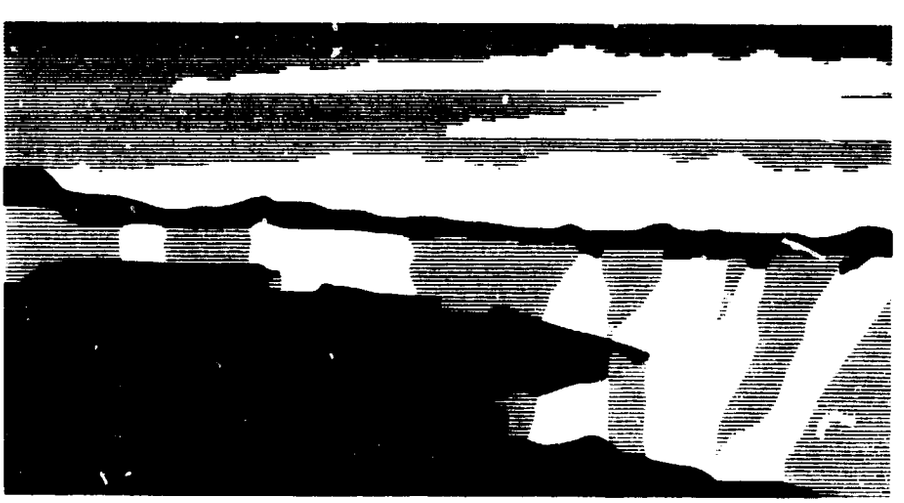
Conf-950472--4

LA-UR- 95-883

Title: SENSOR FUSION AND NONLINEAR PREDICTION FOR ANOMALOUS EVENT DETECTION

Author(s): Jose V. Hernandez, NIS-1
Kurt R. Moore, NIS-1
Richard C. Elphic, NIS-1

Submitted to: SPIE
April 17-21 1995
Orlando, FL



Los Alamos
NATIONAL LABORATORY

Form No. 836 R5
ST 2629 10/91

Sensor Fusion and Nonlinear Prediction for Anomalous Event Detection

J.V. Hernández, K. Moore, and R. Elphic
Los Alamos National Laboratory
NIS-1/D466
Los Alamos, NM 87545

March 7, 1995

Abstract

We consider the problem of using the information from various time series, each one characterizing a different physical quantity, to predict the future state of the system and, based on that information, to detect and classify anomalous events. We stress the application of principal components analysis (PCA) to analyze and combine data from different sensors. We construct both linear and nonlinear predictors. In particular, for linear prediction we use the least-mean-square (LMS) algorithm and for nonlinear prediction we use both backpropagation (BP) networks and fuzzy predictors (FP). As an application, we consider the prediction of gamma counts from past values of electron and gamma counts recorded by the instruments of a high altitude satellite.

1 Introduction

Here we report our progress on the problem of detection and characterization of multi-instrument signatures of anomalous events. Our approach is to combine past multi-instrument information in order to predict the future state of the system. If the predicted and the actual values differ significantly, then

