From :

Tattersall G.D, Chichlowski K, Limb R, *Feature extraction and visualisation of decision support data*, BTTJ, vol 10, pp110-123. July 1992.

1.0 Introduction

Statistical pattern recognition and artificial neural net techniques have been widely applied to tasks such as image and speech recognition. The data in these domains is usually continuously valued, limited in dynamic range and has a value which changes smoothly as the quality of the speech or image signal changes. It is usually possible to form very large training sets of data from these domains so that it is possible to accurately estimate the class probability distributions needed for the operation of a statistical classifier, or to fully train a neural net classifier.

More recently, statistical and neural net recognition has been applied to decision support problems such as fault diagnosis [1] and the prediction of company and individual financial behaviour. The decision support domain is typified by data which consists of a mix of symbolic and numerical variables whose dynamic range is unlimited and which does not always change smoothly as the state of the system generating the data changes. Moreover, due to the difficulty of data collection in the decision support domain, the sets of pattern exemplars needed for training a classifier are often small and even incomplete because values for some of the attributes within a single example of a pattern may be missing.

One of the attractions of using neural net rather than statistical recognisers in these circumstances is that the neural net performs discriminatory learning so that, theoretically, it will learn a good input transformation. The input transformation would ideally "normalise" variables of unrestricted dynamic range, optimally re-code arbitrary numerical code chosen to represent symbolic variables and form class discriminatory features from non-linear combinations of the raw data attributes [2],[3]. However, in practice, the performance surface of the neural net is complex with many local maxima on which gradient ascent optimisation may become "stuck", and unless the data applied to the network is sensibly coded in the first place, the neural net is unlikely to find a good classification solution.

A better approach than using complex neural networks may therefore be use a very simple neural network or statistical recogniser to which is input manually pre-processed pattern data. The desired pre-processing would *explicitly* normalise attribute values, limit their dynamic range, and code symbolic quantities appropriately. It would also deal with the problem of missing data attributes by inserting the optimal default values. However, it is not clear exactly how the pre-processing would be done and it is these problems which are primarily addressed in this paper.

The paper also presents a number of analysis and visualisation tools which can be used to gain an understanding of the structure and "classifiability" of both of the raw and pre-processed data so that the usefulness of the pre-processing can be assessed. The use of these tools are demonstrated using two sets of data, one concerning the diagnosis of faults in an electronic system and the other concerning the prediction of financial behaviour.

2.0 The Process of Pattern Classification

2.1 Feature Space and Class Boundaries:

Successful pattern classification rests upon the ability to define regions of a feature space which belong to the different classes. If a pattern of unknown class is found to lie within a feature space region belonging to class, C_i, then the unknown pattern is classified as C_i. The feature space in which the class regions are defined may be the pattern space formed by the raw data measurements, or more usually, it is a space formed by a transformation of the original data, such that class separation is improved. Intuitively, class separation is enhanced by the transformation in the following ways:

i) Noisy or otherwise unreliable pattern attributes are reduced in significance. This is normally done by scaling down the attribute relative to the other attributes.

ii) Pattern attributes whose values are similar for different classes of pattern, are reduced in significance and attributes which provide good class discrimination are increased in significance. The change in significance of the attributes to enhance class separation may be done by either linear or non-linear scaling of the attributes.

iii) Disjoint regions of the pattern space which belong to the same class, are transformed to form a single contiguous region in the feature space.

As discussed in the introduction, artificial neural networks are theoretically capable of learning an input transformation which embodies all of the desirable properties listed above so that good class separation will be obtained in the "feature space" formed by the values of the network output units. Such a transform would generally be non-linear and complex and, in practice, the networks do not reliably learn the appropriate transformation. It is therefore our purpose to explicitly design a transform. Of course, the designed transform is unlikely to be optimal, but it may perform satisfactorily and be better than a poorly trained neural net.

The explicit pattern space to feature space transform can be interpreted as a data pre-processing stage in the classifier and henceforth the terms input transformation and data pre-processing will be used interchangebly. A number of explicitly designed transform types could be relatively simply implemented and the properties of these transforms will be briefly described here, although the method of choosing their parameters will not be discussed until section 5.

2.2 Linear Scaling of Attributes:

A transform which only scales the pattern attributes yields feature vectors, **F**, of the same dimensionality as the pattern vectors, **A**, and can be written formally as the multiplication of the pattern vector into a diagonal transform matrix **W**. The diagonal elements of the matrix are the attribute scaling factors and are usually set subject to the constraint that the sum of their squares is a constant.

$$\begin{bmatrix} f_1 \\ . \\ . \\ f_N \end{bmatrix} = \begin{bmatrix} w_{11} & & & \\ & w_{22} & & \\ & & \ddots & \\ & & & w_{NN} \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ . \\ . \\ a_N \end{bmatrix} \qquad \ldots 1$$

As previously explained, this type of transform allows attributes whose values vary widely, even in examples of the same class, to be down-weighted. This causes individual class clusters to become more compact and possibly less overlapping. However, there is no guarantee that class clusters will have

enhanced separability after diagonal transformation unless some account is taken of the class separation during the evaluation of the transform coefficients. If the transform is to be evaluated using class interaction information, then it is just as easy to evaluate a full non-diagonal transform as oulined in the next section. However, if class interaction is not to be taken into account, the diagonal transform provides a simple method of compacting class clusters and frequently reducing their overlap as a result.

2.3 Linear Rotational Transform Features:
In this linear transform, feature values are formed from the weighted summation of the pattern attributes such that an M-dimensional feature vector, F, is formed from an N-dimensional pattern attribute vector, A.

$$\begin{bmatrix} f_1 \\ . \\ . \\ f_M \end{bmatrix} = \begin{bmatrix} w_{11} & \cdots & w_{1N} \\ . & & . \\ . & & . \\ w_{M1} & \cdots & w_{MN} \end{bmatrix} . \begin{bmatrix} a_1 \\ . \\ . \\ . \\ a_N \end{bmatrix} \qquad \ldots 2$$

A diagramatic representation of this transform shows that a new set of rotated axes are effectively defined by the transfrom matrix **W** and the feature values are the projections of the data onto the new axis set. Ideally, the projections of different classes of data onto the new axis set would be more separated than their projections onto the pattern space axis set.

