BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 2 2014, pages 242–250
doi:10.1093/bioinformatics/btt662

Data and text mining

Advance Access publication November 20, 2013

Effect of separate sampling on classification accuracy

Mohammad Shahrokh Esfahani[1,2] and Edward R. Dougherty[1,2,*]

[1]Department of Electrical and Computer Engineering and [2]Center for Bioinformatics and Genomic Systems Engineering, Texas A&M University, College Station, TX 77843, USA

Associate Editor: Prof. Martin Bishop

ABSTRACT

Motivation: Measurements are commonly taken from two phenotypes to build a classifier, where the number of data points from each class is predetermined, not random. In this 'separate sampling' scenario, the data cannot be used to estimate the class prior probabilities. Moreover, predetermined class sizes can severely degrade classifier performance, even for large samples.

Results: We employ simulations using both synthetic and real data to show the detrimental effect of separate sampling on a variety of classification rules. We establish propositions related to the effect on the expected classifier error owing to a sampling ratio different from the population class ratio. From these we derive a sample-based minimax sampling ratio and provide an algorithm for approximating it from the data. We also extend to arbitrary distributions the classical population-based Anderson linear discriminant analysis minimax sampling ratio derived from the discriminant form of the Bayes classifier.

Availability: All the codes for synthetic data and real data examples are written in MATLAB. A function called mmratio, whose output is an approximation of the minimax sampling ratio of a given dataset, is also written in MATLAB. All the codes are available at: http://gsp.tamu.edu/Publications/supplementary/shahrokh13b.

Contact: edward@ece.tamu.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on August 5, 2013; revised on November 9, 2013; accepted on November 11, 2013

1 INTRODUCTION

The medical community is being confronted with serious problems of reproducibility in the development of biomarkers. The issue has been highlighted by a recent report regarding comments by Janet Woodcock, FDA drug division head. The report states, 'Based on conversations Woodcock has had with genomics researchers, she estimated that as much as 75 percent of published biomarker associations are not replicable. "This poses a huge challenge for industry in biomarker identification and diagnostics development," she said (Ray, 2011).' Many issues affect reproducibility, including the measurement platform, specimen handling, data normalization and sample compatibility between the original and subsequent studies. These matters concern experimental procedures and are not our concern here; rather, we are interested in the methodology for designing classifiers. One issue in this regard is the impact of inaccurate error estimation owing to small samples. This has been previously quantified (Yousefi and Dougherty, 2012). Here we are interested in a different problem, one that will confront us even if we have large samples and perfect error estimation: the effect of having predetermined sample sizes so that sampling is not random.

In classification studies it is typically a tacit assumption that sampling is random; indeed, it is commonplace for this assumption to be made throughout a text on classification. For instance, Devroye *et al*. declare on page 2 of their text that all sampling is random (Devroye, 1996). The assumption is so pervasive that it can be applied without mention. With regard to the problem at hand, Duda *et al*. (2001) state, 'In typical supervised pattern classification problems, the estimation of the prior probabilities presents no serious difficulties.' But, in fact, there are often serious difficulties.

Under the assumption of random sampling, the data set, $S_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$, is drawn independently from a fixed distribution of feature-label pairs, $(\mathbf{X}, Y)$; in particular, this means that if a sample of size n is drawn for a binary classification problem, then the numbers of sample points, n_0 and n_1, in classes 0 and 1, respectively, are random variables such that $n_0 + n_1 = n$. An immediate consequence of the random-sampling assumption is that the prior probability $c = \Pr(Y = 0)$ can be consistently estimated by the sampling ratio, namely, by $\hat{c} = \frac{n_0}{n}$. Consistency is nothing but Bernoulli's weak law of large numbers, namely, $\frac{n_0}{n} \to c$ in probability. Thus, if the sample is large, we can expect the sampling ratio to be close to the prior probability.

Suppose the sampling is not random, in the sense that the ratios $\frac{n_0}{n}$ and $\frac{n_1}{n}$ are chosen prior to sampling. In this 'separate (stratified) sampling' case, $S_n = S_{n_0} \cup S_{n_1}$, where the sample points in S_{n_0} and S_{n_1} are selected randomly from Π_0 and Π_1 but, given n, the individual class counts n_0 and n_1 are not random. Then, in effect, we have no sensible estimate of c. One could let $\hat{c} = \frac{n_0}{n}$, but there would be no reason to do so.

Since our aim is to use the data to train a classifier, does the inability to consistently estimate c matter? Clearly in the case of linear discriminant analysis (LDA) it does, since the LDA classifier is defined by $\psi_n(\mathbf{x}) = 1$ if $D_{\text{samp}}(\mathbf{x}) \leq 0$ and $\psi_n(\mathbf{x}) = 0$ if $D_{\text{samp}}(\mathbf{x}) > 0$, where

$$D_{\text{samp}}(\mathbf{x}) = \left(\mathbf{x} - \frac{\hat{\boldsymbol{\mu}}_0 + \hat{\boldsymbol{\mu}}_1}{2}\right)^T \hat{\boldsymbol{\Sigma}}^{-1}(\hat{\boldsymbol{\mu}}_0 - \hat{\boldsymbol{\mu}}_1) - \ln\frac{1-\hat{c}}{\hat{c}}, \quad (1)$$

and $\hat{\boldsymbol{\mu}}_0$ and $\hat{\boldsymbol{\mu}}_1$ are the sample means of the class-conditional populations Π_0 and Π_1, respectively, and $\hat{\boldsymbol{\Sigma}}$ is the pooled sample covariance matrix. The rationale for the LDA discriminant is that the estimators converge to the population parameters

*To whom correspondence should be addressed.

as $n \to \infty$, in which case the resulting discriminant, $D_{\text{Bayes}}(\mathbf{x})$, defines the Bayes (optimal) classifier in the two-class Gaussian model with common covariance matrix. It is obvious from Equation (1) that an estimate of c is required for LDA and a bad choice of $\hat{c}$ will negatively impact the classifier. This fact, which is a consequence of separate sampling, has long been recognized (Anderson, 1951).

The situation is less transparent with model-free classification rules such as support vector machines. In this article we use simulation to study the effect of separate sampling on several different classification rules, where the role of c does not appear explicitly in classifier learning. We generate separate samples with different ratios $r = \frac{n_0}{n}$ and consider the expected error, $E[\varepsilon_n|r]$, of the designed classifier, given r, where the error of classifier ψ_n is defined by $\varepsilon_n = \Pr(\psi_n(\mathbf{X}) \neq Y)$, the probability of misclassification. We will see that the penalty for separate sampling without knowledge of c can be severe.

With random (or, 'mixed') sampling, rather than being fixed prior to sampling, r is a sample-dependent random variable. In this case, $E[\varepsilon_n|r]$ denotes the expectation of the error conditioned on r and the expected classification error is given by $E[\varepsilon_n] = E_r[E[\varepsilon_n|r]]$, where the outer expectation is relative to the distribution of r. The classifier error is likely to be smaller when the sampling ratio r is close to c. Hence, if one happens to fix r sufficiently close to c, then $E[\varepsilon_n|r] < E[\varepsilon_n]$. Because $r \to c$ in probability as $n \to \infty$ for mixed sampling, as n gets larger the distribution of r gets more tightly concentrated around c, so that the distribution of $E[\varepsilon_n|r]$ (as function of r) gets more tightly packed around $E[\varepsilon_n]$, which in turn means that to have $E[\varepsilon_n|r] < E[\varepsilon_n]$ one must choose r very close to c. To illustrate this phenomenon, consider 2D Gaussian class-conditional densities with means at (0.3, 0.3) and (0.8, 0.8), possessing common covariance matrix $\sigma^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix and $\sigma^2 = 0.4$, and with $c = 0.6$. For this model, the Bayes error is $\varepsilon_{\text{Bayes}} = 0.27$. Figure 1 shows the difference $E[\varepsilon_n] - E[\varepsilon_n|r]$ for different values of r and different sample sizes when using LDA. If $r = 0.6$, then $E[\varepsilon_n|r] < E[\varepsilon_n]$ for all n. If $r = 0.7$, which is fairly close to $c = 0.6$, then $E[\varepsilon_n|r] < E[\varepsilon_n]$ for $n \leq 25$. Notice the lack of symmetry, both 0.5 and 0.7 being equally close to c. This should not be surprising because we should not expect the distribution of $E[\varepsilon_n|r]$ to be symmetric.

