PATTERN RECOGNITION USING ROBUST DISCRIMINATION AND FUZZY SET THEORETIC PREPROCESSING

BY

NICOLINO JOHN PIZZI

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION PAGE

PATTERN RECOGNITION USING ROBUST DISCRIMINATION AND FUZZY SET THEORETIC PREPROCESSING

BY

NICOLINO JOHN PIZZI

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirements of the degree of DOCTOR OF PHILOSOPHY

Dedication

I dedicate this thesis to:

My brother, Tony,
in memory of his gentle spirit that touched the hearts of many,

and to

My sons, Anthony and Michael,
for touching my heart.

Abstract

Classification is the empirical process of creating a mapping from individual patterns to a set of classes and its subsequent use in predicting the classes to which new patterns belong. Tremendous energies have been expended in developing systems for the creation of the mapping component. Less effort has been devoted to the nature and analysis of the data component, namely, strategies that transform the data in order to simplify, in some sense, the classification process. The purpose of this thesis is to redress somewhat this imbalance by introducing two novel preprocessing methodologies. Fuzzy interquartile encoding determines the respective degrees to which a feature belongs to a collection of fuzzy sets and subsequently using these membership grades in place of the original feature. Burnishing tarnished gold standards compensates for the possible imprecision of a well-established reference test by adjusting, if necessary, the class labels in the design set while maintaining the test's vital discriminatory power. The methodologies were applied to several synthetic data sets as well as biomedical spectra acquired from magnetic resonance and infrared spectrometers.

Both fuzzy encoding and burnishing consistently improved the discriminatory power of the underlying classifiers. They are insensitive to outliers and often reduce the training time for iterative classifiers such as the multi-layer perceptron. With the latter, reclassification only occurs for data within the design set; outliers within the test set are flagged but not altered. Therefore, the accepted gold standard is left in a pristine state sullied only by its original tarnish.

Acknowledgement

I would like to thank Dr. Pedrycz for his infinite patience, gentle prodding, and guidance as I marshaled my intellectual forces to complete these investigations. His penetrating insights tempered by his subtle humour will always be appreciated.

Also, I want to express my gratitude to Dr. Walton, Dr. Roventa, and my good friend, Dr. Barakat, for their thorough review of my work and their useful suggestions.

And, of course, many thanks to my wife, Cindy. Without her love, faith, and support, none of this would have been realized.

Table of Contents

List of Tables

List of Figures

1 Introduction

Classification is the empirical process of creating a mapping from individual patterns to a set of classes and its subsequent use in predicting the classes to which new patterns belong. Tremendous energies have been expended, with some measure of success, in developing systems, and methodologies, for the creation of the mapping component. Less effort has been devoted to the nature and analysis of the data component, namely, strategies that transform the data in order to simplify, in some sense, the classification process. The purpose of this thesis is to redress somewhat this imbalance by introducing new transformational techniques that deal specifically with the data component.

This thesis argues that in practical pattern recognition problems preprocessing seems to be of paramount importance. Advanced technologies contribute ever more sophisticated models upon which to build ever more sophisticated classifiers. Herein lies a major problem: if these models are highly non-linear, they may be unstable, if they are iterative, they may not converge, if they are probabilistic, they may be based on underlying statistical assumptions that are often not true in real-world scenarios. Preprocessing may address these concerns: data may be transformed such that a non-linear model may be replaced by a linear one, the dimensionality of the data may be reduced so that an iterative method may converge or may be substituted for an analytic one, or the data may be "normalized", in some sense, such that the underlying statistical assumptions of a probabilistic model are realized. Years of investigations into pattern recognition problems have led this researcher to conjecture that the 80/20 rule holds in the construction of good classification systems: 20% of the investigator's time should be spent on selecting and tuning a classifier for a particular pattern recognition problem; the initial 80% should be spent on a thorough analysis of the data in order to preprocess it in such a way as to simplify, in some sense, the data that is to be presented to the classifier of choice. To this end, this thesis presents two novel preprocessing methodologies:

- *fuzzy encoding*, the process of determining the respective degrees to which a datum belongs to a collection of fuzzy sets or fuzzy clusters and subsequently using these membership grades in place of the original datum;
- *burnishing tarnished gold standards*, compensating for the possible imprecision of a well-established reference test while maintaining its vital discriminatory power.

Three new fuzzy encoding strategies and three respective variants are presented:

- *fuzzy interquartile encoding*, intervalizing a single input value across a collection of fuzzy sets, thereby producing a list of degrees of membership for each of the fuzzy sets;
- *dimension-preserving fuzzy interquartile encoding*, a variant of fuzzy interquartile encoding that does not increase the dimensionality of the feature space;
- *fuzzy cluster encoding*, transforming the input space using a membership measure to determine how similar an individual is to centroids computed using the fuzzy *c*-means algorithm;
- *class-wise variants*, identical to the above methods except that they take into account class assignments for the data set.

Two new strategies for burnishing tarnished gold standards are presented that may be used independently or may augment the fuzzy encoding methods:

- *robust reclassification*, uses a robust estimation of deviations from class medoids for the reclassification of spectra in a design set;
- *fuzzy gold standard adjustment*, a fuzzy set theoretic preprocessing method to enhance the gold standard by incorporating non-subjective within-class medoid information.

In order to properly discuss these methodologies, this thesis will begin with a preliminary introduction (chapter 2) to some key concepts necessary for their understanding: *classification*,

fuzzy set theory, *robust statistics*, and *artificial neural networks*. Chapter 3 is devoted to the *classification process*, a multi-faceted exercise, realized through a *classification system*. In the most general sense, the system creates a discrimination function mapping individuals to a set of class indices. This chapter also discusses verification issues and presents the data sets that will be used to test the efficacy of the methodologies. While some data sets are synthetic, "real-world" data were also acquired from the biomedical domain.

Specific classifiers are revisited in chapter 4, including linear discriminant analysis, a classical multivariate discrimination technique, as well as several artificial neural network (ANN) architectures: multi-layer perceptrons, probabilistic neural networks, and radial basis function neural networks. Chapter 5 then presents two conventional preprocessing methods: adjustments to the receptive fields of the radial basis function neural network and principal component analysis. In the former case, standard techniques are discussed to determine the location, size, and interaction of the local receptive fields used in the radial basis function neural network. The motivation behind the latter method is to find a set of orthogonal directions that explain as much of the variability of the original data as possible.

Chapter 6 thoroughly discusses fuzzy data encoding and burnishing tarnished gold standards. It begins with a mathematical description of fuzzy interquartile encoding, dimension-preserving fuzzy interquartile encoding, fuzzy cluster encoding, and their class-wise variants. Integration of these methods into a classification system is then presented. A specific classifier, the multi-layer perceptron (MLP), is used. Robust reclassification and fuzzy gold standard adjustments are then presented as well as the motivational differences between reclassification and adjustment.

A set of experiments using synthetic data are performed in Chapter 7 in order to measure the efficacy of the novel preprocessing methods described in the previous chapter. All fuzzy encoding methods are applied to two-class 1-dimensional data with different distributions. Fuzzy encoding and linear separability issues are also presented. The burnishing methods are also tested

using some of these data sets along with contaminated counterparts, that is, data sets where some individuals were intentionally mislabeled. This chapter also explores the performance of fuzzy interquartile encoding when integrated with an artificial neural network, namely, the MLP. The chapter concludes with performance measures for all of the novel strategies using several other synthetic data sets.

A set of experiments using "real-world" data is performed in Chapter 8. These data are from the biomedical domain: infrared and magnetic resonance spectra of human tissue. The gold standard is a pathologist's report concerning the disease state of the tissue specimens. The novel preprocessing methods are applied to these data and are benchmarked against some conventional preprocessing and classification strategies.

2 Preliminaries

A number of essential concepts must be discussed in order to understand properly the nature and intent of the novel preprocessing methodologies presented in this thesis. This chapter begins with a background of the classification process and a typical methodology, artificial neural networks, which may be used to create a classification mapping. As these new techniques involve concepts from fuzzy set theory and robust statistics, overviews of these topics are also presented.

2.1 Classification

Classification, or discrimination, systems involve the process of finding a function mapping N individuals to an index set of k class identifiers, ω_i (i=1,...,k). The individuals normally take the form described in Table 1 where N_i is the total number of individuals in class ω_i. Each individual (case, sample, pattern, vector, or point) comprises n features (measurements or coordinates) and belongs to some class, ω_j. To be more precise, a distinction can be made between *discrimination* and *verification*.

Discrimination is the process of determining a decision rule that partitions the individuals' space into k regions, Ω_i such that if an individual belongs to the class, ω_j, it will also lie in the region Ω_j. For example, Figure 1 is an example of a 2-dimensional 2-class problem with the decision rule, $x_2-x_1=0$, that divides the space into two regions, Ω_1 and Ω_2. If $x_2-x_1<0$ then the individual lies in Ω_1 and is classified as coming from ω_1, otherwise it lies in region Ω_2 and is classified as coming from ω_2. This is a *linearly separable* problem: an n-dimensional hyperplane can be defined that serves as a decision boundary between two classes of n-dimensional individuals. In this specific case, the line, $x_2-x_1=0$, is the decision boundary that divides individuals belonging to ω_1 from those belonging to ω_2.

Since the intent of discrimination is to group classes into regions, the concept of similarity must be quantified. Similarity is often measured using the Euclidean metric

$$d(x,y)=\sqrt{\sum_i (x_i - y_i)^2} \qquad (1)$$

not only for historical reasons but also for analytic ones since it has a derivative at every point (x and y are individuals and x_i and y_i are their respective features). If $d(x,y)$ is near zero, the two individuals are said to be similar. Other similarity measures may also be used.

Individual	**Features**				
	x_1	x_2	...	x_n	
1	x^1_{11}	x^1_{21}	...	x^1_{n1}	Class 1
2	x^1_{12}	x^1_{22}	...	x^1_{n2}	
⋮	⋮	⋮	⋮	⋮	
N_1	$x^1_{1N_1}$	$x^1_{2N_1}$	...	$x^1_{nN_1}$	
1	x^2_{11}	x^2_{21}	...	x^2_{n1}	Class 2
2	x^2_{12}	x^2_{22}	...	x^2_{n2}	
⋮	⋮	⋮	⋮	⋮	
N_2	$x^2_{1N_2}$	$x^2_{2N_2}$	...	$x^2_{nN_2}$	
⋮	⋮	⋮	⋮	⋮	⋮
1	x^k_{11}	x^k_{21}	...	x^k_{n1}	Class k
2	x^k_{12}	x^k_{22}	...	x^k_{n2}	
⋮	⋮	⋮	⋮	⋮	
N_k	$x^k_{1N_k}$	$x^k_{2N_k}$	...	$x^k_{nN_k}$	

Table 1: Typical data for discrimination

Verification, on the other hand, specifically refers to the application of the decision rule to a new individual of unknown class. In the case of Figure 1, the individual denoted by ϑ would be classified as belonging to class ω_1 since it lies in Ω_1. The discrimination process typically employs a subset of the N individuals, specifically; the defining parameters of the decision rule are estimated from this subset. This subset is known as the *design* (or training) set. The verification process uses the remaining individuals, the *test* set to measure the efficacy of the decision rule. A typical measure of the error rate is to divide the number of individuals that lie in a region that do not belong to the corresponding class by the total number of individuals.

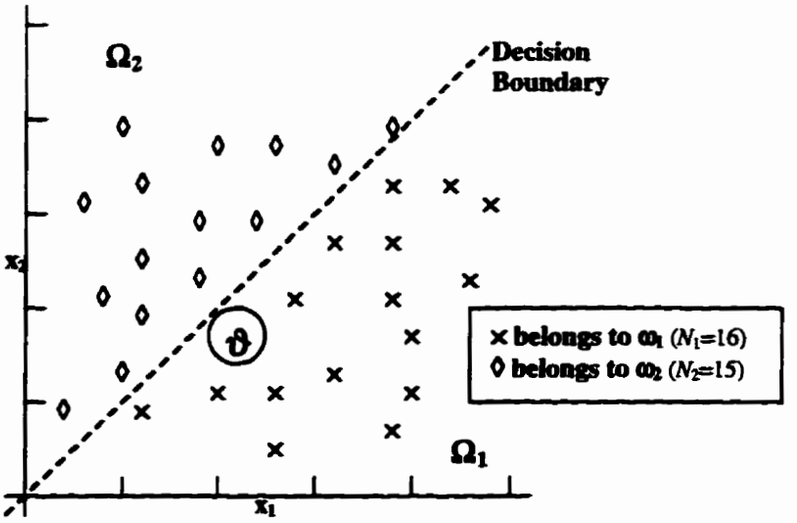

Figure 1: Decision rule, $x_2-x_1=0$, for a 2-dimensional 2-class problem

A number of problematic situations may arise in the discrimination/verification process. For example, the discrimination problem shown in Figure 2 is identical to the one in Figure 1 except that one of the original ω_2 individuals now belongs to ω_1. If the decision rule that produced the original boundary is used here then the suspect individual will lie in the wrong region, Ω_2. This occurs because the problem is no longer linearly separable. If we use the original boundary then misclassifications will occur with individuals in the test set that are near the suspect individual.

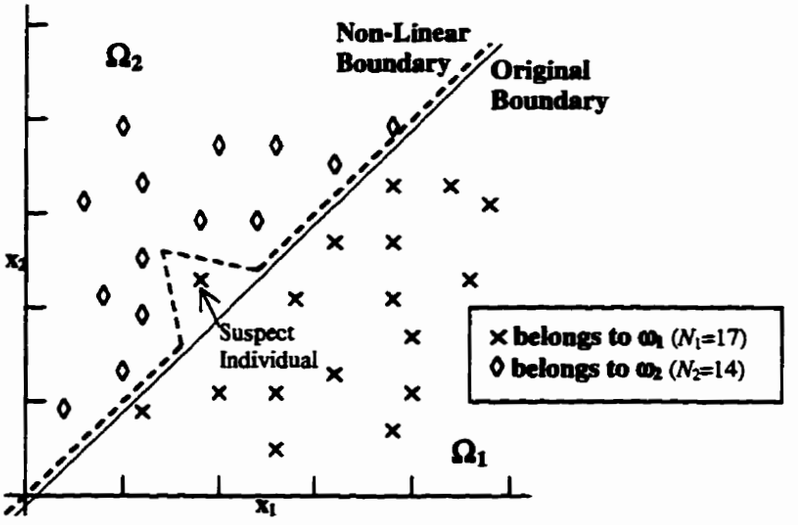

Figure 2: Possible effects on a decision rule by a suspect individual

One solution that would reduce these misclassifications is to use a discrimination method that can produce a non-linear decision rule. In this example, a non-linear boundary can be derived

using a piece-wise linear decision rule. However, suspect individuals may cause problems that are more serious. Figure 3 has the suspect individual further from the original decision boundary. Discrimination methods that produce piece-wise linear decision rules may not work well in this case. Fortunately, other methods exist that can produce decision boundaries that are arbitrarily complex. In this case, such a method could produce a non-linear rule that creates the original boundary as well as a spherical Ω_1 boundary disjoint from the other Ω_1 region. This may be a good solution especially if there is a significant number of other individuals belonging to the class ω_1 near the suspect individual: this would indicate that there are two distinct *clusters* of ω_1 and a more complex decision boundary is therefore required. However, a problem may exist with the suspect individual itself. For instance, an error may have occurred during the measurement process for this individual such that its features were incorrectly recorded. In this case, the suspect individual may be an *outlier* and should not be used during the discrimination process. In this example, if the suspect individual is indeed an outlier and were removed, the problem once again becomes linearly separable. Outlier detection and removal is a standard practice in statistics but there are pitfalls. If there is a paucity of individuals upon which to build a decision boundary, removing an individual that is considered an outlier may not be possible. Detecting outliers in a high dimensional space is an extremely difficult problem (see section 2.4). The individuals that are identified as outliers may not be outliers at all and their existence warrants the use of non-linear decision rules. Finally, the suspect individual's features may have been measured accurately but it may have been mistakenly assigned to a wrong class. In this case, the individual is not an outlier: if the suspect individual in Figure 3 were reclassified as belonging to ω_2 then this problem would again be linearly separable (in fact, it would be the identical problem to Figure 1).

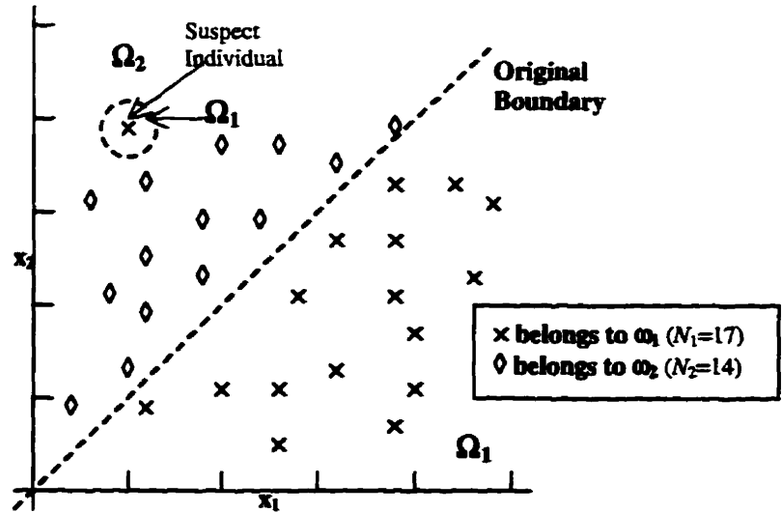

Figure 3: A suspect individual as an outlier

The next few sections will discuss systems that are used to derive decision rules and auxiliary techniques that simplify the data in some fashion in order to produce more succinct decision rules or better discrimination boundaries. The focus of the remainder of this thesis is presented in the final section of this chapter. In the interest of brevity, and since this thesis is concerned with both the discrimination and verification processes, the term classification will be used to refer to both processes.

2.2 Artificial Neural Networks

The artificial neural network paradigm [14,18,40] has consistently demonstrated its effectiveness as a reliable nonlinear classification technique. An ANN is a self-adaptive, massively parallel machine learning system composed of layers of processing elements used primarily for pattern recognition problems. A processing element (PE) is a construct composed of a set of inputs and corresponding weights (input connection strengths) that are combined to produce a result that is passed to a transfer function (used to constrain output to a particular range) ultimately generating an output value that may be used by other PEs. Typically, an ANN is composed of three types of layers: an input layer that passes data vectors to other layers; an output layer that produces an output vector (this vector often represents the classification outcome for the corresponding input vector); and the hidden layers that take data from an input layer or a

previous hidden layer and pass the transformed data to an output layer or a subsequent hidden layer. A learning strategy is used to make incremental changes to the weights in order to optimize some error criterion.

A supervised ANN [95] requires the desired output for each input vector in order that it may be compared to the actual output generated by the ANN (Figure 4). The learning strategy attempts to minimize a global error function for the set of design data. Local errors are computed for each PE in order to adjust the weights. This process is repeated for each input vector in the design set and the ANN continues to iterate through the set until an acceptable minimization of the error is achieved. The back-propagation algorithm is the most common technique used to pass this error back to the network. A feed-forward ANN has an uni-directional data flow from the input layer, through each hidden layer, and finally to the output layer. In other words, no PE may pass its output to a PE in a previous layer nor may it pass the output back to itself (feedback).

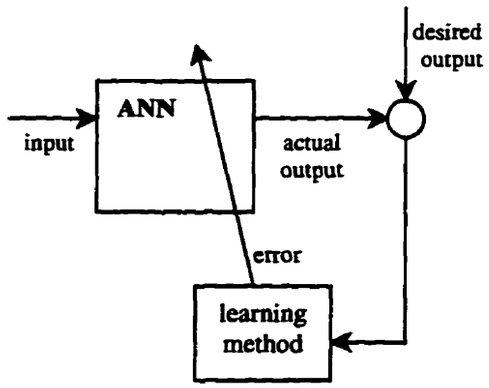

Figure 4: A supervised artificial neural network

2.3 Fuzzy Set Theory

Fuzzy set theory (FST), a generalization of conventional or Boolean set theory, was introduced by Zadeh [123] as a natural and intuitively plausible way to represent vagueness in everyday life. A central generalization of FST is the extension of the notion of elementhood from the range {0, 1} to the entire unit interval [0, 1] [84]. Conventional sets are *crisp*; elements in the universe of discourse must satisfy precise properties required for membership. Let us examine an example

where the universe of discourse is the set of real numbers and the set, A, that is to be defined is the set of numbers that are approximately zero. Conceptually,

$$A = \{x \in \Re | x \cong 0\}. \quad (2)$$

Using a conventional definition for this set, one must first define the upper and lower crisp limits for this set. These limits are, of course, domain-specific. Say, in this example, the limits are ±0.5, then

$$A = \{x \in \Re | -0.5 \le x \le 0.5\} \quad (3)$$

Equivalently, this Boolean set may also be described by its membership function, $A(x)$

$$A(x) = \begin{cases} 1 \; if \; -0.5 \le x \le 0.5 \\ 0 \quad otherwise \end{cases}. \quad (4)$$

Every real number, x, is either in A or it is not. More specifically, $A(x)$ maps all real numbers onto the two points {0, 1}. Hence, x is "approximately zero" if and only if $A(x)=1$. Unfortunately, this sharp transition between inclusion and exclusion is problematic when dealing with values immediately outside the transition (Figure 5). If $x=0.50001$, $A(x)=0$, hence, x is not "approximately zero". In many real-world applications, this sharp transition from truth to falsity is intuitively unappealing. Conceptually, the degree to which 0.50001 belongs to A is certainly not one ($A(x) \neq 1$) but it should be greater than zero, especially since it is approximately equal to 0.5. FST quantifies this gradual transition from falsity to truth by generalizing the membership function such that it maps values into the entire unit interval [0, 1] [63,124].

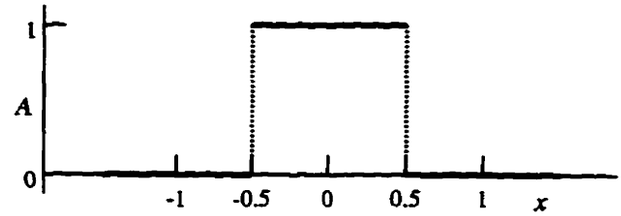

Figure 5: Membership function for the crisp set definition of "approximately zero"

A fuzzy set, F, contains objects that satisfy, possibly *imprecise*, properties to varying degrees. The value of the membership function $F(x)$ is known as the grade of membership of x in F. As

with Boolean sets, there is no unique domain-independent membership function for F. Some plausible properties for a fuzzy set are domain-independent, however. The first property is *normality*, at some point the grade of membership equals one (in the example, $F(0)$ should equal one). A fuzzy set should also satisfy the criterion of *monotonicity* [125]. Although not necessary, a fuzzy set may also satisfy the criterion of *symmetry*. Using our example, the former criterion simply means that as x approaches 0, $F(x)$ approaches 1 (the converse must also hold). The latter criterion is satisfied if numbers equidistant from 0 have the same membership grade. One membership function (see Figure 6) that satisfies the conceptual property "approximately zero" is

$$F(x)=\left(1+10x^2\right)^{-1}. \qquad (5)$$

Note that normality ($F(0)=1$), monotonicity ($F(0.5)=0.29$ and $F(0.1)=0.91$), and symmetry ($F(0.4)=0.38$ and $F(-0.4)=0.38$) are all satisfied by (5). Several advantages occur: conventional set theory is reduced by FST; FST represents vagueness in a more intuitively plausible manner using gradual transitions from falsity to truth; membership grades are more informative (in the example, $A(x_1)=1$ and $A(x_2)=1$ only indicate that x_1 and x_2 are both between –0.5 and 0.5 whereas $F(x_1)=0.88$ and $F(x_2)=0.95$ not only indicate that x_1 and x_2 are "approximately zero" but also that x_2 is "closer" to zero than x_1.

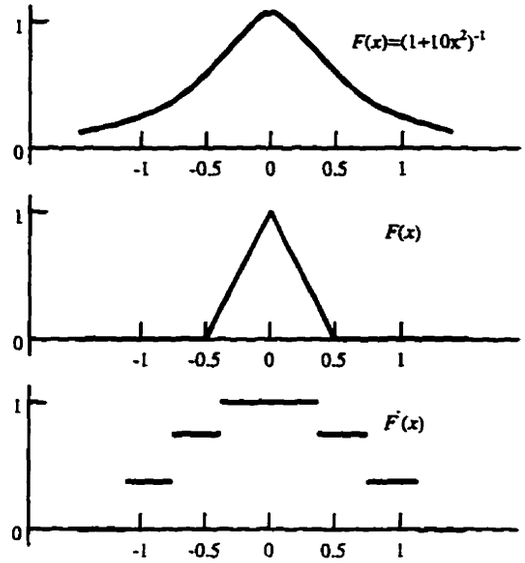

Figure 6: Possible membership functions for the fuzzy set "approximately zero"

The membership function is a measure of the degree to which an object satisfies imprecisely defined properties and in order to combine fuzzy sets a collection of operators must be defined [126]. Let $\Im(X)$ be the family of all fuzzy sets of the domain X and $A,B\in\Im(X)$. For all $x\in X$, the following operations may then be defined [60]:

Complement: $\tilde{A}(x)=1-A(x)$

Intersection: $(A\cap B)(x)=\min\{A(x), B(x)\}$

Union: $(A\cup B)(x)=\max\{A(x), B(x)\}$.

Note that there is no definition for the law of the excluded middle ($\tilde{A}\cap A=\varnothing$ and $\tilde{A}\cup A=X$) [39]. For instance, if $A(x)=0.6$ then $(A\cap\tilde{A})(x)=\min\{A(x), \tilde{A}(x)\}= \min\{0.6, 1-A(x)\}= \min\{0.6, 0.4\}=0.4$. More interestingly, if $A(x)=0.5$ then $A=\tilde{A}= \tilde{A}\cap A= \tilde{A}\cup A$.

It should also be noted that FST is not reduced by probability theory. An example will now be discussed to show the differences; for thorough discussions refer to [48,62,63]. Let P={all philosophers} and $\mathcal{P}$={all empirical philosophers} and let p_1, $p_2\in P$ where $\mathcal{P}(p_1)=0.9$ and $\Pr(p_2\in\mathcal{P})=0.9$. In the latter case, all that can be said about p_2 is that there is a 1 in 10 chance that the philosopher is not an empiricist. In the former case, the philosopher is quite similar to the ideal empiricist (for instance, Hume). In other words, with the latter, the information that is conveyed concerns relative frequency, whereas the former deals with the representation of similarity to imprecisely defined properties. Another fundamental difference involves observational effects on information content. In this example, say it is now observed that p_1=Aristotle and p_2=Plato. The information content in the former case does not change, $\mathcal{P}$(Aristotle)=0.9 but in the latter case the probability now drops from 0.9 to 0.0.

2.4 Robust Statistics

A statistic is considered *robust* if it is resistant to effects caused by extreme values [51]. More specifically, a statistical estimate is robust if it is insensitive to slight deviations from its requisite model assumptions (often normal assumptions) about the underlying distribution [98]. Discussions about robust statistics often go hand in hand with the notion of *outliers*, observations that do not follow the pattern of the majority of the data [4]. For instance, say one is fitting a line through a set of points by minimizing a standard sum-of-squares error

$$E(y) = \sum (y - x_i)^2 . \qquad (6)$$

A potential difficulty with (6) is that it receives its largest contributions from points that have the largest errors and it is outliers that will have the largest errors [27]. In Figure 7(a), the line appears to be a good fit of the systematic aspects of the points. However, in Figure 7(b), a single outlier has dominated the line fitting process, since it produced the largest error, and has, as a result, skewed the line away from the other points.

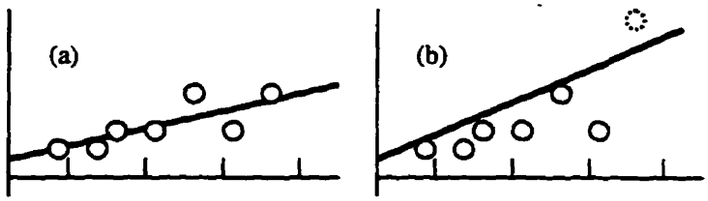

Figure 7: Fitting a line through a set of points. In (a), without outliers. In (b), with an outlier

One solution to this problem is to use the robust error

$$E(y) = \sum |y - x_i| . \qquad (7)$$

Outliers will have smaller errors using (7) rather than (6) and hence their contributions are diminished. Moreover, minimizing (7) with respect to y gives

$$\sum sign(y - x_i) = 0 \qquad (8)$$

which is satisfied when y is the median of all the x_i [15]. If one of the points is an outlier this has no additional effect on the solution.

Another instructive example comparing a standard and robust measure of dispersion is given in [114]. The mean square deviation is

$$s_N = \sqrt{\frac{1}{N}\sum (x_i - \bar{x})^2} \qquad (9)$$

and the mean absolute deviation is

$$d_N = \frac{1}{N}\sum |x_i - \bar{x}|. \qquad (10)$$

Under certain regularity conditions, we can define the *relative efficiency* of d_N to s_N as

$$\lim_{N\to\infty} \frac{\mathrm{var}(s_N)}{\mathrm{var}(d_N)}. \qquad (11)$$

where var(s_N) is the variance of s_N. For instance if the relative efficiency of d_N to s_N were 0.5 then d_N would require twice the sample size needed for s_N in order that both measures have the same statistical power. Tukey took two groups of normally distributed observations having the same mean. The second group, however, had three times the standard deviation of the first (in other words, the errors of some of the observations in the second group are increased by a factor of three). Each observation belongs to the first group with a probability of 1-ε and to the second group with a probability of ε where $0 \le \varepsilon \le 1$. Table 1 lists the relative efficiency of d_N to s_N with increasing contamination of the first group of observations by the second. For exactly normal observations, s_N is 12% more efficient than d_N. However, with as little as 0.2% contamination, the robust measure is slightly more efficient. It becomes more than twice as efficient when there is 5% contamination.

ε	0	.001	.002	.005	.01	.02	.05	.10	.15	.25	.50	1
RE	.88	.95	1.02	1.20	1.44	1.75	2.04	1.90	1.69	1.37	1.02	0.88

Table 2: Relative efficiency (RE) of d_N with increasing contamination (ε)

Looked at another way, this example demonstrates that lengthening the tails of a distribution can greatly increase the variability of s_N. Since d_N is less sensitive to such a change it is

distributionally robust. Moreover, because it is in the long tails where outliers reside, it is concomitantly *outlier resistant*.

In [52], Huber suggests that, with typical "good data" samples in the physical sciences, $0.01 \leq \varepsilon \leq 0.1$, and if this holds then robust statistical measures are invaluable. One may argue that these examples do not corroborate the need for robust statistical procedures but only suggest that outlying observations must be detected and dealt with in some fashion. But outlier detection is a contentious problem. Causes of outliers fall into two broad and somewhat overlapping categories, model weakness and natural variability. Model weakness includes response variables in the wrong scale and isolated measurement and recording errors. Identification of an outlier may lead to its subsequent rejection, important new information contained in concomitant variables that would otherwise have gone unnoticed, its incorporation through model revision, or a recognition of an inherent weakness in the data and thus to additional experimentation [8]. Multiple outlier detection, especially in a high-dimensional space, suffers from two major problems. Two or three outlying observations that are roughly equidistant from their sample mean can drastically inflate the mean as well as the variance, to such a degree that the outliers are not detected. This problem is known as *masking*. Further, as the sample size increases, the masking effect between any two outliers decreases, but unfortunately, the number of outliers increases so the overall masking effect does not change. Nevertheless, some success has been achieved in unmasking multivariate outliers using robust estates of location and covariance [99]. *Swamping* is the converse to masking and occurs in an inappropriate block test for multiple outliers when a highly discordant outlier carries with it another observation that is not an outlier. In [24], it is argued that it is better to defer to domain-dependent technical expertise than any statistical criterion for straight outlier rejection. Finally, with respect to methods for detecting multivariate outliers, Gnanadesikan [38] states that the "*... complexity of the multivariate case suggests that it would be fruitless to search for a truly omnibus outlier detection procedure. A more reasonable approach seems to be to*

tailor detection procedures to protect against specific types of situations, e.g., correlation distortion, thus building up an arsenal of techniques with different sensitivities. This approach recognizes that an outlier for one purpose may not necessarily be one for another purpose!"

3 The Process of Classification

Classification, a multi-faceted exercise, is realized through a classification system. In the most general sense, the system creates a discrimination function mapping individuals to a set of class indices. The performance or accuracy of the system must also be validated.

This chapter describes the general architecture for a classification system including the optional preprocessing and postprocessing blocks and the classification block proper. Issues revolving around normally distributed data, *a priori* knowledge, and verification strategies are discussed. With regards to verification, the data sets that are used throughout this thesis are presented. Also, a chance-corrected measure of agreement is discussed. This agreement measure is used throughout the thesis to measure classification performance. The chapter concludes with a review of the field.

3.1 Classification Systems

In the most general sense, the problem of classification is a problem of function approximation: attempt to estimate an unknown function

$$f : \Re^n \rightarrow [0,1]^k \qquad (12)$$

from observed pair-wise random samples, (x_1, y_1), (x_2, y_2), ..., (x_N, y_N), where $x_i \in \Re^n$ is an n-dimensional input vector and $y_i \in \Re^k$ is an k-dimensional output vector. Normally, an association is established between the output vectors and a set of groups (or classes), ω_1, ω_2, ..., ω_k such that

$$f : \Re^n \rightarrow \{1,2,\ldots,k\}. \qquad (13)$$

Each ω_i comprises a subset (usually non-empty) of N_i input vectors. A common association, 1-of-k classification encoding

$$f : \Re^n \rightarrow \{0,1\}^k \qquad (14)$$

assumes that each input vector, x, belongs to one and only one group, say ω_j, and sets all coordinates of the respective output vector, y, to

$$y_i = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases} \quad (15)$$

A classification system attempts to approximate f using the input/output pairs such that, for any given input vector, its corresponding output vector is generated within some error tolerance. After the approximating function is computed, its effectiveness should be tested by mapping new input vectors to output vectors. An input vector, x, is considered to be correctly classified by the system, if the generated output vector is the same as the desired output vector within the error tolerance. Figure 8 is a diagram of a generalized classification architecture.

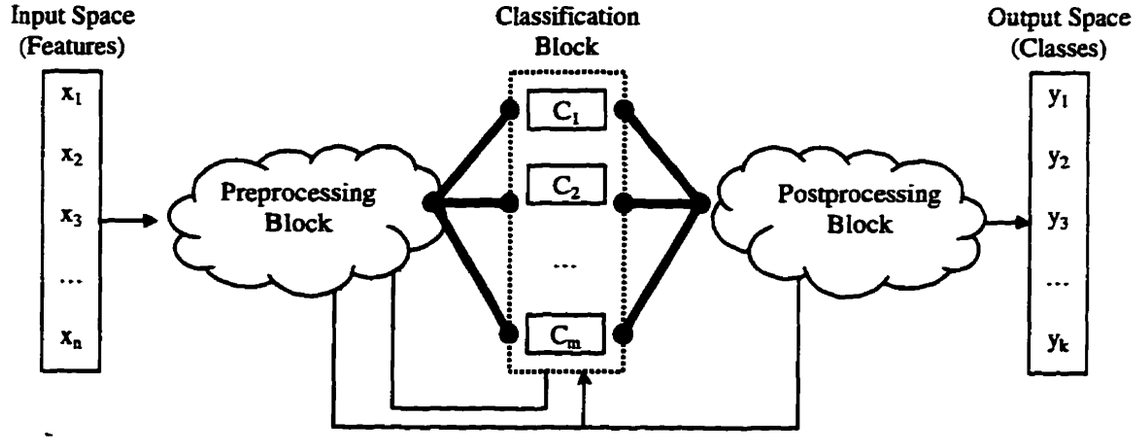

Figure 8: A generalized classification architecture

3.1.1 Classification block

The classification block is the core of the classification system. This block predicts the output vector or group assignment of an input vector or a modified input vector if a preprocessing block is present. Usually, the classification block contains a single classifier but it is not uncommon to have a set of $m>1$ classifiers operating in concert. This set may be comprised of any combination of classifiers including linear discriminant analysis and ANNs. In general, classifiers attempt to minimize some objective function, usually an Euclidean metric, comparing the desired output

with the actual output, in order to correctly predict output vectors given input vectors. The objective function may be minimized iteratively as with ANNs or non-iteratively as with linear discriminant analysis. Although the typical topology is full interconnectivity between all elements from the preprocessing (or input) block to each classifier, it is not a necessary requirement. Each classifier may be connected to only certain, possibly disjoint, regions of the feature space. Moreover, the preprocessing block may also pass the original data to the classification block in addition to any preprocessed features.

FST may also be used as an internal constituent of a classifier system. This may involve the fuzzification of some component or mechanism of the classifier system. In the case of a multi-layer perceptron ANN (see section 4.1) that uses the back-propagation algorithm, for example, one could have a set of fuzzy inference rules that would make dynamic adjustments to the learning parameter, α, based on the change of classification error. These inference rules would be of the form, "If the change in the error is X then the learning rate is Y" where X and Y would be fuzzy sets [44,58]. Another example would be the fuzzification of the probabilistic neural network (see section 4.3). The set of Gaussian receptive fields that are normally used in this network could, in theory, be replaced by fuzzy sets. The role of FST within a classifier system may be much more pervasive, that is, the architecture may have been completely developed using FST. Examples of this approach include referential fuzzy neural networks and neuro-fuzzy networks.

3.1.2 Preprocessing block

It may be difficult to cull discriminatory information from data (for instance, diagnostic information from magnetic resonance spectra) due to their complex nature, the confounding effects of noise, and/or the presence of artifacts produced by the data acquisition process itself. One of the most pernicious effects is the "curse of dimensionality", a phenomenon associated with a paucity of high-dimensional input vectors (that is, n/N is large). In many applications, for

instance, acquisition of biomedical magnetic resonance spectra of tissue in different disease states, this problem is all too common. The optional preprocessing block attempts to deal with issues such as this by simplifying, in some fashion, the original input data prior to presentation to the classification block.

Preprocessing methods fall into several categories and may be used singly or in concert. The first category is the set of techniques that diminish the effects of noise. For instance, input vectors are often smoothed (averaged) with the assumption that the signal–noise ratio is not unreasonably low and that the noise signature, not being predominant, will be "washed away" by the smoothing. Another method used is to add uniform or Gaussian noise to the original noisy input vectors. The rationale is that the additional noise will, on average, cancel out the existing noise signature.

The second category comprises those methods that reduce the dimensionality of the problem. One common method is to average over a fixed number of contiguous coordinates of the input vectors. The average for each "window" may be the mean or median of the coordinates. A more sophisticated average method such as α-*trimmed* or α-*Winsorized* means [32] may also be used. The α-trimmed mean, μ_α, of N observations drops the smallest and largest observations from the sample

$$\mu_\alpha = \frac{1}{N-2j}\sum_{i=j+1}^{N-j} X_{(i)} \qquad (16)$$

where j is the smallest integer greater than or equal to αN and $X_{(i)}$ are the ordered observations. Instead of dropping the extreme observations, in the Winsorized mean, μ_w, they are replaced by the remaining respective largest and smallest observations

$$\mu_w = \frac{1}{N}\left((j+1)(X_{(j+1)} + X_{(N-j)}) + \sum_{i=j+2}^{N-j-1} X_{(i)} \right). \qquad (17)$$

Another common method is principal component analysis (discussed in section 5.2) that performs a linear transformation of the original data such that the coordinates of the transformation, known as principal components, account for decreasing amounts of variance [41]. Another dimensionality reduction method is domain-specific and involves the use of *a priori* knowledge of the input vectors. For instance, it may be known that signals acquired from MR spectrometers of thyroid tissue may have regions that are not particularly relevant to the specific diagnostic issue or that specific regions are highly significant. Appropriate selection or rejection of such regions will achieve dimensionality reduction.

The third category of preprocessing strategies involve the adjustment of the input space dimensionality for reasons other than its reduction. In fact, the adjustments may increase the input space's dimensionality. For instance, one may take all quadratic combinations of the coordinates for submission to the classification block. This is especially effective if the classifier is linear but the original problem is not linearly separable, that is, n-dimensional hyperplanes could not separate the input vector into their respective groups. It may then be the case that the linear classifier will succeed in discriminating between the groups because the problem is still linear in the new input space; only the parameter space is quadratic. For example, say there is a 2-dimensional input space and two groups. If the point is inside a cluster of points bounded by a circle it is in one group, otherwise it is in the other. A linear classifier would perform poorly because a single line cannot be computed to separate the two groups. However, if the input space is adjusted by adding two new coordinates, namely, the square of each of the original coordinates, a linear classifier would be able to successfully discriminate between the two groups.

FST may also be used for this type of preprocessing, for example, data may be fuzzy encoded prior to presentation to the classification block [89]. Fuzzy encoding is the process of determining the respective degrees to which a datum belongs to a collection of fuzzy sets and subsequently

using these membership grades in place of the original datum. This procedure is akin to 1-of-*n* intervalization encoding except that gradual transitions occur at the boundaries [18].

The next category of preprocessing is the set of normalization and scaling methods and their relatives [91]. The intent of these methods is variance stabilization [76]. It may be the case that some features have far greater variance than do others and hence the former may play a more significant role during classification process simply by nature of this greater variance than the latter. This is problematic if the latter features are, at the same time, highly discriminatory. If, however, collinear vectors need to retain their distinctiveness, for instance, pixel values of the same image at different illumination levels, normalization methods cannot be used.

Burnishing tarnished gold standards is another preprocessing category that has only recently been investigated [88]. A reference test, or gold standard, that is used as a benchmark, against which the classification system is measured, may itself be imprecise or even unreliable. Contributing factors include subjective estimates by a domain expert (or panel of experts) or simple clerical errors. Of course, while this preprocessing category addresses the possible imprecision of the gold standard, at the same time, the vital discriminatory power of a well-established reference test must also be retained. One possibility is to use FST or robust deviation measures [109] to enhance the gold standard by incorporating non-subjective within-group centroid information.

The final category of preprocessing is artifact suppression. Discriminatory information within input vectors might be systematically distorted by the very process used to acquire them. Unlike noise, which is introduced due to limitations of physical devices used in data acquisition, an artifact is a phenomenon that is an inherent part of the signal. For example, an infrared spectrum of *ex vivo* thyroid tissue will have little noise but an enormous water signature that completely dominates any interesting metabolites. The baseline signal is usually adjusted to suppress the water signature. Techniques to suppress artifacts are highly domain-specific and are often *ad hoc*.

3.1.3 Postprocessing block

The optional postprocessing block may perform several functions. It may perform some inverse transformation on the outputs generated by the classification block in order to reverse the effects of a preprocessing technique (for instance, scaling) prior to generating a final output vector. If there is more than one classifier then this block combines the outputs from each classifier and produces a final output vector. Combination strategies include, "winner-take-all", weighted competition, consensus, and fuzzy integration.

3.2 Classification Issues

3.2.1 "DeGaussing normality": the law of errors

Many preprocessing techniques as well as certain classifiers assume that data are sampled with a normal distribution. Unfortunately, there is, in general, no guarantee that this is the case. In fact, in many biomedical classification problems, data are sampled with non-normal distributions. Data that correspond to some rare disease state, for example, may be under-represented with respect to data corresponding to a non-disease state. Moreover, the presence of errors in the data, whether manifested as signal noise in spectral information, inaccurate classifications, or imprecise attribute values, may or may not be normal. Part of the problem in dealing with the above issues resides in the fact that Gauss' "law of errors" is often applied inappropriately. The law states that if repeated measurements are made on the same object, the distribution of the random component on the errors can be well approximated by the Gaussian distribution. For instance, this implies that repeated measurements based on the acquisition of an infrared signal from a particular diseased tissue sample would follow the Gaussian distribution. However, this does not necessarily imply that measurements of the infrared signals from all diseased tissue samples would follow the Gaussian distribution.

As a historical remark, Rietz, in his famous 1927 monograph on mathematical statistics [94], contends that one of the factors behind the relative lack of progress in this subject for fifty years

after Laplace's Théorie Analytique des Probabilités published in 1812, was that "*the followers of Gauss retarded progress in the generalization of frequency theory by overpromoting the idea that deviations from the normal law of frequency are due to lack of data*". Cramér also levels this charge in his 1946 book [23]; "*Under the influence of the great works of Gauss and Laplace, it was for a long time more or less regarded as an axiom that statistical distributions of practically all kinds would approach the normal distribution as an ideal limiting form, if only we could dispose of a sufficiently large number of sufficiently accurate observations. The deviation of any random variable from its mean was regarded as an 'error', subject to the 'law of errors' expressed by the normal distribution.*"

The Bayes classifier is often highly touted as the only classification technique to be used because it is the theoretically best classifier. Assuming that the distributions of the random vectors are known then it is the case that the Bayes classifier does indeed give the smallest error that can be achieved from the given distributions [118]. Unfortunately, these distributions often are not known. Furthermore, even if they were known, there is the pragmatic consideration of its implementation. Although the Bayes classifier, under the previous assumption, is optimal, its implementation is often difficult in practice because the probability density function is not accessible, particularly when the dimensionality is high [35].

Now, this should not be construed as some wholesale condemnation of "normality" for "*it is undeniable that, in a large number of important applications, we meet distributions which are at least approximately normal*". Nevertheless, it is prudent not to fall into the trap, described by Lippman, that "*everybody believes in the law of errors, the experimenters because they think it is a mathematical theorem, the mathematicians because they think it is an experimental fact*". "*Mathematical proof tells us that, **under certain qualifying conditions**, we are justified in expecting a normal distribution, while statistical experience shows that, in fact, distributions are often **approximately normal***".

The problem is that, in many "real world" classification scenarios, the qualifying conditions *do not hold*: one class of data may be sampled differently from that of another class; due to the nature of an experiment, all data may be sampled with a skewed distribution; or the curse of dimensionality issue. Practical experience suggests that there are times when distributions are simply *not normal*, not even approximately so. The thrust of this argument is that it is prudent to examine the efficacy of non-parametric methods when dealing with classification problems. This includes judicious use of FST and robust statistics at the preprocessing level of the classification architecture that do not require the satisfaction of normality preconditions.

3.2.2 A priori knowledge

It is often sufficient to classify data using strictly objective general mathematical discrimination techniques that do not take into account the nature of the domain space under investigation. Not only are these objective methods often sufficient but they are often preferred because they eliminate "subjective" bias. All objective methods use some quantitative measure to determine the similarity of one data point to another. These measures include Euclidean distance, Mahalanobis distance, the L_1 norm, correlation/covariance measures, and relative entropy [65]. Nevertheless, there are occasions when it may be extremely worthwhile to exploit the nature of the domain space. A fine line exists between subjective "bias" and subjective information and it may, at times, be crucial to cross that line and exploit any *a priori* knowledge that may be acquired from the specific problem domain [79,93]. In the case of biochemical spectral data, for instance, *a priori* knowledge may allow the investigator to focus his attention to a small set of spectral regions corresponding to the presence or concentration of some metabolites that are known to be significant in the diagnosis or monitoring of a disease state. Dimensionality reduction, therefore, occurs and the subsequent classification process is simplified. Artifact suppression is also extremely domain-dependent. Dealing with tarnished gold standards may also be domain-dependent although general preprocessing techniques can be employed. The

exploitation of *a priori* knowledge is not without its pitfalls, however. The investigator must have a thorough understanding of the underlying domain and this often involves consultation with one or more domain experts and all the requisite perils this knowledge acquisition process entails. Further, the subjective information may obfuscate, otherwise obvious, objective relationships in the data that similarity measures would have uncovered. The former problem has often manifested itself in expert system technology [3]. Knowledge acquisition is of paramount importance in this methodology and poor acquisition leads to poor system performance especially with respect to "brittleness" — steep rapid, and sometimes complete, degradation of performance when the expert system is pushed to the periphery of the particular problem domain [77]. As in the knowledge acquisition versus rule codification dichotomy, the 80/20 rule applies in classification systems; only 20% of the investigator's time should be spent on the mechanics of the selected classifier system, 80% of the time should be spent on a thorough analysis, including preprocessing, of the data and an understanding of the problem domain. The latter problem can be resolved by using subjective information only after a strict objective analysis has been performed. In conclusion, *a priori* knowledge of a particular problem domain may be invaluable in the classification process but caution must be exercised in its exploitation.

3.2.3 Verification

How is the performance of a classification system to be measured? One verification method is to divide the data into a design set and a test set (a 2:1 ratio is often used). The classification system uses the design set to set all the necessary parameters particular to it. The system is then presented with vectors from the test set and the corresponding output vectors are computed. An $n \times n$ contingency table of desired versus actual classification outcomes is constructed. A typical measure of performance is

$$P_o = \frac{\sum_i n_{ii}}{N} (i = 1, \ldots, k) \qquad (18)$$

where N is the total number of vectors in the test set and k is the number of groups.

There are a number of concerns with this method. First, biases are introduced when the data are artificially divided into design and test sets. That is, the measure of agreement, P_o, will change depending on the selection of design and test sets. A simple solution is to build several randomly sampled design and tests set pairs and compute the average P_o of all pairs. Another method is to use the leave-one-out cross-validation strategy: build N (N is now the total number of vectors) design/test set pairs where each test set comprises a single vector and each design set comprises the remaining N–1 vectors.

Another issue is poorly distributed groups in the sample, that is, at least one of the groups has a small number of vectors with respect to the remaining groups. It is important, especially with non-linear iterative classifiers, that the design set has roughly the same number of vectors from each group, otherwise the under-represented group will contribute less significantly to the design process and, therefore, there will be a concomitant loss of agreement between the desired and actual outcomes for the test vectors within the under-represented group. If the smallest group is still large, in absolute terms, one can simply use a percentage of the number of vectors in that group as a design set floor for all other groups. Unfortunately, if the smallest group is small, in absolute terms, this will not work. One could artificially increase the number of samples in the smallest group by adding copies of randomly selected vectors into the group. Additionally, noise can be introduced to the copies.

One final problem is that the measure of agreement, P_o, does not take into account the agreement that might be due to chance [30]

$$P_c = \frac{\sum_i \left(\sum_j n_{ij} \sum_j n_{ji} \right)}{N^2} (i, j = 1, \ldots, k) \qquad (19)$$

For example, Table 3 and Table 4 are two contingency tables for a 300-vector classification problem where 10%, 80%, and 10% of the vectors are in groups 1, 2, and 3, respectively. At first

glance, since P_o=0.66 for both tables, the results seem to indicate equivalent classification performance. In Table 3, however, the vectors were randomly assigned in accordance with their marginal rates, that is, 10% of the vectors were assigned to each of groups 1 and 3 and the remaining 80% were assigned to group 2. This is evident if we compute P_c for both tables. For Table 4 P_c=0.52 but P_c=0.66 for Table 3 clearly indicating, in this case, that all of the measured agreement is due simply to chance.

	Group 1	Group 2	Group 3	
Group 1	3	24	3	N_1= 30
Group 2	24	192	24	N_2=240
Group 3	3	24	3	N_3= 30
P_o		0.66		N= 300

Table 3: 300-vector 3-group contingency table (P_c=0.66)

	Group 1	Group 2	Group 3	
Group 1	15	10	5	N_1= 30
Group 2	37	163	40	N_2=240
Group 3	2	8	20	N_3= 30
P_o		0.66		N= 300

Table 4: 300-vector 3-group contingency table (P_c=0.52)

This example strongly suggests that chance must be accounted for by an agreement measure. One option is to use the κ coefficient [21,33] as a chance-corrected measure of agreement between the desired and actual group assignments

$$\kappa = \frac{P_o - P_c}{1 - P_c} \qquad (20)$$

If the agreement is due strictly to chance then κ=0. If the agreement is greater than chance then κ>0; κ=1 indicates complete agreement. If the agreement is less than chance then κ<0 with the minimum value dependent upon the marginal distributions. Figure 9 plots decaying κ values for P_o=0.66 as P_c increases. Returning to the example, κ=0.00 for Table 3 indicating that the observed agreement is due strictly to chance but κ=0.29 for Table 4.

