Employing Optimized Combinations of One-Class Classifiers for Automated Currency Validation

Chao He [a] Mark Girolami [a,*] Gary Ross [b]

[a] *School of Information and Communication Technologies, University of Paisley, Paisley PA1 2BE, United Kingdom.*

[b] *NCR Financial Solutions Group Ltd, Dundee DD2 3XX, United Kingdom.*

Abstract

Automated currency validation requires a decision to be made regarding the authenticity of a banknote presented to the validation system. This decision often has to be made with little or no information regarding the characteristics of possible counterfeits as is the case for issues of new currency. A method for automated currency validation is presented which segments the whole banknote into different regions, builds individual classifiers on each region and then combines a small subset of the region specific classifiers to provide an overall decision. The segmentation and combination of region specific classifiers to provide optimized false positive and false negative rates is achieved by employing a genetic algorithm. Experiments based on high value notes of Sterling currency were carried out to assess the effectiveness of the proposed solution.

Key words: Novelty detection; One-Class classifier; Classifier combination; Automated currency validation; Genetic algorithm

1 Introduction

There is a growing need for banks and financial institutions to provide their customers with an expanding range of high standard financial self-services, the most popular and well-established service introduced among them is the Automatic Teller Machine (ATM). There is an increasing requirement to provide secure deposit transaction capabilities to complement traditional ATM withdrawal services. Although there are some ATM systems which can accept cheque and banknote deposits, e.g. the *deposit-in-envelope* ATM systems generally adopted by Canadian banks, the validation and confirmation of the authenticity of the enclosed currency or cheques cannot be completed by the ATM. Staff from the bank are then required to manually confirm the validity of the deposit before completing the transaction and authorizing the recirculation of any banknotes included in the envelope. The automated verification and validation of any currency at the ATM would then allow these deposited banknotes to be recirculated as part of the withdrawal service of the ATM. Such a service places additional demands on the existing ATM technology which must then be able to provide a means of discrimination between deposited notes of legal tender and those which are counterfeit. The fundamental danger of *automated currency validation* is the possibility of utilizing a currency recycling facility within ATM's as a means of circulating counterfeit currency.

Considering *automated currency validation* within the classical pattern recog-

* Corresponding author. Tel: +44-141-848-3317; fax: +44-141-848-3542.
Email addresses: chao.he@paisley.ac.uk (Chao He), mark.girolami@paisley.ac.uk (Mark Girolami), rossgar@exchange.scotland.ncr.com (Gary Ross).

nition paradigm it is assumed that a representative sample of both genuine and counterfeit currency is available in devising a discriminator. However, in this particular application it is seldom possible to acquire a suitably representative sample of possible counterfeits to build an effective binary discriminator with good generalization ability. Indeed when new notes of a given currency are issued there are no examples of counterfeits available with which to construct a discriminative template.

In this paper, we investigate the utilization of what has been termed 'one-class classifiers' [1,2] or 'novelty detectors' [3] in assessing the validity of an observed banknote when, at best, sufficient numbers of representative examples of genuine notes and a potentially small, not necessarily representative, number of counterfeits may also be available.

As noted in standard classification problems a single classifier may not sufficiently exploit the necessary discriminative characteristics of the data and so the combination of multiple classifiers in a principled manner to improve discrimination performance has been the focus of much recent investigation [4–7,17,18,21]. In this application we exploit the improved discrimination power potentially available when appropriately combining multiple 'one-class classifiers' *trained* on different sub-sets of the available feature set. It is difficult in the counterfeiting process to provide a uniform quality of imitation across the whole note, and certain regions of the note may be more difficult than others to copy successfully. Therefore, by segmenting the whole note into a number of regions, then building individual one-class classifiers on each region and combining these in an appropriate manner it is hoped that a higher level of discrimination will be obtained.

In the absence of any expert knowledge regarding the possible counterfeiting process, the optimal segmentation of the note into various regions followed by their combination requires an exhaustive search over all possible segmentations and combinations. A practical solution to this problem is to use any one of a number of possible stochastic search methods and in this paper a Genetic Algorithm (GA) [8] is employed. This provides a means of building the optimized banknote validation system automatically from the reference set (genuine notes) and a small number of counterfeit examples that may be available, therefore there is no limitation on currency type in employing this proposed solution.

The remainder of the paper is organized as follows. Section 2 presents two methods for one-class classification considered for this application. In Sect. 3, the whole solution of note segmentation and classifier combination is detailed. The results of experiments [1] are reported in Sect. 4. Finally, conclusions and discussion of possible future work is given in Sect. 5.

2 One-class classification

Somewhat differently from conventional binary classification, one-class classification faces the problem that there is only information available regarding one class. In the case of currency validation as explained above it is very difficult to acquire sufficient numbers of counterfeit examples, therefore, the focus of effort shifts to defining a boundary around the known class such that examples falling outwith this boundary are deemed not to belong to the known

[1] The experiments were conducted on currency from the Bank of Scotland (BOS).

class. Small numbers of examples of counterfeits however can potentially be employed in defining the boundary where we are not attempting to estimate a class conditional density for the counterfeit class. This is similar in spirit to the approaches adopted in [1,3] where 'information' regarding the second class can be exploited from a small number of examples.

There are various one-class classification methods available, see for example [1–3]. The statistical methods are based on maximization of the log-likelihood ratio under the null-hypothesis that the observation under consideration is drawn from the target class and these include the D^2 test [9] which assumes a multivariate Gaussian distribution for the target class (genuine currency). In the case of an arbitrary non-Gaussian distribution the density of the target class can be estimated using for example a mixture of Gaussians and the distribution of the log-likelihood ratio under the null-hypothesis can be obtained by sampling techniques such as the bootstrap [10].

Other methods which can be employed for one-class classification are Support Vector Data Domain Description (SVDD) [1], also known as 'support estimation' [3] and Extreme Value Theory (EVT) [11]. In SVDD the support of the data distribution is estimated which obviates the difficult problem of density estimation in the case of data sparsity in high dimensional situations, whilst EVT estimates the distribution of extreme values. For this particular application, large numbers of examples of genuine notes are available and the feature space, which will be described in the following sections, is 6-dimensional and so in this case it is possible to obtain reliable estimates of the target class distribution. Therefore, it is desirable to choose one-class classification methods that can estimate the density distribution explicitly. In the following subsec-

tions, we will then introduce one-class classification methods based on the parametric D^2 test and the semi-parametric test based on a Mixture of Gaussians which employs the Bootstrap [12].

