Comparative Study of Methods for Linguistic Modeling of Numerical Data

A thesis submitted to the
Division of Research and Advanced Studies
of the University of Cincinnati in partial
fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in the Department of
Electrical and Computer Engineering
and Computer Science
of the College of Engineering

December 2002

by

Sofia Visa

Bachelor of Mathematics and Computer Science, June 1998

Lucian Blaga University, Sibiu, Romania.

Thesis Advisor & Committee Chair: Prof. Anca Ralescu
I dedicate this thesis to my parents

and to Dr. Anca Ralescu
Acknowledgments

First, I would like to thank my research advisor, Dr. Anca Ralescu, for her insight and wisdom during our work on this project, for her patience with me during the thesis writing stage, and for her unconditional support throughout. An excellent teacher and person, she is an inspiration for me.

The committee members, Dr. Ashraf Genaidy, Dr. Dan Ralescu and Dr. Dieter Schmidt, deserve many thanks for reviewing my thesis and for their advice.

I would like to thank our research group, Mircea Ionescu, Rinako Kamei, Prasanna Karunanayaka and Waibhav Tembe for their careful reading of this thesis; their suggestions have helped me a lot. Many thanks to Dr. Simon Yeung from Hong Kong Polytechnic University for making available the data set used in this study.

Special thanks must go to my friends Dr. Simona Doboli, Dr. Elek Doboli, Dr. Elena Teica, Dr. Rajesh Radakrishnan for all the encouragement and help in the course of this work and research.

I am deeply indebted to my parents for their continuous support.
Abstract

This thesis investigates the performance of four different classifiers on a common real data set. A review of the current classification models is presented along with their advantages and limitations. Four approaches to classifier design, a fuzzy set-based approach, neural network approach, support vector machine, and minimum distance classifier were implemented.

The data set used for comparing the performance of these classifiers consists of 4,232 samples. Its characteristics, such as high variability between samples within the same category and overlap between between categories, pose serious challenges for designing the classifier.

Several criteria are considered as a basis for evaluating a classifier performance, including the generalization power, the learning curve and ROC points. For classifier comparison, measures of diversity such as Yule Q statistic and the coefficient of correlation are used. Results of the evaluations are presented and analyzed in the light of the characteristics of the data set considered.

keywords: classifier, fuzzy systems, neural networks, support vectors machine, minimum distance classifier, ROC, confusion matrix, Yule statistic, bias-variance tradeoff.
# Contents

1 Introduction - background and motivation  
   1.1 Dimensionality reduction  
   1.2 Feature extraction  

2 Different models of training/testing  
   2.1 Major approaches in class predictions  
   2.2 Designing classifiers  
   2.3 Error estimation  
   2.4 Training versus Generalization  
   2.5 Comparing and Combining Classifiers  
       2.5.1 Diversity measures: The $Q$ statistic and correlation  

3 The current problem  
   3.1 Problem definition  
   3.2 The data set  
       3.2.1 Data preprocessing  

4 Fuzzy Systems  
   4.1 Determining the membership function  
   4.2 Fuzzy sets for the lifting task variables  
   4.3 Testing the fuzzy model  


4.4 Classification error ............................................. 32
4.5 Modeling error versus predictive error ....................... 34

5 Other classifiers .................................................... 36
5.1 Neural Networks .................................................. 36
5.2 Support Vector Machine-Perceptron ......................... 38
5.3 Minimum Distance Classifier .................................. 41

6 Comparison of classifiers ....................................... 46
6.1 Comparison criteria ............................................. 46
   6.1.1 Comparing the Fuzzy and Neural Network Classifier .. 46
6.2 Correlation between the Fuzzy and Neural Network Classifiers ................................. 48
6.3 Global Comparison of the classifiers ........................... 49
6.4 The ROC statistics ................................................ 51

7 Conclusions .......................................................... 54
7.1 Open problems ................................................... 55

Bibliography
# List of Figures

1.1 XOR problem; the 2D and 3D plots. ............................................. 3

2.1 Example: local versus global optimum. ................................. 9

2.2 The ROC curve. .............................................................. 15

2.3 Bias-Variance trade-off. ...................................................... 17

4.1 Frequency distribution (a), and the corresponding fuzzy sets (b, c, d), for the floor weight (FW) variable. ................................. 31

4.2 Frequency distribution (a), and the corresponding fuzzy sets(b, c, d), for the horizontal distance (HD) variable. ................................. 32

4.3 Frequency distribution and the corresponding fuzzy set for HD extended data (after data adjustment). ............................................. 33

4.4 Generalization power of the fuzzy set modeling ........................ 35

5.1 Neural network results. ....................................................... 37

5.2 Data characteristics for best results(FW variable). ..................... 38

5.3 Data characteristics for worst results(HD variable). .................... 39

5.4 Decision boundary in SVM classification between M and B classes for (a)training and (b)testing data for HD. ................................. 40

5.5 Decision boundary in SVM classification between M and B classes for (a)training and (b)testing extended data for HD. ......................... 41

5.6 Minimum Distance Classifier using Euclidean Distance. ............. 43
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.7</td>
<td>Minimum Distance Classifier for extended data using Euclidean Distance.</td>
<td>44</td>
</tr>
<tr>
<td>5.8</td>
<td>Minimum Distance Classifier using Sup Distance.</td>
<td>44</td>
</tr>
<tr>
<td>5.9</td>
<td>Minimum Distance Classifier for extended data using Sup Distance.</td>
<td>45</td>
</tr>
<tr>
<td>5.10</td>
<td>An example for which Minimum Distance Classifier is not a good approach.</td>
<td>45</td>
</tr>
<tr>
<td>6.1</td>
<td>Comparison of the prediction power for fuzzy sets and neural network.</td>
<td>47</td>
</tr>
<tr>
<td>6.2</td>
<td>The learning curve of all four classifiers for variable TA.</td>
<td>51</td>
</tr>
<tr>
<td>6.3</td>
<td>The ROC points of the four classifiers for class S (variable HD).</td>
<td>52</td>
</tr>
<tr>
<td>6.4</td>
<td>The ROC points of the four classifiers for class M (variable HD).</td>
<td>53</td>
</tr>
<tr>
<td>6.5</td>
<td>The ROC points of the four classifiers for class B (variable HD).</td>
<td>53</td>
</tr>
</tbody>
</table>
List of Tables

1.1 XOR problem; 2-dimensional data. ................................. 3
1.2 XOR problem; 3-dimensional data. ................................. 4

2.1 Confusion matrix ...................................................... 13
2.2 The quantities defining $Q$ and $\rho$ ................................. 18

3.1 Between-label and within-label variability for the variable horizontal distance (HD). ............................................. 22
3.2 The labels (B, M, S), the lifting variables(1), range(2), mean(3) and standard deviation(4). ............................................. 23
3.3 Breakdown of the data by variables and variable labels; floor level (FW), waist level (W), horizontal distance(HD), twisting angle (TA), frequency(F), work duration(WD), vertical distance(VD). ............................................. 24
3.4 Adjustment of variable intervals for the variable horizontal distance (HD): initial variable value (1), adjusted variable value (2) ............................................. 25

4.1 Fuzzy sets versus classical sets to model linguistic labels ................................. 27
4.2 Generalization error of the fuzzy model for original and extended data. For the variables FW and HD corresponding to best and worst case respectively, the error is shown for each label. ............................................. 34

5.1 Prediction of Perceptron for variable HD. ................................. 41
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2</td>
<td>Prediction of Minimum Distance Classifier for variable HD.</td>
<td>43</td>
</tr>
<tr>
<td>6.1</td>
<td>Complete results for fuzzy sets and neural networks</td>
<td>47</td>
</tr>
<tr>
<td>6.2</td>
<td>Prediction results for fuzzy sets and neural networks on the adjusted data</td>
<td>48</td>
</tr>
<tr>
<td>6.3</td>
<td>Confusion matrix for variable HD.</td>
<td>49</td>
</tr>
<tr>
<td>6.4</td>
<td>$Q$ and $\rho$ for the fuzzy sets and neural networks</td>
<td>49</td>
</tr>
<tr>
<td>6.5</td>
<td>The size of test set used to obtain the learning curves for variable HD (the original data set consists of 648 data points).</td>
<td>50</td>
</tr>
<tr>
<td>6.6</td>
<td>The computed FP an TP values used in ROC plots (variable TA).</td>
<td>52</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction - background and motivation

The primary goal of supervised or unsupervised learning is pattern recognition / classification. The design of a recognition system requires careful attention to several issues including: definition of pattern classes, sensing environment, pattern representation, feature extraction and selection, cluster analysis, classifier design and learning, selection of training and test samples and performance evaluation.

In supervised learning the system obtained should be able to produce the correct output $y^{(i)}$ when the input is $x^{(i)}$, where $(x^{(i)}, y^{(i)})_{i=1}^{N}$ is the given training set. However, this is only a partial requirement (had this been the only requirement, the best approach would use a look-up table). In addition to this, the system once constructed, is to produce the correct output for an input $x$ that was not part of the training data (the test data). A system which can do that is said to generalize well. When the output denotes a category (class label) the learning system is a classifier, when the output is a real number (assumes continuous values) the resulting system is an approximator.