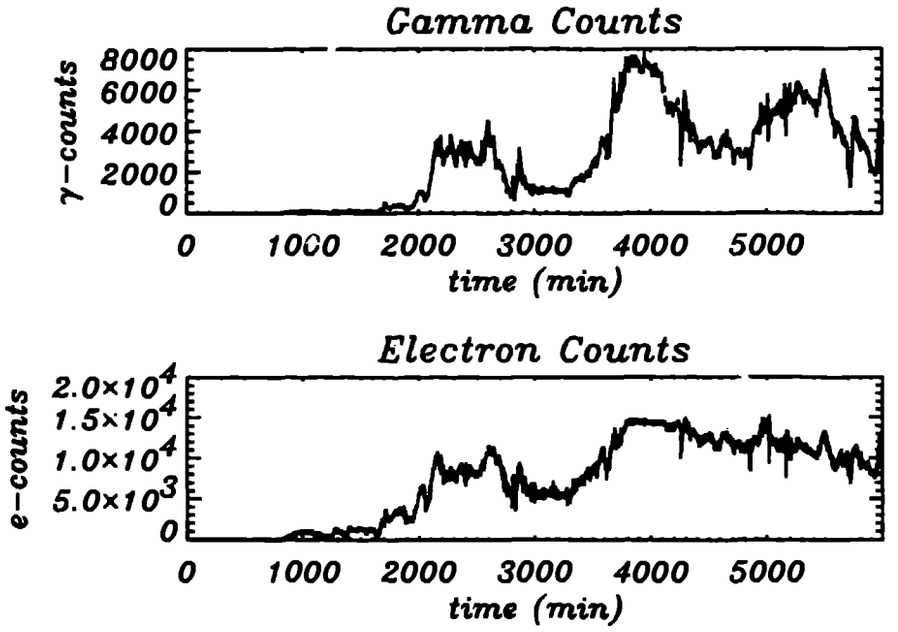


Figure 1: Gamma and electron counts as a function of time. The time resolution of the data is 1 min.

we may interpret that difference as evidence about the possible occurrence of an anomalous event. As a benchmark, we consider the use of past electron and gamma counts, as recorded by several instruments onboard a satellite, in order to predict future gamma counts. Plots for gamma and electron counts as a function of time are shown in Figure 1.

We can split our approach into two coupled problems: the predictor design problem and the combination of multi-instrument measurements problem. In general, the design of a prediction system involves the determination of a function f, which relates past and present information to future values of the quantity that we wish to predict. We can design both linear (the output is proportional to the input) and nonlinear prediction systems.

Recently [1, 2], artificial neural networks have emerged as a flexible nonlinear prediction tool. The reason for this is that feed forward neural networks, under certain conditions that will be discussed in Section 3, are known to be universal approximants to functions. If there is a linear or nonlinear function

relating past information to future values. then a neural network with the appropriate architecture should be able to determine that function.

Backpropagation networks are not the only possible systems which can be used as universal approximators of continuous real-valued functions. There are several approaches. In particular, we use the fuzzy learning algorithm of Wang and Mendel [3] to approximate the function. if any, connecting the past to the future. When constructing fuzzy rules from input-output data. the fuzzy learning algorithm requires a single pass through the training data. This is in sharp contrast with the training of backpropagation networks, which requires multiple passes (epochs) through the training data.

In order to assess the goodness of our predictions we use two diagnostics: the normalized mean squared error $\mathcal{E}$ and the correlation coefficient ρ between actual and predicted values. Both $\mathcal{E}$ and ρ will be formally defined in Section 2. Ideally $\mathcal{E}$ should be as close to 0 as possible and ρ should be as close to 100% as possible.

Regarding the inputs combination issue, we have chosen to represent the input data in the principal components representation. There are several reasons underlying this choice. In particular, if the first principal components are the ones with most of the intrinsic information of the data, then we can get information about the relative importance of the input data by considering the components of the principal vectors. PCA will be discussed in Section 2.

2 Gamma Counts Prediction

The gamma counts prediction problem can be stated as follows: let $\vec{\xi}_t$ be a data vector containing past information on electron and gamma counts up to time t,

$$\vec{\xi}_t = \left(e_t, e_{t-T}, \ldots, e_{t-(p-1)T}, \gamma_t, \gamma_{t-T}, \ldots, \gamma_{t-(q-1)T}\right). \tag{1}$$

where e_t and γ_t denote the number of electron and gamma counts at time t. respectively. Let γ_{t+T} denote the future value of the gamma counts at time $t+T$ and assume that there exists a function f connecting past and present information with the future,

$$\gamma_{t+T} = f(\vec{\xi}_t). \tag{2}$$

The problem is to determine f.

In order to measure the goodness of our predictions we use two quantities: the normalized error $\mathcal{E}$ and the correlation coefficient ρ. In the following we will denote the predicted value of γ_{t+T} by $\hat{\gamma}_{t+T}$. We define the normalized error $\mathcal{E}$ as the ratio of the mean square error MSE.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\gamma_i - \hat{\gamma}_i)^2, \tag{3}$$

where N is the number of points in the sample. to the variance VAR of the actual data.

$$VAR = \frac{1}{N}\sum_{i=1}^{N}(\gamma_i - \langle\gamma\rangle)^2. \tag{4}$$

that is.

$$\mathcal{E} = MSE/VAR. \tag{5}$$

The correlation coefficient ρ is defined as

$$\rho = \frac{\frac{1}{N}(\gamma_i - \langle\gamma\rangle)(\hat{\gamma}_i - \langle\hat{\gamma}\rangle)}{\sigma_\gamma \sigma_{\hat{\gamma}}}, \tag{6}$$

where σ_γ and $\sigma_{\hat{\gamma}}$ are the standard deviations of the actual and the predicted gamma counts, respectively. The prediction is perfect if $\mathcal{E} = 0$ and $\rho = 100\%$.