2.1 Statistical Hypothesis Tests

Consider N independent and identically distributed p-dimensional vector samples (the feature set for each banknote) $\mathbf{x}_1, \cdots, \mathbf{x}_N \in \mathcal{C}$ with an underlying density function with parameters $\boldsymbol{\theta}$ given as $p(\mathbf{x}|\boldsymbol{\theta})$. The following hypothesis test is given for a new point $\mathbf{x}_{N+1}$ such that $\mathcal{H}_0 : \mathbf{x}_{N+1} \in \mathcal{C}$ vs. $\mathcal{H}_1 : \mathbf{x}_{N+1} \notin \mathcal{C}$, where $\mathcal{C}$ denotes the region where the null hypothesis is true and is defined by $p(\mathbf{x}|\boldsymbol{\theta})$. Assuming that the distribution under the alternate hypothesis is uniform then the standard log-likelihood ratio for the null and alternate hypothesis

$$\lambda = \frac{\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathcal{L}_0(\boldsymbol{\theta})}{\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathcal{L}_1(\boldsymbol{\theta})} = \frac{\sup_{\boldsymbol{\theta}} \prod_{n=1}^{N+1} p(\mathbf{x}_n|\boldsymbol{\theta})}{\sup_{\boldsymbol{\theta}} \prod_{n=1}^{N} p(\mathbf{x}_n|\boldsymbol{\theta})} \tag{1}$$

can be employed as a test statistic for the null-hypothesis. In this application we can use the log-likelihood ratio as test statistic for the validation of a newly presented note.

2.1.1 Feature vectors with multivariate Gaussian density

Under the assumption that the feature vectors are multi-variate Gaussian a test, which emerges from the above likelihood ratio, to assess whether each point in a sample shares a common mean is detailed in [9]. Consider N inde-

pendent and identically distributed p-dimensional vector samples $\mathbf{x}_1, \cdots, \mathbf{x}_N$ from a multi-variate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{C}$, whose sample estimates are $\hat{\boldsymbol{\mu}}_N$ and $\hat{\mathbf{C}}_N$. From the sample consider a random selection denoted as $\mathbf{x}_0$, the associated squared Mahalanobis distance [2]

$$D^2 = (\mathbf{x}_0 - \hat{\boldsymbol{\mu}}_N)^{\mathrm{T}} \hat{\mathbf{C}}_N^{-1} (\mathbf{x}_0 - \hat{\boldsymbol{\mu}}_N) \tag{2}$$

can be shown to be distributed as a central F-distribution with p and $N-p-1$ degrees of freedom by

$$F = \frac{(N-p-1)ND^2}{p(N-1)^2 - NpD^2} \ . \tag{3}$$

Then, the null hypothesis of a common population mean vector for $\mathbf{x}_0$ and the remaining $\mathbf{x}_i$ will be rejected if

$$F > F_{\alpha;\, p,\, N-p-1} \ , \tag{4}$$

where $F_{\alpha;\, p,\, N-p-1}$ is the upper $\alpha 100\%$ point of the F-distribution with $(p, N-p-1)$ degrees of freedom. We can make use of the following incremental estimates of the mean and covariance in devising a test for new examples which do not form part of the original sample when an additional datum $\mathbf{x}_{N+1}$ is made available, i.e. the mean

$$\hat{\boldsymbol{\mu}}_{N+1} = \frac{1}{N+1}\{N\hat{\boldsymbol{\mu}}_N + \mathbf{x}_{N+1}\} \ , \tag{5}$$

and the covariance

$$\hat{\mathbf{C}}_{N+1} = \frac{N}{N+1}\hat{\mathbf{C}}_N + \frac{N}{(N+1)^2}(\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}}_N)(\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}}_N)^{\mathrm{T}} \ . \tag{6}$$

[2] This is a generalized distance where the metric is defined by $\hat{\mathbf{C}}_N$.

By using the expressions of (5), (6) and the matrix inversion lemma [3], Equation (2) for an N-sample reference set and an $N+1$'th test point becomes

$$D^2 = \boldsymbol{\delta}_{N+1}^{\mathrm{T}} \hat{\mathbf{C}}_{N+1}^{-1} \boldsymbol{\delta}_{N+1} \ , \tag{7}$$

where

$$\boldsymbol{\delta}_{N+1} = \frac{N}{N+1}(\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}}_N) \ , \tag{8}$$

and

$$\hat{\mathbf{C}}_{N+1}^{-1} = \frac{N+1}{N}\hat{\mathbf{C}}_N^{-1} - \frac{\hat{\mathbf{C}}_N^{-1}\boldsymbol{\delta}_{N+1}\boldsymbol{\delta}_{N+1}^{\mathrm{T}}\hat{\mathbf{C}}_N^{-1}}{N + \frac{N}{N+1}\boldsymbol{\delta}_{N+1}^{\mathrm{T}}\hat{\mathbf{C}}_N^{-1}\boldsymbol{\delta}_{N+1}} \ . \tag{9}$$

Denoting $\boldsymbol{\delta}_{N+1}^{\mathrm{T}}\hat{\mathbf{C}}_N^{-1}\boldsymbol{\delta}_{N+1}$ by $D_{N+1,N}^2$, then

$$D^2 = \frac{N+1}{N} D_{N+1,N}^2 \left\{ 1 - \frac{D_{N+1,N}^2}{D_{N+1,N}^2 + N + 1} \right\} \ . \tag{10}$$

So a new point $\mathbf{x}_{N+1}$ can be tested against an estimated and assumed normal distribution for a common estimated mean $\hat{\boldsymbol{\mu}}_N$ and covariance $\hat{\mathbf{C}}_N$. Employing the log-likelihood ratio for the one-class hypothesis test we can derive the test statistic (10) directly, Appendix A. The assumption of multivariate Gaussian feature vectors often does not hold in practice, though may be an appropriate pragmatic choice in many applications. However, we will relax this assumption and consider arbitrary densities in the following section.

2.1.2 Feature Vectors with arbitrary Density

The density under a mixture model has the following standard form:

$$p(\mathbf{x}) = \sum_{j=1}^{M} p(\mathbf{x}|j)P(j), \tag{11}$$

[3] The inverse of $\mathbf{B} = \mathbf{A} + \mathbf{u}\mathbf{u}^{\mathrm{T}}$ is $\mathbf{B}^{-1} = \mathbf{A}^{-1} - (1 + \mathbf{u}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{u})^{-1}\mathbf{A}^{-1}\mathbf{u}\mathbf{u}^{\mathrm{T}}\mathbf{A}^{-1}$.

where, $P(j), j = 1, \ldots, M$ are the mixing parameters. They are chosen to satisfy the constraints $\sum_{j=1}^{M} P(j) = 1$ and $P(j) \geq 0$. The component density functions $p(\mathbf{x}|j)$ are normalized so that $\int p(\mathbf{x}|j)d\mathbf{x} = 1$. We consider Gaussian mixture models such that

$$p(\mathbf{x}|j) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_j|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x} - \boldsymbol{\mu}_j)\}, \tag{12}$$

and the required parameters can be estimated using the Expectation Maximization (EM) algorithm[13]. This density can then be employed in computing the log-likelihood ratio. Unlike the case for the multivariate Gaussian distribution there is no analytic distribution for the test statistic (λ) under the null-hypothesis so to obtain this distribution numerical bootstrap [12] methods can be employed to obtain the otherwise non-analytic null distribution under the mixture of Gaussians density and so the various critical values of λ_{crit} can be established from the empirical distribution obtained. It can be shown [10] that in the limit as $N \to \infty$, the likelihood ratio can be estimated by the following

$$\lambda = \frac{\sup_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}_0(\boldsymbol{\theta})}{\sup_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}_1(\boldsymbol{\theta})} \to p(\mathbf{x}_{N+1}; \hat{\boldsymbol{\theta}}_N) \tag{13}$$

where $p(\mathbf{x}_{N+1}; \hat{\boldsymbol{\theta}}_N)$ denotes the probability density of $\mathbf{x}_{N+1}$ under the model estimated by the original N samples.