There is no closed analytic method of finding **W** and it is most easily evaluated by gradient descent optimisation of its coefficients to maximise classification accuracy or some measure of class separation. If classification accuracy is used as the performance measure, the surface on which the optimisation takes place is quadratic and learning can be made very reliable.

2.4 Non Linear Transformations:
Although the linear transform may be able to enhance the separation of classes which occupy a single contiguous region of the pattern space, it cannot cause disjoint regions to merge. To do this requires a non-linear transform. Again, there is no closed analytic method of finding an optimal non-linear transform and if gradient descent optimisation is used, it is completely equivalent to the training of an MLP. This is exactly the approach that we set out to avoid!

A more practical approach is to explicitly define a single type of non-linear transform which is known to be useful in merging disjoint class regions in a wide variety of pattern domains. The chosen non-linear transfrom forms features from the *products* of selected pattern attributes. Such a transform is useful if the class type is defined by the relationship between attribute values rather than their individual absolute values. A simple example is the 2-dimensional EX-OR function in which class 1 patterns have values [-1,1] or [1,-1] and class 2 patterns have values of [1,1] or [-1,-1]. It can be seen that the product of the co-ordinate values for class1 patterns is always -1 and +1 for class 2. Thus simple linear separation is possible between the classes in the feature space formed by the product of the pattern space attributes. Formally the k^{th} feature value is defined as:

$$f_k = \prod_{i=1}^{N} a_i^{s_i} \qquad \ldots 3$$

Where $s_i = 1$ if attribute a_i is selected else $s_i = 0$

There is considerable evidence that this type of feature is often useful in the classification of both data from the physical world and symbolic data and several neural network architectures have been developed which are designed to find this type of feature. An example is the "higher order network" described by Maxwell and Giles [6] in which the weights of a single layer linear perceptron are made equal to the computed

average value of the products of chosen pattern attrributes. An alternative approach is described by Durbin and Rumelhart [7] who propose a multi-layer perceptron in which pattern attributes are scaled and logged before being added by the hidden units of the network. This operation corresponds to forming the product of the attributes, each raised to a power equal to the input weights.

Both of these approaches place product features in a neural network context. However, it is proposed here, that these features should be explicitly evaluated before using a neural net or conventional classifier so that the classifier training is simplified and made more reliable.

3.0 Data Analysis and Visualisation Techniques

3.1 Looking For Clusters:

As discussed in the previous section, successful pattern classification depends upon all the examples of different classes of pattern lying in well separated clusters within a suitably chosen feature space. The feature space may just be the original pattern space or, more usually, its dimensions are formed by a transformation of the original attributes. If different classes of pattern lie in separate clusters, class boundaries can be defined between the clusters and an unknown pattern is classified simply by finding in which class region the pattern lies.

It is therefore very desirable to be able to visualise a set of data and see if it consists of separate clusters. This would present no problem if the data were only two dimensional but in practice it has high dimensionality and direct drawing of the data in a 2-dimensional form is impossible.

A solution to the visualisation of high dimensional data is provided by Sammon's Mapping [4]. In this technique, the set of data points in N-dimensional space is mapped to an equal number of corresponding points in a 2-dimensional space. The *relative* positions of the 2-dimensional points are iteratively modified until their relative distances mirrors as closely as possible the *relative* distances between pairs of points in the N-dimensional space. Thus, if clusters exist in the N-dimensional space, corresponding clusters will be formed in the 2-dimensional space which can easily be plotted.

The iterative adjustment of the positions of the points in 2-dimensional space is done using a gradient descent minimisation of a mapping error, E, which is defined as the average of the squared difference between the distance of each pair of points in the N-dimensional space and a corresponding pair of points in the 2-dimensional space.

Let the k^{th} N-dimensional vector in the set of patterns to be visualised be denoted as X_k and the corresponding 2-dimensional vector be Y_k. Assuming that there are a total of M examples in the data set, there are M points in the N-dimensional space and 2-dimensional space. The squared distances between the i^{th} and j^{th} point in the N and 2-dimensional spaces are $d_x(i,j)$ and $d_y(i,j)$ respectively and these can be expressed as:

$$d_x(i,j) = \sum_{l=1}^{N} (x_{i,l} - x_{j,l})^2 \qquad \ldots 4$$

$$d_y(i,j) = \sum_{l=1}^{2} (y_{i,l} - y_{j,l})^2 \qquad \ldots 5$$

Where $x_{i,l}$ and $y_{i,l}$ are the values along the l^{th} dimension of X_i and Y_i respectively.

A measure of the total difference between the distances of pairs of points in the N-dimensional and 2-dimensional spaces is therefore:

$$E = \sum_{i=1}^{M}\sum_{j=1}^{M}(d_x(i,j) - d_y(i,j))^2 \qquad \ldots 6$$

Usually the initial values of Y_k are set randomly and are iteratively modified using the steepest descent algorithm:

$$Y_k^{n+1} = Y_k^n - k \cdot \frac{\bullet E}{\bullet Y_k} \qquad \ldots 7$$

The value of the derivitive of E with respect to the positions of each of the points in the 2-dimensional space is found by chain differentiating equation 6 and 5.

3.2 Measuring The Usefulness of Pattern Attributes:
A classical technique for assessing the usefulness of pattern attributes in discriminating between different classes of pattern is by measuring the *mutual information* provided by each attribute [2].

The mutual information provided by a particular attribute is a measure of the correlation between its value and the occurrence of particular classes. Thus an attribute which always has a value of '1' when different examples of one particular class occurs and has a value of '0' for all examples of other classes is intuitively a useful attribute and carries a high mutual information. Conversely, a feature which may have a value of '1' or '0' with equal probability regardless of the class of input pattern is useless and carries no mutual information.

The mutual information provided by each attribute can be calculated from a set of pattern examples whose classes are known by using the concept of entropy. The entropy of a variable is defined as the average of the base two logarithm of the probabilities of each of its possible values and is measured in bits. Thus a variable which can have one of two values with equal probability will have an entropy of 1 bit whereas a variable which can have one of four values with equal probability has an entropy of 2 bits. Entropy is therefore a measure of the uncertainty about the value of a variable.