If the function f is approximated by a linear method such as the LMS algorithm [4], then the predictor is linear. On the other hand, if f is approximated by a nonlinear method. such as a BP network with nonlinear activation functions or a fuzzy predictor. then the predictor is nonlinear.

2.1 Backpropagation Networks

Feed-forward neural networks with l inputs, one or several hidden layers of units with nonlinear activations, and one output layer with m outputs are known to be universal approximants to mappings of the form $f : R^l \rightarrow R^m$. For an introduction to the theory of neural networks we refer the reader to [5].

In all of our nonlinear predictors, we use feed-forward networks with one hidden-layer of nonlinear activation functions. $g(x) = \tanh(x)$. The networks are trained using the backpropagation algorithm [6] with the addition of a momentum term [7] to accelerate convergence. The inputs to the network

are given by the first few principal components. obtained by projecting ξ_t into the principal components basis. The network has one output for the predicted x_{t+T}.

2.2 Fuzzy Rule Extraction From the Data

The theory of fuzzy sets [8] provides a useful framework for representing and making inferences with vague or uncertain information. Traditional fuzzy inference systems have been constructed using fuzzy rules provided by a human expert. On the other hand. in a neural network the rules are extracted by the network using the input-output training data. Recently, Wang and Mendel [3] have deviced a fuzzy learning algorithm for extracting fuzzy rules from numerical data. Wang and Mendel also showed that the resulting inference system can be used to approximate any continuous real-valued function. In the special case in which we apply the Wang and Mendel algorithm to a prediction problem. we refer to it as the fuzzy predictor.

The fuzzy predictor has several advantages over backpropagation networks:

1. The extraction of fuzzy rules requires a single pass through the training data. On the other hand, backpropagation networks require several passes through the training data in order to achieve good function approximation:

2. Rules generated by a human expert can be easily incorporated into the fuzzy rule base. In contrast. there is not an straightforward and general procedure to implement rules generated by a human expert into a backpropagation network:

3. Fuzzy predictors are local. that is, the effect of each rule is concentrated in the vicinity of the training input which was used to generate the rule. Backpropagation networks implement global mappings. The presence of a new training sample affects, in general. all the weights in the network.

We now describe the fuzzy learning algorithm of Wang and Mendel. The discussion follows [3] and is included here just to make the presentation self-

contained. Suppose we are given a set of K input-output pairs:

$$\left(\vec{x}^{(1)}, y^{(1)}\right) \ldots\ldots \left(\vec{x}^{(K)}\ y^{(K)}\right). \tag{7}$$

where $\vec{x}$ is an m-dimensional vector of input values, y is the corresponding output value, and the superscript denotes the sample number. The task is to approximate a function f relating the inputs $\vec{x}$ to the output y,

$$y = f(\vec{x}). \tag{8}$$

The approach is to approximate f through the generation of a set of fuzzy rules such as:

$$\text{if } [(x_1 \text{ is } A_1) \text{ and } (x_2 \text{ is } A_2) \text{ and } \ldots (x_m \text{ is } A_m)] \text{ then } (y \text{ is } C_j). \tag{9}$$

where A is an antecedent or predicate fuzzy region and C is a consequent fuzzy region.

The Wang and Mendel algorithm consists of the following steps:

1. **Divide the input and output spaces into fuzzy regions.** Let $\left[x_j^-, x_j^+\right]$ and $[y^-, y^=]$ denote the domain intervals for the jth input x_j and the output y, respectively. Divide each domain interval into $2N+1$ regions and assign each region a fuzzy membership function μ, as shown in Figure 2.