After generating B bootstrap samples from the reference data set and using each of these to estimate the parameters of the mixture distribution $\hat{\boldsymbol{\theta}}_N^i$, B bootstrap replicates of the test statistic $\lambda_{crit}^i, i = 1, \ldots, B$ can be obtained by randomly selecting an $N + 1$'th sample and computing $p(\mathbf{x}_{N+1}; \hat{\boldsymbol{\theta}}_N^i) \approx \lambda_{crit}^i$. By ordering λ_{crit}^i in ascending order, the critical value α can be defined to

reject the null-hypothesis at the desired significance level if $\lambda \leq \lambda_\alpha$, where λ_α is the j th smallest value of λ^i_{crit}, and $\alpha = j/(B+1)$. The following section now considers the combination of one-class classifiers in an application specific manner.

3 One-Class Classifier Combination

In recent years multiple classifier combination systems have received increasing attention in the literature [4–7] as it has been realized that such systems can potentially be more robust and more accurate than a single classifier alone [17]. Preliminary first steps in developing a rigorous theoretical framework for the analysis of multiple-classifier systems have been made [18]. However, since the best combination of a set of classifiers depends on the application and on the classifiers to be combined, there is no single, best combination scheme nor any unequivocal relationship between the accuracy of a multiple classifier system and the individual constituent classifiers. One approach is to generate a large number of classifiers and then to select the best combinations to use in particular regions of the input space. This raises two questions: first, how to generate a suitable set of classifiers, and then how best to combine them [17]. Based on this particular application, the first question is answered by segmenting the note, detailed in Sect. 3.1; and due to the drawback of the exponential growth in the search space of possible segmentations and combinations, stochastic search based on a genetic algorithm is developed in Sect. 3.2 to answer the second question.

3.1 Note segmentation and classifier combination strategy

Two different ways to combine one-class classifiers have been considered [4]. The first is to combine the same kind of classifiers trained on different feature sets; the second is to combine the different classifiers trained on a common feature set. Research [4] has indicated that, for one-class classification, combining different feature sets is more useful than combining different classifiers.

In this application a banknote is characterized by the output of a set of narrow-band sensors which provide responses to light reflected from across the surface of the note. In all a six-dimensional feature set is derived from the responses from a fixed range of spectral bands. The six-dimensional response features are then averaged over the whole note (specifically, the effective note region captured by the sensors, e.g. such a region for the BOS 10 Pound note is shown in Fig. 1) and so each note is characterized by a six-dimensional feature vector. Clearly this level of global averaging will loose some detailed information regarding localized regions that may be very important for differentiating genuine notes from some high quality counterfeits. Based on this consideration, we segment the whole note into non-overlapping and equal-sized subregions, and devise individual one-class classifiers (D^2 and Mixture of Gaussians) on the feature set extracted from each region. For example, the 2x2 segmentation of a BOS 10 Pound note and the empirical feature distributions of one feature are shown in Fig. 2.

Segmenting the whole note into $R \times C$ subregions, which can be numbered from 1 to L, left-to-right and then top-to-bottom wise (similar to that in Fig. 2), we denote the classifier trained on the ith subregion as D_i ($i = 1, \ldots, L$), where

Fig. 1. Effective region of BOS 10 Pound note

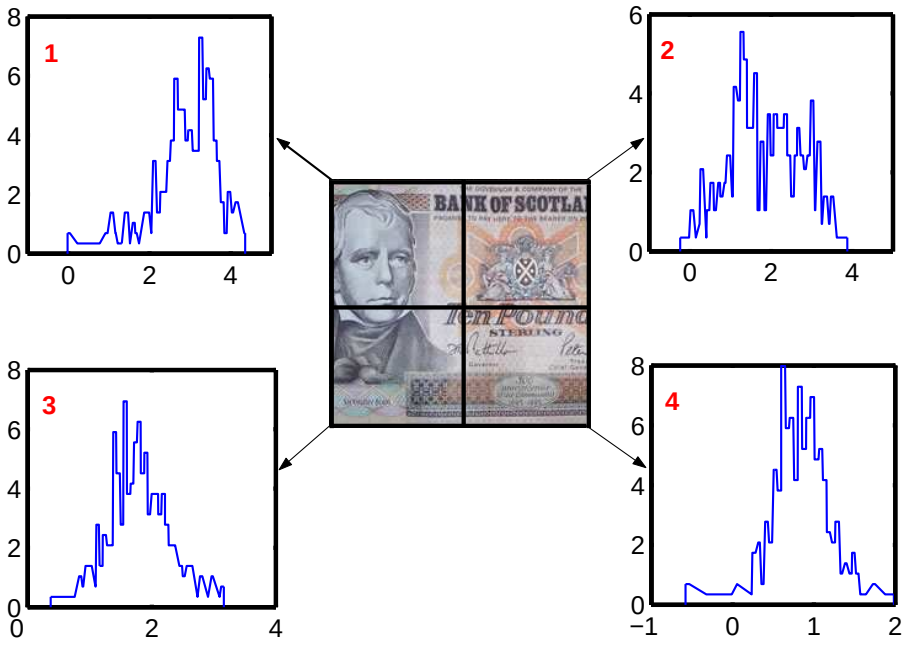

Fig. 2. 2x2 segmentation of BOS 10 Pound note and their feature distributions

R and C are respectively the number of segmented rows and columns, and $L = RC$ is the total number of segmented subregions.

