Optimising Two-Stage Recognition Systems

Thomas Landgrebe, Pavel Paclík, David M.J. Tax, and Robert P.W. Duin

Elect. Eng., Maths and Comp. Sc.,
Delft University of Technology, The Netherlands
{t.c.w.landgrebe, p.paclik, d.m.j.tax, r.p.w.duin}@ewi.tudelft.nl

Abstract. A typical recognition system consists of a sequential combination of two experts, called a detector and classifier respectively. The two stages are usually designed independently, but we show that this may be suboptimal due to interaction between the stages. In this paper we consider the two stages holistically, as components of a multiple classifier system. This allows for an optimal design that accounts for such interaction. An ROC-based analysis is developed that facilitates the study of the inter-stage interaction, and an analytic example is then used to compare independently designing each stage to a holistically optimised system, based on cost. The benefit of the proposed analysis is demonstrated practically via a number of experiments. The extension to any number of classes is discussed, highlighting the computational challenges, as well as its application in an imprecise environment.

1 Introduction

In this paper we view the sequential combination of two classifiers as a Multiple Classifier System (MCS). We illustrate that the independent design of individual classifiers in such sequential systems results in sub-optimal performance, since it ignores the interaction between stages. In this paper we demonstrate that optimality can be obtained by viewing such an MCS in a holistic manner. This research is targeted specifically at two-stage recognition systems, in which the first stage classifier attempts to detect *target* object distributed among a typically poorly sampled, or widely distributed *outlier* class. The second classifier then operates on objects selected by the first, and discriminates between sub-*target* classes. An example is image-based road-sign recognition [9], in which the first stage involves detecting road-signs that are distributed among an arbitrary background, and the second stage consists of a classifier to distinguish between different sign classes. Another application is fault diagnosis, such as [7], in which the first stage classifier is designed to detect a fault from normal operation, and the second stage to characterise the type of fault.

Considering the detector, since the *outlier* class is poorly defined, a two-class discrimination scheme is inappropriate, and other methods that are trained/designed only on the *target* class are typically used, such as correlation. Recently One Class Classification (OCC) was introduced [12], consisting of a formal framework to train models in situations in which data from only a single

N.C. Oza et al. (Eds.): MCS 2005, LNCS 3541, pp. 206–215, 2005.

class is available. This allows a statistical pattern recognition methodology to be taken in designing the detector[1]. Thus we consider these recognition systems as a mixture of one-class and multi-class classifiers.

Evaluating the recognition system involves analysing the classification accuracy, and the rate of *outlier* false acceptances. Importantly, a poor detector that does not detect a large fraction of *target* objects results in poor classification performance. In the opposite case, a very sensitive detector may pass an unacceptably large fraction of *outlier* objects to the classifier, which may for example result in high manual processing costs or computational overload.

The paper is structured as follows: Section 2 presents an analytic example to demonstrate how the two classifiers interact. A cost-based approach using ROC analysis demonstrates how system optimisation can be performed in evaluating the entire system. In Section 3 the multiple-class extension is discussed briefly, highlighting some problems that exist in extending the analysis to a large number of *target* classes. In Section 4, some experiments on real data are performed, consisting of a simple problem with 2 *target* classes, and a 4-class problem involving hand-written digit recognition. In Section 5 we briefly consider the case in which priors or costs cannot be defined precisely, discussing how different system configurations can be chosen in these situations. Conclusions are given in Section 6.