An arbitrary but useful benchmark for the strength of the agreement is discussed in [68], namely, the agreement strength is poor if κ=0, slight if 0.00<κ≤0.20, fair if 0.20<κ≤0.40,

moderate if $0.40<\kappa\leq0.60$, substantial if $0.60<\kappa\leq0.80$, and almost perfect if $0.80<\kappa\leq1.00$. Under a number of assumptions the asymptotic large sample variance of κ may also be computed [30,34].

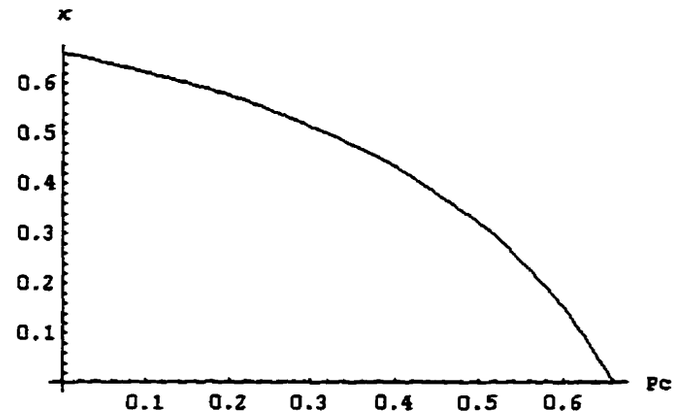

Figure 9: κ as a function of P_c (P_o=0.66)

3.3 *Data Sets*

The data sets that will be used throughout the thesis will now be presented. These data sets are widely divergent in character. Artificial data will be used for a thorough pedagogical examination of the techniques to be described since we have "control" over them. Three "real world" biomedical spectral data sets will also be used: spectra acquired from an infrared spectrometer using diseased and control brain tissue; spectra acquired from a magnetic resonance spectrometer using normal and cancerous thyroid tissue; and magnetic resonance spectra of brain neoplasms. These data are classified according to their respective gold standards; the edict of the medical pathologist after performing a morphological analysis of the tissue samples.

3.3.1 *Bounding problem in n*-dimensions

Consider a data set consisting of points, $x=[x_1, x_2]$, such that $x\in\omega_2$ if $-0.75<x_1<0.75$ and $-0.75<x_2<0.75$, otherwise, $x\in\omega_1$. Figure 10 shows four ($2n$) lines, H1 through H4, that perfectly separate the two classes.

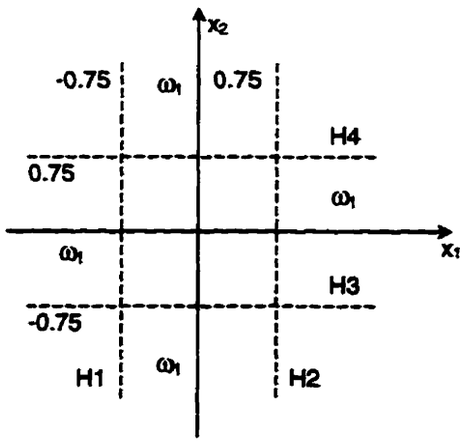

Figure 10: The bounding problem in two dimensions

This bounding problem can easily be extended to the n-dimensional case. Artificial data were generated that fall into two classes: those points that are bounded by a set of hyperplanes and those that are outside the region. A point $x=[x_1, x_2, \ldots, x_n] \in \omega_2$ if $-0.75<x_i<0.75$ ($\forall i=1, 2, \ldots, n$), otherwise $x \in \omega_1$ (note that $2n$ hyperplanes will perfectly separate the two classes). Equal numbers of points were selected from each class: ω_1 points were randomly selected from a uniform distribution in the range $(-0.75,0.75)^n$; ω_2 points were randomly selected from a uniform distribution in the disjoint ranges $[-1.0,-0.75]^n$ and $[0.75,1.0]^n$.

3.3.2 Disk and torus

In this artificial data set, a unit circular disk centred at the origin is surrounded by a 2-dimensional torus of equal area (Figure 11). The data set falls into two classes: those points that lie within the disk, ω_1, and those points that lie within the torus, ω_2. The design and test sets each contain N points: $N/2$ within the disk and $N/2$ within the torus. Points from class ω_1 were randomly selected from a uniform distribution in the range $[-1.0,1.0]^2$ such that they were in the unit circular disk centred at the origin (with area π). Points from class ω_2 were randomly selected from a uniform distribution in the disjoint ranges $[-\sqrt{2},-1.0]^2$ and $[1.0,\sqrt{2}]^2$ such that they were in the torus centred at the origin (also with area π). In order to determine the efficacy of the classifiers when data are not equally distributed, the data in the design set were randomly selected

such that only 10% of the data within the disk had their first coordinate less than zero and only 10% of the data within the torus had their second coordinate greater than zero.

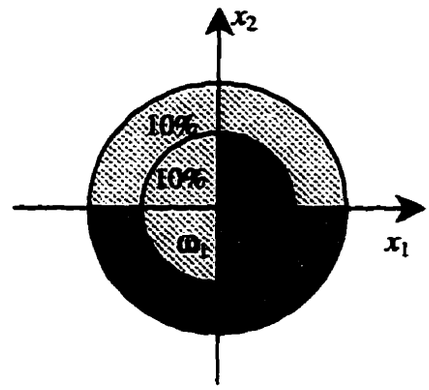

Figure 11: Distribution of design data

3.3.3 One-dimensional points with various distributions

For each of a collection of data sets, 200 one-dimensional points were randomly selected using two different distributions. All points from the first distribution were assigned ω_1 (N_1=100) with the remainder assigned to ω_2 (N_2=100). The design set was comprised of 50 ω_1 points and 50 ω_2 points (N_d=100) with the remaining points assigned to the test set (N_t=100).

For the first data set (Figure 12), the ω_1 points were sampled from the normal distribution $N(0,1)$ (mean of 0 with standard deviation of 1) and the ω_2 points were sampled from $N(3,1)$. For the second data set (Figure 13), the ω_1 points were sampled from the normal distribution $N(0,1)$ and the ω_2 points were sampled from $N(3,2)$. For the third data set (Figure 14), half of the ω_1 points were sampled from the $N(0,1)$ and the other half were sampled from $N(5,1)$ while all of the ω_2 points were sampled from $N(10,1)$. For the fourth and final one-dimensional data set (Figure 15), all ω_1 points were sampled from the $N(10,2)$, while all ω_2 points were sampled from a log normal distribution with mean of 2 and a standard deviation of 1. Hence, ω_2 is a highly skewed group with a probability density function (pdf) that significantly overlaps the pdf of ω_1.

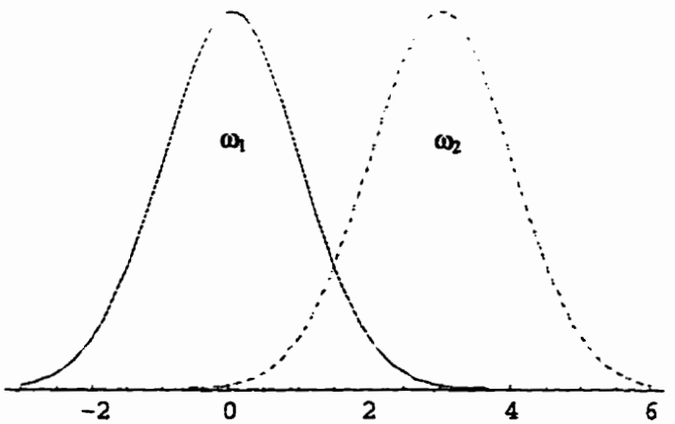

Figure 12: Normal distributions with equal variances

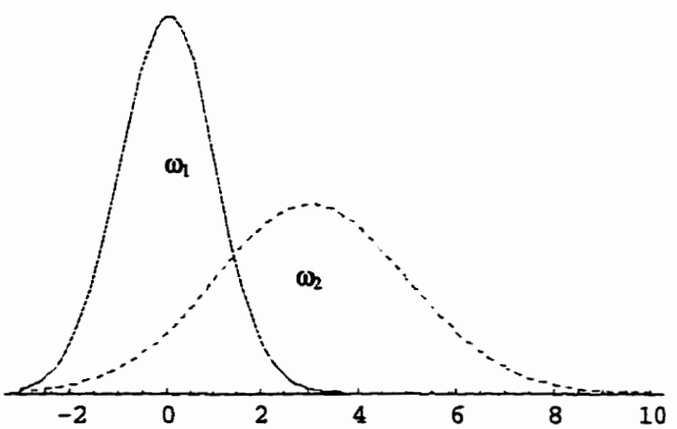

Figure 13: Normal distributions with unequal variances

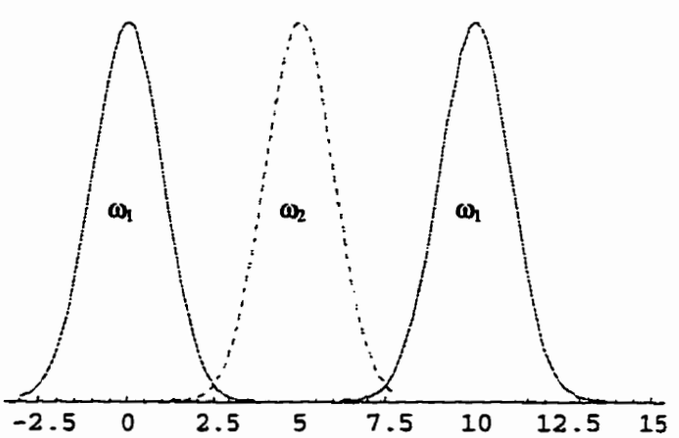

Figure 14: Normal and bimodal distributions

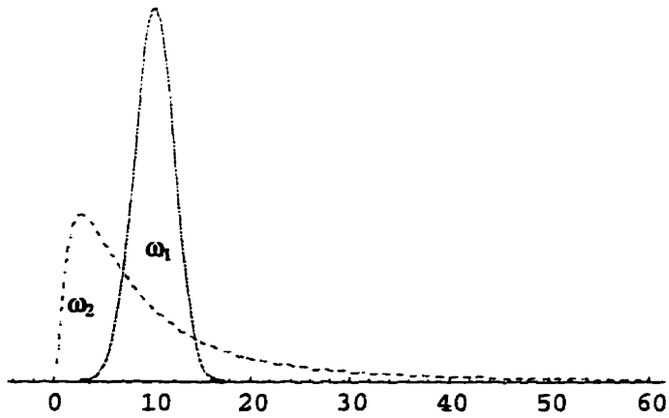

Figure 15: Normal and log normal distributions

3.3.4 Magnetic resonance spectral data

Magnetic resonance (MR) spectroscopy is quickly emerging as an effective noninvasive diagnostic tool. MR spectra reflect altered cellular chemistry before gross morphological changes are manifest.

One-dimensional magnetic resonance spectra were obtained at 360 MHz (37°C) for 25 thyroid biopsies. Of these, 16 had papillary carcinomas and 9 were classified as normal. Two phased spectral regions were analyzed: the main lipid CH_2 and CH_3 peaks, 0.64–2.59 ppm; and the choline-like species, 2.59–3.41 ppm. Analysis was based on 170 input points for the choline region and 400 input points for the lipid region. This data is particularly difficult to classify: it is a small data set, the classification assignments are not particularly crisp, and the spectra are quite noisy.

The other data set comprises 206 ^1H MR spectra (360 MHz, 37°) consisting of 95 meningiomas (M), 74 astrocytomas (A), and 37 control samples of non-tumourous brain tissue from patients with epilepsy (E). The 550 data points in the region of 0.3–4.0 ppm were used in the analyses. The phased spectra were randomly assigned to either a design set (n=80) or a test set (n=126). The design set contained 29 M's, 31 A's, and 20 E's. Figure 16 shows typical spectra. A subsequent normalized data set was also created where each datum was divided by the area of its spectrum.

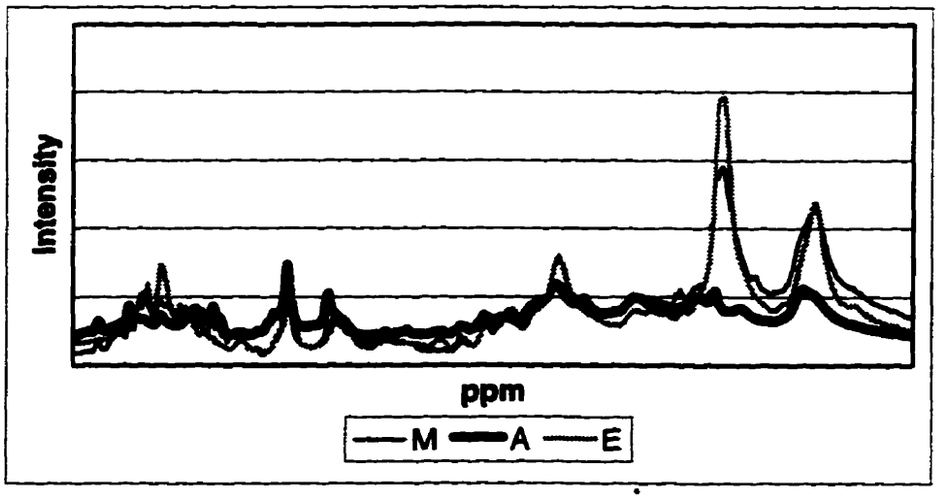

Figure 16: Typical magnetic resonance spectra

3.3.5 Infrared spectral data

Alzheimer's disease (AD) is a progressive brain disease usually occurring in persons over fifty years of age. It is the most common dementia of adult life and is marked by a general atrophy of the brain. Chief symptoms include memory loss, disorientation, and impaired judgment and speech [57]. The pathological hallmarks of AD are the abnormal neuritic plaque deposits and neurofibrillary tangles in the cortical regions of the brain. Neuritic plaque deposits are roughly spherical particles that accumulate extracellularly in the AD brain. Neurofibrillary tangles are bundles of abnormal filaments found intracellularly within neurons that appear up to ten years after neuritic plaque deposition. Alzheimer's disease (like all other disease states) is accompanied by biochemical changes in tissues and cells. Infrared (IR) spectroscopy has been extensively used in the past for characterizing simple organic molecules, and more recently for the structural analysis of biological compounds including lipids, proteins and deoxyribonucleic acid (DNA) [54]. IR spectroscopy, by probing molecular vibrations, is a technique sensitive enough to detect these changes. An IR spectroscopic study [20] indicates considerable variability in spectra of AD grey matter which make it difficult to classify AD tissue based upon a subjective spectroscopic evaluation alone thus making this set a good candidate for investigation.

Grey and white matter were sampled from various regions of histopathologically confirmed non-Alzheimer's (control) and Alzheimer's diseased age matched brains. For each sample, 200 interferograms were accumulated and Fourier transformed to generate spectra with a nominal resolution of 4 cm^{-1} in the regions between 1000–4000 cm^{-1}. Each spectrum was discretized to 416 data points. The initial data set was composed of 114 spectra and divided into 49 control spectra (C) and 65 AD spectra (A). Subsequently, additional spectra were collected and separated into five classes. The original spectra were also subdivided into these five classes. Of the 163 spectra in this subsequent data set, 49 spectra were from control grey matter tissue (CG), 23 spectra were from control white matter tissue (CW), 58 spectra were from Alzheimer's diseased grey matter tissue (AG) and 24 spectra were from Alzheimer's diseased white matter tissue (AW). Finally, nine spectra of tissue from a brain with a condition known as $18q^{-}$, characterized by the presence of neurofibrillary tangles (NT) without any neuritic plaques, were included in the data set. Figure 17 shows some typical IR spectra in this data set. Note that the control and Alzheimer's spectra are significantly different. Distinguishing between gray and white matter within each is much more difficult, however.

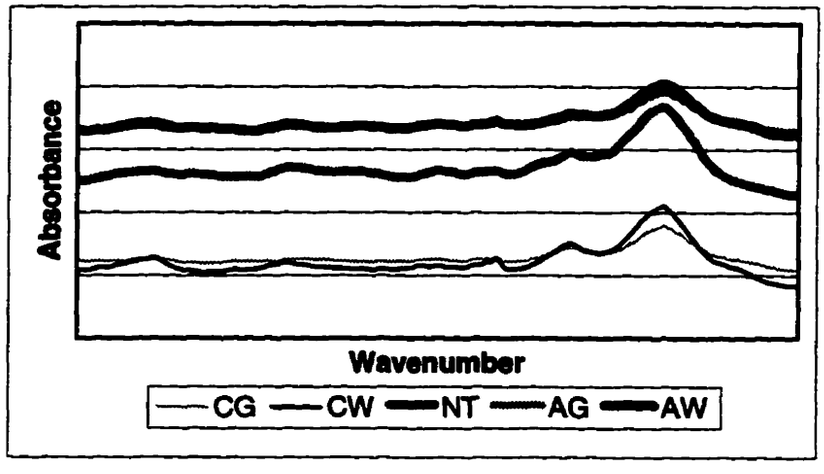

Figure 17: Typical infrared spectra

3.4 Field Review

A set of classifiers whose combination is based on the Dempster-Shafer theory of evidence, which uses statistical information about the relative classification strengths of several classifiers is characterized in [97] and it is reported that misclassifications are reduced by 15–30%. In [1], the fusion of several types of robotic scene data using a fuzziness measure enhanced the recognition capability of an autonomous system. In [53], a bias constraint based on prior knowledge about the underlying distribution of the data is discussed as a means for reducing the overall error measure of a classifier.

Modular neural networks involve adaptive mixtures of local experts [55]. It consists of a group of ANNs competing to learn different aspects of a problem. A gating network controls the competition and learns to assign different regions of the data space to different local expert networks.

Stacked generalization [120] is a scheme for minimizing the generalization error rate of one or more generalizers. Stacked generalization works by deducing the biases of the generalizer(s) with respect to a provided learning set.

A hybrid architecture is described in [73] for classification expert systems that combines semantic networks and ANNs for representing knowledge. A semantic network is used to describe the objects of the problem domain and their relations at the intensional and extensional levels. This hybrid scheme allows the construction of fuzzy expert systems able to inherit useful properties from the sub-symbolic neural networks and symbolic expert systems, such as: expert knowledge representation, integration of multiple expert knowledge sources, heuristic and incremental learning, feature selection, and treatment of vague input data.

Flexible data structures and retrieval specifications within a database are achieved using standard relational formalisms but their implementations are in terms of crisp, static, and deterministic relations whereas real-world applications data are often imprecise, inherently

dynamic and non-deterministic. In [36], it is shown how FST can be incorporated into relational database systems to allow for a wider range of real-world requirements and closer human-machine interaction.

An approach based on fuzzy classification of epileptiform spikes in electroencephologram recordings to minimize the number of false positive classifications returned by the monitoring system is described in [59]. A system is illustrated in [13] that uses fuzzy sets as the representation framework for the classification of multisource remote sensing data. An interesting comment made in this paper is that one of the main limitations to properly classifying remote sensing data is the acquisition of domain knowledge from the experts. In [92], a fuzzy classification method for FFT spectra is described to distinguish abnormal vibrational conditions of rotating machinery. Nonlinear fuzzy operators, optimally tuned using genetic algorithms (optimal tuning of fuzzy sets is a vital area of investigation [5]), have been successfully used as an image processing method [101]. A plethora of research has been undertaken in the area of enhancing classification, signal processing, and sensors using FST [6,26,66,101,102]. Analysis and classification of esophageal motility records were investigated in [2] using signal processing and fuzzy-set pattern recognition techniques. The FST extensions reduced the classification error rate by half. Similar results were achieved in [115] where blood cell anomalies were discriminated.

Computerized alarm systems have been well accepted in clinical medicine but suffer from not being able to handle patient/disease specificity, temporal changes, dynamic patterns, and multivariable combinations. The approach in [46] uses techniques from fuzzy set theory and artificial intelligence in order to initialize the alarm system and interpret the incoming data.

One of the earliest fuzzy neural network hybrids is found in [70]. One of the first practical applications of a fuzzy controller, the operation of a cement kiln, is described in [116]. The classification performance of an ANN in a prognostic problem in aviation medicine was

enhanced using similar FST-based extensions as above to the ANN's PEs [67] that conceptually resembled some components from the physician's decision process. In [117], fuzzy discriminant analysis, based on the technique described in [85], is used to successfully diagnose valvular heart disease. In [81], a fuzzy multi-layer perceptron is used for diagnosing hepatobiliary disorders. A good review of fuzzy neural networks is offered in [16].

Fuzzy integrals have also been used to combine multiple neural networks to improve the classification performance of any individual networks. For instance, [19] reports the success of a fuzzy integration technique that nonlinearly combines objective evidence, in the form of a membership function, with subjective evaluation of the worth of the individual neural networks with respect to the decision. In [28], possibility theory and its use in data fusion in poorly defined environments is discussed in detail.

4 Classifiers

This chapter describes the classifiers that will be used on the transformed data generated by the preprocessing techniques. The first, linear discrimination, is a traditional technique and the remaining three are ANN architectures: the MLP, the probabilistic neural network (PNN), and an ANN implementation of radial basis functions. Some enhancements of each classifier are also presented.

4.1 Linear Discriminant Analysis

When building a decision boundary between groups, it is not sufficient to simply examine the differences between the classes, the error distributions must also be taken into account. Notice the first decision boundary perfectly separating the ω_1 and ω_2 centroids in Figure 18. Now, assuming the first error distribution, this decision boundary continues to perfectly separate the two groups. However, if the second error distribution is the case, the first decision boundary is perpendicular to the best decision boundary (boundary 2). Therefore, in addition to taking into account the between-group variances, a classifier must also account for within-group variances. Linear discriminant analysis is a classifier strategy that builds linear decision boundaries between groups while taking into account between-group and within-group variances [76]. If the error distributions are the same, it can be shown that linear discriminant analysis constructs the optimal linear decision boundary between groups [56]. Unfortunately, this optimality is gained by the underlying assumption that the covariance matrix within each group is the same for each group, that is, the only difference among groups are different central tendencies. In real-world situations, this is seldom the case, different groups may give rise to different distributions. For instance, if the first group in Figure 18 has the first error distribution but the second group has the second error distribution, a linear decision boundary can no longer be constructed to completely separate the two groups.

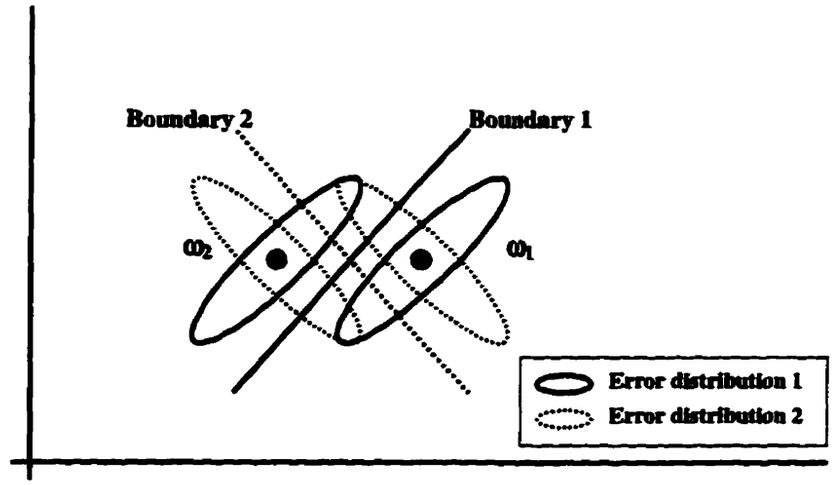

Figure 18: Error distributions affect discriminant functions

If x is an n-dimensional vector it will have a different probability distribution, $f_i(x)$, in each of the groups, $\omega_1, \omega_2, \ldots, \omega_k$, otherwise some or all of the groups are indistinguishable. As described in section 2.1, a classification rule can be defined by a partition of the input space into k exhaustive and mutually exclusive regions, $\Omega_1, \Omega_2, \ldots, \Omega_k$ with the decision rule that assigns to ω_i those vectors that fall in Ω_i. As the theoretical derivation of the following is not germane to the thesis and is thoroughly described in the literature [22], only an intuitive justification of linear discriminant analysis will be offered. Let q_i be the prior probability of observing a vector from ω_i. If the prior probabilities are not known, they may all be set to $1/k$, or proportional probabilities, N_i/N, may be used for each ω_i. A vector, x, should be allocated to the group for which the probability distribution, $f_i(x)$, is greater than any other distribution, while taking into account known prior probabilities. So,

$$x \in \omega_i \text{ if } q_i f_i(x) \geq q_j f_j(x) \; (\forall j = 1,\ldots,k). \qquad (21)$$

The two key assumptions for linear discriminant analysis are: all groups are multivariate normal populations (with different mean vectors, μ_i (i=1,...,k)); all groups have the same covariance matrix, W. Given these assumptions, then

$$q_i f_i(x) = q_i \left((2\pi)^n |W|\right)^{\frac{1}{2}} e^{\left(-\frac{1}{2}(x-\mu_i)^T W^{-1}(x-\mu_i)\right)} \qquad (22)$$

and, with some algebraic manipulations,

$$\log(q_i f_i(x)) = \log q_i + \tfrac{1}{2}\log\left((2\pi)^n |W|\right) - \tfrac{1}{2}(x-\mu_i)^T W^{-1}(x-\mu_i) \qquad (23)$$

hence,

$$\log(q_i f_i(x)) = \tfrac{1}{2}\log\left((2\pi)^n |W|\right) - \tfrac{1}{2}x^T W^{-1} x + \log q_i + \mu_i^T W^{-1}\left(x - \tfrac{1}{2}\mu_i\right). \qquad (24)$$

Given (21), it is clear that x should be allocated to the group for which (23) is highest. The linear discriminant function, $L_i(x)$, may now be defined for ω_i. Since the first two terms in (24) are the same for all groups, x should be assigned to the group ω_i for which

$$L_i(x) = \log q_i + \mu_i^T W^{-1}\left(x - \tfrac{1}{2}\mu_i\right) \qquad (25)$$

is greatest. If the prior probabilities are assumed to be equal then the first term in (25) may be ignored. The difference, $D_{ij}(x)=L_i(x)-L_j(x)=0$, defines the hyperplane in the input space that separates ω_i from ω_j [38]. Figure 19 shows a three class 2-dimensional classification problem and the decision boundaries produced by linear discriminant analysis. The decision rule is straightforward: if $D_{12}(x)>0$ and $D_{13}(x)>0$ then $x\in\omega_1$; if $D_{12}(x)<0$ and $D_{23}(x)>0$ then $x\in\omega_2$; otherwise, $x\in\omega_3$.

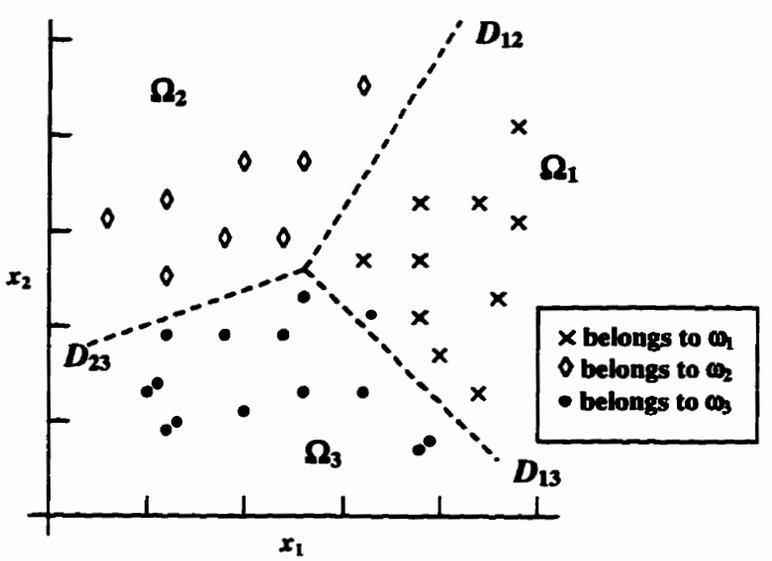

Figure 19: Decision boundaries produced by linear discriminant analysis

4.2 Multi-layer Perceptron

The MLP [45], a supervised feed-forward ANN employing the back-propagation algorithm, has served as a workhorse and a touchstone for many fruitful inquiries (Figure 20). The nonlinear transfer (activation) function, γ, is traditionally the logistic function,

$$\gamma(x) = (1+e^{-x})^{-1} \qquad (26)$$

however, any sigmoid function is permissible. A sigmoid function is an "S-shaped" function, and the logistic form of it maps the interval $[-\infty,\infty]$ onto $[0,1]$. If $|x|$ is small, then γ can be approximated by a linear function. The output of PE j is

$$x_j = \gamma\left(w_{j0} + \sum_i w_{ji} x_i\right). \qquad (27)$$

The transfer function is applied to the summation of the outputs of the PEs from the previous layer multiplied by the respective i weights. The term, w_{j0}, is the PEs bias (or threshold). Assuming n inputs, the geometrical interpretation of (27) is as follows. The summation term defines the orientation of an $(n-1)$-dimensional hyperplane about the origin in the n-dimensional input space. The bias term defines the distance of the hyperplane from the origin [29]. Non-linearity is introduced when multiple PEs are used in the same layer. As a notational convenience, the bias term may be thought of as an additional weight term for the PE except that its input is always equal to one. It may then be absorbed into the summation.

The global error function that is typically used in a MLP is

$$E = 0.5\sum_k ((d_k - o_k)^2) \qquad (28)$$

where the d_k's and o_k's are the respective components of the desired and actual outputs. Different performance indices may be used, however. For instance, an L_1 norm variant can be used instead of the L_2 norm in (28) in order to make the function more robust. The weight changes are calculated using a gradient descent strategy (delta rule)

$$\Delta w_{ji} = -\alpha\left(\frac{\partial E}{\partial w_{ji}}\right) \qquad (29)$$

where α is a learning coefficient in the range [0,1]. The local error for a PE is determined by solving (29) using (26), (27), and (28)(see [100] for a derivation). In general terms, a MLP may be considered a non-linear regression system that performs a gradient descent search through the weight space, searching for minima. Thorough discussions on MLPs and ANNs in general may be found in [25,47,61,113].

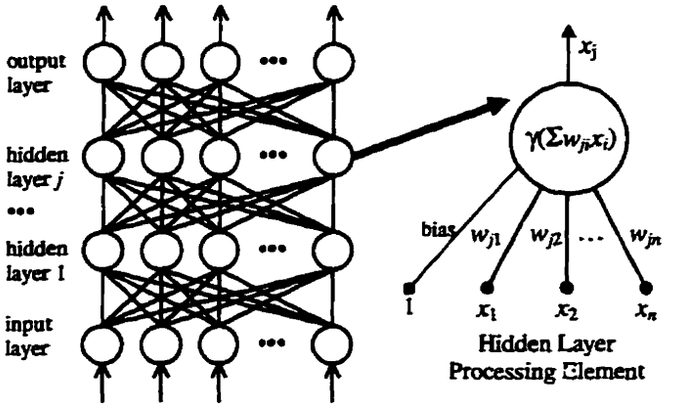

Figure 20: A multi-layer perceptron

4.2.1 Conventional enhancements

A number of enhancements may be made to MLPs that: increase the rate of convergence; increase robustness; or improve the accuracy of the final results. A few of these will now be discussed.

To increase the rate of convergence, a momentum term, β, in the weight update equation

$$\Delta w_{ji} = -\alpha\left(\frac{\partial E}{\partial w_{ji}}\right) + \beta\Delta w_{ji} \qquad (30)$$

may be used [72].

Using the hyperbolic tangent function as the transfer function instead of the logistic function typically improves the performance of a MLP [43]. The transfer function's output is a multiplier

in the weight update formula. The logistic function's range of [0,1] may cause a bias towards learning larger values. However, the hyperbolic tangent function is bipolar hence this will not occur. A gain term, g, may also be introduced into the sigmoid

$$tr(x) = (1+e^{-xg})^{-1} \qquad (31)$$

A large gain value may increase the rate of convergence but at the same time makes the MLP more susceptible to pitted error surfaces and may cause wild oscillations during learning.

Different learning and momentum rates may be used for each layer and/or after each of a set of predetermined number of iterations. A typical scenario is to use large learning and momentum values for the initial layers and/or the initial sets of iterations and successively smaller values for subsequent layers and/or sets of iterations. The end effect of this modulated learning strategy is to search for gross data features at the initial layers and/or during the initial sets of iterations and successively refine these detected features by subsequent layers and/or sets of iterations.

4.3 Probabilistic Neural Networks

The PNN has been successfully used as a general technique for solving pattern classification problems [106,111]. It uses design data to build probability density functions that are used to estimate the likelihood of a given vector falling into a particular category. For example, sonar spectra have been presented to a PNN to determine hull–to–emitter correlations in order to predict the likelihood of a given signal coming from submarines, ships, or other objects [75]. A PNN is an ANN implementation of a Bayesian classifier that uses Parzen estimators to build the required density functions. As a Bayesian classifier, a PNN can take advantage of *a priori* probabilities if they are available (for instance, if a test vector is equally likely to fall into class X or class Y but class X has a higher relative frequency than class Y, then the vector will, *inter pares*, be classified as a class X vector). If relative frequencies are not known then proportional probabilities are used. Compared with MLPs (and gradient descent methods in general), PNNs offer several advantages:

training is typically significantly faster; the design set may be modified without the need for extensive retraining periods; unlike MLPs that only guarantees convergence to a local minimum, as more design data is included, the PNN converges to a Bayesian classifier. This last point is important because a Bayesian classifier provides an optimum approach to pattern classification in terms of minimizing expected risk and, as such, is a benchmark of optimality. At the same time, PNNs retain the same advantages of MLPs: they are universal function approximators — arbitrary nonlinear decision boundaries can be constructed based solely on design data; in other words, they can robustly generalize.

Bayes theorem provides a method for performing optimal classifications; given enough data, it demonstrates how to classify a test vector with the maximum probability of success. Due to its sound theoretical foundations, the Bayesian classifier is often used as a standard against which other methods are evaluated [74].

Suppose that we wish to classify the magnetic resonance spectra of thyroid tissue (as described in section 3.3.3) to determine whether it is normal, moderately diseased, or severely diseased based on a two-dimensional feature vector; namely, the choline and lipid regions of the spectra. Assume that the probabilities of each disease state having the measured choline and lipid spectral properties are known. In other words, the two-dimensional probability density functions (choline and lipid) is known for normal, f_n, moderately diseased, f_m, and severely diseased, f_s, thyroid tissue. Also assume that, from medical history, the *a priori* probabilities of a tissue sample falling into one of the three classes are also known to be h_n, h_m, and h_s, respectively. If the *a priori* probabilities are not known then proportional probabilities may be used. Finally, let l_n, l_m, l_s be the respective loss or penalty incurred for misclassifying a tissue sample as normal, moderately diseased, or severely diseased (for example, there may be a greater financial (and ethical) cost associated with misclassifying moderately-diseased tissue as normal than for normal tissue to be misclassified as moderately-diseased). In general, each class may have a different

misclassification loss for each other class thereby resulting in a loss matrix. This complexity is avoided by assuming the loss to be equal and positive for each class. An optimal classification can be made by assigning the tissue sample to the class whose product, $f_\omega h_\omega l_\omega$, is greatest.

4.3.1 Parzen estimators

The ideal f for each class is its pdf [104]. If they are known then a PNN may be constructed whose architecture would correspond exactly to a Bayesian classifier. Unfortunately, pdfs are rarely known and must be approximated through the construction of a sampling histogram. As the number of sampling bins are increased, the histogram approaches the corresponding pdf assuming appropriate scaling to ensure the integral of the approximating curve is unity (a necessary criterion for pdfs). Parzen [86] developed a technique whereby pdfs may be estimated using sparse or inaccurate data sets. This technique — Parzen estimators — involves constructing unit area Gaussian functions centered at the values of the features for every design set vector. These Gaussian functions are summed and scaled to produce a composite curve. Parzen demonstrated that, as the number of design vectors increase, the composite curve asymptotically approaches the true pdf. Since PNNs use Parzen estimators, it is clear that the more design data used, the more accurate the final classification outcomes. However, it is not possible to determine the number of vectors required to estimate the pdf to a specified accuracy.

Parzen estimators can be easily extended to the n-dimensional case [110]. Moreover, it is unnecessary to compute the approximated pdf but rather only the values at each point, $x=[x_1, x_2, \ldots, x_n]$, to be classified. The value of the pdf of ω_j at x is:

$$f_j(x)=\frac{1}{2\pi^{\frac{n}{2}}\sigma^n N_j}\sum_{i=1}^{N_j} e^{-(x-x_i^j)^T(x-x_i^j)/(2\sigma^2)} \qquad (32)$$

where x^j_i is the i^{th} ω_j design vector, and $\sigma=\sigma(N_j)$ is a smoothing parameter which must satisfy two conditions [17],

$$\lim_{N_j \to \infty} \sigma(N_j) = 0 \qquad (33)$$

and

$$\lim_{N_j \to \infty} N_j \sigma(N_j) = \infty \qquad (34)$$

If

$$\sigma(N_j) = N_j^{-b} \qquad (35)$$

where $0<b<1$, then (33) and (34) are satisfied. If $\sigma<<1$ then each Gaussian constituent of (32), known as Parzen kernels, will be sharply peaked. As σ approaches zero, the PNN approximates a nearest neighbour classifier. Specht [107] demonstrated that classification performance is relatively insensitive to the choice of σ. However, as σ increases, the decision boundaries approach hyperplanes thereby limiting the classifier to functions that are linearly separable. Therefore, it is desirable to keep σ small in order that the more robust nearest neighbor scenario occurs.

An unknown test vector will be classified as belonging to ω_j if

$$h_j l_j f_j \geq h_i l_i f_i \left(\forall i \neq j\right) \qquad (36)$$

where h and l are the respective prior probabilities and loss functions for each class, if available (see section 3.2.1). Now if we rewrite the Parzen kernels from (32) as

$$e^{\left(2x^T x_l' - x^T x - x_l'^T x_l'\right)/(2\sigma^2)} \qquad (37)$$

and if we also normalize all input data ($x^T x=1$), the PNN implementation of the above is straightforward since (37) can then be rewritten as

$$e^{\left(x^T x_l' - 1\right)/\sigma^2} \qquad (38)$$

Normalization can be problematic if collinear vectors need to retain their distinctiveness — for instance, pixel values of the same image at different illumination levels. Although normalization simplifies its architecture, a PNN can deal with Parzen kernels of the form found in (37) rather than (38) with only a corresponding increase in complexity.

The term $\boldsymbol{x}^T\boldsymbol{x}_i^j$ is the inner product of the unknown vector and a design vector. If a PE has its incoming weights set to the design vector, $\boldsymbol{x}_i^j$, then a standard MLP treatment of that PE will implement (38) as a transfer function.

Figure 21 illustrates the architecture of a PNN. The input layer passes an n-dimensional vector to the normalization layer. The weights entering a pattern layer PE, $\boldsymbol{x}_i^j$, are simply the component values of the i^{th} design vector from the class ω_j. The output from each pattern layer is the value of the corresponding Parzen kernel, (38). Each PE, f_j, in the summation layer sums all Parzen kernels for ω_j. The classification layer is basically a competitive layer; if the summation layer PE, f_j, has a value greater than any other PE, f_i ($i \neq j$), then the corresponding classification layer PE, c_j, will output a 1, otherwise it will output a 0, thereby indicating the class of the current input vector.

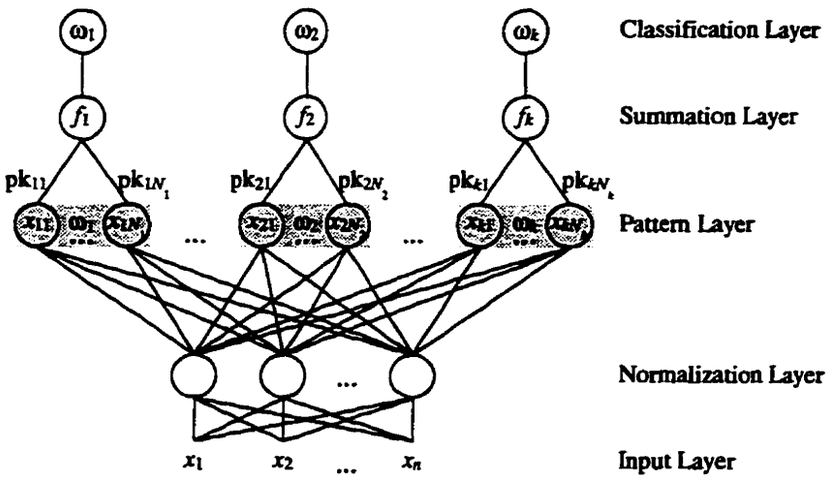

Figure 21: PNN architecture

4.4 Radial Basis Function Neural Networks

A radial basis function neural network (RBFN) has an internal representation of hidden PEs that are radially symmetric [45,121]. It should be noted that the literature has referred to RBFNs by different names: localized receptive fields [82], locally tuned processing units [83], regularization networks [37], and Gaussian potential functions [69]. Since radial symmetry is the essential concept with this architecture, the more descriptive, "RBFN" will be used. The output of a PE possessing radial symmetry is

$$f(x) = \varphi(\|x - \mu\|) \qquad (39)$$

where: μ is the PE's centre, represented by a vector in the input space, that is stored in the weights from the input layer to the PE; the distance metric (often Euclidean distance) determines how far an input vector is from μ; and, the transfer function, φ (typically a Gaussian function), must output high values when the distance from an input vector to μ is small, and low values otherwise. RBFNs are a class of universal function approximators [90] that are often used as classifiers [71]. That is, given an RBFN with enough hidden layer PEs, it can approximate any continuous function with arbitrary accuracy [42]. RBFNs typically train more quickly than traditional MLPs [119]. Also, there is the useful feature that the hidden PEs represent density functions for the input space and may be used as a probability measure for new input vectors. However, there are also several problems with this architecture: since the receptive fields are localized they do not perform well if discriminatory features are globally distributed throughout the input space; the selection of the number of receptive fields is strictly an *ad hoc* procedure.

Figure 22 illustrates the topology of a RBFN. If μ_i is a column vector representing the centre of pattern layer PE i, and σ_i is the diameter of its receptive region, then the PE's output, f_i, for a given test vector, $x=[x_1, x_2, \ldots, x_n]$, is

$$f_i(x) = e^{-(x-\mu_i)^T (x-\mu_i)/(2\sigma_i^2)} \qquad (40)$$

PE j in the output layer generates the sum, y_j, of the product of the pattern layer f_is and the respective weights;

$$y_j = \sum_i f_i w_{ij} \qquad (41)$$

The normalization layer is optional. PE j in the normalization layer generates, c_j, which is the normalized y_j from the output layer

$$c_j = y_j / \sum f_i \qquad (42)$$

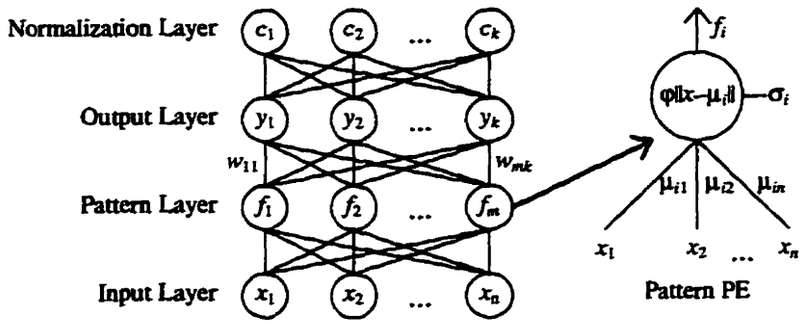

Figure 22: RBFN architecture

5 Conventional Preprocessing Methodologies

As discussed in section 3.1.2, many preprocessing strategies exist that transform the input space prior to presentation to a classifier. Three main problems potentially exist when trying to build a classification system that deals with a small set of high-dimensional data. Of course, the first is the computational burden placed upon the classifier. The second is overfitting: the classifier may focus on meaningless or unimportant idiosyncrasies of individual design cases instead of building a genuine generalization, based on the design set, that may be successfully used on a test set. The final problem is correlation between variables — variables can be highly correlated even in a low-dimensional problem (or completely independent in a high-dimensional space, for that matter), however, the probability of interdependencies in general increases with the dimensionality of the input space. Many classifiers assume that input variables are independent and high correlation can seriously degrade their performance. This chapter reviews two traditional methods: adjustments to the receptive fields of the RBFN, and principal component analysis.

5.1 Receptive Fields

Figure 23 illustrates a two-dimensional example of (40) and (41) where the final outputs are not normalized (hence we can ignore (42)). Figure 24 shows the response of the RBFN, given a two-dimensional input vector. The top of the Gaussian bump is $\mu=[\mu_1, \mu_2]$ and the distance from μ to the point at which the curve flattens out is σ; that is, the function is radially symmetric around μ. When an input vector, $x=[x_1, x_2]$, is equal to μ, the response function produces its maximum output, one. As x deviates from μ, the response quickly drops to zero. The range of the receptive field of the response function is determined by the value of σ. (The receptive field of a RBFN pattern layer PE differs from that of a neuron in the visual cortex as well as other regions of the human brain, where the receptive field is determined by neuronal interconnectivity. In contrast, a RBFN PE's receptive field range is controlled by the shape of the exponential

function. Since this report is not concerned with the biological plausibility of the ANNs under investigation, this is not an issue of contention.) The values μ and σ may analogously be viewed as the mean and standard deviation of the response curve, respectively.

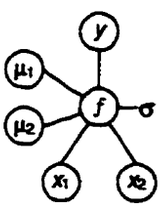

Figure 23: A two-dimensional RBFN PE

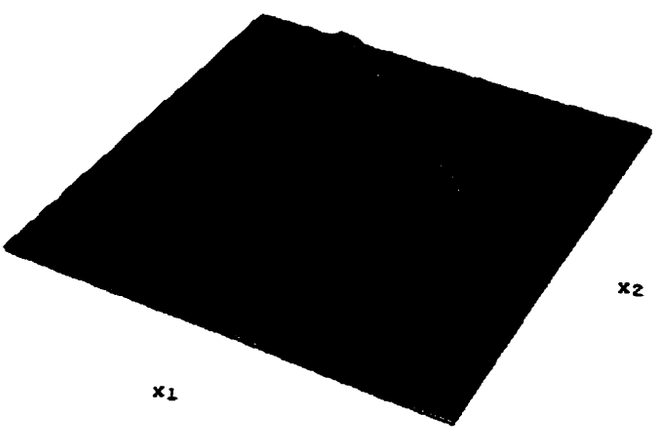

Figure 24: Receptive field of a two-dimensional RBFN

The response function of a RBFN PE diminishes rapidly as an input vector deviates from the PE's mean. Since this function is typically (but not exclusively) characterized by a Gaussian exponential function it gives rise to a localized "Gaussian bump" response. The set of pattern layer PEs is designed so that their responses cover all significant regions of the input vector space. In the simplest case, both the pattern layer and output layer weights remain fixed; there is no training at all. Further there is one pattern layer PE for every design vector. In a slightly more complex extension, only the output layer weights are trained; this is a straightforward, and rapid, training of a single layer linear system. A further extension includes training the pattern layer weights as well as the location and shape of the response curves. In this section, we will examine several strategies for training the different parameters in a RBFN. As with other ANNs, a RBFN has two operational modes: a training mode where the parameters such as μ_i, σ_i, and the weight

matrix, are adjusted in order to minimize the mean error (over the design vector set) between the desired classification outcomes and the actual outcomes produced by the RBFN; and a test mode where the performance of the trained RBFN is evaluated by using previously unseen vectors.

There are several alternatives for determining the location of the centers of the receptive fields of the pattern layer PEs. The simplest alternative is to have one PE for every vector in the design set. However, this may become completely impractical if there are a large number of design vectors; the amount of time required to train such a network as well as to test it would be inordinately great. A more robust strategy is to take advantage of the fact that design vectors typically tend to occur in clusters, and use an unsupervised clustering algorithm to reduce the number of pattern layer PEs.

Standard *k*-means clustering is one possible strategy to compute a set of μ_is. This algorithm assumes that all of the design vectors are available and that there are a fixed number of clusters (centres). Approximately k centres are usually selected (for clarity, assume k centres here). The standard *k*-means algorithm will ensure that the sum of the squares of the distances between each design vector and its closest centre is a local minimum. The algorithm begins with a set of k random centres. Each design vector is examined to determine the closest centre to it. A new set of centres is computed by taking the average of all design vectors, for each centre, and using those averages as the new centres. This step is repeated for a fixed number of iterations or until the membership function no longer changes.

The adaptive *k*-means clustering algorithm is a modification of the previous algorithm that does not require retention of past design vectors. It is basically competitive Kohonen learning [61] that begins with a set of k random centres. For every design vector, x_i, the centre, μ_j, closest to x_i, is modified as follows,

$$\mu_i^{new} = \mu_i^{old} + \alpha(x_i - \mu_i^{old}) \qquad (43)$$

where α is a learning rate that decreases with the number of epochs. This step repeats for a fixed number of iterations or until the learning rate decreases to zero.

One final variation involves the dynamic initialization of the adaptive *k*-means clustering algorithm to ensure that all centres are actually used. The *k* centres are initially disabled. For each design vector, x_i, if it is within a specified distance to the closest enabled centre, then modify that centre using (43), otherwise, enable a new centre at x_i. The termination condition is the same as with the adaptive *k*-means algorithm.

The radius of the receptive region of y_i is determined by the σ_i. If the μ_is are widely separated then the σ_is should be large to cover the gaps. If the μ_is are tightly packed then the σ_is should be small enough to accurately retain the distinctiveness of each receptive field. One technique that may be used to determine the σ_i is to use a *P*-nearest neighbour heuristic. Given a centre, μ_i, let $i_1, i_2, \ldots, i_p$ be the indices of the *P* centres nearest to μ_i. Then the corresponding σ_i is

$$\sigma_i = \sqrt{P^{-1}\sum_{p=1}^{P}\left\|\mu_i - \mu_{i_p}\right\|^2} \qquad (44)$$

In order to simplify the computations required for (44), *P* is often set to one so that only the single nearest neighbour is considered.

Once the μ_is and σ_is have been selected, the output layer weight matrix may then be optimized. A standard technique is to use a supervised training strategy such as gradient descent learning as described by (29). Most of the y_is will be close to zero for a given input vector since that vector will be near only one receptive field. As a consequence, the corresponding weight changes will be small. To improve training time, this fact can be exploited by ignoring the receptive fields with small activations.

An ill-advised strategy to determine the values of the weights is to treat the problem as a solution of the matrix equation, $W=Y^{-1}D$; where *W* is the weight matrix, *D* is a matrix whose rows

are the desired outcomes, and Y is a matrix whose rows are the outputs from the output layer for each design set vector. The matrix Y is generally not invertible because it is typically not square. Further even if a pseudoinverse exists it may not be easily determined [108]. The matrix may be ill-conditioned because it is singular or nearly singular. Even more complex techniques such as singular value decomposition may fail because of the possible limited accuracy of the results.

5.2 Principal Component Analysis

The motivation behind principal component analysis, first described by Pearson [87] with a practical computing method described by Hotelling [49], is to find a set of directions that explain as much of the variability of the original data as possible. In other words, given a set of N n-dimensional points, the principal components are a new set of orthogonal linear coordinates such that the variances of the original points with respect to these derived coordinates are in decreasing orders of magnitude [9]. As a result each principal component is uncorrelated with the other principal components (in a normal distribution, they are statistically independent). Moreover, it can be shown [65] that no other set of k variables can account for more of the variability in the original data than the first k principal components.