2. **Generate the fuzzy rules from the training input-output data pairs.** For each one of the trained samples and using the assigned membership functions: a) determine the degrees of given $x_1^{(j)}, \ldots, x_m^{(j)}$ and y in different fuzzy regions. b) assign a given $x_1^{(j)}, \ldots . x_m^{(j)}$ or y to the region with maximum degree, obtain one rule, such as Eq. (9), from each training input-output sample, d) assign a degree to each of the fuzzy rules, and e) solve conflicts between rules by giving priority to the rule with maximum degree. Fuzzy rules generated by a human expert can be easily implemented into the fuzzy rule base. This is achieved by assigning a degree to the rule generated by the human expert and solving conflicts with other rules in the way just described.

3. **Determine a mapping f based on the fuzzy associative memory (FAM).** Given the out-of-sample data $\{(x_1, x_2, \ldots, x_m)\}$, use some defuzzification procedure to determine the output y. Wang and Mendel use a centroid defuzzification formula.

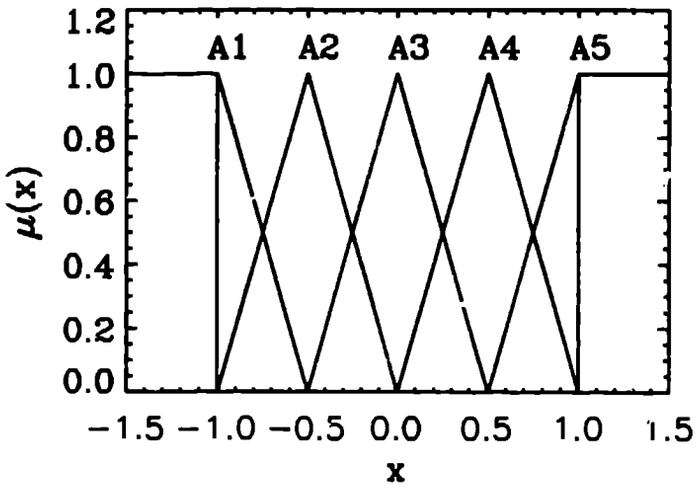


Figure 2: Fuzzy regions and fuzzy membership functions

2.3 Principal Component Analysis

One of the most important issues when applying fuzzy predictors and back-propagation networks is that of data preprocessing. Appropriate data preprocessing leads to a more efficient use of the information contained in the past values vector $\vec{\xi}$ of Eq. (1). In our data preprocessing stage we use PCA. With PCA we get good input data compression (dimensionality reduction) and noise reduction while preserving as much information about the inputs as possible. PCA has been used for image coding [9] and to reduce the dimension of speech signals for vowel classification [10]. For a general discussion on PCA see [11].

We obtain the ith principal component, y_i, projecting $\vec{\xi}$ along the ith unit eigenvector, $\vec{u}^{(i)}$, of the covariance matrix $\mathbf{C}$.

$$C_{jk} = \langle (\xi_j - \langle \xi_j \rangle)(\xi_k - \langle \xi_k \rangle) \rangle . \tag{10}$$

The principal components are ordered in terms of decreasing eigenvalue. The ith eigenvalue λ_i is the variance of the data along the ith direction.

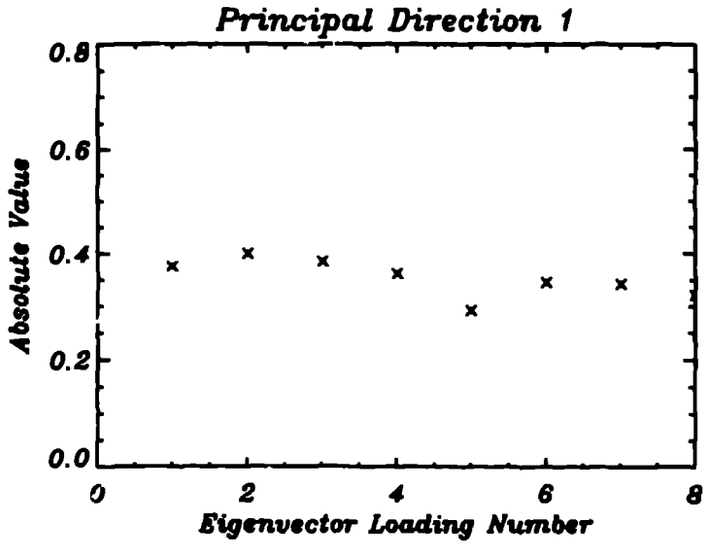


Figure 3: Absolute value of the components of the first principal unit eigenvector of the covariance matrix.