2 The Dependence Between Classifiers

2.1 Two-Stage Recognition Systems

Consider a recognition task in which there are a number (n) *target* classes $\omega_{t1}, \omega_{t2}, \ldots, \omega_{tn}$, and an *outlier* class ω_o. A recognition system, as illustrated in Figure 1, has to classify these objects. A detector D_{DET} classifies incoming objects as either *target* (ω_t), or *outlier* via a detection threshold θ_d:

$$D_{DET}(\mathbf{x}) : \begin{cases} \textit{target} & \text{if } f_{DET}(\mathbf{x}) > \theta_d \\ \textit{outlier} & \text{otherwise} \end{cases} \tag{1}$$

The detector selects objects from $\mathbf{x}$ such that the input to D_{CLF} is $\tilde{\mathbf{x}}$.

$$\tilde{\mathbf{x}} = \{\mathbf{x} | f_{DET}(\mathbf{x}) > \theta_d\} \tag{2}$$

The classifier D_{CLF} then classifies incoming objects (according to $\tilde{\mathbf{x}}$) to any of the n *target* classes via the classification thresholds[2] $\theta_c^{t1}, \theta_c^{t2}, \ldots, \theta_c^{tn}$. The classifier

[1] Note that the MCS view on such a multi-stage system also holds for two-stage recognition systems that are constructed for computational reasons. In this case the first stage is typically designed for fast rejection of very abundant *outlier* objects, with a more complex second stage to discriminate between *target* classes.

[2] In an n-class situation, the classification thresholds can be considered to be the weighting applied to the output posterior density estimates together with priors.

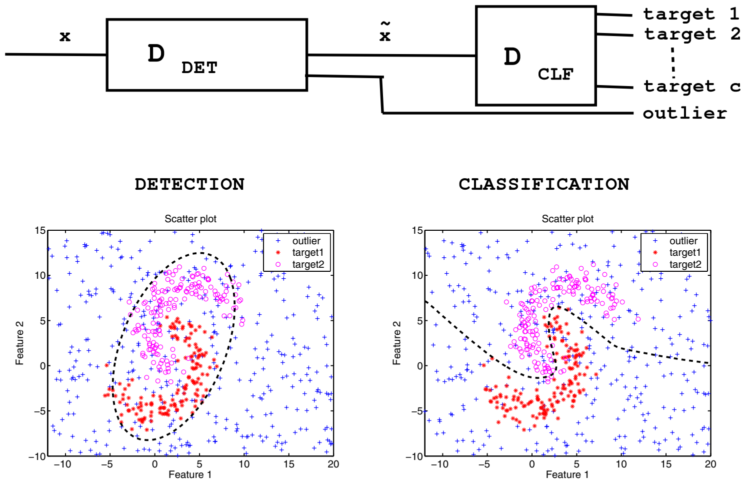

Fig. 1. Illustrating a typical recognition system on a synthetic example. The scatter plots show a 2-dimensional synthetic example with two *target* classes, illustrating the detector in the left plot, and the classifier in the right

outputs are weighted by classification thresholds and priors $p(\omega_{t1}), p(\omega_{t2}), \ldots, p(\omega_{tn})$. The classifier outputs $f_{CLF}(\tilde{\mathbf{x}})$ can then be written as:

$$[\theta_c^{t1} p(\omega_{t1}) f_{CLF}(\omega_{t1}|\tilde{\mathbf{x}}), \theta_c^{t2} p(\omega_{t2}) f_{CLF}(\omega_{t2}|\tilde{\mathbf{x}}), \ldots, \theta_c^{tn} p(\omega_{tn}) f_{CLF}(\omega_{tn}|\tilde{\mathbf{x}})] \quad (3)$$

Here $\sum_{i=1}^{n} \theta_c^{ti} = 1$. The final decision rule is then:

$$D_{CLF}(\tilde{\mathbf{x}}) = \mathrm{argmax}_{i=1}^{n}\, \theta_c^{ti} p(\omega_{ti}) f_{CLF}(\omega_{ti}|\tilde{\mathbf{x}}) \quad (4)$$

The primary distinction between this two-stage system and a multi-class single-stage recognition system is that the input to the classification stage in the two-stage case is a subset of the system input, whereas in the single-stage case all data is processed. We are considering the dependence (in terms of overall system performance) of the 2 stages, and how the system should be optimised.