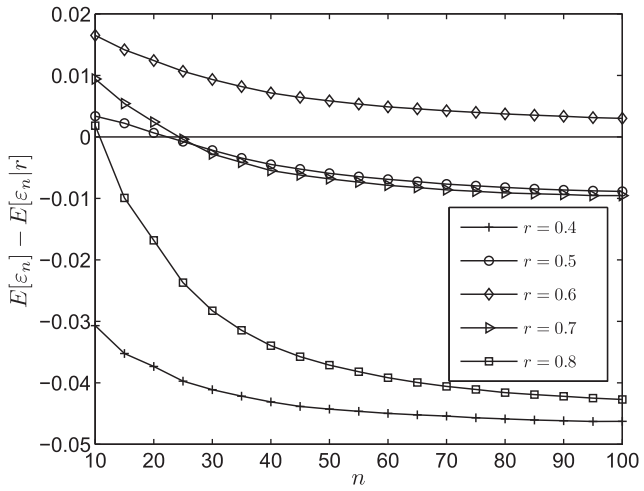

Fig. 1. The difference $E[\varepsilon_n(c)] - E[\varepsilon_n(c)|r]$ for different values of r as a function of sample size, n, with $c = 0.6$. The class conditional densities parameters are as follows: $\boldsymbol{\mu}_0 = (0.3, 0.3)$, $\boldsymbol{\mu}_1 = (0.8, 0.8)$, and $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = 0.4\mathbf{I}$, leading to the Bayes error $\varepsilon_{\text{Bayes}} = 0.27$

Let us examine Figure 1 from the practitioner's perspective. Suppose that cost limits the sample to a given size n. If the sample is random, then the expected error of the designed classifier will be $E[\varepsilon_n]$, which is unknown since the feature-label distribution is unknown. Consider three cases: (i) if c is accurately known from existing population statistics regarding the two classes, say BRCA1 and BRCA2 breast cancer, then no matter what the sample size, it is best to do separate sampling with $n_0 \approx cn$; (ii) if c is approximately known, meaning that the practitioner believes that c is close to c', then, for small n, it may be best, or at least acceptable, to do separate sampling with $n_0 \approx c'n$, and the results will likely still be acceptable for large n, though not as good as with random sampling; (iii) if the practitioner has no idea what c is, then sampling must be random because the penalty for separate sampling can be very large. While, at this point, these comments refer specially to Figure 1, which is for LDA, a salient point to be made in this article is that they are quite general and, moreover, can be extended to the commonplace separate sampling situation where one cannot choose n_0 and n_1.

Why is all of this a major issue for bioinformatics? Simply put separate sampling is ubiquitous in bioinformatics, in particular, with genomic classification, where a standard approach is to take tissue samples from two classes, say, different types of cancer or different stages of cancer, for which the number of specimens in each class is not chosen randomly, and then to design a classifier. The Supplementary Material lists 20 published studies using separate sampling. In each case we give the classification problem, sample sizes, classification rule and error estimator. Even if an error estimate is exact for the problem at hand—that is, for the sampling ratio represented by the data—what does it mean relative to the classification error for future observations (say, patients)? That depends on the true prior probabilities, which we do not know.

2 SYSTEMS AND METHODS

2.1 Effect of sampling ratio—synthetic data

We employ simulation to study the effect of the sampling ratio for different classification rules using a general model based on multivariate Gaussian distributions with a blocked covariance structure. This model conforms to the setting where blocks represent correlated gene groups, say common pathways, and between-block correlation is negligible (Doughtery *et al.*, 2007; Hua *et al.*, 2005; Shmulevich and Dougherty, 2007). The model has several parameters that can generate a battery of covariance matrices. For example, a 4-block covariance matrix with block size 3 has the structure

$$\boldsymbol{\Sigma}_y = \begin{bmatrix} \mathbf{B}_{y,1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{y,2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{y,3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{y,4} \end{bmatrix}, \tag{2}$$

where

$$\mathbf{B}_{y,i} = \begin{bmatrix} \sigma_y^2 & \rho_i\sigma_y^2 & \rho_i\sigma_y^2 \\ \rho_i\sigma_y^2 & \sigma_y^2 & \rho_i\sigma_y^2 \\ \rho_i\sigma_y^2 & \rho_i\sigma_y^2 & \sigma_y^2 \end{bmatrix}, \tag{3}$$

in which σ_y^2 is the variance of each variable and the $\rho_i, i \in \{1,2,3,4\}$, are the correlation coefficients inside blocks. We consider both identical and unequal covariance matrices. We assume common correlation coefficient, $\rho_i = \rho, i \in \{1,2,3,4\}$.

A typical microarray or next-gen RNA sequencing (Mortazavi *et al.*, 2008; Wang *et al.*, 2009) experiment yields expressions for thousands of genes, but a small number of sample points, typically <200. Therefore, data-based feature selection is typically employed; however, since our sole aim is to study the effect of the ratio r on the expected true error, we do not consider feature selection and assume a model containing a reasonable number, D, of features (which is equivalent to assuming that a set of D genes has been chosen by the researcher based on prior biological knowledge). We let $D = 15$. Two covariance matrix settings are considered: identical covariance matrices, $\sigma_0^2 = \sigma_1^2 = 0.4$, and unequal covariance matrices, $\sigma_0^2 = 0.4, \sigma_1^2 = 1.6$, with block size $l = 5$ and correlation coefficient $\rho = 0.8$ corresponding to tight correlation within a block. The parameter settings are summarized in Table 1.

Seven classification rules are considered: 3-nearest neighbor (3NN), 5-nearest neighbor (5NN), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), linear support vector machine (L-SVM), radial basis function SVM (RBF-SVM) and decision tree (DT). The SVM classifiers are trained from the package LibSVM written in MATLAB (Chang and Lin, 2011). A decision tree classifier is trained using the MATLAB classregtree function.

2.2 Effect of sampling ratio—real data

Four microarray real datasets are used: pediatric acute lymphoblastic leukemia (ALL) (Yeoh *et al.*, 2002), acute myeloid leukemia (AML) (Valk *et al.*, 2004), multiple myeloma (Zhan *et al.*, 2006) and breast cancer (Desmedt *et al.*, 2007). We follow the data preparation instructions reported in the cited articles. The properties of these datasets are summarized in Table 2. The right-most column in Table 2 contains the initial feature size, number of sample points in classes 0 and 1, respectively, from left to right. The Supplementary Material provides detailed descriptions of these datasets. The same classification rules as those used for the synthetic data are applied to the real data. T-test feature selection is used to reduce the original set of genes down to $D = 15$.

2.3 Holdout error estimation

Because we are going to use real data, we wish to use holdout error estimation; however, the standard holdout procedure, which is unbiased with random sampling, become biased, perhaps severely so, with separate sampling. Therefore, we redefine holdout for separate sampling.

