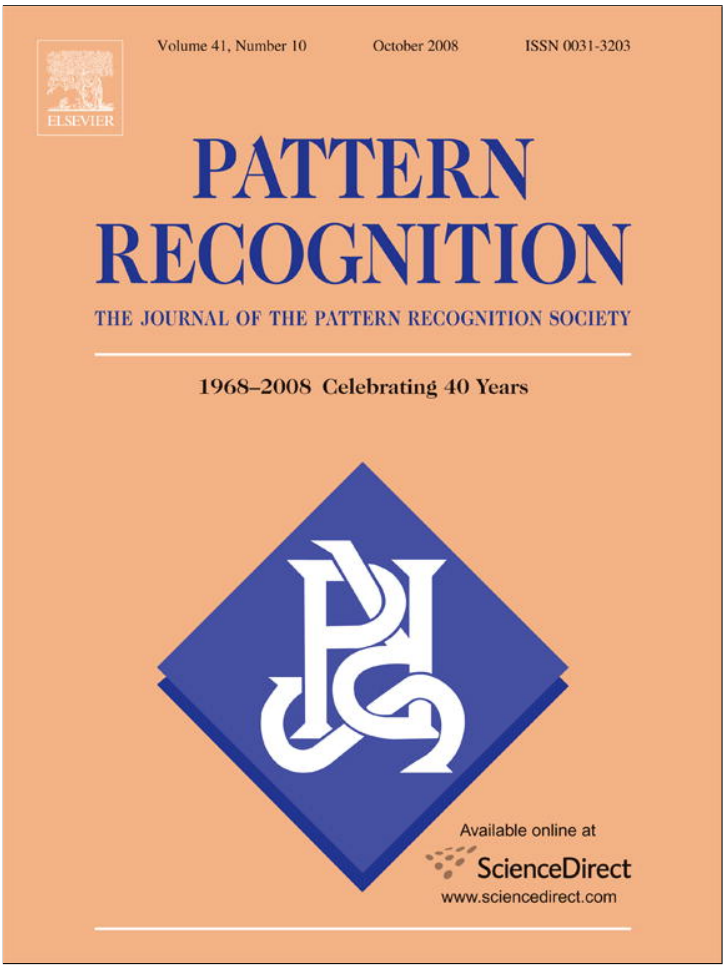

Pattern Recognition 41 (2008) 3010–3020

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A case-study on naïve labelling for the nearest mean and the linear discriminant classifiers

L.I. Kuncheva[a,*], C.J. Whitaker[b], A. Narasimhamurthy[c]

[a]School of Computer Science, Bangor University, Bangor LL57 1UT, UK
[b]School of Psychology, Bangor University 1UT, UK
[c]School of Computer Science and Informatics, University College Dublin (UCD), Dublin, Ireland

ARTICLE INFO

Article history:
Received 2 August 2007
Received in revised form 28 November 2007
Accepted 25 March 2008

Keywords:
Semi-supervised learning
Unlabelled data
On-line classifiers
Naïve labelling

ABSTRACT

The abundance of unlabelled data alongside limited labelled data has provoked significant interest in semi-supervised learning methods. "Naïve labelling" refers to the following simple strategy for using unlabelled data in on-line classification. A new data point is first labelled by the current classifier and then added to the training set together with the assigned label. The classifier is updated before seeing the subsequent data point. Although the danger of a run-away classifier is obvious, versions of naïve labelling pervade in on-line adaptive learning. We study the asymptotic behaviour of naïve labelling in the case of two Gaussian classes and one variable. The analysis shows that if the classifier model assumes correctly the underlying distribution of the problem, naïve labelling will drive the parameters of the classifier towards their optimal values. However, if the model is not guessed correctly, the benefits are outweighed by the instability of the labelling strategy (run-away behaviour of the classifier). The results are based on exact calculations of the point of convergence, simulations, and experiments with 25 real data sets. The findings in our study are consistent with concerns about general use of unlabelled data, flagged up in the recent literature.

1. Introduction

Semi-supervised learning is becoming a centre-stage research theme to answer the challenges of real-life problems [1–3]. The labelled data for training a classifier are usually a small proportion of the total available data. This happens in situations where labelling the objects is time-consuming, expensive, dangerous or destructive. For example, in order to verify a scrapie diagnosis of a sheep, the animal has to be slaughtered and samples of its brain tissue have to be taken for analysis. There are many ways to incorporate unlabelled data in the process of training a classifier.

- *Active learning* seeks to select objects within the unlabelled data whose labelling would lead to the greatest improvement of the chosen classification model or the greatest insight about the problem. These objects are then labelled and used to update the classifier and guide the next selection of candidates for labelling.
- *Transductive learning* assumes that all objects of interest are already collected, and the task is to assign the most plausible labels to the currently presented unlabelled data. There is no concern about a procedure for labelling new data that might become available later.
- *Training generative classifiers* relies on guessing the probabilistic structure of the problem and estimating the parameters of the distributions using both labelled and unlabelled data. Usually expectation maximisation (EM) algorithms are applied for this task.
- *Adaptive learning* constantly modifies the classifier when new data becomes available [4]. On-line EM algorithms have been designed which take a batch of data and re-estimate the parameters of the distributions before proceeding with the next batch [5,6]. An example of this category is the co-training algorithm of Blum and Mitchell [7]. No assumptions are made with respect to the probability distributions. First, two classifiers are trained on different *aspects* of the data (e.g., different subsets of features) using the labelled data only. Then the unlabelled data are run through both classifiers. The most accurately labelled objects by classifier 1 are added to the training set of classifier 2 and vice versa. The classifiers are re-trained using the new respective training sets and the process goes on until some heuristic convergence condition is met. To run this algorithm on-line, batches of data are collected during the on-line operation and the classifier pauses to re-train

* Corresponding author. Tel.: +44 1248383661.
E-mail addresses: l.i.kuncheva@bangor.ac.uk (L.I. Kuncheva), c.j.whitaker@bangor.ac.uk (C.J. Whitaker), anand_mn@yahoo.com (A. Narasimhamurthy).

0031-3203/$30.00 © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.03.028

after each batch. This training methodology is particularly useful when the underlying distribution changes. For example a speech recognition classifier needs to be re-trained for a different speaker.

Within the general enthusiasm and success reports with semi-supervised learning, researchers voice their concerns that the classifiers might deteriorate rather than improve with unlabelled data [3,8–11]. The main result by Cozman et al. [8–10] demonstrates theoretically that if the model is guessed correctly, unlabelled data are expected to improve on the error. This is in unison with the earliest works where normally distributed classes were considered and the batch of unlabelled data was used to evaluate the distribution parameters [12–15]. However, if there is a modelling error (incorrect guess about the shape of the data distribution), using unlabelled data may do more harm than good. It is frustrating that a small modelling error may lead to inadequate results even using a perfect training algorithm, guaranteed to converge to the optimal solution for the assumed classifier.

In this study we relax the training requirements and adopt naïve labelling (NL) to update the classifier. While maximum likelihood estimators (EM being the main representative) work on the batch data, NL works on-line. It takes each new point, labels it, adds it to the training data and updates the classifier. Therefore the multiple passes through the data, which EM relies heavily upon [16], are disallowed for NL. This prevents NL from having the neat convergence properties and asymptotical optimality of EM.

Cozman's results reveal that even the optimal maximum likelihood approximation of the class-conditional densities may be flawed when the model is not guessed correctly. In this context, NL, which makes a single pass through the data and does not optimise the approximations in any way, looks like a lost cause from the start. Logically, if an imperfect approximation method is applied to a correctly guessed model, the benefits from using unlabelled data may disappear. Even worse, when the model is incorrectly guessed, adding imprecise approximation is unlikely to lead to good results. Therefore, any encouraging result with NL would be a bonus. There is anecdotal empirical evidence that NL may improve the classifier [15]. We are interested in NL because it can be viewed as the basic stepping stone for the case where semi-supervised learning is applied for streaming data and non-static environments.

In this paper we look for insights into the asymptotic behaviour of NL for the nearest mean classifier (NMC) and the linear discriminant classifier (LDC). As in the previous studies [12–14] we consider the case of two Gaussian classes. In our scenario the classifier is a true on-line system where each data point is seen once and is then "forgotten". We also investigate the case where the classifier assumptions do not match the true data distribution. To the best of our knowledge, asymptotic results have not been derived for this case. Assuming that we have access to a very small labelled training set and practically unlimited unlabelled data, the question is whether to use NL to update the classifier or stay with the initial classifier.

The rest of the paper is organised as follows. Section 2 explains NL. Section 3 discusses the special case of two Gaussians and one variable. Simulation results are reported in Section 4, and results with real data in Section 5. Section 6 concludes the paper.