The first principal component, Y_1, of the original input variables x_1, x_2, ..., x_n, is the linear combination

$$Y_1 = \sum_{i=1}^{n} a_{1i} x_i, \ \sum a_{1i}^2 = 1 \qquad (45)$$

The constraint on the coefficients is necessary otherwise the variance of Y_1 can be increased simply by increasing the value of any coefficient. The second principal component, Y_2, would be computed in a similar fashion to (45). Figure 25 is a plot of some bivariate data and its two principal components. It is clear from this figure that an additional constraint, orthogonality to the first principal component, is required to compute the second principal component, otherwise it would simply be driven to the first principal component. Orthogonality is ensured by restricting

the variables of the second principal component to those that are uncorrelated with the first principal component. As a result of this orthogonality constraint, if there are n variables then there can be up to n principal components [76]. In fact, if the original variables are completely uncorrelated, then all n principal components must be used to take into account the variance in the original variables. In this particular case, principal component analysis serves no useful purpose since the motivation behind the technique is to reduce the dimensionality of the original input space. However, in "real-world" high-dimensional data, the converse is usually true; the variables are highly correlated and hence only $1 \leq k << n$ principal components are required to account for all of the variation.

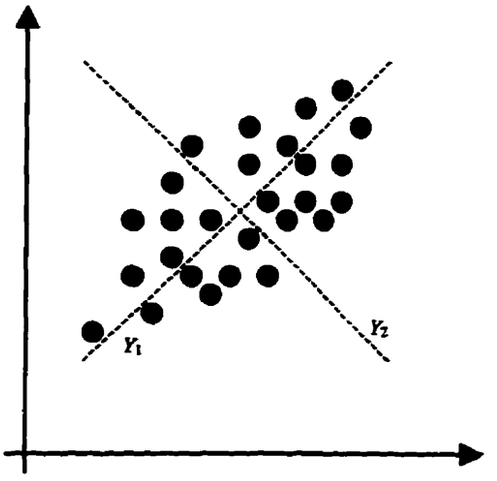

Figure 25: Two principal components, Y_1 and Y_2

Determining the principal components is a straightforward process involving the computation of the eigensystem of the original data's covariance matrix, C, whose element c_{lm} is the sample covariance between variables l and m

$$c_{lm} = \frac{1}{N-1}\sum_{i=1}^{N}(x_{il} - \mu_l)(x_{im} - \mu_m). \qquad (46)$$

The proof that this is the case will not be presented here; it is not difficult but somewhat lengthy and can be found in any good text on multivariate statistics [32,38,65,103]. The variances of the principal components are the eigenvalues of C, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$ (the covariance matrix is

quadratic and hence admits no negative eigenvalues). The variance of a principal component, Y_i, is λ_i and its constants a_{i1}, a_{i2}, ..., a_{in} are the elements of the corresponding eigenvector. A potential problem here is that the significance of a variable in principal component analysis changes with a change of scale of one or more of the variables. In order to avoid a variable having an undue influence on the principal components the original variables can be *standardized* (means of zero and variances of one). If the variables are standardized then, instead of using the sample covariance matrix, C, one may use $C^{\cdot}$, the sample correlation matrix.

The principal components that are computed for the data sets described in section 3.3 use their respective correlation matrices. The strategy employed in this thesis is to take the first k principal components whose cumulative variance exceeds some threshold (>95%). This reduction is significant, that is, $k<<n$: it is often the case that more than 80% of the cumulative variance of sets of high-dimensional (n>500) biomedical spectra acquired from magnetic resonance spectrometers, are accounted for by only the first one or two principal components.

PCA is often an effective preprocessing technique [7] but it suffers from several deficiencies. First, if new data are to be analyzed then the principal components of the original data need to be re-computed and the principal components of the new data must be calculated. Second, it is not possible, in general (and especially for high dimension problems), to determine what input features are relevant in the classification, that is, original input values cannot be determined solely using the principal component values. Unfortunately, it is often important to make such determinations in order to have a better understanding of the problem at hand. Finally, PCA orders the components based on maximal variance. Unfortunately, this does not necessarily translate into maximal discriminatory power [56]. For instance, say the method used to acquire values for a particular variable is extremely prone measurement error, then this variable will have a high variance. Now, assuming this variance is greater than other variables, the first principal component will be approximately equal to this suspect variable, and hence, this principal

component will be useless in discriminating between groups. Conversely, a highly discriminatory variable may have an extremely small variance and hence will not contribute to the first few principal components. Section 8.2 presents examples where better discrimination is achieved using sets of principal components other than the first k. In summary, the moral of all this is: maximal discriminatory power is not the same as maximal variance.

6 Fuzzy Data Encoding and Gold Standard Burnishing

Fuzzy encoding, the process of determining the respective degrees to which a datum belongs to a collection of fuzzy sets and subsequently using these membership grades in place of the original datum, is presented. Two preprocessing strategies are presented to deal with tarnished gold standards. *Enhancing gold standards* by incorporating non-subjective within-group centroid information via a fuzzy set theoretic approach is also discussed. The second uses a robust estimation of deviations from group medians for the reclassification of spectra in a design set. This *robust reclassification* is more radical than enhancing the gold standard in that individuals in the design set may be assigned to another group.

6.1 Fuzzy Encoding of Feature Space

6.1.1 Fuzzy interquartile encoding

Fuzzy encoding involves taking a single input value and intervalizing it across a collection of fuzzy sets, thereby producing a list of degrees of membership for each of the fuzzy sets. In other words, if we have s fuzzy sets, F_1, F_2, ..., F_s, and f_i is the membership function for fuzzy set i then the list of values for a single input value x is $\{f_1(x), f_2(x), ..., f_s(x)\}$. Selecting intervals for the fuzzy sets is usually an experimental or heuristic process and is similar to the techniques used in standard 1–of–k intervalization encodings. The purpose of intervalization is to reduce the effects of noise in the data as well as to transform the problem in such a way that a non-linear regression model such as MLP can provide better solutions. The membership functions are simple enough to define once the intervals have been selected because the definition corresponds to 1–of–k intervalization with the addition of gradual transitions at the respective interval boundaries.

Now let us derive a formula to generate a collection of membership functions. First, select the number of fuzzy sets, m, that are to be used. Let w be the width of the top of the trapezoid of the fuzzy sets. If w=0, then the f_i's are triangular fuzzy sets Let b, $0 \le b \le 1$, be the boundary value at

the intersection of the fuzzy sets. For simplicity, b is constant for each intersection. Let l_i and r_i be the left and right boundary, respectively, of F_i such that $f_i(l_i)= f_i(r_i)=b$. Let $\underline{l}_i$ and $\underline{r}_i$ be the left and right boundary, respectively, of F_i such that $f_i(\underline{l}_i)= f_i(\underline{r}_i)=0$ and for all x if $f_i(x)=0$ then $x<\underline{l}_i$ or $x>\underline{r}_i$. Finally, let x be the original non-encoded (NE) input value. Then,

$$f_i(x)=\begin{cases} 1\wedge\left(0\vee\left(1+w-2\dfrac{1+w-b}{r_i-l_i}\left|x-\dfrac{l_i+r_i}{2}\right|\right)\right) & \text{if} \quad l_i<r_i \\ 1 & \text{if} \quad l_i=r_i=x \\ 0 & \text{if} \quad l_i=r_i\neq x \end{cases} \tag{47}$$

where $\vee$ and $\wedge$ are the *max* and *min* operators, respectively. The bottom two cases define a delta function when $l_i=r_i$. This delta function satisfies the definition of a fuzzy set: it is monotonic and it maps onto the unit interval. Figure 26 shows two trapezoidal fuzzy sets constructed using (47) overlapping at b. Note that since $f_i(r_i)=f_{i+1}(l_{i+1})=b$, $r_i=l_{i+1}$ ($\forall i=1, \ldots, c-1$). It should also be noted that the corresponding fuzzy sets are symmetric about l_i and r_i.

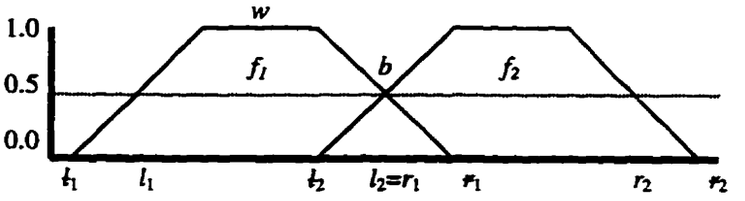

Figure 26: Construction of two fuzzy sets

Substituting l_i for x in (47) gives

$$f_i(l_i)=1\wedge\left(0\vee\left(1+w-\frac{1+w-b}{r_i-l_i}\left|l_i-r_i\right|\right)\right) \tag{48}$$

and, canceling terms,

$$f_i(l_i)=1\wedge(0\vee b))=b. \tag{49}$$

Similarly, substituting r_i for x in (47) gives

$$f_i(r_i) = b \qquad (50)$$

Now let us determine l_i and r_i. We need to find x such that $f_i(x)=0$. We can ignore the $\wedge$ and $\vee$ operators (to solve $1 \wedge \vartheta' = 0$, one must solve $\vartheta' = 0$ and to solve $\vartheta' = 0 \vee \vartheta = 0$, one must solve $\vartheta = 0$). From (47), we have

$$1 + w - 2\frac{1 + w - b}{r_i - l_i}\left| x - \frac{l_i + r_i}{2} \right| = 0 \qquad (51)$$

collecting terms,

$$\left| x - \frac{l_i + r_i}{2} \right| = \frac{(r_i - l_i)(w+1)}{2(1 + w - b)} \qquad (52)$$

and, hence,

$$r_i = \frac{(2w - b + 2)r_i - bl_i}{2(w - b + 1)}, l_i = \frac{(2w - b + 2)l_i - br_i}{2(w - b + 1)} \qquad (53)$$

When b is at least 0.5 then there exists a strict 1–1 correspondence between the fuzzy encoding and the original input value. Since a particular fuzzy encoding can be produced by only one input value, the fuzzy encoding of the data does not change the nature of the problem. If $b<0.5$ then we have a 1–many correspondence and the information content of the fuzzy encoding is reduced and hence the nature of the problem is changed. Furthermore, because of the relationship across each fuzzy set, the encoding does not introduce any extra degrees of freedom into the problem.

Given a data set, a method is now required to determine appropriate values for l_i and r_i. One strategy is to use specific percentiles for each l_i and r_i. The P^{th} percentile of a sample of n observations is a value such that P% of the area under the relative frequency distribution for the observations lies to the left of the P^{th} percentile and (100-P)% of the areas lies to its right [78]. The specific percentiles used are the 25th percentile, or lower quartile (Q_L), the 50th percentile, or midquartile (more commonly referred to as the median (m)), and the 75th percentile, or upper

quartile (Q_U). To calculate the quartiles for small data sets, where it may be difficult to use relative frequency distributions, the measurements may simply be ranked in increasing order of magnitude and the appropriate values selected: Q_L is the measurement with rank ¼(n+1), rounded to the nearest integer (rounded up if it falls halfway); Q_U is the measurement with rank ¾(n+1), rounded to the nearest integer (rounded down if it falls halfway); and, m is the measurement with rank ½(n+1), if n is odd, or the mean of the measurements with ranks ½n and ½(n+2), if n is even. In order to effect uniform coverage, the quartiles are computed for each coordinate, x_i, and the fuzzy sets, F^i_1, F^i_2, F^i_3, F^i_4, are constructed around them. The corresponding membership functions for these four fuzzy sets are used to generate the fuzzy encoded data. To ensure a strict 1–1 mapping between the non-encoded and fuzzy encoded values, w=0 and b=0.5. Specifically, the membership functions are

$$f_1^j(x_j)=1\wedge\left[0\vee\left[1-\left|x-0.5\left(\alpha^j+Q_L^j\right)\right|/\left(Q_L^j-\alpha^j\right)\right]\right] \quad (54)$$

$$f_2^j(x_j)=1\wedge\left[0\vee\left[1-\left|x-0.5\left(Q_L^j+m^j\right)\right|/\left(m-Q_L^j\right)\right]\right] \quad (55)$$

$$f_3^j(x_j)=1\wedge\left[0\vee\left[1-\left|x-0.5\left(m^j+Q_U^j\right)\right|/\left(Q_U^j-m^j\right)\right]\right] \quad (56)$$

$$f_4^j(x_j)=1\wedge\left[0\vee\left[1-\left|x-0.5\left(Q_U^j+\beta^j\right)\right|/\left(\beta^j-Q_U^j\right)\right]\right] \quad (57)$$

where α^j, Q^j_L, m^j, Q^j_U, and β^j are the smallest value, lower quartile, median, upper quartile, and largest values of coordinate j, respectively. The t_i and r_i boundaries (ignoring the coordinate index) for (54)–(57) are

$$t_1=\frac{3\alpha-Q_L}{2}, r_1=\frac{3Q_L-\alpha}{2} \quad (58)$$

$$t_2=\frac{3Q_L-m}{2}, r_2=\frac{3m-Q_L}{2} \quad (59)$$

$$t_3 = \frac{3m - Q_U}{2}, f_3 = \frac{3Q_U - m}{2} \quad (60)$$

$$t_4 = \frac{3Q_U - \beta}{2}, f_4 = \frac{3\beta - Q_U}{2} \quad (61)$$

A graphical representation of the membership functions, (54)–(57), is shown in Figure 27. This strategy is not restricted to four fuzzy sets: any number of percentiles may be used.

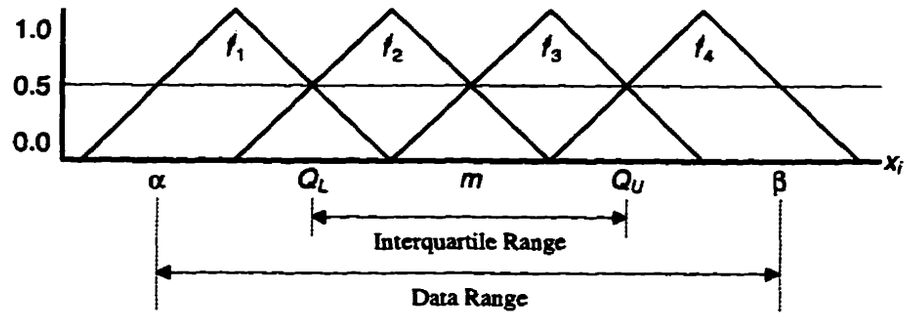

Figure 27: Membership functions used to fuzzy encode coordinate x_i

The fuzzy sets shown in Figure 27 assume that the distribution of the underlying data is normal and that there are sufficient samples to make that distribution apparent. However, this is seldom the case when dealing with real-world data. Fortunately, normality assumptions are not built into this method. For example, assume a set of data where α=-1, Q_L=0, m=Q_U=3, and β=9 (see Figure 28). The underlying distribution of this data is highly skewed: a dense population of points are around 3 and values around 9 may be outliers. Notice that the fuzzy set f_3 is a delta function since m=Q_U.

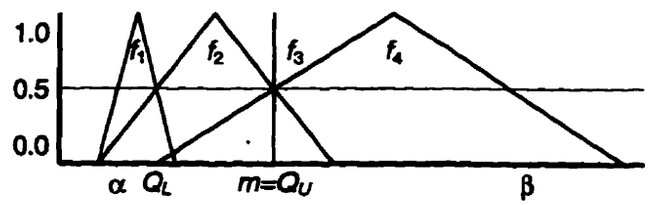

Figure 28: Membership functions used to fuzzy encode highly skewed data

Fuzzy encoding exhibits several useful properties. First, since the membership functions map values onto the unit interval, the data are automatically scaled. This is particularly useful in the classification process since scaled data diminish the effects of extreme variances across features.

Without scaled data, features with large variances will predominate over features with small variances although the latter features may be discriminatory. Another beneficial property is that values that may be considered outliers impact less severely upon classifiers, such as the MLP, that employ any type of iterative adjustments to its error function. This does not mean that samples with features that are outliers are removed during the design or test phases of the classification process, however. The farther a value is from the interquartile range, the fuzzy encoded values all tend to zero. In the case of a MLP where its hidden layer PEs are summing products of weights and input values this is important since, if the fuzzy encoded values of an outlier are all zero or near zero, those values will contribute very little to the learning process regardless of the PEs weights; an extremely useful feature if the original value is indeed an outlier yet if it is not an outlier it still does contribute to a degree. For instance, using the example quartiles from the previous paragraph, if x=10 then the fuzzy encoded values are {0, 0, 0, 1/3}. If x=12, the fuzzy encoded values are {0, 0, 0, 0}. Conversely, values that are within the interquartile range will contribute strongly to the learning process. If x=3, the fuzzy encoded values are {0, ½, 1, ½}. If x=2, the fuzzy encoded values are {0, 5/6, 0, 1/3}.

6.1.2 Dimension-preserving fuzzy interquartile encoding

A variant of fuzzy interquartile encoding exists that does not increase the dimensionality of the feature space. Instead of constructing four triangular fuzzy sets around the quartiles of each feature, a single piece-wise linear fuzzy set is constructed whose vertices are the lower quartile, median, and upper quartile. Figure 29 is an example of a single membership function constructed from a feature's quartiles where $h \in (0, 1)$ is a membership threshold such that feature values within the interquartile range will have membership values greater than h, values outside the interquartile range but within the minimum and maximum values will have values less than h.

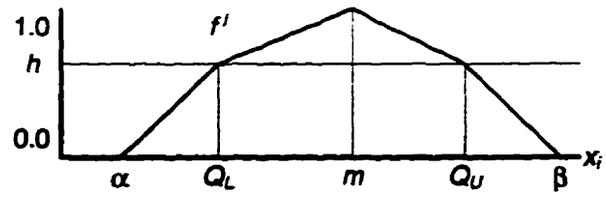

Figure 29: A single membership function constructed from feature quartiles

Assuming $\alpha<Q_L<m<Q_U<\beta$ (respectively, the smallest value, lower quartile, median, upper quartile, and largest value for feature j), the membership function for the fuzzy set of feature j is

$$f^j(x)=\begin{cases} h\dfrac{x-\alpha}{Q_L-\alpha} & \text{if} \quad \alpha \le x < Q_L \\ (1-h)\dfrac{x-Q_L}{m-Q_L}+h & \text{if} \quad Q_L \le x < m \\ (h-1)\dfrac{x-m}{Q_U-m}+1 & \text{if} \quad m \le x < Q_U \\ -h\dfrac{x-Q_U}{\beta-Q_U}+h & \text{if} \quad Q_U \le x < \beta \\ 0 & \text{if} \quad x<\alpha \vee x>\beta \end{cases} \tag{62}$$

For the degenerate cases, occurring when the data are extremely skewed,

$$f^j(x)=\begin{cases} 1 & \text{if} \quad \alpha<Q_L=m \vee \beta>Q_U=m \\ h & \text{if} \quad m>Q_L=\alpha \vee m<Q_U=\beta \end{cases} \tag{63}$$

6.1.3 Fuzzy cluster encoding

This method employs the fuzzy c-means algorithm [11,122] to determine a set of c centroids for the data. A distance measure is then used to determine how similar an individual is to each centroid. These values are substituted for the original data. A particularly useful property of this method is that it changes the dimensionality of the problem space from n to c.

Let $X=\{x_1, x_2, \ldots, x_N\}$ be a set of data where $x_k \in \Re^n$. A fuzzy c-partition or pseudopartition of X is a family of fuzzy subsets of X, denoted by $P=\{u_1, u_2, \ldots, u_c\}$ such that

$$\sum_{i=1}^{c} u_i(x_k) = 1 \qquad \forall k = \{1 \ldots N\}$$
$$0 < \sum_{k=1}^{N} u_i(x_k) < N \qquad \forall i = \{1 \ldots c\} \tag{64}$$

Clustering involves finding the fuzzy c-partition and the associated centroids by which the structure of the data is represented as best as possible, specifically, that the associations are strong within clusters and weak between clusters. The criterion used as a performance index is computed by first calculating the cluster centres associated with the pseudopartition P

$$v_i = \frac{\sum_{k=1}^{N} [u_i(x_k)]^m x_k}{\sum_{k=1}^{N} [u_i(x_k)]^m} \tag{65}$$

where v_i (i=1...c) is the centroid associated with the partition u_i and $m \in (1,\infty)$ governs the influence of membership grades. Using (65), v_i the weighted average of data in u_i where the weight of x_k is the m^{th} power of the membership grade of x_k in the fuzzy set u_i. The performance index $Q(P)$ may now be defined in terms of these centroids

$$Q(P) = \sum_{k=1}^{N} \sum_{i=1}^{c} [u_i(x_k)]^m \|x_k - v_i\|^2 \tag{66}$$

where $\|x_k - v_i\|^2$ is the distance between an individual and a centroid (any inner product-induced norm in $\Re^n$ may be used but the Euclidean norm is most often selected). $Q(P)$ measures the weighted sum of distances between cluster centroids and individuals in the corresponding fuzzy clusters; a small values indicates a good P, hence the objective of the fuzzy c-means algorithm is to find a fuzzy c-partition that minimizes Q. This optimization may be solved using the following steps:

0. Select c, m, and a small positive number, ε, as a stopping criterion. Let τ=0 and select an initial c-partition $P^{(\tau)}$ satisfying (64).

1. Compute the c centroids $v_i^{(\tau)}$ using (65) for $P^{(\tau)}$.

2. Compute $P^{(\tau+1)}$. Do the following $\forall x_k$:

   if $\|x_k - v_i^{(\tau)}\|^2 > 0 \ \forall v_c$, then

$$u_i^{(\tau+1)}(x_k) = \left[\sum_{j=1}^{c} \left(\frac{\left\| x_k - v_i^{(\tau)} \right\|^2}{\left\| x_k - v_j^{(\tau)} \right\|^2} \right)^{\frac{1}{m-1}} \right]^{-1} \qquad (67)$$

   Otherwise: when $\|x_k - v_i\|^2 = 0$, $u_i^{(\tau+1)}(x_k) = 0$; and $u_i^{(\tau+1)}(x_k)$ is set to any non-negative real number such that

$$\sum u_i^{(\tau+1)}(x_k) = 1 \qquad (68)$$

   for all remaining i.

3. If

$$\max_{i \in \{1 \ldots c\}, k \in \{1 \ldots N\}} \left| u_i^{(\tau+1)}(x_k) - u_i^{(\tau)}(x_k) \right| \leq \varepsilon \qquad (69)$$

   then stop. Repeat steps 1–3, otherwise.

As m approaches 1, the fuzzy c-means algorithm converges to a classical hard means algorithm as described in section 5.1 [80]. As m approaches ∞, all cluster centroids tend towards the centroid of X. In other words, the pseudopartition becomes fuzzier as m increases. No theoretical basis exists for an optimal m but Bezdek proved [10] that the algorithm converges for $m \in (1,\infty)$. Empirical evidence suggests that good results are typically obtained for $m \in [1.5,2.5]$ (in the discussion that follows $m=2$) [12].

The membership function, $u_i(x_k)$, that is used is based upon the update equation (67) in the second step of the fuzzy c-means algorithm. Specifically,

$$u_i(x_k)=\begin{cases}\left[\sum_{j=1}^{c}\frac{\|x_k-v_i\|^m}{\|x_k-v_j\|^m}\right]^{-1} & \text{if} \quad x_k \neq v_j \\ 1 & \text{if} \quad x_k = v_{j\neq i}\end{cases} \qquad (70)$$

is a measure of the degree to which an individual x_k belongs to the cluster centroid v_i that also takes into account the individual's membership in other cluster centroids (as previously mentioned m=2). That is, for two individuals, x_1 and x_2, that are equidistant (in a strict Euclidean sense) from a cluster centroid v_1, if x_1 is near another cluster centroid (once again in a strict Euclidean sense) and x_2 is not, then $u_1(x_1)<u_1(x_2)$. Furthermore, as the Euclidean distance between x_k and all cluster centroids approaches ∞, $u_i(x_k)$ approaches $1/c$. Figure 30 is a plot of $u_i(x)$ with two cluster centroids, v_1 and v_2. Note that $u_1(x)$ and $u_2(x)$ both approach ½ as the distance between x and the cluster centroids increase. Note further that $u_1(x)$ and $u_2(x)$ are ½ when x is between v_1 and v_2.

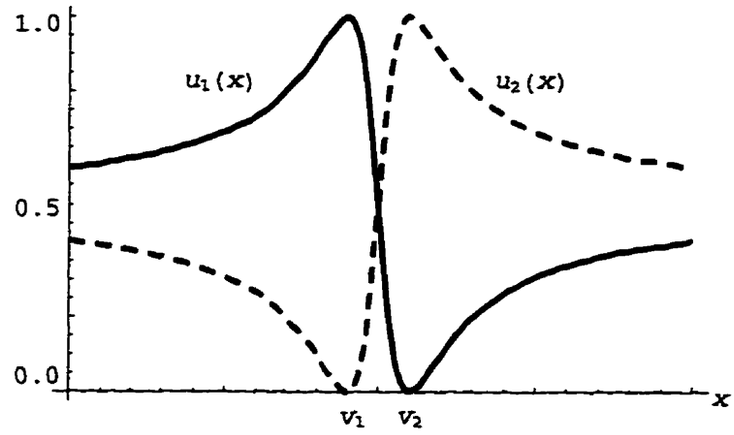

Figure 30: Plot of $u_i(x)$ with 2 cluster centroids

The encoding is straightforward: replace every individual x_k with $[u_1(x_k), u_2(x_k), \ldots, u_c(x_k)]$.

6.1.4 Class-wise variants

The encoding methods described in the previous sections do not take into account any class information. This may be problematic when classes have extremely different distributions or when the underlying probability density functions for each class significantly overlap. For

example, Figure 31 shows the probability density functions for ω_1, whose data are normally distributed, and ω_2, whose data are bimodally distributed about ω_1.

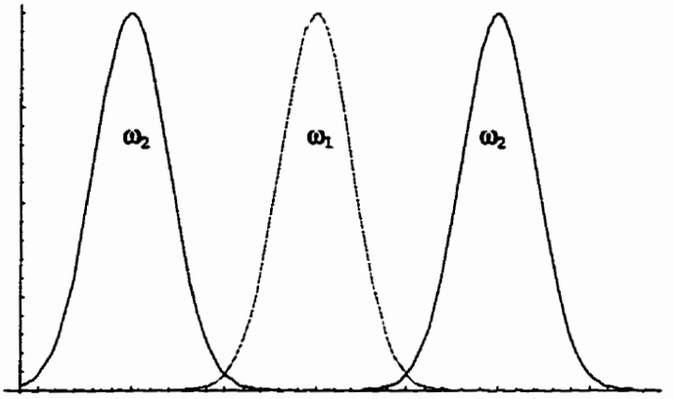

Figure 31: Probability density functions for a normally distributed class between a bimodal distribution

Using the membership function, $u_i(x)$, described in section 6.1.3 and c=3, the fuzzy cluster encoding should find one cluster centroid near the mode of ω_1 and one centroid near each of the two modes of ω_2 (Figure 32). Intuitively, this should give good discriminatory performance since $u_i(x)$ will be high for individuals near the class modes. However, if c=2 one centroid will be placed between the one mode of ω_2 and the mode of ω_1 and the other centroid will fall between the mode of ω_1 and the other mode of ω_2 (Figure 33). Now the discriminatory power will more than likely be poor since $u_i(x)$ will be greatest for a few individuals between the modes.

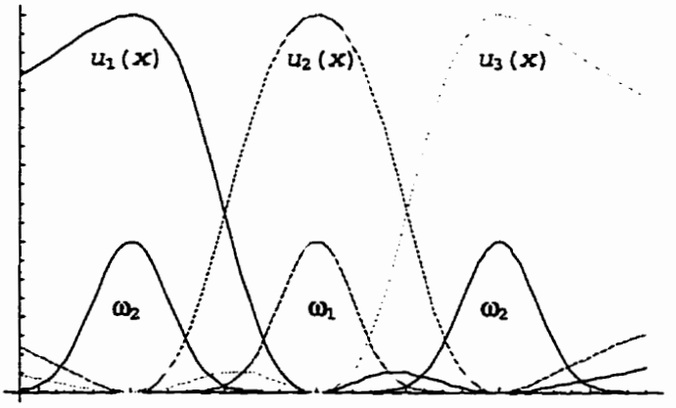

Figure 32: Good discriminatory performance using fuzzy cluster encoding with 3 centroids

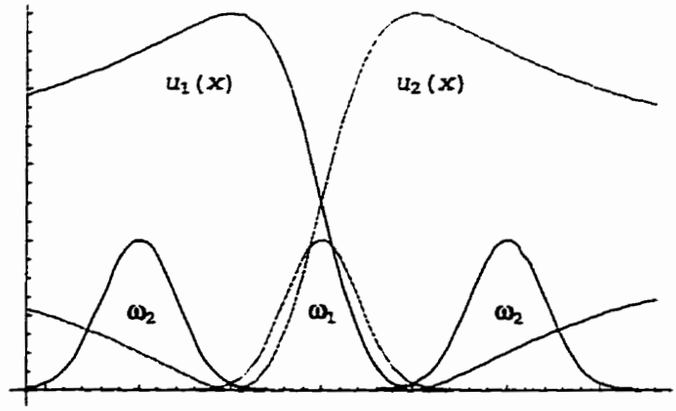

Figure 33: Poor discriminatory performance using fuzzy cluster encoding with 2 centroids

Now, if two cluster centroids are used for each class, the previous problem dissolves. The two cluster centroids for ω_2 will be centred near its two modes, whereas the two cluster centroids for ω_1 will both be near its mode (Figure 34). The discriminatory power should be similar to that in Figure 32.

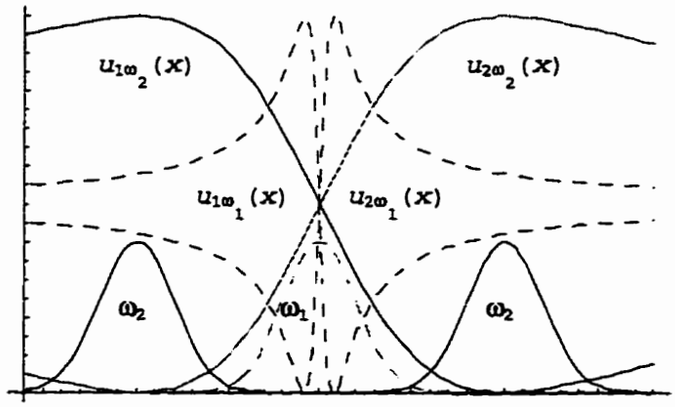

Figure 34: Good discriminatory performance using fuzzy cluster encoding with 2 centroids per class

Class-wise variants of the previously mentioned encoding techniques will now be described. The fuzzy interquartile encoding technique described in section 6.1.1 can be extended to deal with class information by computing the feature quartiles for each ω. So, ignoring the coordinate index j, (54)–(57) may be rewritten as

$$f_{1,k}^{j}(x_j) = 1 \wedge \left[0 \vee \left[1 - \left|x - 0.5(\alpha_k + Q_{L,k})\right| / (Q_{L,k} - \alpha_k)\right]\right] \qquad (71)$$

$$f_{2,k}^{j}(x_j)=1\wedge\left[0\vee\left[1-\left|x-0.5(Q_{L,k}+m_k)\right|/(m_k-Q_{L,k})\right]\right] \quad (72)$$

$$f_{3,k}^{j}(x_j)=1\wedge\left[0\vee\left[1-\left|x-0.5(m_k+Q_{U,k})\right|/(Q_{U,k}-m_k)\right]\right] \quad (73)$$

$$f_{4,k}^{j}(x_j)=1\wedge\left[0\vee\left[1-\left|x-0.5(Q_{U,k}+\beta_k)\right|/(\beta_k-Q_{U,k})\right]\right] \quad (74)$$

where j is the feature coordinate for an ω_k individual and α^j_k, $Q^j_{L,k}$, m^j_k, $Q^j_{U,k}$, and β^j_k are the smallest value, lower quartile, median, upper quartile, and largest value of feature j for the N_k individuals of ω_k.

The class-wise extension to the dimension-preserving fuzzy interquartile (Section 6.1.2) encoding involves constructing a single piece-wise fuzzy set for each ω. As in the previous paragraph, α^j, Q^j_L, m^j, Q^j_U, and β^j must be computed for each class (Figure 35). Extending (62), and ignoring the feature index, the membership function for feature j of ω_k is

$$f_k^j(x)=\begin{cases} h_k\dfrac{x-\alpha_k}{Q_{L,k}-\alpha_k} & \text{if} \quad \alpha_k\le x<Q_{L,k} \\ (1-h_k)\dfrac{x-Q_{L,k}}{m_k-Q_{L,k}}+h_k & \text{if} \quad Q_{L,k}\le x<m_k \\ (h_k-1)\dfrac{x-m_k}{Q_{U,k}-m_k}+1 & \text{if} \quad m_k\le x<Q_{U,k} \\ -h_k\dfrac{x-Q_{U,k}}{\beta_k-Q_{U,k}}+h_k & \text{if} \quad Q_{U,k}\le x<\beta_k \\ 0 & \text{if} \quad x<\alpha_k\vee x>\beta_k \end{cases} \quad (75)$$

For the degenerate cases, the extension to (63) is

$$f_k^j(x)=\begin{cases} 1 & \text{if} \quad \alpha_k<Q_{L,k}=m_k\vee\beta_k>Q_{U,k}=m_k \\ h_k & \text{if} \quad m_k>Q_{L,k}=\alpha_k\vee m_k<Q_{U,k}=\beta_k \end{cases} \quad (76)$$

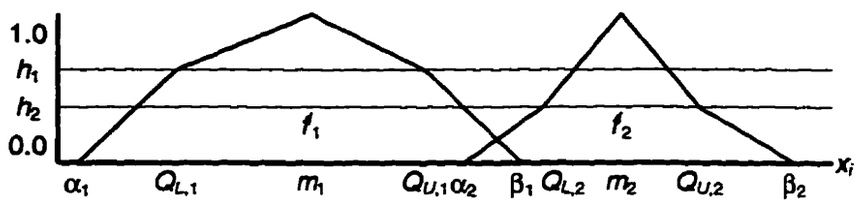

Figure 35: Feature membership functions constructed for ω_1 and ω_2

The class-wise extension to fuzzy cluster encoding involves employing the fuzzy c-means algorithm for each ω instead of once for the entire data set. Let $P_w=\{u_{1w}, u_{2w}, \ldots, u_{cw}\}$ be the pseudopartition for ω_w such that

$$\begin{aligned} \sum_{i=1}^{c} u_{iw}(x_k) &= 1 \qquad \forall k = \{1 \ldots N_w\} \\ 0 < \sum_{k=1}^{N_w} u_{iw}(x_k) &< N_w \qquad \forall i = \{1 \ldots c\} \end{aligned} \tag{77}$$

and the associated cluster centroids are

$$v_{iw} = \frac{\sum_{k=1}^{N_w} [u_{iw}(x_k)]^{m_w} x_k}{\sum_{k=1}^{N_w} [u_{iw}(x_k)]^{m_w}} \tag{78}$$

where v_{iw} (i=1, 2, ..., c) is the centroid associated with the partition u_{iw} and $m_w \in (1,\infty)$ governs the influence of membership grades for ω_w individuals. The performance index (66) is used on a class-by-class basis. The algorithm described in section 6.1.3 is now applied to each ω. The membership function (70) is used for each set of ω cluster centroids v_{iw}

$$u_{iw}(x_k) = \begin{cases} \left[\sum_{j=1}^{c} \frac{\|x_k - v_{iw}\|^m}{\|x_k - v_{jw}\|^m} \right]^{-1} & \text{if} \quad x_k \neq v_{jw} \\ 1 & \text{if} \quad x_k = v_{jw \neq iw} \end{cases} \tag{79}$$

6.1.5 Fuzzy encoded multi-layer perceptron

Fuzzy encoding may be integrated into a MLP classifier. The basic ANN structure of a fuzzy encoded multi-layer perceptron (FE-MLP) is described in section 4.2. Specifically, the global error function (28) is used. The learning rate, α, may be in the range [0,1], but in all of the experiments described in the next chapters $\alpha \in [0.7,0.9]$. No momentum term is ever employed. The logistic function (26) is used as the transfer function. A single hidden layer is used with the

number of hidden PEs determined experimentally (empirical evidence suggests using one or two more hidden PEs than the number of classes). The output layer has k PEs corresponding to the k classes. The FE-MLP assigns an input vector to ω_j if output PE j has the largest activation.

Recall in section 4.2.1 that the logistic function's range of [0,1] may cause a bias towards learning larger values and hence the bipolar hyperbolic tangent function is often used instead. However, the logistic function bias may be exploited within the FE-MLP. A large fuzzy encoded value indicates that the original value is similar to a "typical" value for that feature. For instance, with fuzzy interquartile encoding, one or two large fuzzy encoded values indicate that the original value was within the interquartile range for the feature whereas no large values indicate that the original value was an outlier. Therefore, the logistic function's natural bias will further diminish the impact of feature values that are far outside the interquartile range. This is similarly the case for the dimension-preserving fuzzy encoding as well as the fuzzy cluster encoding.

In the case of fuzzy interquartile encoding, the input layer will have pn PEs where n is the dimensionality of the original input space and p is the number of fuzzy sets used for the fuzzy encoding (more generally, different numbers of fuzzy sets may be used for each feature). Figure 36 shows the architecture of a FE-MLP using four fuzzy sets constructed around the quartiles and two original input variables, x_1 and x_2.

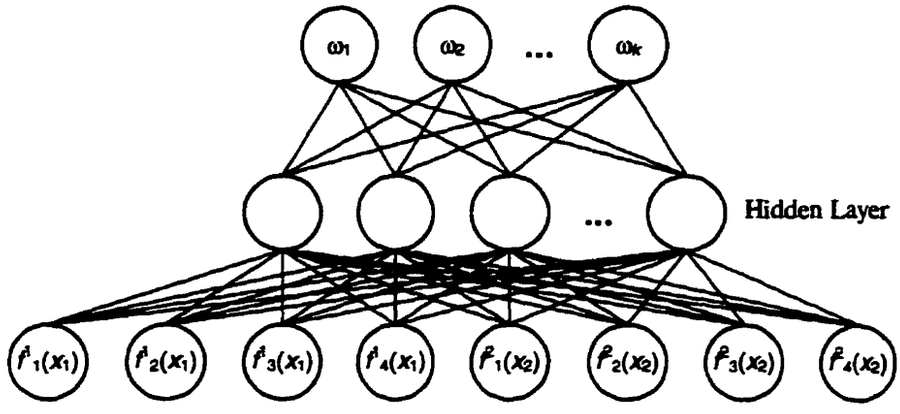

Figure 36: Fuzzy interquartile FE-MLP with two inputs, x_1 and x_2, four fuzzy sets, and k classes

At first glance, it may appear that fuzzy encoding would significantly increase the complexity of the design phase of the classification process since it is increasing the dimensionality of the feature space (in the previous example by a factor of four). The experimental results in the next two chapters consistently demonstrate not only does fuzzy encoding not increase the complexity of the design phase but, in fact, it actually dramatically reduces the number of iterations required by MLPs to converge during the design phase. This reduction is, at times, greater than an order of magnitude while, at the same time, improving the classification accuracy for the test phase. Several factors contribute to this efficiency. First, as previously mentioned, fuzzy encoding naturally scales the data to the unit interval and scaled data improves the performance of an MLP (and neural networks, in general) by reducing the impact of variance disparities across the features [96]. Second, many of the fuzzy encoded values are zero (or near zero) and, the corresponding terms in the PE summations are zero (or near zero) regardless of the respective weights. Ultimately, this means that they contribute little to the overall error of the FE-MLP so resultant errors propagated back through the network are not caused (to any great extent) by these values. Finally, outliers can be problematic to a standard MLP since they can cause large resultant errors and many more iterations will typically be required for the classifier to converge. Fuzzy encoding reduces the impact of outlying feature values and hence improves the convergence time.

In the case of the class-wise extension to fuzzy interquartile encoding, the input layer will have kpn PEs where n is the dimensionality of the original input space, p is the number of fuzzy sets used for the fuzzy encoding, and k is the number of classes. Figure 37 shows the architecture of a FE-MLP using four fuzzy sets constructed around the quartiles and two original input variables, x_1 and x_2 for a two class problem.

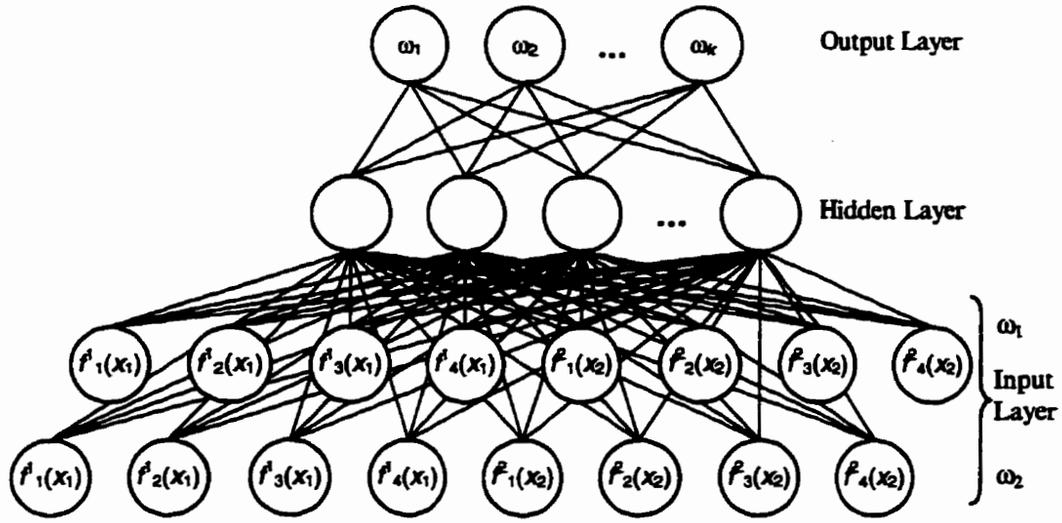

Figure 37: A FE-MLP using a class-wise extension to fuzzy interquartile encoding

FE-MLPs employing dimension-preserving and class-wise dimension-preserving fuzzy encoding have more straightforward architectures, with n input layer PEs, than those using the fuzzy interquartile encoding counterparts (Figure 38 and Figure 39, respectively).

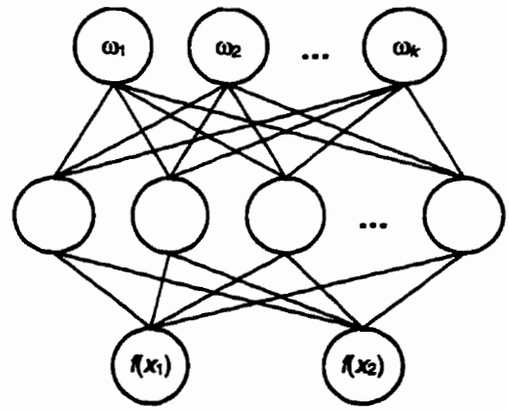

Figure 38: Dimension-preserving FE-MLP with two inputs, x_1 and x_2, and k classes

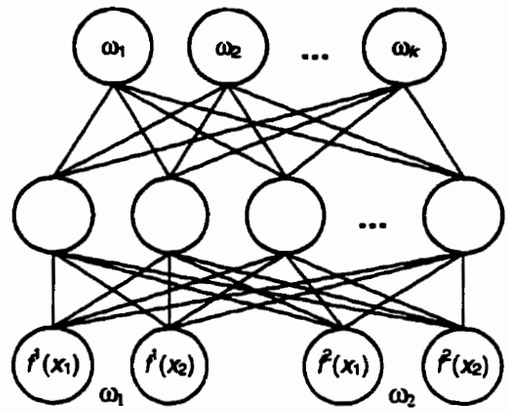

Figure 39: Class-wise dimension-preserving FE-MLP

FE-MLPs employing fuzzy cluster encoding will have a simpler architecture than its non-encoded MLP counterpart when the number of dimensions of the feature space, n, is greater than the number of clusters, c, used in the fuzzy c-means algorithm. In general, when the MLP counterpart has n input layer PEs, the FE-MLP will have c input layer PEs. Figure 40 is an example of a FE-MLP architecture for a 10-dimensional, 2 class, feature space using 2 cluster centroids.

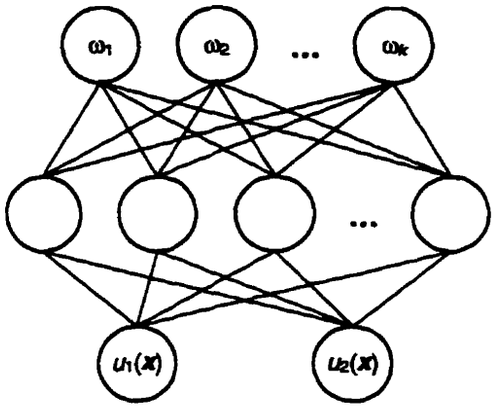

Figure 40: FE-MLP employing fuzzy cluster encoding with input vectors $x=[x_1, x_2, \ldots, x_n]$

The class-wise extension to the FE-MLP employing fuzzy encoding will have kc input layer PEs instead of n (Figure 41 is an example of a 10-dimensional, 2 class, feature space with 2 cluster centroids per class). This architecture can be further generalized by having different numbers of cluster centroids for each class.

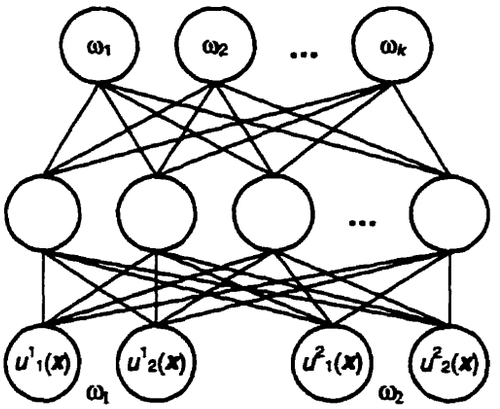

Figure 41: FE-MLP employing class-wise fuzzy cluster encoding with input vectors $x=[x_1, x_2, \ldots, x_n]$

6.2 Burnishing Tarnished Gold Standards

Data, such as magnetic resonance spectra, are often difficult to analyze due to their complex nature and the presence of noise. Many preprocessing methods exist that transform the original input data in order to eliminate or diminish the effects of noise and/or reduce the dimensionality of the input space. Unfortunately, culling diagnostic information is further exasperated by the fact that the reference test or gold standard, against which a new and possibly imperfect diagnostic test is measured, may itself be imprecise or even unreliable. However, little work has been done to investigate a methodology whereby the possible imprecision of a well-established but tarnished gold standard may be addressed while at the same time maintaining its vital discriminatory power.

Two strategies are discussed to burnish such tarnished gold standards. The first uses a robust estimation of deviations from class medoids (the robust equivalent of a centroid) for the reclassification of spectra in a design set. The second uses a fuzzy set theoretic preprocessing method to enhance the gold standard by incorporating non-subjective within-class medoid information. Either strategy may be used to augment any of the fuzzy encoding approaches described in section 6.1.1.

6.2.1 Robust reclassification

This preprocessing strategy involves the robust reclassification (RR) of vectors in a design set using a robust estimation of deviations from class medians. The median of the absolute deviations (MAD)

$$\tau(x) = \frac{median|x - median(x)|}{0.6745} \qquad (80)$$

is a robust estimator of the standard deviation (the constant is used so that as the error distribution becomes more normal the MAD estimate converges to the standard deviation) [50]. Only 40%

efficient for normal data [98], it is robust to outliers and long-tailed distributions, nevertheless [103].

Although a univariate estimator, it may be extended to the multivariate case by computing a vector, τ_l, whose elements are dispersion measures for each feature of ω_l vectors. Specifically, $\forall x_j \in \omega_l$ (j=1, 2, ..., n_l), feature i of τ_l is

$$\tau_{li} = \frac{median\left|x_{ji} - median(x_{ji})\right|}{0.6745} \qquad (81)$$

A feature of x_j is considered to be an ω_l feature i outlier if

$$\left|x_{ji} - median_{li}\right| > c\tau_{li} \qquad (82)$$

where $median_{li}$ is element i of the ω_l medoid. The constant, $c \geq 1$, is the spread across the median indicating whether or not a feature is an outlier. Specifically, it is a robust version of the empirical corollary of Tchebysheff's theorem. Tchebysheff's theorem states that for $c \geq 1$, at least $(1-1/c^2)$ of a set of N measurements will lie within c standard deviations of their mean [78]. The empirical corollary states that, if a data set has a normal distribution, then the following heuristics may be used to describe the data set: approximately 68% of the measurements will lie within c=1 standard deviation of their mean; approximately 95% of the measurements will lie within c=2 standard deviations of their mean; and, almost all the measurements will lie within c=3 standard deviations of their mean. For the robust case, the standard deviation is replaced by (80) and the mean is replaced by the median. For the remainder of the thesis, c=2.5, hence, approximately 99% of the measurements will lie within 2.5 MADs of their median. Finally, for each vector, x_j, compute its membership in each class medoid using

$$D_l^{(j)} = \sum_i d_{ji}^{(l)} \qquad (83)$$

where

$$d_{ji}^{(l)}(x_j)=\begin{cases}0 & if \quad \left|x_{ji}-median_{li}\right|>2.5\tau_{li}\\ \left(1+\left|x_{ji}-median_{li}\right|\right)^{-1} & otherwise\end{cases} \qquad (84)$$

A vector, x_j, from ω_l will be reassigned to ω_p, if $D_l^{(j)}<D_p^{(j)}$. In other words, a vector must be sufficiently distant from its class' medoid and sufficiently near another class' medoid. Note that reclassification may only occur for vectors in the design set.

6.2.2 Fuzzy gold standard adjustment

Given an input vector $x_i=[x_1, x_2, \ldots, x_n]\in\omega_l$, the associated gold standard may be encoded using output vectors of the form, $y_i=[y_1, y_2, \ldots, y_k]$ where

$$y_j=\begin{cases}1 if \ j=l\\ 0 if \ j\neq l\end{cases} \qquad (85)$$

The weighted distance, d_{il}, of x_i from the ω_l medoid is defined as

$$d_{il}=\sum_j\left|\frac{x_{ij}-m_{lj}}{\tau_{lj}}\right| \qquad (86)$$

where m_l and τ_l are the respective feature-wise median and MAD of the ω_l vectors. This distance measure is then incorporated into the gold standard using membership functions (see section 2.3) — monotonic functions that are continuous in the interval [0,1] indicating the degree to which an element belongs to a set. The membership function for ω_l is defined as

$$f_l(x_i)=\left[1+(d_{il}/q)^p\right]^{-1} \qquad (87)$$

where $p>1$ and $q>0$ describe the shape and amount of fuzziness for the membership function. Figure 42 plots $f(x)$ for different values of p with a constant q. Note that $f(x)$ is sigmoidal and that as p increases, f approaches a step function. The crossover point, which occurs when the membership function is ½, occurs when the distance equals q.

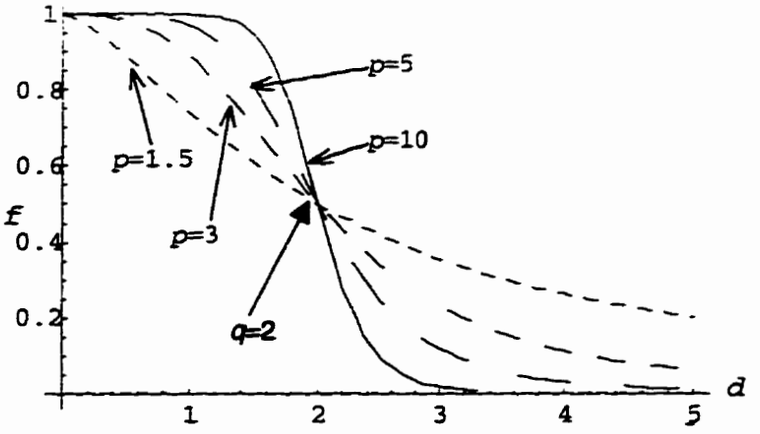

Figure 42: Plot of $f(x)$ with varying p (q=2)

Figure 43 plots $f(x)$ for different values of q with a constant p. As q increases, f becomes fuzzier; that is, membership values will remain high even at great distances.