Table 1. Distribution model parameters

Parameters	Value/description
Mean	$\mu_0 = 0.3\mathbf{1}_D$, $\mu_1 = 0.8\mathbf{1}_D$
Covariance matrix	$\sigma_0^2 = 0.4, \sigma_1^2 = 0.4$ (identical covariance) $\sigma_0^2 = 0.4, \sigma_1^2 = 1.6$ (unequal covariance)
Block size	$l = 5$
Feature size	$D = 15$
Feature block correlation	$\rho = 0.8$

The true error of a designed classifier ψ_n is given by

$$\begin{aligned} \varepsilon_n &= \Pr(\psi_n(\mathbf{X}) \neq Y) \\ &= c\Pr(\psi_n(\mathbf{X}) \neq 0 | Y = 0) \\ &\quad + (1-c)\Pr(\psi_n(\mathbf{X}) \neq 1 | Y = 1) \\ &= c\varepsilon_n^0 + (1-c)\varepsilon_n^1. \end{aligned} \tag{4}$$

Relative to a random sample, S_n, the expected true error is

$$E_{S_n}[\varepsilon_n] = cE_{S_n}[\varepsilon_n^0] + (1-c)E_{S_n}[\varepsilon_n^1]. \tag{5}$$

For standard holdout error estimation, the sample is split into t points (the training set) to train the classifier and m points (the test set) to estimate the error, where in this scenario the notation indicates that the total sample size is $n = t + m$. Let S_t, S_m, S_{m_0}, and S_{m_1} denote the set of training data, the full set of test data, the class-0 test points, and the class-1 test points, respectively. The holdout estimator is

$$\begin{aligned} \hat{\varepsilon}(\psi_n) &= \frac{1}{m}\sum_{(\mathbf{X}_i, Y_i)\in S_m} 1_{\psi_n(\mathbf{X}_i)\neq Y_i} \\ &= \frac{m_0}{m}\frac{1}{m_0}\sum_{(\mathbf{X}_i, Y_i)\in S_{m_0}} 1_{\psi_n(\mathbf{X}_i)\neq Y_i} \\ &\quad + \frac{m_1}{m}\frac{1}{m_1}\sum_{(\mathbf{X}_i, Y_i)\in S_{m_1}} 1_{\psi_n(\mathbf{X}_i)\neq Y_i} \\ &= \frac{m_0}{m}\hat{\varepsilon}^0(\psi_n) + \frac{m_1}{m}\hat{\varepsilon}^1(\psi_n), \end{aligned} \tag{6}$$

where $\hat{\varepsilon}^0$ and $\hat{\varepsilon}^1$ denote the holdout estimators of ε_n^0 and ε_n^1. Taking expectations in (6) yields

$$E_{S_n}[\hat{\varepsilon}] = cE_{S_n}[\hat{\varepsilon}^0] + (1-c)E_{S_n}[\hat{\varepsilon}^1]. \tag{7}$$

Because the test data are independent from the training data, the holdout estimator is unbiased given the training data, which means that $E_{S_n}[\hat{\varepsilon}|S_t] = \varepsilon_n$. Taking the expectation relative to the training data yields $E_{S_n}[\hat{\varepsilon}] = E_{S_t}[E_{S_n}[\hat{\varepsilon}|S_t]] = E_{S_n}[\varepsilon_n]$. Similar expressions apply to $\hat{\varepsilon}^0$ and $\hat{\varepsilon}^1$, namely, $E_{S_n}[\hat{\varepsilon}^0] = E_{S_n}[\varepsilon_n^0]$ and $E_{S_n}[\hat{\varepsilon}^1] = E_{S_n}[\varepsilon_n^1]$. Thus, $\hat{\varepsilon}, \hat{\varepsilon}^0$, and $\hat{\varepsilon}^1$ are unbiased estimators of $\varepsilon_n, \varepsilon_n^0$, and ε_n^1, respectively, and Expression (7) corresponds term by term to (5).

With separate sampling, taking expectations in (6) yields

$$\begin{aligned} E_{S_n}[\hat{\varepsilon}] &= \frac{m_0}{m}E_{S_n}[\hat{\varepsilon}^0] + \frac{m_1}{m}E_{S_n}[\hat{\varepsilon}^1] \\ &= \frac{m_0}{m}E_{S_n}[\varepsilon_n^0] + \frac{m_1}{m}E_{S_n}[\varepsilon_n^1], \end{aligned} \tag{8}$$

because the ratio $\frac{m_0}{m}$ is fixed. Hence, $\hat{\varepsilon}$ is not unbiased. The bias depends on the difference between c and $\frac{m_0}{m}$. If c is known, then the holdout estimator can be redefined as

$$\hat{\varepsilon}_c = c\hat{\varepsilon}^0 + (1-c)\hat{\varepsilon}^1, \tag{9}$$

for both random and separate sampling. In both cases it is unbiased: taking expectations on the right-hand side of (9) yields the right-hand

Table 2. Real datasets used in this article

Dataset	Dataset type	Feature|Sample size
Yeoh *et al.*, 2002	Pediatric ALL	5077|149/99
Valk *et al.*, 2004	AML	22215|116/157
Zhan *et al.*, 2006	Multiple myeloma	54613|156/78
Desmedt *et al.*, 2007	Breast cancer	22215|98/77

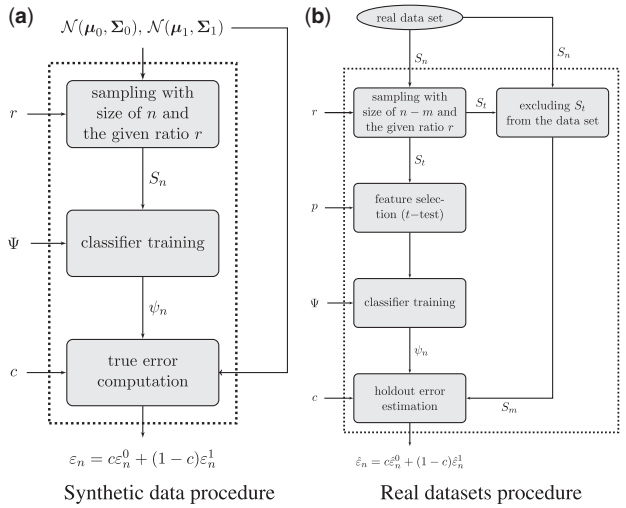

Fig. 2. Flowcharts of the processes implemented for the synthetic and real dataset examples. The dashed boxes show one iteration of the MC simulation, repeated to find an approximation for the expected true error, i.e. $E_{S_n}[\varepsilon_n(c)|r]$, and the expected holdout error estimation, i.e. $E[\hat{\varepsilon}_n(c)|r]$, respectively, for the synthetic and real dataset examples

side of (5). (We are unaware of this approach to holdout error estimation in the case of separate sampling having been previously used.) Absent an independent estimate of c, use of (6) for separate sampling is unacceptable because it is biased and the extent of the bias is unknown. We use (9) for our real-data examples.

2.4 Implementation

For a given synthetic model parameter setting, sample size n, ratio r and classification rule Ψ, we approximate the expected true error rate $E_{S_n}[\varepsilon_n(c)|r]$ via Monte-Carlo (MC) simulation. Each repetition of the MC simulation is depicted in Figure 2(a).

The first set of experiments is done using the flowchart in Figure 2(a). In these experiments, the sample size, n, is fixed but the class sample sizes vary according to the sampling ratio r. Samples S_n are generated using the model determined by (μ_0, Σ_0) and (μ_1, Σ_1) described in Table 1 in accordance with r and n. Assuming a classification rule Ψ, a classifier is trained. The last stage in Figure 2(a) is the true error computation for the designed classifiers, which is also done via MC with 10 000 repetitions. The whole procedure is repeated 5000 times.

