On Neyman-Pearson optimisation for multiclass classifiers

Thomas Landgrebe, and Robert P.W. Duin

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

Abstract

Typically two procedures are used in optimising classifiers. The first is cost-sensitive optimisation, in which given priors and costs, the optimal classifier weights/thresholds are specified corresponding to minimum loss, followed by model comparison. This procedure extends naturally to the multiclass case. The second is Neyman-Pearson optimisation, in which costs may not be certain, and the problem involves specification of one of the class errors, which subsequently fixes the corresponding error (in the two class case), followed by comparisons between different models. This optimisation is well understood in the two-class case, but less so in the multiclass case. In this paper we study the extension of Neyman-Pearson optimisation to the multiclass case, involving specifying various classification errors, and minimising the others. It is shown empirically that the optimisation can indeed be useful for the multiclass case, but obtaining a viable solution is only guaranteed if a single error is specified. Specifying more than one error may result in a solution depending on the data and classifier, which is determined via a multiclass ROC analysis framework.

1. Introduction

In statistical pattern recognition, a typical design procedure involves gathering representative data for each class, and estimating model parameters to derive a discrimination function (e.g. density estimation, support vector classification), as well as a suitable representation (e.g. feature extraction, feature selection [1]). Once a suitable model is found, the next step is to optimise the various classification weights/thresholds. This optimisation is defined by the nature of the problem at hand. In some situations, the optimisation can be posed as a loss-minimisation problem. In this case classification costs are known, and the respective loss can be computed for different classification weights by summing confusion matrix errors, weighted by the respective costs and priors. This is commonly known as cost-sensitive optimisation [1], [2].

In other situations, referring specifically to the 2-class case, precise costs may be unknown, and a different optimisation strategy needs to be taken. Two types of classification errors occur in the 2-class case, namely the false negative rate (FN_r), consisting of class 1 errors misclassified as class 2, and the false positive rate (FP_r) in the opposite case. In this situation, it is often desirable to specify a fixed FN_r or FP_r, and select a model with the corresponding lowest FP_r or FN_r. This is referred to as Neyman-Pearson optimisation[1]. The optimisation is in selecting the best model. A well-known classifier analysis approach that is useful in this context is receiver operator characteristic (ROC) analysis [4], consisting of a graph representing all possible classification conditions as the classification weights are varied. It is important to note that in the 2-class case, any FN_r or FP_r specification can be achieved, and a corresponding weight obtained. This Neyman-Pearson design is useful in a number of areas such as detection problems, and medical decision making.

In the multiclass case, several possible classification outputs result, with $C^2 - C$ interclass errors, and C intraclass correct classifications, in a C-class problem. In this situation, it has been shown that both cost-sensitive optimisation and Neyman-Pearson optimisation extend theoretically [5], involving the use of multiclass ROC hypersurfaces. In the case of Neyman-Pearson optimisation, it has thus been shown that specifying a particular classification error in a C-class problem is achievable, with the subsequent objective to minimise all other $C^2 - C$ classification errors. However, a practical situation may demand the specification of a number of classification errors, and subsequent minimisation of remaining classification errors. This type of optimisation could

[1] A more fundamental formalisation and derivation of Neyman-Pearson theory in a detection context can be found in [3], with application in a classifi cation sense in [1].

be applicable to areas such as medical decision making involving multiple diseases, or remote sensing, in which the objective is to identify various types of terrain, and minimise the false positive rates with respect to the desired classes of all other terrain types. This type of analysis has not yet (to our knowledge) been studied. In this paper we formalise multiclass ROC analysis, allowing for an implementation of a multiclass Neyman-Pearson optimisation procedure. Extendibility and limitations are identified, primarily discussing the fact that a feasible point on the ROC hypersurface is only guaranteed if just one interclass classification error is specified. However, some experiments show that in practical situations, specifying a number of classification outputs does result in a feasible solution. This is a very interesting result, which may have a large potential for Neyman-Pearson type problems in the multiclass case.