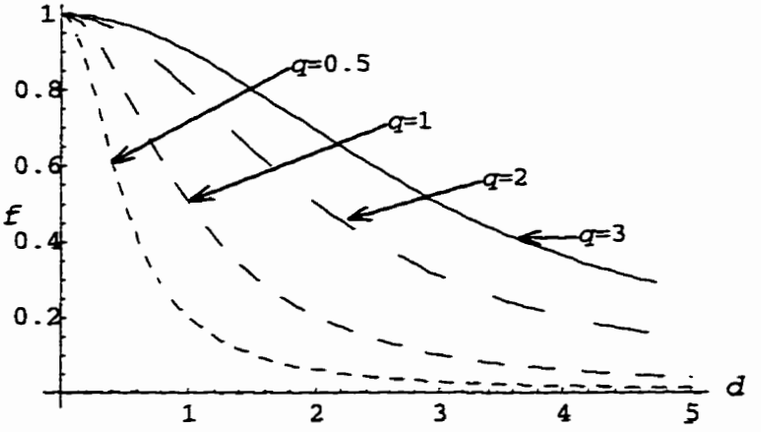

Figure 43: Plot of $f(x)$ with varying q (p=2)

In general, the further a vector is from a class medoid, the lower its membership value for that class. It is possible for a vector, x, that was originally assigned to ω_k to be closer to the medoid of another class, ω_t. In such cases, $f_k(x)<f_t(x)$ and, hence, the original gold standard assignment will no longer predominate. To rectify this situation, let $f_k(x)=f_t(x)$. That is, a vector will never be reassigned to a class different from the class to which it was originally assigned. However, if a vector is near another class medoid then the corresponding output element for that vector will not be zero. The fuzzy gold standard adjustment (FA) may now be encoded by the vector $y_i'=[y_1', y_2', \ldots, y_k']$ where

$$y_j = \begin{cases} 2f_j^2 & \text{if } 0 \le f_j \le 0.5 \\ 1-2(1-f_j)^2 & \text{if } 0.5 \le f_j \le 1.0 \end{cases} \quad (88)$$

This operation is known as contrast intensification [124] and here the intent is to increase values of f that are above 0.5 and reduce those that are below this point, in other words, contrast intensification has the effect of reducing the fuzziness of f.

6.2.3 Reclassification versus adjustment

The FST gold standard adjustment strategy described in section 6.2.2 is not as radical as the robust reclassification strategy discussed in section 6.2.1 where an individual may actually be reclassified in the design set if it is sufficiently distant from the class to which it was originally assigned and sufficiently near another class' medoid. A conservative variant of the robust reclassification strategy may be defined that mirrors the intent of the FST gold standard adjustment. Similarly, a radical variant of the latter may be defined that mirrors the radical nature of the robust reclassification strategy.

For each vector, $\boldsymbol{x}_j$, the robust reclassification variant produces the output vector, an augmented gold standard, $\boldsymbol{y}'=[D_1^{(j)}, D_2^{(j)}, \ldots, D_k^{(j)}]$, where $D_l^{(j)}$ are defined using (83) and (84). Since $D_l^{(j)} \in [0,1]$, and it approaches 1 as the vector approaches the ω_l medoid, this new gold standard mimics the behaviour of FA. It is still possible, however, for a vector assigned to some class by the original gold standard, to have a distance value that is greater for some other class. Hence, although more conservative than the original strategy, this variant may still reclassify the vector. To ensure that the original gold standard predominates, its corresponding distance value may be set to the maximum of all distance values for the input vector.

The radical variant of FA is straightforward: for an input vector's enhanced gold standard encoded by the vector $\boldsymbol{y}_i'=[y_1', y_2', \ldots, y_k']$ defined by (88), reassign it to ω_l where y_l' is maximum.

Finally, most classifiers, such as ANNs, accommodate class labels that have been encoded as 1–of–k output vectors. Hence, these classifiers may also accommodate output vectors of the form $[0,1]^k$. However, some classifiers, such as linear discriminant analysis, admit only discrete integer values from 1 to k. For this latter case, only RR or the FA variant may be used to burnish tarnished gold standards.

7 Experiments Using Synthetic Data

7.1 Two-Class 1-Dimensional Data Sets

In the experiments described in this section, 200 one-dimensional points were randomly generated using two different distributions. All points from the first distribution were assigned to ω_1 (N_1=100) with the remainder assigned to ω_2 (N_1=100). The design set was comprised of 50 ω_1 points and 50 ω_2 points (N_d=100) with the remaining points assigned to the test set (N_t=100). All performance results for both design and test sets are measured using the chance-corrected measure of agreement, κ (section 3.2.3). Linear discriminant analysis is used as the classifier for all experiments. Four fuzzy sets are used for the fuzzy interquartile encodings. The quartiles for the fuzzy interquartile encodings and the dimension-preserving encodings are computed using only the points in the design set. Similarly, the cluster centroids for the fuzzy cluster encodings are computed using only design points. The threshold for the dimension-preserving fuzzy encoding was set at ¾ for all experiments. Each subsection will contain two pairs of performance tables, one for the design set and one for the test set. One pair of tables contains performance results using different cluster centroids for the fuzzy cluster encodings. The other pair of tables contains κ results for: the non-encoded data (NE); fuzzy interquartile encoding (IQ); dimension-preserving fuzzy encoding (DP); and the best fuzzy cluster encoding (CL) results from the previously mentioned table; and the respective class-wise variants (IQc, DPc, and CLc). For each experiment, a plot is listed showing the underlying probability density functions for each class, the misclassified points (large grey points on the axis), and the underlying fuzzy sets (for IQ and DP) or membership functions (for CL) for the different encodings.

7.1.1 Normal distributions with equal variances

In this experiment, the ω_1 points were sampled from the normal distribution $N(0,1)$ and the ω_2 points were sampled from $N(3,1)$. This is an ideal data set in the sense that the two classes are normally distributed with equal variance, hence LDA will produce an optimal decision boundary

for the design set as is the case here (see Table 5). The only errors that should occur should be where the probability density functions of the two classes overlap (see Figure 44). Apart from DP, all methods produced comparable results for the design set.

N_d	NE		IQ		DP		CL (c=2)		IQc		DPc		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	47	3	47	3	18	32	47	3	47	3	47	3	44	6
ω_2	2	48	2	48	21	29	2	48	2	48	3	47	2	48
κ	0.90		0.90		-0.06		0.90		0.90		0.88		0.84	

Table 5: Design set results using normally distributed data

Note that CLc with two clusters produced poorer results than other CL methods (Table 6).

N_d	CL (c=2)		CL (c=3)		CLc (c=2)		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	47	3	46	4	50	0	44	6
ω_2	2	48	2	48	21	29	2	48
κ	0.90		0.88		0.58		0.84	

Table 6: Design set results for fuzzy cluster encoding using different cluster numbers

For each method, concomitant results were obtained using the test set (Table 7) with IQc and DPc producing slightly better results than NE.

N_t	NE		IQ		DP		CL (c=2)		IQc		DPc		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	45	5	45	5	15	35	45	5	45	5	45	5	35	15
ω_2	6	44	6	44	15	35	6	44	4	46	4	46	2	48
κ	0.78		0.78		0.00		0.78		0.82		0.82		0.66	

Table 7: Test set results using normally distributed data

N_t	CL (c=2)		CL (c=3)		CLc (c=2)		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	45	5	41	9	50	0	35	15
ω_2	6	44	3	47	27	23	2	48
κ	0.78		0.76		0.46		0.66	

Table 8: Test set results for fuzzy cluster encoding using different cluster numbers

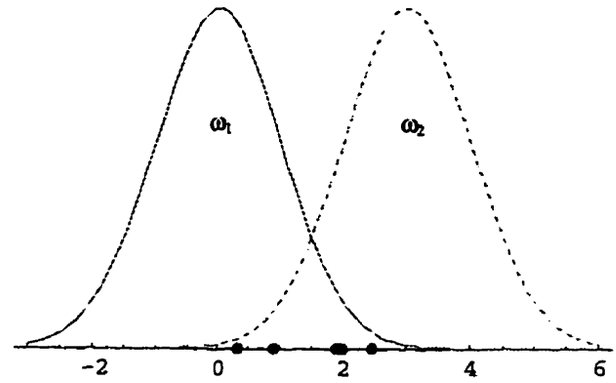

Figure 44: Non-encoded design set results

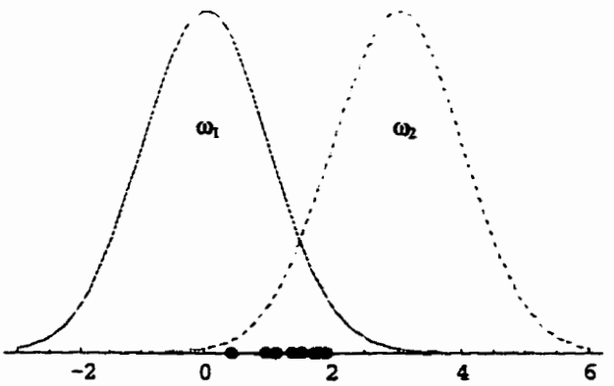

Figure 45: Non-encoded test set results

Figure 46 and Figure 47 show that IQ, like NE (Figure 44 and Figure 45), only misclassified some points that overlapped the pdfs of the two classes.

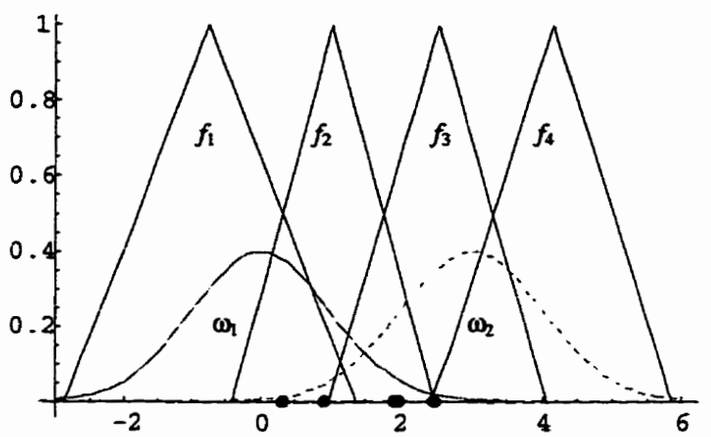

Figure 46: Fuzzy interquartile encoded design set results (α=-1.81, Q_l=0.31, m=1.76, Q_u=3.30, β=5.00)

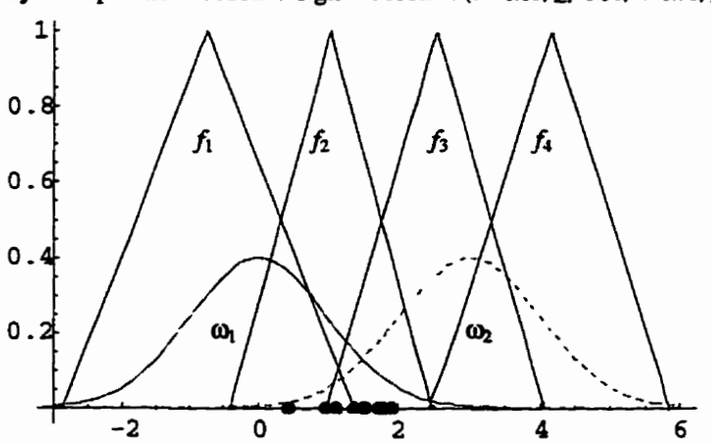

Figure 47: Fuzzy interquartile encoded test set results

Figure 48 and Figure 49 clearly demonstrate the problem with DP: not only does it suffer from misclassification of points at the overlap of the pdfs of the two classes, it also misclassifies many

points that are to the right of the overlap. This occurs because the encoded value for a point in this region will be nearly identical to the encoded value to the left of the overlap. Since ω_2 points predominate to the right and ω_1 predominate to the left this will wash away the discrimination between the two classes. This weakness is precisely the strength in the class-wise variant of this method (see below).

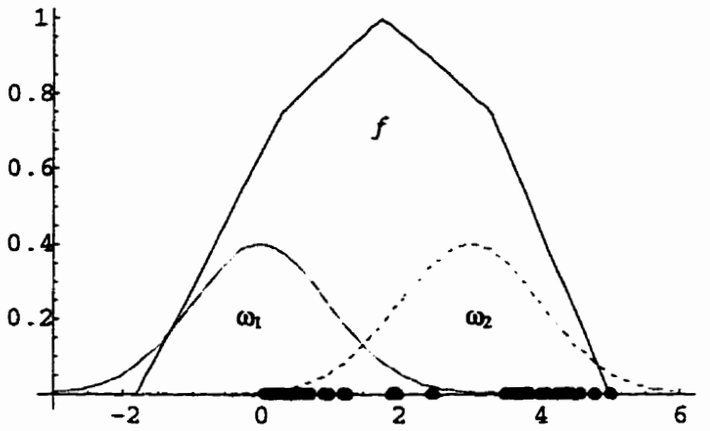

Figure 48: Dimension-preserving design set results (α=-1.81, Q_f=0.31, m=1.76, Q_e=3.30, β=5.00)

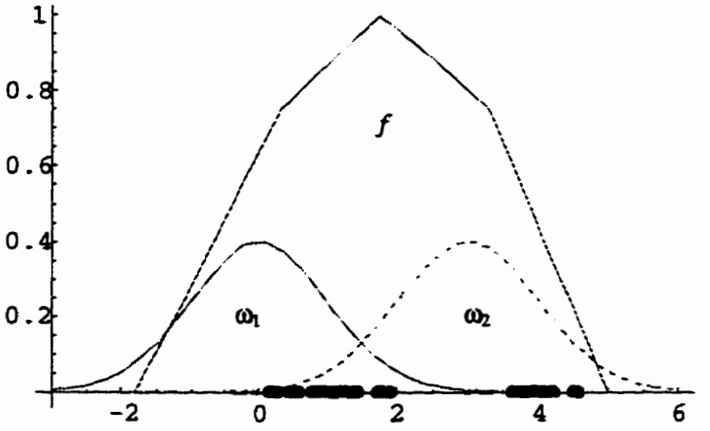

Figure 49: Dimension-preserving test set results

As with the NE and IQ methods, CL with two cluster centroids misclassified only points that fell between the class' pdfs (Figure 50 and Figure 51). Note that u_1 is maximum at the mode of ω_1 and u_2 is maximum at the mode of ω_2 and that the fuzzy c-means algorithm found centroids near these modes (0.16 and 3.30, respectively).

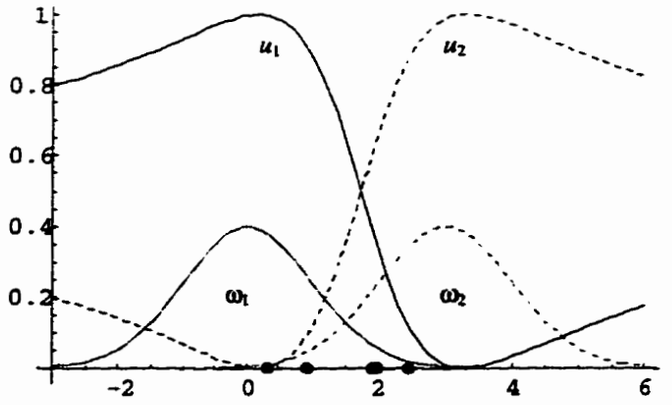

Figure 50: Fuzzy cluster (c=2) encoding using design set (v_1=0.16, v_2=3.30)

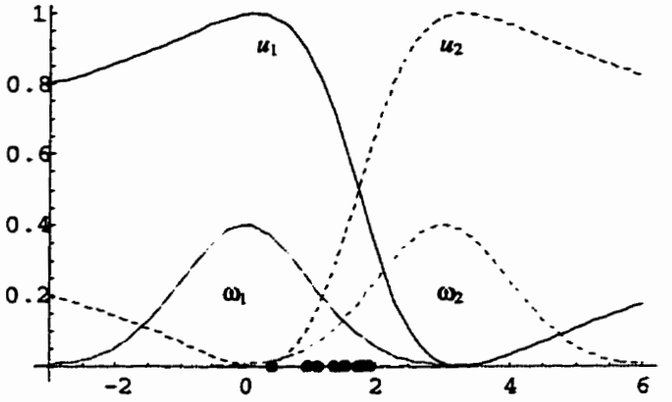

Figure 51: Fuzzy cluster (c=2) encoding using test set

Figure 52 and Figure 53 show CL using three cluster centres with results comparable to CL using only two clusters. However, while u_1 is maximum at the mode of ω_1 and the corresponding cluster centre (0.05) is near its mode the other centres (2.23 and 3.81) are to either side of the mode of ω_2 and, hence, u_2 and u_3 are not maximum at the mode. Nevertheless, since u_1 is near zero at this point, it does not confound this method.

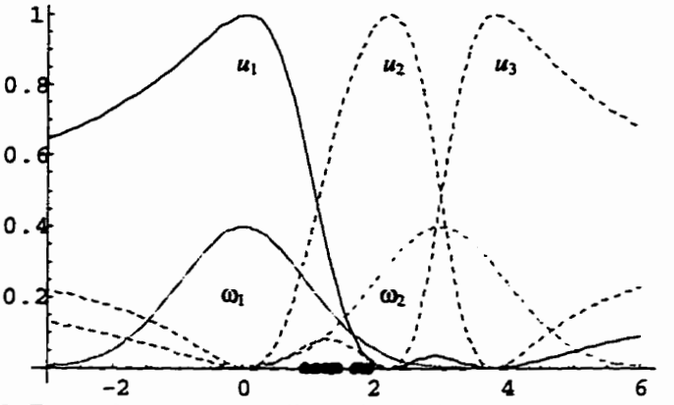

Figure 52: Fuzzy cluster (c=3) encoding using design set (v_1=0.05, v_2=2.23, v_3=3.81)

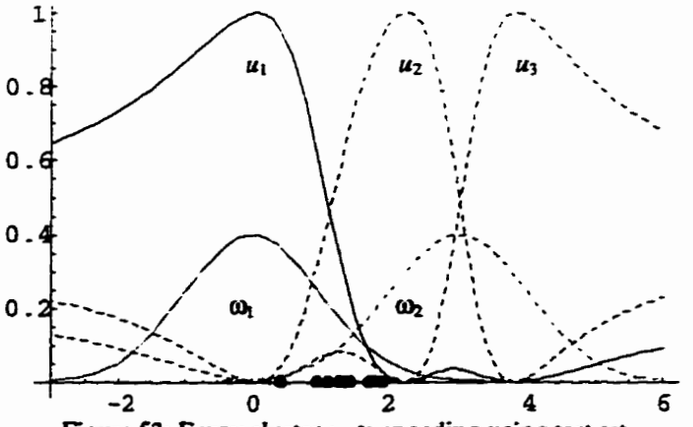

Figure 53: Fuzzy cluster (c=3) encoding using test set

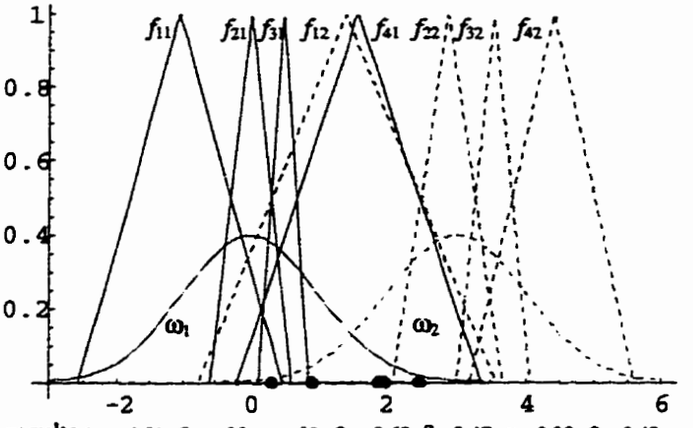

Figure 54: IQc design results (α_1=-1.81, Q_{l1}=-.28, m_1=.32, Q_{u1}=0.69, β_1=2.47, α_2=0.32, Q_{l2}=2.49, m_2=3.28, Q_{u2}=3.80, β_2=5.00)

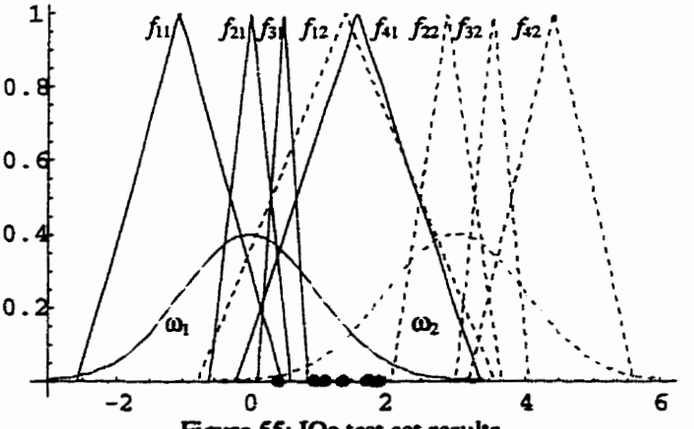

Figure 55: IQc test set results

A dramatic improvement occurs with DPc compared to DP (Figure 56 and Figure 57) since each membership function nearly uniquely encodes each class' points. Apart from the typical errors at the overlap of the pdfs, an additional two points were misclassified by this method. These points were the maximum and minimum points of ω_2, β_2=5.00 and α_2=0.32, respectively, which were both encoded as (0,0).

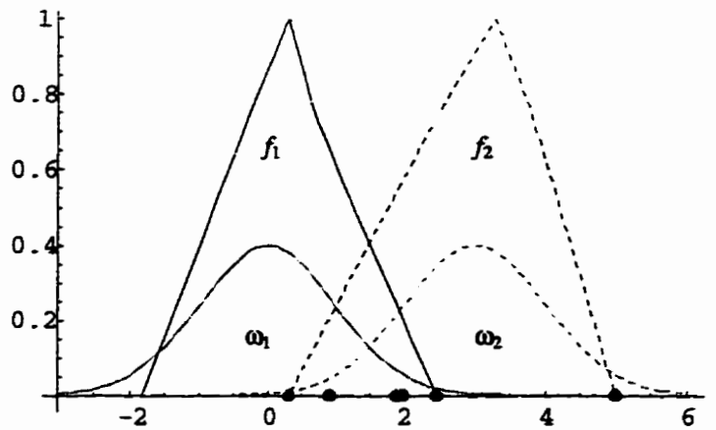

Figure 56: DPc design results (α_1=-1.81, Q_{l1}=-.28, m_1=.32, Q_{u1}=0.69, β_1=2.47, α_2=0.32, Q_{l2}=2.49, m_2=3.28, Q_{u2}=3.80, β_2=5.00)

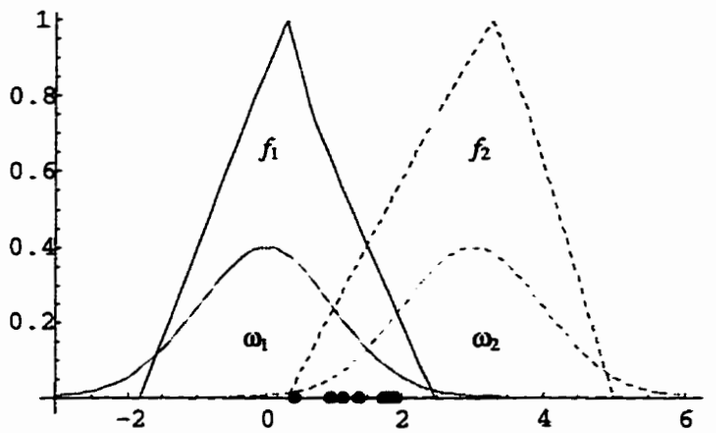

Figure 57: Class-wise dimension-preserving encoding using test set

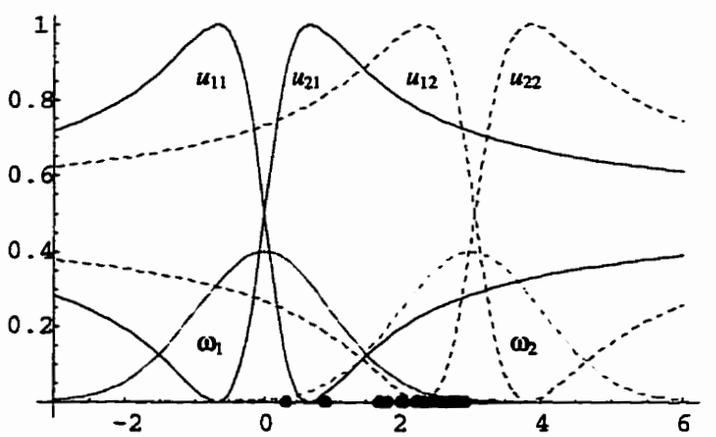

Figure 58: Class-wise fuzzy cluster (c=2) encoding using design set (v_{11}=-0.69, v_{21}=0.68, v_{12}=2.30, v_{22}=3.83)

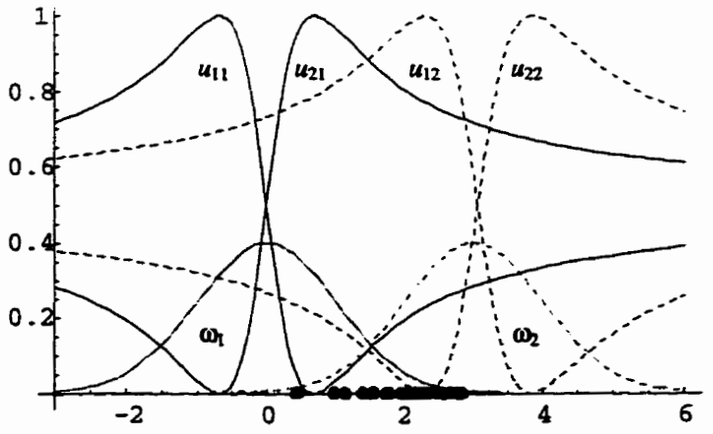

Figure 59: Class-wise fuzzy cluster (c=2) encoding using test set

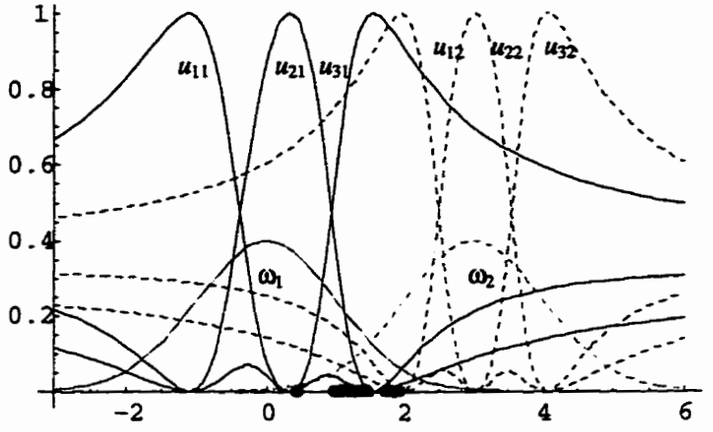

Figure 60: CLc (c=3) using design set (v_{11}=-1.11, v_{21}=0.33, v_{31}=1.53, v_{12}=1.97, v_{22}=3.04, v_{32}=4.07)

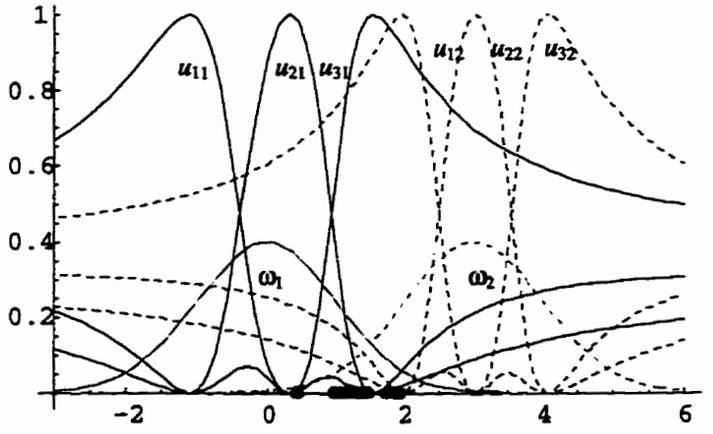

Figure 61: CLc (c=3) using test set

7.1.2 Normal distributions with unequal variances

In this experiment, the ω_1 points were sampled from the normal distribution $N(0,1)$ and the ω_2 points were sampled from $N(3,2)$. Note that apart from DP, all methods produced comparable results using the design set (Table 9 and Table 10).

N_d	NE		IQ		DP		CL (c=3)		IQc		DPc		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	45	5	42	8	35	15	41	9	45	5	38	12	42	8
ω_2	14	36	14	36	24	26	10	40	14	36	11	39	11	39
κ	0.62		0.56		0.22		0.62		0.62		0.54		0.62	

Table 9: Design set results using normal distributions with unequal variances

N_d	CL (c=2)		CL (c=3)		CLc (c=2)		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	46	4	41	9	49	1	42	8
ω_2	15	35	10	40	21	29	11	39
κ	0.62		0.62		0.56		0.62	

Table 10: Design set results for fuzzy cluster encoding using different cluster numbers

N_t	NE		IQ		DP		CL (c=3)		IQc		DPc		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	46	4	46	4	32	18	45	5	47	3	40	10	45	5
ω_2	7	43	7	43	26	24	5	45	12	38	7	43	5	45
κ	0.78		0.78		0.12		0.80		0.70		0.66		0.80	

Table 11: Test set results for each method using normally distributed data

N_t	CL (c=2)		CL (c=3)		CLc (c=2)		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	47	3	45	5	50	0	45	5
ω_2	13	37	5	45	16	34	5	45
κ	0.68		0.80		0.68		0.80	

Table 12: Test set results for fuzzy cluster encoding using different cluster numbers

As with the data in section 7.1.1, the points misclassified by NE occur where the pdfs for each class overlap. Since the variances are unequal, the overlap is greater, and the κ value decreases (Figure 62 and Figure 63). This also occurs with IQ (Figure 64 and Figure 65). Also note, with IQ, how f_4 has a wider span than the other membership functions to account for the greater variance in ω_2.

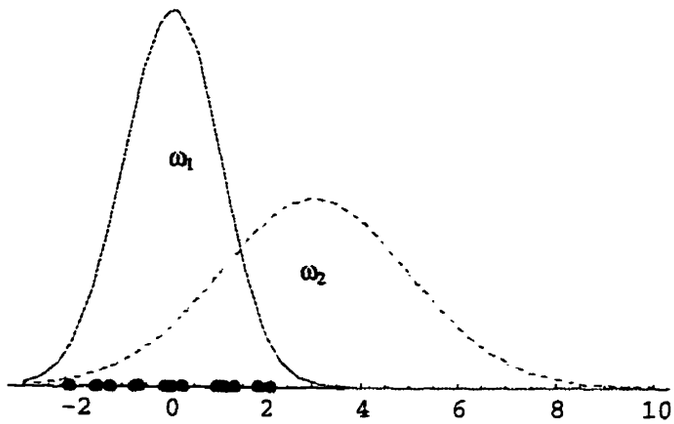

Figure 62: Non-encoded design set results

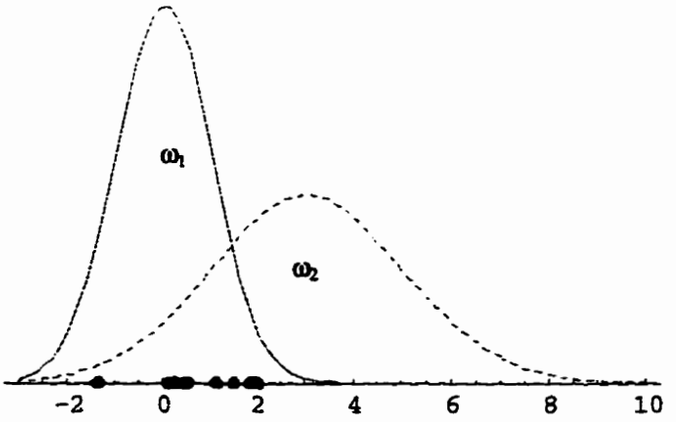

Figure 63: Non-encoded test set results

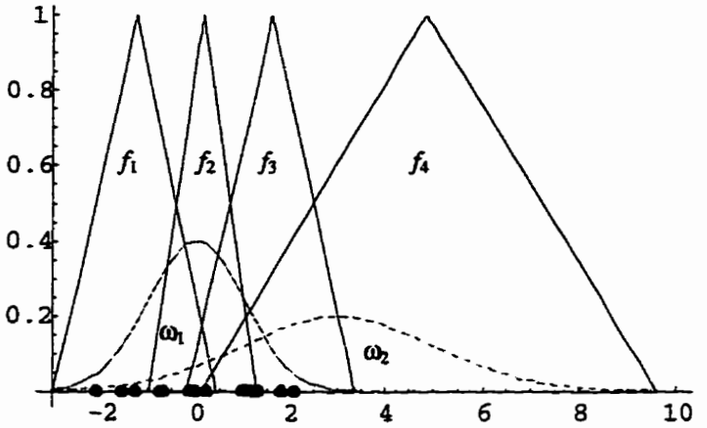

Figure 64: Fuzzy interquartile encoded design set results (α=-2.10, Q_1=-0.42, m=0.70, Q_u=2.48, β=7.23)

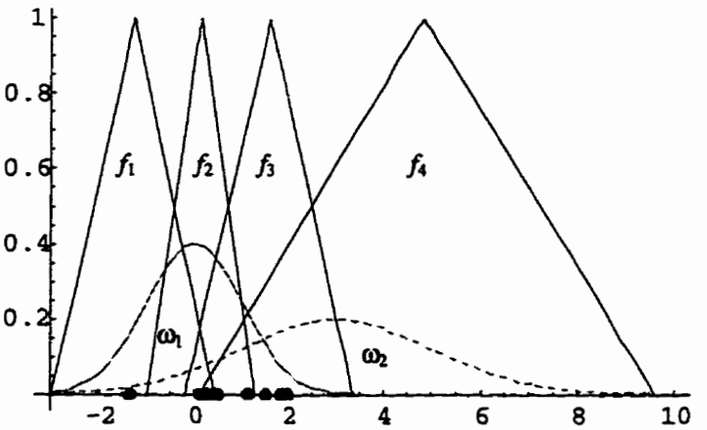

Figure 65: Fuzzy interquartile encoded test set results

Although faring slightly better than its counterpart in section 7.1.1, DP still underperformed compared to all other methods. The marginal improvement can be attributed to the skewing of the membership function caused by the unequal variances of the two classes.

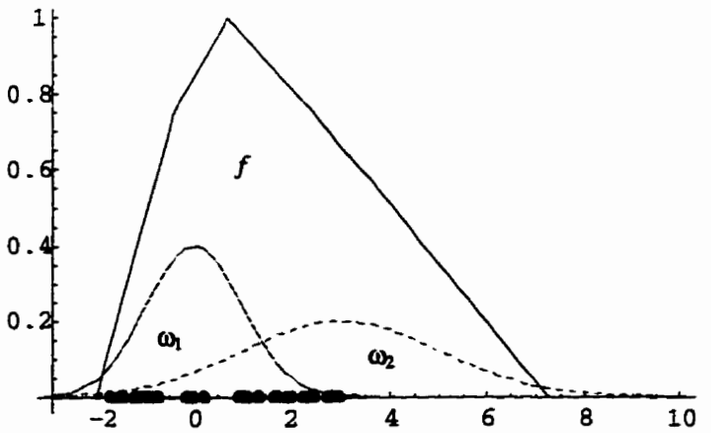

Figure 66: Dimension-preserving design set results (α=-2.10, Q_c=-0.42, m=0.70, Q_*=2.48, β=7.23)

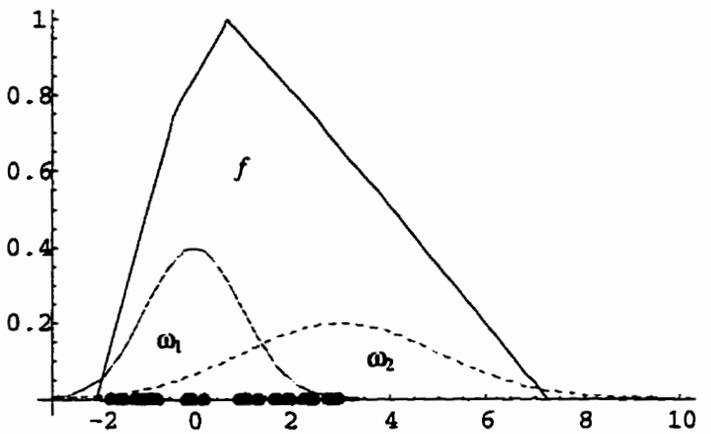

Figure 67: Dimension-preserving test set results

CL with two cluster centres produced poorer, but acceptable, results (Figure 68 and Figure 69). Part of this may be attributed to the location of the second cluster centre at 4.14 which is to the right of the ω_2 mode. This is due to the greater variance for the second class.

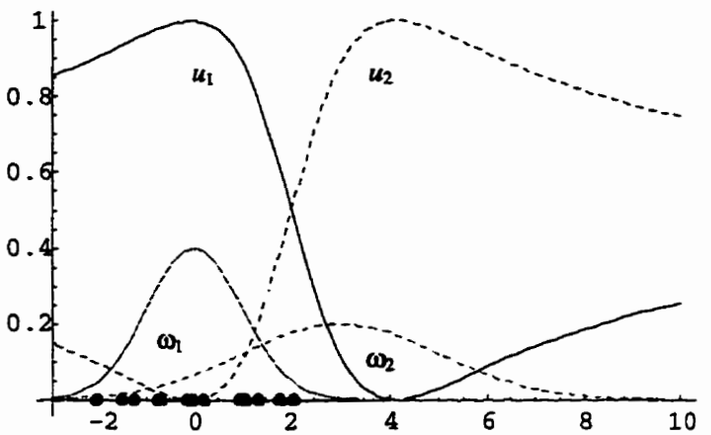

Figure 68: Fuzzy cluster (c=2) encoding using design set (v_1=-0.05, v_2=4.14)

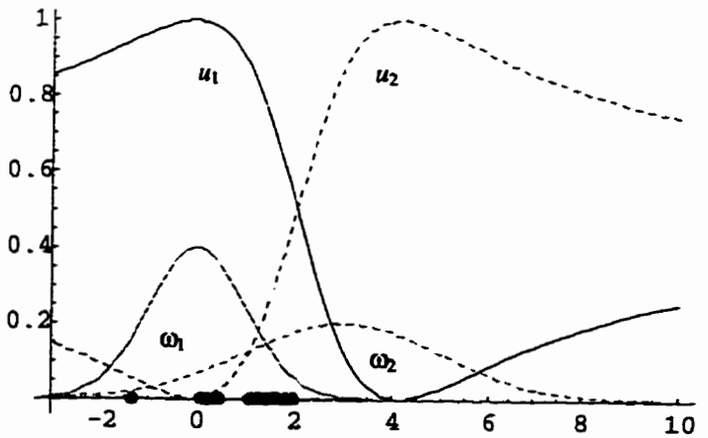

Figure 69: Fuzzy cluster (c=2) encoding using test set

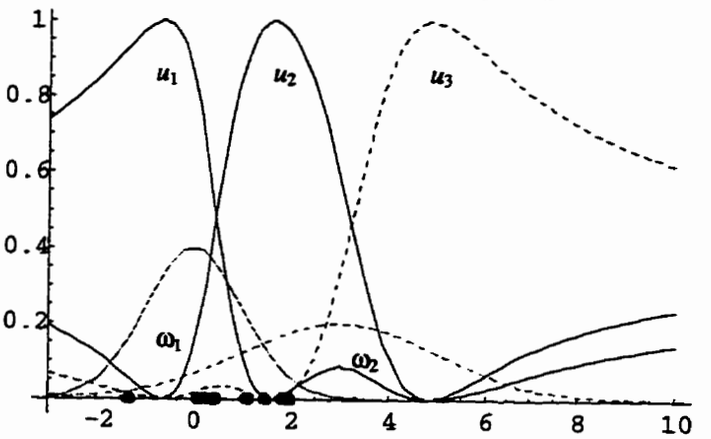

Figure 70: Fuzzy cluster (c=3) encoding using design set (v_1=-0.64, v_2=1.62, v_3=4.85)

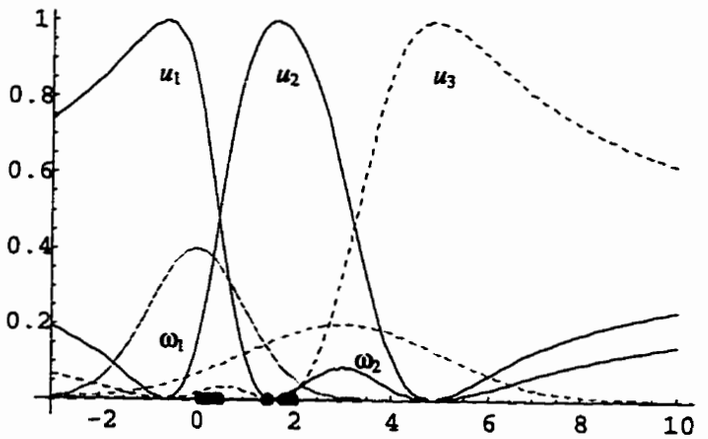

Figure 71: Fuzzy cluster (c=3) encoding using test set

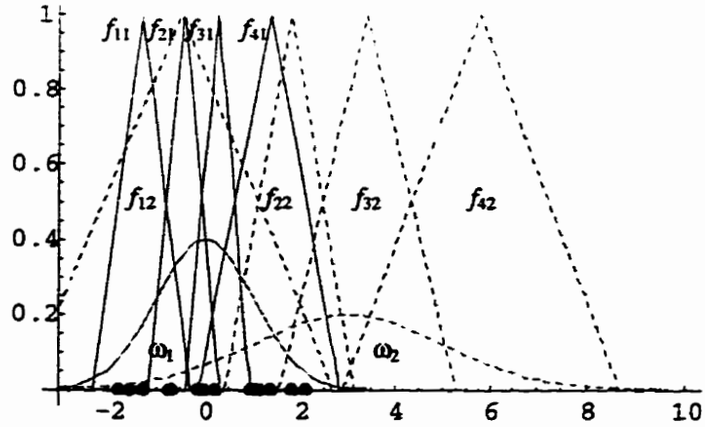

Figure 72: IQc design results (α_1=-1.80, Q_{l1}=-.80, m_1=-.06, Q_{s1}=.61, β_1=2.10, α_2=-2.10, Q_{l2}=1.10, m_2=2.47, Q_{s2}=4.35, β_2=7.23)

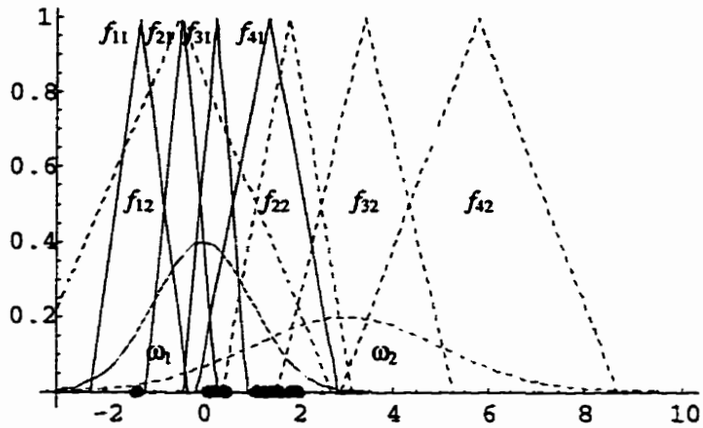

Figure 73: Class-wise fuzzy interquartile encoding using test set

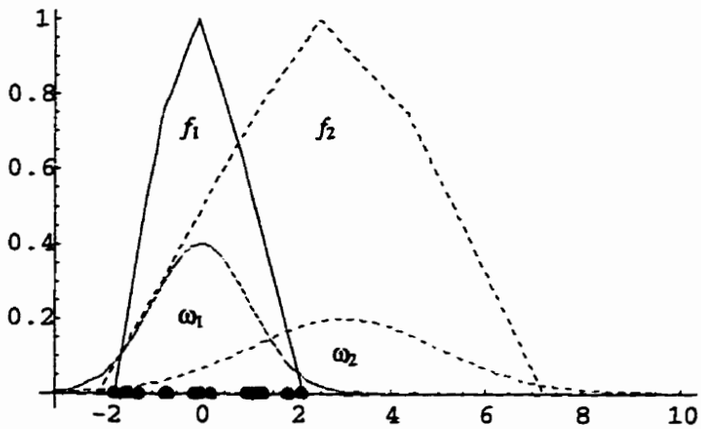

Figure 74: DPc design results (α_1=-1.80, Q_{l1}=-.80, m_1=-.06, Q_{s1}=.61, β_1=2.10, α_2=-2.10, Q_{l2}=1.10, m_2=2.47, Q_{s2}=4.35, β_2=7.23)

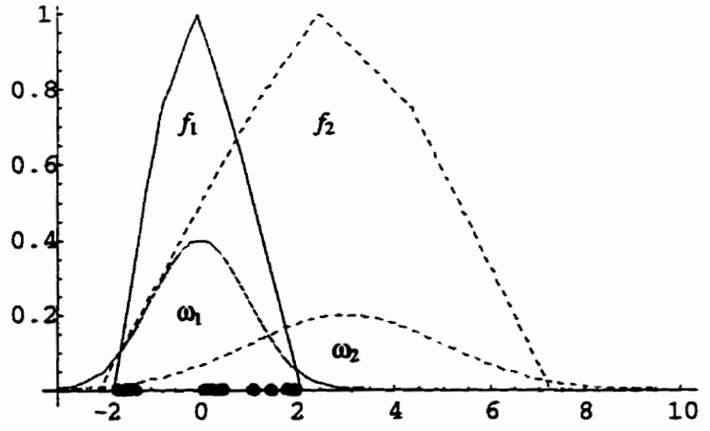

Figure 75: Class-wise dimension-preserving encoding using test set

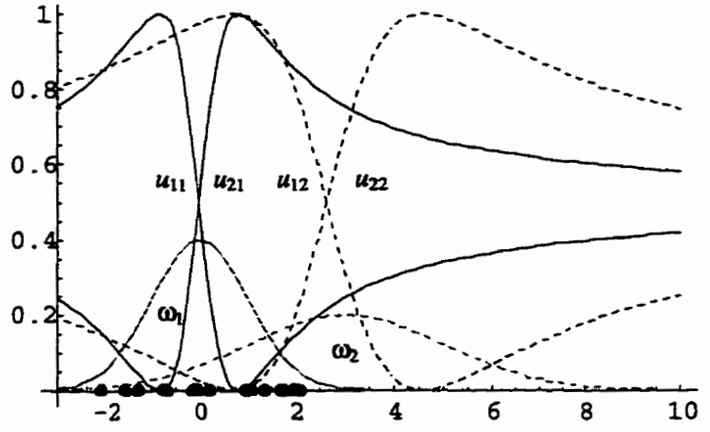

Figure 76: Class-wise fuzzy cluster (c=2) encoding using design set (v_{11}=-0.83, v_{21}=0.82, v_{12}=0.71, v_{22}=4.59)

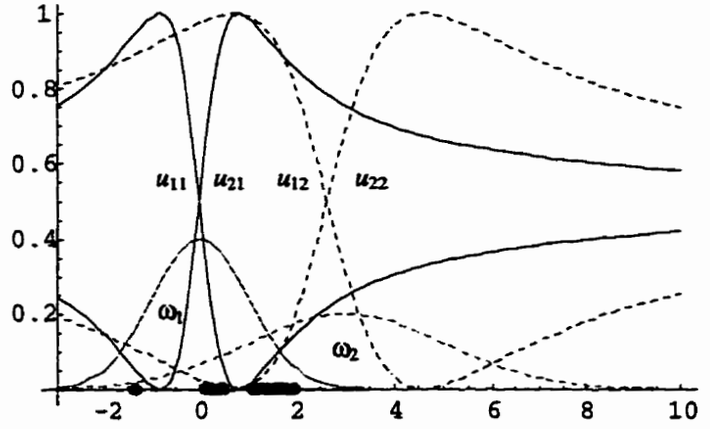

Figure 77: Class-wise fuzzy cluster (c=2) encoding using test set

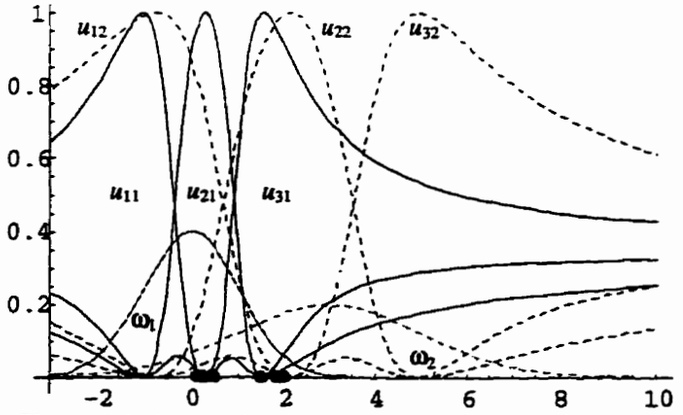

Figure 78: CLc (c=3) design results (v_{11}=-1.03, v_{21}=0.29, v_{31}=1.53, v_{12}=-0.76, v_{22}=2.12, v_{32}=4.94)

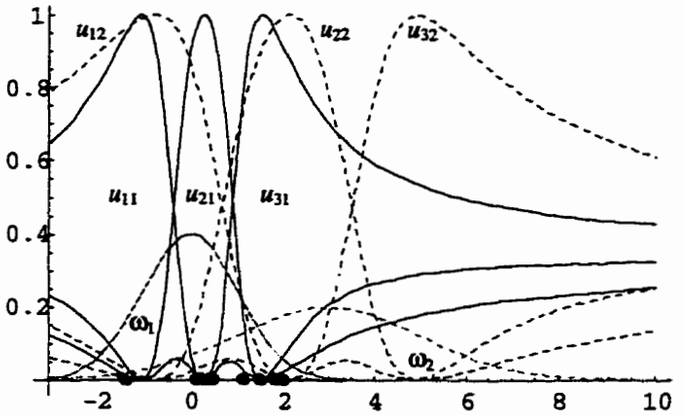

Figure 79: Class-wise fuzzy cluster (c=3) encoding using test set

7.1.3 Bimodal distribution

In this experiment, half of the ω_1 points were sampled from $N(0,1)$ and the other half were sampled from $N(5,1)$ while all of the ω_2 points were sampled from $N(10,1)$. The nature of this data set is such that NE cannot construct a discriminatory decision boundary. Since the ω_2 pdf is between the two modes of the ω_1 pdf a linear decision boundary can, at best, only produce classification results that misclassify about half of the points (Table 13). NE produced design results worse than chance and poor results using the test set (Table 14). Misclassified points occur with roughly equal frequency on either side of the ω_2 mode (Figure 80 and Figure 81).