2. NL—the general scenario

We consider the general pattern recognition problem where the data come from a mixture of c distributions corresponding to c mutually exclusive classes $\Omega = \{\omega_1, \ldots, \omega_c\}$. Denote by $\mathbf{x}$ a data point in the feature space of the problem. Without loss of generality we may assume that the feature space is the n-dimensional real space $\Re^n$. Let $p(\mathbf{x}|\omega_i)$ be the class-conditional probability density function for class ω_i and $P(\omega_i)$ be the prior probability for this class. Regardless of what the true distributions are, suppose that a parametric classifier $C(\vec{\theta})$ has been chosen and trained on a labelled data set X_l sampled from the distribution of the problem. We denote by $\vec{\theta} \in \Theta$ the vector of parameters that specify the classifier completely. Applying the classifier to the feature space is equivalent to partitioning the
feature space into c classification regions $\mathscr{R}_1(C(\vec{\theta})), \ldots \mathscr{R}_c(C(\vec{\theta}))$. The class label of a point $\mathbf{x}$ is determined by the region it belongs to. The classification error of $C(\vec{\theta})$ is

$$E(C(\vec{\theta})) = 1 - \sum_{i=1}^{c} \int_{\mathscr{R}_i(C(\vec{\theta}))} P(\omega_i) p(\mathbf{x}|\omega_i)\, \mathrm{d}\mathbf{x}. \tag{1}$$

The error depends on $\vec{\theta}$ through the definition of the classification regions by the respective classifier.

Consider the following scenario. Classifier $C(\vec{\theta})$ is initialised by using a small labelled training data set sampled from the distribution of the problem. Denote the initial parameter values by $\vec{\theta}_l$. Assume that beyond the initial stage we have access to practically unlimited supply of unlabelled data sampled from the same distribution. The classifier has an adaptation mechanism so that each data point is first labelled by the current classifier, added to the training set and immediately used to re-train the classifier. The restriction that we pose here is one of the hallmarks of on-line classifiers [17,18]: the classifier must see each data point only once. This restriction precludes using algorithms such as EM, which loop through the data until the maximum of the likelihood is achieved with a desirable precision. However, later in the experiment we use a single iteration of EM as a "soft" alternative of NL. The re-training itself consists in updating the parameter vector $\vec{\theta}$. We assume that the re-training does not require storing of all previously seen data but only depends upon the previous parameter values, the number of data points seen thus far, and the new data point $\mathbf{x}$

$$\vec{\theta}(k) = f(\vec{\theta}(k-1), k^{(1)}, k^{(2)}, \ldots, k^{(c)}, \mathbf{x}), \tag{2}$$

where $k^{(i)}$ is the number of points from class ω_i within the k seen points.

For obvious reasons we term this adaptation strategy NL [1]. Assuming the sequence $\vec{\theta}(1), \vec{\theta}(2), \ldots, \vec{\theta}(k), \ldots$ converges ($\vec{\theta}(1) = \vec{\theta}_l$), denote by $\vec{\theta}_u$ the parameter vector it converges to. Also, let $\vec{\theta}^*$ be the optimal set of parameters for this classifier with respect to classification error, defined by

$$\vec{\theta}^* = \arg\max_{\vec{\theta} \in \Theta} \sum_{i=1}^{c} \int_{\mathscr{R}_i(C(\vec{\theta}))} P(\omega_i) p(\mathbf{x}|\omega_i)\, \mathrm{d}\mathbf{x}. \tag{3}$$

Modelling the general case of NL requires specifying the underlying distributions, the classifier and the update rule (2). In this study we investigate the simple case of

- two normal distributions in $\Re$,
- the nearest mean classifier (NMC) and the linear discriminant classifier (LDC) and
- maximum likelihood estimates of $\vec{\theta}$ (Robbins-Monro update formulas for the means and the standard deviations [19]), taking the guessed label of the data point as the correct label.

Cozman and Cohen [8] already showed that using unlabelled data is expected to improve the classifier if the modelling assumptions are correct. Included within the modelling assumption is the procedure of finding the asymptotically optimal value $\vec{\theta}^*$. In our scenario, even with a correct guess of the underlying distribution, the updating process is not guaranteed to lead to $\vec{\theta}^*$. Here we study the convergence of NL, the errors incurred by classifiers $C(\vec{\theta}^*)$, $C(\vec{\theta}_l)$ and $C(\vec{\theta}_u)$, and their relationship to the Bayes error.

3. NL for two Gaussians

Let $x \in \Re$ be the variable of interest, and let $p(x|\omega_1) \sim N(\mu_1, \sigma_1^2)$ and $p(x|\omega_2) \sim N(\mu_2, \sigma_2^2)$ be the class-conditional pdfs. Denote by λ the prior probability for class ω_1, i.e., $P(\omega_1)=\lambda$, $P(\omega_2)=1-\lambda$. Without loss of generality we may assume that $\mu_1 < \mu_2$.

We start with the NMC as the simplest parametric classifier. NMC estimates the means of the classes from the available data and labels an unseen point in the class with the nearest mean. NMC is optimal in Bayesian sense (minimum classification error) if the two classes have the same variance $\sigma = \sigma_1 = \sigma_2$ and the same priors $\lambda = 0.5$. For the case of two classes and $x \in \Re$, NMC requires two parameters, m_1 and m_2 for the respective means. Note that the optimal values of these parameters, m_1^* and m_2^*, found through Eq. (3) are not unique. The unique optimal classification boundary $b^* = (m_1^* + m_2^*)/2$ will be the same for infinitely many transformations of the two means which keep them symmetrical about b^*. Therefore we will consider the boundary b to be the only parameter of NMC, $\theta = \{b\}$.

Knowing the true distribution of the classes, we can construct a classifier in order to derive a possible convergence point $\theta_u = b_u$. Since we assumed that the labelled data set is finite and small, it will merely initialise the boundary by $b = b_l$. The convergence will be evaluated under the assumptions that infinite amount of unlabelled data is available.

Let b be the boundary obtained by updating the NMC at some stage of the training process. NMC will label all points to the left in class 1, and all points to the right in class 2, as illustrated in Fig. 1. The true distributions scaled by the priors $\lambda = 0.4$ and $1-\lambda = 0.6$ (subplot (a)) are $p(x|\omega_1) \sim N(0, 1)$ and $p(x|\omega_2) \sim N(4, 16)$. The boundary b defines a new set of distributions. As all points $x < b$ belong to the new class 1, the pdf for class 1 will be

$$p_1(x) = \begin{cases} \dfrac{1}{Z(b)} p(x) = \dfrac{1}{Z(b)} [\lambda p(x|\omega_1) \\ \quad +(1-\lambda) p(x|\omega_2)], & x < b, \\ 0, & x \geqslant b, \end{cases} \tag{4}$$

where $Z(b)$ is a normalising constant so that the integral of $p_1(x)$ across x is 1,

$$\begin{aligned} Z(b) &= \int_{-\infty}^{b} p(x)\,\mathrm{d}x \\ &= \int_{-\infty}^{b} [\lambda p(x|\omega_1) + (1-\lambda) p(x|\omega_2)]\,\mathrm{d}x. \end{aligned} \tag{5}$$

Accordingly,

$$p_2(x) = \begin{cases} 0, & x < b, \\ \dfrac{1}{1-Z(b)} p(x) = \dfrac{1}{1-Z(b)} [\lambda p(x|\omega_1) \\ \quad +(1-\lambda) p(x|\omega_2)], & x \geqslant b. \end{cases} \tag{6}$$

Fig. 1(b) shows the new distributions. If we sample long enough from the original distribution and let the classifier learn by labelling the data according to b, the two class means will be the means with respect to $p_1(x)$ and $p_2(x)$

$$\begin{aligned} m_1(b) &= \int_{-\infty}^{\infty} x p_1(x)\,\mathrm{d}x = \frac{1}{Z(b)} \int_{-\infty}^{b} x[\lambda p(x|\omega_1) \\ &\quad + (1-\lambda) p(x|\omega_2)]\,\mathrm{d}x, \end{aligned} \tag{7}$$

$$\begin{aligned} m_2(b) &= \int_{-\infty}^{\infty} x p_2(x)\,\mathrm{d}x = \frac{1}{1-Z(b)} \int_{b}^{\infty} x[\lambda p(x|\omega_1) \\ &\quad + (1-\lambda) p(x|\omega_2)]\,\mathrm{d}x. \end{aligned} \tag{8}$$

Having estimated the two means, a new boundary can be calculated. Then a sequence of parameter values can be constructed by taking $b_0 = b_l$, $b_1 = (m_1(b_0) + m_2(b_0))/2$, and so on. The general term for the NMC is

$$b_k = \frac{m_1(b_{k-1}) + m_2(b_{k-1})}{2}. \tag{9}$$

Denote by $f(b)$ the function determining the new boundary, given the current boundary b. If the sequence $b_0, b_1, \ldots, b_k, \ldots$ converges, the stationary point would be the solution of

$$b = f(b). \tag{10}$$

The equation does not have an analytical solution even for the simple case of NMC and two Gaussian distributions because of the scaling constant $Z(b)$ which has to be evaluated through integration. The derivation of $f(b)$ is given in the Appendix. From numerical analysis, the sequence b_k can be regarded as steps in solving $b = f(b)$ using the fixed point method (see, for example, Ref. [20]). The sequence b_k is guaranteed to converge if $f(b)$ is differentiable and, in some vicinity of the solution, $|\mathrm{d}f(b)/\mathrm{d}b| < 1$. It is difficult to analyse $f(b)$ even for the simple case considered here. Hence we provide an illustration of the behaviour of $f(b)$ for three choices of parameters