This idea can be applied to the evaluation of the mutual information provided by a pattern attribute by first claculating the entropy of the class set. Let C_k denote the k^{th} class of pattern and $P(C_k)$ be the probability of occurrence of a pattern of class C_k. Before any attributes are examined, the average uncertainty about the class of an unknown pattern is given by the entropy of the class set, H(C).

$$H(C) = \sum_{k=1}^{K} P(C_k)\log_2\{\frac{1}{P(C_k)}\} \qquad \ldots 8$$

If the entropy of the class set is now calculated given that the l^{th} possible value of the i^{th} attribute value $a_i(l)$ has been observed, the attribute conditional entropy of the class set, $H(C|a_i)$, is obtained. This measure is evaluated by averaging over all possible feature values and classes.

$$H(C|a_i) = -\sum_{k=1}^{K}\sum_{l=1}^{L_i} P(C_k|a_i(l))\log_2 P(C_k|a_i(l)) \qquad \ldots 9$$

Where $P(C_k , a_i(l))$ and $P(C_k \mid a_i(l))$ are the joint and conditional probabilities of a pattern of class C_k and the attribute $a_i(l)$, and L_i is the number of quantised levels of the i^{th} attribute.

If the feature, a_i, is useful for discriminating between classes, the conditional entropy will approach zero, indicating that the attribute almost completely determines the class. Conversely, the conditional entropy associated with a poor attribute will be almost as high as the unconditional entropy, indicating that almost no information is gained by knowing its value. Thus the information about the class of a pattern given by a particular attribute is simply measured as the difference between the original class set entropy and the attribute conditional entropy. This is the mutual information, $I(C,a_i)$, of the attribute.

$$I(C, a_i) = H(C) - H(C \mid a_i) \qquad \ldots 10$$

The probability distributions, $P(C_k \mid a_i)$ and $P(C_k)$ are estimated from the set of class labelled pattern examples and are then used in equation 9 to derive a mutual information value for each attribute.

3.3 Correlation Analysis:
An alternative method of assessing the usefulness of pattern attributes is by measuring their correlation with each of the possible classes of pattern. An attribute whose value is highly correlated with a particular class is intuitively more useful than an attribute whose value varies widely in different examples of patterns of the same class.

The correlation between pattern class and attribute value can be evaluated using the usual definition of correlation coefficient, R_{xy}:

$$R_{xy} = \frac{< (x-\mu_x) . (y-\mu_y) >}{std(x) . std(y)} \qquad \ldots 11$$

Where x and y are the two values whose correlation is to be measured, std(x) and std(y) are their respective standard deviations, and m_x and m_y are their mean values.

Equation 11 defines the correlation coefficient between two scalar variables. However, in the class-attribute correlation problem, the class is a symbolic quantity and must therefore be replaced by a suitable scalar value before the correlation coefficient can be evaluated. This is simply done by assigning a scalar value to the class whose attribute correlation is being measured, and a different, but common, scalar value to all other classes. The value assigned is not important, and for convenience it is made unity if the class of the pattern is the same as the class whose attribute correlation is being measured and is made minus one if the pattern belongs to any other class.

A complete expression for the correlation coefficient between a pattern attribute, a_j, and the pattern class, C_i, is therefore given by $R(a_j , C_i)$ which is evaluated using N class labelled example patterns in which the k^{th} example has an attribute value of a_{jk} and belongs to class(k).

$$R(a_j, C_i) = \frac{\frac{1}{N} . \sum_{k=1}^{N} (a_{jk} - \mu_{a_j}) . (v_k - \mu_v)}{std(a_j)} \qquad \ldots 12$$

The variable v_k is the scalar value representation of the class and:

v_k =+1 if class(k) = C_i

v_k = -1 if class(k) < > C_i.

If the attribute a_j is not correlated with class C_i, equation 12 will return a value near zero indicating that a_j is not a useful attribute for classifying patterns of that class. Conversely if the attribute is highly correlated with the class C_i, the equation will return a value near to unity, indicating that the attribute is a powerful discriminator for that class.

4.0 Data Pre-Processing Strategies To Enhance Classification Accuracy

In this section we examine some practical approaches to formulating input transformations which will enhance class separation in a given problem. The transformations can all be viewed as variants of the linear diagonal, linear rotational, and product transforms which were introduced in section 2.

4.1 Zeroing The Mean of Pattern Attributes:

It is unusual for the mean of each of the pattern attributes used by a classifier to be zero. However, the mean value of an attribute, measured across all classes, carries no class discriminatory information. It is therefore possible to form a new set of attributes with zero mean value by subtracting the mean of each original attribute from each attribute value.

Using zero mean attributes is particularly desirable if a neural net classifier is being used because it avoids the need for the neural net bias weights to undergo many iterative updates until they compensate for the non zero mean of the input attributes. However, it should be emphasised that zeroing the means of pattern attributes does not effect the separability of the different classes in any way.

Formally, the process of zeroing the attribute means can be expresssed as follows. If the i^{th} attribute in the k^{th} example of a class j pattern vector is a_{ikj}, the mean value of the ith attribute over all examples of all classes is:

$$E\{ a_i \} = \frac{1}{\text{no classes}} \cdot \sum_{j=1}^{\text{no classes}} \frac{1}{\text{no examples in class j}} \cdot \sum_{k=1}^{\text{no examples in class j}} a_{ikj} \qquad ..13$$

And the new zero mean attributes are a_{ikj}:

$$a_{ikj} = a_{ikj} - E\{ a_i \} \qquad ...14$$

4.2 Normalising The Variance of Pattern Attributes:
The dynamic range and variance of each pattern attribute used in a particular classification problem can vary widely depending on the way in which the attribute measurements are made, and consequently some form of range normalisation is required.

For example, in an electronic fault diagnosis problem, the attributes might be a current measured in milliamps and a resistance measured in ohms. The range of the current might be 0 to 10 and the resistance 0 to 1000. If these attributes are used without normalisation by a classifier, such as k-nearest neighbour [5], which relies on measures of the Euclidean distance in the attribute space, the resistance attribute will have a disproportionate importance even though it may carry less discriminatory information than the current attribute. This is undesirable because the classification decision will be dominated by those attributes contributing most the Euclidean distance value.

In practice, explicit attribute normalisation should not be required if parametric statistical recognisers or neural nets are used. In the first case, the normalisation takes place implicitly because the attribute probability distribution is typically modelled by a Gaussian in which the exponent consists of the square of the attribute value *divided by its variance* .