2.4 Results

The way in which we preprocess our input data follows. We start with the input vector $\vec{\xi}_t$ of Eq. (1) with 8 components (4 for past values of e and 4 for past values of γ) and project it into the principal components representation [12]. The input $\vec{\chi}_t$ to the predictors is given by the first 3 principal components. Predictors with more than 3 principal components as inputs did not lead to any improvement in the predictions.

Figure 3 is a plot of the absolute value of the components or loadings of the first unit eigenvector $\vec{u}^{(1)}$ of the covariance matrix. The first principal component χ_1 is obtained by projecting $\vec{\xi}$ along $\vec{u}^{(1)}$. The components of the input vector $\vec{\xi}$ are past values of e and past values of γ. For the case shown in Figure 3, the time lag is $T = 120$ min. In Figure 3 the components 1-4 of $\vec{u}^{(1)}$ determine the contribution of past values of e to χ_1. Similarly, the components 5-8 of $\vec{u}^{(1)}$ determine the contribution of past values of γ to χ_1. From Figure 3 we have that the relative importance of the contribution of past electron and gamma counts to χ_1 is similar.

Figure 4 is a plot of the eigenvalues of C as a function of the principal di-

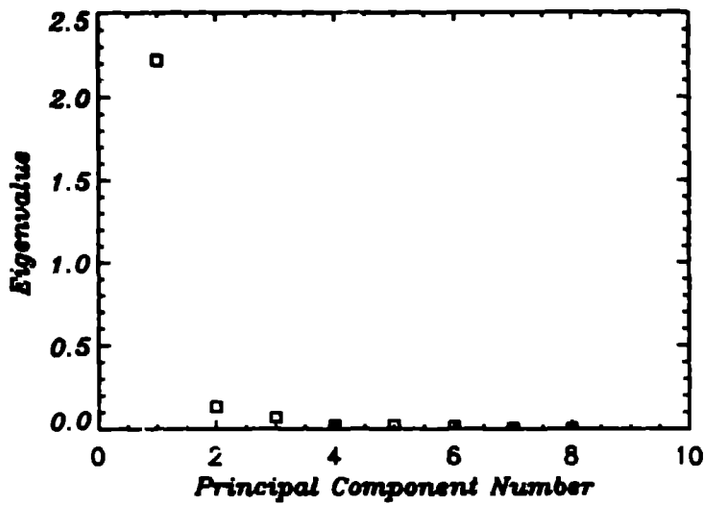


Figure 4: Eigenvalues of the covariance matrix as a function of the principal component number.

rection number. We observe that the only significant eigenvalues are the first three. This is consistent with the observation that, in all our trials, the best predictions were obtained using only the first three principal components. It is interesting to note that whenever we used the raw data vector $\vec{\xi}$ as input to our predictors, the prediction error was larger than in the case when we used only the first three principal components. The moral is that PCA is an effective tool for combining diverse signals, for dimensionality reduction, and for damping the effect of noise.

Figure 5 is a plot of the normalized prediction error and the correlation coefficient as a function of the prediction time T. The results shown correspond to single-step predictions of gamma counts using the previous value (stars) and fuzzy predictors (boxes). In all our trials, the prediction results of both linear LMS predictors and nonlinear BP networks were only comparable to the results obtained using the previous value as the predicted value. The poor performance of BP networks compared to fuzzy predictors is due to the fact that BP networks implement global mappings between past information and future gamma counts and then, given the nonstationary character of