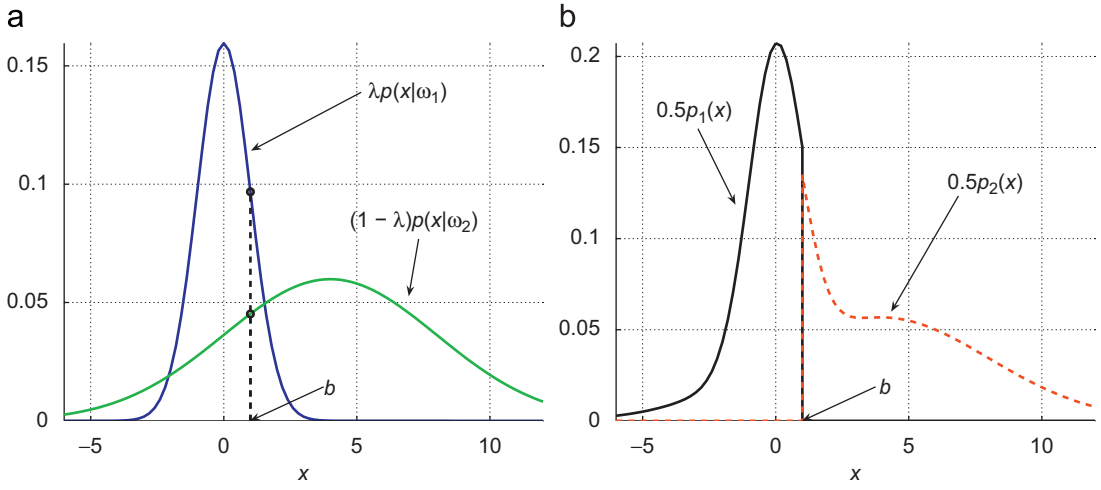

Fig. 1. Illustration of the true distributions ($p(x|\omega_1)$ and $p(x|\omega_2)$) and the corresponding boundary-induced distributions ($p_1(x)$ and $p_2(x)$). (a) True distributions and (b) distributions induced by boundary b.

Table 1
Optimal and asymptotic boundaries and error rates for two Gaussian classes

Distribution	Bayes error	Optimal boundary b^*	Error rate at b^*	NMC		LDC	
				Asymptotic b_u	Error at b_u	Asymptotic b_u	Error at b_u
$p(x\|\omega_1) \sim N(0,1)$ $p(x\|\omega_2) \sim N(4,1)$ $\lambda = 0.5$	0.0228	2.0000	0.0228	2.0000	0.0228	2.0000	0.0228
$p(x\|\omega_1) \sim N(0,1)$ $p(x\|\omega_2) \sim N(4,1)$ $\lambda = 0.1$	0.0122	1.4507	0.0122	2.3313	0.0438	1.4147	0.0122
$p(x\|\omega_1) \sim N(0,1)$ $p(x\|\omega_2) \sim N(4,16)$ $\lambda = 0.5$	0.1400	1.7570	0.1635	3.5246	0.2265	4.7498	0.2872

of the true distributions

- *Equal priors and equal variances*: In this case the optimal boundary b^* (satisfying Eq. (3)) is $b = (\mu_1 + \mu_2)/2$. This is also the Bayes-optimal boundary, and NMC is equivalent to the Bayes-optimal classifier for this problem. We use $p(x|\omega_1) \sim N(0,1)$ and $p(x|\omega_2) \sim N(4,1)$, $\lambda = 0.5$, so the optimal boundary is $b^* = b = 2$.
- *Different priors, equal variances*: For equal variances σ^2 and different priors the optimal boundary is

$$b^* = \frac{\mu_1 + \mu_2}{2} - \frac{\sigma^2}{\mu_1 - \mu_2} \ln \frac{\lambda}{1-\lambda}. \tag{11}$$

  In our examples, $p(x|\omega_1) \sim N(0,1)$ and $p(x|\omega_2) \sim N(4,1)$, $\lambda = 0.1$, so $b^* \approx 1.4507$.
- *Equal priors, different variances*: For non-equal variances, the optimal boundary consists of two points found as the real roots $a_{1,2}$ (if they exist) of the quadratic equation

$$\left(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2}\right) a^2 + 2\left(\frac{\mu_1}{\sigma_1^2} - \frac{\mu_2}{\sigma_2^2}\right) a + 2\ln\frac{\lambda\sigma_2}{(1-\lambda)\sigma_1} - \left(\frac{\mu_1^2}{\sigma_1^2} - \frac{\mu_2^2}{\sigma_2^2}\right) = 0. \tag{12}$$

  As NMC only calculates one boundary, b^* in Eq. (3), NMC can never be Bayes-optimal in this case. It can be proved, however, that a single optimal boundary b^* is one of a_1 or a_2. As there will be two points of intersection of the discriminant curves, one of the classes will have its classification region $\mathcal{R}_i(C(b))$ split into two subregions spanning the distribution tails. The boundary that cuts off the sub-region with the larger area will be b^*. For our example, $p(x|\omega_1) \sim N(0,1)$ and $p(x|\omega_2) \sim N(4,16)$, $\lambda = 0.5$. The two boundaries are $a_1 = -2.2903$ and $a_2 = 1.7570$. The optimal boundary is $b^* = a_2 = 1.7570$. Note that Eq. (12) can be used to derive the optimal boundaries for any priors λ.

Table 1 shows the Bayes error rate, the optimal boundaries b^* and the error rates for the three distributions. We calculated $f(b)$ for values of b varying from -1 to 5 and plotted the graphs in Fig. 2. Plotted is also the diagonal line $f(b) = b$. The intersection of this line with the curve $f(b)$ gives the stationary point b_u. To find out b_u we ran the iterative procedure $b_k = f(b_{k-1})$ from an initial point which ensured convergence, and set as a terminating condition $|b_k - b_{k-1}| < 10^{-6}$. The values of $f(b)$ were calculated by Eq. (32) in the Appendix. The convergence values b_u for NMC are shown in Table 1 along with the incurred error rate. Note that in all three cases we are solving iteratively a theoretical equation that carries no uncertainty related to the choice of training data. The point of convergence can be used to measure the bias of the classifier, i.e., the deviation from the optimal boundary.

Like NMC, the LDC will also determine one boundary between the classes. However, instead of only the means of the distributions induced by the boundary b, LDC also evaluates the class variance, assumed to be the same for both classes, as well as the priors. The new boundary is calculated as

$$b_{\text{new}} = f(b) = \frac{m_1(b) + m_2(b)}{2} - \frac{s(b)^2}{m_1(b) - m_2(b)} \ln \frac{Z(b)}{1 - Z(b)}, \tag{13}$$

where $s(b)^2$ is the common variance found as

$$\begin{aligned} s(b) &= \frac{Z(b)}{Z(b)} \int_{-\infty}^{b} (x - m_1(b))^2 p(x)\,dx + \frac{1 - Z(b)}{1 - Z(b)} \int_{b}^{\infty} (x - m_2(b))^2 p(x)\,dx \\ &= \int_{-\infty}^{b} (x - m_1(b))^2 p(x)\,dx + \int_{b}^{\infty} (x - m_2(b))^2 p(x)\,dx. \end{aligned} \tag{14}$$

The expression for $f(b)$ in this case is even less tractable than the one for NMC. Thus the calculations of $f(b)$ for this case use Eqs. (5), (7), (8) and (14), and plug them into Eq. (13). Using Eq. (13) we found b_u for the three distributions in the above example (results are in Table 1), and also plotted the graphs for LDC in Fig. 2.

Table 1 and Fig. 2 reveal several interesting phenomena. For the simplest case of equal variances and equal priors, NL lands both classifiers on the optimal boundary, $b^* = b_u = 2$. In other words, NL will indeed improve upon an initial boundary. However, while NL with NMC is guaranteed to improve the boundary starting from any initial b_0, NL may diverge for LDC if b_0 is slightly to the left of μ_1 or slightly to the right of μ_2.

For equal variances and non-equal priors, NMC converges to a non-optimal boundary with an error at b_u more than 3.5 times the Bayes error. LDC, if NL converges, is only slightly off the optimal boundary, and the error at b_u is practically the same as the Bayes error.

For the case of unequal variances, LDC is misled by NL in a larger degree than NMC is, and finds a boundary far away from the optimal boundary, with an error around twice the Bayes error. This agrees with the conclusion in Ref. [10] that when the model guess is incorrect, unlabelled data may adversely affect the performance of the classifier. The assumption for LDC is that variances of the classes are the same. NMC has the advantage in this case of assuming additionally (correctly!) that the priors are the same, and does not estimate these from the data. This explains the better solution by NMC.