In the case of a neural net classifier, discriminatory learning is used, and theoretically the net should eventually learn input weights which attenuate or excentuate pattern attributes in an optimal way regardless of their range. In practice, learning may be unreliable if the dynamic range of the input attributes is not normalised. This is because the magnitude of the learning step applied to each weight in the network is related to the magnitude of the inputs. If their range is too great, the learning becomes unstable, whereas if their range is very small, the learning becomes extremely slow.

In general it is therefore good practice to explicitly normalise the range of pattern attributes before they are applied to any type of classifier. In the absence of any information about the discriminatory significance of each of the attributes, it is intuively appropriate to normalise the variance of each attribute so that each contributes equally to the Euclidean distances on which classification is ultimately based. Thus, if the i^{th} attribute in the k^{th} example of a class j pattern vector is a_{ikj}, the variance of the i^{th} attribute over all examples of all classes is $V\{a_i\}$:

$$V\{a_i\} = \frac{1}{\text{no classes}} \cdot \sum_{j=1}^{\text{no classes}} \frac{1}{\text{no examples in class j}} \cdot \sum_{k=1}^{\text{no examples in class j}} (a_{ikj} - E\{a_i\})^2 \qquad ..15$$

And the set of new attributes with normalised variance is a_{ikj}:

$$\alpha_{ikj} = \frac{a_{ikj}}{\sqrt{V\{a_i\}}} \qquad ...16$$

This process of variance normalisation is an example of the diagonal linear transform described in section 2.2 in which the diagonal elements of the transform matrix are the reciprocals of the standard deviations of each or the raw pattern attributes.

4.3 Linear Scaling of Attributes:
The variance normalisation described in the previous section was proposed in cases where no information is available about the discriminatory power of each feature and it is therefore assumed that each attribute is equally important.

However, a method of measuring the usefulness of attributes on the basis of their mutual information has already been described in section 3.2 and the mutual information values can be used to furthur scale the variance normalised attributes to form compact class clusters. Attributes which exhibit high discriminatory power will have a high mutual information value and are scaled up in value so that they are contribute proportionately more to the Euclidean distances evaluated in the classifier. Conversely, attributes which contribute little to class discrimination will have a low mutual information and are consequently scaled down.

If the i^{th} attribute in the k^{th} example of a pattern vector of class C_j is a_{ikj}, the new attribute values are a_{ikj} which are given by:

$$\alpha_{ikj} = \frac{a_{ikj} . I(C_j, a_i)}{\sqrt{V\{a_i\}}} \qquad ...17$$

Once again, this pre-processing is a linear transformation in which the transform matrix only has non zero values on its diagonal.

The mutual information of the feature a_i is $I(S,a_i)$,which can be calculated using equations 9 and 10 once the required probability distributions have been estimated. Equation 9 shows that the equivocation of a feature a_i is given by:

$$H(C|a_i) = -\sum_{k=1}^{K} \sum_{l=1}^{l_r} P(C_k, a_i(l)) \log_2 P(C_k \mid a_i(l)) \qquad ..9$$

In this expression it is assumed that the attribute a_i has discrete values $a_i(l)$, and in some cases a_i will be naturally discrete valued, but often it will be a continuously valued quantity. In the latter case, the expression for mutual information can still be used if the attributes are quantised to a set of suitably spaced discrete levels. Ideally a very large number of levels would be used so that the discrete representation approximated to the continuous value. However, in practice the number of levels which can be defined will be limited by the number of available examples of the attribute and by the computational effort in evaluating equation 9.

Assuming that a_i is a continuously valued feature, the required discrete distribution $P(C_k \mid a_i(l))$ can be estimated using the following steps:

i) Divide the range of a_i into a L_i of uniformly spaced quantisation levels.

ii) Select the set of examples belonging to class C_k and quantise the values of a_i in this set.

iii) Count the frequency with which each quantisation level is used and normalise to the total number of examples. The resulting normalised frequency values are estimates of $P(a_i(l)|C_k)$.

iv) Count the number of examples of each class and normalise to the total number of examples to obtain an estimate of $P(C_k)$.

v) Evaluate $P(a_i(l))$:

$$P(a_i(1)) = \sum_{k=1}^{\text{no classes}} P(a_i(l) \mid C_k) \; . \; P(C_k) \qquad ...18$$

vi) Evaluate $P(C_k,a_i(l)) = P(a_i(l)|C_k) * P(C_k)$ ...19

vii) Evaluate $P(C_k|a_i(l)) = P(a_i(l)|C_k) * P(C_k) / P(a_i(l))$ ...20

4.4 Forming Rotational Transform Features:
The linear scaling of attributes described in previous sections does not explicitly take account of class separability nor does it reduce the dimensionality of the feature vectors used by the classifier. A more powerful form of pre-processing scheme is a linear rotational transform whose parameters are adjusted to maximise some measure of class separation given a feature dimensionality which is less than the original pattern dimensionality. A simple way of producing such a transform is by applying the raw pattern attributes to an MLP classifier which potentially can make its first layer weights equal the coefficients of an optimal rotational transform. The weights are adjusted using back propagation of the error between the actual and desired output of the MLP. Unfortunately, this is exactly the system which we have been arguing to be unreliable. Moreover, it does not yield a rotational transform which could be used with other types of classifier since the input transform implemented by the first layer in the MLP is matched to the subsequent non-linear transforms created by the hidden layers of the MLP.

A more appealing method of implementing a rotational transform which will lead to reduced dimensionality in the feature vectors and enhanced class separation is as follows. A feature vector is defined with M elements, f_i, and the number of elements is made equal to the number of pattern classes. The value of f_i is given by the weighted summation of the variance normalised pattern attributes, a_j, where the weighting coefficients are the correlation coefficients between the i^{th} class and the j^{th} pattern attribute.

$$f_i = \sum_{j=1}^{N} R(C_i, a_j) . \frac{a_j}{\sqrt{V\{a_j\}}} \qquad ...21$$

The important property designed into the transform is that an attempt is made to associate each class with a single feature vector element and that the contributions to that element from each of the pattern attributes are weighted in proportion to their correlation with the class. Thus pattern attributes which are uncorrelated with class C_i will make no contribution to the feature value associated with C_i.

The correlation coefficients are evaluated by the method described in section 3.3 and the variance of each attribute is given by equation 15 .