When the output $y^{(i)}$ is not given at all the learning problem falls into the category of unsupervised learning. Patterns are learned by the system based on some measure
of similarity and optimization criteria. The current study is concerned with supervised learning, more exactly, it compares various approaches to supervised learning and analyzes their performance on a common real data set [18].

1.1 Dimensionality reduction

Many times the training data provided for a pattern extraction problem come from a large dimensional space. This means that in order to achieve effective generalization a very large amount of training data (usually not available) is required. This "curse of dimensionality" motivates the need to create a lower dimension feature space for use as input to the classifier. The reduced feature space can be constructed based on the knowledge of the problem or using other methods of dimensionality reduction. There are two main reasons to keep the dimensionality of the pattern representation (the number of features) as small as possible: measurement cost and classification accuracy. On the other hand, a reduction in the number of features may also lead to a loss in the discrimination power and thereby lower the accuracy of the resulting recognition system. In selecting features an important issue arises concerning the desirability (from modeling and prediction points of view) of any one of the features. A careful choice of the features is necessary since it is possible to make two arbitrary patterns similar by encoding them with a sufficiently large number of redundant features. Therefore, the main issue in dimensionality reduction is the choice of a criterion function for feature selection. A commonly used criterion is the classification error of a feature subset.

1.2 Feature extraction

It is important to distinguish between feature selection and feature extraction. The term feature selection refers to algorithms that select the best subset of the input feature set. Methods that create new features based on transformations or combinations of
the original feature set are called feature extraction algorithms. When patterns are not separable in the original feature space, a possible approach is to consider a higher dimension feature space where, hopefully, the patterns become separable (i.e. extract more features from each input). A very well known example to illustrate this idea is the ”XOR problem” detailed in Example 1.2.1.

**Example 1.2.1** The input to the XOR problem consists of a 2-dimensional vector \( x^{(i)} = (x_1^{(i)}, x_2^{(i)}) \), where \( x_j^{(i)} = 0 \) or 1 for \( j = 1, 2 \). The desired output is the ”exclusive-or” function of the two inputs; that is, the output is a 1 if either (but not both) of the two inputs are 1; and is 0 if either both are 1 or both 0.

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

*Table 1.1: XOR problem; 2-dimensional data.*

*Figure 1.1: XOR problem; the 2D and 3D plots.*
<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1.2: XOR problem; 3-dimensional data.

The data set generated by all possible assignments of $x^{(i)}$ is shown in table 1.1 and illustrated in Figure 1.1.

If one class is labeled 1 and the second 0 is obvious that the two classes are not linearly separable in the 2-dimension space (see figure 1.1). However, by considering one additional feature - for example $x_3 = x_1 \times x_2$ - then the classes become linearly separable as shown in figure 1.1. Now the input will be a 3-dimensional vector and the data set is shown in table (1.2).
Chapter 2

Different models of training/testing

2.1 Major approaches in class predictions

Several approaches for pattern recognition have been developed over the years [4]. Among them the best known include the following:

Template matching: In this approach a prototype (also called template) of the pattern to be recognized is available. The pattern to be recognized is matched against the stored template. The similarity measure, often a correlation, may be optimized based on the available training set. Moreover, the template itself may be learned from the training set. In general, template matching is computationally demanding, in addition to being a rigid and brittle approach. For example, when used in an image processing/image understanding the system, it may fail if the patterns are distorted due to imaging process, view-point change, or large intra-class variations among the patterns.

Statistical approach: Each pattern is represented in terms of $d$ features and is viewed as a point in the $d$-dimensional space. Given a set of training data points from
each class, the objective is to establish decision boundaries in the feature space which separate the data belonging to different classes. In the statistical decision theoretic approach, the decision boundaries are determined by the probability distributions underlying each class, which must either be specified or learned. An example of this kind of classifier is a Bayes classifier, also called an optimum statistical classifier because it minimizes the total average loss.

**Neural networks:** Viewed as massively parallel computational units, consisting of a (sometimes extremely) large number of simple processors with many interconnections, neural network models attempt to use some organizational principles (such as learning, generalization, adaptivity, fault tolerance and distributed representation and computation) in order to capture relationships between the data belonging to one class. Neural networks are represented as directed graphs in which the nodes are artificial neurons; the directed edges (with weights) are connections between neuron outputs and neuron inputs. The main differences between neural networks and the other approaches to pattern recognition are that these networks have the ability to learn complex non-linear input-output relationships. Moreover, they have the general characteristic of adapting themselves to the data.

The neural networks most commonly used for pattern classification tasks are feed-forward networks. These networks are organized into layers and have unidirectional weighted connections between the layers.

Neural networks provide nonlinear algorithms for feature extraction (using hidden layers) and classification (e.g. multilayers perceptrons). In training a multilayer neural network classifier, the network weights are corrected so that the overall error (the sum of squared errors between the network outputs and the desired outputs) is minimized. But since the decision boundaries between classes are not directly determined, the generalization ability depends on the training method
and it decreases when the number of training data is small. A feed-forward network with an associated training algorithm provides a powerful tool for function approximation and classification.

Note that each neuron establishes a linear decision boundary. A single layered network can only capture linearly separable pattern classification. To achieve classification of linearly non-separable data a multi-layer network must be used. However, if the neurons are linear in the multi-layered neural network, then this is equivalent to a single layered network. Hence for multi-layered network a non-linear activation function (at least in the hidden layer) must be used. A main reason for using a neural network classifier is the fact that it generalize well.

Limitations of neural networks include slow speed of training and possibility for the network to be stuck in a local minimum (failure to converge). Another difficulty concerns the choice of network size best suited for the problem at hand. This includes the number of nodes in the hidden layer, the activation function.

**Fuzzy Sets/Fuzzy Systems:** Another approach that has gained much popularity is based on the concepts of *fuzzy sets* and *fuzzy logic*. Systems based on the notions of fuzzy sets and fuzzy logic are referred to as *fuzzy systems*. Introduced in 1965 [11] fuzzy sets are meant to capture the meaning of concepts which cannot be defined in terms of necessary and sufficient conditions, and which have un-sharp boundaries. Like neural network approaches, fuzzy systems are universal approximators and establish implicitly class boundaries. The dual, symbolic/quantitative, semantics of fuzzy sets allows the system designer to translate his/her knowledge directly into some initial model for the concept (that is, to determine the membership function) and therefore is apt to improve the model convergence. On the other hand, work has been done in data driven learning of fuzzy models. In either case the advantage of the fuzzy approach is the ability to form a correspondence
between quantitative results and their qualitative interpretation. Fuzzy classifiers are fuzzy models of the classes under study and are usually selected when the training data is imprecise and/or incomplete, and when there is overlap between training data points for various classes. Chapter 4 of this study presents a review of the fuzzy system approach in connection with the work carried out for a fuzzy classifier.

2.2 Designing classifiers

The choice of a classifier is a difficult problem. It depends on many factors including the type of data to be classified, the availability of data for training and last, but not least, on the availability of classifiers and knowledge of these by the user.

The simplest and the most intuitive approach to classifier design is based on the concept of similarity which implements the idea that patterns which are similar should be assigned to the same class. In fact, in supervised learning, the fact that two data points are labeled by the same class label means that they must be similar in some sense: that is, in some metric, in some feature space (which may not be the original feature space). Part of the problem is to find these, the feature space and the metric. Defining corresponding similarity measures, patterns can be learned and then recognized using template matching or a minimum distance and a few prototypes per class. The choice of the metric and the prototypes is crucial to the success of this approach based on similarity. In the nearest mean classifier, selecting prototypes is very simple: each pattern class is represented by a single prototype which is the mean vector of all the training patterns in the class. However, this choice does not always lead to good results.

Another important approach used for designing classifiers is based on the probabilistic approach. The optimal Bayes rule (with the 0/1 loss function) assigns a pattern to the class with the maximum posterior probability. This rule can be modified to take into account costs associated with different types of missclassifications.
Finally, another approach is to construct **decision boundaries** directly by optimizing certain error criterion. The driving force of the training procedure is the minimization of a criterion such as the apparent classification error or the mean squared error between the classifier output and some preset target value. Most neural network approaches fall into this category. A classical example is the single-layer perceptron, where classes are separated by a hyperplane which is iteratively updated as a function of the distances of the misclassified patterns from the hyperplane. If the sigmoid function is used in combination with the mean squared error criterion, as in feed forward neural networks (also called multilayer perceptrons) classifier behaves as other linear classifiers [15]. It is important to note that neural networks can lead to many different classifiers depending on how they are trained. While the hidden layers in multilayer perceptrons allow nonlinear decision boundaries, they also increase the danger of overtraining the classifier since the number of networks parameters increases as more layers and more neurons per layer are added. Therefore, the regularization of neural networks may be necessary. Many regularization mechanisms such as slow training in combination with early stopping, are already built in. Other regularization methods include the addition of noise and weight decay [19].