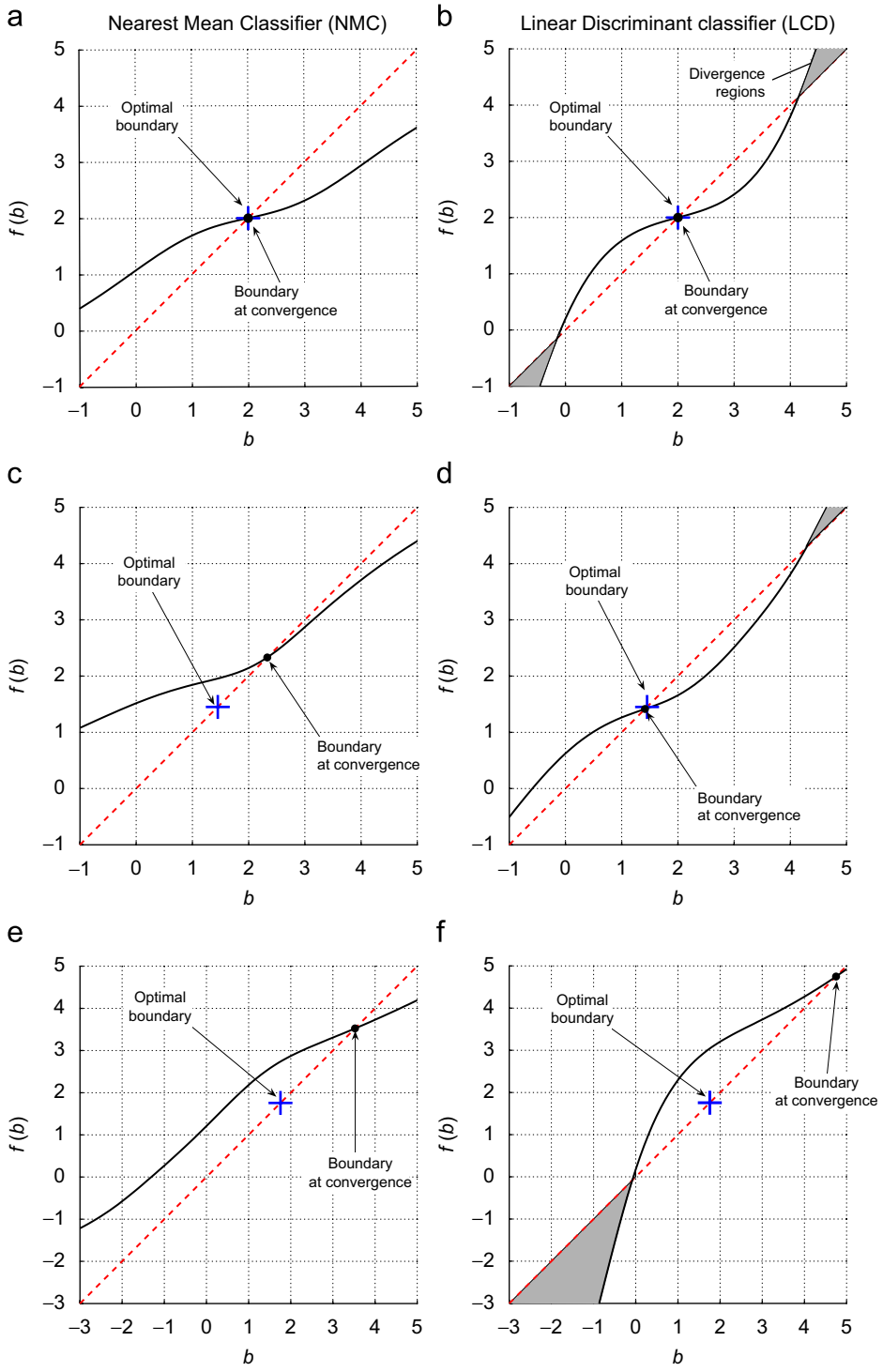

Fig. 2. Illustration of the optimal and the asymptotical boundary points for the linear discriminant classifier (LDC) and the nearest mean classifier (NMC) for two classes with $p(x|\omega_1) \sim N(\mu_1, \sigma_1^2)$, $p(x|\omega_2) \sim N(\mu_2, \sigma_2^2)$ and prior probabilities $P(\omega_1) = \lambda$, $P(\omega_2) = 1 - \lambda$. (a) $N(0, 1)$, $N(4, 1)$, $\lambda = 0.5$, (b) $N(0, 1)$, $N(4, 1)$, $\lambda = 0.5$, (c) $N(0, 1)$, $N(4, 1)$, $\lambda = 0.1$, (d) $N(0, 1)$, $N(4, 1)$, $\lambda = 0.1$, (e) $N(0, 1)$, $N(4, 16)$, $\lambda = 0.5$ and (f) $N(0, 1)$, $N(4, 16)$, $\lambda = 0.5$.

4. Simulations

In this section we put the asymptotic boundaries found above to the test. The question we seek to answer is whether these boundaries are achievable in a more realistic scenario where each new observation is used to re-calculate the current boundary.

In evaluating the asymptotic boundary b_u in Section 3, it was assumed that the parameters of the two distributions $p_1(x)$ and $p_2(x)$ (Fig. 1(b)) are readily available for each b_k in the sequence of boundaries. For this to correspond to a real experiment, we need an infinite sample for a fixed boundary b_k. NL is, in fact, quite far from this scenario because the boundary might change after each new sample. Then the parameters being evaluated are heavily affected by

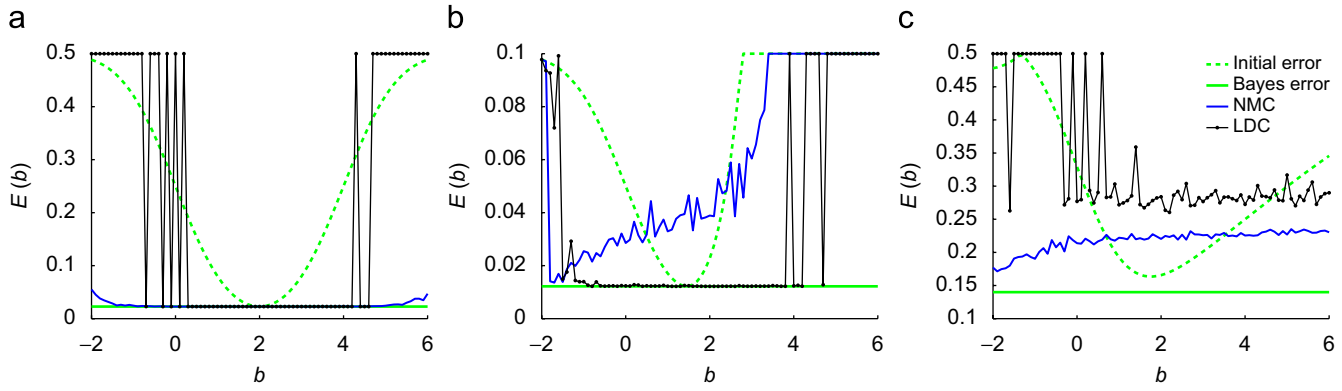

Fig. 3. Classification error $E(b)$ as a function of the boundary b for the simulation experiment with the three distributions ($p(x|\omega_1) \sim N(\mu_1, \sigma_1^2)$, $p(x|\omega_2) \sim N(\mu_2, \sigma_2^2)$ and prior probabilities $P(\omega_1) = \lambda$, $P(\omega_2) = 1 - \lambda$). (a) $N(0, 1)$, $N(4, 1)$, $\lambda = 0.5$, (b) $N(0, 1)$, $N(4, 1)$, $\lambda = 0.1$, (c) $N(0, 1)$, $N(4, 16)$, $\lambda = 0.5$.