Given R, C and reference (training) samples with size N, their feature matrices $\mathbf{X}_k = \left(\mathbf{x}_i^k\right),\ (k = 1, \ldots, N; i = 1, \ldots, L)$ can be extracted beforehand, where $\mathbf{x}_i^k = \left[x_1^k, x_2^k, \ldots, x_p^k\right]^{\mathrm{T}}$ is a p-dimensional feature vector of the kth sample in the ith subregion. When a test sample $\mathbf{t}$ is presented, according to the same

rule, its feature matrix $\mathbf{T} = (\mathbf{t}_i),\ (i = 1, \ldots, L)$ can be extracted. Then, using the ith one-class classifier D_i trained by $\{\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^N\}$, we can test $\mathbf{t}_i$, and give the decision $y_i(\mathbf{t})$ whether it is rejected or accepted according to the log-likelihood ratio test. If we define y_i as follows:

$$y_i(\mathbf{t}) = \begin{cases} 0 & \text{if } \mathbf{t}_i \text{ is rejected by } \mathrm{D}_i \\ 1 & \text{if } \mathbf{t}_i \text{ is accepted by } \mathrm{D}_i \end{cases} , \tag{14}$$

the final decision for $\mathbf{t}$ can be calculated by combining the product of individual decisions $y_i(\mathbf{t})$ obtained by the classifiers belonging to the optimized combination set $\mathbf{S}$, i.e.

$$Y(\mathbf{t}) = \prod_{i \in \mathbf{S}} y_i(\mathbf{t}) \ . \tag{15}$$

Then, we reject the note $\mathbf{t}$ as a forgery if $Y(\mathbf{t}) = 0$, or accept it as a genuine note if $Y(\mathbf{t}) = 1$. The product combination decision rule defined in (15) can be seen to be equivalent to a unanimous vote which means that only if a note obtained common recognition of it being genuine by all selected classifiers, would it be accepted, otherwise it would be rejected. Unanimous vote is an extreme situation of commonly-used majority voting when requiring all the voters to agree with consensus. The reason that unanimous vote is adopted here is due to the higher cost assigned to incorrectly accepting forgeries.

Considering combining all the individual classifiers D_i $(i = 1, \ldots, L)$, the final decision $Y_{\mathrm{MV}}(\mathbf{t})$ made by employing majority vote and $Y_{\mathrm{UV}}(\mathbf{t})$ by unanimous vote for a note under test $\mathbf{t}$ can be defined respectively as follows:

$$Y_{\mathrm{MV}}(\mathbf{t}) = \mathtt{INT}\left[\frac{\sum_{i=1}^{L} y_i(\mathbf{t})}{\mathtt{INT}(L/2) + 1}\right] \ , \tag{16}$$

$$Y_{\mathrm{UV}}(\mathbf{t}) = \prod_{i=1}^{L} y_i(\mathbf{t}) \ , \tag{17}$$

where $\texttt{INT}(\cdot)$ means rounding down the element to its nearest integer, and both $Y_{\mathrm{MV}}(\mathbf{t})$ and $Y_{\mathrm{UV}}(\mathbf{t})$ can only have the value of either 1 or 0. By writing Eq. (17) in the same form as (16), we obtain

$$Y_{\mathrm{UV}}(\mathbf{t}) = \texttt{INT}\left[\frac{\sum_{i=1}^{L} y_i(\mathbf{t})}{L}\right] \quad . \tag{18}$$

Given N_f counterfeit samples $\left\{\mathbf{t}_f^{k_1}\right\}$, $(k_1 = 1, \ldots, N_f)$ and N_g genuine samples to test $\left\{\mathbf{t}_g^{k_2}\right\}$, $(k_2 = 1, \ldots, N_g)$, the False Negative (FN: false acceptance rate of forgeries) and False Positive (FP: false rejection rate of genuine notes) can be calculated by

$$\mathrm{FN} = \frac{\sum_{k_1=1}^{N_f} Y(\mathbf{t}_f^{k_1})}{N_f} \quad , \tag{19}$$

$$\mathrm{FP} = \frac{N_g - \sum_{k_2=1}^{N_g} Y(\mathbf{t}_g^{k_2})}{N_g} \quad , \tag{20}$$

where $Y(\cdot)$ can be either $Y_{\mathrm{MV}}(\cdot)$ or $Y_{\mathrm{UV}}(\cdot)$.

From (16) and (18), we see $Y_{\mathrm{UV}}(\cdot)$ has a larger denominator than $Y_{\mathrm{MV}}(\cdot)$, which leads $Y_{\mathrm{UV}}(\cdot)$ to be more likely to become 0 than $Y_{\mathrm{MV}}(\cdot)$. Therefore when respectively substituting $Y_{\mathrm{UV}}(\cdot)$ and $Y_{\mathrm{MV}}(\cdot)$ into (19) and (20), the former (unanimous vote) achieves lower FN than the latter (majority vote), however at the same time unanimous vote also produces higher FP than majority voting. This can be explained intuitively, majority vote looks at the average information rather than the individual characteristics. Therefore, for testing genuine notes, as their features are relatively uniform within the reference samples in all subregions (only one bank authorizes the printing of each note), majority vote can potentially achieve better results. However, for testing counterfeits, very high quality counterfeits have similar feature distributions as genuine notes in a number of subregions and differences exist in a few specific subregions where the genuine features might be too complex to be completely duplicated by the counterfeiting process. So by taking account of the average information,

majority vote may give the final decision according to most of the subregions that might give an incorrect judgement for counterfeits. While unanimous vote does not have the problem of giving wrong decisions towards counterfeits, it will however suffer from falsely rejecting, for example, poor quality genuine notes due to the worn nature of the note in some subregions.

For statistical tests, there is always a trade-off between FN and FP. In this particular application, FN is more important than FP, the proposed log-likelihood ratio tests (D^2 test and Mixture of Gaussians + Bootstrap) allow us to specify the expected FP level of individual classifiers by setting the significance levels of each test. Therefore, the extreme unanimous vote is chosen to balance the overall FN and FP performance after combination. As an illustrative example, by segmenting a whole note into 3×3 regions and combining all 9 classifiers, in testing 1000 genuine notes and 1000 forgeries, majority vote achieved the following results FN=23.90% and FP=0.50%; while unanimous vote achieved values of FN=2.30% and FP=8.20%. It can be seen that in this example the ten-fold decrease in FN is accompanied by an almost sixteen-fold increase in FP. When applying the unanimous vote combination rule, selecting appropriate classifiers to be combined is vital. Not all the classifiers built on subregions are necessary to be combined and indeed doing so may reduce the robustness of the whole classification system as we mentioned above. Therefore, the next problem which needs to be addressed is how to find the optimized note segmentation and classifier combination strategy so that we can achieve the best trade-off between FN and FP performance.