![Diagram](image)

Figure 2.1: Example: local versus global optimum.

A special type of classifier is **the decision tree**, which is trained by an iterative selection of individual features that are most salient at each node of the tree. Dur-
ing classification, these features only are used for the test pattern under consideration. Therefore, in this approach, feature selection is implicitly built-in. The most commonly used decision tree classifiers are binary in nature and use a single feature at each node, resulting in decision boundaries that are parallel to the feature axes [8]. Consequently, such decision trees are intrinsically suboptimal for most applications. In general, decision trees achieve a local optimum and not the global optimum. For example figure 2.1 illustrates a data set for which a decision tree classifier it takes three steps to separate, while a neural network can achieve it in two steps.

The main advantage of the tree classifier, besides its speed, is the possibility to interpret the decision rule in terms of individual features. Like neural networks, decision trees can be easily overtrained. When this happens, post-pruning the decision tree can be used to avoid overfitting the data [13].

An interesting recent development in classifier design is the introduction of the support vector classifier [22] which evolved from the theoretical work on statistical learning done by Vapnik. It is primarily a two class classifier (a multi-class problem is converted into several two-class problems) seeking to separate the two classes by a hyperplane. The optimization criterion here is the width of the margin between the classes and the separating hyperplane. The margin is defined as the distance of the nearest training data points to the separating hyperplane. These data points, called support vectors, finally define the classification function. The number of support vectors is determined by maximizing the margins. The support vector classifiers aim at honing into the optimal hyperplane which is found as a solution of the margin optimization problem and therefore it is apt to have the best generalization among those classifiers which seek linear separability.

Each of the classifiers discussed above is admissible, in the sense that, given a classifier, there exists some classification problem for which this classifier is the best choice. It is showed [7] that there is no overall optimal classification rule. This motivates the
present comparative study for the particular set of data described in [18].

2.3 Error estimation

Data is almost always limited. On the other hand, obtaining unbiased and accurate estimates of the error of a classifier requires a fair amount of data. To some extent, lack of data and/or lack of knowledge about its other (statistical) properties can be compensated by good models on training, testing and error evaluation.

The classification error, or simply the error test rate, is the ultimate measure of the performance of a classifier. In practice, the error rate of a recognition system must be estimated from all the available samples which are split into training and test sets [13]. The classifier is first designed using training samples, and then is evaluated based on its classification performance on the test samples. The percentage of misclassified test samples is taken as an estimate of the error rate. In order for this error estimate to be reliable in predicting future classification performance, the training set and the test set must be sufficiently large and the training samples and the test samples must be representative and independent.

Another issue concerns the decision of the available samples to form training and test sets. If the training set is small, then the resulting classifier will not be very robust and will have a low generalization ability. On the other hand, if the test set is small, then the confidence in the estimated error rate will be low. Various methods that are commonly used to estimate the error rate include the following:

**Bootstrap** uses resampling (often with substitution) to generate the training data.

In effect, the underlying principle is to use computational power in lieu of large data sets.

**Leave-one-out** uses one of the N data points for testing and the remaining N – 1 for training. This is done N times. Leave-one-out is an unbiased estimator of the
expected error rate but has large variance.

**Generalized k-fold cross validation:** In this approach the data is partitioned into $k$ equal groups, $k - 1$ of which are used for training with the remaining groups used for testing. The procedure is iterated $k$ times, each time using a unique partition for testing. For $k = N$ this coincides with the Leave-one-out method. A weakness of this estimator is the variability within the results according to the value of $k$ selected.

**Holdout method:** Here half the data is used for training and the remaining data is used for testing; training data and test data are independent. This is considered to be a pessimistically biased estimate. Also, different partitioning will give different estimates.

**The confusion matrix:** In many pattern recognition applications, it is not adequate to characterize the performance of a classifier only by the global error rate of a system. The *confusion matrix* contains information about actual and predicted classification done by a classification system. Table 2.1 shows the confusion matrix for a two class classifier. The entries in the confusion matrix have the following meaning:

- $a$ is the number of correct negative predictions;
- $b$ is the number of incorrect positive predictions;
- $c$ is the number of incorrect negative predictions;
- $d$ is the number of correct positive predictions;

Six measures of classifier performance are defined based on the confusion matrix entries as follows:

1. the *accuracy* ($AC$) is the proportion of the number of correct predictions that
<table>
<thead>
<tr>
<th></th>
<th>Predicted</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Negative</td>
<td>Positive</td>
</tr>
<tr>
<td>Actual</td>
<td>Negative</td>
<td>$a$</td>
</tr>
<tr>
<td></td>
<td>Positive</td>
<td>$c$</td>
</tr>
</tbody>
</table>

Table 2.1: Confusion matrix

is:

$$AC = \frac{a + d}{a + b + c + d}$$

2. the *true positive rate* (TP) is the proportion of positive cases that were correctly identified, that is:

$$TP = \frac{d}{c + d}$$

3. the *false positive rate* (FP) is the proportion of negatives cases that were incorrectly classified as positive:

$$FP = \frac{b}{a + b}$$

4. the *true negative rate* (TN) is defined as the proportion of negatives cases that were classified correctly:

$$TN = \frac{a}{a + b}$$

5. the *false negative rate* (FN) is the proportion of positive cases that were incorrectly classified as negative:

$$FN = \frac{c}{c + d}$$
6. the *precision* ($P$) is the proportion of the predicted positive cases that were correct:

$$P = \frac{d}{b + d}$$

The **ROC graph** : The quantities calculated based on the confusion matrix entries are used to define the ROC (Receiver Operating Characteristic) graph: a point on the ROC graph is a plot with the false positive rate, $FP$, on the $x$-axis and the true positive rate, $TP$, on the $y$-axis. An ROC curve is independent of class distribution. It encapsulates all information contained in the confusion matrix and provides a visual tool for examining the tradeoff between the ability of a classifier to correctly identify positive cases and the number of negative cases that are incorrectly classified. More precisely, the point $(0, 1)$ is the perfect classifier: it classifies all positive cases and negative cases correctly. The point $(0, 0)$ represents a classifier that predicts all cases to be negative, while the point $(1, 1)$ corresponds to a classifier that predicts every case to be positive. The point $(1, 0)$ is the classifier that is incorrect for all classifications. In many cases, a classifier has a parameter that can be adjusted to increase $TP$ at the cost of an increased $FP$ or decrease $FP$ at the cost of a decrease in $TP$. Each parameter setting provides a $(FP, TP)$ pairs and a series of such pair can be used to plot an ROC curve. A non-parametric classifier is represented by a single ROC point, corresponding to its $(FP, TP)$ pair. Figure 2.2 shows an example of an ROC graph for two classifiers. From the discussion above it is obvious that for a classifier, in order to be acceptable, its corresponding ROC graph should be in the upper shaded area of the figure.

The **learning curve**: The learning curve is a plot of classifier predictive power when different size training data sets are used. In general, given a data set with $N$
elements, \( p \% \) of \( N \) are used for training with the remaining \((100 - p)\%\) used for testing. The learning curve is obtained by plotting the predictive power of the classifier, that is, the percentage of correct classification on the test set, when \( p \) varies (for example, \( p \) can take values in \{10, 20, 30, \cdots, 90\}). A smooth learning curve with a rapidly increasing slope is preferable.

### 2.4 Training versus Generalization

It is now widely accepted that no single procedure will completely solve a complex classification problem [7]. On the other hand, as already mentioned, there are many admissible approaches, each capable of discriminating patterns in certain portions of the feature space. The combination of classifiers has, therefore, become a heavily studied topic [6].

However, regardless of the classification or decision rule selected, the classifier must be trained using the available training samples. As a result, the performance of clas-
sifier depends on both the number of available training samples as well as the specific values of the samples. Therefore, optimizing a classifier to maximize its performance on the training set may not always result in the desired performance on a test set. The generalization ability of a classifier refers to its performance on test data which were not used during the training stage. A poor generalization ability of a classifier can be attributed to any of the following factors:

(i) the number of features is too large compared to the number of training samples (curse of dimensionality [3]);

(ii) the number of unknown parameters associated with the classifier is large (e.g., polynomial classifiers or a large neural network);

(iii) a classifier is too intensively optimized on the training set (overtrained); this is analogous to the phenomenon of overfitting in regression when there are too many free parameters.

Overtraining has been investigated theoretically for classifiers that minimize the apparent error rate (the error on the training set). The classical studies by Cover [19] and Vapnik [21] on classifier capacity and complexity provide a good understanding of the mechanism behind overtraining. Complex classifiers (e.g., those having many independent parameters) may have a large capacity, i.e. they are able to represent many dichotomies for a given dataset. A frequently used measure for the capacity is the Vapnik-Chervonenkis (VC) dimension [21].

While an exact relationship between the probability of misclassification, the number of training samples, the number of features and the true parameters of the class-conditional densities is very difficult to establish, some guidelines have been suggested regarding the ratio of the sample size to dimensionality. It is generally accepted that using at least ten times as many training samples per class as the number of features \((n/d > 10)\) is a good practice to follow in classifier design [3]. The more complex
the classifier, the larger should be this ratio in order to avoid problems due to high dimensionality.