2.2 One-Dimensional Example

In this section a simple 1-dimensional analytical example is studied in order to illustrate how the detection and classification stages are related. Two Gaussian-distributed *target* classes ω_{t1} and ω_{t2} are to be detected from a uniformly-distributed *outlier* class ω_o, and subsequently discriminated. The *target* classes have means of -1.50 and 1.50 respectively, and variances of 1.50. The ω_o class has a density of 0.05 across the domain x. The class conditional densities for ω_{t1}, ω_{t2} and ω_o are denoted $p(x|\omega_{t1})$, $p(x|\omega_{t2})$, and $p(x|\omega_o)$ respectively, with priors $p(\omega_{t1})$, $p(\omega_{t2})$, and $p(\omega_o)$, which are assumed equal here. For the total probability distribution of x therefore holds:

$$p(x) = p(\omega_{t1})p(x|\omega_{t1}) + p(\omega_{t2})p(x|\omega_{t2}) + p(\omega_o)p(x|\omega_o) \quad (5)$$

For this 1-dimensional data, the classifier is defined consisting of only a single threshold, denoted θ_c. The position of θ_c determines the classification performance, and can be used to set an operating point to achieve a specified false-negative rate FN_r (with respect to ω_{t1}) or false-positive rate (FP_r). These two errors are known as the Error of Type I and II respectively (ϵ_I and ϵ_{II}). As θ_c varies, so do the respective ϵ_I and ϵ_{II}, resulting in the ROC (receiver-operator curve [8]) between ω_{t1} and ω_{t2}. In a typical discrimination problem (ignoring the detector) across domain x, we can define ϵ_I and ϵ_{II} in terms of θ_c as:

$$\epsilon_I = 1 - \int_{-\infty}^{\infty} p(x|\omega_{t1}) I_1(x|\theta_c)dx, \; \epsilon_{II} = 1 - \int_{-\infty}^{\infty} p(x|\omega_{t2}) I_2(x|\theta_c)dx \qquad (6)$$

The indicator functions $I_1(x|\theta)$ and $I_2(x|\theta)$ specify the relevant domain:

$$\begin{array}{r} I_1(\mathbf{x}|\theta_c) = 1 \text{ if } \; p(\omega_{t1})p(\mathbf{x}|\omega_{t1}) - p(\omega_{t2})p(\mathbf{x}|\omega_{t2}) < \theta_c, \; 0 \text{ otherwise} \\ I_2(\mathbf{x}|\theta_c) = 1 \text{ if } p(\omega_{t1})p(\mathbf{x}|\omega_{t1}) - p(\omega_{t2})p(\mathbf{x}|\omega_{t2}) \geq \theta_c, \; 0 \text{ otherwise} \end{array} \qquad (7)$$

A two-stage recognition system consists of two sets of thresholds, namely a classification threshold θ_c (of which there are a number of thresholds according to the number of classes), and a detection threshold θ_d. Evaluating the recognition system involves estimating both classification performance (ϵ_I and ϵ_{II}), and the fraction of *outlier* objects incorrectly classified as *target*, denoted FP_r^o. Thus one axis of the evaluation is concerned with how well the system performs at detecting and discriminating *target* classes, and the other is concerned with the amount of false alarms that the system must deal with. Therefore the system must be evaluated with respect to ϵ_I, ϵ_{II}, and FP_r^o. In this simple example, we can write these as:

$$\begin{array}{rcl} \epsilon_I & = & 1 - \int_{-\infty}^{\infty} p(x|\omega_{t1}) I_1(x|\theta_c) I_R(x|\theta_d, \omega_{t1}) dx \\ \epsilon_{II} & = & 1 - \int_{-\infty}^{\infty} p(x|\omega_{t2}) I_2(x|\theta_c) I_R(x|\theta_d, \omega_{t2}) dx \\ FP_r^o & = & \int_{-\infty}^{\infty} p(x|\omega_o) I_1(x|\theta_c) I_R(x|\theta_d, \omega_{t1}) + p(x|\omega_o) I_2(x|\theta_c) I_R(x|\theta_d, \omega_{t2}) dx \end{array} \qquad (8)$$

$$I_R(\mathbf{x}|\theta_d, \omega) = 1 \; if \; p(\mathbf{x}|\omega) > \theta_d, \; 0 \; otherwise \qquad (9)$$