Considering an $R_{\max} \times C_{\max}$ search space where $R_{\max}$ and $C_{\max}$ are preset maximum numbers of rows and columns of the segmentation, the total number

of possible ways to segment and combine the individual regions would be

$$P = \sum_{i=1}^{R_{\max}} \sum_{j=1}^{C_{\max}} \sum_{k=1}^{ij} \begin{pmatrix} k \\ \\ ij \end{pmatrix} , \tag{21}$$

where $\begin{pmatrix} k \\ \\ ij \end{pmatrix}$ is the number of combinations of ij regions taken k at a time. For example, if $R_{\max} = 15$ and $C_{\max} = 15$, $P = 5.3923 \times 10^{67}$. This value will exponentially increase following the changes of bigger $R_{\max}$ and $C_{\max}$. Therefore, it is impossible to carry out exhaustive searching experiments. To achieve a useful result, parameters R, C of the optimized segmentation and $\mathbf{S}$, the optimized classifier combination set, will all be identified by a stochastic search algorithm in this case a genetic algorithm (GA) described in Sect. 3.2.

3.2 Genetic algorithm design

Different currencies have different feature distributions, so it is impossible to define a uniform segmentation and combination that is suitable for all possible currencies. However, inspired by the success of using GA's to build multiple classifier systems in [19], we design an application specific GA in this section. The methodology of note segmentation plus classifier combination presented in Sect. 3.1 can then be conveniently applied to the discrimination of notes from any arbitrary currency.

3.2.1 Chromosome design

In order to optimize the note segmentation as well as the classifier combination at the same time, one chromosome $\mathbf{c}$ is composed of three gene groups $\mathbf{c}_r$, $\mathbf{c}_c$ and $\mathbf{c}_d$ one by one, where $\mathbf{c}_r = [r_1, r_2, \ldots, r_{n_1}]$, $\mathbf{c}_c = [c_1, c_2, \ldots, c_{n_2}]$ and $\mathbf{c}_d = [d_1, d_2, \ldots, d_{n_3}]$ are all binary strings and defined as follows:

The first two gene groups $\mathbf{c}_r$ and $\mathbf{c}_c$ respectively represent the number of rows and columns for a note segmentation. Their relationships with R and C are given by

$$R = \texttt{BinToDec}(\mathbf{c}_r) \ , \tag{22}$$

$$C = \texttt{BinToDec}(\mathbf{c}_c) \ , \tag{23}$$

where $\texttt{BinToDec}(\cdot)$ means transforming a number from a binary expression into decimal expression. Given the number of maximum rows $R_{\max}$ and maximum columns $C_{\max}$ which define the size of the intended segmentation search space, $\mathbf{c}_r$'s length n_1 equals the length of the binary string expression of $R_{\max}$; and $\mathbf{c}_c$'s length n_2 equals the length of the binary string expression of $C_{\max}$. The third gene group $\mathbf{c}_d$ represents the combination of classifiers. Its elements d_i and length n_3 are defined as

$$d_i = \begin{cases} 0 & \text{if } \mathrm{D}_i \notin \mathbf{S} \\ & \quad , \quad (i = 1, 2, \ldots, n_3) \ , \\ 1 & \text{if } \mathrm{D}_i \in \mathbf{S} \end{cases} \tag{24}$$

$$n_3 = R_{\max} C_{\max} \ , \tag{25}$$

where $\mathbf{S}$ has the same definition as in Sect. 3.1, means the optimized classifier combination set. For example, given $R_{\max} = 3$ and $C_{\max} = 3$, then n_1, n_2 and

n_3 would respectively equal to 2, 2 and 9; and a chromosome

$$\mathbf{c} = \left[\mathbf{c}_r \vdots \mathbf{c}_c \vdots \mathbf{c}_d\right] = \left[1\ 1 \vdots 1\ 0 \vdots 1\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\right]$$

means segmenting a note into 3×2 (i.e. $R = 3$, $C = 2$) and combining the classifiers D_1, D_2, D_4 and D_6 respectively built on the 1st, 2nd, 4th and 6th subregions.

3.2.2 Gene operations

Three gene operations are designed for the chromosome in our GA algorithm. The first operation is *gene crossover*. Considering the characteristics of our chromosomes, two crossover points in one chromosome are set to complete the crossover operation with other chromosomes. One is the point between $\mathbf{c}_r$ and $\mathbf{c}_c$; and the other is the point in the middle of $\mathbf{c}_d$. For example, if there are two chromosomes $\mathbf{c}_1$ and $\mathbf{c}_2$, where

$$\mathbf{c}_1 = \left[\mathbf{c}_r^1 \vdots \mathbf{c}_c^1 \vdots d_1^1\ d_2^1\ \ldots\ d_{n_3}^1\right] \ ,$$

$$\mathbf{c}_2 = \left[\mathbf{c}_r^2 \vdots \mathbf{c}_c^2 \vdots d_1^2\ d_2^2\ \ldots\ d_{n_3}^2\right] \ ,$$

after crossover, two new chromosomes $\mathbf{c}_1^{\text{new}}$ and $\mathbf{c}_2^{\text{new}}$ will be produced, where

$$\mathbf{c}_1^{\text{new}} = \left[\mathbf{c}_r^1 \vdots \mathbf{c}_c^2 \vdots d_1^1\ \ldots\ d_{\mathtt{INT}(n_3/2)}^1 \vdots d_{\mathtt{INT}(n_3/2)+1}^2\ \ldots\ d_{n_3}^2\right] \ ,$$

$$\mathbf{c}_2^{\text{new}} = \left[\mathbf{c}_r^2 \vdots \mathbf{c}_c^1 \vdots d_1^2\ \ldots\ d_{\mathtt{INT}(n_3/2)}^2 \vdots d_{\mathtt{INT}(n_3/2)+1}^1\ \ldots\ d_{n_3}^1\right] \ ,$$

and $\mathtt{INT}(\cdot)$ has the same definition as previous, means rounding the element to the nearest integer less than or equal to it.

The second operation is *gene mutation*. We use the standard gene mutation operation here, i.e. every time randomly selecting a gene (bit) to mutate from 1 to 0 or from 0 to 1. The third operation is *validation*, which is specifically

designed for this algorithm. Because once the $\mathbf{c}_r$ and $\mathbf{c}_c$ have changed, only the first RC genes in $\mathbf{c}_d$ would be effective, it is necessary to clean up the ineffective genes in order to avoid them affecting other chromosomes in further operations. Therefore the validation operation is designed to set all the $n_3 - RC$ ineffective genes in $\mathbf{c}_d$ to be zeros.