The paper is structured as follows: Section 2 formalises multiclass classification, allowing for derivation of the various inter- and intra-class outputs inherent to multiclass classifiers. The foundation of the Neyman-Pearson optimisation is a multiclass ROC framework, defined in Section 3. Neyman-Pearson optimisation using ROC analysis is then discussed in Section 4, with some experiments to demonstrate the optimisation in realistic situations in Section 5. A final discussion and conclusions are given in Section 6.

2. Formalisation of multiclass classification

Consider a multiclass problem with C classes, $\omega_1, \omega_2, \ldots, \omega_C$, with input data $\mathbf{x}$, and d dimensions. The objective of a multiclass classifier $f(\mathbf{x})$ is to discriminate between the various classes as well as possible, according to the requirements of the problem. The classifier is usually trained based on independent training data. Many strategies are possible, but the outcome is typically a vector of continuous values, with higher values supporting higher confidence (e.g. probability, distance to a decision boundary, support vector etc.) in particular classes. For class i, the classifier output is written as $f(\omega_i|\mathbf{x})$. In the density-based case, this would be the posterior estimate for the respective class. Irrespective of the classification type, the class assignment is typically:

$$argmax_{i=1}^{C} f(\omega_i|\mathbf{x}) \qquad (1)$$

For example in Figure 1, a scatterplot is shown of a 5-class synthetic example, together with the multiclass decision boundary of a Bayes quadratic classifier.

An observation that can be made is that there are a number of Degrees Of Freedom (DOF) that can be used

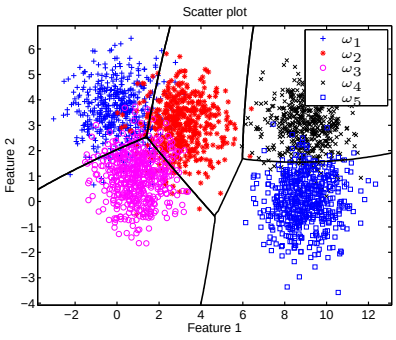

Figure 1: Scatter plots of a synthetic 5-class problem, illustrating the decision boundaries (degrees of freedom) for one operating point.

to adjust the classifier (analogous to *thresholds*). In fact, there are $C-1$ degrees of freedom in a $C-$class problem. Thus in the 2-class case, there is only one DOF, and in the 10-class case, there are 9 DOF to optimise the classifier.

When evaluating a classifier, both intraclass outputs (correct classifications), and interclass classifications (between-class errors) are of interest. These are specified by a confusion matrix CM, with a size C^2. Thus the number of errors increases quadratically with increasing C. The CM is typically constructed by applying an independent test set to a trained classifier. In the 2-class case, only 2 interclass errors occur, namely the familiar False Negative rate (FN_r) and False Positive rate (FP_r), with respect to one of the classes. ROC analysis involves inspecting the interplay between these two errors as a function of the single weight/threshold. The CM is defined in Table 1. The output between class i and j is denoted $cm_{i,j}$. CM outputs are usually normalised by the absolute number of objects N_i per class ω_i, $N = [N_1, N_2, \ldots, N_C]^T$, resulting in

		estimated				
		ω_1	ω_2	$\ldots$	ω_C	
true	ω_1	$cm_{1,1}$	$cm_{1,2}$	$\ldots$	$cm_{1,C}$	N_1
	ω_2	$cm_{2,1}$	$cm_{2,2}$	$\ldots$	$cm_{2,C}$	N_2
	$\ldots$					
	ω_C	$cm_{C,1}$	$cm_{C,2}$	$\ldots$	$cm_{C,C}$	N_C

Table 1: A multi-class confusion matrix.

the normalised confusion matrix Ξ, where each element is now referenced as $\xi_{i,j}$, and $\xi_{i,j} = \frac{cm_{i,j}}{N(i)}$. We now consider the computation of each element in Ξ via an example. Consider the class-conditional distributions in Figure 2, consisting of five Gaussian-distributed classes $\omega_1, \omega_2, \ldots, \omega_5$, with means occurring at $\mu_1 = -6, \mu_2 = -3, \mu_3 = 0, \mu_4 = 6, \mu_5 = 9$, and unit variance. The respective priors are assumed to be equal. The class-conditional density of class i is denoted $p(\mathbf{x}|\omega_i)$, and the prior $p(\omega_i)$.