The conflict arising from trying to learn well and generalize well is known as the bias-variance trade off [14]. For example, for an approximator whose output is a constant (irrespective of the training data) the bias is high as no attention is paid to the data but the variance is 0. On the other hand, for a much complex approximator the bias would be small but there would be high variance (with different data set the approximation would be greatly different). These situations are illustrated in figure 2.3.

![Bias-Variance trade-off](image_url)

Figure 2.3: Bias-Variance trade-off.

For a classifier like neural network, for example, an increased number of hidden neurons would be more sensitive to the noisy data, i.e. for a small noise (change) in the input we get a large change in the output, so the variance increases (for small variations in the training sets the network output will be very different). But the bias decrease leads to better approximation results.

If the data set is small there is always a bias-variance trade-off; if the data set is large (infinite) bias and variance can be reduced simultaneously. Now the question is, which one of these should be reduced? The answer is both but for a small data set it is impossible. Sometimes, a bias is introduced voluntarily, if it allows variance to reduce significantly.
2.5 Comparing and Combining Classifiers

As it is clear from the preceding, given a classification problem, several approaches are available. Classifiers can be compared on the curves ROC and learning curves which convey information on their predictive performance. However, it is conceivable for two classifiers to exhibit similar performance from the point of view of their percentages of error, while, being actually quite different. Such classifiers are complementary in the sense that their prediction error occurs on a different subset of the test data. Such observations have lead researchers to consider the issue of combining classifiers [6], in a synergistic way the result being a classifier that performs better than its components. A significant improvement in classification performance for a combination of classifiers, depends on the complementarity of the combined classifiers, that is, on how much they agree and disagree. To evaluate this, measures other than the ROC and learning curves are necessary.

2.5.1 Diversity measures: The $Q$ statistic and correlation

The ROC and learning curves provide qualitative measures for classifiers comparison. Classifiers are preferred according to the slope of these curves: the sharper the slope, the better the classifier. Quantitative measures for comparison of classifier performance include the Yule statistic, $Q$, and the coefficient of correlation, $\rho$. In the following $C1$ and $C2$ denote two different classifiers and $a$, $b$, $c$ and $d$, such $a + b + c + d = 1$, the quantities associated to their performance, as described in table (2.2). The $Q$ statistic and the correlation coefficient $\rho$ are defined by (2.1) and (2.2) respectively.

<table>
<thead>
<tr>
<th>$C2_{correct}$</th>
<th>$C1_{correct}$</th>
<th>$C1_{wrong}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>$b$</td>
<td></td>
</tr>
<tr>
<td>$c$</td>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2: The quantities defining $Q$ and $\rho$
\[ Q = \frac{ad - bc}{ad + bc} \]  

(2.1)

\[ \rho = \frac{ad - bc}{\sqrt{(a + b)(c + d)(a + c)(b + d)}} \]  

(2.2)

It can be easily seen that \(-1 \leq \rho \leq Q \leq +1\). Both convey correlation between the classifiers, with values close to +1 indicating high correlation between the two classifiers. The Yule statistic and the coefficient of correlation are both used in this study to compare the classifiers under consideration.
Chapter 3

The current problem

3.1 Problem definition

With the background of the previous two chapters, the problem considered in the remaining part of this thesis consists in designing and implementing four different classifiers with the goal of comparing their results when applied to a particular data set. All approaches considered are instances of supervised learning. While benchmark data sets for assessing classifier performance exist, the data set considered in this thesis comes from a real world application in the domain of safety evaluation in a physical work environment and it poses certain challenges. This make this study interesting in several ways, as follows:

From a theoretical point of view, by considering a specific data set the work presented here points out to some difficulties when non-standard data sets must be processed (in this case classified) and how some of these difficulties can be solved, in order to obtain an accurate and robust solution;

From an application point of view, the study carried out is among the few of its type, attempting to introduce ideas from system modeling in the application field of work environment safety evaluation and, on a larger scale, in the field of
ergonomics, where the quantitative methods used usually include mostly standard statistical methods.

The data set used for the study and its characteristics are presented next. The data set consists of verbal descriptions of perceptions of the physical characteristics of a lifting task. A full and detailed description of the data set can be found in [16]. In the following, a description of its characteristics to the extent needed to formulate the classifier modeling problem is provided.

Two hundred seventeen manual workers were interviewed in order to collect the data, as part of a study on work environment safety carried out in collaboration at University of Cincinnati and Hong Kong Polytechnic University [17], [18]. The data were collected from a questionnaire filled out by manual workers whose daily activities require them to lift various weights. Subjects were required to imagine various lifting conditions (described verbally) on several aspects (variables) of the lifting task.

Seven variables were used to describe the lifting task as follows:

1. Floor Weight (FW),
2. Waist Weight (W),
3. Horizontal Distance (HD),
4. Twisting Angle (TA),
5. Frequency (F),
6. Work Duration (WD),
7. Vertical Distance (VD).

The values of the lifting tasks variables were assessed by each individual and labeled as Small (S), Medium (M) or Big (B). Therefore, the total data set consists of $217 \times 7 \times 3 = 4557$ samples.
Table 3.1 illustrates the kind of data used for this problem. This table contains actual entries for the variable \( HD \). Column 1 contains the label describing the values, recorded as intervals, in column 2, of a typical lifting task variable. The classifier problem is then stated as follows: for each lifting task variable, given the available data set of label-value correspondences on a subset of the data set, find a model that can capture this correspondence as accurately as possible and which can be used to predict the verbal category for a given numerical description of the input.

<table>
<thead>
<tr>
<th>Label</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>30 - 40</td>
</tr>
<tr>
<td></td>
<td>40 - 55</td>
</tr>
<tr>
<td></td>
<td>28 - 32</td>
</tr>
<tr>
<td>M</td>
<td>40 - 65</td>
</tr>
<tr>
<td></td>
<td>50 - 60</td>
</tr>
<tr>
<td></td>
<td>35 - 45</td>
</tr>
<tr>
<td>B</td>
<td>33 - 45</td>
</tr>
<tr>
<td></td>
<td>50 - 59</td>
</tr>
<tr>
<td></td>
<td>48 - 63</td>
</tr>
</tbody>
</table>

Table 3.1: Between-label and within-label variability for the variable horizontal distance (HD).

### 3.2 The data set

The data in Table 3.1 shows also the variability and similarity between and within the three labels describing the variable \( HD \). The overlap between two different labels and variability within one label is not restricted to the variable \( HD \). Table 3.2 shows a summary of the data for all seven variables which include the range, mean and standard deviation for each label of each variable.

#### 3.2.1 Data preprocessing

The major difficulty in processing this type of data arises from the two aspects discussed above: on one hand, the data exhibits variability within a given label; on the other hand,
<table>
<thead>
<tr>
<th>Labels</th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FW</td>
<td>10-45</td>
<td>18.2-24</td>
<td>3.0-3.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4-25</td>
<td>9.8-15.9</td>
<td>2.2 - 2.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.27-14</td>
<td>2.7 - 7.4</td>
<td>0.9-2.1</td>
<td></td>
</tr>
<tr>
<td>W</td>
<td>10-40</td>
<td>19.2-24.2</td>
<td>2.87-2.82</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5-25</td>
<td>11.1-17.2</td>
<td>2.5-2.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.27-17</td>
<td>3-8.7</td>
<td>1.5-2.6</td>
<td></td>
</tr>
<tr>
<td>HD</td>
<td>33-75</td>
<td>52.7-65.3</td>
<td>8.1-9.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25-70</td>
<td>38.3-49.6</td>
<td>5.5-7.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25-55</td>
<td>25.5-35.2</td>
<td>1.7-5.2</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>21-180</td>
<td>56-78.3</td>
<td>13.6-16.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11-80</td>
<td>32.3-51.4</td>
<td>8.6-12.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10-51</td>
<td>10.8-27.1</td>
<td>2.6-7.6</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>4-4.5</td>
<td>11.5-16.5</td>
<td>3.7-4.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1-40</td>
<td>5.9-10.5</td>
<td>2.6-3.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0-16</td>
<td>0.6-4.8</td>
<td>1.3-2.4</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>1-8</td>
<td>4.3-6.4</td>
<td>1.4-1.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.5-7</td>
<td>2.2-3.8</td>
<td>0.9-1.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.3-3</td>
<td>0.3-1.6</td>
<td>0.3-0.8</td>
<td></td>
</tr>
<tr>
<td>VD</td>
<td>55-200</td>
<td>106.8-146.7</td>
<td>23.9-30.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>36-175</td>
<td>54.3-98.7</td>
<td>17.2-22.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25-205</td>
<td>27.4-55.7</td>
<td>7.4-14.8</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: The labels (B, M, S), the lifting variables(1), range(2), mean(3) and standard deviation(4).

data for two different labels may be identical. These reflect the variability among, and the subjectivity of the individuals evaluating a given lifting task and therefore, they cannot be easily eliminated. Nor should they be only to make processing easier.