N_d	NE		IQ		DP		CL (c=3)		IQc		DPc		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	25	25	50	0	44	6	50	0	50	0	50	0	37	13
ω_2	27	23	0	50	0	50	0	50	0	50	6	44	21	29
κ	-0.04		1.00		0.88		1.00		1.00		0.88		0.32	

Table 13: Design set results using a bimodal distribution

N_d	CL (c=2)		CL (c=3)		CLc (c=2)		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	25	25	50	0	25	25	37	13
ω_2	12	38	0	50	20	30	21	29
κ	0.26		1.00		0.10		0.32	

Table 14: Design set results for fuzzy cluster encoding using different cluster numbers

N_t	NE		IQ		DP		CL (c=3)		IQc		DPc		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	25	25	49	1	48	2	50	0	50	0	50	0	38	12
ω_2	22	28	0	50	0	50	0	50	6	44	9	41	12	38
κ	0.06		0.98		0.96		1.00		0.88		0.82		0.52	

Table 15: Test set results using a bimodal distribution

N_t	CL (c=2)		CL (c=3)		CLc (c=2)		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	25	25	50	0	26	24	38	12
ω_2	12	38	0	50	28	22	12	38
κ	0.26		1.00		-0.04		0.52	

Table 16: Test set results for fuzzy cluster encoding using different cluster numbers

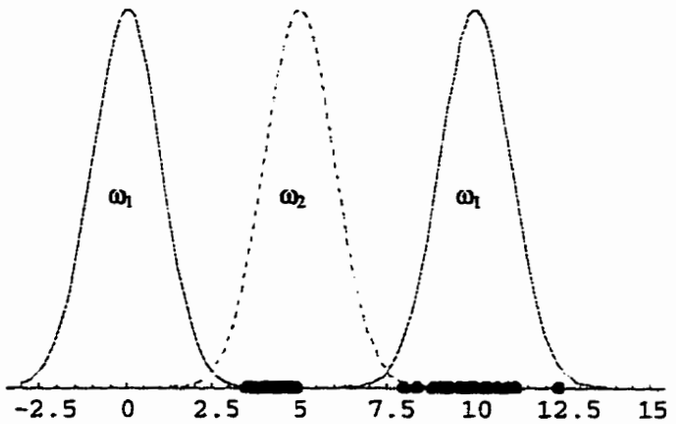

Figure 80: Non-encoded design set results

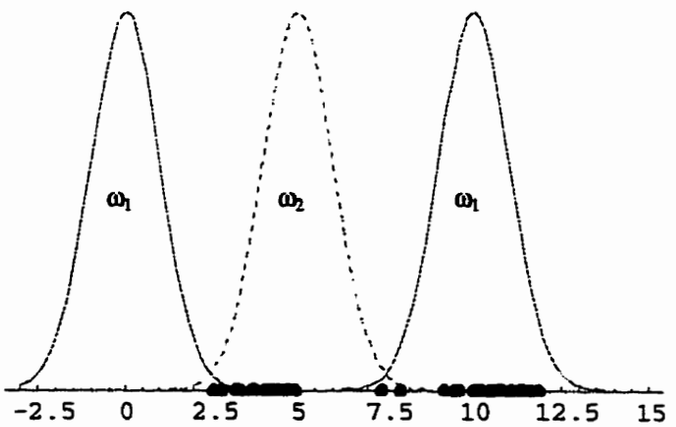

Figure 81: Non-encoded test set results

Figure 82 and Figure 83 show that IQ is immune to bimodally distributed data. The encoding is such that f_1 is at a maximum near the first mode of ω_1 and f_4 is at a maximum near the other mode of ω_1. However, f_2 and f_3 both approach ½ at the ω_2 mode.

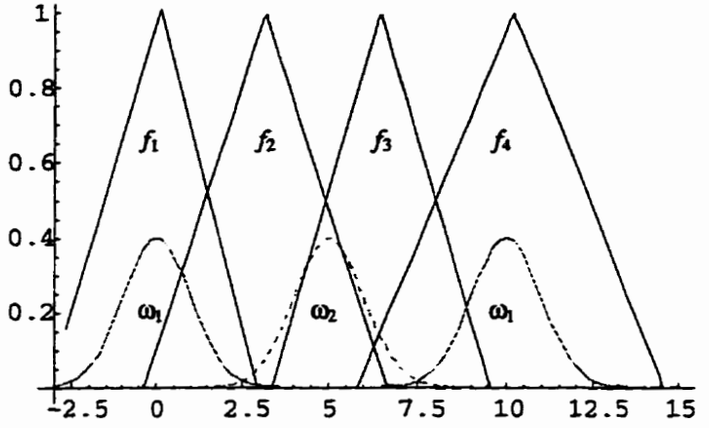

Figure 82: Fuzzy interquartile encoded design set results (α=-1.78, Q_f=1.40, m=4.91, Q_u=8.00, β=12.37)

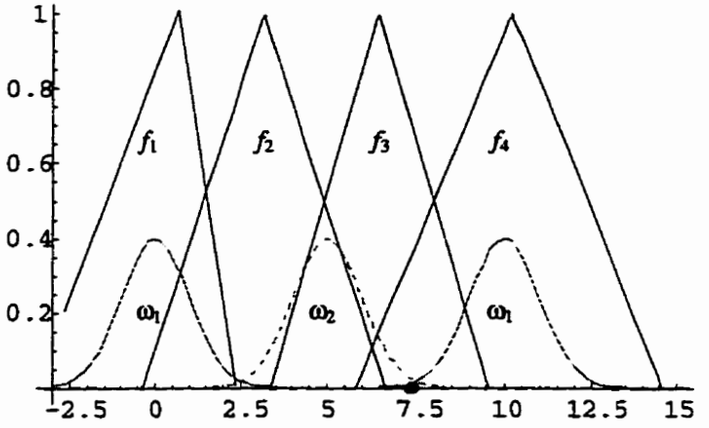

Figure 83: Fuzzy interquartile encoded test set results

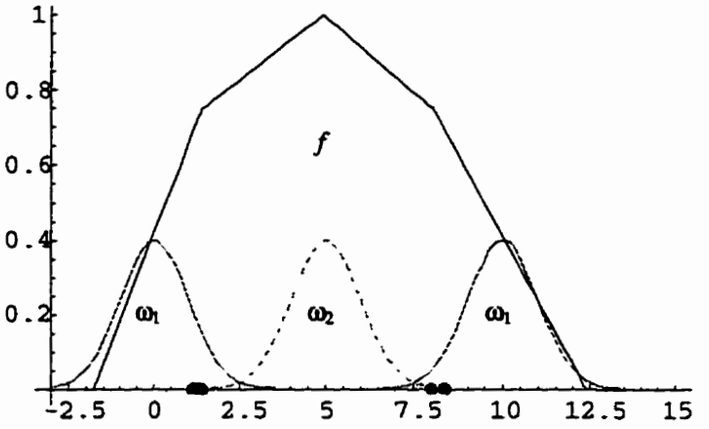

Figure 84: Dimension-preserving design set results (α=-1.78, Q_f=1.40, m=4.91, Q_u=8.00, β=12.37)

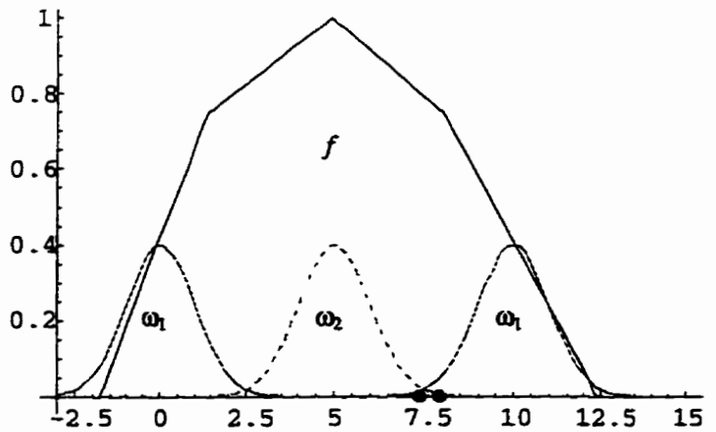

Figure 85: Dimension-preserving test set results

Figure 86 and Figure 87 demonstrate a potential problem with CL; using fewer cluster centres than numbers of modes. In this case, only two cluster centres were used while the data had three modes; hence, the cluster centres (0.60 and 7.19) were not near any of the modes (0, 5, or 10). In fact, the centres are situated near the overlaps of the pdfs about ω_2.

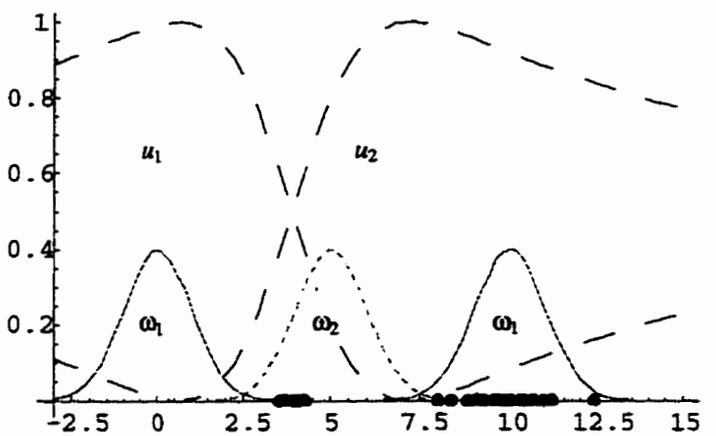

Figure 86: Fuzzy cluster (c=2) encoding using design set (v_1=0.60, v_2=7.19)

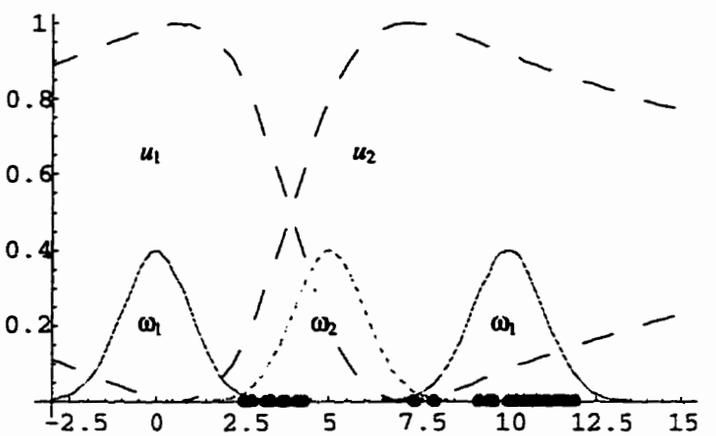

Figure 87: Fuzzy cluster (c=2) encoding using test set

Conversely, when 3 clusters are used (-0.1, 5.0, and 9.8) they are situated near each of the 3 modes. Hence, u_1 is at a maximum near the first mode of ω_1, u_3 is at a maximum near the other ω_1 mode, and u_2 is at a maximum near the ω_2 mode (Figure 88 and Figure 89). This was the only test set case where there was perfect agreement between the actual and desired outcomes.

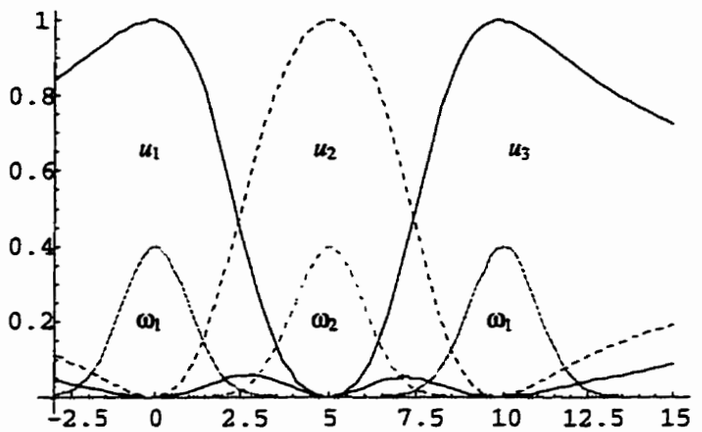

Figure 88: Fuzzy cluster (c=3) encoding using design set (v_1=-0.09, v_2=4.98, v_3=9.83)

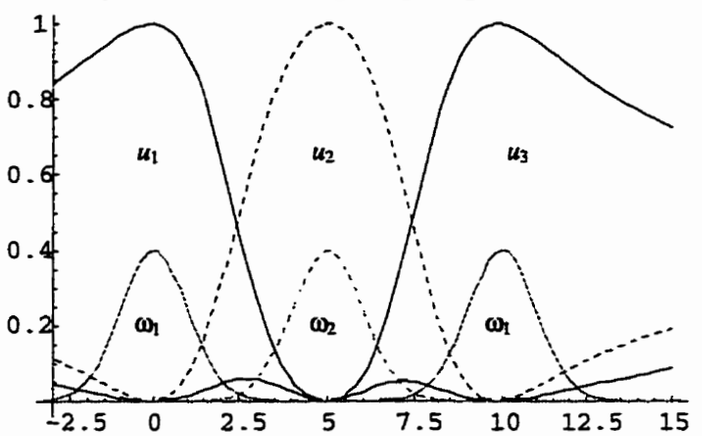

Figure 89: Fuzzy cluster (c=3) encoding using test set

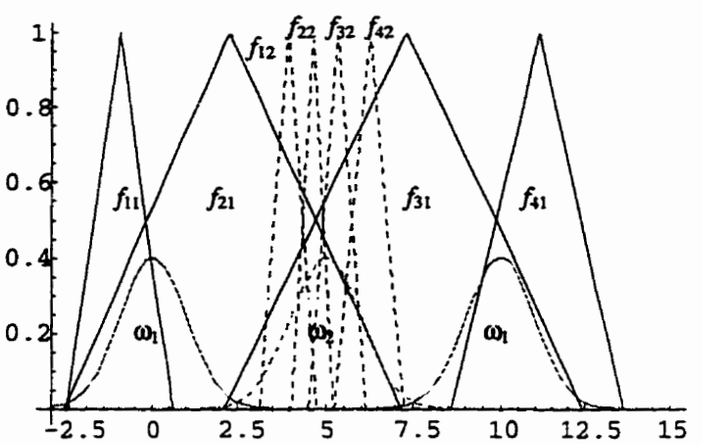

Figure 90: IQc design results (α_1=-1.8, Q_{l1}=-0.2, m_1=4.7, Q_{u1}=9.9, β_1=12.4, α_2=3.5, Q_{l2}=4.3, m_2=4.9, Q_{u2}=5.7, β_2=6.7)

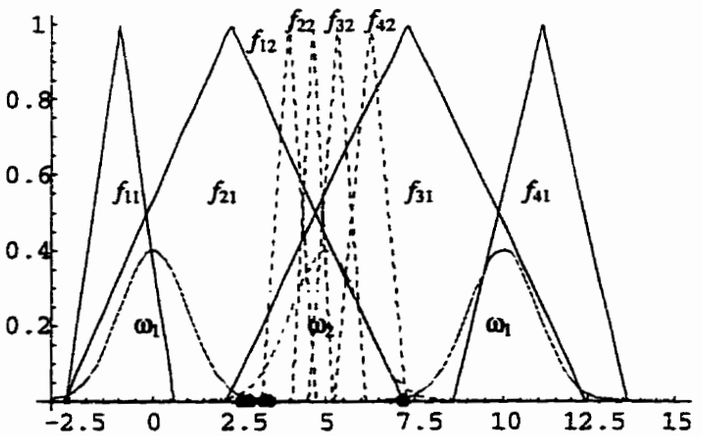

Figure 91: Class-wise fuzzy interquartile encoding using test set

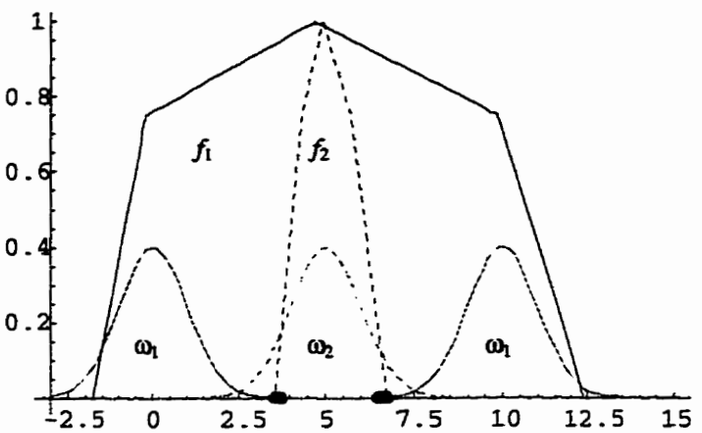

Figure 92: DPc design results (α_1=-1.8, Q_{l1}=-0.2, m_1=4.7, Q_{u1}=9.9, β_1=12.4, α_2=3.5, Q_{l2}=4.3, m_2=4.9, Q_{u2}=5.7, β_2=6.7)

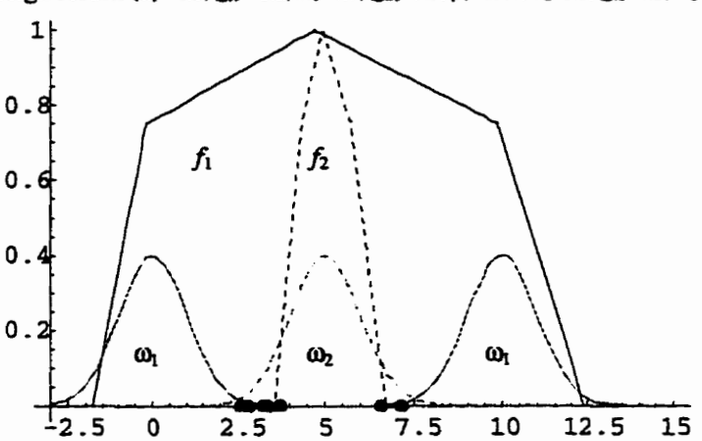

Figure 93: Class-wise dimension-preserving encoding using test set

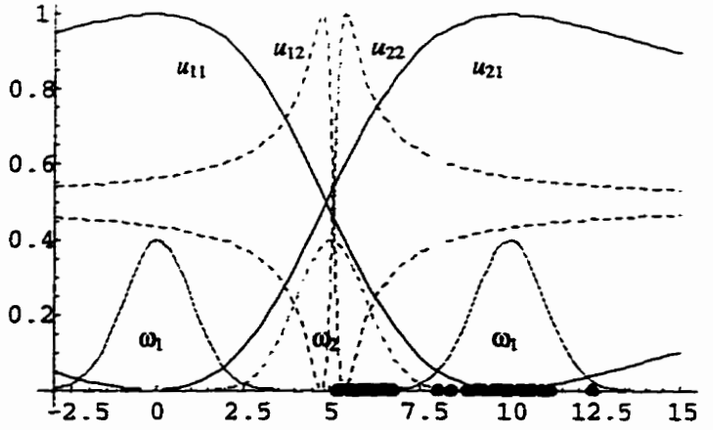

Figure 94: Class-wise fuzzy cluster (c=2) encoding using design set (v_{11}=-0.13, v_{21}=9.88, v_{12}=4.75, v_{22}=5.40)

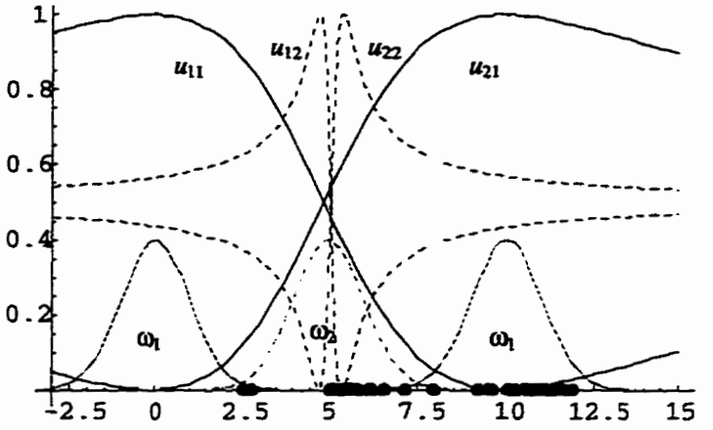

Figure 95: Class-wise fuzzy cluster (c=2) encoding using test set

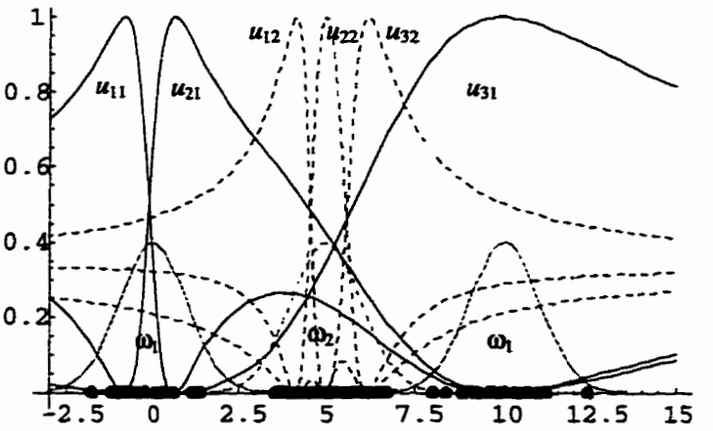

Figure 96: Class-wise fuzzy cluster (c=3) encoding using design set (v_{11}=-.9, v_{21}=.6, v_{31}=9.9, v_{12}=4.1, v_{22}=5.0, v_{32}=6.2)

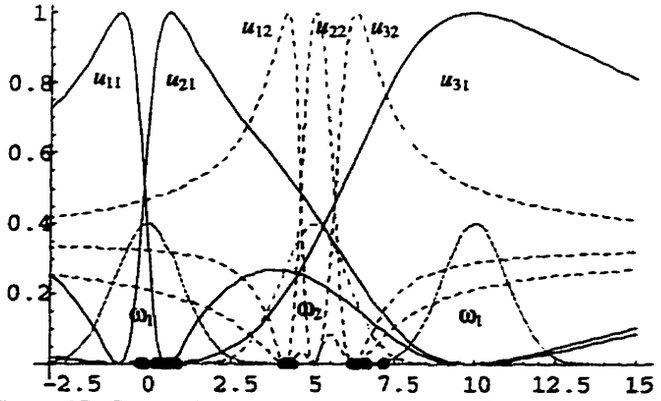

Figure 97: Class-wise fuzzy cluster (c=3) encoding using test set

7.1.4 Skewed distribution

In this experiment, all ω_1 points were sampled from $N(10,2)$, while all ω_2 points were sampled from a log normal distribution with a mean of 2 and a standard deviation of 1. Hence, ω_2 is a highly skewed class with a pdf that significantly overlaps the pdf of ω_1. NE performed poorly with both the design and test sets (Figure 98 and Figure 99). DP and IQc performed surprisingly well with such a highly skewed data set (Table 17 and Table 19).

N_d	NE		IQ		DP		CL (c=2)		IQc		DPc		CLc (c=2)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	42	8	43	7	48	2	40	10	48	2	42	8	36	14
ω_2	28	22	9	41	13	37	17	33	14	36	8	42	12	38
κ	0.28		0.68		0.70		0.46		0.68		0.68		0.48	

Table 17: Design set results using skewed data

N_d	CL (c=2)		CL (c=3)		CLc (c=2)		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	50	0	40	10	36	14	38	12
ω_2	35	15	17	33	12	38	15	35
κ	0.30		0.46		0.48		0.46	

Table 18: Design set results for fuzzy cluster encoding using different cluster numbers

N_t	NE		IQ		DP		CL (c=3)		IQc		DPc		CLc (c=2)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	26	24	35	15	47	3	40	10	47	3	38	12	36	14
ω_2	35	15	8	42	9	41	11	39	9	41	7	43	6	44
κ	-0.18		0.54		0.76		0.58		0.76		0.62		0.60	

Table 19: Test set results using skewed data

N_t	CL (c=2)		CL (c=3)		CLc (c=2)		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	50	0	40	10	36	14	28	22
ω_2	36	14	11	39	6	44	14	36
κ	0.28		0.58		0.60		0.28	

Table 20: Test set results for fuzzy cluster encoding using different cluster numbers

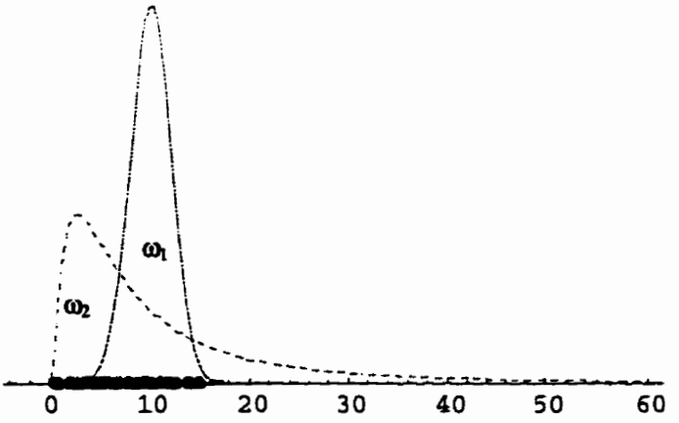

Figure 98: Non-encoded design set results

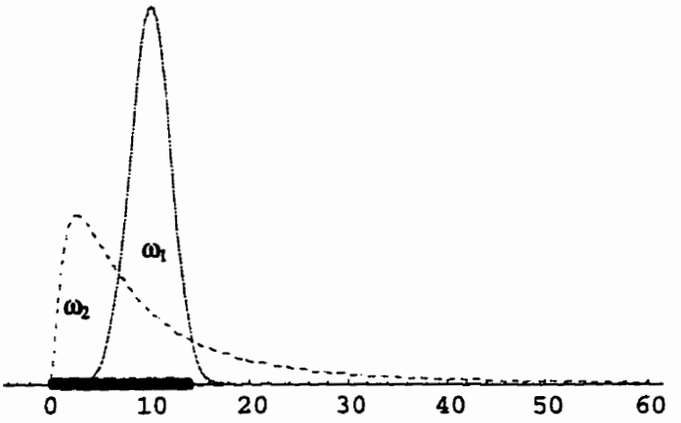

Figure 99: Non-encoded test set results

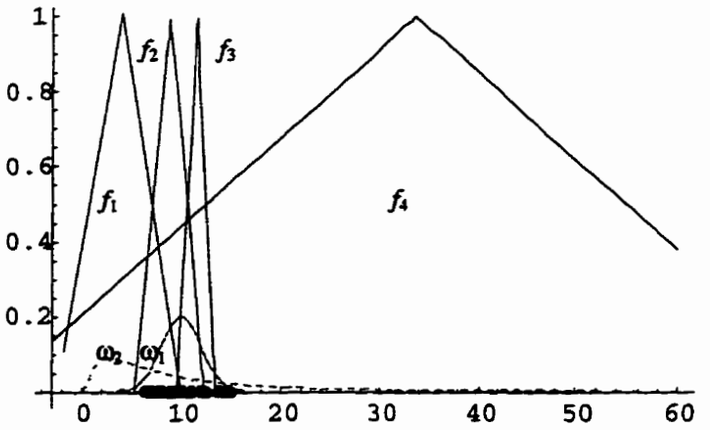

Figure 100: Fuzzy interquartile encoded design set results (α=0.51, Q_c=7.07, m=10.54, Q_w=12.45, β=54.80)

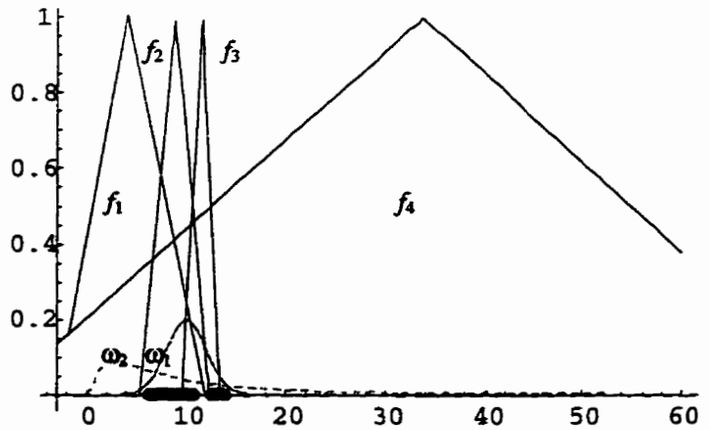

Figure 101: Fuzzy interquartile encoded test set results

Figure 102 and Figure 103 show that DP mimics the skewness of the design and test sets. The results from this and previous sections suggest that DP classification performance improves as the data become less normal.

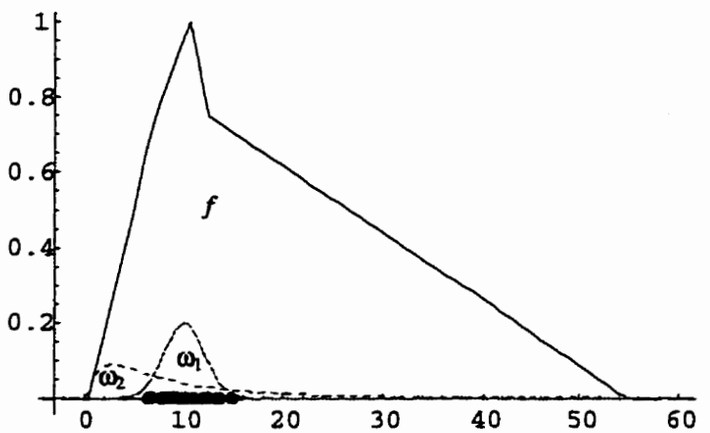

Figure 102: Dimension-preserving design set results (α=0.51, Q_l=7.07, m=10.54, Q_u=12.45, β=54.80)

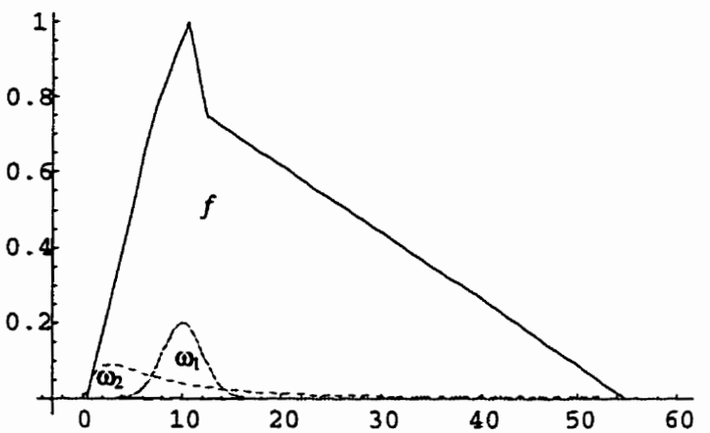

Figure 103: Dimension-preserving test set results

Figure 104 and Figure 105 show that CL with two cluster centres completely breaks down producing classification results worse than chance with all of the errors occurring around the pdf overlap of the two classes. The situation improves dramatically with three cluster centres with the errors distributed throughout the pdfs (Figure 106 and Figure 107).

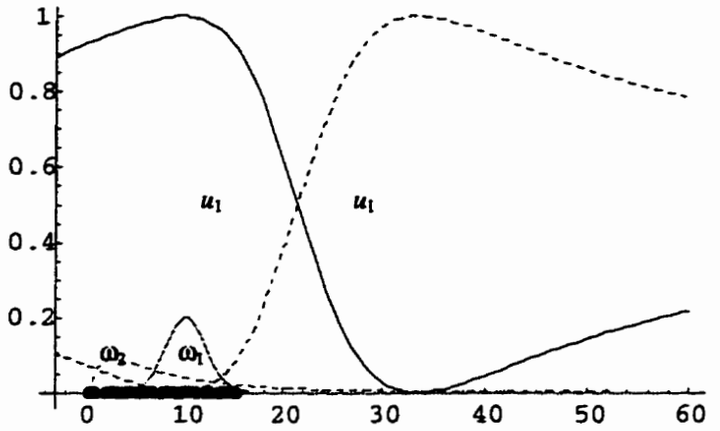

Figure 104: Fuzzy cluster (c=2) encoding using design set (v_1=9.49, v_2=33.41)

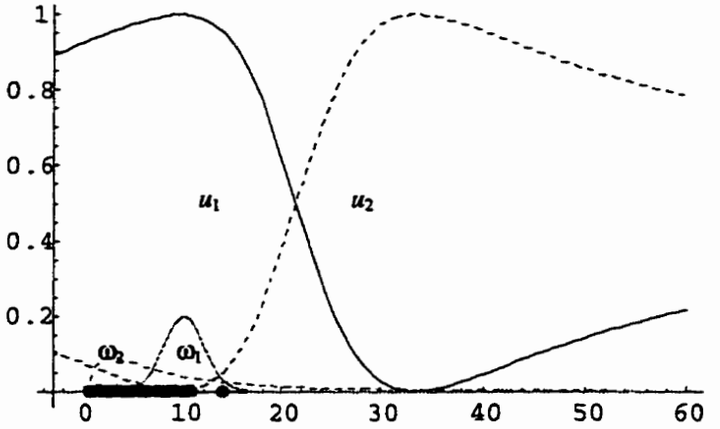

Figure 105: Fuzzy cluster (c=2) encoding using test set

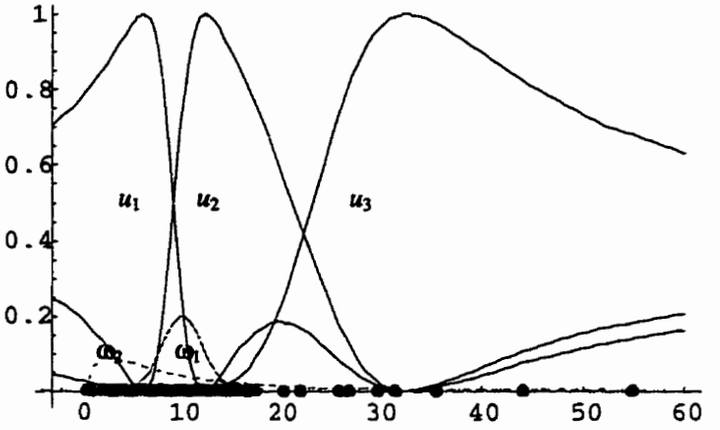

Figure 106: Fuzzy cluster (c=3) encoding using design set (v_1=6.06, v_2=12.21, v_3=32.71)

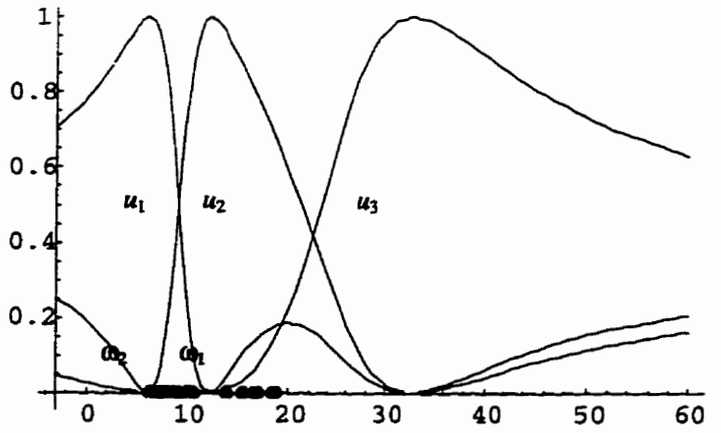

Figure 107: Fuzzy cluster (c=3) encoding using test set

Figure 108 and Figure 109 demonstrate an advantage of IQc; membership functions capture the skewness of their respective classes. The membership functions f_{11}–f_{41} are all narrow and near the mode of ω_1. The membership functions f_{12}–f_{32} are slightly less narrow and similarly surround the ω_2 mode but f_{42} spans the entire pdf of ω_2 compensating for its significant amount of skewness. Once again, the misclassifications all occur at the overlap.

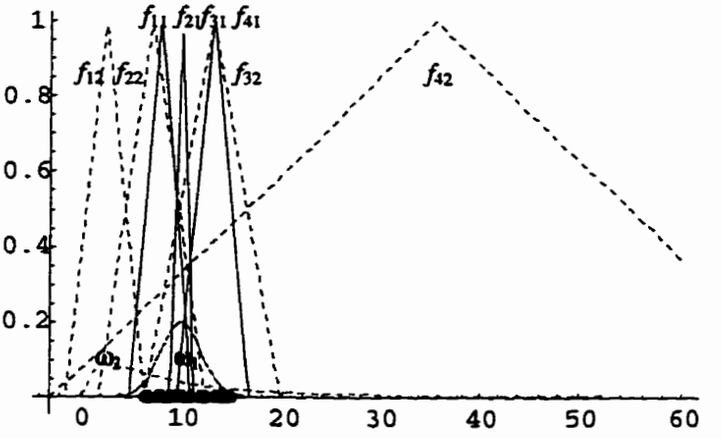

Figure 108: IQc design results (α_1=6.4, Q_{l1}=9.4, m_1=10.6, Q_{u1}=11.4, β_1=14.9, α_2=0.5, Q_{l2}=4.4, m_2=9.7, Q_{u2}=16.5, β_2=54.8)

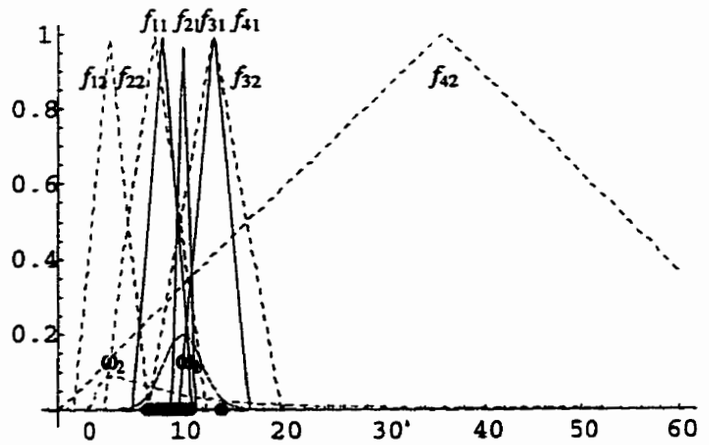

Figure 109: Class-wise fuzzy interquartile encoding using test set

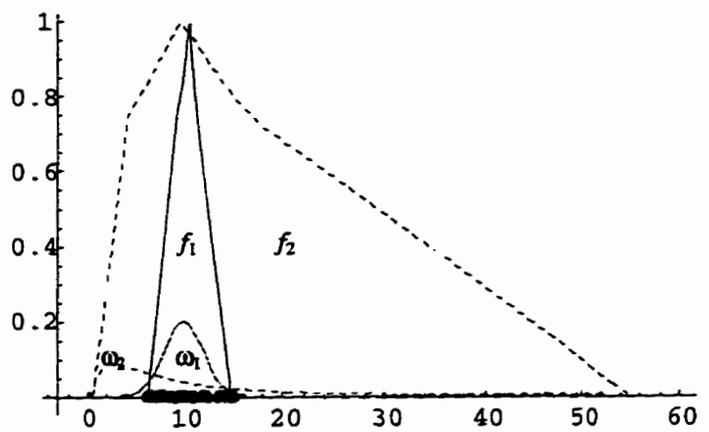

Figure 110: DPc design results (α_1=6.4, Q_{n1}=9.4, m_1=10.6, Q_{u1}=11.4, β_1=14.9, α_2=0.5, Q_{n2}=4.4, m_2=9.7, Q_{u2}=16.5, β_2=54.8)

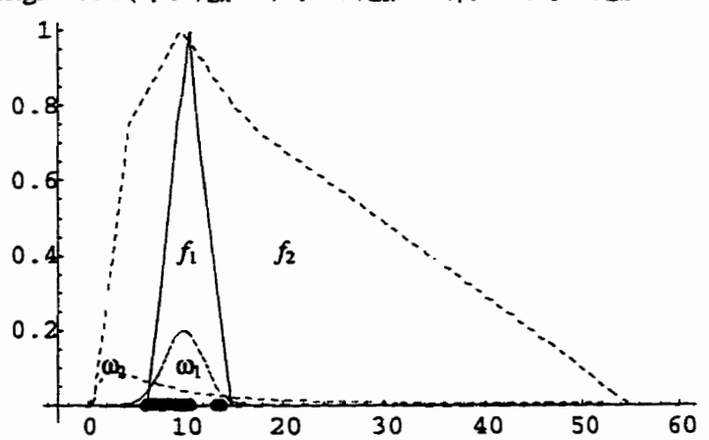

Figure 111: Class-wise dimension-preserving encoding using test set

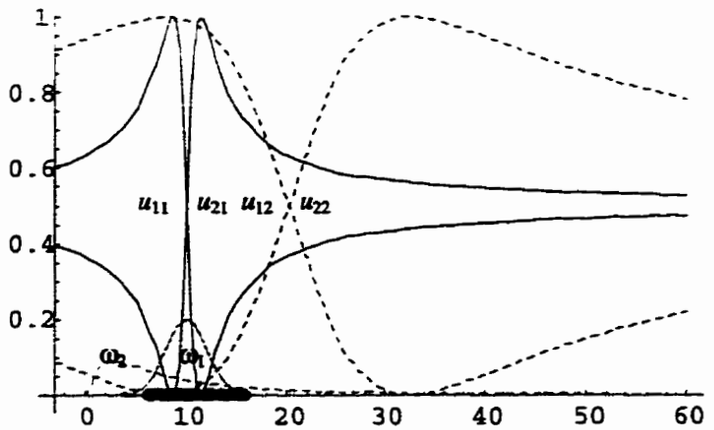

Figure 112: Class-wise fuzzy cluster (c=2) encoding using design set (v_{11}=8.63, v_{21}=11.36, v_{12}=7.94, v_{22}=32.46)

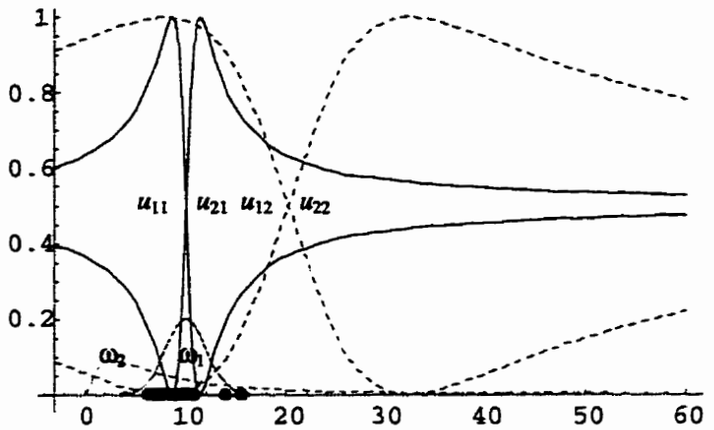

Figure 113: Class-wise fuzzy cluster (c=2) encoding using test set

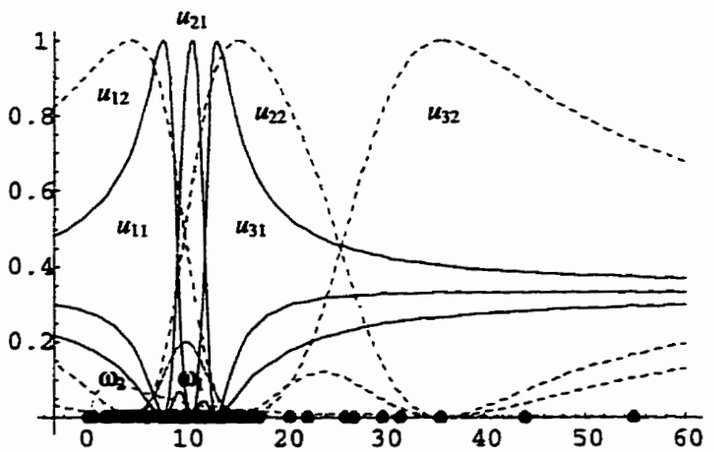

Figure 114: CLc (c=3) design results (v_{11}=7.77, v_{21}=10.66, v_{31}=13.07, v_{12}=4.64, v_{22}=15.22, v_{32}=35.93)

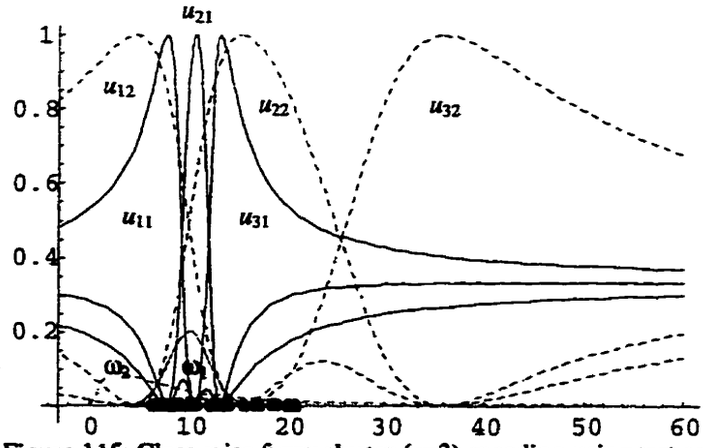

Figure 115: Class-wise fuzzy cluster (c=3) encoding using test set

7.2 Fuzzy Encoding and Linear Separability

Section 7.1 clearly demonstrated that linear discriminant analysis is a good classifier when classes possess a normal distribution and are well separated, the only misclassified points occurring where the pdfs of the two classes overlap. Classification performance significantly degrades as the data become less normal. As a whole, the fuzzy encoding methods are more robust to skewed data. IQ and IQc gave consistently good results for all data sets. As class pdfs became less symmetric, DP produced better κ scores. In some cases, taking class information into account produced better results using the class-wise variants of the encoding methods (as a whole). CL is the most variable fuzzy encoding method, at times, producing the best results over all methods. This variability is due, in large part, to the number of clusters that are selected *a priori*: good results are typically obtained when there is one cluster for every mode.

Although some of the variability in classification results may be attributed to changes in the information content of the fuzzy encoded transformations, another significant factor is the nature of the classifier. Linear discriminant analysis is a linear classifier, it can only discriminate classes that are linearly separable (see section 4.1). In section 7.1.3, for instance, LDA could not discriminate between the two classes because of the bimodal distribution of one of them. None of the fuzzy encoding methods necessarily performs a transformation of the data that is strictly linear. It is possible, therefore, to have an original data set that was linearly separable become

linearly inseparable after a transformation using one of the fuzzy encoding methods. In such a case, LDA would be able to successfully discriminate using the original data set but fail using the encoded data. The converse may also occur: a data set that is linearly inseparable may be encoded such that the transformation becomes linearly separable. For instance, assume a data set with points inside, ω_1, or outside, ω_2, the unit circle. Figure 116 illustrates that this data set is linearly inseparable; the unit circle is the optimal decision boundary. However, if each coordinate is squared, this transformation becomes linearly separable, and LDA will successfully discriminate points inside and outside the circle; the line in Figure 117 is the optimal decision boundary.

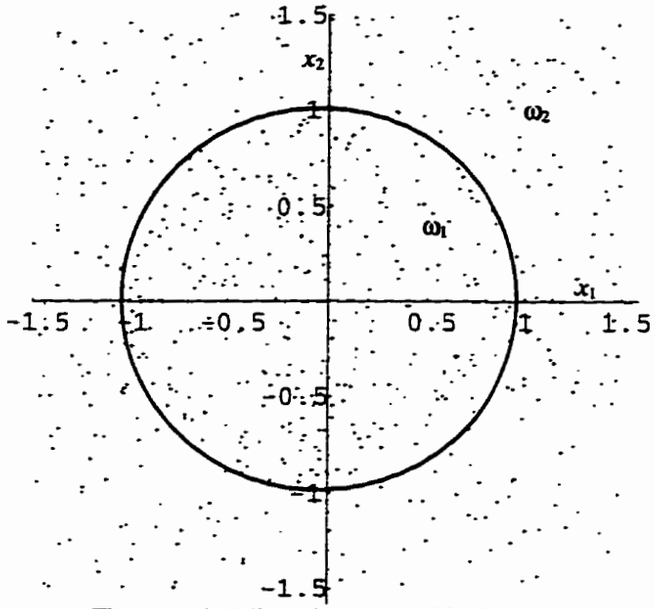

Figure 116: A linearly inseparable data set

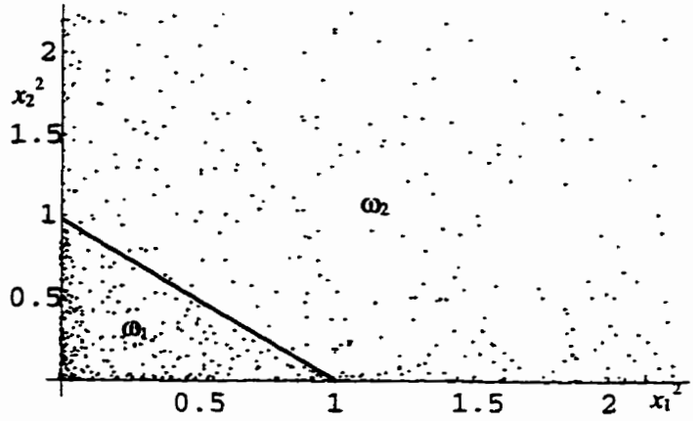

Figure 117: A linearly separable transformation

Recall from section 7.1.3, that CL with two clusters had an agreement measure of κ=0.26 for the design set where one class had a bimodal distribution. Figure 118, a plot of u_1 versus u_2 for this case, demonstrates the reason for the poor performance: the transformation is linearly inseparable and LDA will, consequently, perform poorly (the dashed line is a possible decision boundary). With the same original design set, CL with three clusters produced perfect results. Figure 119, a three-dimensional scatter plot of u_1 versus u_2 versus u_3, demonstrates the reason for the perfect agreement: the transformation into three-dimensional space is linearly separable (the dashed plane is a possible decision boundary).

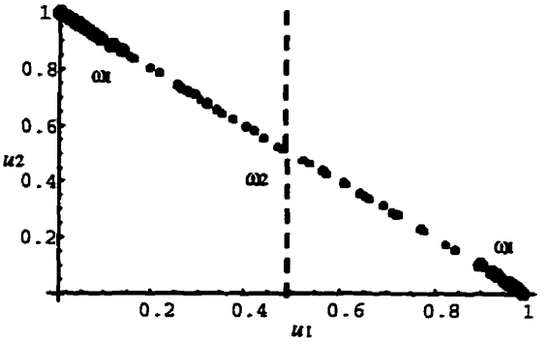

Figure 118: Linearly inseparable transformation using CL (c=2)

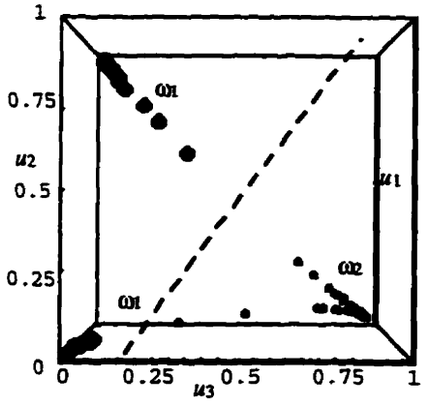

Figure 119: Linearly separable transformation using CL (c=3)

The experiment in section 7.1.3, will now be repeated except that a non-linear classifier, MLP, will be used instead of LDA. The architecture has two output PEs: if the first one is larger than the second then the actual outcome indicates the corresponding input belongs to ω_1, otherwise, it belongs to ω_1. The number of input PEs varies depending upon the method used: 1 for NE and DP; 2 and 3 for CL using 2 and 3 clusters, respectively (double for the class-wise variants); 2 for

DPc; and 4 for IQ (double for IQc). One hidden layer with 3 PEs was used with all methods. For IQ and DP, α=-1.6, Q_l=1.3, m=5.3, Q_u=8.2, β=12.5. For IQc and DPc, α_1=-1.6, Q_{l1}=0.3, m_1=4.7, Q_{u1}=10.3, β_1=12.5, α_2=2.3, Q_{l2}=4.5, m_2=5.3, Q_{u2}=5.8, β_2=7.0. Table 21 and Table 22 list the κ scores using the design set with all encoding methods and their class-wise variants, respectively. Table 23 and Table 24 are the corresponding results using the test set. Note the across the board improvement using MLP as the classifier instead of LDA. NE, which originally gave results no better than chance using LDA, now produces good results for both the design and test sets. This indicates that MLP was able to produce a non-linear decision boundary. IQ now produces perfect results for the design and test sets. IQc and DPc show improvement with the test set. The most dramatic improvements occurred with CL and CLc: all variations now give good results for both design and test sets. This demonstrates that CL transformations are non-linear. While LDA may or may not discriminate using these transformations, MLP is not affected by the non-linearities.