3.2.3 Fitness function

The fitness function is calculated by balancing the performance of FN and FP in the validation set within the GA. Given N_1 genuine samples $\mathbf{G} = \{\mathbf{g}_{k_1}\},\ (k_1 = 1, \ldots, N_1)$ and N_2 counterfeit samples $\mathbf{F} = \{\mathbf{f}_{k_2}\},\ (k_2 = 1, \ldots, N_2)$ where both $\mathbf{g}_{k_1}$ and $\mathbf{f}_{k_2}$ are p-dimensional column vectors, then the decisions $y_i(\mathbf{g}_{k_1})$ and $y_i(\mathbf{f}_{k_2})$ of the ith classifier D_i towards validation samples $\mathbf{g}_{k_1}$ and $\mathbf{f}_{k_2}$ can be calculated by using (14). Finally, we can define the fitness function as

$$f = \left(\frac{1}{N_1} \sum_{k_1=1}^{N_1} \prod_{\{i:\ \mathrm{D}_i \in \mathbf{S}\}} y_i(\mathbf{g}_{k_1}) \right)^2 + \left(\frac{1}{N_2} \left(N_2 - \sum_{k_2=1}^{N_2} \prod_{\{i:\ \mathrm{D}_i \in \mathbf{S}\}} y_i(\mathbf{f}_{k_2}) \right) \right)^2 , \quad (26)$$

where the first term indicates the True Negative (TN: = 1 - FP) performance and the second term indicates the True Positive (TP: = 1 - FN) performance. A chromosome which has a larger value of fitness function will have better performance when applying the segmentation and combination that this chromosome represents. An alternative approach to evaluate the fitness of chromosomes is to use Receiver Operating Characteristics (ROC) [20]. However, this requires running tests for each individual subregion by setting different significance levels which will lead to far higher computation costs. Furthermore, for our application of currency validation, we hope to achieve the objective of rejecting as many counterfeits as possible but as few as possible genuine

notes, which means the performances of classifiers obtained under the very low significance levels (top partial ROC curve) are more important than overall performance of classifiers obtained under a large range of significance levels (whole ROC curve). Therefore (26) is less costly in application and only needs to run tests by setting a very small significance level e.g. 1%.

The whole GA algorithm can be described as follows:

☐ 1. Initialize a random population of n chromosomes

☐ 2. Perform crossover and mutation operations to create another n offspring

☐ 3. Carry out the validation operation, and calculate the fitness for each chromosome by Eq. (26)

☐ 4. Select n chromosomes with the best fitness among all the parents and offspring as the next generation

☐ 5. If convergent or over the preset maximum iteration steps then stop, otherwise go to step 2

For the procedure of selecting the segmentation and combination of regions described above, we require some validation samples in the opposite class (counterfeits) to evaluate the performance of the selected segmentation and combination. However, compared with the amount of counterfeits required to build a two-class classifier, small amounts of counterfeit validation samples suffice for this validation purpose.

4 Experiments

The first set of experiments assess the performance of a single classifier (one-class) based on a single feature vector defining the whole note and this was

carried out as a baseline. Then, by setting different values of $R_{\max}$ and $C_{\max}$, the proposed GA method was used to select the optimized note segmentation and classifier combination, based on which, additional tests were run to allow comparisons. The feature sets used in all the reported experiments are extracted from one single currency value (BOS 10 Pound Sterling notes).

4.1 Single feature vector experiments

4.1.1 Using one-class classifiers

Using 3324 genuine notes as a reference set, based on a single feature vector defining the whole note, the D^2 and Mixture of Gaussians (MoG) + Bootstrap tests were employed on 798 counterfeit notes by setting different critical values (test significance levels) $\alpha = 0.01$, $\alpha = 0.05$ and $\alpha = 0.10$ to specify the expected FP rate for each test. To obtain the FP rate for genuine notes 10-fold cross-validation was employed. Test results for both tests are listed in Tab. 1. It can be seen that the semi-parametric test (MoG) for all significance levels provides significantly lower levels of false negatives (counterfeits accepted as genuine) than that obtained by the D^2 tests.

4.1.2 Using two-class discriminators

As a comparison we test simple binary classifiers for this problem, specifically estimating the class conditional densities based on both a Gaussian and Mixture of Gaussian models. In this experiment, 3324 genuine and 798 counterfeit samples are used to train the two-class discriminators. Test results are shown in Tab. 2, where 10-fold cross-validation is used for both classes to calculate

Table 1

One-class D^2 test and MoG + Bootstrap test by using the whole note

	D^2 Test		MoG + Bootstrap	
α	FN	FP	FN	FP
0.01	26.94 %	0.75 %	18.17 %	0.99 %
0.05	16.42 %	4.45 %	8.77 %	5.14 %
0.10	11.15 %	9.95 %	5.76 %	10.10 %

the values of FN and FP.

Table 2

Two-class discriminators applied to single vector descriptions of a whole note

	Two-class Gaussian Discriminator	Two-class MoG Discriminator
FN	11.90 %	10.03 %
FP	3.97 %	3.25 %

It is clear that employing simple binary classifiers can provide an improvement of around 4% in FN rates over the D^2 test, whilst the difference in performance for the MoG + Bootstrap tests is less pronounced. We observe that employing the one-class classifiers it is possible to reduce the costly FN rate by increasing the significance level of the test and so increase the number of genuine notes rejected (FP). Clearly in cases where large numbers of examples of counterfeits are available then employing binary classifiers will be advantageous, however for the purposes of this work we focus on the case where the numbers of examples from both classes is highly imbalanced such that the distribution of only one class can be reliably estimated.

4.2 Note segmentation and classifier combination experiments

4.2.1 Experiments based on the D^2 test

Setting ($R_{\max}$, $C_{\max}$) to be the different pairs [4]: $(3,3)$, $(7,7)$ and $(15,15)$, then identifying the corresponding optimized note segmentation and classifier combinations, individual D^2 tests at 1% significance levels are carried out. In the experiments the GA used 50 chromosomes in each population. A sample of 2500 genuine notes was used as the reference set for the D^2 test. The GA was trained using 10-fold cross-validation on the 2500 genuine samples as well as 300 counterfeit examples. Finally, tests were performed on an independent test set of 824 genuine notes and 498 counterfeits. The reported test results, shown in Tab. 3, are the best selected from 200 repeated runs of the GA. We will discuss the variability in the solutions obtained in a further section.

Table 3

Test results on the optimized note segmentation (Opt_seg) and classifier combination (Opt_comb) using local D^2 tests at the 1% significance level.

α	$R_{\max}$	$C_{\max}$	Opt_seg	Opt_comb	GA training set		Independent test set	
					FN	FP	FN	FP
0.01	3	3	3×3	D_4, D_6	8.67 %	1.68 %	5.02 %	1.46 %
0.01	7	7	1×6	D_2, D_5, D_6	5.67 %	2.88 %	**2.41 %**	**3.03 %**
0.01	15	15	1×11	D_2, D_8, D_{11}	**4.00 %**	**3.24 %**	4.02 %	3.16 %

The optimized note segmentation and classifier combination strategies ob-

[4] $R_{\max}$ and $C_{\max}$ do not necessarily require to be equal. They can be any positive integer other than 0.

tained by running the GA using the D^2 test are visualized in Fig. 3 - 5.