4.5 Forming Product Features:
Section 2.4 discussed the need for non-linear transforms to enhance class separation and to cause disjoint class regions in the pattern space to become merged into one contiguous region in the feature space. A simple non-linear transform which is known to be useful if the class type is defined by the relationship between attribute values rather than their individual absolute values is the product transform in which the elements of the feature vector, f_i, are given by the product of selected pattern attributes such that: $f_i = a_r . a_s . ..... a_t$.

The number of possible features which can be generated in this manner is immense and there is a problem in choosing which attributes should be selected to form part of a product feature. The most straightforward method of finding suitable product features is by an exhaustive search in which the mutual information or class correlation coefficients of all possible 2-tuple products are first measured, followed by measurements on all possible 3-tuple products and so on. In practice, the size of the search is likely to make it difficult to explore the behaviour of products whose order is greater than two.

5.0 Case Example of Data Analysis And Pre-Processing

5.1 Description of The problems:
Two sets of data are used as case examples. The first data set concerns fault diagnosis of electronic circuit cards and consists of a set of 77 pattern attributes which are measurements of current, voltage and resistance at various points in the circuit, along with a fault diagnosis performed by a human expert. There are eleven types of faults possible, corresponding to the failure of individual components such as relays and fuses.

The complete data set consists of 290 examples of faulty cards, each example consisting of a vector of 77 real valued pattern attributes, (the measurements), followed by a one out of eleven code to represent the type of fault. This data is characterised by very wide variations in range for the different measured attributes as well as a wide range in relative importance in discriminating between the different fault classes.

The second data base concerns the behaviour of a financial system. Each example consists of sixteen real valued attributes of the system along with a single two valued class attribute. The complete data base contains 980 different examples of seventeen dimensional vectors.
Some of the real valued pattern attributes used in this data are arbitrary encodings of symbolic quantities in the raw data and any classifier must therefore deal with the peculiarities of the chosen encoding, moreover, the scalings of the different attributes is quite arbitrary in the raw data, and this presents a complex task to the classifier.

The pre-processing and data analysis techniques described earlier in this paper have been applied to both these databases in order to gain insight into the structure of the data and hence determine the suitability of different types of classification algorithm to the data. We commence the investigation by analysing the raw form of both data bases using measurements of the correlation between each of the pattern attributes and the class of the pattern, the mutual information of each of the attributes, and finally Sammon's Mapping technique.

5.2 Analysis of Raw Financial Data:
Figure 1 shows the values of the correlation between each of the sixteen attributes of the financial data and the class attribute. To provide a comparative scale in these results, the correlation of the class attribute with itself is shown as well as its correlation with the input attributes. The self correlation of the class attribute is, of course, unity.

Figure 1 shows very clearly which attributes are important in determining the class of the pattern, and could be used as the basis of attribute selection if it was desired to reduce the dimensionality of the pattern examples. Alternatively, the coefficients could be used to decide on which attributes should be collected in future.

Figure 2 shows the mutual information values of the attributes of the financial data. It should be noted that this is a two class problem with roughly equal probabilities of occurrence for each class. The entropy of the classification is therefore 1 bit. Figure 2 shows that attribute 5 has a mutual information of 0.9 bits which implies that, using this attribute alone for classification, would leave a residual classification entropy of only 0.1 bits. This can be interpreted as a probability of correct classification of $2^{-0.1}$.

Finally, we examine the distribution of the pattern examples in the pattern space using Sammon's Mapping technique. This is presented in Figure 3 which shows very clear separation of the two classes of pattern. In fact, the figure indicates that a simple linear discriminant function classifier would successfully classify all the examples in this data set.

5.3 Analysis of The Raw Fault Diagnosis Data:

The same analysis techniques were applied to the raw fault diagnosis data. Due to the difficulty of plotting correlation and entropy values for all 77+11 attributes in this data set, only the results for the first 53 pattern attributes and 11 class attributes are presented. These results adequately show the form of the data.

Figure 4 shows the correlation coefficients between the pattern attributes and an arbitrarily chosen class attribute, class attribute 9. In this data it is evident that no single attribute is very strongly correlated with the class attribute. One point of interest is that significant correlations exist between the class attributes. Specifically, class attribute 5 is negatively correlated with class attribute 9, implying that the occurrence of a class 5 fault can be used as evidence against the occurrence of a class 9 fault on the same circuit.

Figure 5 shows the mutual information values of the attributes of the fault diagnosis data. In this case there are eleven classes and if these were equiprobable, the entropy of the classification would be between 3 and 4 bits. Examining the attribute mutual information values, it is evident that no single attribute has a mutual information greater than 0.83 bits, and hence only poor classification would be expected by use of any individual attribute. It also suggests that it may be necessary to form new, more discriminatory features, if accurate classification is required. Such features might be formed from the non-linear transformation of the raw data.

Figure 6 presents the Sammon Map for the raw data. To simplify the map, the data has been re-defined as belonging to class 9 or not belonging to class 9. Evidently the data consists of a large number of examples of both classes which are very close together, and a few rogue points which are relatively far apart. The automatic normalisation of scales in the Sammon Mapping routine means that it is not possible to see if the majority of data examples, which lie in the bottom left corner of the map, form separate class clusters.The separability of the class clusters will be made more apparent in subsequent sections in which mappings of normalised data are presented.

5.3 Analysis of Variance Normalised Financial Data:
It has previously been argued that in the absence of any information about the relative importance of different pattern attributes, their variances should be normalised prior to use by a classifier. Variance normalisation will not change the correlations between pattern attributes and class attributes, nor will it effect the mutual information values of the attributes. However, it will cause the class clusters to become more less elongated and hence make it easier to visualise their intrinsic separability. This is demonstrated in Figure 7 which shows the Sammon Mapping for the variance normalised financial data. Comparison with the Sammon Map for the raw data shows that the clusters in the normalised data are more uniformly shaped, and in some cases this may make the data more easily separable. However, this is not guaranteed.

5.4 Analysis of Normalised Fault Diagnosis Data:
Similar results are shown for the normalised fault diagnosis data in the Sammon Map of Figure 8. Again, the data has been re-defined as class 9 or not class 9 to simplify presentation. The map shows that the classes cluster rather poorly, and that there are several non-contiguous regions associated with each class. This implies that simple linear classification would not work well and that either a non-linear classifier such as an MLP would be required, or that new features should be formed by the non-linear transformation of the original attributes.