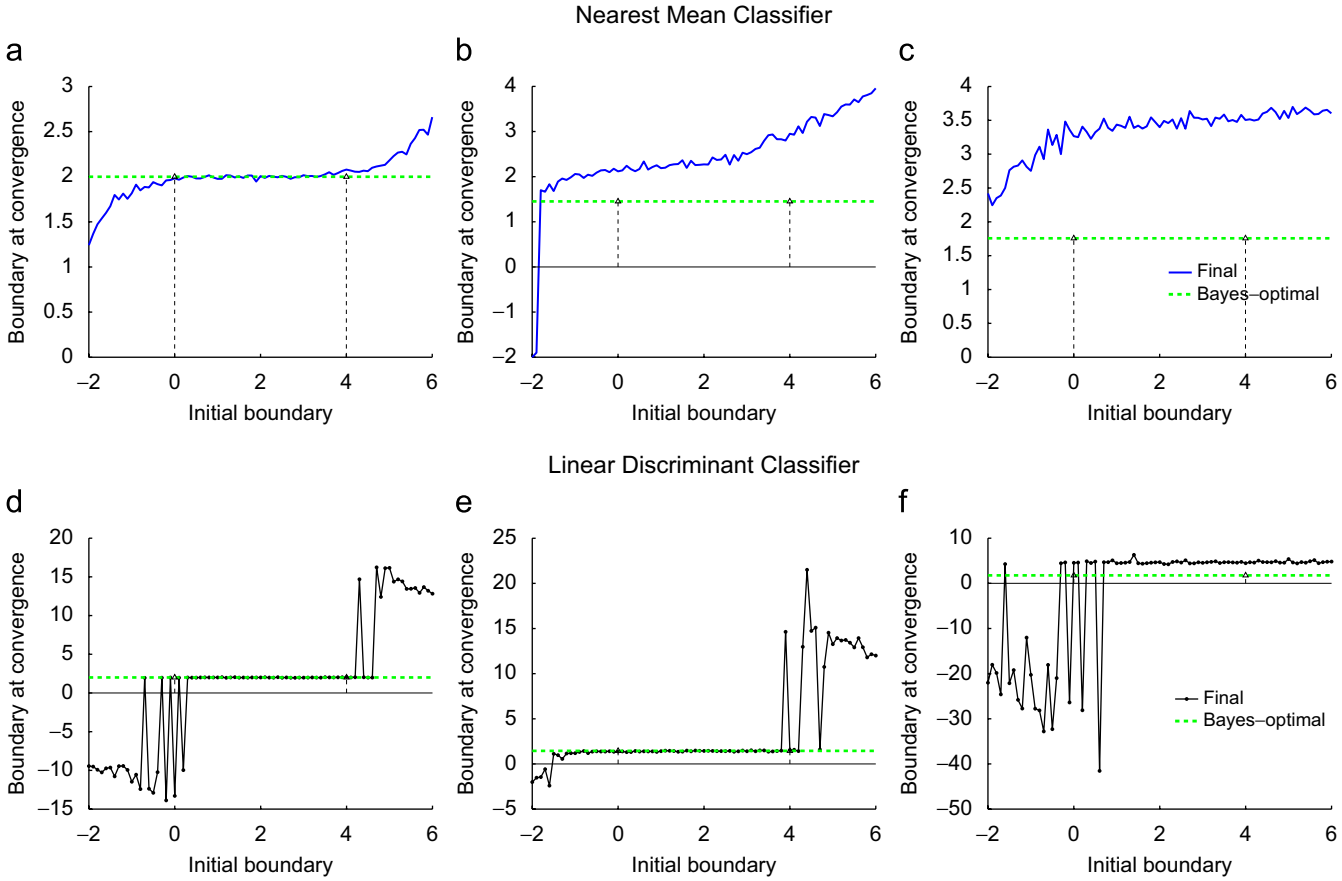

Fig. 4. Initial boundaries and boundaries at convergence for the simulation experiment with the three distributions. The optimal boundary (b^*) and the two means ($\mu_1 = 0, \mu_2 = 4$) are also indicated. (a) $N(0, 1)$, $N(4, 1)$, $\lambda = 0.5$, (b) $N(0, 1)$, $N(4, 1)$, $\lambda = 0.1$, (c) $N(0, 1)$, $N(4, 16)$, $\lambda = 0.5$, (d) $N(0, 1)$, $N(4, 1)$, $\lambda = 0.5$, (e) $N(0, 1)$, $N(4, 1)$, $\lambda = 0.1$, (f) $N(0, 1)$, $N(4, 16)$, $\lambda = 0.5$.

the labelling of all previous data. To demonstrate this effect, we ran simulation experiments with the following protocol.

- The initial boundary b_0 was varied taking all values in the set $\{-2.00, -1.99, -1.98, \ldots, 5.99, 6.00\}$.
- Data samples have been randomly generated from the distribution of the problem.
- To start off the procedure, boundary $b_1 = f(b_0)$ was calculated as soon as the number of objects in the smaller of the two classes reached a pre-specified limit (set to 20 in this experiment).
- The procedure of estimating the parameters of the classifier and subsequently $b_k = f(b_{k-1})$ was run for 5000 iterations. Thus we obtained the convergence boundary $b_u = b_{5000}$.

To answer the main question, whether NL will improve on the initial boundary, Fig. 3 shows the error rate of NL as a function of the *initial* boundary. Overlaid in the figure is the error of the classifier

if the initial boundary was not changed at all (dashed line). When the dashed line is above the solid lines (classifiers' errors) the initial boundary incurs higher classification error.

Fig. 3(a) suggests that in the case of equiprobable classes with equal variances, both classifiers benefit substantially from NL. NMC (the solid line without a marker) is the more stable of the two classifiers, giving error that almost coincides with the Bayes error for any initial boundary. LDC, on the other hand (the solid line with the dot marker), suffers from major instability outside the region from $\mu_1 = 0$ to $\mu_2 = 4$. This could be expected in view of the two regions of divergence in Fig. 1(b). Subplot 3(b) reveals that the success of NMC is short-lived. When the assumptions for optimality of NMC are not met (unequal priors in this case), the classifier behaves inadequately. LDC is again very well behaved between the two means and unstable outside this region. This shows that when the distribution is indeed correctly guessed, a very simple procedure such as NL is expected to improve on the initial boundary. Finally, subplot 3(c) suggests that neither of the two classifiers offers an improvement on the initial boundary, especially if it happens to be near the optimal boundary. The failure of the classifiers can be attributed to the wrong modelling assumptions. NMC is again the more stable of the two classifiers, but is not of much use in terms of error rate.

Fig. 4 displays the boundary at convergence, b_u, as a function of the original boundary b_0. The trends of the graphs with NMC (subplots (a)–(c)) suggest the possibility of a *runaway classifier*, i.e., existence of divergence regions. LDC (subplots (d)–(f)) is relatively stable outside its divergence regions.

The results show that the theoretical convergence properties for the synthetic cases considered here carry forward when the boundary is re-calculated with each new observation. This gives us the reassurance that NL is not likely to show a dramatic change in its behaviour in real-life scenarios. Even though NL has not been proved to be a sound density approximation method, it is capable of leading to the correct solution.

5. Experiments with real data

In this section we examine NL for NMC and LDC with real data that hardly conform to the distribution assumptions which make NMC and LDC Bayes-optimal. Getting sensible error rates is a tall order anyway bearing in mind that NMC and LDC are among the simplest classifiers and are going to be trained on a very small labelled data set. A negative outcome of this experiment would not be a surprise, hence any favourable result would be welcome.

5.1. Data

Twenty five real data sets were used in the experiment (Table 2). All features are numerical and there are no missing values. In the table, the data sets are sorted by the total number of samples N.

5.2. Experimental protocol

Although there is no strict guideline about what a sufficient data size is, the common wisdom (which we quote after Nagy [4]) is that the size of the training data should be around $10 \times n \times c$, where n is the number of features and c is the number of classes in a problem. To simulate a small data set we took the size of the labelled data set to be $1 \times n \times c$. The choices for the experimental protocol with a single data set are listed below:

- 100 runs were carried out with 90% of the data used for training and 10% used for testing. The splits were done using stratified sampling.
- From each training part of the data, a random stratified sample of $N_l = 1 \times n \times c$ was taken as the initial labelled data. An initial classifier (NMC or LDC) was trained on the labelled part. The error of this classifier is called "the initial error".
- The remaining part of the training data was used as the new coming unlabelled data. To simulate an i.i.d. sequence of unlabelled data the data were shuffled before each of the 100 runs.
- One point from the unlabelled data was fed to the system at a time. The point was labelled, added to the training set and the classifier parameters were updated accordingly. The classification error was evaluated on the (labelled) testing set. In this way we created a "progression curve" which is the classification error as a function of the number of unlabelled samples seen by the classifier.
- The results were averaged across the 100 runs giving a single progression curve for the data set.

Table 2
Data sets used in the experiment

Data set	Features	Classes	Objects	Source
iris	4	3	150	UCI[a]
wine	13	3	178	UCI
sonar	60	2	208	UCI
laryngeal1	16	2	213	Collection[b]
glass	9	6	214	UCI
thyroid	5	3	215	UCI
votes	16	2	232	UCI
voice3	10	3	238	Collection
breast	9	2	277	UCI
heart	13	2	303	UCI
liver	6	2	345	UCI
spect	44	2	349	Collection
ionosphere	34	2	351	UCI
laryngeal3	16	3	353	Collection
voice9	10	9	428	Collection
wbc	30	2	569	UCI
palynomorphs	31	3	609	Private[c]
laryngeal2	16	2	692	Collection
pima	8	2	768	UCI
vehicle	18	4	846	UCI
vowel	11	10	990	UCI
german	24	2	1000	UCI
image	19	7	2310	UCI
scrapie	14	2	3113	Private[d]
spam	57	2	4601	UCI

[a]UCI [21] http://www.ics.uci.edu/~mlearn/MLRepository.html.

[b]Collection http://www.informatics.bangor.ac.uk/~kuncheva/activities/real_data_full_set.htm.

[c]Images of pieces of kerogen extracted from microscope images of palynomorphs.

[d]Data on scrapie disease in sheep (related to BSE in cows), provided by DEFRA, UK, http://www.defra.gov.uk/.