The real datasets in Table 2 are sufficiently large to be divided for training and testing and we use holdout error estimation, as previously described for separate sampling. The procedure is graphically illustrated in Figure 2(b).

Fixing the total sample size n, assuming different values for the parameter r, we choose $n_0 = \lceil r(n-m) \rceil$ points from class 0 and $n_1 = (n-m) - n_0$ from class 1, where $\lceil a \rceil$ is the smallest integer greater than or equal to a. The remaining data points are used for holdout estimation. The expected holdout estimate, $E[\hat{\varepsilon}_n(c)|r]$, is computed via MC approximation. The process is repeated 3000 times.

3 RESULTS AND DISCUSSION

The full set of results appears in the Supplementary Material. Herein, we provide some results covering a variety of cases. We show results of four classification rules: 3NN, 5NN, L-SVM and RBF-SVM. Results for the synthetic examples with dimension 15 are shown. Also, we only provide the results for $n = 100$ and $n = 80$, for the synthetic and real datasets, respectively. For the real datasets, a two-sample t-test is used to reduce the dimensions in Table 2 to $D = 15$.

3.1 Expected true error

The expected true errors for synthetic data with common covariance matrix are given in Figure 3(a)-(d), where each plot gives the expected true error versus the parameter r for different class prior probabilities, i.e. $c \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$. Similar behavior is observed, regardless of classification rule. There is, however, different sensitivsities of different rules to the sampling ratio r. Results for the second model with different covariance matrices are shown in Figure 3(e)–(h). In this case, there is more varied behavior among the classification rules. Note the point in each figure where the curves cross. This point corresponds to the minimax solution and will be discussed in detail when we analyze the properties of the curves in a later subsection. For equal covariance matrices, the point is close to 0.5 for all the classification rules; however, it can be far from 0.5 for unequal covarinace matrices, depending on the classification rule.

Figure 3(d)–(p) show results for two real datasets, where the performance is assessed via holdout error estimation. Each plot includes the expected holdout error estimate versus r for five values of c. In contrast to Figure 3(a)–(h), the curves in Figure 3(d)–(p) are not smooth, which is a result of discrete error estimation. Nonetheless, there still is a crossing point. The solid vertical lines in Figure 3(i)–(p) are fixed on the initial sampling ratios.

3.2 Properties of the error curves

The most obvious characteristic of the error curves is that in each figure they appear to cross at a single value of r. We will now examine this phenomenon. We must be careful because the figures show continuous curves but r is a discrete variable. Hence, we will have to carefully define what it means to 'cross'.

According to the standard definition, a classification rule is said to be 'smart' if the expected value of the error is monotonically decreasing as a function of sample size for all feature-label distributions (Devroye, 1996). This is in accord with the intuition (not always correct) that more data cannot hurt classifier design. We adapt the notion of smartness to the present circumstances by defining a classification rule to be 'class-wise smart' relative to a family of feature-label distributions $f_c(\mathbf{x}, y) = cf(\mathbf{x}|0) + (1-c)f(\mathbf{x}|1), c \in [0,1]$, if, for all $c \in [0,1]$ and $r_2 > r_1$, $E[\varepsilon_n^0|r_2] \leq E[\varepsilon_n^0|r_1]$ and $E[\varepsilon_n^1|r_2] \geq E[\varepsilon_n^1|r_1]$. Intuitively, $r_2 > r_1$ means that there are more data available from class 0 when designing the classifier when conditioning on r_2 than when conditioning on r_1, so that one would intuitively expect that the class-0 error when conditioning on r_2 is not greater than when conditioning on r_1, which is what is stated by the first inequality. The situation reverses relative to the class-1 error and that is what is stated by the second inequality. A classification rule is 'strictly class-wise smart' if $r_2 > r_1$ implies $E[\varepsilon_n^0|r_2] < E[\varepsilon_n^0|r_1]$ and $E[\varepsilon_n^1|r_2] > E[\varepsilon_n^1|r_1]$.

In Figure 3 we observe that, if $c_2 > c_1$, then for sufficiently small r, $E[\varepsilon_n(c_2)|r] > E[\varepsilon_n(c_1)|r]$, where the notation $E[\varepsilon_n(c)|r]$