Still, a mild preprocessing step was applied which eliminated entries deemed as incorrect. Such entries include those in which one of the endpoints of the interval for values of the variable was not specified, or the endpoints were specified in reverse order.

The final data set, shown in Table 3.2, for each variable, has 4,232 entries. The holdout method of Chapter 2, according to which, half of the data set, selected randomly, is used to model the classifier; the remaining data are used with each classification model for testing purposes.

Further, a rationality assumption is introduced as follows: when a subject esti-
<table>
<thead>
<tr>
<th>Variable</th>
<th>Total data points</th>
<th>Data per label</th>
<th>Data sets for training and testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>FW</td>
<td>512</td>
<td>S: 176, M: 168, B: 168</td>
<td>256 train, 256 test</td>
</tr>
<tr>
<td>W</td>
<td>480</td>
<td>S: 160, M: 160, B: 160</td>
<td>240 train, 240 test</td>
</tr>
<tr>
<td>HD</td>
<td>648</td>
<td>S: 216, M: 216, B: 216</td>
<td>324 train, 324 test</td>
</tr>
<tr>
<td>TA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VD</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3: Breakdown of the data by variables and variable labels; floor level (FW), waist level (W), horizontal distance(HD), twisting angle (TA), frequency(F), work duration(WD), vertical distance(VD).

mates the interval \([a, b]\) as his (the subjects were all male) perception of the concept *small/light* assume that this subject will also accept any interval \([x, b]\) for any \(x \leq a\). Indeed, it is possible that the reason for which the subject produced the interval \([a, b]\) is that for values \(x < a\) this subject would have associated another label, e.g. *very small*. Yet in the experiments which produced the data for this study there was no other option except that of using labels from the set \(\{S, M, B\}\). In addition, *very small* commonly subsumes *small* so the assumption is justified. A similar adjustment, of the upper limit of the interval, is done for intervals which are used to estimate the label *big/large*. Intervals corresponding to the label *medium* are not adjusted. Equation (3.1) describes the above adjustment procedure.

\[
adj([a, b], L) = \begin{cases} 
(m_L, b) & \text{if } L = S \\
(a, b) & \text{if } L = M \\
(a, M_L) & \text{if } L = B 
\end{cases} 
\]
where $m_L = \min_L \{a; [a, b]\}$ and $M_L = \max_L \{b; [a, b]\}$.

<table>
<thead>
<tr>
<th>Label</th>
<th>(1)</th>
<th>(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>30 - 40</td>
<td>25 - 40</td>
</tr>
<tr>
<td></td>
<td>40 - 55</td>
<td>25 - 55</td>
</tr>
<tr>
<td></td>
<td>28 - 32</td>
<td>25 - 32</td>
</tr>
<tr>
<td>M</td>
<td>40 - 65</td>
<td>40 - 65</td>
</tr>
<tr>
<td></td>
<td>50 - 60</td>
<td>50 - 60</td>
</tr>
<tr>
<td></td>
<td>35 - 45</td>
<td>35 - 45</td>
</tr>
<tr>
<td>B</td>
<td>33 - 45</td>
<td>33 - 75</td>
</tr>
<tr>
<td></td>
<td>50 - 59</td>
<td>50 - 75</td>
</tr>
<tr>
<td></td>
<td>48 - 63</td>
<td>48 - 75</td>
</tr>
</tbody>
</table>

Table 3.4: Adjustment of variable intervals for the variable horizontal distance (HD): initial variable value (1), adjusted variable value (2)

Table (3.4) illustrates the result of the adjustment for the data in Table 3.1. From this point onwards the adjusted data is referred to as the extended-data.
Chapter 4

Fuzzy Systems

Due to the variability of the data within one class on one hand, and some overlap between different classes on the other hand, it is appropriate to try a fuzzy classifier. The real data set used in this study reflects the perception of the lifting task variables by manual workers whose activity consists in frequent lifting. The values of the lifting task variables are described by words rather than exact numbers. For example, weight of load (W) is a variable which determines the difficulty of lifting. Verbal descriptions of measurements, such as light, medium light, heavy are natural to people. Yet, these descriptions are subjective, the quantities which one description applies to vary, and different such descriptions may describe, eventually to a different degree, the same value.

If a computer based approach is to capture the correspondence between the actual values of a variable and their linguistic description, a paradigm that accommodates both the numeric and symbolic/linguistic descriptions of values, that allows for categories to be both distinct and somewhat overlapping, that distinguishes between members of the same category is needed. Fuzzy sets theory [11] is one such paradigm which has proved successful in many similar problems [5] and is adopted in this study. Although the use of fuzzy sets for the purpose described above is not new, their use in this particular application is and it is hoped that this study contribute to the use of fuzzy set theory
in the field of ergonomics.

Introduced in 1965 [11], fuzzy sets aim to capture and describe concepts that do not have sharp boundaries. Such natural concepts, abound in the real world and are to be distinguished from man-made, technical (mathematical) concepts defined in terms of necessary and sufficient conditions. Given a universe of discourse, \( S \), a (classical) set \( A \) is identified by an indicator function, \( I_A : S \rightarrow \{0, 1\} \), such that \( I_A(x) = 1 \) if \( x \in A \) and \( I_A(x) = 0 \) otherwise. Similarly, a fuzzy set, \( \tilde{A} \) is identified by a membership function, \( \mu_{\tilde{A}} : S \rightarrow [0,1] \) such that \( \mu_{\tilde{A}}(x) \) is the degree to which \( x \) is in \( \tilde{A} \). Fuzziness arises in the process of assigning linguistic descriptions to numerical values, and is due what can be called *lack of definition* of the linguistic descriptions. By contrast, probabilistic or statistical uncertainty arises in the process of summarization of numerical data and is due to *lack of data*.

Fuzzy sets and classical sets do share certain similarities but they are also different in what they can capture. Table (4.1) summarizes the features of these two representations.

<table>
<thead>
<tr>
<th>Features</th>
<th>Fuzzy Sets</th>
<th>Classical Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overlapping</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Exhaustive</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Between-label distinction</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Within-label distinction</td>
<td>✓</td>
<td>×</td>
</tr>
</tbody>
</table>

Table 4.1: Fuzzy sets versus classical sets to model linguistic labels

4.1 Determining the membership function

An important issue in connection with fuzzy sets is that of determining their membership function. It can be seen that, given a label, e.g. *tall* for the variable height, the
membership function that describes it is not necessarily unique. More precisely, while it is accepted that for heights under a certain value, $\alpha$, the membership function is 0, for heights exceeding a value $\beta > \alpha$ it is 1, and it is expected to be nondecreasing between $\alpha$ and $\beta$, the exact values as well as exact definition in the interval $[\alpha, \beta]$ are not fixed, their choices reflecting various conditions under which the concept tall may be considered.

Without doubt there is a statistical aspect underlying the notion of a fuzzy set, in the sense that several different instances must have been given as examples of the concept to be represented as a fuzzy set (alternatively, the notion of degree of membership seems to make little sense, when all that is known about such a concept is only one instance). It is therefore natural to relate the notion of fuzziness to the notion of frequency and, moreover, to do so in a formal manner. This idea is at the basis of work done by [9], in which a formal correspondence between fuzzy sets on a universe of discourse and a probability distribution on it is established. The same was further studied in [1] and was the basis for an approach to derive subjective membership functions [2], [10].

For the current problem the basic mechanism for converting a relative frequency distribution (discrete probability distribution) into fuzzy sets is used. This mechanism can be presented independently of the more advanced aspects of the general theory.

In the following, $A$ denotes a discrete fuzzy set with membership function $\mu_A(x_i)$, $i = 1, ..., n$. For $0 < \alpha \leq 1$ the $\alpha$-level, $A_\alpha$ is the crisp set defined as $A_\alpha = \{x | \mu_A(x) \geq \alpha\}$. It can be easily seen that $(A_\alpha)_{0 \leq \alpha \leq 1}$ is a family of nested classical (non-fuzzy) sets, such that if $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n$, and $A_i$ their corresponding level sets, $A_1 \in A_2 \in \cdots \in A_n$. It has been shown that, for a fuzzy set, $A$, $\mu_A(x) = \sup_{\alpha} \alpha I_{A_\alpha}(x)$. On the other hand, the representation theorem [20] states the conditions (quite general) under which a collection of classical sets correspond to the levels of a fuzzy set. The correspondence between the levels sets and a fuzzy set has been traditionally used to derive mathematical properties of fuzzy sets from the ones already known for classical sets. In this thesis,
this correspondence is used to derive the fuzzy set membership function, under certain assumptions on the level sets.

When $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$, and $A_i$ is the level set corresponding to $\alpha_i$ it can be verified that $\sum_{i=1}^{n}(\alpha_i - \alpha_{i+1}) = \alpha_1$. In particular, when $\alpha_1 = 1$, that is for a normal fuzzy set, $\sum_{i=1}^{n}(\alpha_i - \alpha_{i+1}) = 1$.

Since $0 \leq \alpha_i - \alpha_{i+1} \leq 1$ it follows that $(\alpha_i - \alpha_{i+1})$ can be viewed as a discrete probability distribution. This observation is the starting point of deriving the membership function of a fuzzy set when examples of its members are given.