Equation 8 yields the full operating characteristics of the system, shown in Figures 2 and 3 for the example. Referring first to Figure 2, this shows how the system operating characteristics vary for a number of fixed detection thresholds. The top row illustrates the position of the detection threshold, and the bottom row shows ϵ_I, ϵ_{II}, and FP_r^o for all classification thresholds (similar to standard ROC analysis, except an additional dimension is introduced to account for the detection threshold). In these plots, it is desirable for ϵ_I, ϵ_{II}, and FP_r^o to be minimal, indicating good classification and detection.

In Figure 2, as θ_d is increased, the plots show how FP_r^o progressively decreases. In the left-column, a very sensitive detector is used, with θ_d placed in the tails of the *target* distribution. It is clear that the classification performance is almost maximal for this threshold, but FP_r^o is very high i.e. the system will accept a very high percentage of *outlier* objects. The centre column plots show the case for which a higher detection threshold has been used ($\theta_d = 0.05$), resulting

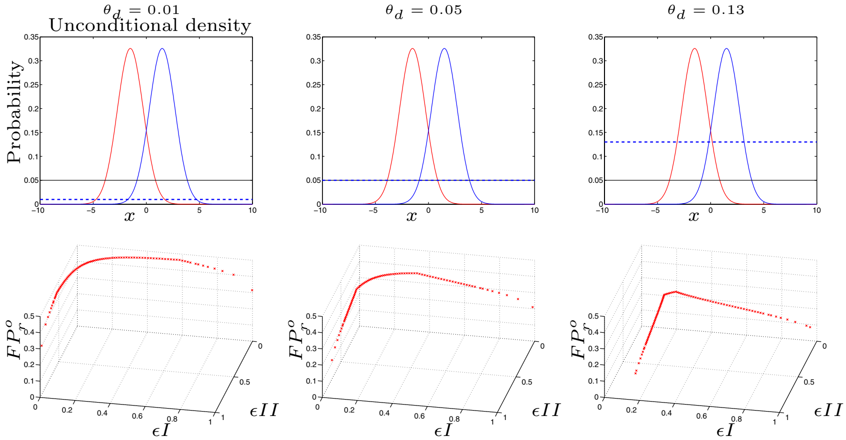

Fig. 2. Operating characteristics for a fixed θ_d, and varying θ_c. The left column is where $\theta_d = 0.01$, followed by $\theta_d = 0.05$ in the middle column, and $\theta_d = 0.13$ in the right column. The top row plots illustrate the distribution, with two Gaussian *target* classes, and a uniformly distributed *outlier* class. The position of the detection threshold is shown via the dotted line. The full operating characteristics for all possible θ_c are shown in the bottom row

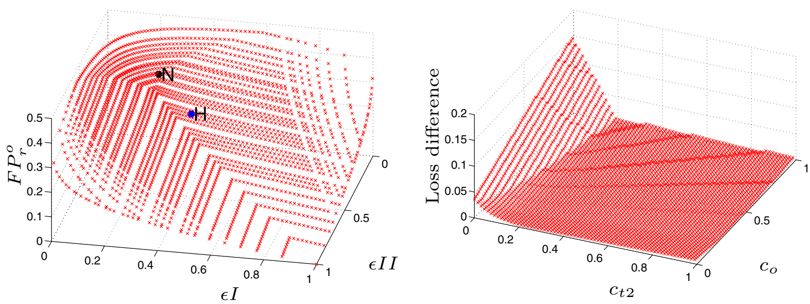

Fig. 3. Results of analytic experiment. The left plot shows the full operating characteristics, with ϵ_I plotted against ϵ_{II}, and FP_r^o. The right plot shows the loss difference between an independent and holistic design approach for all combinations of c_{t2}, and c_o over a $\{0, 1\}$ range, where c_{t1} is fixed to 0.55