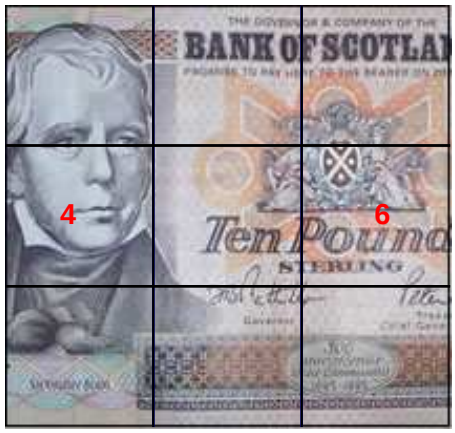

Fig. 3. Optimized note segmentation and classifier combination for D^2 test (search space: $(3, 3)$)

Fig. 4. Optimized note segmentation and classifier combination for D^2 test (search space: $(7, 7)$)

In comparison to the best results obtained by considering the note as a whole we can observe that the segmentation and combination strategy has reduced the number of counterfeits wrongly accepted (FN) from 5.76% to 2.41% whilst also significantly reducing the number of genuine notes being erroneously rejected (FP) from 10.1% to 3.03%. It is also interesting to observe the nature of the segmentations identified and the regions selected for combination. We can observe that regions of both high contrast with little detail and regions

Fig. 5. Optimized note segmentation and classifier combination for D^2 test (search space: $(15, 15)$)

with high levels of detail are combined. Intuitively we can argue that the high contrast regions help to reduce the number of genuine notes rejected whilst the highly detailed regions assist in reducing the number of counterfeits wrongly accepted - as these are the regions which are typically more difficult to copy successfully. We now consider restricting the assumptions made on the distribution of the features in each region and employ a mixture of Gaussians to estimate the density and the Bootstrap (MoG + Bootstrap) to obtain the empirical distribution of the null hypothesis.

4.2.2 Experiments based on the Mixture of Gaussians test

We now repeat the test described in Sec. 4.2.1, this time using the MoG + Bootstrap test. Test results (the best selected from 200 repeated runs) are shown in Tab. 4. The optimized note segmentation and classifier combination strategies obtained by running the GA for MoG + Bootstrap are visualized in Fig. 6 - 8. In Tab. 4, the best performance for the training set and independent test set are indicated by bold fonts. When segmenting the note into 4×11 regions, the best performance for the training samples is achieved, whilst the

best performance for the independent test set is achieved by segmenting the note into 7×3 regions. It indicates that a more complex segmentation will not necessarily achieve better performance on an independent test set and so a level of 'overfitting' can be observed. In comparison to the best D^2 test results obtained the best MoG + Bootstrap provides a lower FP value (number of genuine notes rejected) with a marginal increase in the number of FN's. The decrease in FP value is due to the reduced number of regions being tested (two) and so only two tests at the 1% significance level are being performed. In contrast three regions are combined in the D^2 test and so the expected FP rate will be approximately $3\times$ the individual significance levels. Considering the segmentations and combinations again we can observe that similar regions of low detail and high detail content have been selected as in the previous case.

Table 4

Test results on the optimized note segmentation (Opt_seg) and classifier combination (Opt_comb) using local MoG + Bootstrap tests at the 1% significance level.)

α	$R_{\max}$	$C_{\max}$	Opt_seg	Opt_comb	GA training set		Extra testing set	
					FN	FP	FN	FP
0.01	3	3	2×3	D_3, D_4, D_6	5.67 %	2.96 %	4.82 %	5.22 %
0.01	7	7	7×3	D_{12}, D_{13}	3.67 %	1.88 %	**2.61 %**	**1.46 %**
0.01	15	15	4×11	D_{18}, D_{22}, D_{24}	**0.67 %**	**2.64 %**	2.81 %	2.79 %

Fig. 6. Optimized note segmentation and classifier combination for MoG + Bootstrap test(search space: $(3, 3)$)

Fig. 7. Optimized note segmentation and classifier combination for MoG + Bootstrap test (search space: $(7, 7)$)

4.2.3 The variability of solutions obtained by the GA based search method

As the GA is a stochastic optimization algorithm, it will yield different results in different randomly initialized runs. We can repeatedly run the GA N_r times and select the note segmentation and classifier combination strategy that achieves the best performance, as has been done in the reported experiments. To investigate the level of solution variability, in the following experiments, GA using MoG + Bootstrap is repeatedly run $N_r = 200$ times and the results are recorded as follows.

Fig. 8. Optimised note segmentation and classifier combination for MoG + Bootstrap test (search space: $(15, 15)$)

□1. Search space $(3, 3)$

Only two different results (segmentation and combination strategies) are obtained during 200 randomly initialized runs, which are shown in Tab. 5.

Table 5

GA variability using MoG + Bootstrap (Search space: $(3, 3)$). The 2×3 solution was achieved 188 times and the 1×3 solution was obtained twelve times out of the 200 runs.

No. of runs	Opt_seg	Opt_comb	GA training set		Test set	
			FN	FP	FN	FP
188	2×3	D_3, D_4, D_6	5.67 %	2.96 %	4.82 %	5.22 %
12	1×3	D_1, D_3	8.67 %	1.92 %	5.62 %	1.21 %

□2. Search space $(7, 7)$

Thirty-four different segmentation and combination strategies are obtained from the 200 runs. Among them, the best strategy is a 7×3 segmentation of the note and then combining classifiers D_{12} and D_{13}, which achieves a training

performance: FN = 3.67 %, FP = 1.88 % and the testing performance: FN = 2.61 %, FP = 1.46 %. The interquartile range (IQR) of values of the training FN, FP and testing FN, FP for those 34 strategies over all runs are respectively: 1.67%, 1.04% and 1.61%, 3.03%. IQR computes the difference between the 75th and the 25th percentiles of the sample. It is a robust estimate of the spread of the data, since changes in the upper and lower 25 % of the data do not affect it. On observation the distribution of FN and FP is not normal, IQR is more representative than the standard deviation as an estimate of the spread of the body of the data.

□2. Search space $(15, 15)$ One hundred twenty-eight different strategies are obtained throughout the 200 runs. Among them, the best strategy is a 4×11 segmentation followed by the combination of classifiers D_{18}, D_{22} and D_{24}, which achieves training performance: FN = 0.67 %, FP = 2.64 % and testing performance: FN = 2.81 %, FP = 2.79 %. The IQR of training FN, FP and testing FN, FP of those 128 strategies over all runs are respectively: 3.50%, 1.10% and 1.61%, 1.70%.

5 Conclusions

A step towards solving the automated currency validation problem has been made in this paper. A solution has been presented which is based on the segmentation of the whole note into different subregions followed by the combination of the individual classifiers built on those regions. It has the advantage of being able to be applied to notes of any currency when there is insufficient numbers of counterfeit examples to build a robust binary classifier. In this paper statistical one-class classifiers are employed and a genetic

algorithm is used in selecting the optimized note segmentation and classifier combinations. Experiments were carried out to assess the effectiveness of the proposed method and the results show that, compared with considering the note as a whole, dramatically improved discrimination performance based on false negative and false positive levels is achieved after employing the GA optimized note segmentation and classifier combination. Further investigation will consider combining different individual (one-class and binary) classifiers according to the different feature distributions of the subregions and possibly segmenting notes based on morphological characteristics. The adaptive updating of the segmentation and combination of tests as and when examples of counterfeits become available is also the subject of further investigation.