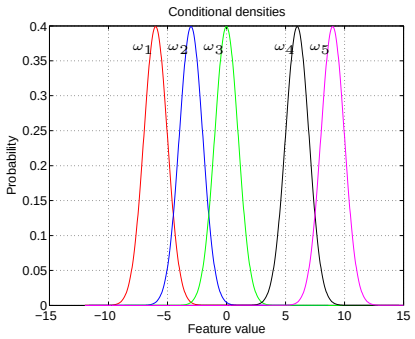

Figure 2: Probability density functions for the 5-class example with known distributions.

This example results in a 5×5 element confusion matrix. In order to compute each confusion $\xi_{i,j}$ (the percentage of of ω_i misclassified as ω_j), the following integration is performed:

$$\xi_{i,j}(\mathbf{x}) = p(\omega_i) \int p(\mathbf{x}|\omega_i) I_{ij}(\mathbf{x}) dx \qquad (2)$$

The indicator function $I_{ij}(\mathbf{x})$ specifies the relevant domain:

$$I_{ij}(\mathbf{x}) = \begin{cases} 1 \text{ if } p(\omega_j|\mathbf{x}) > p(\omega_k|\mathbf{x}) \ \forall k, \\ k \neq j,\ i \neq j \\ 1 \text{ if } p(\omega_i|\mathbf{x}) > p(\omega_k|\mathbf{x}) \ \forall k, \\ k \neq i,\ i = j \\ 0 \text{ otherwise} \end{cases} \qquad (3)$$

Equation 2 allows any confusion matrix output to be computed, generalised for both diagonal elements (performances), and off-diagonal elements (errors).

3. Multiclass ROC analysis

Referring again to Figure 1, the plot shows only a single operating point, corresponding to a single weight setting. In fact, any combination of weightings results in a different operating point (the challenge in multiclass optimisation is in understanding the relation between a weight modification and the corresponding alteration of the confusion matrix). Application of this weighting Φ involves modification of Equation 1, in which the class assignment is now based on each output $f(\omega_i|\mathbf{x})$, multiplied by a corresponding weight, denoted ϕ_i, resulting in:

$$argmax_{i=1}^{C} \phi_i f(\omega_i|\mathbf{x}) \qquad (4)$$

The concept of classifier optimisation can be formalised as the process by which the optimal set of weights is found to suit the problem at hand. ROC analysis involves generation of a hypersurface consisting of all possible combinations of Φ, where $\Phi = [\phi_1, \phi_2, \ldots, \phi_{C-1}, 1 - \phi_1]$. A multiclass ROC consists of $C^2 - C$ dimensions (diagonal elements are superfluous), which can be constructed using a similar equation to 2. In this case, each output between class i and j is weighted by ϕ_i as follows:

$$\xi_{i,j}(\mathbf{x}|\Phi) = \phi_i p(\omega_i) \int p(\mathbf{x}|\omega_i) I_{ij}(\mathbf{x}|\Phi) dx \qquad (5)$$

The indicator function $I_{ij}(\mathbf{x}|\Phi)$ is as in 3, except each posterior is multiplied by the corresponding class weight. Note that there are only $C-1$ weights, and thus $\phi_C = 1 - \phi_1$.