N_d	NE		IQ		DP		CL (c=2)		CL (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	47	3	50	0	50	0	50	0	50	0
ω_2	0	50	0	50	0	50	6	44	0	50
κ	0.94		1.00		1.00		0.88		1.00	

Table 21: Design set results using an MLP

N_d	IQc		DPc		CLc (c=2)		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	50	0	50	0	50	0	50	0
ω_2	0	50	2	48	1	49	0	50
κ	1.00		0.96		0.98		1.00	

Table 22: Design set results for class-wise variants using an MLP

N_t	NE		IQ		DP		CL (c=2)		CL (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	41	9	50	0	48	2	50	0	50	0
ω_2	0	50	0	50	0	50	10	40	0	50
κ	0.82		1.00		0.96		0.80		1.00	

Table 23: Test set results using an MLP

N_t	IQc		DPc		CLc (c=2)		CLc (c=3)	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	46	4	50	0	50	0	49	1
ω_2	0	50	1	49	0	50	0	50
κ	0.92		0.98		1.00		0.98	

Table 24: Test set results for class-wise variants using an MLP

For completeness, a set of tables follows that list the points, the CL encodings, and class labels, for both the design and test set: Table 25and Table 26 for CL using two clusters; Table 27 and Table 28 for CL using three clusters; Table 29and Table 30 for CLc using two clusters; Table 31and Table 32 for CL using three clusters. Entries in italics indicate points that were misclassified.

Point	u_1	u_2	ω	Point	u_1	u_2	ω	Point	u_1	u_2	ω
10.44	0.06	0.94	1	0.09	0.96	0.04	1	5.18	0.44	0.56	2
0.50	0.97	0.03	1	9.38	0.02	0.98	1	4.04	0.77	0.23	2
9.70	0.03	0.97	1	-0.52	0.94	0.06	1	*6.46*	*0.13*	*0.87*	2
0.73	0.98	0.02	1	9.29	0.02	0.98	1	6.10	0.20	0.80	2
8.19	0.00	1.00	1	0.31	0.97	0.03	1	5.32	0.40	0.60	2
0.33	0.97	0.03	1	12.47	0.13	0.87	1	3.71	0.84	0.16	2
10.34	0.06	0.94	1	1.27	1.00	0.00	1	5.26	0.42	0.58	2
-0.85	0.92	0.08	1	11.23	0.09	0.91	1	4.53	0.64	0.36	2
10.32	0.06	0.94	1	0.55	0.98	0.02	1	4.46	0.66	0.34	2
0.53	0.98	0.02	1	10.01	0.04	0.96	1	5.82	0.26	0.74	2
9.98	0.04	0.96	1	0.32	0.97	0.03	1	4.64	0.61	0.39	2
-0.15	0.95	0.05	1	9.72	0.03	0.97	1	*2.91*	*0.96*	*0.04*	*2*
8.53	0.00	1.00	1	1.01	0.99	0.01	1	4.65	0.61	0.39	2
0.19	0.96	0.04	1	11.90	0.12	0.88	1	5.84	0.26	0.74	2
10.71	0.07	0.93	1	0.73	0.98	0.02	1	5.08	0.47	0.53	2
0.72	0.98	0.02	1	9.95	0.04	0.96	1	6.29	0.16	0.84	2
9.96	0.04	0.96	1	0.68	0.98	0.02	1	*6.58*	*0.11*	*0.89*	*2*
-0.85	0.92	0.08	1	5.80	0.27	0.73	2	*2.32*	*0.99*	*0.01*	*2*
10.94	0.08	0.92	1	5.52	0.34	0.66	2	5.61	0.32	0.68	2
0.24	0.97	0.03	1	*6.79*	*0.08*	*0.92*	2	5.76	0.28	0.72	2
10.59	0.07	0.93	1	6.01	0.22	0.78	2	4.05	0.77	0.23	2
0.26	0.97	0.03	1	4.49	0.65	0.35	2	5.48	0.36	0.64	2
9.60	0.03	0.97	1	6.26	0.16	0.84	2	5.85	0.25	0.75	2
-1.59	0.90	0.10	1	5.66	0.30	0.70	2	4.27	0.71	0.29	2
10.28	0.05	0.95	1	4.92	0.52	0.48	2	4.76	0.57	0.43	2
-1.49	0.90	0.10	1	5.40	0.38	0.62	2	4.91	0.53	0.47	2
8.75	0.01	0.99	1	4.87	0.54	0.46	2	4.24	0.72	0.28	2
0.15	0.96	0.04	1	6.40	0.14	0.86	2	4.35	0.69	0.31	2
11.99	0.12	0.88	1	5.05	0.48	0.52	2	3.81	0.82	0.18	2
0.69	0.98	0.02	1	5.71	0.29	0.71	2	5.59	0.32	0.68	2
11.21	0.09	0.91	1	4.79	0.57	0.43	2	4.26	0.72	0.28	2
0.51	0.97	0.03	1	5.60	0.32	0.68	2	*7.00*	*0.05*	*0.95*	*2*
10.47	0.06	0.94	1	4.02	0.78	0.22	2	5.30	0.41	0.59	2
								5.75	0.28	0.72	2

Table 25: CL (c=2) design results using MLP (v_1=1.76, v_2=8.24)

Point	u_1	u_2	ω	Point	u_1	u_2	ω	Point	u_1	u_2	ω
10.16	0.05	0.95	1	1.29	1.00	0.00	1	5.35	0.39	0.61	2
1.45	1.00	0.00	1	10.56	0.06	0.94	1	5.14	0.46	0.54	2
10.60	0.07	0.93	1	-0.48	0.94	0.06	1	4.16	0.74	0.26	2
0.92	0.99	0.01	1	8.58	0.00	1.00	1	*3.54*	*0.87*	*0.13*	2
10.27	0.05	0.95	1	-1.49	0.90	0.10	1	4.43	0.67	0.33	2
0.77	0.98	0.02	1	10.23	0.05	0.95	1	5.37	0.39	0.61	2
8.99	0.01	0.99	1	0.89	0.99	0.01	1	5.75	0.28	0.72	2
0.91	0.99	0.01	1	11.14	0.09	0.91	1	6.23	0.17	0.83	2
8.34	0.00	1.00	1	1.09	0.99	0.01	1	4.05	0.77	0.23	2
1.28	1.00	0.00	1	8.80	0.01	0.99	1	3.96	0.79	0.21	2
10.92	0.08	0.92	1	1.92	1.00	0.00	1	4.56	0.63	0.37	2
1.06	0.99	0.01	1	8.33	0.00	1.00	1	5.23	0.43	0.57	2
11.19	0.09	0.91	1	0.28	0.97	0.03	1	4.69	0.60	0.40	2
1.75	1.00	0.00	1	9.69	0.03	0.97	1	6.20	0.17	0.83	2
10.86	0.08	0.92	1	-1.43	0.90	0.10	1	*6.54*	*0.11*	*0.89*	2
0.96	0.99	0.01	1	10.91	0.08	0.92	1	*3.14*	*0.93*	*0.07*	2
10.84	0.08	0.92	1	1.81	1.00	0.00	1	4.29	0.71	0.29	2
1.72	1.00	0.00	1	5.83	0.26	0.74	2	4.17	0.74	0.26	2
10.49	0.06	0.94	1	5.09	0.47	0.53	2	4.34	0.69	0.31	2
1.18	0.99	0.01	1	3.77	0.83	0.17	2	3.72	0.84	0.16	2
9.75	0.03	0.97	1	4.55	0.64	0.36	2	*6.57*	*0.11*	*0.89*	2
1.12	0.99	0.01	1	5.81	0.26	0.74	2	4.82	0.55	0.45	2
10.18	0.05	0.95	1	6.04	0.21	0.79	2	4.16	0.74	0.26	2
0.21	0.96	0.04	1	*3.39*	*0.90*	*0.10*	2	6.00	0.22	0.78	2
9.72	0.03	0.97	1	4.85	0.55	0.45	2	5.00	0.50	0.50	2
-0.21	0.95	0.05	1	5.35	0.39	0.61	2	5.30	0.41	0.59	2
9.37	0.02	0.98	1	4.76	0.57	0.43	2	4.27	0.71	0.29	2
0.02	0.96	0.04	1	4.29	0.71	0.29	2	4.79	0.57	0.43	2
10.46	0.06	0.94	1	6.09	0.20	0.80	2	5.63	0.31	0.69	2
0.52	0.98	0.02	1	*2.96*	*0.95*	*0.05*	2	*6.49*	*0.12*	*0.88*	2
8.14	0.00	1.00	1	4.06	0.77	0.23	2	*2.43*	*0.99*	*0.01*	2
1.37	1.00	0.00	1	5.91	0.24	0.76	2	*3.53*	*0.88*	*0.12*	2
11.78	0.11	0.89	1	5.93	0.23	0.77	2	4.27	0.71	0.29	2
								6.51	*0.12*	*0.88*	2

Table 26: CL (c=2) test results using MLP

Point	u_1	u_2	u_3	ω	Point	u_1	u_2	u_3	ω
10.44	0.00	1.00	0.00	1	5.80	0.01	0.02	0.97	2
0.50	1.00	0.00	0.00	1	5.52	0.00	0.00	0.99	2
9.70	0.00	0.98	0.02	1	6.79	0.04	0.16	0.80	2
0.73	0.99	0.00	0.01	1	6.01	0.02	0.03	0.95	2
8.19	0.04	0.64	0.32	1	4.49	0.03	0.02	0.95	2
0.33	1.00	0.00	0.00	1	6.26	0.03	0.06	0.91	2
10.34	0.00	1.00	0.00	1	5.66	0.01	0.01	0.99	2
-0.85	0.96	0.01	0.03	1	4.92	0.00	0.00	0.99	2
10.32	0.00	1.00	0.00	1	5.40	0.00	0.00	1.00	2
0.53	1.00	0.00	0.00	1	4.87	0.01	0.00	0.99	2
9.98	0.00	0.99	0.00	1	6.40	0.03	0.08	0.89	2
-0.15	0.99	0.00	0.01	1	5.05	0.00	0.00	1.00	2
8.53	0.03	0.75	0.21	1	5.71	0.01	0.01	0.98	2
0.19	1.00	0.00	0.00	1	4.79	0.01	0.01	0.98	2
10.71	0.00	0.99	0.01	1	5.60	0.00	0.01	0.99	2
0.72	0.99	0.00	0.01	1	4.02	0.09	0.03	0.87	2
9.96	0.00	0.99	0.00	1	5.18	0.00	0.00	1.00	2
-0.85	0.96	0.01	0.03	1	4.04	0.09	0.03	0.88	2
10.94	0.00	0.98	0.01	1	6.46	0.03	0.09	0.88	2
0.24	1.00	0.00	0.00	1	6.10	0.02	0.04	0.94	2
10.59	0.00	1.00	0.00	1	5.32	0.00	0.00	1.00	2
0.26	1.00	0.00	0.00	1	3.71	0.16	0.04	0.80	2
9.60	0.01	0.97	0.02	1	5.26	0.00	0.00	1.00	2
-1.59	0.91	0.02	0.07	1	4.53	0.03	0.01	0.96	2
10.28	0.00	1.00	0.00	1	4.46	0.03	0.02	0.95	2
-1.49	0.92	0.02	0.06	1	5.82	0.01	0.02	0.97	2
8.75	0.03	0.82	0.15	1	4.64	0.02	0.01	0.97	2
0.15	1.00	0.00	0.00	1	2.91	0.41	0.05	0.53	2
11.99	0.02	0.92	0.06	1	4.65	0.02	0.01	0.97	2
0.69	0.99	0.00	0.01	1	5.84	0.01	0.02	0.97	2
11.21	0.01	0.97	0.02	1	5.08	0.00	0.00	1.00	2
0.51	1.00	0.00	0.00	1	6.29	0.03	0.06	0.91	2
10.47	0.00	1.00	0.00	1	6.58	0.04	0.11	0.85	2
0.09	1.00	0.00	0.00	1	2.32	0.64	0.04	0.32	2
9.38	0.01	0.95	0.05	1	5.61	0.00	0.01	0.99	2
-0.52	0.98	0.01	0.02	1	5.76	0.01	0.01	0.98	2
9.29	0.01	0.93	0.06	1	4.05	0.09	0.03	0.88	2
0.31	1.00	0.00	0.00	1	5.48	0.00	0.00	1.00	2
12.47	0.03	0.89	0.08	1	5.85	0.01	0.02	0.97	2
1.27	0.93	0.01	0.06	1	4.27	0.05	0.02	0.92	2
11.23	0.01	0.97	0.02	1	4.76	0.01	0.01	0.98	2
0.55	1.00	0.00	0.00	1	4.91	0.01	0.00	0.99	2
10.01	0.00	1.00	0.00	1	4.24	0.06	0.03	0.92	2
0.32	1.00	0.00	0.00	1	4.35	0.04	0.02	0.94	2
9.72	0.00	0.98	0.02	1	3.81	0.13	0.04	0.83	2
1.01	0.96	0.01	0.03	1	5.59	0.00	0.01	0.99	2
11.90	0.02	0.93	0.05	1	4.26	0.06	0.02	0.92	2
0.73	0.99	0.00	0.01	1	7.00	0.05	0.21	0.74	2
9.95	0.00	0.99	0.00	1	5.30	0.00	0.00	1.00	2
0.68	0.99	0.00	0.01	1	5.75	0.01	0.01	0.98	2

Table 27: CL (c=3) design results using MLP (ν_1=0.27, ν_2=5.24, ν_3=10.28)

Point	u_1	u_2	u_3	ω	Point	u_1	u_2	u_3	ω
10.16	0.00	1.00	0.00	1	5.83	0.01	0.02	0.97	2
1.45	0.90	0.02	0.09	1	5.09	0.00	0.00	1.00	2
10.60	0.00	1.00	0.00	1	3.77	0.14	0.04	0.82	2
0.92	0.97	0.00	0.02	1	4.55	0.03	0.01	0.96	2
10.27	0.00	1.00	0.00	1	5.81	0.01	0.02	0.97	2
0.77	0.98	0.00	0.01	1	6.04	0.02	0.03	0.95	2
8.99	0.02	0.88	0.10	1	3.39	0.25	0.05	0.70	2
0.91	0.97	0.00	0.02	1	4.85	0.01	0.01	0.99	2
8.34	0.04	0.69	0.27	1	5.35	0.00	0.00	1.00	2
1.28	0.93	0.01	0.06	1	4.76	0.01	0.01	0.98	2
10.92	0.00	0.98	0.01	1	4.29	0.05	0.02	0.92	2
1.06	0.96	0.01	0.03	1	6.09	0.02	0.04	0.94	2
11.19	0.01	0.97	0.02	1	2.96	0.39	0.05	0.55	2
1.75	0.83	0.03	0.15	1	4.06	0.09	0.03	0.88	2
10.86	0.00	0.99	0.01	1	5.91	0.01	0.02	0.96	2
0.96	0.97	0.01	0.03	1	5.93	0.01	0.02	0.96	2
10.84	0.00	0.99	0.01	1	5.35	0.00	0.00	1.00	2
1.72	0.83	0.02	0.14	1	5.14	0.00	0.00	1.00	2
10.49	0.00	1.00	0.00	1	4.16	0.07	0.03	0.90	2
1.18	0.94	0.01	0.05	1	3.54	0.20	0.05	0.75	2
9.75	0.00	0.98	0.01	1	4.43	0.04	0.02	0.95	2
1.12	0.95	0.01	0.04	1	5.37	0.00	0.00	1.00	2
10.18	0.00	1.00	0.00	1	5.75	0.01	0.01	0.98	2
0.21	1.00	0.00	0.00	1	6.23	0.03	0.05	0.92	2
9.72	0.00	0.98	0.02	1	4.05	0.09	0.03	0.88	2
-0.21	0.99	0.00	0.01	1	3.96	0.10	0.04	0.86	2
9.37	0.01	0.94	0.05	1	4.56	0.02	0.01	0.96	2
0.02	1.00	0.00	0.00	1	5.23	0.00	0.00	1.00	2
10.46	0.00	1.00	0.00	1	4.69	0.02	0.01	0.98	2
0.52	1.00	0.00	0.00	1	6.20	0.02	0.05	0.93	2
8.14	0.05	0.62	0.34	1	6.54	0.04	0.10	0.86	2
1.37	0.91	0.01	0.07	1	3.14	0.33	0.05	0.62	2
11.78	0.02	0.94	0.05	1	4.29	0.05	0.02	0.92	2
1.29	0.93	0.01	0.06	1	4.17	0.07	0.03	0.90	2
10.56	0.00	1.00	0.00	1	4.34	0.05	0.02	0.93	2
-0.48	0.98	0.00	0.02	1	3.72	0.16	0.04	0.80	2
8.58	0.03	0.77	0.20	1	6.57	0.04	0.11	0.85	2
-1.49	0.92	0.02	0.06	1	4.82	0.01	0.01	0.99	2
10.23	0.00	1.00	0.00	1	4.16	0.07	0.03	0.90	2
0.89	0.98	0.00	0.02	1	6.00	0.02	0.03	0.95	2
11.14	0.01	0.97	0.02	1	5.00	0.00	0.00	1.00	2
1.09	0.95	0.01	0.04	1	5.30	0.00	0.00	1.00	2
8.80	0.02	0.83	0.14	1	4.27	0.05	0.02	0.92	2
1.92	0.78	0.03	0.19	1	4.79	0.01	0.01	0.98	2
8.33	0.04	0.69	0.27	1	5.63	0.01	0.01	0.99	2
0.28	1.00	0.00	0.00	1	6.49	0.04	0.10	0.87	2
9.69	0.00	0.98	0.02	1	2.43	0.60	0.05	0.35	2
-1.43	0.92	0.02	0.06	1	3.53	0.21	0.05	0.75	2
10.91	0.00	0.98	0.01	1	4.27	0.05	0.02	0.92	2
1.81	0.81	0.03	0.16	1	6.51	0.04	0.10	0.87	2

Table 28: CL (c=3) test results using MLP

Point	u_{11}	u_{21}	u_{12}	u_{22}	ω	Point	u_{11}	u_{21}	u_{12}	u_{22}	ω
10.44	1.00	0.00	0.36	0.64	1	5.80	0.62	0.38	0.00	1.00	2
0.50	0.00	1.00	0.66	0.34	1	5.52	0.56	0.44	0.06	0.94	2
9.70	1.00	0.00	0.34	0.66	1	6.79	0.78	0.22	0.13	0.87	2
0.73	0.00	1.00	0.67	0.33	1	6.01	0.65	0.35	0.01	0.99	2
8.19	0.94	0.06	0.27	0.73	1	4.49	0.36	0.64	0.98	0.02	2
0.33	0.00	1.00	0.66	0.34	1	6.26	0.70	0.30	0.05	0.95	2
10.34	1.00	0.00	0.36	0.64	1	5.66	0.59	0.41	0.01	0.99	2
-0.85	0.01	0.99	0.63	0.37	1	4.92	0.44	0.56	0.69	0.31	2
10.32	1.00	0.00	0.36	0.64	1	5.40	0.54	0.46	0.13	0.87	2
0.53	0.00	1.00	0.66	0.34	1	4.87	0.43	0.57	0.74	0.26	2
9.98	1.00	0.00	0.35	0.65	1	6.40	0.72	0.28	0.07	0.93	2
-0.15	0.00	1.00	0.64	0.36	1	5.05	0.47	0.53	0.52	0.48	2
8.53	0.96	0.04	0.29	0.71	1	5.71	0.60	0.40	0.01	0.99	2
0.19	0.00	1.00	0.65	0.35	1	4.79	0.42	0.58	0.82	0.18	2
10.71	1.00	0.00	0.37	0.63	1	5.60	0.58	0.42	0.03	0.97	2
0.72	0.00	1.00	0.67	0.33	1	4.02	0.28	0.72	0.98	0.02	2
9.96	1.00	0.00	0.35	0.65	1	5.18	0.49	0.51	0.35	0.65	2
-0.85	0.01	0.99	0.63	0.37	1	4.04	0.28	0.72	0.98	0.02	2
10.94	1.00	0.00	0.37	0.63	1	6.46	0.73	0.27	0.08	0.92	2
0.24	0.00	1.00	0.65	0.35	1	6.10	0.67	0.33	0.02	0.98	2
10.59	1.00	0.00	0.36	0.64	1	5.32	0.52	0.48	0.20	0.80	2
0.26	0.00	1.00	0.65	0.35	1	3.71	0.23	0.77	0.93	0.07	2
9.60	1.00	0.00	0.34	0.66	1	5.26	0.51	0.49	0.27	0.73	2
-1.59	0.02	0.98	0.61	0.39	1	4.53	0.37	0.63	0.97	0.03	2
10.28	1.00	0.00	0.36	0.64	1	4.46	0.35	0.65	0.99	0.01	2
-1.49	0.02	0.98	0.61	0.39	1	5.82	0.62	0.38	0.00	1.00	2
8.75	0.97	0.03	0.30	0.70	1	4.64	0.39	0.61	0.93	0.07	2
0.15	0.00	1.00	0.65	0.35	1	2.91	0.12	0.88	0.81	0.19	2
11.99	0.98	0.02	0.39	0.61	1	4.65	0.39	0.61	0.92	0.08	2
0.69	0.00	1.00	0.67	0.33	1	5.84	0.62	0.38	0.00	1.00	2
11.21	0.99	0.01	0.38	0.62	1	5.08	0.47	0.53	0.48	0.52	2
0.51	0.00	1.00	0.66	0.34	1	6.29	0.71	0.29	0.05	0.95	2
10.47	1.00	0.00	0.36	0.64	1	6.58	0.75	0.25	0.10	0.90	2
0.09	0.00	1.00	0.65	0.35	1	*2.32*	*0.07*	*0.93*	*0.76*	*0.24*	*2*
9.38	0.99	0.01	0.33	0.67	1	5.61	0.58	0.42	0.03	0.97	2
-0.52	0.00	1.00	0.63	0.37	1	5.76	0.61	0.39	0.00	1.00	2
9.29	0.99	0.01	0.33	0.67	1	4.05	0.28	0.72	0.98	0.02	2
0.31	0.00	1.00	0.66	0.34	1	5.48	0.55	0.45	0.08	0.92	2
12.47	0.97	0.03	0.40	0.60	1	5.85	0.62	0.38	0.00	1.00	2
1.27	0.01	0.99	0.69	0.31	1	4.27	0.32	0.68	1.00	0.00	2
11.23	0.99	0.01	0.38	0.62	1	4.76	0.41	0.59	0.85	0.15	2
0.55	0.00	1.00	0.66	0.34	1	4.91	0.44	0.56	0.70	0.30	2
10.01	1.00	0.00	0.35	0.65	1	4.24	0.31	0.69	1.00	0.00	2
0.32	0.00	1.00	0.66	0.34	1	4.35	0.33	0.67	1.00	0.00	2
9.72	1.00	0.00	0.34	0.66	1	3.81	0.24	0.76	0.94	0.06	2
1.01	0.01	0.99	0.68	0.32	1	5.59	0.57	0.43	0.03	0.97	2
11.90	0.98	0.02	0.39	0.61	1	4.26	0.32	0.68	1.00	0.00	2
0.73	0.00	1.00	0.67	0.33	1	7.00	0.82	0.18	0.16	0.84	2
9.95	1.00	0.00	0.35	0.65	1	5.30	0.52	0.48	0.22	0.78	2
0.68	0.00	1.00	0.67	0.33	1	5.75	0.61	0.39	0.00	1.00	2

Table 29: CLc (c=2) design results using MLP (ν_{11}=0.18, ν_{21}=10.24, ν_{12}=4.30, ν_{22}=5.83)

Point	u_{11}	u_{21}	u_{12}	u_{22}	ω	Point	u_{11}	u_{21}	u_{12}	u_{22}	ω
10.16	1.00	0.00	0.35	0.65	1	5.83	0.62	0.38	0.00	1.00	2
1.45	0.02	0.98	0.70	0.30	1	5.09	0.48	0.52	0.47	0.53	2
10.60	1.00	0.00	0.36	0.64	1	3.77	0.24	0.76	0.94	0.06	2
0.92	0.01	0.99	0.68	0.32	1	4.55	0.37	0.63	0.96	0.04	2
10.27	1.00	0.00	0.36	0.64	1	5.81	0.62	0.38	0.00	1.00	2
0.77	0.00	1.00	0.67	0.33	1	6.04	0.66	0.34	0.01	0.99	2
8.99	0.98	0.02	0.31	0.69	1	3.39	0.18	0.82	0.88	0.12	2
0.91	0.01	0.99	0.68	0.32	1	4.85	0.43	0.57	0.77	0.23	2
8.34	0.95	0.05	0.28	0.72	1	5.35	0.53	0.47	0.17	0.83	2
1.28	0.01	0.99	0.69	0.31	1	4.76	0.41	0.59	0.84	0.16	2
10.92	1.00	0.00	0.37	0.63	1	4.29	0.32	0.68	1.00	0.00	2
1.06	0.01	0.99	0.68	0.32	1	6.09	0.67	0.33	0.02	0.98	2
11.19	0.99	0.01	0.38	0.62	1	2.96	0.13	0.87	0.82	0.18	2
1.75	0.03	0.97	0.72	0.28	1	4.06	0.28	0.72	0.98	0.02	2
10.86	1.00	0.00	0.37	0.63	1	5.91	0.64	0.36	0.00	1.00	2
0.96	0.01	0.99	0.68	0.32	1	5.93	0.64	0.36	0.00	1.00	2
10.84	1.00	0.00	0.37	0.63	1	5.35	0.53	0.47	0.17	0.83	2
1.72	0.03	0.97	0.72	0.28	1	5.14	0.49	0.51	0.41	0.59	2
10.49	1.00	0.00	0.36	0.64	1	4.16	0.30	0.70	0.99	0.01	2
1.18	0.01	0.99	0.69	0.31	1	3.54	0.20	0.80	0.90	0.10	2
9.75	1.00	0.00	0.34	0.66	1	4.43	0.35	0.65	0.99	0.01	2
1.12	0.01	0.99	0.69	0.31	1	5.37	0.53	0.47	0.16	0.84	2
10.18	1.00	0.00	0.35	0.65	1	5.75	0.60	0.40	0.00	1.00	2
0.21	0.00	1.00	0.65	0.35	1	6.23	0.69	0.31	0.04	0.96	2
9.72	1.00	0.00	0.34	0.66	1	4.05	0.28	0.72	0.98	0.02	2
-0.21	0.00	1.00	0.64	0.36	1	3.96	0.27	0.73	0.97	0.03	2
9.37	0.99	0.01	0.33	0.67	1	4.56	0.37	0.63	0.96	0.04	2
0.02	0.00	1.00	0.65	0.35	1	5.23	0.50	0.50	0.30	0.70	2
10.46	1.00	0.00	0.36	0.64	1	4.69	0.40	0.60	0.90	0.10	2
0.52	0.00	1.00	0.66	0.34	1	6.20	0.69	0.31	0.04	0.96	2
8.14	0.94	0.06	0.27	0.73	1	6.54	0.75	0.25	0.09	0.91	2
1.37	0.02	0.98	0.70	0.30	1	3.14	0.15	0.85	0.84	0.16	2
11.78	0.98	0.02	0.39	0.61	1	4.29	0.32	0.68	1.00	0.00	2
1.29	0.02	0.98	0.69	0.31	1	4.17	0.30	0.70	0.99	0.01	2
10.56	1.00	0.00	0.36	0.64	1	4.34	0.33	0.67	1.00	0.00	2
-0.48	0.00	1.00	0.64	0.36	1	3.72	0.23	0.77	0.93	0.07	2
8.58	0.96	0.04	0.29	0.71	1	6.57	0.75	0.25	0.10	0.90	2
-1.49	0.02	0.98	0.61	0.39	1	4.82	0.42	0.58	0.79	0.21	2
10.23	1.00	0.00	0.36	0.64	1	4.16	0.30	0.70	0.99	0.01	2
0.89	0.01	0.99	0.68	0.32	1	6.00	0.65	0.35	0.01	0.99	2
11.14	0.99	0.01	0.38	0.62	1	5.00	0.46	0.54	0.59	0.41	2
1.09	0.01	0.99	0.69	0.31	1	5.30	0.52	0.48	0.22	0.78	2
8.80	0.97	0.03	0.30	0.70	1	4.27	0.32	0.68	1.00	0.00	2
1.92	0.04	0.96	0.73	0.27	1	4.79	0.42	0.58	0.82	0.18	2
8.33	0.95	0.05	0.28	0.72	1	5.63	0.58	0.42	0.02	0.98	2
0.28	0.00	1.00	0.66	0.34	1	6.49	0.74	0.26	0.08	0.92	2
9.69	1.00	0.00	0.34	0.66	1	2.43	0.08	0.92	0.77	0.23	2
-1.43	0.02	0.98	0.62	0.38	1	3.53	0.20	0.80	0.90	0.10	2
10.91	1.00	0.00	0.37	0.63	1	4.27	0.32	0.68	1.00	0.00	2
1.81	0.04	0.96	0.72	0.28	1	6.51	0.74	0.26	0.09	0.91	2

Table 30: CLc (c=2) test results using MLP

Pt	u_{11}	u_{21}	u_{31}	u_{12}	u_{22}	u_{32}	ω	Pt	u_{11}	u_{21}	u_{31}	u_{12}	u_{22}	u_{32}	ω
10.44	0.00	0.98	0.02	0.31	0.47	0.22	1	5.80	0.21	0.33	0.46	0.10	0.88	0.02	2
0.50	1.00	0.00	0.00	0.30	0.20	0.51	1	5.52	0.26	0.32	0.43	0.53	0.42	0.05	2
9.70	0.00	0.01	0.99	0.30	0.50	0.20	1	6.79	0.10	0.35	0.54	0.14	0.81	0.05	2
0.73	0.99	0.00	0.00	0.29	0.19	0.52	1	6.01	0.18	0.34	0.48	0.00	1.00	0.00	2
8.19	0.02	0.31	0.67	0.27	0.59	0.15	1	4.49	0.45	0.24	0.31	0.46	0.05	0.49	2
0.33	1.00	0.00	0.00	0.30	0.20	0.50	1	6.26	0.15	0.35	0.50	0.02	0.97	0.01	2
10.34	0.00	1.00	0.00	0.31	0.48	0.21	1	5.66	0.23	0.32	0.44	0.28	0.68	0.04	2
-0.85	0.98	0.01	0.01	0.31	0.22	0.46	1	4.92	0.36	0.28	0.36	0.98	0.01	0.01	2
10.32	0.00	1.00	0.00	0.31	0.48	0.21	1	5.40	0.28	0.31	0.42	0.73	0.23	0.05	2
0.53	1.00	0.00	0.00	0.30	0.20	0.51	1	4.87	0.37	0.27	0.36	0.95	0.02	0.03	2
9.98	0.00	0.52	0.48	0.31	0.49	0.21	1	6.40	0.14	0.35	0.51	0.06	0.93	0.02	2
-0.15	1.00	0.00	0.00	0.30	0.21	0.48	1	5.05	0.34	0.29	0.38	1.00	0.00	0.00	2
8.53	0.01	0.27	0.72	0.28	0.56	0.16	1	5.71	0.23	0.32	0.45	0.21	0.76	0.03	2
0.19	1.00	0.00	0.00	0.30	0.20	0.50	1	4.79	0.39	0.27	0.35	0.89	0.03	0.08	2
10.71	0.00	0.88	0.12	0.31	0.47	0.22	1	5.60	0.24	0.32	0.44	0.39	0.57	0.05	2
0.72	0.99	0.00	0.00	0.29	0.19	0.52	1	4.02	0.55	0.20	0.25	0.00	0.00	1.00	2
9.96	0.00	0.46	0.54	0.31	0.49	0.21	1	5.18	0.31	0.29	0.39	0.96	0.03	0.02	2
-0.85	0.98	0.01	0.01	0.31	0.22	0.46	1	4.04	0.54	0.20	0.26	0.01	0.00	0.99	2
10.94	0.00	0.81	0.19	0.31	0.46	0.23	1	6.46	0.13	0.35	0.52	0.07	0.91	0.02	2
0.24	1.00	0.00	0.00	0.30	0.20	0.50	1	6.10	0.17	0.34	0.49	0.00	1.00	0.00	2
10.59	0.00	0.92	0.08	0.31	0.47	0.22	1	5.32	0.29	0.30	0.41	0.84	0.12	0.04	2
0.26	1.00	0.00	0.00	0.30	0.20	0.50	1	3.71	0.61	0.17	0.21	0.04	0.01	0.95	2
9.60	0.00	0.00	1.00	0.30	0.50	0.20	1	5.26	0.30	0.30	0.40	0.90	0.07	0.03	2
-1.59	0.95	0.02	0.02	0.32	0.24	0.45	1	4.53	0.44	0.25	0.32	0.52	0.06	0.42	2
10.28	0.00	1.00	0.00	0.31	0.48	0.21	1	4.46	0.45	0.24	0.31	0.41	0.05	0.54	2
-1.49	0.96	0.02	0.02	0.31	0.23	0.45	1	5.82	0.21	0.33	0.46	0.08	0.90	0.01	2
8.75	0.01	0.24	0.75	0.28	0.54	0.17	1	4.64	0.42	0.25	0.33	0.70	0.05	0.24	2
0.15	1.00	0.00	0.00	0.30	0.21	0.49	1	2.91	0.77	0.10	0.13	0.18	0.08	0.74	2
11.99	0.01	0.66	0.33	0.32	0.44	0.24	1	4.65	0.41	0.26	0.33	0.72	0.05	0.22	2
0.69	0.99	0.00	0.00	0.29	0.19	0.52	1	5.84	0.21	0.33	0.46	0.07	0.91	0.01	2
11.21	0.01	0.75	0.24	0.32	0.45	0.23	1	5.08	0.33	0.29	0.38	1.00	0.00	0.00	2
0.51	1.00	0.00	0.00	0.30	0.20	0.51	1	6.29	0.15	0.35	0.50	0.03	0.96	0.01	2
10.47	0.00	0.97	0.03	0.31	0.47	0.22	1	6.58	0.12	0.35	0.53	0.10	0.87	0.03	2
0.09	1.00	0.00	0.00	0.30	0.21	0.49	1	2.32	0.87	0.06	0.07	0.24	0.12	0.64	2
9.38	0.00	0.07	0.93	0.30	0.51	0.19	1	5.61	0.24	0.32	0.44	0.37	0.59	0.05	2
-0.52	0.99	0.00	0.01	0.31	0.22	0.47	1	5.76	0.22	0.33	0.45	0.15	0.83	0.02	2
9.29	0.00	0.10	0.90	0.30	0.52	0.19	1	4.05	0.54	0.20	0.26	0.01	0.00	0.99	2
0.31	1.00	0.00	0.00	0.30	0.20	0.50	1	5.48	0.26	0.31	0.42	0.60	0.35	0.05	2
12.47	0.02	0.62	0.36	0.32	0.43	0.25	1	5.85	0.21	0.33	0.46	0.06	0.92	0.01	2
1.27	0.97	0.01	0.02	0.28	0.17	0.55	1	4.27	0.49	0.22	0.28	0.14	0.03	0.83	2
11.23	0.01	0.75	0.25	0.32	0.45	0.23	1	4.76	0.39	0.26	0.34	0.86	0.04	0.10	2
0.55	1.00	0.00	0.00	0.30	0.20	0.51	1	4.91	0.36	0.28	0.36	0.97	0.01	0.02	2
10.01	0.00	0.61	0.39	0.31	0.49	0.21	1	4.24	0.50	0.22	0.28	0.11	0.02	0.87	2
0.32	1.00	0.00	0.00	0.30	0.20	0.50	1	4.35	0.48	0.23	0.29	0.24	0.04	0.72	2
9.72	0.00	0.02	0.98	0.30	0.50	0.20	1	3.81	0.59	0.18	0.23	0.02	0.00	0.98	2
1.01	0.98	0.01	0.01	0.29	0.18	0.53	1	5.59	0.24	0.32	0.44	0.40	0.56	0.05	2
11.90	0.01	0.66	0.32	0.32	0.44	0.24	1	4.26	0.50	0.22	0.28	0.12	0.02	0.86	2
0.73	0.99	0.00	0.00	0.29	0.19	0.52	1	7.00	0.08	0.35	0.56	0.17	0.76	0.07	2
9.95	0.00	0.45	0.55	0.31	0.49	0.21	1	5.30	0.29	0.30	0.40	0.86	0.10	0.03	2
0.68	0.99	0.00	0.00	0.29	0.19	0.52	1	5.75	0.22	0.33	0.45	0.16	0.82	0.03	2

Table 31: CLc (c=3) design results using MLP (v_{11}=0.10, v_{21}=9.63, v_{31}=10.31, v_{12}=3.96, v_{22}=5.03, v_{32}=6.06)

Pt	u_{11}	u_{21}	u_{31}	u_{12}	u_{22}	u_{32}	ω	Pt	u_{11}	u_{21}	u_{31}	u_{12}	u_{22}	u_{32}	ω
10.16	0.00	0.92	0.08	0.31	0.48	0.21	1	5.83	0.21	0.33	0.46	0.08	0.91	0.01	2
1.45	0.96	0.02	0.02	0.28	0.17	0.56	1	5.09	0.33	0.29	0.38	0.99	0.00	0.00	2
10.60	0.00	0.92	0.08	0.31	0.47	0.22	1	3.77	0.60	0.18	0.22	0.02	0.01	0.97	2
0.92	0.99	0.01	0.01	0.29	0.18	0.53	1	4.55	0.43	0.25	0.32	0.56	0.06	0.38	2
10.27	0.00	1.00	0.00	0.31	0.48	0.21	1	5.81	0.21	0.33	0.46	0.09	0.89	0.02	2
0.77	0.99	0.00	0.00	0.29	0.19	0.52	1	6.04	0.18	0.34	0.48	0.00	1.00	0.00	2
8.99	0.00	0.19	0.81	0.29	0.53	0.18	1	3.39	0.68	0.14	0.18	0.10	0.04	0.86	2
0.91	0.99	0.01	0.01	0.29	0.18	0.53	1	4.85	0.38	0.27	0.35	0.94	0.02	0.04	2
8.34	0.02	0.29	0.69	0.27	0.57	0.16	1	5.35	0.28	0.31	0.41	0.79	0.16	0.04	2
1.28	0.97	0.01	0.02	0.28	0.17	0.55	1	4.76	0.39	0.26	0.34	0.86	0.04	0.10	2
10.92	0.00	0.82	0.18	0.31	0.46	0.22	1	4.29	0.49	0.22	0.29	0.16	0.03	0.82	2
1.06	0.98	0.01	0.01	0.29	0.18	0.53	1	6.09	0.18	0.34	0.48	0.00	1.00	0.00	2
11.19	0.00	0.75	0.24	0.32	0.46	0.23	1	2.96	0.76	0.11	0.13	0.17	0.08	0.75	2
1.75	0.93	0.03	0.04	0.26	0.15	0.58	1	4.06	0.54	0.20	0.26	0.01	0.00	0.99	2
10.86	0.00	0.83	0.17	0.31	0.46	0.22	1	5.91	0.20	0.33	0.47	0.03	0.97	0.01	2
0.96	0.99	0.01	0.01	0.29	0.18	0.53	1	5.93	0.20	0.33	0.47	0.02	0.98	0.00	2
10.84	0.00	0.84	0.16	0.31	0.46	0.22	1	5.35	0.28	0.31	0.41	0.79	0.16	0.04	2
1.72	0.94	0.03	0.03	0.27	0.15	0.58	1	5.14	0.32	0.29	0.39	0.98	0.01	0.01	2
10.49	0.00	0.96	0.04	0.31	0.47	0.22	1	4.16	0.52	0.21	0.27	0.05	0.01	0.94	2
1.18	0.98	0.01	0.01	0.28	0.18	0.54	1	3.54	0.65	0.16	0.19	0.07	0.03	0.90	2
9.75	0.00	0.05	0.95	0.30	0.50	0.20	1	4.43	0.46	0.24	0.30	0.36	0.05	0.59	2
1.12	0.98	0.01	0.01	0.28	0.18	0.54	1	5.37	0.28	0.31	0.41	0.77	0.19	0.04	2
10.18	0.00	0.94	0.06	0.31	0.48	0.21	1	5.75	0.22	0.33	0.45	0.16	0.81	0.03	2
0.21	1.00	0.00	0.00	0.30	0.20	0.50	1	6.23	0.16	0.34	0.50	0.02	0.98	0.01	2
9.72	0.00	0.02	0.98	0.30	0.50	0.20	1	4.05	0.54	0.20	0.26	0.01	0.00	0.99	2
-0.21	1.00	0.00	0.00	0.31	0.21	0.48	1	3.96	0.56	0.20	0.25	0.00	0.00	1.00	2
9.37	0.00	0.07	0.93	0.30	0.51	0.19	1	4.56	0.43	0.25	0.32	0.59	0.06	0.36	2
0.02	1.00	0.00	0.00	0.30	0.21	0.49	1	5.23	0.30	0.30	0.40	0.93	0.05	0.02	2
10.46	0.00	0.97	0.03	0.31	0.47	0.22	1	4.69	0.41	0.26	0.33	0.78	0.05	0.17	2
0.52	1.00	0.00	0.00	0.30	0.20	0.51	1	6.20	0.16	0.34	0.49	0.01	0.98	0.00	2
8.14	0.02	0.31	0.66	0.26	0.59	0.15	1	6.54	0.12	0.35	0.52	0.09	0.88	0.03	2
1.37	0.96	0.02	0.02	0.28	0.17	0.55	1	3.14	0.73	0.12	0.15	0.15	0.06	0.79	2
11.78	0.01	0.67	0.31	0.32	0.44	0.24	1	4.29	0.49	0.22	0.29	0.15	0.03	0.82	2
1.29	0.97	0.01	0.02	0.28	0.17	0.55	1	4.17	0.51	0.21	0.27	0.05	0.01	0.94	2
10.56	0.00	0.93	0.07	0.31	0.47	0.22	1	4.34	0.48	0.23	0.29	0.23	0.04	0.74	2
-0.48	0.99	0.00	0.00	0.31	0.22	0.47	1	3.72	0.61	0.17	0.22	0.03	0.01	0.96	2
8.58	0.01	0.27	0.72	0.28	0.56	0.16	1	6.57	0.12	0.35	0.53	0.10	0.87	0.03	2
-1.49	0.96	0.02	0.02	0.31	0.23	0.45	1	4.82	0.38	0.27	0.35	0.92	0.03	0.06	2
10.23	0.00	0.98	0.02	0.31	0.48	0.21	1	4.16	0.52	0.21	0.27	0.05	0.01	0.94	2
0.89	0.99	0.01	0.01	0.29	0.19	0.53	1	6.00	0.19	0.34	0.48	0.00	0.99	0.00	2
11.14	0.00	0.77	0.23	0.32	0.46	0.23	1	5.00	0.35	0.28	0.37	1.00	0.00	0.00	2
1.09	0.98	0.01	0.01	0.28	0.18	0.54	1	5.30	0.29	0.30	0.41	0.86	0.11	0.04	2
8.80	0.01	0.23	0.76	0.29	0.54	0.17	1	4.27	0.49	0.22	0.28	0.14	0.02	0.84	2
1.92	*0.92*	*0.04*	*0.05*	*0.26*	*0.15*	*0.60*	*1*	4.79	0.39	0.27	0.35	0.89	0.03	0.08	2
8.33	0.02	0.30	0.69	0.27	0.57	0.15	1	5.63	0.24	0.32	0.44	0.33	0.63	0.04	2
0.28	1.00	0.00	0.00	0.30	0.20	0.50	1	6.49	0.13	0.35	0.52	0.08	0.90	0.03	2
9.69	0.00	0.01	0.99	0.30	0.50	0.20	1	2.43	0.85	0.07	0.08	0.23	0.12	0.66	2
-1.43	0.96	0.02	0.02	0.31	0.23	0.45	1	3.53	0.65	0.16	0.19	0.08	0.03	0.90	2
10.91	0.00	0.82	0.18	0.31	0.46	0.22	1	4.27	0.49	0.22	0.28	0.14	0.02	0.84	2
1.81	0.93	0.03	0.04	0.26	0.15	0.59	1	6.51	0.13	0.35	0.52	0.08	0.89	0.03	2

Table 32: CLc (c=3) test results using MLP

7.3 Data Sets with Tarnished Gold Standards

In the experiments described in this section, 200 one-dimensional points were randomly selected from two different distributions. All points from the first distribution were assigned to ω_1 (N_1=100) with the remainder assigned to ω_2 (N_2=100). The design set was comprised of 50 ω_1 points and 50 ω_2 points (N_d=100) with the remaining points assigned to the test set (N_t=100). All performance results for both design and test sets are measured using the chance-corrected

measure of agreement, κ (section 3.2.3). Unlike the previous section, an MLP is used as the classifier. The MLP has one input PE, two output PEs, and two hidden layer PEs with the learning rate set to 0.7. The classifier is presented with the non-encoded data from the design set. The design set is then subjected to the robust gold standard reclassification described in section 6.2.1 and subsequently presented to the MLP. For each case, the test set is presented to the trained network and results recorded.

7.3.1 Robust reclassification and normal distributions

In this experiment, the ω_1 points were sampled from $N(0,1)$ while the ω_2 points were sampled from $N(3,1)$. Table 33 shows that when reclassification occurred perfect agreement was obtained using the design set as opposed to κ=0.90 with NE. A concomitant improvement was also obtained using the test set (Table 34).

N_d	NE		Robust	
	ω_1	ω_2	ω_1	ω_2
ω_1	47	3	49	0
ω_2	2	48	0	51
κ	0.90		1.00	

Table 33: Design set results

N_t	NE		Robust	
	ω_1	ω_2	ω_1	ω_2
ω_1	46	4	45	5
ω_2	4	46	2	48
κ	0.84		0.86	

Table 34: Test set results

Figure 120 shows that, with NE, any misclassifications with the design set occurred at the overlap of the pdfs of ω_1 and ω_2. Specifically, five points were misclassified: three ω_1 points, 2.11, 1.59, and 1.67; and two ω_2 points, 0.73 and 1.02. Figure 121 shows that eight test set misclassifications also occurred at the overlap: four ω_1 points, 2.13, 1.97, 1.85, and 2.16; and four ω_2 points, 1.05, 1.41, 0.81, and 1.43.

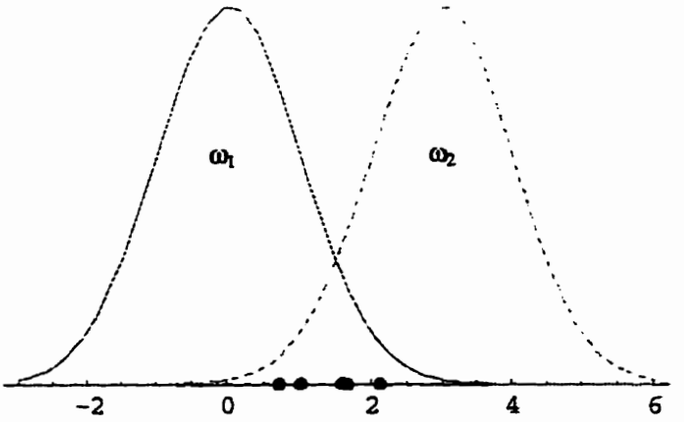

Figure 120: MLP non-encoded results using design set

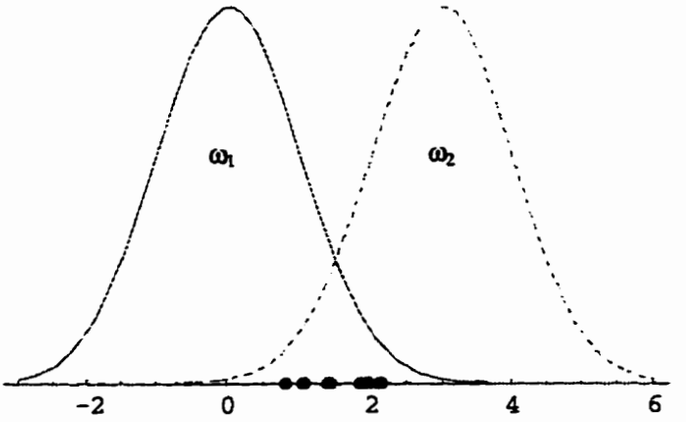

Figure 121: MLP non-encoded results using test set

Table 35 is a list of points in the design set, their class labels (ω), and their membership values for ω_1 (D_1) and ω_2 (D_2). The points that were misclassified by NE are shown in italics and points that were reclassified are shown in bold. In this case, all and only those points that were originally misclassified were reclassified. Of course, this, in general, is not the case. Figure 122 shows that no design points were misclassified when robust reclassification of the gold standard was employed.

Point	ω	D_1	D_2	Point	ω	D_1	D_2	Point	ω	D_1	D_2
-0.82	1	0.64	0	-0.34	1	0.92	0	3.67	2	0	0.63
-0.00	1	0.79	0	-0.07	1	0.84	0	2.79	2	0.24	0.76
1.23	1	0.40	0.34	-0.51	1	0.80	0	***0.73***	***2***	***0.50***	***0.30***
1.04	1	0.43	0.32	0.28	1	0.64	0	2.54	2	0.26	0.63
-0.11	1	0.86	0	-0.10	1	0.86	0	4.90	2	0	0.35
2.11	***1***	***0.30***	***0.50***	0.89	1	0.46	0.31	2.19	2	0.28	0.52
1.07	1	0.42	0.32	-0.26	1	0.99	0	***1.02***	***2***	***0.44***	***0.32***
-1.57	1	0.43	0	-2.27	1	0.33	0	2.96	2	0	0.87
1.40	1	0.37	0.37	-2.13	1	0.34	0	2.43	2	0.27	0.59
-0.95	1	0.59	0	0.63	1	0.52	0.28	3.52	2	0	0.70
-0.31	1	0.94	0	0.67	1	0.51	0.29	3.17	2	0	0.93
-0.25	1	0.99	0	-0.33	1	0.93	0	3.30	2	0	0.83
0.84	1	0.47	0.30	-0.65	1	0.71	0	4.17	2	0	0.48
-1.87	1	0.38	0	-0.51	1	0.79	0	4.73	2	0	0.38
0.76	1	0.49	0.29	0.50	1	0.56	0	1.81	2	0.32	0.43
-0.49	1	0.81	0	-0.94	1	0.59	0	2.82	2	0.24	0.78
-1.83	1	0.38	0	1.90	2	0.31	0.45	1.52	2	0.35	0.38
-0.61	1	0.74	0	3.35	2	0	0.80	3.04	2	0	0.94
-1.02	1	0.56	0	3.88	2	0	0.56	3.84	2	0	0.57
0.79	1	0.48	0.30	4.57	2	0	0.40	3.77	2	0	0.59
-1.55	1	0.43	0	2.81	2	0.24	0.77	1.97	2	0.30	0.46
-2.54	1	0.30	0	4.07	2	0	0.50	3.35	2	0	0.79
0.56	1	0.54	0	1.79	2	0.32	0.43	3.02	2	0	0.92
1.20	1	0.40	0.34	1.85	2	0.32	0.44	3.90	2	0	0.55
0.45	1	0.58	0	2.59	2	0.25	0.66	3.21	2	0	0.90
-2.26	1	0.33	0	2.72	2	0.25	0.72	3.86	2	0	0.56
1.59	***1***	***0.35***	***0.40***	4.71	2	0	0.38	3.22	2	0	0.89
1.67	***1***	***0.34***	***0.41***	2.66	2	0.25	0.69	3.97	2	0	0.53
-1.50	1	0.44	0	3.08	2	0	0.98	2.69	2	0.25	0.70
-1.38	1	0.47	0	3.12	2	0	0.98	3.95	2	0	0.54
0.52	1	0.55	0	4.01	2	0	0.52	2.78	2	0.24	0.75
-0.63	1	0.73	0	3.01	2	0	0.91	3.58	2	0	0.67
-0.99	1	0.57	0	2.33	2	0.27	0.56	4.24	2	0	0.46
0.60	1	0.53	0								

Table 35: Robust reclassification using the design set

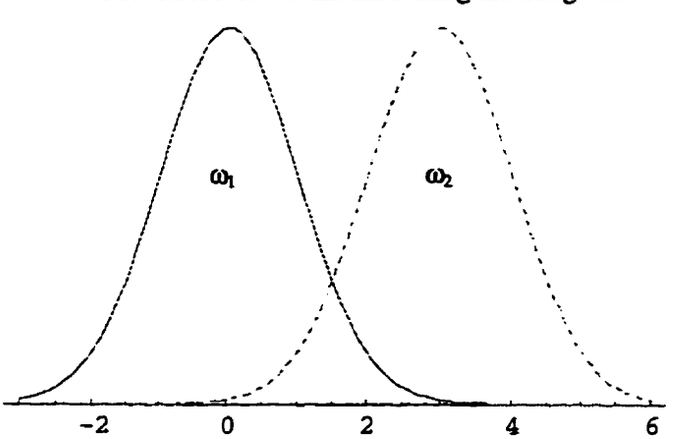

Figure 122: Design results for MLP with robust reclassification

Figure 123 shows that fewer points in the test set were misclassified, at the overlap of the pdfs. Specifically, seven points were misclassified: five ω_1 points, 1.41, 2.13, 1.97, 1.85, and 2.16; and two ω_2 points were misclassified, 1.05 and 0.81.