References

[1] Tax, D.M.J., Duin, R.P.W.: Support vector domain description. Pattern Recognition Letters. **20**(11-13) (1999) 1191–1199.

[2] Tax, D.M.J.: One-class classification. Technische Universiteit Delft Ph.D. Thesis, 2001.

[3] Hayton,P., Schölkopf, B., Tarassenko, L., Anuzis, P.: Support Vector Novelty Detection Applied to Jet Engine Vibration Spectra. Advances in Neural Information Processing Systems 13, eds Leen, Todd K. and Dietterich, Thomas G. and Tresp, Volker, MIT Press, (2001) 946 - 952.

[4] Tax, D.M.J., Duin, R.P.W.: Combining one-class classifiers. In: Kittler, J., Roli, F. (eds.): Multiple Classifier Systems, Proceedings Second International Workshop MCS 2001 (Cambridge, UK, July), Lecture Notes in Computer Science, Vol. 2096, Springer Verlag, Berlin, (2001) 299–308.

[5] Florian, R., Yarowsky, D.: Modeling consensus: classifier combination for word sense disambiguation. Proceedings of EMNLP'02, Philadelphia, PA, USA, (2002) 25–32.

[6] Xu, L., Krzyzak, A., Suen, C.Y.: Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Transactions on Systems, Man, and Cybernetics. **22**(3) (1992) 418–435.

[7] Zhilkin, P.A., Somorjai, R.L.: Application of several methods of classification fusion to magnetic resonance spectra. Connection Science. **8**(3,4) (1996) 427–442.

[8] Mitchell, M.: Introduction to genetic algorithms. MIT Press, Cambridge, MA, USA, 1996.

[9] Morrison, D.F.: Multivariate Statistical Methods (third edition). McGraw-Hill Publishing Company, New York, 1990.

[10] Wang, S., Woodward, W.A., Gary, H.L. et al: A new test for outlier detetion from a multivariate mixture distribution. Journal of Computational and Graphical Statistics. **6**(3) (1997) 285–299.

[11] Roberts, S.J.: Novelty detection using extreme value statistics. IEE Proceedings on Vision, Image & Signal Processing. **146**(3) (1999) 124–129.

[12] Efron, B., Tibshirani, R.J.: An introduction to the bootstrap. Chapman & Hall/CRC Press LLC, Boca Raton, Florida, 1998.

[13] Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, New York, 1995.

[14] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification (second edition). John Wiley & Sons, INC., New York, 2001.

[15] Judge, G.G., Hill, R.C., Griffiths, W.E. et al: Introduction to the theory and practice of econometrics (2nd edition). Wiley, New York, 1998.

[16] Srivastava, M.S., Carter, E.M.: An introduction to applied multivariate statistics. Elsevier Science Publishing Co.,Inc., New York, 1983.

[17] Buxton, B.F., Langdon, W.B. and Barrett, S.J.: Data fusion by intelligent classifier combination. Measurement and Control. **34**(8) (2001) 229–234.

[18] Kuncheva, L.I., Whitaker, C.J.: Feature subsets for classifier combination: an enumerative experiment. In Kittler, J., Roli, F. (eds.): Multiple Classifier Systems, Proceedings Second International Workshop MCS 2001 (Cambridge, UK, July), Lecture Notes in Computer Science, Vol. 2096, Springer Verlag, Berlin, (2001) 229–236.

[19] Ruta, D., Gabrys, B.: Application of the evolutionary algorithms for classifier selection in multiple classifier systems with majority voting. Proceedings of the MCS'2001 Workshop, Cambridge, UK, ISBN 3-540-42284-6 (2001) 399–408.

[20] Langdon, W.B., Buxton, B.F.: Genetic programming for combining classifiers. In Spector, L., Goodman, E.D., Wu, A. et al (eds.): Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001), San Francisco, USA, Morgan Kaufmann, 2001.

[21] Guermeur, Y.: Combining discriminant models with new multi-class SVMs. Pattern Analysis and Applications. **5** (2002) 168–179.

A Derivation of D^2 statistic

Consider the generalised likelihood ratio when the null hypothesis is that N points plus one additional $N+1$'th point are distributed commonly as $\mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_o, \mathbf{C}_o)$ and the alternate hypothesis is given by $\mathbf{x}_{N+1} \sim \mathcal{U}(\bar{\mathcal{C}})$. Then

$$\lambda = \frac{\sup_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}_0(\boldsymbol{\theta})}{\sup_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}_1(\boldsymbol{\theta})} = \frac{\sup_{\boldsymbol{\mu}_o, \mathbf{C}_o} \prod_{n=1}^{N+1} \mathcal{N}_{\mathbf{x}_n}(\boldsymbol{\mu}_o, \mathbf{C}_o)}{\sup_{\boldsymbol{\mu}_1, \mathbf{C}_1} \prod_{n=1}^{N} \mathcal{N}_{\mathbf{x}_n}(\boldsymbol{\mu}_1, \mathbf{C}_1)} = (2\pi e)^{-\frac{p}{2}} \frac{|\hat{\mathbf{C}}_1|^{\frac{N}{2}}}{|\hat{\mathbf{C}}_o|^{\frac{N+1}{2}}}$$

$$= (2\pi e)^{-\frac{p}{2}} |\hat{\mathbf{C}}_1|^{-\frac{1}{2}} \left\{ \frac{N}{N+1} + \frac{N}{(N+1)^2} (\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}_1})^{\mathrm{T}} \hat{\mathbf{C}}_1^{-1} (\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}_1}) \right\}^{-\frac{N+1}{2}}$$

$$= (2\pi e)^{-\frac{p}{2}} |\hat{\mathbf{C}}_1|^{-\frac{1}{2}} \left\{ \frac{N}{N+1} + D_1^2 \right\}^{-\frac{N+1}{2}}$$

Where $D_1^2 = \frac{N^2}{(N+1)^2} (\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}_1})^{\mathrm{T}} \hat{\mathbf{C}}_1^{-1} (\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}_1})$ and so

$$D^2 = (\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}_o})^{\mathrm{T}} \hat{\mathbf{C}}_o^{-1} (\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}_o}) = \frac{(N+1) D_1^2}{N} \left\{ 1 - \frac{D_1^2}{D_1^2 + N + 1} \right\}$$

This is a monotonically increasing function of D_1^2 and so we can use D^2 as the test statistic. This is convenient as it is known that D^2 is distributed as a central F-distribution.