5.5 Analysis of Mutual Information Scaled Financial Data:
We have previously argued that scaling pattern attributes by their mutual information (MI) values before they are used by a classifier can improve class cluster separability. This is demonstrated by the Sammon Map of Figure 9 for the financial data after MI scaling. Comparison of this map with the map for the unscaled data shows the dramatic improvement in class separation which is possible with this technique. A practical consequence of pre-processing data in this manner, would be a reduction in the complexity required of the subsequent classifier, and hence better generalisation performance.

5.6 Analysis of Mutual Information Scaled Fault Diagnosis Data:

Unfortunately, MI scaling cannot ameliorate the problem of class cluster non-contiguity which was evidenced in the Sammon Map of the fault diagnosis data in Figure 8. Figure 10 shows the Sammon Map of the same data after MI scaling and there is no obvious change in the separation of the class examples.

5.7 Analysis of Rotationally Transformed Financial Data:
It is often desirable to rotationally transform a pattern set so as to enhance class separability and reduce the dimensionality of the resulting feature space in which classification will be performed. We have suggested that a simple rotational transform is given by using the coefficients of correlation between of the pattern attributes and each of the class attributes. In the case of the financial data, there are only two classes and so the transform maps the 16 pattern attributes to just one feature on which classification can be performed. The usefulness of the resulting feature is demonstrated by measuring its correlation coefficient with the class attribute as shown in Figure 11. In this particular case, the correlation of the new feature is 0.89 which is marginally less than the correlation between the best pattern attribute and the class. However, other experiments have shown that if the input pattern to the transformer only consists of attributes having rather low class correlation, the class correlation of the output of the transform is usually much higher than the correlations of the individual pattern attributes. The transform is therefore able to combine the discriminatory power of individual attributes into a single feature.

The Sammon Map of the one Dimensional feature generated by the rotational transform of the financial data is presented in Figure 12 which indicates that, with the exception of three pattern examples, the classes are completely separable in the new one dimensional feature space. This slight reduction in separability might be a small penalty to exchange for the reduction in classifier complexity which results from using the low dimensional feature.

5.8 Analysis of Rotationally Transformed Fault Diagnosis Data:
Similar results are presented for rotationally transformed fault diagnosis data in Figures 13 and 14. Figure 13 shows the correlation coefficient between the single dimensional feature formed by rotationally transforming the 77 dimensional pattern attribute vector, and comparison with the correlation coefficients of the individual pattern attributes in Figure 4 shows a slight improvement. Figure 14 shows the Sammon Map for the transformed data from which it can be seen that class separation is slightly improved compared to the untransformed data shown in Figure 8.

5.9 Analysis of Product Features Formed From Attributes of Financial Data:
We have argued earlier in the paper that the class of a pattern is frequently determined by the relationship between its attributes rather than their absolute values. This suggests that useful class information might be gained from new features which are formed from the products of the original pattern attributes. Clearly, the number of possible product terms is enormous, and so we only aim to illustrate the process by selecting features which are formed by the product of one arbitrarily chosen pattern attribute with all the other attributes. It is important to use pattern attributes whose mean value has been set to zero, so that the new product features contain no components of the original attributes. In the case of the financial data, the new features are formed from the product of attribute 1 with all other attributes.

Figure 15 shows the class correlations for the resulting features, which although small in this case, could be usefully used in conjunction with the original attribute set to improve class separation. Figure 16 shows the corresponding Sammon Map, from which it can be seen that some class separation is possible using the new product features by themselves.

5.10 Analysis of Product Features From Attributes of Fault Diagnosis Data:
Similar experiments have been performed on features formed from the products of pairs of pattern attributes in the fault diagnosis data. Again, the chosen product pairs are pattern attribute 1 with all other attributes. The correlation coefficients for the product features are shown in Figure 17 from which it can be seen that the correlation for some of the new features is slightly higher than that of the original attributes. This is an important result because it suggests that class separation, and hence classifier performance, could be

significantly improved by selecting the best product features and best original features and using them in combination.

Confirmation of the usefulness of the product features from this data base is given by their mutual information values which are shown in Figure 18. These results show that one of the product features has an MI value which is significantly higher (1.0 bits) than the MI values of any of the original attributes.

6.0 Conclusions

This paper has argued that the type of data which is generated in decision support problems is often ill suited to conventional statistical classifiers because it contains a mix of symbolic and scalar values, the dynamic range of the data may be unlimited, and it does not always change smoothly as the state of the system generating the data changes.

In recent years, a popular solution has been to use artificial neural networks for classifying such data because the neural net performs discriminatory learning and can theoretically learn to suitably pre-process the data internally before classification. In practice, such learning is unreliable using currently available training algorithms and a better approach may therefore be to use a very simple neural network or statistical recogniser to which is input explicitly pre-processed pattern data.

This paper has reviewed the pattern classification problem and examined various pre-processing strategies which could be used to simplify the classification of a given data set. In particular, the use of normalisation, mutual information scaling, rotational transforms and product features have been considered.
In order to assess the effects of these pre-processing schemes, a number of data structure analysis tools have been presented which can provide insight into the structure and "classifiability" of a data set.

The use of the pre-processing strategies and the data analysis tools has been demonstrated using two case examples. The first is data concerning a financial system and the second involved the diagnosis of faults in an electronic system. In both cases, it has been shown that significant improvements in class separability are possible using quite simple and explicit pre-processing.

7.0 References

1. Totton K.A, Limb P.R, *Experience in using neural networks for electronic diagnosis.*
Proc. IEE 2nd Int. Conf. On Artificial neural Networks, pp115-118, Bournemouth,
November 1991.

2. Andrews H.C , *Introduction to Mathematical Techniques in Pattern Recognition,* Kreiger, 1983.

3. Abramson N, *Information Theory and Coding*, McGraw-Hill, 1963.

4. Sammon J.W, *A Non Linear Mapping For Data Structure Analysis*, IEEE Trans. on Computers, vol C-18, no 5, May 1969.

5. Wilpon J.G, Rabiner L.R, *A modified K-means Clustering Algorithm for Use in Isolated Word Recognition*, IEEE TRans. on Acoustics, Speech and Signal Proc., vol 33, pp587-594, 1985

6. Maxwell,T., Giles,C.,Lee Y.,*Generalisation in Neural Networks, the Contiguity Problem.*
Proc IEEE First Int. Conf. on Neural Networks, vol. 2, pp41-45, 1987.