Consider the 2-class case between ω_1, and ω_2, in which a weighting ϕ is applied to obtain the most appropriate threshold. In this case, the classifier output can be written as:

$$p(\mathbf{x}|\phi) = [\phi p(\omega_1|\mathbf{x}) \ (1 - \phi) p(\omega_2|\mathbf{x})] \qquad (6)$$

For $0 \leq \phi \leq 1$, $\xi_{1,2}$ and $\xi_{2,1}$ vary across all possible combinations, resulting in the ROC plot. In the multiclass case ($C > 2$), Equation 5 can be used to construct a multiclass ROC, resulting in a $C^2 - C$ dimensional surface. For example, in the 5-class Gaussian example, a 4-D grid of weights was computed ($C-1$ DOF), and a step resolution of 30 was chosen, resulting in $8.1e5$ weight combinations. Application of Equation 5 resulted in a 20 dimensional surface. Even though this surface cannot be visualised, for demonstration purposes, Figure 3 shows the ROC between the dimensions $\xi_{1,2}$, $\xi_{2,1}$, and $\xi_{2,3}$.

4. Neyman-Pearson optimisation

Classifier optimisation in a Bayesian framework involves estimates (or given) class conditional densities (pdf's), denoted $f(\mathbf{x}|\omega_i), \forall i$, prior estimates $\pi_i, \forall i$, and misclassification costs corresponding to each off-diagonal output of the confusion matrix CM, denoted

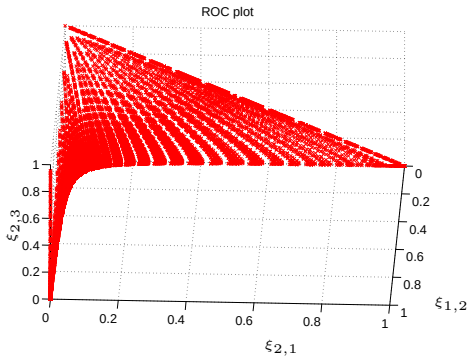

Figure 3: Plotting $\xi_{1,2}$, $\xi_{2,1}$, and $\xi_{2,3}$ for the example in Figure 2 as a function of Φ.

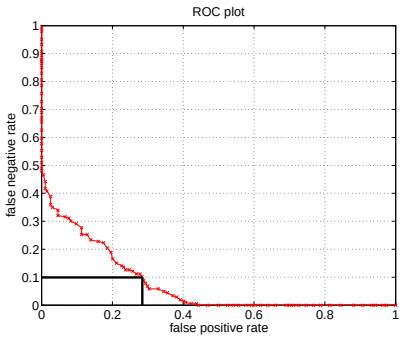

Figure 4: ROC plot example illustrating Neyman-Pearson design. In this case the false negative rate has been specified at 10.00%.

$c_{ij}, i \neq j$. The optimisation involves deriving the optimal weight vector Φ such that the overall system loss is minimised, where the loss is computed via:

$$L = \sum_{i=1}^{C} \frac{\phi_i \pi_i}{N_i} \Big(\sum_{j=1, i \neq j}^{C} c m_{i,j} c_{ij} \Big) \tag{7}$$

In some problems (such as detection problems and medical decision making), respective costs cannot be defined, and the cost-sensitive optimisation procedure cannot be taken. However, it is assumed that the individual classes can be modelled in some way, e.g. pdf estimates $f(\mathbf{x}|\omega_i), \forall i$. In the 2-class case, this implies that an ROC curve can be estimated between $\xi_{1,2}$ and $\xi_{2,1}$ (referring to the previous example), the false negative, and false positive rates respectively, written as a function of the weight ϕ as (with population priors $\pi_i \ \forall i$ estimated):

$$\begin{aligned} \xi_{1,2}(\mathbf{x}|\phi) &= \phi \pi_1 \textstyle\int f(\mathbf{x}|\omega_1) I_{12}(\mathbf{x}|\phi) dx \\ \xi_{2,1}(\mathbf{x}|\phi) &= \phi \pi_2 \textstyle\int f(\mathbf{x}|\omega_2) I_{21}(\mathbf{x}|\phi) dx \end{aligned} \tag{8}$$