Let $\mu_{(i)}$ denote the values of $\mu_A(x_i)$ arranged in nondecreasing order and $A_i$ the level set corresponding to $\mu_A(x_i)$. For simplicity, it is assumed that $\mu_{(i)} = \mu_A(x_i)$. Then $A_i = \{x_1, \ldots, x_i\}$. Let $p_{ik}$ denote a selection rule from the level $A_i$ as defined in [10]. That is, $p_{ik} = p_i(x_k)$, $k = 1, \ldots, i$, denotes the probability of selecting the element $x_k$ from the level set $A_i$ such that $0 \leq p_{ik} \leq 1$ and $\sum_{k=1}^{i} p_{ik} = 1$. Let $m : \mathcal{P}(\mathcal{S}) \rightarrow [0,1]$ be defined as

$$m(A) = \begin{cases} 
\alpha_i - \alpha_{i+1} & \text{if } A = A_i \\
0 & \text{otherwise}
\end{cases} \quad (4.1)$$

and as in [2], let $f_k : \mathcal{S} \rightarrow [0,1]$ be defined as

$$f_k(x) = \sum_{i=1}^{n} m(A_i) \times p_{ik}(x) \quad (4.2)$$

It can be easily verified that $0 \leq f_k(x) \leq 1$ and that $\sum_{k=1}^{n} f_k(x) = 1$. That is, $\{f_k\}_k$ is a probability distribution on $\{x_1, \ldots, x_n\}$. It can also be seen easily that $\{x | f_k(x) \geq f_k(x_i)\} = A_i$. As shown in [10] (4.2) can be used to derive the membership function $\mu_A$ when the frequency $f_k$ and the selection rules $p_{ik}$ are known (setting $f_{n+1} = 0$). Two special cases for $p_{ik}$ yield:
1. *least prejudiced distribution* (*lpd*) when \( p_{ik} = \frac{1}{\mid M \mid} = \frac{1}{n} \), in which case

\[
\mu(k) = k f(k) + \sum_{i=k+1}^{n} f(i), \text{ for } k = 1, \ldots, n
\]  

(4.3)

and

2. *the most prejudiced distribution* (*mpd*) when \( p_{ik} = \delta_{ik} \), where \( \delta_{ik} = 0 \) if \( i = k \) and 0 otherwise, in which case

\[
\mu(k) = \sum_{i=k}^{n} f(i), \text{ for } k = 1, \ldots, n
\]  

(4.4)

In equations (4.4) and (4.3) \( f(k) \) and \( \mu(k) \) denote the \( k \)th largest value of the frequency distribution and membership function respectively. Both of these equations were known in the literature of fuzzy sets for some time (see for example [9] for the (*lpd*) case). However, a discussion of the general case, and the fact that equations (4.3) and (4.4) are special cases of it, can be found in [2], [1].

In this study (4.3) is used to derive the membership functions for fuzzy sets, representing the labels \( S, M, \) and \( B \).

### 4.2 Fuzzy sets for the lifting task variables

Using the mechanism outlined above, the data for seven lifting task variables were assigned to three linguistic labels on their corresponding universe of discourse, represented as fuzzy sets. Half, selected randomly, are used to model the fuzzy sets and train the neural network. The remaining are used with each model for testing purposes.

Figures 4.1 and 4.2 show the frequency distributions and the fuzzy sets for the variable FW and HD.

Figure 4.3 shows the resulting frequency distribution and fuzzy sets for the variable HD using extended data. The recognition procedure remains the same as for the case
Figure 4.1: Frequency distribution (a), and the corresponding fuzzy sets (b, c, d), for the floor weight (FW) variable.

of the original data.

4.3 Testing the fuzzy model

The fuzzy sets obtained are used for classification of a given data point as follows. Given the value $x$, a point or interval value, for a lifting variable the following steps are applied:

1. **Match $x$ to each label**: Calculate the degrees $\mu_S(x)$, $\mu_M(x)$, $\mu_B(x)$.

2. **Assign $x$ to a label**: Based on the degrees computed at the preceding step assign $x$ to that category for which its degree is highest. More precisely, if $pred_j$ denotes
Figure 4.2: Frequency distribution (a), and the corresponding fuzzy sets (b, c, d), for the horizontal distance (HD) variable.

the label predicted for the data $x$, then

$$\text{pred}_L = \text{arg}\{\max_{y \in \{S,M,B\}} \mu_L(x)\}$$

(4.5)

4.4 Classification error

Due to the variability of descriptions among various subjects, and to the overlap between various labels, a given value $x$ may be misclassified. To track the errors of classification
a simple error model (4.6) is used.

$$\text{error}(x) = \begin{cases} 0 & \text{pred.} J(x) = \text{actual} J(x) \\ 1 & \text{otherwise} \end{cases}$$ (4.6)

For a collection of data, $V$, the overall error is simply the average number of errors (4.7).

$$\text{error}(V) = \frac{\sum_{x \in V} \text{error}(x)}{|V|} \times 100\%$$ (4.7)

It should be noted that the fuzzy set model allows for a finer measure of error, based on the degree of a classification; however for the purpose of comparing its results with those of other methods, where such a finer measure of error cannot be obtained (unless
introduced in an *ad-hoc* manner) this simple model is adopted.

### 4.5 Modeling error versus predictive error

For fuzzy models (as well as in other approaches, e.g. neural networks) the error of classification cannot be excluded even for the training data. This is known as the *modeling error* and the corresponding correct classification of the training data is known as *modeling power*. For the test data these are called *generalization error* and *generalization power* respectively. The idea behind modeling approaches such the one used in this study is to *give up some of the modeling power, in order to achieve acceptable levels of generalization power of the model*. Table 4.2 shows the generalization error of the fuzzy

<table>
<thead>
<tr>
<th>Variable</th>
<th>test data</th>
<th>extended test data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FW</strong></td>
<td>7.43 =</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.9 – B</td>
<td>4.6</td>
</tr>
<tr>
<td></td>
<td>2.6 – M</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.93 – S</td>
<td></td>
</tr>
<tr>
<td>W</td>
<td>8.42</td>
<td>4.75</td>
</tr>
<tr>
<td>TA</td>
<td>14.42</td>
<td>8.34</td>
</tr>
<tr>
<td>VD</td>
<td>14.64</td>
<td>8.05</td>
</tr>
<tr>
<td>D</td>
<td>17.39</td>
<td>12.6</td>
</tr>
<tr>
<td>F</td>
<td>19.55</td>
<td>13</td>
</tr>
<tr>
<td><strong>HD</strong></td>
<td>22.42 =</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12.7 – B</td>
<td>8.5</td>
</tr>
<tr>
<td></td>
<td>7.6 – M</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 – S</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2: Generalization error of the fuzzy model for original and extended data. For the variables FW and HD corresponding to best and worst case respectively, the error is shown for each label.

sets based model for the seven variables considered. Figure 4.4 shows the corresponding generalization power. The modeling error for the two variables highlighted in Table 4.2, **FW** (lowest error) and **HD** (largest error), are 7.43% and 22.40% respectively. Note that these are cumulative errors for each variable, and that within each label, the errors are much smaller. It should also be noted that the fuzzy model loses very little from its power when it is applied to test data.
Figure 4.4: Generalization power of the fuzzy set modeling

The overlap between data evident from Table 3.2 indicates how fuzzy the labels and therefore, the underlying fuzzy sets, are. It can be seen that the extent to which data corresponding to different labels overlap in the original data set affects the accuracy of the prediction.
Chapter 5

Other classifiers

As the results of the previous chapter indicate the fuzzy set based classifiers are accurate within acceptable error rates. For comparison purposes, other classifiers, including neural networks, support vector machines (perceptron) and minimum distance based, have been applied to the data set considered in this study. Although results have been obtained for the seven variables, detailed discussion is included only for the variable HD for which was the largest overlap and hence the most difficult to classify.

5.1 Neural Networks

Multi-layer neural networks are universal classifiers in that they can classify any labeled data correctly if there are no identical data in different classes. For this study a Feed Forward Neural Network (NN) was trained to capture the meaning of the linguistic labels for each variable. One hidden layer of ten neurons, sigmoidal activation function, back-propagation learning algorithm, training error of 0.1, two input values and one output were used. The training and test data are shown in Table 3.2. The input values correspond to the endpoints of the interval given as data to be assigned to a linguistic label. The output, corresponding to the class label, is encoded as $-1$ for Small, 0 for Medium and $+1$ for Big respectively. A recurrent neural network approach would not
be helpful for this data because the order in which the data points are presented to the network does not matter.

After training, for each input data the neural network outputs a value in the interval [-1, 1]. A particular test data, the endpoints of an interval, is classified to the label $S$, $M$, $B$ according to whether the corresponding output is close enough to -1, 0, or +1 respectively. This rule of thumb is implemented simply by selecting a threshold $\gamma$ such that if $NN_V(x)$ denotes the neural network output for the variable $V$, at $x$, $x$ is assigned the label $L_0$ if $|NN_V(x) - L_0| \leq \gamma$.