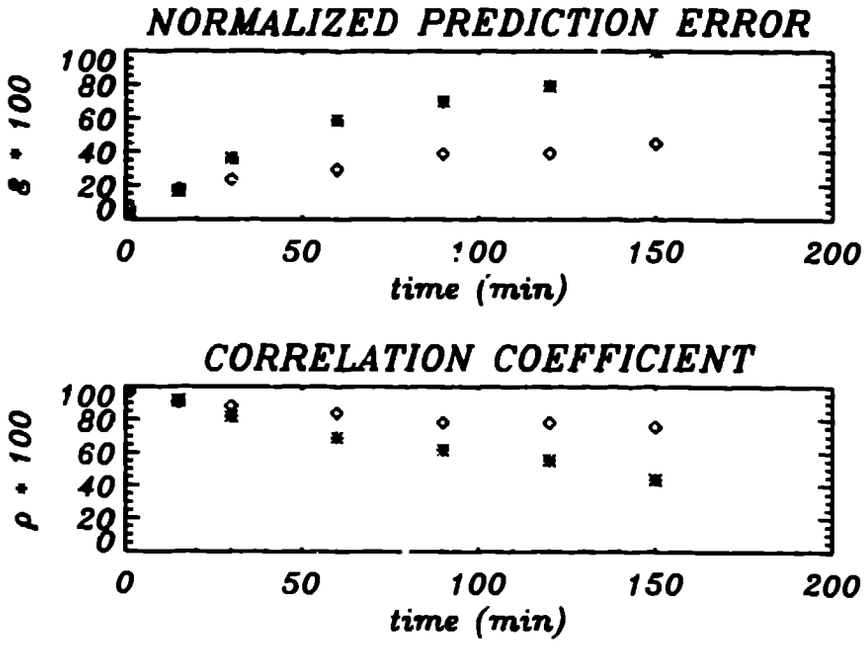


Figure 5: Normalized prediction error $\mathcal{E}$ and correlation coefficient ρ as a function of the prediction time for single-step predictions. The results were generated using the previous value as the predicted value (stars) and fuzzy predictors (boxes). It is interesting to note that, due to their local character, fuzzy predictors outperformed backpropagation networks and linear predictors.

the time series displayed in Figure 1. BP networks capture only the average properties of the mapping. On the other hand, fuzzy predictors implement mappings between the past an the future using local inference rules.

The results of single-step predictions $T = 120$ min ahead of time are shown in Figure 6. The solid curve represents the actual gamma counts and the dotted curve represents the predicted gamma counts using a fuzzy predictor. Figure 7 is a scatter plot of the predicted and the actual gamma counts. Perfect predictions would lie along the diagonal (dashed) line.

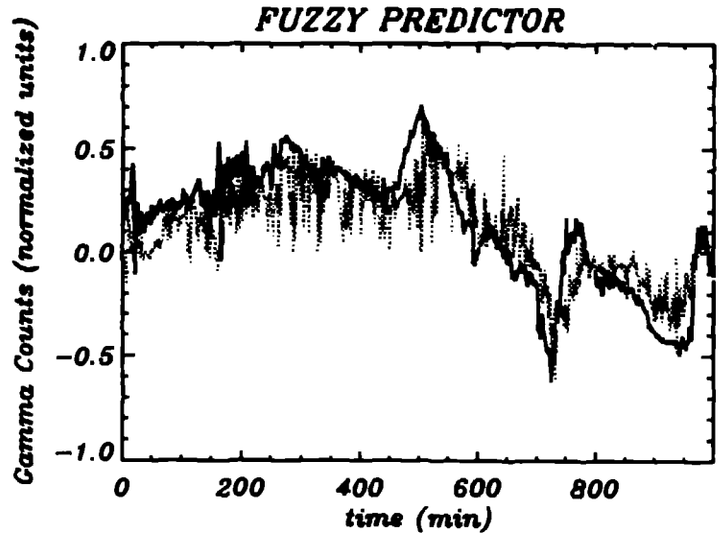


Figure 6: Actual (solid) and predicted (dots) gamma counts. The results correspond to single-step predictions 120 min ahead of time using a fuzzy predictor.