in a substantially lower FP_r^o, for a small sacrifice in classification performance. The third column shows a situation in which θ_d is again increased, resulting in a further decrease in classification performance. In this case the detector only accepts very probable *target* objects, reducing the volume of the *target* class decision space, at the expense of all *target* objects appearing outside the decision boundary. The left plot of Figure 3 shows the operating characteristics for all combinations of θ_c and θ_d. Next we show how using the full operating characteristic can be advantageous in system design.

2.3 Cost-Based Analysis

From the system perspective, the cost of misclassifying a ω_t object (as *outlier*) is c_t, and the cost of misclassifying a ω_o object (as *target*) is c_o. The individual *target* class misclassification costs can be written as $c_{t1}, c_{t2}, \ldots c_{t_n}$, which must sum to c_t together with the priors (note that we do not consider the entire loss matrix as defined in [2], but only consider the loss incurred due to misclassification, irrespective of the class to which it is assigned). The expected overall system loss L can be written as:

$$L = c_t p(\omega_t) FN_r + c_o p(\omega_o) FP_r^o, = \sum_{i=1}^{n} c_{t_i} p(\omega_{t_i}) FN_r^{t_i} + c_o p(\omega_o) FP_r^o, \quad \sum_{i=1}^{n} c_{t_i} = c_t \tag{10}$$

The priors are denoted $p(\omega_t)$ and $p(\omega_o)$, and the false negative rate of ω_t is denoted FN_r. The *target* class misclassification costs are denoted c_{t_i} for *target* class ω_{t_i}. Cost-based classifier design involves minimising of L for the given costs, resulting in the optimal threshold values. The ROC is a tool that can be used to facilitate this minimisation, since it consists of performances for all possible threshold values (all FN_r and FP_r results). In a 2-class problem, the costs (and priors) specify the gradient of the cost line (also known as an *iso-performance* line as defined in [10]), and the intersection of the normal of this line with the ROC (plotting FN_r against FP_r) results in the optimal operating point[3]. We now demonstrate a cost-analysis for the example in order to emphasise the importance of designing the entire system holistically. Two different design approaches are compared, the first of which we refer to as the *independent* approach, and the second as the *holistic* approach. In the first case, we optimise the recognition and classification stages independently, and compare the expected system loss to the second case, in which the entire system is optimised holistically. We assume that the cost specification for the recognition system is such that misclassifying a ω_t object has a cost of 5, and the cost of classifying a ω_o object as *target* is 10. Among the two *target* classes ω_{t1} and ω_{t2}, these have misclassification costs of 2 and 3 respectively (summing to 5), i.e. ω_{t2} is favoured. From Equation 10, we can write the system loss (assuming equal priors) for the chosen θ_c and θ_d as $L(\theta_c, \theta_d) = 2\epsilon_I(\theta_c, \theta_d) + 3\epsilon_{II}(\theta_c, \theta_d) + 10FP_r^o(\theta_c, \theta_d)$. In the *independent* approach, the detector is optimised using ω_t and ω_o data only (with operating characteristics generated for these classes only). The classifier is then optimised on ω_{t1} and ω_{t2}. The corresponding thresholds are indicated by the point marked **N** in the left plot of Figure 3. In the holistic approach, ω_{t1}, ω_{t2}, and ω_o are analysed simultaneously in the optimisation, resulting in the point marked **H**. The two points **N** and **H** are significantly apart on the operating characteristic. In the *independent* approach, the overall expected loss is thus **4.18**, and in the