![Figure 5.1: Neural network results.](image)

Figure 5.1 shows the neural network prediction results for the twisting angle (TA) variable. These results are typical for all variables. It can be seen from these that the neural network model distinguishes better between the labels $S$ and $M$ than between $M$ and $B$; this is due to the higher overlap between the classes $M$ and $B$ than between the
classes S and M.

Figures 5.2 and 5.3 shows the range and standard deviation for the variables FW and HD. \( L \) and \( H \) denote the two end points of an interval (the two components of a data point). It can be observed that for the class corresponding to the worst prediction results (HD) the range is almost the same for the first component of a data point for all the three classes. That shows a significant overlap between all three classes. This can be also observed from the frequency distribution used in deriving the fuzzy sets on Chapter 4 (see figure 4.2).

5.2 Support Vector Machine-Perceptron

Support Vector Machines are motivated by the idea that for patterns projected non-linearly into a high dimensional space (feature space), linear separability is possible [12]. The image of a hyperplane defined in the feature space is the decision boundary in the input space. Using support vector machines, the decision boundaries are determined directly from the training data such that the separating margin of decision boundaries
Figure 5.3: Data characteristics for worst results (HD variable).

is maximized. This learning strategy minimizes the classification error of training and test data.

If the data are linearly separable in the original or increased feature space, then the Support Vector Machines classifier has best generalization. Note that the perceptron establishes a linear decision boundary in the original input space and hence can do only linearly separable classification. Oscillations will occur if classes represented by data are not linearly separable. To prevent this from occurring the so called "pocket" version of the algorithm can be used. In this algorithm weights change only if more than a threshold number of data points are misclassified. However if the patterns are linearly separable, the perceptron learning rule is guaranteed to find a solution in a finite amount of time.

\[ w_{new} = w_{old} + (y - \hat{y}) \cdot x \]  \hspace{1cm} (5.1)

In training the perceptron, after presenting each data point the weights are changed by adding/subtracting the amount by which the pattern was misclassified (in this case
+1 or -1) as shown in equation (5.1), where \( y = \pm 1 \) is the real class label and \( \hat{y} = \pm 1 \) is the predicted class. For the variable HD the pocket version of SVM for 2-class problem is applied (because the problem is not linearly separable): first to find the hyperplane between class \textit{light} and the class \textit{medium} and after between the class \textit{medium} and \textit{heavy}.

Figure 5.4: Decision boundary in SVM classification between M and B classes for (a) training and (b) testing data for HD.

Figure 5.4 shows the big overlap existing between the classes M and B; when the extended data (see figure 5.5) are used the results are improved significantly since the data from class B are “pulled up” and this way the overlap is greatly reduced.

Table 5.1 summarizes the results obtained by using SVM for the variable HD.
Figure 5.5: Decision boundary in SVM classification between M and B classes for (a) training and (b) testing extended data for HD.

5.3 Minimum Distance Classifier

This is template matching approach in which a prototype of the patterns to be recognized is available. An intuitive approach is to represent each class $C_i$ by its mean (average) vector (prototype vector) $p_i$ as shown in equation (5.2).

<table>
<thead>
<tr>
<th>Variable</th>
<th>SVM for original data</th>
<th>SVM for Extended Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD</td>
<td>72.79%</td>
<td>98.77%</td>
</tr>
</tbody>
</table>

Table 5.1: Prediction of Perceptron for variable HD.
\[ p^{(i)} = \frac{1}{|C_i|} \sum_{x \in C_i} x^{(j)} \]

where \(|C_i|\) denotes the number of data points in class \(C_i\). A data point \(x\) is then classified to the class whose prototype it is closest to.

\[ x \in C_{i_0} \text{ where } i_0 = \arg\{ \min d(x, p_i) \} \]

\[ d(i, k) = \left( \sum_{j=1}^{m} |x_{ij} - x_{kj}|^r \right)^{1/r} \]

where \(r \geq 1\). Two special cases of the Minkowski metric obtained for two values of \(r\) are used here, as follows:

1. For \(r = 2\) the Minkowski distance reduces to the usual Euclidean distance

\[ d_E(i, k) = \left( \sum_{j=1}^{d} |x_{ij} - x_{kj}|^2 \right)^{1/2} \]

2. For \(r \rightarrow \infty\) the Minkowski distance results in the Sup distance defined as

\[ d_{Sup}(i, k) = \max_{1 \leq j \leq d} |x_{ij} - x_{kj}| \]

The results depend on the metric used and on the distribution of data. The metric used in this study is the Minkowski metric defined in equation (5.4).

The prediction rates for the Euclidean and Sup distance are shown in presented in the Table 5.2 for the original data and for extended-data. Figures 5.6, 5.7, 5.8

42
\[
\begin{array}{|c|c|c|c|c|}
\hline
\text{Variable} & \text{Original Data} & \text{Extended Data} \\
& \text{d}_E & \text{d}_{Sup} & \text{d}_E & \text{d}_{Sup} \\
\hline
\text{HD} & 82.1\% & 88.88\% & 91.87\% & 96\% \\
\hline
\end{array}
\]

Table 5.2: Prediction of Minimum Distance Classifier for variable HD.

and 5.9 show the test points for all three classes and the separating hyperplanes. In these figures, diamonds represent the class mean (the prototype vector) while triangles represent the misclassified points. It can be observed that for the extended data the number of misclassified points is considerably reduced.

![Minimum Distance Classifier using Euclidean Distance.](image)

Figure 5.6: Minimum Distance Classifier using Euclidean Distance.

In closing this section, the following summarize the Minimum Distance Classifier.

On the positive side, this classifier

- works well if the distance between class means is large compared to the spread of the data in each class;

- yields optimum performance when the distribution of class patterns around its mean is a "spherical hyper-cloud" in the feature (input) space;

on a less positive side,
Figure 5.7: Minimum Distance Classifier for extended data using Euclidean Distance.

Figure 5.8: Minimum Distance Classifier using Sup Distance.

- it makes the assumption that the mean is a good representative of the class. However, as shown in Figure 5.10 this is not always the case;

- it requires the classes to be linearly separable since the decision boundary is linear.
Figure 5.9: Minimum Distance Classifier for extended data using Sup Distance.

Figure 5.10: An example for which Minimum Distance Classifier is not a good approach.
Chapter 6

Comparison of classifiers

6.1 Comparison criteria

In comparing the four different classifiers presented in the previous chapters a set of criteria must be decided on. As already described in Chapter 2 a classifier’s performance can be captured in terms of the prediction error, which is perhaps the most important aspect. At the same time, for two classifiers with comparable prediction error, other aspects must be taken into consideration, including their respective confusion matrices, the difficulty (complexity) of training, their correlation as indicated by the $Q$ statistic and the coefficient of correlation, $\rho$, as well as the ROC graph and learning curve.

6.1.1 Comparing the Fuzzy and Neural Network Classifier

With the error model of (4.6) and (4.7) and $\gamma = 0.5$ for FW, W, HD, TA, F, and $\gamma = 0.4$, for D and VD the best prediction power of the neural network approach compared to that of the fuzzy approach, is shown in figure 6.1 and table 6.1. From these it can be concluded that based on their predictive power, these two classifiers have comparable performance. The neural network described in Chapter 5 was also trained (with training error 0.1) and tested (with $\gamma = 0.5$) on the extended data. The complete results for the fuzzy model and the neural network are in table 6.1.
Figure 6.1: Comparison of the prediction power for fuzzy sets and neural network

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Good prediction</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fuzzy sets</td>
<td>neural networks</td>
<td>γ</td>
</tr>
<tr>
<td>F</td>
<td>84.87</td>
<td>87.65</td>
<td>0.5</td>
</tr>
<tr>
<td>FW</td>
<td><strong>94.50</strong></td>
<td><strong>94.14</strong></td>
<td><strong>0.5</strong></td>
</tr>
<tr>
<td>W</td>
<td>93.30</td>
<td>92.5</td>
<td>0.5</td>
</tr>
<tr>
<td>D</td>
<td>84.80</td>
<td>87.30</td>
<td>0.4</td>
</tr>
<tr>
<td>HD</td>
<td>84.25</td>
<td>84.87</td>
<td>0.5</td>
</tr>
<tr>
<td>VD</td>
<td>86.72</td>
<td>89.81</td>
<td>0.4</td>
</tr>
<tr>
<td>TA</td>
<td>84.25</td>
<td>91.04</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 6.1: Complete results for fuzzy sets and neural networks

The major problem in Neural Networks approach is that it depends on too many parameters (e.g. the NN model, number of hidden neurons, accuracy in training, activation functions, threshold, etc.) and that it is not known for what network configuration it achieves the best generalization. Also for contradictory data (the same data point belonging to two or more classes in the same time) Fuzzy Systems always gives a model for prediction but NN might not converge. For Neural Networks methods the selection of the threshold values requires *an added step after the training of the network*. No such step is required by the fuzzy model.