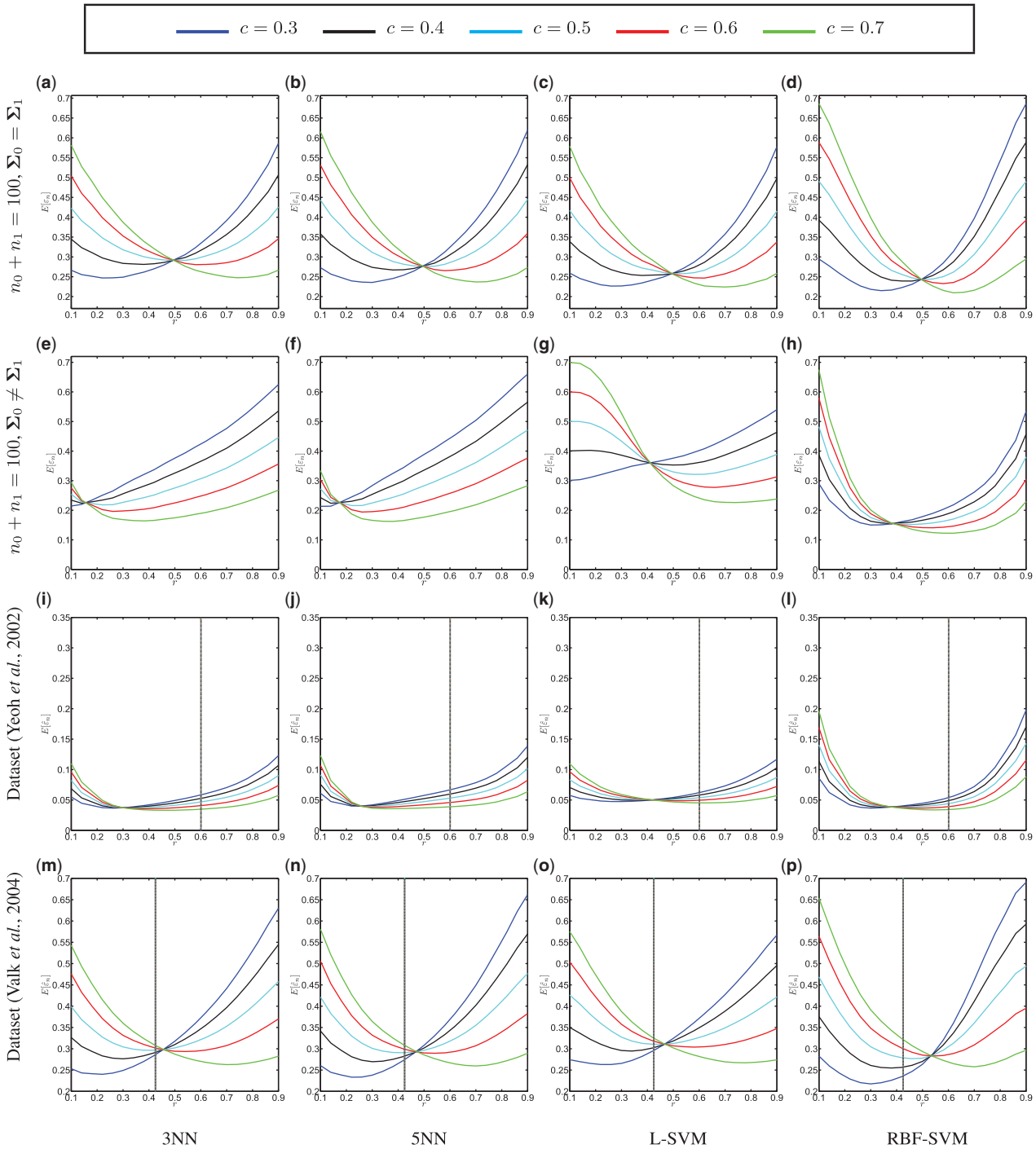

Fig. 3. The first and the second rows show the expected true error rates of four classification rules when covariance matrices are identical and unequal, respectively. In these plots $n_0 + n_1 = 100$ is fixed, where n_0 and n_1 are chosen according to the ratio r. The third and the fourth rows show the expected holdout error estimate of the datasets (Yeoh *et al.*, 2002) and (Valk *et al.*, 2004), respectively, where $n_0 = \lceil 80r \rceil$ and $n_1 = 80 - n_0$ points are randomly selected from classes 0 and 1, respectively, as the training set. The rest of points are held out for error estimation computed via (9)

indicates that the error is with respect to $f_c(\mathbf{x}, y)$. To obtain intuition behind this phenomenon, decompose the error into class-wise errors:

$$E[\varepsilon_n(c)|r] = cE[\varepsilon_n^0|r] + (1-c)E[\varepsilon_n^1|r]. \quad (10)$$

For small r, the preponderance of data for classifier design is in class 1 with very little data in class 0. Hence, the class-0 error tends to be greater than the class-1 error, so that

$$E[\varepsilon_n(c_2)|r] - E[\varepsilon_n(c_1)|r] = (c_2 - c_1)(E[\varepsilon_n^0|r] - E[\varepsilon_n^1|r]) > 0. \quad (11)$$

Analogously, for sufficient large r,

$$E[\varepsilon_n(c_2)|r] - E[\varepsilon_n(c_1)|r] < 0. \quad (12)$$

We shall assume that whatever classification rule and feature-label distribution we are considering, (11) and (12) hold for sufficiently small and sufficiently large r, respectively.

The next lemma, whose proof is in the Supplementary Material, states a fundamental property of the error curves.

LEMMA 3.2.1 *If a classification rule is strictly class-wise smart relative to the family* $\{f_c(\mathbf{x}, y)\}$, *then, for* $c_2 > c_1$, $E[\varepsilon_n(c_2)|r] - E[\varepsilon_n(c_1)|r]$ *is a strictly decreasing function of* r.

If we only assume class-wise smart, then $E[\varepsilon_n(c_2)|r] - E[\varepsilon_n(c_1)|r]$ would only be sure to be a decreasing function of r.

The next lemma, whose proof is in the Supplementary Material, shows that constraining c results in a corresponding constraining of the expected error.

LEMMA 3.2.2 *Suppose* $c_1 < c_2 < c_3$. *If* $E[\varepsilon_n(c_3)|r] \geq E[\varepsilon_n(c_1)|r]$, *then* $E[\varepsilon_n(c_3)|r] \geq E[\varepsilon_n(c_2)|r] \geq E[\varepsilon_n(c_1)|r]$. *If* $E[\varepsilon_n(c_3)|r] \leq E[\varepsilon_n(c_1)|r]$, *then* $E[\varepsilon_n(c_3)|r] \leq E[\varepsilon_n(c_2)|r] \leq E[\varepsilon_n(c_1)|r]$.

To ease notation, we will say that $E[\varepsilon_n(c_2)|r]$ is *between* $E[\varepsilon_n(c_1)|r]$ and $E[\varepsilon_n(c_3)|r]$ if either $E[\varepsilon_n(c_3)|r] \geq E[\varepsilon_n(c_2)|r] \geq E[\varepsilon_n(c_1)|r]$ or $E[\varepsilon_n(c_3)|r] \leq E[\varepsilon_n(c_2)|r] \leq E[\varepsilon_n(c_1)|r]$.

The salient proposition concerning the error curves involves a strictly decreasing function $g(r)$ of r that is positive for sufficiently small values of r and negative for sufficiently large values of r. Since r is a discrete variable in $(0, 1)$, we have a sequence of values $0 < r_1 < \ldots < r_{n-1} < 1$. If g were continuous, then there would exist a unique value r^* such that $g(r^*) = 0$, $g(r) > 0$ for $r < r^*$, and $g(r) < 0$ for $r > r^*$. But since g is discrete, this basic proposition is slightly altered. Rather, there are two possibilities: (i) there exists a unique value $r^* = r_j$ for some value j such that $g(r^*) = 0$, $g(r) > 0$ for $r < r^*$, and $g(r) < 0$ for $r > r^*$ or (ii) there is a unique value r_j such that $g(r) > 0$ for $r \leq r_j$ and $g(r) < 0$ for $r \geq r_{j+1}$. In the second case, we select a point $r^* \in (r_j, r_{j+1})$, say, the mid-point, and then we have $g(r) > 0$ for $r < r^*$ and $g(r) < 0$ for $r > r^*$, as in the first case. In the next theorem, whose proof is in the Supplementary Material, we will be interested in a 'unique' point $r^* \in (0, 1)$. For the second case, we interpret this to mean that there is a unique interval (r_j, r_{j+1}) and r^* is the selected point in that interval.

Given the preceding discrete interpretation, we shall say that a function $p(r)$ 'crosses' function $q(r)$ at r^* if $p(r) \geq q(r)$ for $r < r^*$ and if $p(r) \leq q(r)$ for $r > r^*$.

THEOREM 3.2.3 *If a classification rule is strictly class-wise smart relative to* $\{f_c(\mathbf{x}, y)\}$, *then there exists a unique point* r^* *such that for any* $c_2 > c_1$, $E[\varepsilon_n(c_2)|r]$ *crosses* $E[\varepsilon_n(c_1)|r]$ *at* r^*.

This is precisely the theorem we want because it means that all error curves cross at r^*.

In the error curves of Figure 3, we observe that r^* provides a minimax value; that is, $r^{\mathrm{mm}} = r^*$ yields the minimum value of $E[\varepsilon_n(c)|r]$ when taking the maximum error over all values of $E[\varepsilon_n(c)|r]$ for $r \in (0, 1)$:

$$r^{\mathrm{mm}} = \arg\min_r \max_c E[\varepsilon_n(c)|r], \quad (13)$$

where we must keep in mind that $r \in \mathcal{R} = \{r_1, \ldots, r_{n-1}\}$ is a discrete variable. The next theorem, proven in the Supplementary Material, formalizes this observation.

THEOREM 3.2.4 *Consider a classification rule that is strictly class-wise smart relative to* $\{f_c(\mathbf{x}, y)\}$ *and let* r^{mm} *be the minimax value defined by (13). If Theorem 3.2.3 yields a unique point* $r^* = r_j$, *then* $r^{\mathrm{mm}} = r_j$; *otherwise, if Theorem 3.2.3 yields an interval* (r_j, r_{j+1}), *then either* r_j *or* r_{j+1} *is the minimax ratio, determined by*

$$r^{\mathrm{mm}} = \begin{cases} r_j & \text{if } E[\varepsilon_n^0|r_j] \leq E[\varepsilon_n^1|r_{j+1}] \\ r_{j+1} & \text{if } E[\varepsilon_n^0|r_j] > E[\varepsilon_n^1|r_{j+1}] \end{cases}. \quad (14)$$

3.3 Practical implications of the error curves

Recall the practical implications we drew regarding Figure 1 in the Section 1: (i) if c is known, then do separate sampling with $n_0 = cn$; where the equal sign means 'as close to cn as possible'; (ii) if $c \approx c'$, then for small n do separate sampling with $n_0 = c'n$; (iii) if one has no idea regarding the value of c, then sampling must be random. Looking at the curves in Figure 3 (and similar figures in the Supplementary Material), we see that the curve for c has its minimum value at $r = c$ or $r = c' \approx c$ and, in the latter case, $E[\varepsilon_n(c)|r] \approx E[\varepsilon_n(c')|r]$. Hence, the first two recommendations hold for the other classification rules examined.