[3] We deal with multi-dimensional ROC plots in this paper. Cost-based optimisation involves intersecting a plane (the gradient based on the cost associated with misclassifying each class) with the multi-dimensional ROC surface, resulting in optimised thresholds.

holistic approach, the loss is **4.02**. Thus independent approach is sub-optimal here. Depending on the problem and the costs, the *independent* approach may vary in the degree of sub-optimality. To assess how the holistic approach will improve performance in general, refer to the right plot of Figure 3. This plot shows the difference between the *independent* and *holistic* loss performances (where a positive score indicates superiority of the *holistic* approach) for all combinations of costs over a range. The cost c_{t1} is fixed to 0.55, and c_{t2} and c_o are varied for all combinations over the $\{0, 1\}$ range. It can be seen that for this artificial example, only imbalanced costs result in significant improvements. In the experiments, it will be shown models that do not fit the data well in real problems can benefit even more from this approach, including balanced cases.

3 Multiple Class Extension

The analytic example involved a recognition system with 2 *target* classes, resulting in a 3-dimensional ROC surface. As the number of *target* classes increase, the dimensionality of the ROC increases. The analysis extends to any number of classes [11]. However, as the number of dimensions increase, the computational burden becomes infeasible [5]. In this paper, experiments involved up to 3 *target* classes. In this case, the processing costs were already very high. and only a very sparsely sampled ROC could be generated. Extending this analysis to N classes would be infeasible. This is the topic of future work, exploring approaches that can be used to either approximate the full ROC, or to use search techniques in optimising the thresholds. Attention is drawn to [6], in which an initial set of thresholds is used, and a hill-climbing greedy-search is used.

4 Experiments

In this section a number of experiments are conducted on real data in order to demonstrate the holistic system design approach practically, and how model (or system configuration) selection can be performed. Two datasets are used, described as follows:

- *Banana*: A simple 2 dimensional problem with 2 *target* classes distributed non-linearly (the banana distribution [4]), in which there are 600 examples each of ω_{t1} and ω_{t2}, and 2400 *outlier* examples. The distribution is shown in Figure 1.
- *Mfeat*: This is a dataset consisting of examples of ten handwritten digits, originating from Dutch utility maps[4]. In this dataset, Fourier components have been extracted from the original images, resulting in a 76-dimensional representation of each digit. 200 examples of each digit are available. In these experiments, digits *3*, *4* and *8* are to be distinguished (i.e. 3 *target* classes ω_{t1}, ω_{t2}, and ω_{t3}), distributed among all other digit classes, which are considered to be *outlier*.

[4] Available at ftp://ftp.ics.uci.edu/pub/machine-learning-databases/mfeat/

We follow the same analysis approach as in Section 2. Classification and detection thresholds are generated across the full range. In the *Banana* case, 200 evenly sampled classification thresholds are used, and similarly 100 detection thresholds are used. For computational reasons, the *Mfeat* experiments only uses 10 detection thresholds, and 12 samples per classification threshold. Each experiment involves a 10-fold randomised hold-out procedure, with 80% of the data used in training, and the remainder for testing. The evaluation consists of evaluating the loss incurred for a number of chosen misclassification costs, using the ROC to find an optimal set of thresholds. In this evaluation it is assumed that the costs (and priors) are known beforehand, and as in Section 2, we only consider misclassification costs, applying Equation 10.

In the *Banana* experiments, 3 different system configurations are implemented, comparing the *independent* and *holistic* approaches for each case. The same detector is used for all 3 configurations, consisting of a Gaussian one class classifier (OCC) [12]. Three different classifier models are used, consisting of a Bayes linear, quadratic, and mixture of Gaussians classifier (with two mixtures per class), denoted LDC, QDC, and MOG respectively. In Table 1 the *Banana* experimental results are shown for 4 different system costs. These are shown in the four right-most columns, with the costs denoted $[c_{t1}, c_{t2}, c_o]$. For all 3 system configurations, the *holistic* design approach results in a lower overall expected loss than the *independent* approach. In some cases the difference in performance is not significant (see the MOG results for the case in which $c_{t1} = 3.0$, $c_{t2} = 1.0$, and $c_o = 4.0$). These experiments show that the benefit of an overall design approach can in many cases result in significant improvements in performance.