5.3. Results

The experiments were run with the 25 data sets and the two classifiers NMC and LDC. The results are displayed in Fig. 5. The x-axis corresponds to the number of processed unlabelled samples and the y-axis is the progression of the classification error, evaluated on the testing sets and averaged across 100 runs. The thin black line shows the error for the NMC and the thick red line shows the error for the LDC.

The graphs are meant to visualise the direction of the curves rather than the details. Several typical patterns can be observed. The error rates have completely opposite trends for data sets iris, votes, breast, heart, laryngeal3, wbc, laryngeal2, pima and spam. While NMC becomes a "run-away classifier" as a consequence of NL, the error of LDC gradually decreases with more unlabelled samples. This pattern, however, is not matched on all data sets. There are sets where the errors are in agreement, both markedly decreasing (german) or increasing (spect, ionosphere, vowel, image).

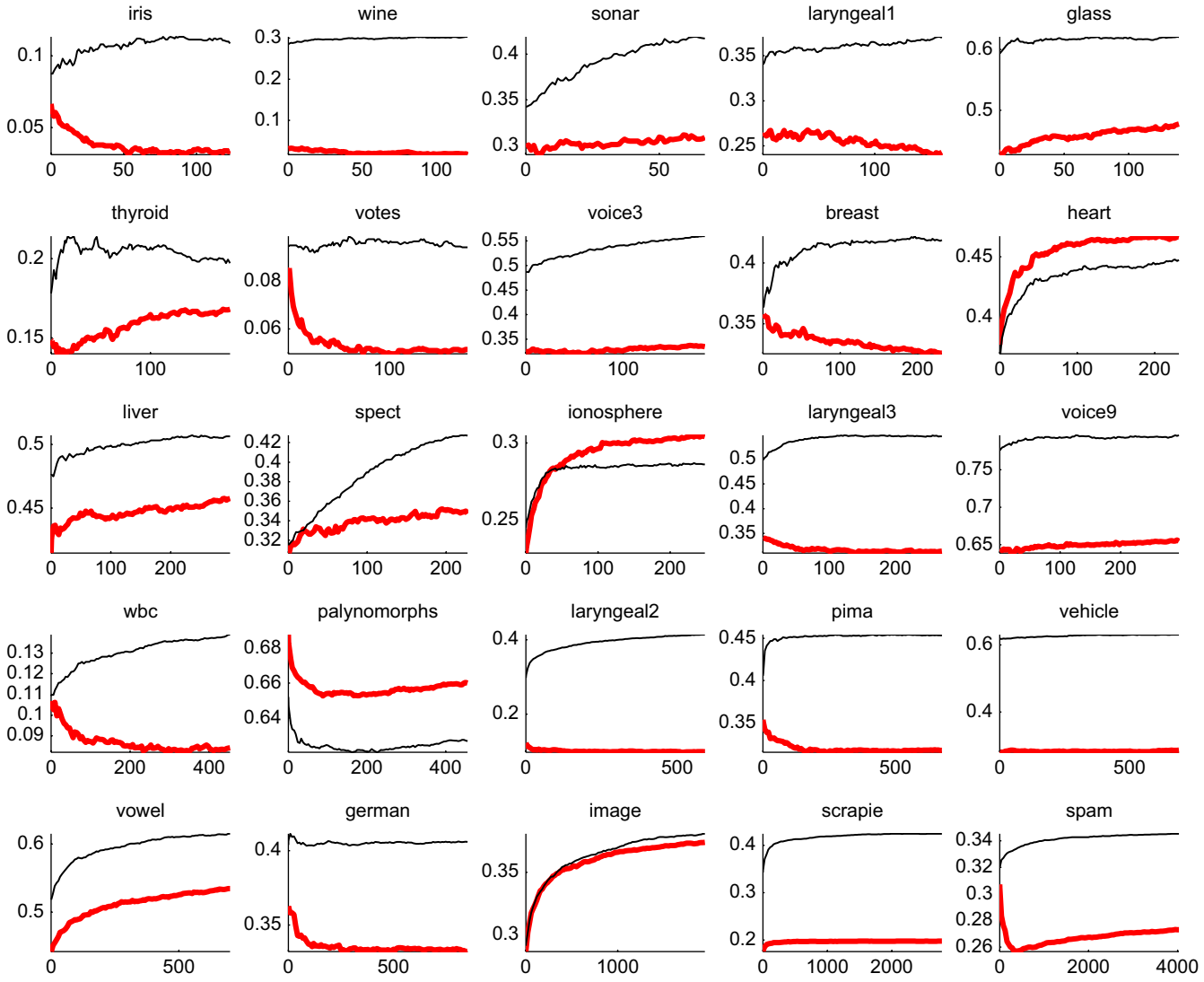

Fig. 5. Error progression with sequential processing of unlabelled data through naïve labelling. The x-axis corresponds to the number of processed unlabelled samples and the y-axis is the classification error on the testing set, averaged across 100 runs. The thin black line shows the error for the nearest mean classifier (NMC) and the thick red line shows the error for the linear discriminant classifier (LDC).

To examine the effect of NL in more detail, Table 3 shows the initial and the final classification errors for the two classifiers. A paired t-test was carried out. Indicated with "•" is a case where the initial error is significantly smaller than the final error ($p < 0.05$), and therefore NL is harmful. The opposite case where the initial error is significantly higher than the final error ($p < 0.05$) is marked with "◦".

5.4. The "soft" NL

We took NL a step further. Instead of the "hard" labelling in one of the classes, we consider a "soft" label made up by estimates of the posterior probabilities $\hat{P}(\omega_i|\mathbf{x})$, $i = 1, \ldots, c$. The two classifiers were updated using these estimates. This approach constitutes a single iteration of the EM algorithm (denoted further "EM1"). To calculate the updates, a soft count is maintained for each class. The count for class ω_i, denoted $C(\omega_i)$, is initially set to N_i, the number of samples from ω_i in the labelled training data set. To be Bayes-optimal, LDC needs the assumption that all class-conditional pdfs are Gaussians and have the same common covariance matrix. The initial covariance matrix was estimated from the training data.[1] Let $\mathbf{m}_i$ be the estimate of the mean for class ω_i, S be the estimate of the common covariance matrix and $C = \sum_{i=1}^{c} C(\omega_i)$ be the total sum of the soft counts. The EM1 updates for LDC are

$$\hat{P}(\omega_i) \leftarrow \frac{C(\omega_i) + \hat{P}(\omega_i|\mathbf{x})}{C+1}, \quad i = 1, \ldots, c \text{ (priors)}, \tag{15}$$

$$\mathbf{m}_i \leftarrow \frac{C(\omega_i)\mathbf{m}_i + \hat{P}(\omega_i|\mathbf{x})\mathbf{x}}{C(\omega_i) + \hat{P}(\omega_i|\mathbf{x})}, \quad i = 1, \ldots, c \text{ (class means)}, \tag{16}$$

$$S \leftarrow \frac{CS + \sum_{i=1}^{c} \hat{P}(\omega_i|\mathbf{x})(\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^{\mathrm{T}}}{C+1} \text{ (covariance matrix)}, \tag{17}$$

$$C(\omega_i) \leftarrow C(\omega_i) + \hat{P}(\omega_i|\mathbf{x}), \quad i = 1, \ldots, c \text{ (soft counts)}. \tag{18}$$

[1] If this covariance matrix happened to be singular, the identity matrix was used instead.

Table 3
The initial classification error and the final classification error after naïve labelling, both in (%), for NMC and LDC

Data set	Initial error (NMC)	Final NL error (NMC)	Final EM1 error (NMC)	Initial error (LDC)	Final NL error (LDC)	Final EM1 error (LDC)
iris	8.8	10.9•	12.3•	5.9	3.4○	21.9•
wine	28.5	30.3•	32.6•	3.2	1.7○	9.4•
sonar	34.2	41.7•	35.1•	30.0	30.9–	28.9–
laryngeal1	34.0	37.0•	35.9–	26.4	24.3○	24.6○
glass	59.3	62.0•	51.0○	43.1	47.7•	45.0•
thyroid	17.8	19.7–	45.0•	14.8	16.8•	17.2–
votes	9.5	9.4–	11.1•	8.6	5.1○	6.1–
voice3	48.7	56.0•	52.2•	32.5	33.5–	35.8•
breast	36.6	41.9•	40.5•	35.8	32.6○	35.3○
heart	37.2	44.7•	38.5–	38.5	46.6•	37.2–
liver	47.6	50.6•	47.7•	42.6	45.8•	43.5–
spectcontinuous	31.5	42.7•	39.7•	30.7	34.7•	33.2•
ionosphere	24.5	28.6•	29.5•	23.2	30.4•	30.9•
laryngeal3	49.8	54.6•	57.8•	34.1	31.4○	33.4○
voice9	77.6	79.6•	72.8○	64.4	65.5•	64.9–
wbc	10.9	13.9•	8.9○	10.7	8.4○	14.4•
palynomorphs	65.2	62.7○	61.1○	68.3	66.0○	57.7○
laryngeal2	29.8	41.4•	46.9•	12.1	9.9○	11.2○
pima	40.5	45.3•	44.1•	35.2	31.7○	29.1○
vehicle	61.8	63.0•	62.7–	28.2	28.6–	31.0•
vowel	51.9	61.5•	59.5•	44.2	53.4•	63.7•
german	40.4	40.6–	41.1•	36.3	33.2○	32.9○
image	29.1	38.1•	51.2•	28.7	37.4•	50.9•
scrapie	34.3	42.4•	48.2•	17.7	19.8•	20.2•
spam	32.0	34.5•	32.3–	30.6	27.3○	23.6○

• indicates that the final error is significantly worse than the initial error (loss).