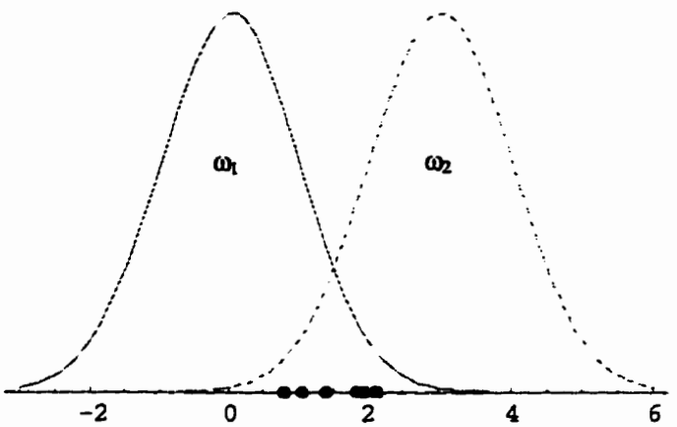

Figure 123: Test results for MLP with reclassified design points

Table 36 lists the points in the test set, their class label (ω), and the membership values for ω_1 (D_1) and ω_2 (D_2). Points in italics indicate they were misclassified whereas points in bold indicate that they would have been reclassified had they been in the design set. The last points needs to be emphasized; robust reclassification never alters the test set, to do so would be to ignore the relevance of the established gold standard. Nevertheless, it can be quite informative to, at least, flag points in the design set that are considered to be outliers or suspect points. Note that all points that were misclassified would have been reclassified.

Points	ω	D_1	D_2	Points	ω	D_1	D_2	Points	ω	D_1	D_2
0.64	1	0.54	0.29	-0.13	1	0.95	0	3.94	2	0	0.53
0.10	1	0.78	0	-0.94	1	0.56	0	***1.05***	***2***	***0.45***	***0.33***
0.15	1	0.75	0	0.44	1	0.61	0	2.35	2	0	0.58
-0.37	1	0.83	0	-0.70	1	0.65	0	2.38	2	0	0.59
-0.36	1	0.84	0	0.62	1	0.55	0.29	2.72	2	0	0.74
-0.79	1	0.62	0	-0.21	1	0.96	0	3.35	2	0	0.77
-0.25	1	0.93	0	-0.41	1	0.80	0	1.95	2	0.31	0.47
1.41	*1*	*0.39*	*0.38*	0.90	1	0.48	0.31	3.90	2	0	0.54
-0.15	1	0.97	0	-1.25	1	0.48	0	3.35	2	0	0.77
-1.44	1	0.44	0	0.53	1	0.58	0	4.64	2	0	0.38
0.64	1	0.54	0.29	0.67	1	0.53	0.29	1.41	2	0.38	0.37
0.80	1	0.50	0.30	-2.17	1	0.33	0	***0.81***	***2***	***0.50***	***0.31***
0.40	1	0.63	0	-0.51	1	0.74	0	2.09	2	0.30	0.50
2.13	***1***	***0***	***0.52***	1.22	1	0.41	0.35	3.25	2	0	0.83
1.97	***1***	***0.32***	***0.48***	-0.71	1	0.65	0	3.58	2	0	0.65
-1.37	1	0.45	0	-0.39	1	0.82	0	**1.43**	**2**	**0.38**	**0.38**
-1.28	1	0.47	0	3.74	2	0	0.59	2.49	2	0	0.63
1.07	1	0.44	0.33	3.05	2	0	0.98	1.94	2	0.31	0.47
-0.94	1	0.56	0	3.33	2	0	0.78	4.45	2	0	0.41
0.85	1	0.49	0.31	2.28	2	0	0.56	2.36	2	0	0.58
-0.47	1	0.76	0	2.74	2	0	0.75	3.74	2	0	0.59
-0.63	1	0.68	0	2.89	2	0	0.85	2.40	2	0	0.60
-0.67	1	0.66	0	4.73	2	0	0.37	2.92	2	0	0.87
1.85	***1***	***0.33***	***0.45***	2.39	2	0	0.59	4.12	2	0	0.48
2.16	***1***	***0***	***0.52***	3.39	2	0	0.75	3.29	2	0	0.81
0.12	1	0.76	0	3.56	2	0	0.66	3.85	2	0	0.56
0.38	1	0.63	0	2.97	2	0	0.91	3.09	2	0	0.97
-0.20	1	0.97	0	4.36	2	0	0.43	1.78	2	0.33	0.43
-0.59	1	0.70	0	4.32	2	0	0.44	3.10	2	0	0.96
-0.24	1	0.93	0	3.00	2	0	0.94	3.07	2	0	0.98
1.18	1	0.42	0.34	4.17	2	0	0.47	3.65	2	0	0.62
0.41	1	0.62	0	2.85	2	0	0.82	3.86	2	0	0.55
-0.84	1	0.59	0	2.06	2	0.30	0.49	1.45	2	0.37	0.38
-0.71	1	0.65	0								

Table 36: Robust distance measures for the test set

7.3.2 Robust reclassification with contamination

The data from the previous section is again used except that four ω_2 points from the design set, 3.95, 2.79, 3.59, and 4.25, have been relabeled as ω_1 points. This contamination significantly affects NE; κ for both design (Table 37) and test (Table 38) results degrade and misclassifications occur not only at the overlap of the pdfs of the two classes but also where the contamination occurred (Figure 124). The robust reclassification strategy is able to compensate for this contamination. NE misclassified nine points from the design set: six ω_1 points, 3.95, 2.79, 3.59, 4.25, 2.11, and 1.67; and three ω_2 points, 0.73, 1.02, and 1.52. All four mislabeled points were

misclassified. Nine points were misclassified by NE from the test set (Figure 125): four ω_1 points 2.13, 1.97, 1.85, and 2.16; and five ω_2 points, 1.05, 1.41, 0.81, 1.43, and 1.46.

N_d	NE		Robust	
	ω_1	ω_2	ω_1	ω_2
ω_1	48	6	49	0
ω_2	3	43	0	51
κ	0.82		1.00	

Table 37: Design set results

N_t	NE		Robust	
	ω_1	ω_2	ω_1	ω_2
ω_1	46	4	45	5
ω_2	5	45	2	48
κ	0.82		0.86	

Table 38: Test set results

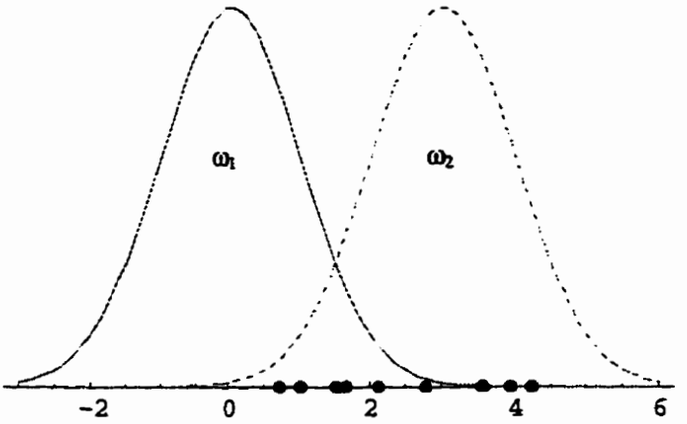

Figure 124: MLP NE design results using contaminated data

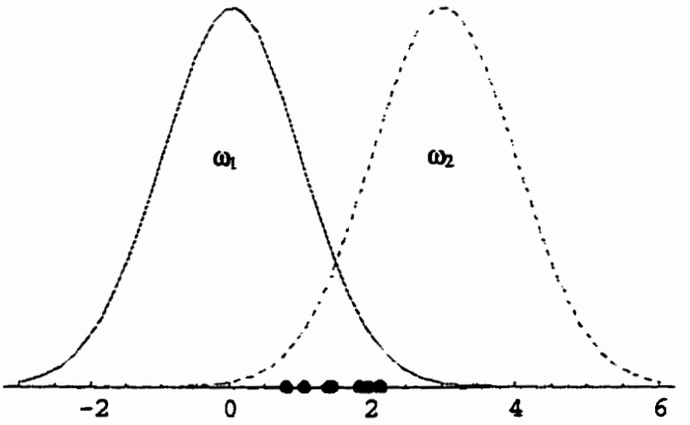

Figure 125: MLP test set with contamination

Table 39 lists the design points, their class label (ω), and the membership values for ω_1 (D_1) and ω_2 (D_2). Points in italics indicate that they have been misclassified whereas points in bold indicate that they were reclassified. Note that all mislabeled points have been reclassified.

Points	ω	D_1	D_2	Points	ω	D_1	D_2	Points	ω	D_1	D_2
3.95	***1***	***0***	***0.53***	0.52	1	0.61	0	3.12	2	0.23	0.94
2.79	***1***	***0.26***	***0.78***	-0.63	1	0.65	0	4.01	2	0	0.51
3.59	***1***	***0***	***0.66***	-0.99	1	0.52	0	3.01	2	0.24	0.94
4.25	***1***	***0***	***0.46***	0.60	1	0.58	0.28	2.33	2	0.29	0.57
-0.82	1	0.58	0	-0.34	1	0.80	0	3.67	2	0	0.62
-0.00	1	0.90	0	-0.07	1	0.97	0	2.79	2	0.25	0.78
1.23	1	0.42	0.35	-0.51	1	0.71	0	***0.73***	***2***	***0.55***	***0.30***
1.04	1	0.46	0.33	0.28	1	0.71	0	2.54	2	0.27	0.65
-0.11	1	0.99	0	-0.10	1	0.99	0	4.90	2	0	0.35
2.11	***1***	***0.31***	***0.51***	0.89	1	0.50	0.31	2.19	2	0.30	0.53
1.07	1	0.45	0.33	-0.26	1	0.86	0	***1.02***	***2***	***0.46***	***0.32***
-1.57	1	0.40	0	-2.27	1	0.31	0	2.96	2	0.24	0.90
1.40	1	0.39	0.37	-2.13	1	0.33	0	2.43	2	0.28	0.61
-0.95	1	0.54	0	0.63	1	0.57	0.29	3.52	2	0	0.68
-0.31	1	0.82	0	0.67	1	0.56	0.29	3.17	2	0	0.90
-0.25	1	0.86	0	-0.33	1	0.81	0	3.30	2	0	0.81
0.84	1	0.51	0.31	-0.65	1	0.64	0	4.17	2	0	0.47
-1.87	1	0.36	0	-0.51	1	0.71	0	4.73	2	0	0.37
0.76	1	0.53	0.30	0.50	1	0.61	0	1.81	2	0.34	0.44
-0.49	1	0.72	0	-0.94	1	0.54	0	2.82	2	0.25	0.80
-1.83	1	0.36	0	1.90	2	0.33	0.46	1.52	2	0.38	0.39
-0.61	1	0.66	0	3.35	2	0	0.77	3.04	2	0.24	0.98
-1.02	1	0.52	0	3.88	2	0	0.55	3.84	2	0	0.56
0.79	1	0.52	0.30	4.57	2	0	0.39	3.77	2	0	0.58
-1.55	1	0.40	0	2.81	2	0.25	0.79	1.97	2	0.32	0.47
-2.54	1	0.29	0	4.07	2	0	0.49	3.35	2	0	0.77
0.56	1	0.59	0	1.79	2	0.34	0.44	3.02	2	0.24	0.95
1.20	1	0.43	0.34	1.85	2	0.33	0.45	3.90	2	0	0.54
0.45	1	0.63	0	2.59	2	0.27	0.67	3.21	2	0	0.87
-2.26	1	0.31	0	2.72	2	0.26	0.74	3.86	2	0	0.55
1.59	***1***	***0.37***	***0.40***	4.71	2	0	0.37	3.22	2	0	0.86
1.67	***1***	***0.36***	***0.42***	2.66	2	0.26	0.71	3.97	2	0	0.52
-1.50	1	0.41	0	3.08	2	0.23	0.98	2.69	2	0.26	0.72
-1.38	1	0.43	0								

Table 39: Robust reclassification using the contaminated design set

Figure 126 shows that seven test set points were misclassified by MLP when robust reclassification was performed on the design set: five ω_1 points, 1.41, 2.13, 1.97, 1.85, and 2.16; and two ω_2 points, 1.05 and 0.81.

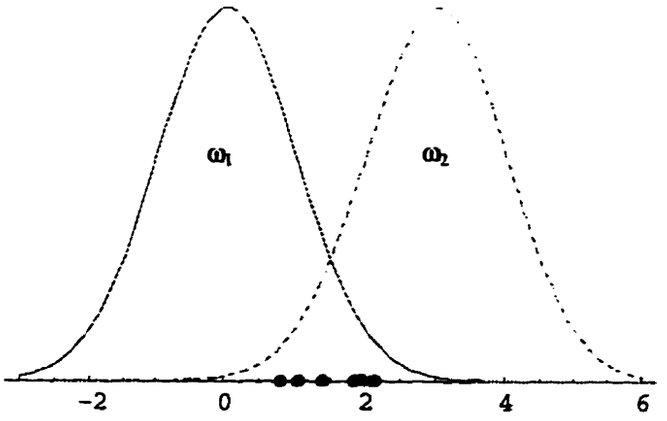

Figure 126: MLP robust results using test set

7.3.3 Fuzzy gold standard adjustment and normal distributions

In this experiment, the ω_1 points were sampled from $N(0,1)$ while the ω_2 points were sampled from $N(3,1)$ and p and q are both set to 2. FST gold standard adjustment was employed. Table 40 shows that NE and the encoded method produced identical design results but the encoded method produced slightly better results with the test set (Table 41). Both methods misclassified the same five design points: three ω_1 points, 2.11, 1.59, and 1.67; and two ω_2 points, 0.73 and 1.02.

N_d	NE		Fuzzy	
	ω_1	ω_2	ω_1	ω_2
ω_1	47	3	47	3
ω_2	2	48	2	48
κ	0.90		0.90	

Table 40: Design set results

N_t	NE		Fuzzy	
	ω_1	ω_2	ω_1	ω_2
ω_1	46	4	47	3
ω_2	4	46	3	47
κ	0.84		0.88	

Table 41: Test set results

Table 42 lists the design set points, their associated class label (ω), and the FST adjustment to the original gold standard (y_1' and y_2'). Rows in italics indicate points that were misclassified in the design set. Note that in all cases the adjusted gold standards are such that y_1' is identical or nearly identical to y_2' indicating that the associated point is nearly equidistant to the centroids of

both classes. The FST adjusted method misclassified six points in the design set: three ω_1 points, 2.13, 1.97, and 2.15; and three ω_2 points, 1.41, 0.81, 1.43.

Points	ω	y_1'	y_2'	Points	ω	y_1'	y_2'	Points	ω	y_1'	y_2'
-0.82	1	0.99	0.08	-0.34	1	0.99	0.12	3.67	2	0.16	0.98
-0.00	1	0.99	0.16	-0.07	1	0.99	0.15	2.79	2	0.32	0.99
1.23	1	0.86	0.55	-0.51	1	0.99	0.10	*0.73*	2	*0.96*	*0.96*
1.04	1	0.91	0.46	0.28	1	0.99	0.22	2.54	2	0.39	0.98
-0.11	1	0.99	0.15	-0.10	1	0.99	0.15	4.90	2	0.07	0.59
2.11	*1*	*0.92*	*0.92*	0.89	1	0.93	0.39	2.19	2	0.52	0.94
1.07	1	0.90	0.47	-0.26	1	1.00	0.13	*1.02*	*2*	*0.91*	*0.91*
-1.57	1	0.90	0.04	-2.27	1	0.69	0.02	2.96	2	0.28	0.99
1.40	1	0.81	0.64	-2.13	1	0.74	0.03	2.43	2	0.43	0.97
-0.95	1	0.99	0.07	0.63	1	0.97	0.30	3.52	2	0.18	0.99
-0.31	1	1.00	0.12	0.67	1	0.97	0.32	3.17	2	0.24	0.99
-0.25	1	1.00	0.13	-0.33	1	0.99	0.12	3.30	2	0.22	0.99
0.84	1	0.94	0.38	-0.65	1	0.99	0.09	4.17	2	0.11	0.89
-1.87	1	0.83	0.03	-0.51	1	0.99	0.10	4.73	2	0.08	0.67
0.76	1	0.95	0.34	0.50	1	0.98	0.27	1.81	2	0.67	0.82
-0.49	1	0.99	0.10	-0.94	1	0.99	0.07	2.82	2	0.31	0.99
-1.83	1	0.84	0.03	1.90	2	0.63	0.85	1.52	2	0.77	0.77
-0.61	1	0.99	0.09	3.35	2	0.21	0.99	3.04	2	0.26	0.99
-1.02	1	0.98	0.07	3.88	2	0.14	0.96	3.84	2	0.14	0.97
0.79	1	0.95	0.36	4.57	2	0.09	0.75	3.77	2	0.15	0.97
-1.55	1	0.91	0.04	2.81	2	0.32	0.99	1.97	2	0.60	0.88
-2.54	1	0.59	0.02	4.07	2	0.12	0.92	3.35	2	0.21	0.99
0.56	1	0.98	0.28	1.79	2	0.67	0.81	3.02	2	0.27	0.99
1.20	1	0.87	0.54	1.85	2	0.65	0.83	3.90	2	0.14	0.96
0.45	1	0.98	0.25	2.59	2	0.38	0.99	3.21	2	0.23	0.99
-2.26	1	0.69	0.02	2.72	2	0.34	0.99	3.86	2	0.14	0.96
1.59	*1*	*0.75*	*0.73*	4.71	2	0.08	0.68	3.22	2	0.23	0.99
1.67	*1*	*0.76*	*0.76*	2.66	2	0.36	0.99	3.97	2	0.13	0.94
-1.50	1	0.92	0.04	3.08	2	0.26	1.00	2.69	2	0.35	0.99
-1.38	1	0.94	0.05	3.12	2	0.25	1.00	3.95	2	0.13	0.95
0.52	1	0.98	0.27	4.01	2	0.13	0.94	2.78	2	0.32	0.99
-0.63	1	0.99	0.09	3.01	2	0.27	0.99	3.58	2	0.17	0.99
-0.99	1	0.98	0.07	2.33	2	0.46	0.96	4.24	2	0.11	0.87
0.60	1	0.97	0.29								

Table 42: FST GS Adjustment of the design set

7.3.4 Normal Distributions with Contamination

The contaminated data from section 7.3.2 is again used in this experiment. NE misclassifed nine points from the design set (Table 43): five ω_1 points, 3.95, 2.79, 3.59, 4.25, 2.11, and 1.67; and three ω_2 points, 0.73, 1.02, and 1.52. Note that all mislabeled points were misclassified. NE misclassified nine points from the test set (Table 44): four ω_1 points, 2.13, 1.97, 1.85, and 2.16; and five ω_2 points, 1.05, 1.41, 0.81, 1.43, and 1.46.

N_d	NE		Fuzzy	
	ω_1	ω_2	ω_1	ω_2
ω_1	48	6	48	6
ω_2	3	43	3	43
κ	0.82		0.82	

Table 43: Design set results

N_t	NE		Fuzzy	
	ω_1	ω_2	ω_1	ω_2
ω_1	46	4	47	3
ω_2	5	45	3	47
κ	0.82		0.88	

Table 44: Test set results

Table 45 lists the points in the design set, their class labels (ω), and their FST adjusted gold standard. Rows in italics indicate points that were misclassified in the design set. Specifically, nine points were misclassified: six ω_1 points, 3.95, 2.79, 3.59, 4.25, 2.11, and 1.67; and three ω_2 points, 0.73, 1.02, and 1.52. Note that all four mislabeled points were misclassified. Also note that, for all four mislabeled points, y_1' is identical to y_2'. Recall that this method will not reclassify a point in the design set. In these cases, y_1' was assigned the value y_2': all of which were near one, clearly indicating that they were much nearer to the ω_2 centroid than to the ω_1 centroid.

Six test set points were misclassified: three ω_1 points, 2.13, 1.97, and 2.16; and three ω_2 points, 0.81, 1.43, and 1.46.

Points	ω	y_1'	y_2'	Points	ω	y_1'	y_2'	Points	ω	y_1'	y_2'
3.95	*1*	*0.95*	*0.95*	0.52	1	0.99	0.28	3.12	2	0.31	0.99
2.79	*1*	*1.00*	*1.00*	-0.63	1	0.99	0.10	4.01	2	0.16	0.93
3.59	*1*	*0.99*	*0.99*	-0.99	1	0.97	0.07	3.01	2	0.33	0.99
4.25	*1*	*0.86*	*0.86*	0.60	1	0.99	0.31	2.33	2	0.56	0.97
-0.82	1	0.99	0.08	-0.34	1	0.99	0.12	3.67	2	0.20	0.98
-0.00	1	0.99	0.17	-0.07	1	1.00	0.16	2.79	2	0.39	0.99
1.23	1	0.91	0.57	-0.51	1	0.99	0.11	*0.73*	*2*	*0.98*	*0.98*
1.04	1	0.94	0.48	0.28	1	0.99	0.22	2.54	2	0.48	0.99
-0.11	1	1.00	0.15	-0.10	1	1.00	0.15	4.90	2	0.09	0.57
2.11	*1*	*0.93*	*0.93*	0.89	1	0.96	0.41	2.19	2	0.61	0.94
1.07	1	0.94	0.49	-0.26	1	0.99	0.13	*1.02*	*2*	*0.95*	*0.95*
-1.57	1	0.88	0.04	-2.27	1	0.66	0.02	2.96	2	0.35	0.99
1.40	1	0.87	0.66	-2.13	1	0.71	0.03	2.43	2	0.52	0.98
-0.95	1	0.98	0.07	0.63	1	0.98	0.32	3.52	2	0.23	0.99
-0.31	1	0.99	0.13	0.67	1	0.98	0.33	3.17	2	0.29	0.99
-0.25	1	0.99	0.13	-0.33	1	0.99	0.12	3.30	2	0.27	0.99
0.84	1	0.97	0.39	-0.65	1	0.99	0.09	4.17	2	0.14	0.88
-1.87	1	0.80	0.03	-0.51	1	0.99	0.11	4.73	2	0.10	0.65
0.76	1	0.98	0.36	0.50	1	0.99	0.28	1.81	2	0.75	0.83
-0.49	1	0.99	0.11	-0.94	1	0.98	0.07	2.82	2	0.39	0.99
-1.83	1	0.81	0.03	1.90	2	0.72	0.86	*1.52*	*2*	*0.84*	*0.84*
-0.61	1	0.99	0.10	3.35	2	0.26	0.99	3.04	2	0.33	1.00
-1.02	1	0.97	0.07	3.88	2	0.17	0.95	3.84	2	0.18	0.96
0.79	1	0.97	0.37	4.57	2	0.11	0.73	3.77	2	0.19	0.97
-1.55	1	0.88	0.04	2.81	2	0.39	0.99	1.97	2	0.69	0.89
-2.54	1	0.56	0.02	4.07	2	0.15	0.91	3.35	2	0.26	0.99
0.56	1	0.99	0.29	1.79	2	0.75	0.83	3.02	2	0.33	0.99
1.20	1	0.91	0.56	1.85	2	0.73	0.85	3.90	2	0.17	0.95
0.45	1	0.99	0.26	2.59	2	0.46	0.99	3.21	2	0.29	0.99
-2.26	1	0.66	0.02	2.72	2	0.42	0.99	3.86	2	0.18	0.96
1.59	1	0.82	0.74	4.71	2	0.10	0.67	3.22	2	0.28	0.99
1.67	*1*	*0.80*	*0.78*	2.66	2	0.44	0.99	3.97	2	0.16	0.94
-1.50	1	0.90	0.05	3.08	2	0.32	1.00	2.69	2	0.43	0.99
-1.38	1	0.92	0.05								

Table 45: FST GS adjustment results for contaminated data

7.4 Fuzzy Interquartile Encoded Multi-Layer Perceptron

The n-dimensional bounding set (see section 3.3.1) is used here to experimentally justify the efficacy of the fuzzy encoding preprocessing strategy. Recall that $2n$ hyperplanes are required as an accurate decision boundary. In the case of an MLP classifier, this translates into the requirement that at least $2n$ PEs in a hidden layer are needed where each PE corresponds to one of the hyperplanes. Figure 127 illustrates the weights and biases for an n-dimensional MLP solution.

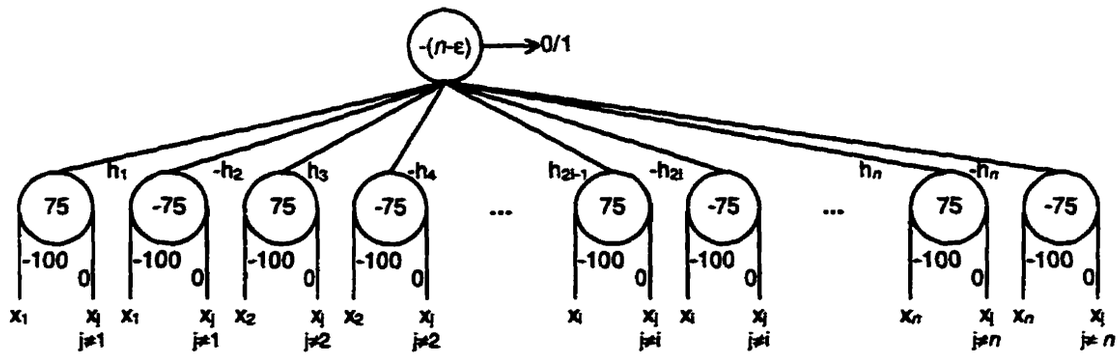

Figure 127: An ideal n-dimensional MLP solution

Figure 10 suggests that the ideal solution for the n-dimensional problem requires exactly $2n$ hyperplanes. If a step function

$$\gamma(x) = \begin{cases} 1 \text{ if } x > 0 \\ 0 \text{ if } x \leq 0 \end{cases} \tag{89}$$

was used as the transfer function then the solution is straightforward. For each dimension, i, we have a pair of hidden PEs corresponding to the pair of hyperplanes used for that dimension. The weights for the corresponding coordinate, x_i, are set to 1. The weights are set to 0 for the remaining features. The weight value between the first PE and the output node is 1 and -1 for the second. The bias for the first PE is 0.75 and -0.75 for the second. Finally, the bias for the output PE is $-(n-\varepsilon)$, where ε is a small positive real. If x_i is bounded by the corresponding hyperplanes then the summation of the pair of PEs is large, otherwise, it tends towards zero. If all features, $x_1, x_2, \ldots, x_n$, are bounded by their respective hyperplanes then the summation of the outputs of the $2n$ hyperplanes is large. Figure 128 illustrates the solution to the 2-dimensional boundary problem shown in Figure 129. Figure 130 and Figure 131 illustrate the 3- and 4-dimensional solutions.

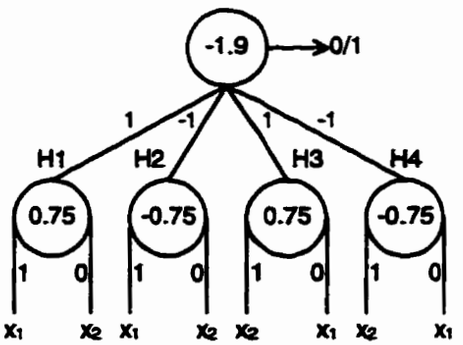

Figure 128: An ideal 2D solution (step function or logistic function with gain)

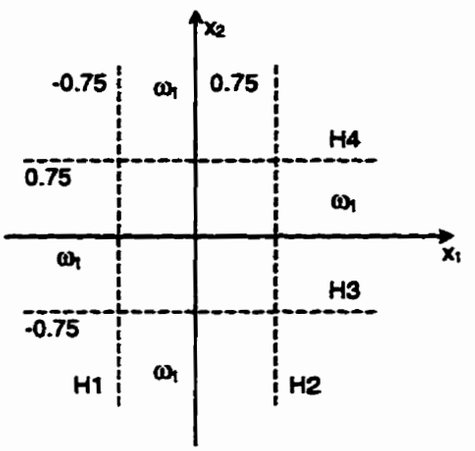

Figure 129: A geometrical interpretation of the 2-dimensional problem

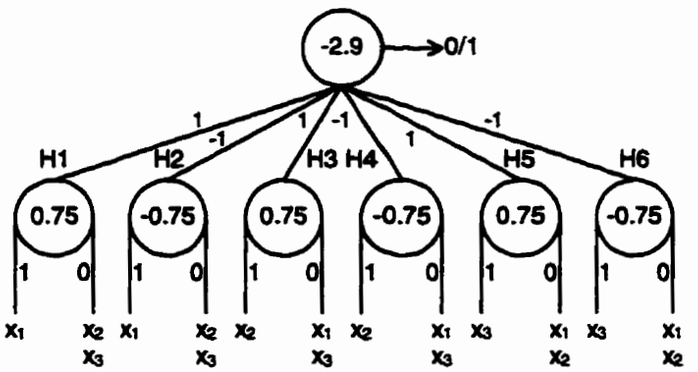

Figure 130: An ideal 3D solution (step function or logistic function with gain)

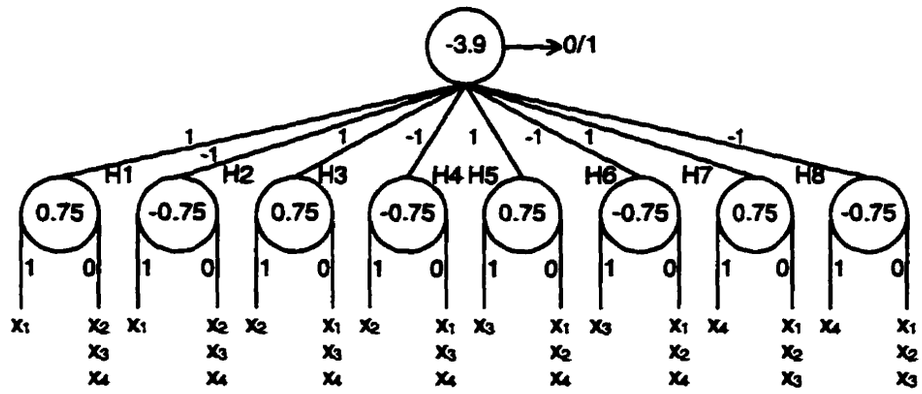

Figure 131: An ideal 4D solution (step function or logistic function with gain)

Of course, an MLP cannot use the step function as a transfer function because the gradient descent strategy requires a differentiable transfer function. Moreover, because the logistic function produces continuous values between 0 and 1, it smoothes the output values instead of providing a discrete, non-continuous jump from 0 to 1. The smoothing nature of the sigmoid tends to affect the results such that data points near the boundaries become misclassified. One way to compensate for this is to use a gain term with the logistic function. As the gain term approaches infinity, the logistic function tends towards a step function. It was experimentally determined that if the gain term is set to 80 the same weight and bias values used with the step function would work with the logistic function. Unfortunately, such a large gain term usually causes the MLP to wildly oscillate so this strategy is of little use.

However, if the logistic function is used without any gain (g=1), an ideal solution may still be obtained if the bias values are changed for the hidden PEs and the weights from the input values to them. In fact, the larger values (two orders of magnitude) tend to produce the same results as those where a large gain term is used. The advantage, though, is that this approach does not tend to cause wild oscillations. Figure 132, Figure 133, and Figure 134, illustrate the weights and biases for 2-, 3-, and 4-dimensional solutions using this strategy.

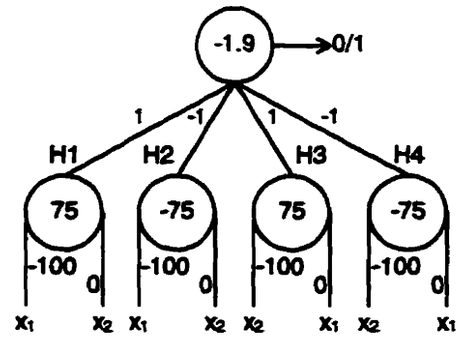

Figure 132: An ideal 2D solution (the logistic function with no gain)

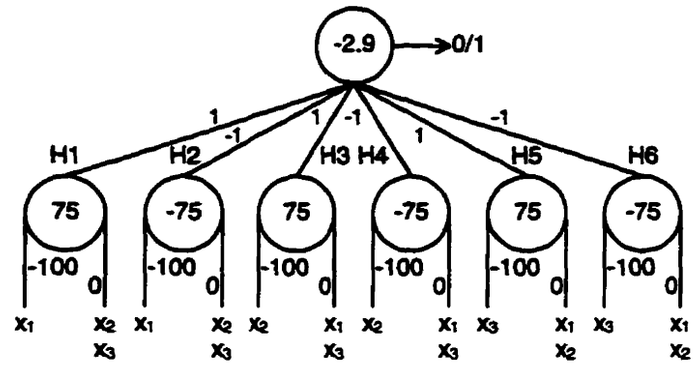

Figure 133: An ideal 3D solution (logistic function with no gain)

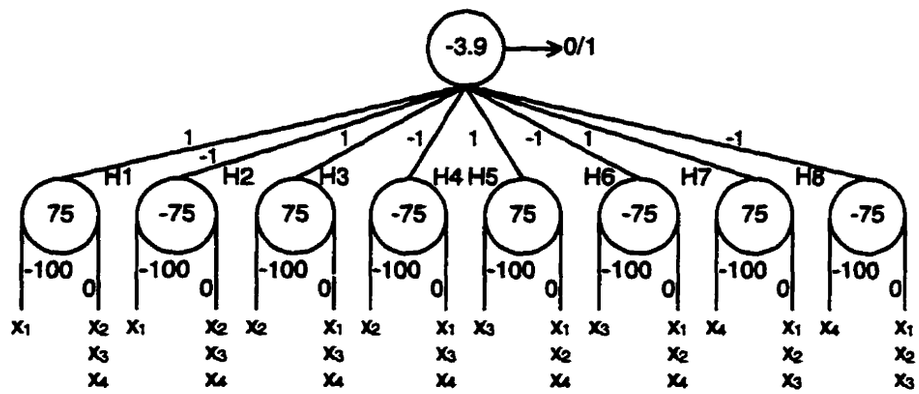

Figure 134: An ideal 4D solution (logistic function with no gain)

However, in practice a MLP may not find these hyperplanes. Figure 135 illustrates a sub-optimal solution for the 2-dimensional problem using three lines. In this case, one of two events will have occurred: one of the hidden PEs will have weights that are similar to one of the other three PEs in the hidden layer (in which case it will duplicate the functionality of the other PE); or, the weights of one of the PEs are near zero in which case it contributes negligibly to the outcome. It should be noted that even when only three hyperplanes are used, a MLP might converge to a

point where a majority of the vectors will be correctly classified. However, this benefit may also be considered a disadvantage — when it begins to converge to a solution, a MLP is not able to escape from the associated local minimum to determine if better solutions exist. This is a result of the gradient descent strategy — the error cannot increase, thus when the algorithm begins to converge towards a solution it cannot diverge from it.

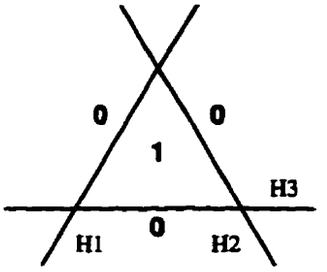

Figure 135: A non-ideal solution

The data range for the classification problem is [-1,1] and is discretized in intervals of 0.1. Apart from ensuring that vectors were randomly selected from the entire pool, the overriding constraint was to ensure that there was an equal number of class 0 and class 1 vectors in the design sets. Another constraint was to attempt to select approximately 2/3 of the total number of vectors for the design set. As the dimensionality of the problem increases, this constraint begins to conflict with the one ensuring an equal number of vectors from each class. The following strategy was used in order to maximally satisfy these two constraints. For each case, 2/3 of the vectors from the class with the fewer number of vectors was randomly selected for inclusion in the design set. The same number of vectors were then randomly selected from the other class for inclusion in the design set. The remaining vectors were then used as test vectors. Table 46, Table 47, and Table 48 list the number of vectors used for the 2-, 3-, and 4-dimensional cases, respectively, as well as their classification and how many were used for the design and test sets. In the interest of achieving convergence in a reasonable period of time, the same number and breakdown of vectors was used in the 20-dimensional case as in the 4-dimensional case. All the experiments discussed in this section used the same MLP architecture. The learning rate was set

at 0.9 and no momentum term was used. The transfer function is the logistic function and the learning rule is the generalized delta rule. As data were carefully generated for this paper, they were neither scaled, normalized, nor was any noise introduced into the MLPs. For each specific n-dimensional problem, one hidden layer was used that contained $2n$ PEs. After some initial trials, the number of iterations was fixed for each set of experiments in order to more accurately compare the performance of the MLP using NE data versus the corresponding MLP using FE data. Finally, each pair of NE and FE runs used the same set of initial randomized weights. Four triangular fuzzy sets were selected at intervals of [-1,-0.5], [-0.5,0], [0,0.5], and [0.5,1], respectively. The membership functions were computed to be

$$\begin{aligned} f_1(x) &= 0 \vee (1-2|x+.75|) \\ f_2(x) &= 0 \vee (1-2|x+.25|) \\ f_3(x) &= 0 \vee (1-2|x-.25|) \\ f_4(x) &= 0 \vee (1-2|x-.75|) \end{aligned} \tag{90}$$

Additional runs were made using eight triangular fuzzy sets for each input value by simply splitting the original boundaries in half.

	Design (%)	Test (%)	Total (%)
Class 0	144 (67)	72 (33)	216 (49)
Class 1	144 (64)	81 (36)	225 (51)
Total	288 (65)	153 (35)	441

Table 46: Vectors in the 2-dimensional case

	Design (%)	Test (%)	Total (%)
Class 0	2250 (38)	3636 (62)	5886 (64)
Class 1	2250 (67)	1125 (33)	3375 (36)
Total	4500 (49)	4761 (51)	9261

Table 47: Vectors in the 3-dimensional case

	Design (%)	Test (%)	Total (%)
Class 0	33750 (23)	110106 (77)	143856 (74)
Class 1	33750 (67)	1125 (33)	50625 (26)
Total	67500 (35)	126981 (65)	194481

Table 48: Vectors in the 4-dimensional case

For each 2-, 3-, 4-, and 20-dimensional case, 100 design and test sets were generated in order to provide a more statistically accurate set of observations. Each set was then fuzzy encoded and

paired with its corresponding NE set. The generated data were neither scaled nor normalized. For each specific n-dimensional problem, one hidden layer was used that contained $2n$ PEs. After some initial trials, the number of iterations was fixed for each set of experiments in order to more accurately compare the performance of a MLP using NE data versus the corresponding MLP using FE data. After the training phase terminated, the test sets were classified and the performance results were recorded.

In all cases, the FE MLPs that used four fuzzy sets attained their κ values with an iteration count of roughly an order of magnitude less than their NE counterparts. Moreover, when eight fuzzy sets were used an additional order of magnitude reduction in the number of iterations was achieved. These significant reductions do not precisely translate to corresponding increases in speed because there are roughly four times the number of computations that have to be performed for the FE MLPs using four fuzzy sets (eight times for the FE MLPs using eight fuzzy sets). Nevertheless, taking this fact into account, the FE MLPs performance were still many times better. It should also be noted that when eight fuzzy sets were used the FE MLPs were somewhat sensitive to overtraining. That is, as the iteration count increased, their performance with respect to classification success was slightly degraded. Table 49i clearly indicates that the FE MLPs outperformed their NE MLPs counterparts for the 2-, 3-, 4-, and 20-dimensional cases.

In the following discussion, representative experiment pairs were selected from each case. The weights were recorded for subsequent analysis. The ensuing sections will clearly demonstrate that FE data does improve the performance of MLPs. Not only are the results consistently better for every pair of experiments, but the MLPs that used FE data also produced these superior results in far fewer iterations.

	NE(κ)	Iters	FE4(κ)	Iters	FE8(κ)	Iters
i) 2 dimensions	0.86	300	1.00	50	1.00	5
3 dimensions	0.83	600	1.00	100	0.99	10
4 dimensions	0.90	2000	1.00	200	0.99	100
20 dimensions	0.85	5000	0.98	500	0.98	200
ii) 2D Noise (5%)	1.00	400	0.99	90	0.99	9
Noise(10%)	0.99	400	0.99	90	0.99	9
Noise(20%)	0.81	400	0.99	90	0.98	9
Noise(30%)	0.92	1500	0.92	400	0.97	50
Noise(40%)	0.88	1500	0.90	400	0.90	50
iii) 2D Bimodal I	0.39	2000	0.97	60	0.99	5
Bimodal II	0.95	2000	0.92	60	0.98	5
Skewed I	0.80	2000	1.00	60	0.97	5
Skewed II	0.87	2000	1.00	60	0.97	5
(FE4=fuzzy-encoded data using 4 fuzzy sets, NE=non-encoded data FE8=fuzzy-encoded data using 8 fuzzy sets, Iters=number of iterations (x1,000))						

Table 49: Classification results averaged over 100 runs

7.4.1 The 2-dimensional case

In the 2-dimensional case, the NE version of experiment 87 (Figure 136) that yielded perfect classifications, is very similar in structure to the MLP found in Figure 132. That is, the relative magnitudes are similar and the signs identical for each respective weight and bias value. This suggests that each hidden PE corresponds to a unique and significant hyperplane. The NE version of experiment 31 (Figure 137) produced an accuracy rate of 86%. Note that the PE, H4 (shaded), contributes very little to the final outcome. In this case, only three hyperplanes are used thereby degrading overall performance. The NE version of experiment 23 (Figure 138) produced the poorest results which is to be expected since each hidden PE duplicates the functionality of the others and this implies that only one hyperplane is used. The NE version of experiment 8 (Figure 139) produced results only slightly worse than experiment 31 using only two hyperplanes. In the FE versions of all the experiments, perfect results were achieved (see Table 50). The structures of the FE MLPs suggest that the information content is more uniformly distributed than the NE MLP counterparts.

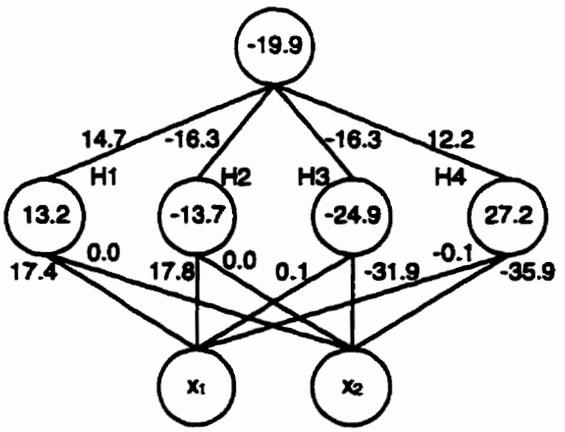

Figure 136: NE MLP with four hyperplanes

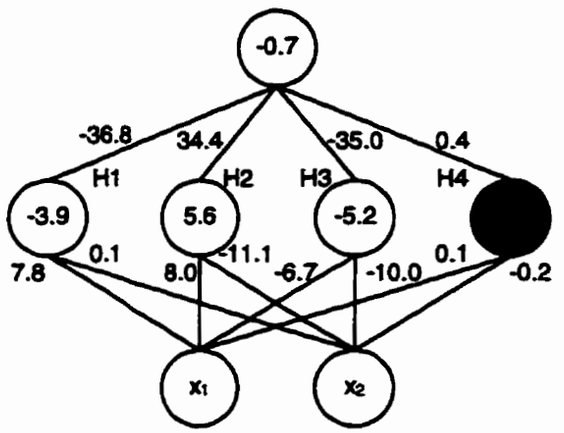

Figure 137: NE MLP with three hyperplanes

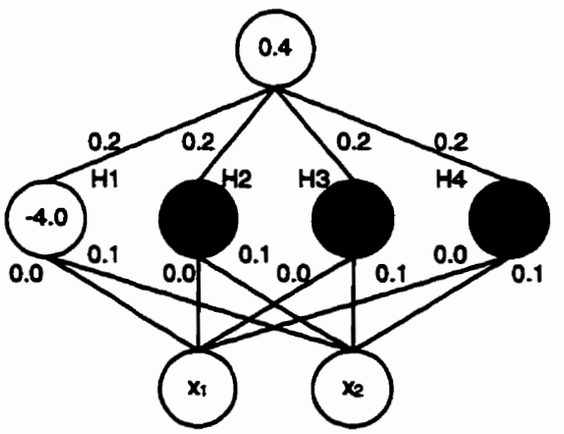

Figure 138: NE MLP with one hyperplane

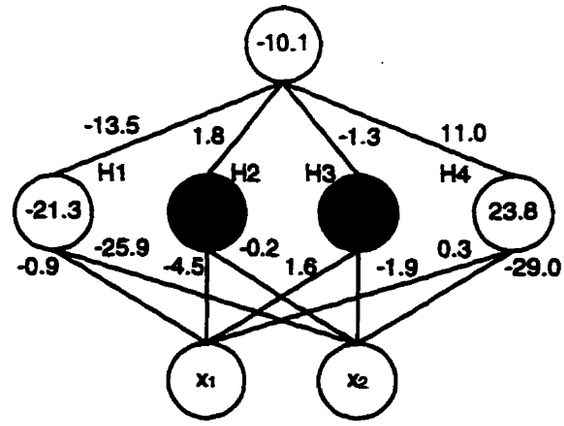

Figure 139: NE MLP with two hyperplanes

Accuracy	Class 0	Class 1	Total
Exp.8: NE(%)	43 (60)	81(100)	124 (81)
FE(%)	72(100)	81(100)	153(100)
Exp.23:NE	0 (0)	81(100)	81 (53)
FE	72(100)	81(100)	153(100)
Exp.31:NE	60 (83)	72 (89)	132 (86)
FE	72(100)	81(100)	153(100)
Exp.87:NE	72(100)	81(100)	153(100)
FE	72(100)	81(100)	153(100)
Total Vectors	72	81	153

Table 50: Sample 2D results — NE versus FE

7.4.2 The 3-dimensional case

In the 3-dimensional case, the NE version of experiment 19 (see Table 51), which produced a correct classification for all test vectors, is similar in structure to the MLP described in Figure 133. The hidden PEs, H2 and H3, represent the hyperplanes for the first coordinate, H1 and H6 represent the hyperplanes for the second coordinate and H4 and H5 represent the hyperplanes for the third coordinate. The NE versions of experiments 51 and 69 are quite similar in structure. The hidden PE pairs H1/H4 and H3/H6 correspond to the hyperplanes for the second and third coordinates, respectively, whereas H2 and H5 contribute far less to the final outcome. A testament to the robustness of MLPs can be found in these two runs: with only four of six hyperplanes, the MLP still achieved an accuracy rate of 83%. In the NE version of experiment 93, five of six hyperplanes are well defined: H1 and H5 for the second coordinate; H2 and H6 for the third coordinate; and H3 for the first coordinate. If this MLP ran for several thousand more

iterations, H4 would probably settle to correspond to the last required hyperplane. As in the 2-dimensional case, all FE runs had uniformly distributed values for weights and biases.

Accuracy	Class 0	Class 1	Total
Exp.19:NE(%)	3636(100)	1125(100)	4761(100)
FE(%)	3636(100)	1125(100)	4761(100)
Exp.51:NE	2812 (77)	1125(100)	3937 (83)
FE	3636(100)	1125(100)	4761(100)
Exp.69:NE	2812 (77)	1125(100)	3937 (83)
FE	3636(100)	1125(100)	4761(100)
Exp.93:NE	3297 (91)	863 (77)	4169 (87)
FE	3636(100)	1125(100)	4761(100)
Total Vectors	3636	1125	4761

Table 51: Sample 3D results — NE versus FE

7.4.3 The 4-dimensional case

In the NE version of experiment 28, good results were achieved after two million iterations (Table 52). All hyperplanes are evident and with more iterations it is suspected that the MLP would produce even better results. The FE versions of this experiment produced perfect results. Hidden PEs are paired as in Figure 134 (for example, H1 and H2 represent hyperplanes for the first feature – $f_i(x_j)$ (i=1, 2, 3, 4) is near 0 for $j \neq i$ and large, ≈100, for j=1) but neither the biases nor the weights to the output PE alternate their signs. This suggests that the value of a coordinate x_i need not be between its corresponding hyperplanes but rather it needs only be on one side of a single hyperplane (therefore we need only sum the outputs of all the hidden PEs and determine if the sum exceeds a threshold). This suggests that the dimensionality of the problem has been reduced. Specifically, the original 4-dimensional problem (8 hyperplanes) has been reduced to a 2-dimensional problem (4 hyperplanes) while still producing perfect classifications.

Accuracy	Class 0	Class 1	Total
Exp.28:NE(%)	99981 (91)	16875(100)	116856 (92)
FE(%)	110106(100)	16875(100)	126981(100)
Total Vectors	110106	16875	126981

Table 52: Sample 4D results — NE versus FE

7.4.4 Noisy data and non-normal distributions

Table 49ii lists performance results when varying amounts of Gaussian noise were added to the first coordinate of the 2-dimensional data sets. The FE MLPs produced comparable or more accurate classifications with far fewer iterations. It should also be noted however that NE MLPs tended to produce better results than their noise-free counterparts. This suggests that the introduction of noise is indeed a useful enhancement to MLPs.

The distribution of the design data in all of the previous experiments was uniform. Additional experiments were run for the 2-dimensional case to determine how well the two types of networks performed if the design data were not uniformly distributed. Design data were carefully reselected to ensure non-uniform distributions: two distinct bimodal distributions and two distinct skewed distributions. Results in Table 49iii indicate that FE MLPs again consistently outperformed NE MLPs and with far fewer iterations.