A similar set of experiments are conducted for the *Mfeat* problem, with costs denoted $[c_{t1}, c_{t2}, c_{t3}, c_o]$. Results are shown for four different cost specifications in the right-most columns of Table 1. Three different system configurations are considered, and in each case the *independent* and *holistic* design approaches are compared. The first configuration consists of a principal component analysis (PCA) mapping with 3 components and a Gaussian OCC as the detector, followed by a Fisher mapping and LDC as the classifier. The second configuration uses a 3-component PCA mapping Gaussian OCC for the detector, and a 3-component PCA LDC for the classifier. Finally the third system consists of a 5-component PCA with Gaussian OCC detector, and a 2-component PCA MOG classifier with 2 mixtures for the classifier. As before, the *holistic* approach consistently results in either a similar or lower overall loss compared to the *independent* approach. Once again, the improvement is dependent on the cost specification. For costs $[1, 8, 1, 10]$ (favouring ω_{t2}) and $[1, 1, 1, 12]$ (favouring ω_o), there is no significant improvement in using the *holistic* approach for all 3 systems. However, when the costs are in favour of ω_{t1}, the holistic approach leads to a significantly lower system loss. This suggests that the ω_{t1} threshold has more effect over the detection performance. In this case θ_d should be adjusted accordingly for optimal performance. The same observation is made for balanced costs $[1, 1, 1, 3]$. An interesting observation made in these experiments is models that do not fit the data well (e.g. the LDC in the *Banana* experiments, compared

Table 1. Results of cost-based analysis for the *Banana* and *Mfeat* datasets, comparing an *independent* (I) and *holistic* (H) design approach for a number of different system configurations (low scores are favourable). Standard deviations are shown

Detector	**Classifier**	Cost 1	Cost 2	Cost 3	Cost 4
Banana		$[5, 5, 10]$	$[3, 1, 4]$	$[1, 3, 4]$	$[1, 1, 4]$
Gauss	LDC I	0.081 ± 0.009	0.370 ± 0.049	0.233 ± 0.056	0.244 ± 0.046
Gauss	LDC H	0.067 ± 0.008	0.326 ± 0.039	0.171 ± 0.015	0.189 ± 0.027
Gauss	QDC I	0.089 ± 0.017	0.418 ± 0.051	0.260 ± 0.060	0.265 ± 0.053
Gauss	QDC H	0.072 ± 0.010	0.354 ± 0.036	0.179 ± 0.025	0.182 ± 0.030
Gauss	MOG I	0.059 ± 0.008	0.252 ± 0.033	0.206 ± 0.032	0.205 ± 0.030
Gauss	MOG H	0.049 ± 0.007	0.230 ± 0.035	0.170 ± 0.019	0.169 ± 0.021
Mfeat		$[1, 1, 1, 3]$	$[8, 1, 1, 10]$	$[1, 8, 1, 10]$	$[1, 1, 1, 12]$
PCA3 Gauss	Fisher LDC I	0.648 ± 0.050	0.212 ± 0.018	0.225 ± 0.017	1.385 ± 0.316
PCA3 Gauss	Fisher LDC H	0.547 ± 0.110	0.146 ± 0.014	0.223 ± 0.017	1.317 ± 0.435
PCA3 Gauss	PCA3 LDC I	0.654 ± 0.053	0.214 ± 0.018	0.225 ± 0.017	1.389 ± 0.316
PCA3 Gauss	PCA3 LDC H	0.551 ± 0.110	0.146 ± 0.015	0.224 ± 0.017	1.305 ± 0.432
PCA5 Gauss	PCA2 MOG2 I	0.442 ± 0.029	0.146 ± 0.011	0.154 ± 0.011	0.929 ± 0.202
PCA5 Gauss	PCA2 MOG2 H	0.380 ± 0.079	0.112 ± 0.024	0.148 ± 0.018	0.847 ± 0.124

to MOG), tend to benefit more from the holistic optimisation, suggesting that the interaction is more prominent for all costs.