Going beyond the case where c is known or approximately known, consider the third implication, where one has no good idea concerning the value of c. Then the minimax r^{mm} is an option. Its suitability depends upon the classification rule and feature-label distribution. As we can see from Figure 3, except for extreme values of c, $E[\varepsilon_n(c)|r^{\mathrm{mm}}]$ tends not to be too much greater than $E[\varepsilon_n(c)|c]$. Of course, there is a practical problem: while we may well know the classification rule, we will not know the feature-label distribution.

3.4 Algorithm to approximate r^{mm}

Algorithm 1 provides an iterative algorithm for approximating r^{mm} when the feature-label distribution is unknown. The procedure is an empirical illustration of Theorem 3.2.4, which requires $E[\varepsilon_n(c)|r]$, which now needs to be approximated to approximate r^{mm}. Algorithm 1 uses holdout error estimation. The expectation of this error estimate is taken by iterative random sampling from the dataset. Here we give a brief overview of the algorithm.

The inputs to the algorithm are: dataset denoted by S_N, classification rule, number of points to be held out for error estimation from classes 0 and 1, denoted, respectively, by n^0_{test}, and n^1_{test} and number of iterations, MaxIters, for computing the expected holdout error estimate. The maximum number of points after holding out test sample points is $N_{\mathrm{new}} = N - (n^0_{\mathrm{test}} + n^1_{\mathrm{test}})$, denoted class-wise as N^0_{new} and N^1_{new}. The algorithm searches over possible values for r, from 0 to 1, until a stopping criterion is met. Suppose we fix the total sample size n. Then, considering the first extreme case, $r = 0$, we need to have at least n points in class 1 to draw sample points from, randomly, i.e. $N^1_{\mathrm{new}} \geq n$. On the other hand, when $r = 1$, we similarly should have $N^0_{\mathrm{new}} \geq n$. Hence, we should have $n \leq \min\{N^0_{\mathrm{new}}, N^1_{\mathrm{new}}\}$, whereby we set $n = \min\{N^0_{\mathrm{new}}, N^1_{\mathrm{new}}\}$.

The algorithm's search criterion is based on Theorem 3.2.4: in a 'while loop' over an increasing sequence of the ratios r, the algorithm computes the estimated slope of the expected error (as a function of c), this being $\text{slope}_{\text{new}} = E[\hat{\varepsilon}_n^0|r] - E[\hat{\varepsilon}_n^1|r]$ (line 22 of the algorithm), obtained by plugging the error estimates (lines 7–21 of the algorithm) into the unknown slope formula $E[\varepsilon_n^0|r] - E[\varepsilon_n^1|r]$. Because the classification rule is strictly class-wise smart, for sufficiently small r, the slope is positive, and it becomes negative for sufficiently large r (refer to Supplementary Material file for further explanation). Once a point is reached at which the sign of the slope becomes non-positive, the 'while loop' stops increasing r. Thereafter, the three different possibilities given by Theorem 3.2.4 are checked, in lines 26–32, and finally a single r^{mm} is returned. Although the returned minimax ratio is only computed for sample size n defined above, the class-sizes can still be conservatively adjusted per r^{mm} in the dataset S_N because, for a ratio given by the algorithm, if one increases the sample size, then in the worst case the error is as large as the minimax value returned by the algorithm.

Algorithm 1 Iterative algorithm to approximate r^{mm} (an implementation of Theorem 3.2.4 using an estimate of the expected error estimate)

1: **Input:** Dataset S_N, Classification rule Ψ, n_{test}^0, n_{test}^1, MaxIters
2: **Output:** r^{mm}
3: **Define:** $N_{\text{new}}^0 = N^0 - n_{\text{test}}^0$, $N_{\text{new}}^1 = N^1 - n_{\text{test}}^1$
$n = \min\{N_{\text{new}}^0, N_{\text{new}}^1\}$
4: **Initialize:** $j = 0, r = 0, \text{slope}_{\text{new}} = 1, \text{sign} = 1$
5: **while** sign > 0 **do**
6: Set: $a \leftarrow \hat{E}[\hat{\varepsilon}_n^0|r]$, $\text{slope}_{\text{old}} \leftarrow \text{slope}_{\text{new}}$, $r \leftarrow \frac{j}{n}$
7: Reset: $\hat{E}[\hat{\varepsilon}_n^0|r] = 0, \hat{E}[\hat{\varepsilon}_n^1|r] = 0$
8: **for** $i = 1$ **to** MaxIters **do**
9: $S_{n_{\text{test}}}^{\text{test},0} \leftarrow n_{\text{test}}^0$ randomly drawn points from S_N^0
10: $S_{n_{\text{test}}}^{\text{test},1} \leftarrow n_{\text{test}}^1$ randomly drawn points from S_N^1
11: $S_{n_{\text{test}}}^{\text{test}} \leftarrow S_{n_{\text{test}}}^{\text{test},0} \cup S_{n_{\text{test}}}^{\text{test},1}$
12: $S_{N_{\text{new}}}^0 \leftarrow S_N^0 \backslash S_{n_{\text{test}}}^{\text{test},0}$, $S_{N_{\text{new}}}^1 \leftarrow S_N^1 \backslash S_{n_{\text{test}}}^{\text{test},1}$
13: $S_{N_{\text{new}}} \leftarrow S_{N_{\text{new}}}^0 \cup S_{N_{\text{new}}}^1$
14: $S_n^0 \leftarrow rn$ randomly drawn points from $S_{N_{\text{new}}}^0$
15: $S_n^1 \leftarrow (1-r)n$ randomly drawn points from $S_{N_{\text{new}}}^1$
16: $S_n \leftarrow S_n^0 \cup S_n^1$
17: $\psi_n \leftarrow \Psi(S_n)$
18: Compute $\hat{\varepsilon}_n^0, \hat{\varepsilon}_n^1$ of ψ_n using $S_{n_{\text{test}}}^{\text{test}}$
19: Add $\hat{\varepsilon}_n^0$, and $\hat{\varepsilon}_n^1$ to $\hat{E}[\hat{\varepsilon}_n^0|r]$ and $\hat{E}[\hat{\varepsilon}_n^1|r]$, respectively
20: **end for**
21: $\hat{E}[\hat{\varepsilon}_n^0|r] \leftarrow \frac{\hat{E}[\hat{\varepsilon}_n^0|r]}{\text{MaxIters}}$, $\hat{E}[\hat{\varepsilon}_n^1|r] \leftarrow \frac{\hat{E}[\hat{\varepsilon}_n^1|r]}{\text{MaxIters}}$
22: $\text{slope}_{\text{new}} \leftarrow \hat{E}[\hat{\varepsilon}_n^0|r] - \hat{E}[\hat{\varepsilon}_n^1|r]$
23: $\text{sign} \leftarrow \text{slope}_{\text{new}}\text{slope}_{\text{old}}$
24: $j \leftarrow j + 1$
25: **end while**
26: **if** sign $= 0$ **then**
27: $r^{\text{mm}} \leftarrow r$
28: **else if** $a < \hat{E}[\hat{\varepsilon}_n^1|r]$ **then**
29: $r^{\text{mm}} \leftarrow r - \frac{1}{n}$
30: **else**
31: $r^{\text{mm}} \leftarrow r$
32: **end if**
33: return r^{mm}

3.5 Adjusting sample sizes

Consider the common situation in which n_0 and n_1 have been determined beforehand, but suppose one knows c. The curves of Figure 3 still apply but we are not free to choose n_0 and n_1, so that we cannot choose $n_0 = cn$. Nevertheless, we desire the training data to be apportioned according to c and we want to use as much data as is possible. These conditions mean that for training we want class sample sizes m_0 and m_1 such that $m = m_0 + m_1$ is maximized given the constraints $m_0 = cm$, $m_1 = (1-c)m$, $m_0 \leq n_0$, and $m_1 \leq n_1$. The solution is to let $m = \left\lceil \min\{\frac{n_0}{c}, \frac{n_1}{1-c}\} \right\rceil$.