From a different point of view, once an input data to a label is done, the neural
<table>
<thead>
<tr>
<th>variable name</th>
<th>% prediction fuzzy sets</th>
<th>% prediction neural networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>87</td>
<td>87.34</td>
</tr>
<tr>
<td>FW</td>
<td>95.40</td>
<td>88.28</td>
</tr>
<tr>
<td>W</td>
<td>95.25</td>
<td>98.33</td>
</tr>
<tr>
<td>D</td>
<td>87.4</td>
<td>87.6</td>
</tr>
<tr>
<td>HD</td>
<td>91.5</td>
<td>98.14</td>
</tr>
<tr>
<td>VD</td>
<td>91.95</td>
<td>90.74</td>
</tr>
<tr>
<td>TA</td>
<td>91.66</td>
<td>96.29</td>
</tr>
</tbody>
</table>

Table 6.2: Prediction results for fuzzy sets and neural networks on the adjusted data

network cannot distinguish between different instances of the same label, implementing in effect, each label as a crisp category. Alternatively, one could come up with a way of differentiating between the elements of the same label. However, such a step would be yet another additional step and, more importantly, it would be ad-hoc without a clear meaning.

6.2 Correlation between the Fuzzy and Neural Network Classifiers

To further compare the fuzzy and neural network classifier the Yule, $Q$ statistic, and coefficient of correlation $\rho$ defined in (2.1) and (2.2) respectively, are used.

For the variable HD the $Q$ statistic is evaluated based on the table 6.3. Its corresponding value, $Q = 0.88$, shows that the two classifiers to compare are strongly correlated and tend to recognize the same data points. The correlation coefficient is computed according with the equation (2.2) and its value is $\rho = 0.5$, indicating a high similarity in prediction of the two classifiers.

Table 6.4 shows the complete values of $Q$ and $\rho$ for the fuzzy and neural network classifier. As it can be seen from this table, these values indicate a high correlation between the two classifiers which means that basically, they agree both in correct and
<table>
<thead>
<tr>
<th>Fuzzy System Classifier</th>
<th>Neural Network Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>253/324</td>
</tr>
<tr>
<td>Wrong</td>
<td>20/324</td>
</tr>
</tbody>
</table>

Table 6.3: Confusion matrix for variable HD.

<table>
<thead>
<tr>
<th>Variable name</th>
<th>$Q$</th>
<th>$\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>0.99</td>
<td>0.83</td>
</tr>
<tr>
<td>FW</td>
<td>0.99</td>
<td>0.74</td>
</tr>
<tr>
<td>W</td>
<td>0.98</td>
<td>0.74</td>
</tr>
<tr>
<td>D</td>
<td>0.94</td>
<td>0.61</td>
</tr>
<tr>
<td>HD</td>
<td>0.88</td>
<td>0.5</td>
</tr>
<tr>
<td>VD</td>
<td>0.87</td>
<td>0.43</td>
</tr>
<tr>
<td>TA</td>
<td>0.73</td>
<td>0.28</td>
</tr>
</tbody>
</table>

Table 6.4: $Q$ and $\rho$ for the fuzzy sets and neural networks

incorrect classification.

From the table 6.1 it can be seen that on the whole the neural network approach does slightly better for most of the variables, to better for the twisting angle (TA). However, the performance of the neural network on the test data depends on the values of the threshold $\gamma$. Two different values for $\gamma$ were required to reach the best performance for the neural network.

### 6.3 Global Comparison of the classifiers

The prediction results for the four classifiers indicate that they have similar prediction powers. To compare their learning curves, the four classifiers were trained on data sets of increasing sizes, of 5%, 10%, 30%, 50%, 80% and 90% of the available data set and tested, in each case, on the remaining data. Figure 6.2 shows the learning curves for the four classifiers for the variable twisting angle (TA). The learning curves for the other variables are somewhat similar and they convey the following:
<table>
<thead>
<tr>
<th>Percent</th>
<th>95%</th>
<th>90%</th>
<th>70%</th>
<th>50%</th>
<th>10%</th>
<th>5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Size</td>
<td>616</td>
<td>583</td>
<td>454</td>
<td>324</td>
<td>65</td>
<td>32</td>
</tr>
</tbody>
</table>

Table 6.5: The size of test set used to obtain the learning curves for variable HD (the original data set consists of 648 data points).

- The neural network classifier has overall best performance followed closely by the fuzzy classifier;

- When the training set size is small the fuzzy classifier outperforms slightly the neural network. Using only 5% of data for training and testing on the remaining 95% gives a generalization power of 82.2% for the fuzzy classifier and 81.65% for the neural network classifier.

- The learning curves for the fuzzy, neural network and support vector machine classifiers increase sharply when the training set exceeds 10% of the data set, and for all of the classifiers when it exceeds 90%.

- For the training set size between 30% to 90% of the data, the learning curves for the fuzzy and support vector machine classifiers tend to be somewhat flat. This behavior is not surprising for the support vector machine as its performance depends on the support vectors, which make up a relatively small part of the training data. Only when many more training data are added, it is likely that the number of support vectors will increase. For the fuzzy system this behavior indicates that the classifier can perform well on a relatively small amount of training data, and more importantly, this behavior is achieved after **one pass** through the training data (this is not the case for the neural network, where the training usually required in excess of 3,000 epochs).
Figure 6.2: The learning curve of all four classifiers for variable TA.

6.4 The ROC statistics

For comparison purpose the ROC graph was obtained based on the model described in Chapter 2. The graph contains all the information in the confusion matrix and can better convey the ability of each classifier to recognize the patterns.

All the classifiers are admissible, in the sense that their corresponding ROC points are in the desired area (shaded region of the 2.2). In fact, fairly close to (0,1) corresponding to a perfect classifier. Figures 6.3, 6.4 and 6.5 show the ROC points plotted for the four classifiers for classes $S$, $M$ and $B$, respectively, based on the model described on Section 2.5 (see also figure 2.2). The worst results are obtained for class $M$. This does not come as a surprise as class $M$ overlaps both with class $S$ and class $B$ (figure 6.4). For this class the fuzzy classifier has the best ROC point supporting the argument
Figure 6.3: The ROC points of the four classifiers for class S (variable HD).

<table>
<thead>
<tr>
<th></th>
<th>Fuzzy Systems</th>
<th>Neural Networks</th>
<th>Perceptron</th>
<th>Minimum Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>False Positive (FP)</td>
<td>0.12</td>
<td>0.1</td>
<td>0.16</td>
<td>0.097</td>
</tr>
<tr>
<td>True Positive (TP)</td>
<td>0.78</td>
<td>0.62</td>
<td>0.75</td>
<td>0.55</td>
</tr>
</tbody>
</table>

Table 6.6: The computed FP and TP values used in ROC plots (variable TA).

that for classes, such as class $M$, that show a large variability and overlap with other classes, the fuzzy model is the most suitable.
Figure 6.4: The ROC points of the four classifiers for class M (variable HD).

Figure 6.5: The ROC points of the four classifiers for class B (variable HD).
Chapter 7

Conclusions

This study addressed the problem of modeling human perception of the linguistic descriptions of numerical values. Four approaches to classifier design, a fuzzy set-based approach, neural network approach, support vector machine, and minimum distance classifier were implemented. The results were evaluated on a real data set obtained from a survey carried out at the Hong Kong Polytechnic University as part of collaborative work with the University of Cincinnati, Department of Mechanical, Nuclear and Industrial Engineering.

The four approaches considered give similar results in so far as the error rates are concerned. Taking into account other criteria, some model may be better than the others. The choice of one of the models may, finally, depend on several factors, including the goal of the modeling, further use of the results, etc. For example, if the user of the model is also interested in being able to differentiate between instances of the same class, then the fuzzy model should be used. If this aspect is not issue then, perhaps the neural network might be selected. If complexity of learning is important, then the fuzzy model, support vector machine and the minimum distance classifier each reach a solution in one pass of the data.

From the application point of view, the results presented here support the idea that a
machine learning approach (complementary to statistical methods) can be used to capture the relation between hand data (such as numerical descriptions of the lifting task characteristics) and their soft descriptions in words. This approach can be extended to model the relationship between other aspects of physical tasks and their perception by those performing such task, and help understand this complex relationship. Ultimately, several such models connected can produce a system that can be used for task design based on physical and psychological aspects of the subjects as well as on the objective task measurements.

7.1 Open problems

As conveyed in the last paragraph the approach proposed here could be extended. This remains to be seen, and depends on whether the subjects interviewed for other tasks and their characteristics have a common understanding of these.

From the modeling point of view it is clear that the similarity of performance of the four classifiers can be improved but very little by combining them. In general, when this situation arises it would be useful to find a way to characterize a data set in terms of the best learning that can be obtained from it. This could perhaps be done by analyzing its statistical characteristics. Some of these characteristics, when provided in this study for the best and worst performance, conveyed some relation between the data set and the quality of learning that can be achieved on it.

Machine learning, in general, classifier design, in particular, remain topics of strong interest among those working in the field of intelligent systems. Theoretical developments have recently pointed to formal treatments. However, such treatments are always kept in check by the application problem and domain considered.
Bibliography