5 Imprecise Environments

The approach taken thus far showed that, given both misclassification costs and priors, the optimal set of thresholds can be found. In many practical situations the costs or priors cannot be obtained or specified precisely [10]. In these situations we may still wish to choose the best system configuration, and have some idea of a good set of system thresholds that may, for example, be suitable for a range of operating conditions or costs (see [1] and [3]). We do not go into more detail here due to space constraints, but emphasise the fact that real problems are often within an imprecise setting, requiring an alternative evaluation to the cost-based approach. One strategy for this situation is to compute the AUC (Area Under the ROC curve) for a range operating points. An integrated error results that is useful for model selection. The next step is to choose thresholds, which may for example be specified by considering operating regions that are relatively insensitive to changes in cost or priors.

6 Conclusion

A two-stage recognition system was considered as an MCS, consisting of a detection and classification stage, with the objective of optimising the overall system. An analysis of a simple analytic problem was performed, in which the full operating characteristics were computed for all combinations of detection and classification thresholds. The holistic design approach was compared to the case in which the two stages are designed independently, showing that the holistic approach may result in a lower expected loss. The N-class extension was discussed, highlighting the computational difficulties in scaling the analysis to any number of classes. Some experiments with real data were then undertaken for a number of system configurations to demonstrate practical application of the analysis, consistently demonstrating the advantage of the holistic design approach. It was observed that the performance improvements vary according to the cost specification, and the respective degree of interference a class may impose on the detection stage. Models that fit the data well only seem to benefit for imbalanced costs/priors, whereas ill-fitting models can result in improvements for any costs. Finally, a short discussion on application of the methodology to imprecise environments was given. Future work includes exploring efficient multi-class ROC analysis, and application to an imprecise environment.

Acknowledgements. This research is/was supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs.

References

[1] N.M. Adams and D.J. Hand. Comparing classifiers when misallocation costs are uncertain. *Pattern Recognition*, 32(7):1139–1147, 1999.

[2] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press Inc., New York, first edition, 1995.

[3] A.P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[4] R.P.W. Duin, P. Juszcak, P. Paclik, E. Pekalska, D. de Ridder, and D.M.J. Tax. Prtools, a matlab toolbox for pattern recognition, January 2004. version 4.0.

[5] C. Ferri, J. Hernndez-Orallo, and M.A. Salido. Volume under the roc surface for multi-class problems. *Proc. of 14th European Conference on Machine Learning*, pages 108–120, 2003.

[6] N. Lachiche and P. Flach. Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves. *Proc. 20th International Conference on Machine Learning (ICML-2003), Washington DC*, pages 416–423, 2003.

[7] A. Lipnickas, J.S. da Costa, and C.D. Bocaniala. FDI based on two stage classifiers for fault diagnosis of valve actuators. *11th Int. Power Electronics and Motion Control Conference*, pages 3,147–153, September 2004.

[8] C. Metz. Basic principles of roc analysis. *Seminars in Nuclear Medicine*, 3(4), 1978.

[9] P. Paclík. Building road sign classifiers. *PhD thesis, CTU Prague, Czech Republic*, December 2004.

[10] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42:203–231, 2001.

[11] A. Srinivasan. Note on the location of optimal classifiers in N-dimensional ROC space. *Oxford University Computing Laboratory Technical report PRG-TR-2-99*, October 1999.

[12] D.M.J. Tax. One-class classification. *PhD thesis, TU Delft, The Netherlands*, June 2001.