To see the effect of adjusting sample sizes, we consider the difference, $\Delta(r,c) = E[\varepsilon_n(c)|r] - E[\varepsilon_m(c)|c]$, between the expected true errors of two cases, n being the original sample size and m the adjusted sample size. When the sampling ratio is r and the true prior probability is c, $\Delta(r,c)$ can be interpreted as the penalty incurred. Figure 4 shows $\Delta(r,c)$ for L-SVM and RBF-SVM for the equal covariance model described in Table 1. The result for the case with unequal covariance matrices can be found in the Supplementary Material. The two parameters r and c take values from 0.06 to 0.94 with the step size of 0.04. As expected, as $|r-c|$ increases, $\Delta(r,c)$ significantly increases. When $r \approx c$, $\Delta(r,c) \approx 0$, which is always the minimum. The figure shows that except when r is very close to c, $\Delta(r,c) > 0$, meaning that, even though $m < n$, a correct sampling ratio more than compensates for the loss of data due to subsampling.

3.6 Population-based minimax theory

The minimax value in the error curves of Figure 3 depends on the sampling distribution and results from the fact that $E[\varepsilon_n(c)|r]$ is minimized over c for a single value r^*. In Anderson (1951), a population-based minimax approach was taken to arrive at a 'best' choice for $\hat{c}$ in (1) in the Gaussian model with common covariance matrix under separate sampling. Here we extend the population-based mimimax approach to arrive at much more general solution than that given by Anderson. It is based upon the fact that the Bayes classifier can be determined via a discriminant involving the class-conditional densities. Anderson also utilized the Bayes classifier in his analysis but he restricted it to the Gaussian model with common covariance matrix, in which case the Bayes classifier is given by LDA using the actual parameters rather than their estimates as in (1).

Given the class-conditional distributions and prior probabilities, the Bayes classifier, ψ_{Bayes}, is determined by the discriminant

$$D_{\text{Bayes}}(\mathbf{x}) = \log\frac{f(\mathbf{x}|0)}{f(\mathbf{x}|1)} - \log\frac{1-c}{c}, \tag{15}$$

where $\psi_{\text{Bayes}}(\mathbf{x}) = 1$ if $D_{\text{Bayes}}(\mathbf{x}) \leq 0$ and $\psi_{\text{Bayes}}(\mathbf{x}) = 0$ if $D_{\text{Bayes}}(\mathbf{x}) > 0$. The regions assigned to the two classes are $R_1 = \{\mathbf{x} : D_{\text{Bayes}}(\mathbf{x}) \leq 0\}$ and $R_0 = \{\mathbf{x} : D_{\text{Bayes}}(\mathbf{x}) > 0\}$. If c is unknown and replaced by $\hat{c}$, then the discriminant becomes

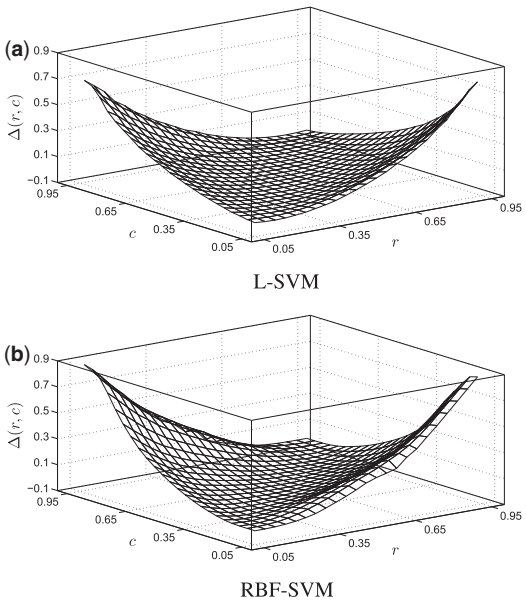

Fig. 4. The parameter $\Delta(r,c)$ for classification rules, L-SVM and RBF-SVM trained on the sample data with size $n=100$ from the common covariance matrix model described in Table 2

$$D_{\text{prior}}(\mathbf{x}) = \log\frac{f(\mathbf{x}|0)}{f(\mathbf{x}|1)} - \log\frac{1-\hat{c}}{\hat{c}}, \tag{16}$$

which defines the classifier $\psi_{\hat{c}}$, with corresponding class regions $R_1(\hat{c}) = \{\mathbf{x} : D_{\text{prior}}(\mathbf{x}) \le 0\}$ and $R_0(\hat{c}) = \{\mathbf{x} : D_{\text{prior}}(\mathbf{x}) > 0\}$. The error associated with $\psi_{\hat{c}}$ is

$$\varepsilon(\hat{c}, c) = c\varepsilon^0(\hat{c}) + (1-c)\varepsilon^1(\hat{c}), \tag{17}$$

where

$$\varepsilon^y(\hat{c}) = \int_{R_{1-y}(\hat{c})} f(\mathbf{x}|y)d\mathbf{x}, \tag{18}$$

for $y \in \{0,1\}$. $R_1(\hat{c})$ and $R_0(\hat{c})$ are strictly increasing and decreasing, respectively, for increasing values of $\log\frac{1-\hat{c}}{\hat{c}}$. Hence, if the conditional densities are strictly positive, then $\varepsilon^1(\hat{c})$ and $\varepsilon^0(\hat{c})$ are strictly decreasing and increasing, respectively, for increasing values of $\log\frac{1-\hat{c}}{\hat{c}}$.

The minimax choice selects the value of $\hat{c}$ that yields the minimum value of the error $\varepsilon(\hat{c}, c)$ when taking the maximum error over all values of $c \in (0,1]$:

$$\hat{c}^{\text{mm}} = \arg\min_{\hat{c}} \max_{c} \varepsilon(\hat{c}, c). \tag{19}$$

We state a lemma and a theorem, whose proofs are given in the Supplementary Material that can be used to find minimax solutions.

LEMMA 3.6.1 *If the class-conditional distributions are strictly positive and $\varepsilon^y(\hat{c}), y=0,1$, is a continuous function of $\hat{c}$, then there exists a unique point $\hat{c}^{\text{mm}}$ such that $\varepsilon^0(\hat{c}^{\text{mm}}) = \varepsilon^1(\hat{c}^{\text{mm}})$ and this point corresponds to the minimax solution defined in* (19).

THEOREM 3.6.2 *If the class-conditional distributions are strictly positive and $\varepsilon^y(\hat{c}), y=0,1$, is a continuous function of $\hat{c}$, then the minimax solution for the discriminant in* (16) *is the value of c that gives rise to the maximum Bayes error for the discrimination problem of* (15).