7. Durbin D.,Rumelhart D., *Product Units: A Computationally Powerful and Biologically Plausible Extension to Backproagation Networks*, Neural Computation, no.1, pp133-142,1989.

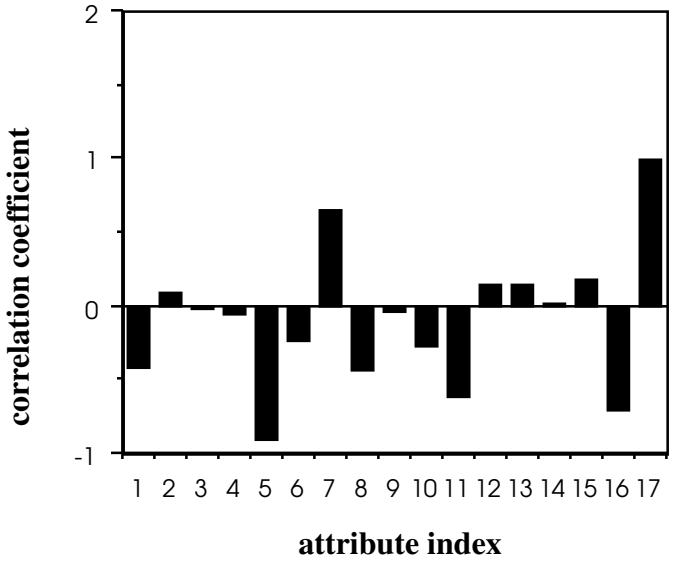

Fig1. Class - Attribute Correlation Coefficients of Unprocessed Financial Data

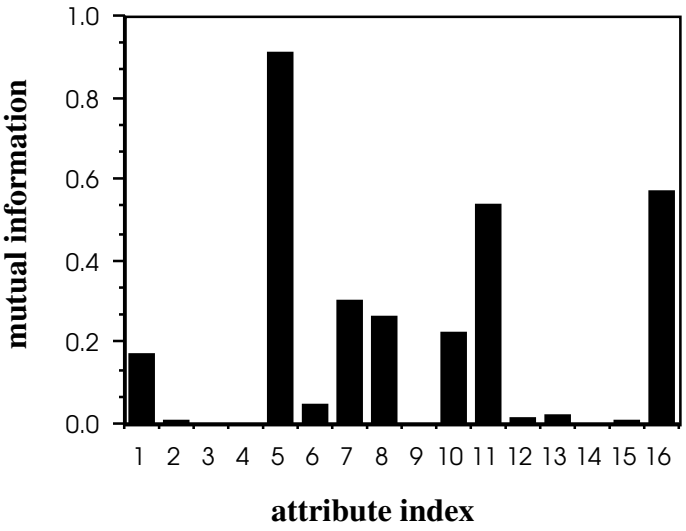

Fig 2. Mutual Information of The Attributes of The Financial Data

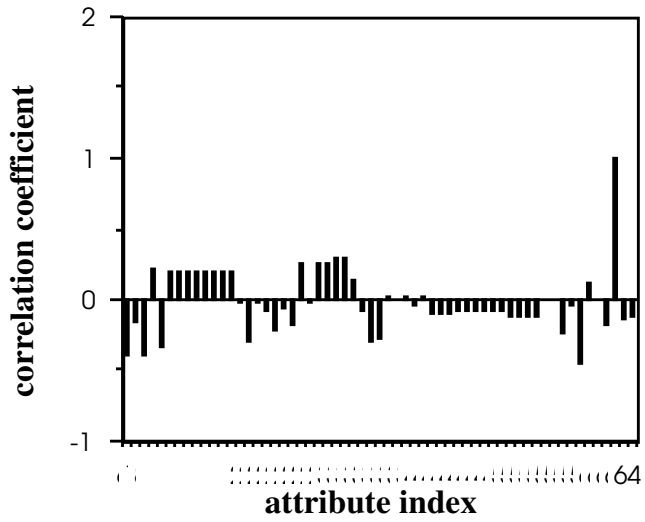

Fig4. Class - Attribute Correlation Coefficients of Unprocessed Fault Diagnosis Data

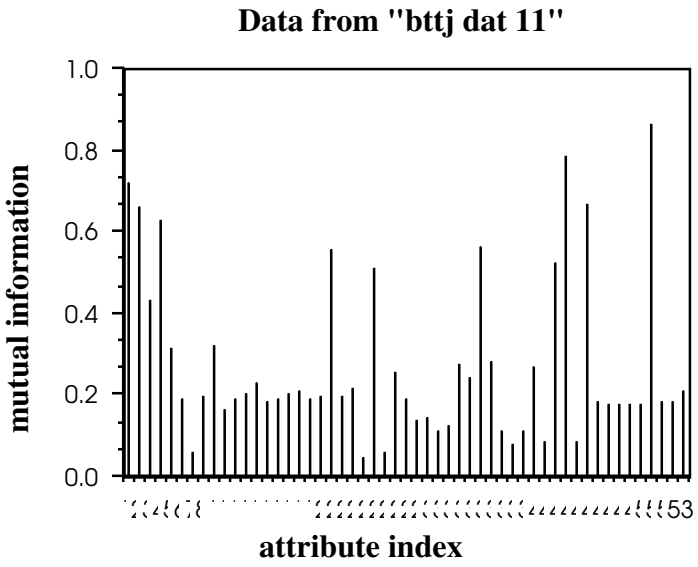

Fig 5. Mutual Information of The Unprocessed Attributes of The Fault Diagnosis Data