In Neyman-Pearson optimisation, either $\xi_{1,2}$ or $\xi_{2,1}$ is fixed at a specified value α. The optimisation then involves computing a value for the weighting ϕ such that the specification holds, and the dependent variable is obtained. In the 2-class case, optimisation occurs only across models. This is illustrated on the ROC plot in Figure 4 (plotting the false negative rate against false positive rate). In this example, $\xi_{1,2}$ is fixed at 10.00%. The ROC curve is a useful tool in this case, immediately resulting in the corresponding $\xi_{2,1}$, which is approximately 26%. The ϕ resulting in this point can then be used as the optimal Neyman-Pearson threshold (Note that each point on the ROC plot corresponds to some ϕ value, and in the multiclass ROC, each point on the ROC hypersurface corresponds to a C dimensional weight vector Φ).

The Neyman-Pearson optimisation is well studied and extensively used in 2-class problems, but this is not the case in the multiclass context. A practical implementation involves the use of multiclass ROC analysis which is also a relatively new research area (see some recent works in [6],[7], [5]). This was also formalised in Section 3. Recently, the theoretical extension of Neyman-Pearson optimisation to multiclass optimisation was proven in [5], with applicability to multiple-diagnosis in medical decision making. This showed that by fixing a single classification error in the confusion matrix, a solution is guaranteed. Thus the first step involves specifying some classification error, and subsequently to return a Φ corresponding to a minimisation of all other $C^2 - C - 1$ classification errors in Ξ. Section 5 demonstrates this optimisation in a few experiments.

In this paper, we also wish to generalise the optimisation procedure such that multiple errors in Ξ can be specified, followed by a subsequent minimisation of remaining errors. It is obvious that when specifying more than one error, a solution is not guaranteed (multiple dependencies). However, we argue that in many practical situations, it may still be feasible to specify a number of outputs, and obtain a solution. The usefulness is, however, data and problem dependent. Next, an algorithm is developed that is generalised in the sense that the original Neyman-Pearson optimisation holds (specifying one error), and can also be used to specify multiple errors.

The proposed optimisation algorithm is implemented by the introduction of two $C \times C$ matrices, namely M_I and M_e (many other implementations are possible). The elements of these correspond to confusion matrix outputs, allowing for a direct input of required specifications. The M_I matrix is a binary indicator matrix specifying which errors are to be specified. A

Algorithm 1: Multiclass Neyman-Pearson optimisation
Inputs: ROC resolution $step$, C classes, trained classifier D, error specification matrix M_e, specification index matrix M_I, independent test set $\mathbf{x}_{ts}$
Outputs: Optimal weights Φ_{opt}
1) Construct weight matrix Φ, with resolution $step$
2) Compute $C \times C$ multiclass ROC E, using Equation 5, with $step$ resolution, applying $\mathbf{x}_{ts}$ to D, for all Φ
3) $m = 0$
For each row of E i, and column j ($i \neq j$):
If ($M_I(i,j) = 1$ and $M_e(i,j) > 0$)
$ind_{(m)} = E(i,j,k) \leq M_e(i,j)\ \forall k$)
Increment m
End
4) Find hypersurface regions that fulfil M_e specifications:
$ind_{com} = ind_{(1)} \cap ind_{(2)} \cap \ldots ind_{(m)}$
If size of $ind_{com} = 0$, no solution - specifications not met
5) Minimise all other non-specified errors
$m = 0$
For each row i, and column j of E ($i \neq j$):
If ($M_I(i,j) = 0$)
$err_{(m)} = \sum(\sum_{i=1}^{C} \sum_{j=1}^{C}(E(i,j,k)\ \forall k)\ i \neq j)$
Increment m
End
Index corresponding to minimum error:
$ind_{min} = index(min(err_{(m)}))$, $\forall m$
Final classifier weights: $\Phi_{opt} = \Phi(ind_{min})$
Return: Φ_{opt}, or no solution

1 at position $M_I(i,j)$ indicates that $\xi_{i,j}$ is to be specified, and a 0 at position $M_I(k,l)$ indicates that $\xi_{k,l}$ is to be minimised. Working in conjunction with M_I is M_e, a matrix used to specify the respective errors as specified in M_I. For example, requiring an error rate lower or equal to 5.00% for $\xi_{3,1}$ would require a $M_I(3,1) = 1$ and $M_e(3,1) = 0.05$.