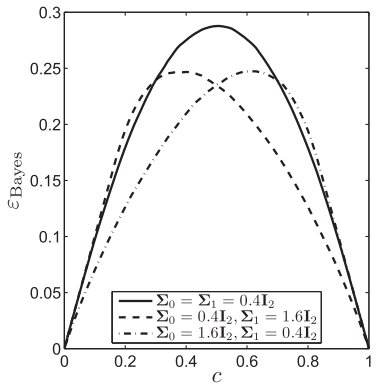

Fig. 5. Bayes error as a function of class prior probability c for different settings with multivariate Gaussian distributions with $\mu_0 = (0.3, 0.3)$ and $\mu_1 = (0.8, 0.8)$

To apply the lemma to the Gaussian model with common covariance matrix, note that the discriminant takes the form

$$D_{\text{prior}}(\mathbf{x}) = \left(\mathbf{x} - \frac{\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1}{2}\right)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) - \ln\frac{1-\hat{c}}{\hat{c}} \tag{20}$$

and the error of the classifier induced by D_{prior} is given by

$$\begin{aligned}\varepsilon(\hat{c}, c) = c\underbrace{\phi\left(-\frac{\frac{(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^T}{2}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1) - \log\frac{1-\hat{c}}{\hat{c}}}{\sqrt{(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)}}\right)}_{\varepsilon^0(\hat{c})} \\ + (1-c)\underbrace{\phi\left(\frac{\frac{(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^T}{2}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1) - \log\frac{1-\hat{c}}{\hat{c}}}{\sqrt{(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)}}\right)}_{\varepsilon^1(\hat{c})},\end{aligned} \tag{21}$$

where ϕ is the standard normal cumulative distribution function. It is immediate that for $\hat{c} = 0.5$, $\varepsilon^0(\hat{c}) = \varepsilon^1(\hat{c})$. Hence, $\hat{c}^{\text{mm}} = 0.5$, which is precisely Anderson's result for this special case.

To illustrate the effect of the underlying feature-label distribution on $\hat{c}^{\text{mm}}$, we consider a situation similar to that used for Figure 1, except that we allow unequal covariance matrices. Figure 5 contains Bayes-error curves as functions of c for different covariance models. It shows that, except for a common covariance matrix, $\hat{c}^{\text{mm}} \neq 0.5$, $\hat{c}^{\text{mm}}$ being the value of c at which the curve attains its maximum. The curve for equal covariance matrices is constructed analytically; for other values, MC simulation is employed.

Obviously, if $\hat{c}^{\text{mm}} = c$, then the minimax value will perform well. But what happens when $\hat{c}^{\text{mm}} \neq c$? In fact, the minimax value can work well so long at it is close to the true value, how close depending on the particulars of the problem. For a Gaussian model with common covariance matrix, as used for Figure 1, we consider LDA with random sampling under three scenarios: (i) known c, (ii) minimax $\hat{c}^{\text{mm}}$ and (iii) the maximum-likelihood estimate $\hat{c}^{\text{ml}} = \frac{n_0}{n}$. Figure 6 shows the expected errors (MC estimates) as a function of c for $n=20$ and 80. In all cases, known c is the best. When the sample is small, $\hat{c}^{\text{mm}}$ outperforms

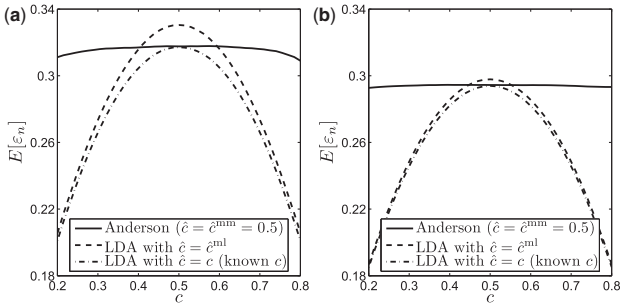

Fig. 6. Expected true error for different scenarios with random sampling (fixed $n_0 + n_1$ with random n_0 and n_1) for the same model as used in Fig. 1

$\hat{c}^{\mathrm{ml}}$ for a fairly wide range of c, but this advantage disappears rapidly as n grows. The reason for this behavior is the difficulty of estimating c by $\hat{c}^{\mathrm{ml}}$ for small samples. All curves show that when c is large or small the minimax solution gives poor results. Let us close this section by noting that the Bayes classifier is intrinsic to the feature-label distribution and, since the minimax choice depends only on the form of the Bayes classifier, it is independent of any particular classification rule.

3.7 Concluding remarks

We have shown, via simulations on both synthetic and real examples, that separate sampling with an inappropriate sampling ratio can significantly degrade classification accuracy for classification rules that do not use an explicit estimate of the prior probability. We have demonstrated some fundamental properties of the expected-error curves, developed the minimax sample-based theory for those curves, proposed an algorithm to approximate the minimax value in practice and extended the classical Anderson minimax theory for prior probabilities. We have provided heuristics on how to proceed when the prior probability is known (or known within a small range) and we have proposed a subsampling methodology to implement these heuristics when the class sample sizes are predetermined. Given the ubiquity of separate sampling in biomedicine, it would behoove the medical community to record incidence rates of patient sub-types (population statistics), so that very accurate estimates of class prior probabilities would be available. While this would certainly incur some cost, that cost would be minuscule compared to the costs incurred by the irreproducibility of classification studies.

Conflict of Interest: none declared.

REFERENCES

Anderson,T.W. (1951) Classification by multivariate analysis. *Psychometrika*, **16**, 31–50.

Chang,C.-C. and Lin,C.-J. (2011) LIBSVM: A library for support vector machines. *ACM Transact. Intell. Syst. Technol.*, **2**, 1–27.

Desmedt,C. *et al.* (2007) Strong time dependence of the 76-gene prognostic signature for node-negative breast cancer patients in the transbig multicenter independent validation series. *Clin. Cancer Res.*, **13**, 3207–3214.

Devroye,L. (1996) *A Probabilistic Theory of Pattern Recognition*. Vol. 31, Springer, New York.

Doughtery,E.R. *et al.* (2007) Validation of computational methods in genomics. *Curr. Genom.*, **8**, 1.

Duda,R.O. *et al.* (2001) *Pattern Classification*. John Wiley, New York.

Hua,J. *et al.* (2005) Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, **21**, 1509–1515.

Mortazavi,A. *et al.* (2008) Mapping and quantifying mammalian transcriptomes by rna-seq. *Nat. Methods*, **5**, 621–628.

Ray,T. (2011) FDA's Woodcock says personalized drug development entering 'long slog' phase. *Pharmacogen. Rep.*, http://www.genomeweb.com/mdx/fdaswoodcock-says-personalized-drug-development-entering-long-slog-phase (26 October 2011, date last accessed).

Shmulevich,I. and Dougherty,E.R. (2007) *Genomic Signal Processing (Princeton Series in Applied Mathematics)*. Princeton University Press, Princeton, New Jersey.

Valk,P.J. *et al.* (2004) Prognostically useful gene-expression profiles in acute myeloid leukemia. *New England J. Med.*, **350**, 1617–1628.

Wang,Z. *et al.* (2009) Rna-seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.*, **10**, 57–63.

Yeoh,E.-J. *et al.* (2002) Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer cell*, **1**, 133–144.

Yousefi,M.R. and Dougherty,E.R. (2012) Performance reproducibility index for classification. *Bioinformatics*, **28**, 2824–2833.

Zhan,F. *et al.* (2006) The molecular classification of multiple myeloma. *Blood*, **108**, 2020–2028.