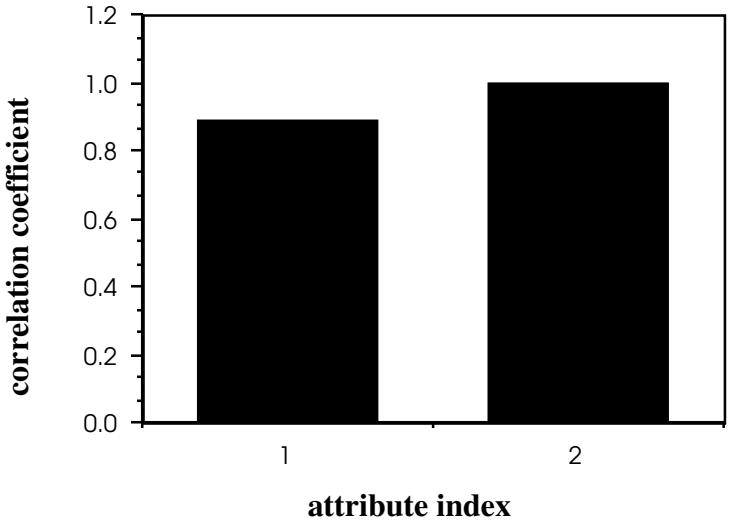

Fig 11. Class - Attribute Correlation Coefficients of Rotationally Transformed Financial Data

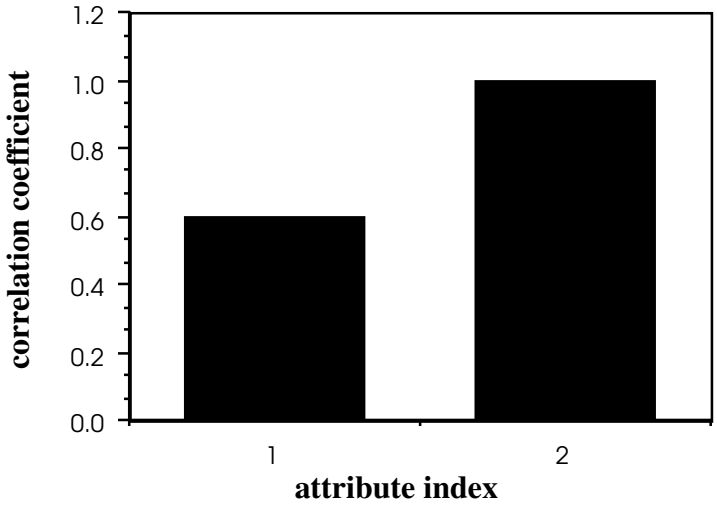

Fig 13. Class - Attribute Correlation Coefficients of Rotationally Transformed Fault Diagnosis Data

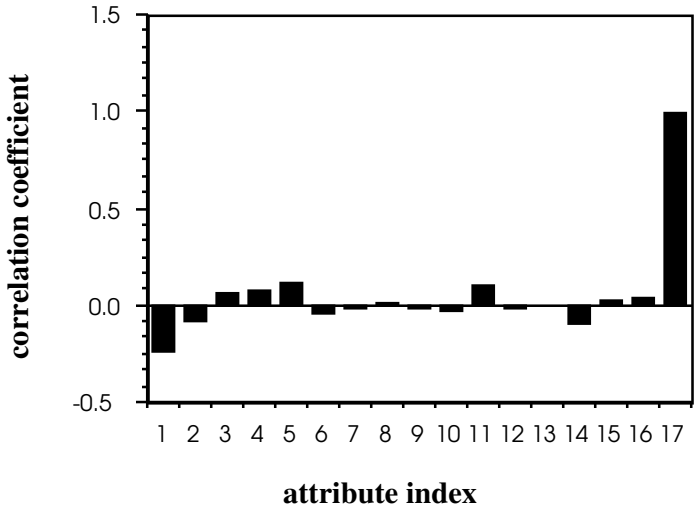

Fig 15. Class - Attribute Correlation Coefficients of Attributes Formed From the Product of Attribute 1 and All Other Attributes of Financial Data

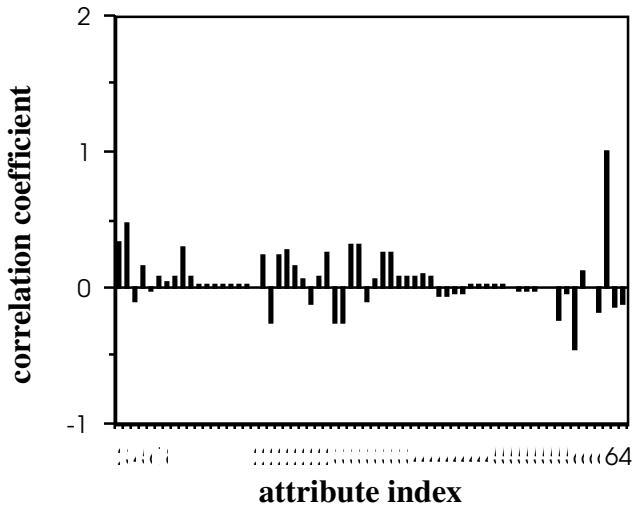

Fig 17. Class - Attribute Correlation Coefficients of Attributes Formed From the Product of Attribute 1 and All Other Attributes of Fault Diagnosis Data

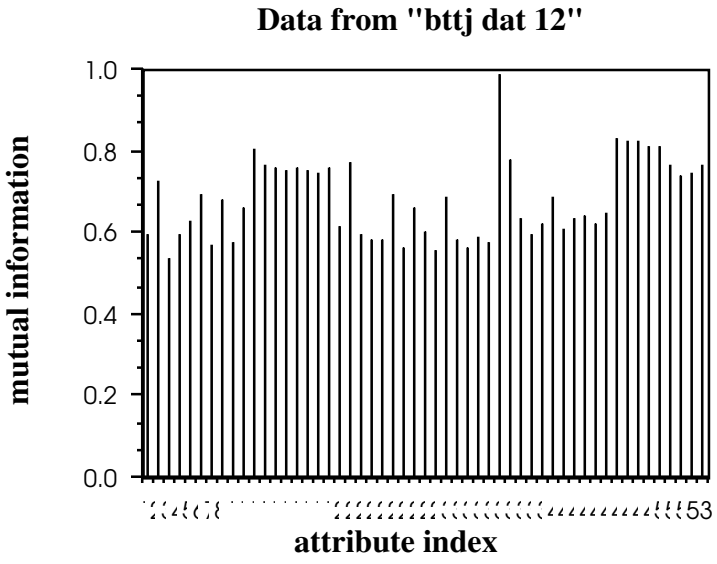

Fig 18. Mutual Information of Features Formed From The Product of Attribute 1 and All Other Arributes in the Raw Fault Diagnosis Data.