Algorithm 1 presents a practical procedure that can be used to perform a multiclass Neyman-Pearson optimisation[2].

In step 2, the multi-class ROC is computed, denoted E using a matrix of weights with $C-1$ columns, as per step 1 (we assume dense sampling). It is convenient to store this matrix in a similar form to the confusion matrix, resulting in a $C \times C \times step^{C-1}$ dimensional matrix. The diagonal elements are ignored, and need not be computed. In step 3, all ROC dimensions corresponding to error specifications are inspected, and the portion of the surface (in that dimension m) fulfilling the specification is stored. The range corresponding to dimension m is stored in $ind_{(m)}$. Note that the leq operator is used here since the ROC is discretised. The next step 4 involves intersecting each of these ranges, and identifying common indices (denoted ind_{com}). If no intersection occurs, this implies that the specifications cannot be met (no point on the ROC hypersurface fulfils specifications). If a solution does exist, step 5 involves minimisation of all non-specified errors. This is achieved by summing all non-specified errors for each Φ weighting corresponding to indices ind_{com}. The weighting resulting in the lowest error sum is then chosen as Φ_{opt}.

[2]This algorithm can easily be adapted for the case in which the distributions are known. We focus here on the practical situation in which the true distributions are unknown, and data samples are assumed to originate from the true distribution.

5. Experiments

To demonstrate the optimisation in a practical situation, the *Satellite* dataset is considered[3]. This dataset consists of 6435 multi-spectral values of a satellite image, with 36 dimensions (4 spectral bands in a 9 pixel neighbourhood). Six classes have been identified to characterise the topography. These consist of *red soil*, *cotton crop*, *grey soil*, *damp grey soil*, *soil with vegetable stubble*, and *very damp grey soil* classes. In experiments, all the *grey soil* classes are grouped together, resulting in a 4 class problem. As per dataset recommendations, the first 4435 spectra are used as a training set, and the remaining data as the test set.

The first experiment involves training a base classifier on the data. The first 17 principal components are used to represent the spectra, and a Bayes linear discriminant classifier is then trained on this representation. The following normalised confusion results following application of the independent test set:

	red	cotton	veget	grey
red	0.9491	0.0000	0.0081	0.0428
cotton	0.0047	0.8826	0.0986	0.0141
veget	0.0553	0.0046	0.7051	0.2350
grey	0.0009	0.0000	0.0037	0.9954

Following classical Neyman Pearson, we now experiment by only specifying single classification errors, and minimising all others. Two experiments are demonstrated. In the first experiment, $\epsilon_{veget,grey} = 8.00\%$, and in the second, $\epsilon_{cotton,veget} = 5.00\%$, resulting in the following two normalised confusion matrices:

	red	cotton	veget	grey
red	0.9572	0.0000	0.0204	0.0224
cotton	0.0047	0.8826	0.1127	0.0000
veget	0.0461	0.0046	0.8756	0.0737
grey	0.0009	0.0000	0.0454	0.9537

	red	cotton	veget	grey
red	0.9552	0.0000	0.0000	0.0448
cotton	0.0047	0.9061	0.0469	0.0423
veget	0.0599	0.0046	0.4516	0.4839
grey	0.0009	0.0000	0.0009	0.9981

As expected, these result in a solution. Comparing with the original normalised confusion matrix, it can clearly be seen that this new operating point has resulted

[3]UCI repository of machine learning databases, ftp://ftp.ics.uci.edu/pub/machine-learning-databasesx.

in different compromises between the various classes (which may or may not be acceptable). In the next experiments, we attempt to specify a number of interclass errors, and minimise the remaining ones. Three experiments are undertaken with the following specifications:

1. Specify $\epsilon_{veget,red} = 5.00\%$, $\epsilon_{red,grey} = 5.00\%$, and $\epsilon_{veget,grey} = 5.00\%$.
2. Specify $\epsilon_{veget,red} = 5.00\%$, $\epsilon_{cotton,veget} = 15.00\%$, and $\epsilon_{veget,grey} = 5.00\%$.
3. Specify $\epsilon_{veget,red} = 5.00\%$, $\epsilon_{cotton,veget} = 10.00\%$, and $\epsilon_{veget,grey} = 5.00\%$.

The optimisation is successful in the first two experiments, but fails in the third. This implies that the specifications cannot be achieved by the classifier in this case. The first two cases result in the following respective normalised confusion matrices:

```
       |   red     cotton     veget     grey
------|-----------------------------------
red    | 0.9593    0.0000    0.0224    0.0183
cotton| 0.0047    0.8826    0.1127    0.0000
veget  | 0.0461    0.0046    0.9171    0.0323
grey   | 0.0009    0.0000    0.1186    0.8804

       |   red     cotton     veget     grey
------|-----------------------------------
red    | 0.9593    0.0000    0.0224    0.0183
cotton| 0.0047    0.8592    0.1362    0.0000
veget  | 0.0461    0.0046    0.9171    0.0323
grey   | 0.0009    0.0000    0.1186    0.8804
```

These normalised confusion matrices should be compared to the original one. It can be seen in the first case that the output $\xi_{veget,veget}$ has improved from around 70% to over 90%, but $\xi_{grey,grey}$ is around 11% lower. Note that many solutions are often possible. These specifications were also not met in the first examples, in which single errors were specified. These experiments (which are limited due to space constraints) demonstrate the potential and practicality of the proposed approach. Once a multiclass ROC has been computed, a practitioner can quickly and easily enter and modify a specification via the matrices M_I and M_e.

6. Conclusion

This paper considered the applicability of Neyman-Pearson optimisation to multiclass problems. This is a well-studied and extensively used technique in 2-class problems (stemming from detection applications), applicable in situations where costs cannot be defined, and it is more practical to specify a fixed true- or false- positive rate. ROC analysis is a tool facilitating this optimisation, allowing for a direct query of the corresponding classification threshold/weight. In the multiclass case, several possible classification outputs result. Work performed in [5] showed that Neyman-Pearson optimisation does hold in the multiclass case, in which case a single output/error is specified. A more practical multiclass scenario may involve the necessity to specify multiple error rates. This paper investigated this empirically, with results showing that this is possible, but a solution is not guaranteed. Since the optimisation hinges on ROC analysis, a multiclass ROC framework was developed, and an algorithm designed to perform the optimisation. The algorithm attempts to find regions on the ROC hypersurface that meets the specifications, and then subsequently minimises all other classification errors. The algorithm terminates in the case that a solution cannot be found. Some real experiments demonstrated the potential of this approach. On-going research is focused on obtaining efficient representations of the ROC hypersurface for large C problems, which remains a significant challenge to multiclass design.

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, The Netherlands.

References

[1] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, Wiley - Interscience, second edition, 2001.

[2] F. Provost and T. Fawcett, "Robust classification for imprecise environments," *Machine Learning*, vol. 42, pp. 203–231, 2001.

[3] D. Kazakos and P. Papantoni-Kazakos, *Detection and Estimation, ISBN 0-7167-8181-6*, Computer Science Press, 1st edition, 1990.

[4] C. Metz, "Basic principles of ROC analysis," *Seminars in Nuclear Medicine*, vol. 3, no. 4, 1978.

[5] D.C. Edwards, C.E. Metz, and M.A. Kupinski, "Ideal observers and optimal ROC hypersurfaces in N-class classification," *IEEE Transactions on Medical Imaging*, vol. 23, no. 7, pp. 891–895, July 2004.

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

[7] 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*, pp. 416–423, 2003.