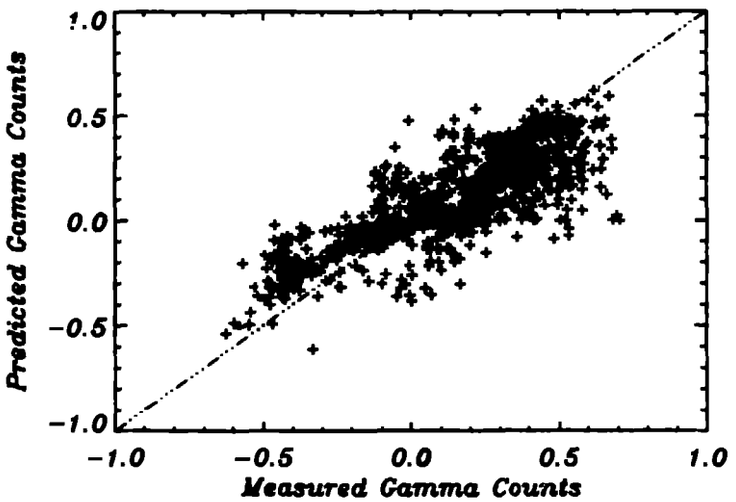


Figure 7: Scatter plot of the results shown in Figure 6. Perfect predictions lie along the diagonal line (dashed), which is shown just for reference.

3 Conclusions

We consider the problem of anomalous event detection from multi-instrument information. Our approach consists of two parts: combine past multi- instrument information in order to predict the future state of the system and use significant deviations between the predicted and actual values as evidence for the occurrence of an anomalous event. As a benchmark, we consider the prediction of future gamma counts from past electron and gamma counts recorded by two instruments onboard a satellite.

We have found that the principal components representation provides a useful framework to combine past multi-instrument information for prediction purposes. In particular, PCA allows us to compress the input data, to determine the relevant variables, and to reduce noise.

We have applied both backpropagation networks and the fuzzy learning algorithm of Wang and Mendel to the prediction of future gamma counts problem. The fuzzy predictor consistently outperformed backpropagation networks in the prediction task. The reason for this is that backpropagation

networks implement a mapping from past to future information using global information. On the other hand, the fuzzy predictor implements the mapping from local inference rules. Given the nonstationary character of the electron and gamma counts time series, a backpropagation network learns the average properties of the time series, whereas a fuzzy predictor exploits the details in the time series.

It is important to note that when we used the raw multi-instrument data as input to our predictors, the prediction accuracy was always smaller than that obtained using only the first principal components. This shows how useful can PCA be for data preprocessing and noise reduction.

Acknowledgements

This work was carried out under the auspices of the United States Department of Energy

References

[1] Lapedes, A. S. and R. M. Farber. Nonlinear signal processing using neural networks: prediction and system modeling. Technical Report LAUR-87-2662, Los Alamos National Laboratory, 1987.

[2] Weigend, A. S., B. A. Huberman and D. E. Rumelhart, Predicting sunspots and exchange rates with connectionist networks. in *Nonlinear Modeling and Forecasting*, M. Casdagli and S. Eubanks, eds., Addison-Wesley, 1992.

[3] Wang, L. and J. M. Mendel, Generating fuzzy rules by learning from examples, IEEE Trans. Syst., Man. Cybern., 22, pp. 1414-1427, 1992.

[4] Widrow, B. and M. E. Hoff, Adaptive switching circuits, in *1960 IRE WESCON Convention Record*, part 4, pp 96-104, New York: IRE, 1960.

[5] Hertz, J., A. Krogh and R. G. Palmer. *Introduction to the Theory of Neural Computation*, ch. 6, Addison-Wesley, 1991.

[6] Rumelhart, D. E., G. E. Hinton and R. J. Williams. Learning representations by backpropagating errors, *Nature*, *323*, 533, 1986.

[7] Plaut, D., S. Nowlan, and G. Hinton. Experiments on learning by back-propagation. Technical Report CMU-CS-86-126. Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA. 1986.

[8] Zadeh, L. D., Fuzzy sets, *Information and Control, 8*, pp. 338-352. 1965.

[9] Sanger, T. D., Optimal unsupervised learning in a single-layer linear feedforward neural network, *Neural Networks, 12*, pp. 459-473, 1989.

[10] Leen, T. K., M. Rudnik, and D. Hammerstrom, Hebbian feature discovery improves classifier efficiency, *International Joint Conference on Neural Networks*, Vol. 1, pp. 51-61, San Diego, CA, 1990.

[11] Jolliffe, I. T., *Principal Component Analysis*, New York, NY: Springer-Verlag, 1986.

[12] Linsker, R., Self-organization in a perceptual network, *Computer*, pp. 105-117, March 1988.