○ indicates that the final error is significantly better than the initial error (win).

Table 4
The win/loss scores for NL and EM1 on the 25 data sets

Factor	NMC(NL)	NMC(EM1)	LDC(NL)	LDC(EM1)
0.5	2 (0)/23 (20)	7 (4)/18 (14)	15 (12)/10 (9)	13 (10)/12 (7)
1.0	3 (1)/22 (21)	6 (4)/19 (17)	12 (12)/13 (10)	9 (8)/16 (11)
2.0	3 (2)/21 (18)	7 (4)/17 (15)	8 (3)/16 (12)	7 (4)/17 (12)
3.0	2 (2)/18 (17)	5 (4)/15 (12)	5 (2)/15 (13)	4 (1)/16 (9)

Given in parentheses are the statistically significant scores out of the total number of wins/losses. "Factor" indicates the size of the initial labelled data set used for training (Factor$\times n \times c$).

For NMC, only the updates of the means (16) and the soft counts (18) are needed. The final error rates for EM1 are displayed in Table 3. The statistically significant differences with the initial errors are indicated next to each value.

Finally, we ran all the experiments for different sizes of the initial labelled set used for training. Taking the size to be Factor $\times\, n \times c$, we varied Factor in the set $\{0.5, 1, 2, 3\}$. The differences between the initial and the final error are summarised in Table 4. The notation $a(b)/c(d)$ in the table means that in a out of the 25 data sets the final error rate has been smaller than the initial error rate (win), and in b of these cases the difference has been statistically significant (one-sided paired t-test, $\alpha = 0.05$). On the other hand, for c data sets the final error rate has been larger than the initial error rate (loss), and in d of these cases the difference has been statistically significant.[2]

5.5. Discussion

The results in Table 3 indicate, almost unequivocally, that NL is inadequate for NMC. EM1 is slightly better but the losses outweigh the wins by a large number. On the other hand, LDC may or may not benefit significantly from NL and EM1. This means that applying unsupervised on-line training to LDC is a gamble. The reason why LDC behaves better than NMC is rooted in the underlying models of the two classifiers. NMC would be the optimal classifier if the underlying distributions were Gaussians, with equal diagonal covariance matrices with the same variance along all features, and equal priors. When the assumptions are met, NMC may benefit from using unlabelled data, even in the simple naïve way considered in this study. However, if the true distribution is not close to the one that is a prerequisite for optimality of NMC, NL will do more harm and will cause the run-away behaviour observed in the experiment. LDC takes the assumptions a step further. In order for LDC to be the optimal classifier for the problem, the classes must be Gaussian with equal covariance matrices. This is a more plausible assumption than the one for NMC, which makes LDC the more adequate classifier between the two. Presumably classifiers relying on more sophisticated models, e.g., where each class-conditional pdf is represented as a mixture of Gaussians, will fare better than LDC. However, if the assumption is far from the real situation NL is likely to do more harm than good.

Table 4 shows that NMC is not affected much by the size of the training set. This is to be expected knowing that NMC needs only estimates of the class means. With a larger training set these estimates will be slightly more accurate. However, this improved accuracy is not sufficient to prevent the run-away behaviour of both NL and EM1. While both labelling strategies are rather harmful than useful with NMC, EM1 is the better of the two. For some data sets where NL failed, EM1 managed to drive the error rate to a smaller final value. With LDC there is a clear pattern in Table 4 indicating that when training sets of adequate size are available, unsupervised on-line training by NL or EM1 is worthless.

The empirical findings in our experiment resonate with the conclusions of the series of studies by Cozman et al. [8–10]. One interesting result in favour of NL was that for very small labelled training sets (Factor = 0.5, Table 4), LDC with NL brought the initial error down in 60% of the data sets, and $\frac{2}{3}$ of the differences were statistically significant. With larger training sets sampled from the same data sets, NL was not that successful. This comes to show that the optimality assumptions for LDC are not likely to hold. Nonetheless, using a very small labelled sample appears to lead to such an inac-

[2] The reason that $a + c < 25$ for Factor = 3 and 4 is that we dropped from the experiment the data sets for which the size of the training set was larger than 30% of the total size. This was done because in these cases the on-line training was considered to be too short for the error rates to reach convergence.

curate initial classifier that NL can do well in spite of all the expectations to the contrary. This suggests that there may be a bound on the size of the training data limiting the scope of NL as an on-line training strategy.

6. Conclusions

Previous studies have proved that unlabelled data would damage the classifier when the model is not guessed correctly and will improve the classifier if the guess is correct. The further assumption there is that there is an approximation procedure in place for the pdfs, which is allowed to iterate through the data, and is asymptotically optimal. Here we chose the naïve labelling (NL) strategy to augment the training set, which is the basic stepping stone for on-line semi-supervised learning. NL makes a single pass through the data, thereby denying repeating iterative optimisation of the pdf approximations to take place. Our study extends previous theoretical results in the following way. We considered a special case of two Gaussian pdfs and one feature. There is no theoretical proof of optimality of NL even for the case of correctly guessed pdfs. For our special case we found that NL will converge to the optimal boundary if the distributions are guessed correctly for the given classifier (NMC or LDC). We carried out a simulation experiment and an experiment with real data. While our findings are mostly in the same vein with the studies by Cozman and co-authors, the experiments where very small labelled data sets were used for training suggest that LDC may benefit from NL even when the optimality assumptions are likely to be false.

In our scenario *all* new coming points are used for re-training the classifier. In most studies where NL has been attempted for real-life problems, there has been a confidence threshold for accepting a new point in the training set. The classifier is re-trained only if the new point is classified with certainty greater than the threshold. This modification introduces an extra parameter—the confidence threshold—which needs to be picked in advance. Besides, this strategy may not be as successful as intuition suggests [4].

Another question here is whether we need to update the classifier with all the streaming data if the underlying distribution does not change. Instead of updating the classifier on-line, it makes more sense to collect a large enough amount of unlabelled data, put the classifier on hold, and run an EM algorithm on the whole data set. The reason for adopting NL is that we are looking for potential application of the on-line classifier to non-stationary environments where the distribution of the problem may change with time.

Acknowledgement

This work was sponsored by EPSRC Grant #EP/D04040X/1.

Appendix

This appendix details the derivation of $f(b)$ for the NMC, two classes with normal distributions, $p(x|\omega_1) \sim N(\mu_1, \sigma_1^2)$, $p(x|\omega_2) \sim N(\mu_2, \sigma_2^2)$, $P(\omega_1) = \lambda$ and a fixed boundary b.

The new boundary $b_{\text{new}} = f(b)$ is calculated as the middle of the means of the two distributions $p_1(x)$ and $p_2(x)$ defined in Eq. (4). Thus

$$f(b) = \frac{1}{2Z(b)} \underbrace{\int_{-\infty}^{b} xp(x)\,\mathrm{d}x}_{A} + \frac{1}{2(1-Z(b))} \underbrace{\int_{b}^{\infty} xp(x)\,\mathrm{d}x}_{B}. \tag{19}$$

Taking the two integrals separately and substituting $\lambda p(x|\omega_1) + (1-\lambda)p(x|\omega_2)$ for the unconditional pdf $p(x)$, we get

$$A = \int_{-\infty}^{b} x \left[\frac{\lambda}{\sqrt{2\pi}\sigma_1} \exp\left\{ -\frac{(x-\mu_1)^2}{2\sigma_1^2} \right\} + \frac{(1-\lambda)}{\sqrt{2\pi}\sigma_2} \exp\left\{ -\frac{(x-\mu_2)^2}{2\sigma_2^2} \right\} \right] \mathrm{d}x. \tag{20}$$