7.5 Additional Experiments

7.5.1 20-dimensional hypercube

A number of different classification systems are now used with the 20-dimensional hypercube data set described in section 3.3.1:

- LDA is linear discriminant analysis (section 4.1);
- MLP is a multi-layer perceptron (section 4.2) with 20 input PEs, 2 output PEs, and 40 hidden layer PEs;
- E-MLP is an enhanced MLP (section 4.2.1);
- RBFN is a radial basis function network (section 4.4) with 40 receptive fields;
- PNN is a probabilistic neural network (section 4.3) with 40 kernels.

The fuzzy encoding techniques all employ an underlying MLP identical to the one mentioned above except that there are different numbers of input PEs depending on the encoding method:

- DP (section 6.1.2) and DPc are the dimension-preserving fuzzy interquartile encoding method and its class-wise variant;
- IQ (section 6.1.1) and IQc are the interquartile encoding method and its class-wise variant;
- CL (section 6.1.3) and CLc are the fuzzy cluster encoding method and its class-wise variant (the number of clusters were varied from 2 to 20 and the best result is listed; respectively, c=7 and c=3).

The boundaries, averaged over all 20 dimensions, for IQ and DP are α=-1.00, Q_l=-0.50, m=0.03, Q_u=0.40, β=1.00. The boundaries, averaged over all 20 dimensions, for IQc and DPc are α_1=-0.70, Q_{l1}=-0.40, m_1=0.00, Q_{u1}=0.40, β_1=0.70 and α_2=-1.00, Q_{l2}=-0.70, m_2=-0.05, Q_{u2}=0.50, β_2=1.00.

Performance results for these methods using the test set are listed in Table 53. As this problem is not linearly separable, LDA fails to discriminate; specifically, LDA achieved a κ=0.00 by computing the hyperplane to fall directly through the hypercube. The non-encoded ANNs, on the other hand, perform adequately, especially RBFN. DP and IQ produced superior results whereas CL and CLc did not perform as well as the non-encoded ANNs.

N_t	LDA		MLP		E-MLP		RBFN		PNN	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	25	25	44	6	41	9	43	7	44	6
ω_2	25	25	14	36	10	40	6	44	14	36
κ	0.00		0.60		0.62		0.74		0.60	

N_t	DP		CL (c=7)		IQ	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	46	4	37	13	46	4
ω_2	4	46	8	42	4	46
κ	0.84		0.58		0.84	

N_t	DPc		CLc (c=3)		IQc	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	43	7	42	8	40	10
ω_2	7	43	16	34	34	16
κ	0.72		0.52		0.48	

Table 53: Test results using different classification systems with a hypercube

The above classification systems are again used with the 20-dimensional hypercube bounding problem except that 20% Gaussian noise is added to each coordinate. The boundaries, averaged over 20 dimensions, for IQ and DP are α=-1.18, Q_l=-0.48, m=0.03, Q_u=0.42, β=0.91. The boundaries, averaged over 20 dimensions, for IQc and DPc, are α_1=-0.90, Q_{l1}=-0.34, m_1=0.00, Q_{u1}=0.38, β_1=0.79 and α_2=-1.02, Q_{l2}=-0.63, m_2=-0.02, Q_{u2}=0.48, β_2=1.10.

Performance results using these methods with the test set are listed in Table 54. Performance results significantly degrade for all methods except for IQ that appears to be robust to the noise.

N_t	LDA		MLP		E-MLP		RBFN		PNN	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	23	27	35	15	35	15	38	12	36	14
ω_2	25	25	12	38	12	38	11	39	15	35
κ	-0.04		0.46		0.46		0.54		0.42	

N_t	DP		CL (c=6)		IQ	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	33	17	24	26	45	5
ω_2	5	45	17	33	6	44
κ	0.56		0.14		0.78	

N_t	DPc		CLc (c=4)		IQc	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	37	13	42	8	36	14
ω_2	17	33	16	34	15	35
κ	0.40		0.52		0.42	

Table 54: Test results using different classification systems with a hypercube with noise

7.5.2 Disk and torus

The classification systems described in section 7.5.1 are used with the disk/torus problem described in section 3.3.2. The only difference is that MLP and E-MLP have 4 hidden layer PEs, RBFN has 4 receptive fields, and PNN has 4 kernels. The boundaries for IQ and DP are α=[0.01, 0.00], Q_l=[0.23, 0.24], m=[0.43, 0.45], Q_u=[0.81, 0.70], β=[0.99, 0.98]. The boundaries for IQc and DPc are α_1=[0.01, 0.00], Q_{l1}=[0.10, 0.13], m_1=[0.63, 0.41], Q_{u1}=[0.86, 0.78], β_1=[0.89, 0.98] and α_2=[0.13, 0.15], Q_{l2}=[0.31, 0.35], m_2=[0.42, 0.47], Q_{u2}=[0.69, 0.63], β_2=[0.89, 0.89]. The three cluster centres for CL are v_1=[0.81, 0.35], v_2=[0.36, 0.79], v_3=[0.25, 0.31]. The four cluster centres for CLc are v_{11}=[0.18, 0.65], v_{21}=[0.80, 0.29], v_{12}=[0.36, 0.38], v_{22}=[0.60, 0.60].

Performance results using these methods with the test set are listed in Table 55. Performance results significantly degrade for all methods except for IQ that appears to be robust to the noise. Note that RBFN was the only non-encoded ANN to perform reasonably well with the disk/torus data set; a single receptive field at the centre of the unit circular disk is all that would be required to get good classification performance. As a whole, the fuzzy encoding methods performed extremely well.

N_t	LDA		MLP		E-MLP		RBFN		PNN	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	36	14	37	13	32	18	42	8	31	19
ω_2	32	18	8	42	12	38	2	48	12	38
κ	0.08		0.58		0.40		0.80		0.38	

N_t	DP		CL (c=3)		IQ	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	47	3	48	2	46	4
ω_2	2	48	18	32	0	50
κ	0.90		0.60		0.92	

N_t	DPc		CLc (c=2)		IQc	
	ω_1	ω_2	ω_1	ω_2	ω_1	ω_2
ω_1	45	5	47	3	38	12
ω_2	0	50	1	49	14	36
κ	0.90		0.92		0.48	

Table 55: Test results using different classification systems with a disk/torus

8 Experiments Using Biomedical Spectra

8.1 Magnetic Resonance Spectra of Thyroid Biopsies

The magnetic resonance (MR) spectra set of thyroid biopsies was used to test the effectiveness of fuzzy interquartile encoding in a "real-world" scenario. It has been demonstrated in [105] that a MLP can be constructed that produces a robust classification of thyroid biopsies given their MR spectra. The inputs to the MLP were the ten best principal components of the original data that accounted for 97% of the total variance. Here, MLPs using the original spectral regions are used without any PCA preprocessing and compared with MLPs using the corresponding FE spectral regions. Twenty experiments were run for each case described below. Unlike the results discussed previously that were based solely on the test data, the average performance results listed in Table 56i–ii are based on all of the data (due to the paucity of data).

	NE(κ)	Iters	FE4(κ)	Iters	FE8(κ)	Iters
i) Choline I	0.64	1400	0.92	3	0.84	0.1
Lipid I	0.80	4000	0.88	5	0.88	0.4
ii) Choline II	0.96	600	0.96	10	0.76	1.0
Lipid II	1.00	2000	0.92	25	0.80	3.0
(FE4=fuzzy-encoded data using 4 fuzzy sets, NE=non-encoded data FE8=fuzzy-encoded data using 8 fuzzy sets, Iters=number of iterations (x1,000))						

Table 56: Classification results averaged over 100 runs

Four fuzzy sets were computed for each feature and the FE data were generated (680 and 1600 input points for the choline and lipid regions, respectively). The intersection, b, was set to 0.5 for all sets. Subsequently, eight fuzzy sets were computed by dividing each quartile in half. Table 56i lists the performance results. Again FE-MLPs outperformed their NE counterparts. What is particularly surprising is the rate of convergence for the FE-MLPs (for instance, the NE-MLPs used to classify the lipid regions are 800 times slower than the corresponding FE-MLPs).

Finally, comparisons were made using MLPs with some enhancements: momentum term; modulated learning; hyperbolic tangent function instead of the logistic function; and data scaling. In this case, the FE-MLPs using four fuzzy sets performed as well as their NE-MLP counterparts for the choline region but slightly poorer results were obtained for the lipid region (Table 56ii).

Although convergence still occurred much more quickly with the FE-MLPs, the NE-MLPs converged approximately twice as quickly with enhancements as without, whereas the FE-MLPs converged roughly 3–5 times more slowly. Moreover, when eight fuzzy sets were used, the overall classification rates were significantly poorer. Since data scaling occurred after the data were fuzzy encoded, the information content of the FE data may have actually changed, thereby affecting the nature of the problem. It was noted that when at least one of the MLP enhancements was deactivated, the FE-MLPs performance results approached those found in the FE-MLPs without any enhancements.

8.2 Infrared Spectra of Alzheimer's Diseased Brain Tissue

Three architectures are used to classify the original infrared (IR) spectra of the Alzheimer's diseased brain tissue (as described in section 3.3.5), an MLP employing an enhanced back-propagation algorithm as described in section 4.2.1, an FE-MLP as described in section 6, and an RBFN (section 4.4). The enhanced MLP (E-MLP) has ten hidden PEs and Gaussian noise was added to the system. The noise-free version produced significantly poorer results. The hyperbolic tangent was selected as the transfer function and modulated learning (α=0.7–0.02) and momentum (β=0.4–0.01) are used across layers as well as across epochs. The FE-MLP did not employ modulated learning. The transfer function was the logistic function and the learning and momentum rates were 0.7 and 0.4, respectively. Four fuzzy sets were computed for each feature and the FE data were generated (1664 input points). Triangular fuzzy sets were chosen and the intersection point was set at 0.5. Although there are four times as many inputs for this MLP than in the E-MLP case, mean square error convergence occurred in approximately an order of magnitude fewer iterations. For this investigation, the RBFN used three, five, and six prototypes for the 2, 4, and 5 class problems, respectively. The transfer point from unsupervised *k*-means clustering (to determine the centroids) to supervised gradient descent learning from the hidden

prototype layer to the output was varied in order to achieve good results for each of the two, four, and five class problems.

The spectra were randomly assigned to either the design or test set. Once assigned these sets were fixed for all runs. The ANN results are averages of ten runs using different initial random weight assignments and different random presentation sequences of the design set. All performance results are calculated using the test set. In the conventional preprocessing cases, PCA (using correlation matrices) is performed using all spectra. The first k principal components that accounted for 99.9% of the cumulative variance were used as inputs for both the LDA and ANN techniques. A MLP is used for the preprocessed spectra. Apart from the hidden layer consisting of three PEs rather than ten, this network's architecture is the same as the E-MLP above.

8.2.1 Two class problem

Of the 114 spectra used in the two class problem, 66 were placed in the design set (33 C and 33 A) and 48 in the test set (16 C and 32 A). The first nine principal components accounted for 99.99% of the cumulative variance (the first five, two, and one principal components accounted for 99.90%, 99.79%, and 99.24% of the cumulative variance, respectively). The ANNs consistently outperformed LDA in all cases by approximately 4% (Table 57). This performance gain can be explained by the non-linear nature of ANNs. Unlike LDA, they are not restricted to hyperplanar decision boundaries. It is interesting to note that the performance results were slightly better for the two principal component case than for the five principal component case. This is a testament to the fact that accounting for maximal variance does not necessarily translate into maximal discriminatory power. Table 58 lists results from the three ANNs using the original spectra. FE-MLP outperformed E-MLP and RBFN and produced results comparable to the best case from Table 57. This can be attributed to the fact that FE-MLP preprocesses the spectra by transforming each discrete data point into an ordered set of membership values whose

corresponding fuzzy sets are centred around the quartiles of each discrete spectral coordinate. As a result, a point that lies significantly outside the upper or lower quartile will have a diminished impact during learning (all but one of the membership values will be zero and the non-zero membership value will go to zero the further the data point is from the lower [or upper] quartile). Such spectral points are often considered to be outliers. Hence, FE-MLPs naturally reduce the negative role that outliers often play during training.

#PCs		LDA C	LDA A	ANN C	ANN A
9	C	16	0	16	0
	A	1	31	0	32
	κ	**0.95**		**1.00**	
5	C	12	4	14	2
	A	5	27	6	26
	κ	**0.58**		**0.65**	
2	C	13	3	14	2
	A	5	27	4	28
	κ	0.64		0.73	
1	C	10	6	9	7
	A	7	25	4	28
	κ	**0.40**		**0.46**	

Table 57: ANN versus LDA (principal components and two classes)

	FE-MLP C	FE-MLP A	E-MLP C	E-MLP A	RBFN C	RBFN A
C	16	0	13	3	14	2
A	1	31	0	32	7	25
κ	***0.95***		***0.85***		***0.61***	

Table 58: Classification results (original spectra and two classes)

8.2.2 Five class problem

Of the 163 spectra used in the five class problem, 104 were used in the design set (33 CG, 16 CW, 6 NT, 33 AG, and 16 AW) and 59 in the test set (16 CG, 7 CW, 3 NT, 25 AG, and 8 AW). The first ten principal components accounted for 99.99% of the cumulative variance (the first eight and five principal components accounted for 99.95% and 99.90% of the cumulative variance, respectively). Again the MLPs outperformed the corresponding LDAs (Table 59) but this time by a wider margin. The fact that both methods did not perform as well as in the two class problem is to be expected. It is more difficult to discriminate between control grey and

white matter and Alzheimer's diseased grey and white matter. Moreover, the increased discriminatory complexity and the concomitant requirement for finer discriminatory hypersurfaces also accounts for the wider performance margin between ANN and LDA. It should also be noted that although there were more misclassifications in this five class problem using the first ten principal components, both methods exhibited conservative misclassification (except for one spectrum in the LDA case). That is, control tissue was always classified as control tissue and AD tissue was always classified as AD tissue. When errors occurred it was only in the determination of the tissue as white or grey matter. Unfortunately, all of the NT spectra were misclassified in all cases. This problem is due to the paucity of NT spectra in the design set. Nevertheless, these spectra were classified as being either AG or AW. Since NTs are one of the hallmarks of AD, it is at least more preferable that NT spectra be misclassified as AD tissue rather than control tissue.

		LDA					MLP				
#PC		CG	CW	NT	AG	AW	CG	CW	NT	AG	AW
	CG	16	0	0	0	0	16	0	0	0	0
	CW	0	7	0	0	0	0	7	0	0	0
10	NT	0	0	0	2	1	0	0	0	1	2
	AG	1	0	5	19	0	0	0	2	23	0
	AW	0	0	2	0	6	0	0	1	0	7
	κ			**0.75**					**0.86**		
	CG	16	0	0	0	0	16	0	0	0	0
	CW	1	6	0	0	0	0	7	0	0	0
8	NT	0	0	0	2	1	0	0	0	2	1
	AG	1	0	2	21	1	0	0	0	24	1
	AW	0	0	2	0	6	0	0	1	1	6
	κ			**0.76**					**0.81**		
	CG	13	1	0	2	0	12	2	0	2	0
	CW	2	5	0	0	0	2	5	0	0	0
5	NT	1	0	0	1	1	1	0	0	1	1
	AG	3	0	4	17	1	3	0	3	19	0
	AW	1	0	0	3	4	1	0	0	2	5
	κ			**0.52**					**0.57**		

Table 59: ANN versus LDA (principal components and five classes)

Table 60 lists the performance results of the three ANNs using the original spectra. Once again FE-MLP produced the best results. It performed almost as well as the best case from Table 59 and outperformed all LDA results.

		CG	CW	NT	AG	AW
FE-MLP	CG	14	1	0	1	0
	CW	3	4	0	0	0
	NT	0	0	3	0	0
	AG	0	0	0	25	0
	AW	1	0	0	3	4
	κ			**0.78**		
E-MLP	CG	14	1	0	1	0
	CW	0	6	0	1	0
	NT	3	0	0	0	0
	AG	3	0	0	22	0
	AW	3	0	0	5	0
	κ			**0.57**		
RBFN	CG	13	2	0	0	1
	CW	2	5	0	0	0
	NT	0	0	2	1	0
	AG	4	0	1	19	1
	AW	1	0	1	3	3
	κ			**0.60**		

Table 60: Classification results (original spectra and five classes)

8.2.3 Effect of auxiliary data on principal components

It is often the case that new spectra collected after the initial classification process has been completed need to be classified. In cases where principal components are used to preprocess the data, this means that the principal components must be computed for the new spectra. Further, the components for the original spectra need to be recomputed. We now investigate this issue using a four class variant of the problem discussed above. The spectra were divided into 3 sets: a design set with 33 CG, 16 CW, 33 AG, and 16 AW for a total of 98 spectra; a test set with 16 CG, 7 CW, 25 AG, and 8 AW for a total of 56 spectra; and, an auxiliary test set with 9 NT spectra. Table 61 lists the results using the first ten principal components that accounted for 99.99% of the cumulative variance. In this case, the principal components were computed using only the spectra in the design and test set. The overall results were κ=0.92 for the ANN and κ=0.87 for LDA.

	LDA				**ANN**			
	CG	CW	AG	AW	CG	CW	AG	AW
CG	16	0	0	0	16	0	0	0
CW	0	7	0	0	0	7	0	0
AG	1	0	22	2	0	0	24	1
AW	0	0	2	6	0	0	2	6
κ		**0.87**				**0.92**		

Table 61: ANN versus LDA results (principal components and four classes)

Finally, principal components were re-computed using the above spectra as well as the auxiliary test set (for a total of 163 spectra). The first ten principal components accounted for 99.99% of the cumulative variance. In this case, the classification results are calculated using only the test set. However, since the principal components have been calculated for the auxiliary test set, their classification outcomes can be generated (Table 62). The overall classification rate for the ANN remained unchanged at κ=0.92 but the LDA rate dropped to κ=0.84. The notion of accuracy is meaningless with regard to the classification of the auxiliary test spectra because no desired outcomes were associated with them. Nevertheless, it is interesting to note that, as in the five class problem, both the ANN and LDA classified the auxiliary test set as spectra from AD tissue (either AG or AW).

	LDA				ANN			
	CG	CW	AG	AW	CG	CW	AG	AW
CG	16	0	0	0	16	0	0	0
CW	1	6	0	0	0	7	0	0
AG	1	0	22	2	0	0	24	1
AW	0	0	2	6	0	0	2	6
κ		**0.84**				**0.92**		
NT	0	0	5	4	0	0	7	2

Table 62: ANN versus LDA results (four classes and PCs based on all five classes)

Diagnosis of AD from autopsy material by IR spectroscopy has proven to be difficult based simply upon a spectroscopic analysis, due to the different degree of involvement of brain tissue, the difficulty in staging the disease and the extensive biochemical changes associated with gross degeneration of the grey matter. Classification of IR spectra of control and Alzheimer's disease tissue has been achieved with a high degree of accuracy by both LDA and ANNs. Separation of grey and white matter into distinct classes is not surprising, given the known biochemical differences between the two tissue types. The separation of AD and control grey matter presumably reflects spectral differences associated with the pathological features of AD, namely general atrophy of the cerebral cortex and the presence of neuritic plaques and neurofibrillary tangles. The ability to distinguish between AD and control white matter may be considered surprising as AD is a disease of the grey matter. However, recent studies [112] have shown that

significant variations in the phospholipid composition of white matter is also associated with AD, a finding which probably explains the ability of LDA and ANNs to discriminate between control and AD white matter.

Table 57, Table 59, and Table 62 clearly show that ANNs consistently outperform their LDA counterparts in all cases where PCA was used as a preprocessing technique. Of course, as the complexity of the problem increased (from two to five classes) both techniques suffered some loss in classification accuracy but this loss was more pronounced with LDA than ANN.

Although preprocessing the spectra using PCA is quite useful and often improves performance results, there is a concomitant loss in flexibility in the addition or deletion of data as well as a loss (in general) of the ability to analyze relevant features in the original spectra that contributed to the discriminatory power of the underlying method.

Finally, in the cases where the original spectra were used, FE-MLPs outperformed E-MLPs and RBFNs and had classification results that were only slightly worse than the best results achieved using PCA. This may be expected since FE-MLP explicitly employs fuzzy encoding as a preprocessing technique as opposed to the other two architectures. Further, since the fuzzy sets are constructed around the quartiles for each data point any outlier values end up having a smaller influence during the iterative training of the net.

8.3 Burnishing Tarnished Gold Standards

The data set used to test the efficacy of FA and RR is the MR spectra of human brain neoplasms described in section 3.3.3. The phased spectra were normalized (each datum was divided by the area of the spectrum) and randomly assigned to either a design set (n=80) or a test set (n=126). The design set contained 29 M's, 31 A's, and 20 E's. The GS was provided by a pathologist and was encoded using the procedure described in section 3.1.

The FA method described in section 6.2.1 was integrated into an MLP architecture consisting of two hidden layers with three PEs in the output layer (one for each class). The networks were trained on spectra from the design set using the GS and its adjustments.

FA improved the overall diagnostic performance of the MLP compared to the original GS. Table 63 lists the performance results using the test MR spectra. A 13% improvement in the κ score was achieved using FA. FA does not alter the original GS classification of the design spectra but it does modify the traversal of the MLP's weight space during the training process; those spectra that are near class medoids other than their own contribute less to the incremental changes to the MLP compared to those design spectra that are near their own medoids.

In the RR case, when the outliers were reclassified in the design set, a 10% improvement in the κ score was achieved. Interestingly, if these outlying spectra were removed from the design set, κ degraded to 0.62 for the test set (a 13% decrease from the original GS κ score). Although none of the spectra in the test set was reclassified, using the MAD criterion two test spectra were flagged as outliers (two M spectra were flagged as A spectra). In all three cases, GS, FA, and RR, these spectra were indeed misclassified as A spectra.

Finally, the classification errors were also more conservative for both FA and RR as compared to the original GS. That is, while the original GS classified 5 E's (non-tumourous) as either M's or A's (tumours) and 4 tumours as E's, FA and RR classified only 1 E as tumourous and 3 tumours as E's.

	GS			FA			RR			RR*		
	M	E	A	M	E	A	M	E	A	M	E	A
M	61	2	3	58	2	6	57	1	8	56	4	6
E	1	12	4	0	16	1	0	16	1	0	16	1
A	9	2	32	5	1	37	5	2	36	13	5	25
κ		**0.71**			**0.80**			**0.78**			**0.62**	

Table 63: Performance results using test spectra (RR*, outliers removed)

It is informative to examine the κ scores for the design spectra using the different methods (Table 64). Although the GS and FA contingency tables are identical, the weights of the

underlying MLP's are sufficiently different to exact a significant performance gain in the classification of the test spectra.

With RR, three spectra were reclassified in the design set: a M spectrum to A; a E spectrum to A; and, an A spectrum to E. When reclassification took place, κ improved from 0.97 to 1.00. However, if those spectra were removed from the design set, as is often the case in classification problems, κ actually degrades to 0.93, suggesting that these spectra, although identified as outliers, have sufficient import to affect the learning cycle of the MLP. Further, the M and E spectra that were identified as outliers were classified as M and E spectra, respectively, using the GS. However, the A spectrum that was identified as an outlier (and reclassified as an E spectrum using RR) was misclassified as an E spectrum using the GS.

	GS			FA			RR			RR*		
	M	E	A	M	E	A	M	E	A	M	E	A
M	29	0	0	29	0	0	28	0	0	28	0	0
E	0	20	0	0	20	0	0	20	0	0	19	0
A	0	2	29	0	2	29	0	0	32	2	2	26
κ		**0.97**			**0.97**			**1.00**			**0.93**	

Table 64: Performance results using design spectra (RR*, outliers removed)

The results demonstrate that the adjustment of a GS using a fuzzy or robust measure of deviation of MR spectra from their respective class medoids leads to a reduction in classification errors. Moreover, misclassifications tend to be more conservative. Recall that, if reclassification occurs, it only occurs for spectra within the design set; outliers within the test set are simply flagged but not altered using this method. Therefore, the accepted GS is left in a pristine state sullied only by its original tarnish.

8.4 Additional Experiments

The final set of experiments were performed using the magnetic resonance spectra, described in section 3.3.4 (N=206, N_d=80, N_t=126, n=550), consisting of 95 meningiomas (M), 74 astrocytomas (A), and 37 control samples of non-tumourous brain tissue from patients with epilepsy (E). The following classification systems were used:

- linear discriminant analysis (LDA);
- linear discriminant analysis with principal component analysis (PC);
- multi-layer perceptron (MLP);
- multi-layer perceptron with principal component analysis (PCM);
- multi-layer perceptron with fuzzy gold standard adjustments (FA);
- multi-layer perceptron with robust reclassification (RR);
- radial basis function neural network (RBF);
- probabilistic neural network (PNN);
- fuzzy interquartile encoding (IQ) and its class-wise variant (IQc);
- dimension-preserving interquartile encoding (DP) and its class-wise variant (DPc);
- fuzzy cluster encoding (CL) and its class-wise variant (CLc).

For PC and PCM, the first 90 principal components were used and accounted for 99.99% of the cumulative variance of the original data. MLP, FA, and RR, have exactly the same structures described in section 8.3. RBF and PNN have 50 receptive fields and 50 kernels, respectively. For CL and CLc, the best results obtained using 2–20 clusters are listed (c=19 and c=5, respectively).

For all methods, κ>0.90 for the design set, however, κ varied widely for the test set as is shown in Figure 140. Note the dramatic improvement of PC (κ=0.74) compared to LDA (κ=0.52). While part of this is certainly due to the use of the principal components instead of the original data another contributing factor is that the original 550×550 covariance matrix that LDA had to invert was nearly singular and hence ill-conditioned. Using the same principal components, PCM (κ=0.70) fared slightly worse than MLP (κ=0.71) using the original spectra. Section 8.3 discusses the efficacy of FA (κ=0.80) and RR (κ=0.78). Neither PNN (κ=0.43) nor RBF (κ=0.53)

performed particularly well; the former suffered from round-off error in computing the Parzen kernels. The class-wise variants. CLc (κ=0.55), DPc (κ=0.50), and IQc (κ=0.44), also were underperformers. IQc classification results were especially poor but this was due to the large increase in the dimensionality of the transformed space; from n=550 to n=6600 (550 × 4 fuzzy sets × 3 classes). DP performed well (κ=0.72) as did CL (κ=0.74). IQ (κ=0.83) had the best agreement measure in spite of the fact that n=2200.

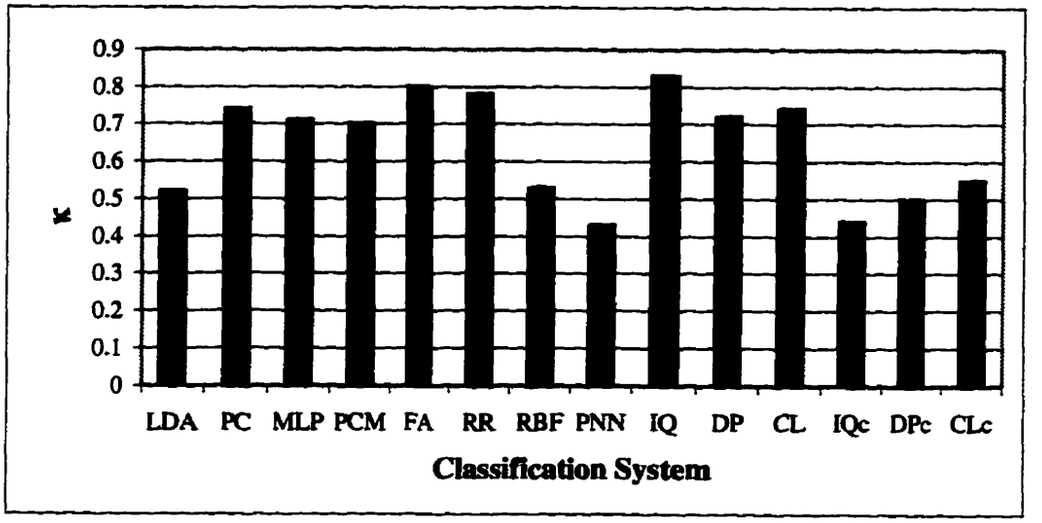

Figure 140: κ scores for classification systems

9 Conclusion

The intent of this thesis was to introduce, derive, implement, and determine the efficacy of two new preprocessing methodologies, *fuzzy feature space encoding*, and *burnishing tarnished gold standards*. The former comprises a collection of methods (fuzzy interquartile encoding, dimension-preserving fuzzy interquartile encoding, fuzzy cluster encoding, and their class-wise variants) for determining the respective degrees to which a datum belongs to a collection of fuzzy sets or fuzzy clusters and subsequently using these membership grades in place of the original datum. The latter comprises methods (robust reclassification, fuzzy gold standard adjustment, and their variants) to compensate for the possible imprecision or unreliability of a well-established gold standard while, at the same time, maintaining its vital discriminatory power by incorporating non-subjective within-class medoid information. The underlying purpose of these methodologies is to simplify the feature space prior to presentation to a classifier. As they are independent of any particular classification method, they may be integrated into any classification system.

9.1 Summary

This thesis began with an introduction to some essential concepts necessary for the understanding of the fuzzy encoding and gold standard burnishing methodologies:

- *classification*, the construction of a discrimination function mapping individuals to a set of class indices;
- *artificial neural networks*, a self-adaptive, non-linear, massively parallel machine learning system composed of layers of processing elements used primarily for pattern recognition problems;
- *fuzzy set theory*, a generalization of Boolean set theory, extending the notion of elementhood from the range {0, 1} to the interval [0, 1];

- *robust statistics*, statistics resistant to outlier effects; they are insensitive to slight deviations from their requisite model (often normal) assumptions about the underlying distribution.

Issues concerning the classification process were then discussed:

- the different stages of classification
  - classifiers,
  - preprocessing and postprocessing;
- linear separability and linear classifiers;
- conventional preprocessing methods such as principal component analysis;
- artificial neural networks as non-linear classifiers
  - multi-layer perceptron,
  - probabilistic neural network,
  - radial basis function neural network;
- misconceptions concerning Gauss' law of errors and normality assumptions;
- synthetic data
  - 1-dimensional 2-class sets with different distributions,
  - disk/torus,
  - hypercubes;
- "real-world" data
  - magnetic resonance biomedical spectra,
  - infrared biomedical spectra;
- measuring classification performance.

Concerning the last point, the chance-corrected measure of agreement, κ, was selected to assess the performance of all classification strategies used as it is more accurate than the conventional measure of the ratio of number of correctly classified data and the entire data set.

Fuzzy feature space encoding and gold standard burnishing are then introduced and mathematically described. Integration of these methods into classifiers is discussed. The burnishing methods, robust reclassification and fuzzy gold standard adjustments are also detailed as well as the motivational differences between reclassification and adjustment. A set of experiments using synthetic data were performed in order to measure the efficacy of these methods and benchmarked against linear discrimination and a multi-layer perceptron. A set of experiments using magnetic resonance and infrared biomedical spectra were also performed and the results presented and discussed.

9.2 Concluding Remarks

As a general preprocessing methodology, fuzzy encoding is effective in improving classification performance by transforming the feature space prior to presentation to a classifier. The fuzzy encoding methods, applicable to any classifier, exhibit several useful properties. First, since the membership functions map values onto the unit interval, data are automatically scaled. This is particularly useful in the classification process since scaled data diminish the effects of extreme variances across features. Without scaled data, features with large variances will predominate over features with small variances although the latter features may be discriminatory. Another beneficial property is that values that may be considered outliers impact less severely upon classifiers, such as the multi-layer perceptron, that employ any type of iterative adjustments to its error function. This does not mean that samples with features that are outliers are removed during the design or test phases of the classification process, however. With the interquartile encoding methods, as the value moves away from the interquartile range, the fuzzy encoded values tend to zero. In the case of a multi-layer perceptron where its hidden layer

processing elements are summing products of weights and input values this is important since, if the fuzzy encoded values of an outlier are all zero or near zero, those values will contribute very little to the learning process regardless of the processing elements weights; an extremely useful property if the original value is indeed an outlier (nevertheless, if it is not an outlier it still does contribute to a degree).

Fuzzy interquartile encoding is the most robust encoding method; it was least sensitive to changes to the underlying distributions of the synthetic data and performed well with all biomedical spectra. Dimension-preserving interquartile encoding and fuzzy cluster encoding are more erratic in their performance. The former method does not work well if features have a unimodal distribution; values that are equidistant from a feature's median will then have the same encoded value. However, this weakness is also its strength when features do not have an unimodal distribution or the distribution is skewed. Fuzzy cluster encoding is sensitive to the correlation of initial clusters to the underlying clusters of the data. The experiments with the synthetic data sets indicate that if the initial clusters are near the modes of the underlying distributions of the data then the performance results are excellent. If not, the results may degrade significantly. The class-wise variants of all three methods also produced variable results.

Concerning the gold standard burnishing methodology, the results from the synthetic data sets and the biomedical spectra demonstrate that the fuzzy gold standard adjustment and robust reclassification methods improved the classification performance compared to the original gold standard assignments. Moreover, misclassifications with the biomedical spectra tend to be more conservative. If reclassification occurs, it only occurs for data within the design set; outliers within the test set are simply flagged but not altered using this method. Therefore, the accepted gold standard is left in a pristine state sullied only by its original tarnish.

In conclusion, this thesis has argued that, in pattern recognition problems, preprocessing is of paramount importance and the methodologies of fuzzy feature space encoding and gold standard

burnishing are good additions to the preprocessing arsenal. In the construction of good classification systems, the 80/20 rule most certainly holds: only 20% of an investigator's effort should be devoted to the selection and tuning of a classifier; the remaining, and more crucial, effort should be devoted to a thorough analysis and preprocessing of the data in order to reduce the complexity of the feature space prior to its presentation to the classifier of choice.

10 References

1. Abidi MA, Abdulghafour M, Chandra T. Fusion of visual and range features using fuzzy logic. *Control Engineering Practice* 1994;**2**:833–47.
2. Abou-Chadi FEZ, Ezzat FA, El–Din AAS. A Fuzzy Pattern Recognition Method to Classify Esophageal Motility Records. *Ann Biomed Eng* 1994;**22**:112–9.
3. Aikens J. Prototypical knowledge for expert systems. *Artif Intel* 1983;**20**:163–210.
4. Andrews DF, Bickel PJ, Hampel FR, Huber PJ, Rogers WH, Tukey JW. Robust Estimates of Location: Surveys and Advances. Princeton: Princeton University Press, 1972.
5. Angelov P. A generalized approach to fuzzy optimization. *International Journal of Intelligent Systems* 1994;**9**:261–8.
6. Arakawa K, Arakawa Y. Digital signal processing using fuzzy clustering for nonstationary signals. *Proceedings of Fuzzy Engineering: Toward Human Friendly Systems*. Yokohama, November 13–15, 1992.
7. Azimi-Sadjadi MR, Ghaloum S, Zoughi R. Terrain classification in SAR images using principal component analysis and neural networks. *IEEE Trans Geosci Remote Sens* 1993;**31**:511–5.
8. Beckman RJ, Cook RD. Outlier.....s. *Technometrics* 1983;**25**:119–49.
9. Bernstein IH, Garbin CP, Teng GK. Applied Multivariate Analysis. New York: Springer-Verlag, 1988.
10. Bezdek JC. A convergence theorem for the fuzzy ISODATA clustering algorithms. *IEEE Trans Patt Anal Mach Intell* 1980;**2**:1–8.
11. Bezdek JC, Ehrlich R, Full W. FCM: the fuzzy c–means clustering algorithm. *Computers and Geosciences* 1984;**10**:191–203.
12. Bezdek JC, Hathaway RJ. Numerical convergence and interpretation of the fuzzy c–shells clustering algorithm. *IEEE Transactions on Neural Networks* 1992;**3**:787–93.
13. Binaghi E, Rampini A. Fuzzy decision making in the classification of multisource remote sensing data. *Opt Eng* 1993;**32**:1193–204.
14. Bishop CM. Neural networks and their applications. *Rev Sci Instrum* 1994;**65**:1803–32.
15. Bishop CM. Neural Networks for Pattern Recognition. Oxford: Clarendon Press, 1995.
16. Buckley JJ, Hayashi Y. Fuzzy neural networks: A survey. *Fuzzy Sets Syst* 1994;**66**:1–13.
17. Chen T, Chen H. Approximation capability to functions of several variables, nonlinear functionals, and operators by radial basis function neural networks. *IEEE Transactions on Neural Networks* 1995;**6**:904–10.
18. Cheng B, Titterington DM. Neural Networks: A Review from a Statistical Perspective. *Statistical Science* 1994;**9**:2–54.
19. Cho S-B, Kim JH. Combining mulitple neural networks by fuzzy integrals for robust classification. *IEEE Trans Syst Man Cybern* 1995;**25**:380–4.
20. Choo L-P, Mansfield JR, Pizzi N, Somorjai RL, Jackson M, Halliday WC, et al. Infrared spectra of human central nervous system tissue: diagnosis of Alzheimer's disease by multivariate analyses. *Biospectrosc* 1995;**1**:141–8.
21. Cohen J. A coefficient of agreement for nominal scales. *Educ Psychol Meas* 1960;**20**:37–46.
22. Cooley WW, Lohnes PR. Multivariate Data Analysis. New York: John Wiley & Sons, 1971.
23. Cramer H. Mathematical Methods of Statistics. Princeton: Princeton University Press, 1946.
24. Daniel C, Wood FS. Fitting Equations to Data. New York: John Wiley, 1971.
25. Dayhoff JE. Neural Network Architectures. New York: Von Nostrand Reinhold, 1990.
26. Degani R, Bortolan G. Fuzzy numbers in computerized electrocardiography. *Fuzzy Sets Syst* 1987;**24**:345–62.
27. Devlin SJ, Gnanadesikan R, Kettenring JR. Robust estimation and outlier detection with correlation coefficients. *Biometrika* 1975;**62**:531–45.
28. Dubois D, Prade H. Possibility theory and data fusion in poorly informed environments. *Control Engineering Practice* 1994;**2**:811–23.
29. Duda RO, Hart PE. Pattern Classification and Scene Analysis. New York: Wiley & Sons, 1973.
30. Everitt BS. Moments of the statistics kappa and weighted kappa. *Br J Math Stat Psychol* 1968;**21**:97–103.
31. Everitt BS. The Analysis of Contingency Tables. London: Chapman & Hall, 1992.

32. Fisher LD, VanBelle G. Biostatistics: A Methodology for the Health Sciences. New York: John Wiley & Sons, 1993.
33. Fleiss JL. Measuring agreement between judges on the presence or absence of a trait. *Biometrics* 1975;**31**:651–9.
34. Fleiss JL, Cohen J, Everitt BS. Large sample standard errors of kappa and weighted kappa. *Psychol Bull* 1969;**72**:323–7.
35. Fukunaga K. Introduction to Statistical Pattern Recognition. Boston: Academic Press, 1990.
36. Gaines BR. Logical foundations for database systems. In: Mamdani EH, Gaines BR, editors. *Fuzzy Reasoning and its Applications.* London: Academic Press, 1981:289–308.
37. Girosi F, Poggio T, Caprile B. Extensions of a theory of networks for approximations and learning: outliers and negative examples. In: Moody JE, Touretzky DS, editors. *Advances in Neural Information Processing Systems.* San Mateo: Morgan Kaufmann, 1991:750–756.
38. Gnanadesikan R. Methods for Statistical Data Analysis of Multivariate Observations. New York: John Wiley, 1977.
39. Gottwald S. Fuzzy set theory: some aspects of the early development. In: Skala HJ, Termini S, Trillas E, editors. *Aspects of Vagueness.* Dordrecht: D. Reidel Publ. Comp., 1984:13–29.
40. Grossberg S. Nonlinear neural networks: principles, mechanisms, and architectures. *Neural Netw* 1988;**1**:17–61.
41. Hand DJ. Discrimination and Classification. New York: John Wiley & Sons, 1981.
42. Hartman E, Keeler JD. Predicting the future: advantages of semilocal units. *Neural Computation* 1991;**3**:566–78.
43. Hassoun MH. Fundamentals of Artificial Neural Networks. Cambridge: MIT Press, 1995.
44. Hayashi Y, Buckley JJ. Fuzzy neural network with fuzzy signals and weights. *International Journal of Intelligent Systems* 1993;**8**:527–37.
45. Haykin S. Neural Networks: A Comprehensive Foundation. New York: Macmillan College Publishing Company, 1994.
46. Henkind SJ, Yager RR, Benis AM, Harrison MC. A clinical alarm system using techniques from artificial intelligence and fuzzy set theory. In: Sanchez E, Zadeh LA, editors. *Approximate Reasoning in Intelligent Systems, Decision and Control.* New York: Pergamon Press, 1987:91–104.
47. Hertz J, Krogh A, Palmer RG. Introduction to the Theory of Neural Computation. Redwood City: Addison–Wesley Publishing Company, 1991.
48. Hisdal E. Are grades of membership probabilities? *Fuzzy Sets Syst* 1988;**25**:325–48.
49. Hotelling H. Analysis of a complex of statistical variables into principal components. *J Educ Psychol* 1933;**24**:417–41.
50. Huber PJ. Robust estimation of a location parameter. *Ann Math Statist* 1964;**35**:73–101.
51. Huber PJ. Robust statistics: a review. *Annals of Mathematical Statistics* 1972;**43**:1041–67.
52. Huber PJ. Robust Statistical Procedures. Bristol: Arrowsmith, 1977.
53. Intrator N. On the combination of supervised and unsupervised learning. *Physica A* 1993;**200**:655–61.
54. Jackson M, Mantsch HH. Biomembrane structure from FT–IR spectroscopy. *Spectrochim Acta Rev* 1993;**15**:53–69.
55. Jacobs RA, Jordan MI, Nowlan SJ, Hinton GE. Adaptive mixtures of local experts. *Neural Computation* 1991;**3**:79–87.
56. Jobson JD. Applied Multivariate Data Analysis: Categorical and Multivariate Methods. New York: Springer–Verlag, 1992.
57. Katzman R, Jackson JE. Alzheimer Disease: basic and clinical advances. *J Am Geriatr Soc* 1991;**39**:516–29.
58. Keller JM, Hunt DJ. Incorporating Fuzzy Membership Functions into the Perceptron Algorithm. *IEEE Trans Patt Anal Mach Intell* 1985;**7**:693–9.
59. Kittel WA, Epstein CM, Hayes MH. EEG monitoring based on fuzzy classification. *Proceedings of the 35th Midwest Symposium on Circuits and Systems.* New York. August 9–12, 1992.
60. Klir GJ, Folger TA. Fuzzy Sets, Uncertainty, and Information. New Jersey: Prentice Hall, 1988.
61. Kohonen T. Self–Organization and Associative Memory. New York: Springer–Verlag, 1989.
62. Kosko B. Fuzziness versus probability. *Int J General Systems* 1990;**17**:211–40.
63. Kosko B. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Englewood Cliffs: Prentice Hall, 1992.
64. Kosko B. Fuzzy Engineering. Englewood Cliffs: Prentice Hall, 1997.

65. Krzanowski W. Principles of Multivariate Analysis. New York: Oxford University Press, 1988.
66. Kummert A. fuzzy technology implemented in sonar systems. *IEEE J Ocean Eng* 1993;**18**:483–90.
67. Kuncheva L. An aggregation of pro and con evidence for medical decision support systems. *Comput Biol Med* 1993;**23**:417–24.
68. Landis JR, Koch GG. The measurements of observer agreement for categorical data. *Biometrics* 1977;**33**:159–74.
69. Lee S, Kil RM. A Gaussian potential function network with hierarchically self-organizing learning. *Neural Netw* 1991;**4**:207–24.
70. Lee SC, Lee ET. Fuzzy neural networks. *Math Biosci* 1975;**23**:151–77.
71. Leonard JA, Kramer MA. Radial basis function networks for classifying process faults. *IEEE Control Systems* 1991;**4**:31–8.
72. Lippmann RP. An introduction to computing with neural networks. *IEEE ASSP* 1987;**4**:4–22.
73. Machado RJ, Da Rocha AF. A hybrid architecture for fuzzy connectionist expert systems. In: Kandel A, Langholz G, editors. *Hybrid Architectures for Intelligent Systems.* Boca Raton: CRC Press Inc., 1992:136–52.
74. MacKay DJC. Bayesian neural networks and density networks. *Nucl Instrument Methods Phys Res Section A Accel Spectromet Detect Ass Equip* 1995;**354**:73–80.
75. Maloney PS, Specht DF. The use of probabilistic neural networks to improve solution times for hull-to-emitter correlation problems. *Proceedings of the International Joint Conference on Neural Networks.* New Jersey: IEEE Press, 1989.
76. Manly BFJ. Multivariate Statistical Methods: A Primer. New York: Chapman & Hall, 1986.
77. McDermott J. R1 revisited: Four years in the trenches. *AI Magazine* 1984;**5**:21–32.
78. Mendenhall W, Sincich T. Statistics for the Engineering and Computer Sciences. San Francisco: Dellen Publ. Comp., 1988.
79. Michalski RS, Chilausky RL. Learning by being told and learning from examples. *International Journal of Policy Analysis and Information Systems* 1980;**4**:125–61.
80. Mitchell J, Abe S. Fuzzy clustering networks: design criteria for approximation and prediction. *IEICE Trans Inf & Syst* 1996;**79**:63–71.
81. Mitra S. Fuzzy MLP based expert system for medical diagnosis. *Fuzzy Sets Syst* 1994;**65**:285–96.
82. Moody J, Darken C. Learning with localized receptive fields. In: Touretzky D, Hinton G, Sejnowski T, editors. *Proceedings of the Connectionist Models Summer School.* San Mateo: Morgan Kaufmann, 1988:133–43.
83. Moody J, Darken CJ. Fast learning networks of locally-tuned processing units. *Neural Computation* 1989;**1**:381–294.
84. Negoita CV. Fuzzy Systems. Tunbridge Wells: Abacus Press, 1981.
85. Norris D, Pilsworth BW, Baldwin JF. Medical diagnosis from patient records – a method using fuzzy discrimination and connectivity analyses. *Fuzzy Sets Syst* 1987;**23**:73–87.
86. Parzen E. On estimation of a probability density function and mode. *Annals of Mathematical Statistics* 1962;**33**:1065–76.
87. Pearson K. On lines and planes of closest fit to a system of points in space. *Philosophical Magazine* 1901;**2**:557–72.
88. Pizzi, N, Somorjai, RL. To burnish tarnished gold standards: a classification strategy for MR spectra as applied to human brain neoplasm diagnosis. *Proceedings of the Third Scientific Meeting and Exhibition of the Society of Magnetic Resonance.* Nice, France, August 19–25, 1995.
89. Pizzi N, Somorjai RL. Fuzzy encoding as a preprocessing method for artificial neural networks. *Proceedings of the World Congress on Neural Networks.* San Diego, USA, June 5–9, 1994.
90. Platt J. A resource-allocating network for function interpolation. *Neural Computation* 1991;**3**:213–25.
91. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical Recipes in C. Cambridge: Cambridge University Press, 1992.
92. Qu L, Chen Y, Liu X. A new approach to computer-aided vibration surveillance of rotating machinery. *International Journal of Computer Applications in Technology* 1989;**2**:108–17.
93. Reiter R. A logic for default reasoning. *Artif Intel* 1980;**13**:81–132.
94. Rietz HL. Mathematical Statistics. La Salle: The Open Court Publishing Company, 1927.
95. Ripley BD. Neural networks and related methods for classification. *J R Stat Soc [B]* 1994;**56**:409–56.
96. Ripley BD. Pattern Recognition and Neural Networks. Cambridge: Cambridge Univ. Press, 1996.
97. Rogova G. Combining the results of several neural network classifiers. *Neural Netw* 1994;**7**:777–81.

98. Rousseeuw PJ, Leroy AM. Robust Regression and Outlier Detection. New York: John Wiley & Sons, 1987.
99. Rousseeuw PJ, Zomeren BC. Unmasking multivariate outliers and leverage points. *J Am Stat Assoc* 1990;**85**:633–9.
100. Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL, editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Cambridge: MIT Press, 1986:318–62.
101. Russo F, Ramponi G. Nonlinear fuzzy operators for image processing. *Signal Processing* 1994;**38**:429–40.
102. Schodel H. Utilization of fuzzy techniques in intelligent sensors. *Fuzzy Sets Syst* 1994;**63**:271–92.
103. Seber G. Multivariate Observations. Toronto: Wiley & Sons, 1984.
104. Sokal RR, Rohlf FJ. Introduction to Biostatistics. New york: W. H. Freeman, 1987.
105. Somorjai RL, Nikulin AE, Pizzi N, Jackson D, Scarth G, Dolenko B, et al. Computerized consensus diagnosis: a classification strategy for the robust analysis of MR spectra. I. Application to ^{1}H spectra of thyroid neoplasms. *Magn Reson Med* 1995;**33**:257–63.
106. Specht D. Probabilistic neural networks for classification, mapping or associative memory. *Proceedings of the IEEE International Conference on Neural Networks.* New Jersey: IEEE Press, 1988.
107. Specht DF. Probabilistic neural networks. *Neural Netw* 1990;**3**:109–18.
108. Srivastava MS, Carter EM. An Introduction to Applied Multivariate Statistics. New York: North–Holland, 1983.
109. Staudte RG, Sheather SJ. Robust Estimation and Testing. New York: Wiley & Sons, 1990.
110. Streit RL, Luginbuhl TE. Maximum likelihood training of probabilistic neural networks. *IEEE Transactions on Neural Networks* 1994;**5**:764–83.
111. Sweeney WP, Musavi MT, Guidi JN. Classification of chromosomes using a probabilistic neural network. *Cytometry* 1994;**16**:17–24.
112. Svennerholm L, Gottfries C–G. Membrane lipids, selectively diminshed in Alzheimer brains, suggest synapse loss as a primary event in early–onset form (Type I) and demyelination in late–onset form (Type II). *J Neurochem* 1994;**62**:1039–47.
113. Taylor JG, Plumbley MD. Information theory and neural networks. In: Taylor JG, editor. *Mathematical Approaches to Neural Networks.* Amsterdam: Elsevier Science Publs, 1993:307–40.
114. Tukey JW. A survey of sampling from contaminated distributions. In: Olkin I, editor. *Contributions to Probability and Statistics.* Stanford: Stanford University Press, 1960:292–301.
115. Uebele V, Abe S, Lan M–S. A neural–network–based fuzzy classifier. *IEEE Trans Syst Man Cybern* 1995;**25**:353–61.
116. Umbers IG, King PJ. An analysis of human decision–making in cement kiln control and the implications for automation. *Int J Man–mach Stud* 1980;**12**:11–23.
117. Watanabe H, Yakowenko WJ, Kim Y–M, Anbe J, Tobi T. Application of a fuzzy discriminant analysis for diagnosis of valvular heart disease. *IEEE Transactions on Fuzzy Systems* 1994;**2**:267–76.
118. Weiss SM, Kulikowski CA. Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems. San Mateo: Morgan Kaufmann Publ., 1991.
119. Wilkins MF, Morris CW, Boddy L. A comparison of Radial Basis Function and backpropagation neural networks for identification of marine phytoplankton from multivariate flow cytometry data. *Comput Appl Biosci* 1994;**10**:285–94.
120. Wolpert DH. Stacked generalization. *Neural Netw* 1992;**5**:241–59.
121. Xu L, Krzyzak A, Yuille A. On radial basis functions nets and kernel regression: statistical consistency, convergence rates, and receptive field size. *Neural Netw* 1994;**7**:609–28.
122. Yang M–S. A survey of fuzzy clustering. *Mathl Comput Modelling* 1993;**18**:1–16.
123. Zadeh LA. Fuzzy sets. *Information and Control* 1965;**8**:338–53.
124. Zadeh LA. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans Syst Man Cybern* 1973;**SMC–3**:28–44.
125. Zadeh LA. A computational approach to fuzzy quantifiers in natural languages. *Comp & Maths with Appls* 1983;**9**:149–84.
126. Zimmermann HJ. Fuzzy Set Theory and Its Applications. Boston: Kluwer–Nijhoff, 1985.