The term A breaks into two integrals. Making the substitutions $t = (x-\mu_1)/\sigma_1$ in the first integral and $u = (x-\mu_2)/\sigma_2$ in the second integral, we arrive at

$$A = \frac{\lambda}{\sqrt{2\pi}} \int_{-\infty}^{(b-\mu_1)/\sigma_1} (\sigma_1 t + \mu_1) \exp\left\{ -\frac{t^2}{2} \right\} \mathrm{d}t + \frac{(1-\lambda)}{\sqrt{2\pi}} \int_{-\infty}^{(b-\mu_2)/\sigma_2} (\sigma_2 t + \mu_2) \exp\left\{ -\frac{u^2}{2} \right\} \mathrm{d}u \tag{21}$$

$$= \frac{\lambda\sigma_1}{\sqrt{2\pi}} \int_{-\infty}^{(b-\mu_1)/\sigma_1} t \exp\left\{ -\frac{t^2}{2} \right\} \mathrm{d}t + \lambda\mu_1 \underbrace{\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{(b-\mu_1)/\sigma_1} \exp\left\{ -\frac{t^2}{2} \right\} \mathrm{d}t}_{\Phi((b-\mu_1)/\sigma_1)} \tag{22}$$

$$+ \frac{(1-\lambda)\sigma_2}{\sqrt{2\pi}} \int_{-\infty}^{(b-\mu_2)/\sigma_2} u \exp\left\{ -\frac{u^2}{2} \right\} \mathrm{d}u + (1-\lambda)\mu_2 \underbrace{\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{(b-\mu_2)/\sigma_2} \exp\left\{ -\frac{u^2}{2} \right\} \mathrm{d}u}_{\Phi((b-\mu_2)/\sigma_2)} \tag{23}$$

$$= \frac{\lambda\sigma_1}{\sqrt{2\pi}} \exp\left\{ -\frac{(b-\mu_1)^2}{2\sigma_1^2} \right\} + \lambda\mu_1 \Phi\left(\frac{b-\mu_1}{\sigma_1} \right) \tag{24}$$

$$+ \frac{(1-\lambda)\sigma_2}{\sqrt{2\pi}} \exp\left\{ -\frac{(b-\mu_2)^2}{2\sigma_2^2} \right\} + (1-\lambda)\mu_2 \Phi\left(\frac{b-\mu_2}{\sigma_2} \right), \tag{25}$$

where $\Phi(.)$ is the cumulative distribution function of the standardised normal distribution. In a similar way, term B in Eq. (19) can be expressed as

$$B = \int_{b}^{\infty} xp(x)\,\mathrm{d}x = -\int_{-\infty}^{-b} yp(-y)\,\mathrm{d}y \tag{26}$$

$$= -\int_{-\infty}^{-b} x \left[\frac{\lambda}{\sqrt{2\pi}\sigma_1} \exp\left\{ -\frac{(-x-\mu_1)^2}{2\sigma_1^2} \right\} + \frac{(1-\lambda)}{\sqrt{2\pi}\sigma_2} \exp\left\{ -\frac{(-x-\mu_2)^2}{2\sigma_2^2} \right\} \right] \mathrm{d}x \tag{27}$$

$$= \frac{\lambda\sigma_1}{\sqrt{2\pi}} \exp\left\{ -\frac{(b-\mu_1)^2}{2\sigma_1^2} \right\} - \lambda\mu_1 \Phi\left(-\frac{b-\mu_1}{\sigma_1} \right) \tag{28}$$

$$+ \frac{(1-\lambda)\sigma_2}{\sqrt{2\pi}} \exp\left\{ -\frac{(b-\mu_2)^2}{2\sigma_2^2} \right\} - (1-\lambda)\mu_2 \Phi\left(-\frac{b-\mu_2}{\sigma_2} \right). \tag{29}$$

To simplify notations, let

$$\Phi_1 = \Phi\left(\frac{b-\mu_1}{\sigma_1}\right) \quad \text{and} \quad \Phi_2 = \Phi\left(\frac{b-\mu_2}{\sigma_2}\right). \tag{30}$$

Putting A and B back in Eq. (19), and taking into account that $\Phi(\zeta) = 1 - \Phi(-\zeta)$, we obtain the final expression for $f(b)$

$$f(b) = \frac{1}{\sqrt{8\pi}Z(b)(1-Z(b))}\left[\lambda\sigma_1 \exp\left\{-\frac{(b-\mu_1)^2}{2\sigma_1^2}\right\}\right.$$

$$\left. + (1-\lambda)\sigma_2 \exp\left\{-\frac{(b-\mu_2)^2}{2\sigma_2^2}\right\}\right] \tag{31}$$

$$+ \frac{\lambda\mu_1\Phi_1}{Z(b)} + \frac{(1-\lambda)\mu_2\Phi_2}{1-Z(b)}$$

$$- \frac{1}{2}\left[\frac{\lambda\mu_1}{Z(b)} + \frac{(1-\lambda)\mu_2}{1-Z(b)}\right], \tag{32}$$

where the normalising constant $Z(b)$ is

$$Z(b) = \lambda\Phi_1 + (1-\lambda)\Phi_2. \tag{33}$$

References

[1] F. Roli, Semi-supervised multiple classifier systems: background and research directions, in: Proceedings of the Multiple Classifier Systems Workshop (MCS2005), CA, USA, 2005, pp. 1–11.

[2] M. Seeger, Learning with labeled and unlabeled data, Technical Report, University of Edinburgh, 2002.

[3] X. Zhu, Semi-supervised learning literature survey, Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005 ⟨http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf⟩.

[4] G. Nagy, Classifiers that improve with use, in: Proceedings of the Conference on Pattern Recognition and Multimedia (IEICE), Tokyo, Japan, 2004, pp. 79–86.

[5] Z. Liu, J. Almhana, V. Choulakian, R. McGorman, Online EM algorithm for mixture with application to Internet traffic modeling, Comput. Stat. Data Anal. 50 (4) (2004) 1052–1071.

[6] N. Suematsu, K. Maebashi, A. Hayashi, An online EM algorithm using component reduction, in: Proceedings of the 15th European Conference on Machine Learning (ECML), Pisa, Italy, 2004.

[7] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT), Madison, WI, USA, 1998, pp. 92–100.

[8] I. Cohen, F.G. Cozman, N. Sebe, M.C. Cirelo, T.S. Huang, Semisupervised learning of classifiers: theory, algorithms, and their application to human–computer interaction, IEEE Trans. Pattern Anal. Mach. Intell. 26 (12) (2004) 1553–1568.

[9] F.G. Cozman, I. Cohen, M.C. Cirelo, Semi-supervised learning of mixture models, in: Proceedings of the 20th International Conference on Machine Learning, Washington, DC, USA, 2003, pp. 99–106.

[10] F.G. Cozman, I. Cohen, Unlabeled data can degrade classification performance of generative classifiers, in: Proceedings of the 15th International FLAIR Conference, Pensacola, FL, USA, 2002, pp. 327–331.

[11] K.P. Nigam, Using unlabeled data to improve text classification, Ph.D. Thesis, Pittsburgh, US, 2001.

[12] S. Ganesalingam, G.J. McLachlan, The efficiency of a linear discriminant function based on unclassified initial samples, Biometrika 65 (3) (1978) 658–662.

[13] G.J. McLachlan, Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis, J. Am. Stat. Assoc. 70 (1975) 365–369.

[14] T.J. O'Neill, Normal distribution with unclassified observations, J. Am. Stat. Assoc. 73 (1978) 821–826.

[15] D.M. Titterington, Updating a diagnostic system using unconfirmed cases, Appl. Stat. 25 (3) (1976) 238–247.

[16] G.J. McLachlan, S. Ganesalingam, Updating a discriminant function on the basis of unclassified data, Commun. Stat. Simulation Comput. 11 (6) (1982) 753–767.

[17] P. Domingos, G. Hulten, A general framework for mining massive data streams, J. Comput. Graphical Stat. 12 (2003) 945–949.

[18] W.N. Street, Y.S. Kim, A streaming ensemble algorithm (SEA) for large-scale classification, in: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York, 2001, pp. 377–382.

[19] H. Robbins, S. Monro, A stochastic approximation method, Ann. Math. Stat. 22 (3) (1951) 400–407.

[20] L.W. Johnson, R.D. Riess, Numerical Analysis, second ed., Addison-Wesley, USA, 1977.

[21] C.L. Blake, C.J. Merz, UCI repository of machine learning databases, 1998 ⟨http://www.ics.uci.edu/mlearn/MLRepository.html⟩.

About the Author—LUDMILA KUNCHEVA received the M.Sc. degree from the Technical University, Sofia, in 1982 and the Ph.D. degree from the Bulgarian Academy of Sciences in 1987. Until 1997 she worked at the Central Laboratory of Biomedical Engineering, Bulgarian Academy of Sciences, as a senior research associate. Dr. Kuncheva is currently a reader at the School of Electronics and Computer Science, Bangor University, UK. Her interests include pattern recognition and classification, machine learning, classifier combination and nearest neighbour classifiers.

About the Author—CHRISTOPHER WHITAKER received the M.Sc. degree in mathematical statistics from the University of Manchester in 1974. Until 1980 he was a research associate in the Department of Occupational Health, University of Manchester. Since 1981 he has worked as a lecturer in statistics in the Faculty of Science and since 2004 as university statistics advisor in the School of Psychology at the Bangor University, UK. His interests cover the applications of statistics in the psychological and biomedical sciences.

About the Author—ANAND NARASIMHAMURTHY received the M.Eng. and Ph.D. degrees from the Pennsylvania State University, Pennsylvania, USA, in 2003 and 2006, respectively. Dr. Narasimhamurthy is currently a post doctoral researcher at the School of Computer Science and Informatics, University College Dublin (UCD), Dublin, Ireland. His interests include pattern recognition and classification, machine learning, classifier combination and clustering.