Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

Scienze Statistiche

Ciclo XXX°

Settore Concorsuale: 13/D1 - Statistica

Settore Scientifico Disciplinare: SECS – S/01

HIGH-DIMENSIONAL AND ONE-CLASS CLASSIFICATION

Presentata da: Francesca Fortunato

Coordinatore Dottorato **Supervisore**

Prof. ssa Alessandra Luati **Prof.ssa Angela Montanari**

Esame finale anno 2018

Abstract

When dealing with high-dimensional data and, in particular, when the number of attributes p is large comparatively to the sample size n, several classification methods cannot be applied. Fisher's linear discriminant rule or the quadratic discriminant one are unfeasible, as the inverse of the involved covariance matrices cannot be computed.

A recent approach to overcome this problem is based on Random Projections (RPs), which have emerged as a powerful method for dimensionality reduction. In 2017, Cannings and Samworth introduced the RP method in the ensemble context to extend to the high-dimensional domain classification methods originally designed for low-dimensional data. Although the RP ensemble classifier allows improving classification accuracy, it may still include redundant information. Moreover, differently from other ensemble classifiers (e.g. Random Forest), it does not provide any insight on the actual classification importance of the input features. To account for these aspects, in the first part of this thesis, we investigate two new directions of the RP ensemble classifier. Firstly, combining the original idea of using the Multiplicative Binomial distribution as the reference model to describe and predict the ensemble accuracy and an important result on such distribution, we introduce a stepwise strategy for post-pruning (called Ensemble Selection Algorithm). Secondly, we propose a criterion (called Variable Importance in Projection) that uses the feature coefficients in the best discriminant projections to measure the variable importance in classification.

In the second part, we faced the new challenges posed by the high-dimensional data in a recently emerging classification context: one-class classification. This is a special classification task, where only one class is fully known (the target class), while the information on the others is completely missing. In particular, we address this task by using Gini's transvariation probability as a measure of typicality, aimed at identifying the best boundary around the target class.

Contents

Chapter 1

Introduction

High-dimensional data arise when the number of observed variables, p, is much larger than the sample size, n. Image processing, information retrieval in text documents, food authentication studies are only a few examples of the applications in which data of that kind have to be analyzed. In those contexts, standard statistical methods cannot be applied, as the matrices involved in the computations are, in general, not full rank and, thus, cannot be inverted. A solution to this problem, which has attracted large attention in the statistical literature, suggests to impose a sparse structure on the estimated vector parameters by the introduction of an L_1 penalty on their norm. Lasso-based approaches to regression, classification and dimension reduction methods have been populating the statistical literature since Tibshirani's seminal paper in 1996 [115]. See Buhlmann, van de Geer [18] and Hastie, Tibshirani, Wainwright [55] for detailed references.

A different approach is based on the recourse to Random Projections (RPs), which have recently emerged as a powerful method for dimensionality reduction. Theoretical results indicate that this method preserves distances quite nicely. The original p-dimensional data is projected onto a d-dimensional ($d \ll p$) subspace through the origin, using a random $d \times p$ matrix A, whose columns have been generated according, for example, to the *Haar* measure (so that they are unit length and orthogonal). Using matrix notation where $X_{p\times n}$ is the original set of n p-dimensional observations, $X^{RP}_{d\times n} = A_{d\times p}X_{d\times n}$ is the projection of the data onto a lower d-dimensional subspace. The key idea of random mapping arises from the Johnson-Lindenstrauss lemma, which states that if points in a vector space are projected onto a randomly selected subspace of suitably high dimension, the distances between the points are approximately preserved. Following this theorem, when p is large compared to n, we may project the data at random into a lower dimensional space and run the

statistical procedure on the projected data, potentially making great computational savings, while achieving comparable or even improved statistical performance.

The idea of combining random projections with ensemble methods has given very nice results in the supervised classification context [20], where the task consists in assigning an object (or a number of objects) to one of two or more groups, on the basis of a sample of labelled training data. In the high-dimensional context, popular methods such as Fisher's linear discriminant rule or the quadratic discriminant one cannot be applied, as the involved covariance matrix cannot be inverted.

In 2017, Cannings and Samworth introduce a general method for high-dimensional classification, based on a careful combination of the results obtained by applying an arbitrary base classifier to random projections of the feature vectors into a lower dimensional space. The random projections are divided into disjoint groups, and, within each group, the projection yielding the smallest estimate of the test error is selected. Then, the Random Projection ensemble classifier aggregates results of applying the base classifier on the selected projections, with a data-driven voting threshold to determine the final assignment. Theoretical results elucidate the effects on performance of increasing the number of projections.

The first part of this thesis presents some new results in the field of random projection ensemble classification.

It is well known that the performance of ensemble classifier methods is strongly driven by the degree of the dependence between the classifiers in the ensemble [16]. Including in the ensemble negatively dependent classifiers can improve the performance, while positively correlated classifiers make the ensemble classifier redundant and, therefore, may worsen its effectiveness. Following that line, many researches have proved that ensemble post-pruning is a relevant strategy for the identification of the ensemble minimizing the misclassification rate.

Even assuming independent random projections, the classifiers in the RP ensemble are not independent, as they are trained on the same data. This implies that the performances of the ensemble cannot be well described by the Binomial model and that a distribution accounting for the Bernoulli variables dependence is required. Among the several solutions proposed in the literature, we have found that the Multiplicative Binomial distribution, introduced by Altham [4] and Lovison [78] is able to provide a better approximation to the ensemble accuracy than the standard Binomial one. We have derived some further theoretical results on the asymptotic distribution of the Multiplicative Binomial and an interesting property showing that the marginal probability of success is larger than the one of the Bernoulli compo-

nents, only if those components are negatively related to each other.
Based on these results, we have developed a stepwise strategy for post-pruning (called Ensemble Selection Algorithm, ESA) involving a pruning function which combines both the accuracy and the dependence between classifiers and accounts for them by using the Multiplicative Binomial model parameters. The performances of this method are tested on both real and simulated data and show that, in many circumstances, the solution proposed sensibly improves the ensemble accuracy, while reducing the ensemble size.
Furthermore, despite of ensemble methods are known to have good predictive performances, they are a sort of black box and no longer allow detecting the most relevant variables for classification purposes. Thus, we have exploited the characteristics of random projections to propose a method that uses the variable coefficients in the best discriminant projections in order to assess variable importance in classification. This method, that we have called VIP (Variable Importance in Projections) has shown very good ability to correctly detect the most relevant featues for classification purposes, while improving the ensemble accuracy.

The second part of the thesis deals with the new challenges posed by the high-dimensional data in a recently emerging classification context, that is one-class classification. This is a special classification problem, where only one class is fully known (called *target* class), while the information on the others is totally vague [111]. In this sense, a very typical example is given by the food authentication issue, where the characteristics of "good" food (i.e. the target class) are known, while those of "counterfeit" food may arise in many and almost unpredictable ways. Misclassification rate is no longer meaningful in this context; the goal instead consists in finding a boundary around the target class so that the probability of labelling as "counterfeit" a unit belonging the this class is minimized.
We have proposed a new one-class classification method based on Gini's transvariation probability as a measure of typicality aimed at identifying the boundary around the target class. Furthermore, we have addressed dimension reduction issues by proposing various strategies; one of them is still based on random projections and exploits a variant of our VIP criterion for variable ranking.

Chapter 2

High-dimensional supervised classification

2.1 Introduction

In the last decades, dramatic advances in data capture, processing power, data transmission and storage have been accomplished. The resulting availability of large amounts of information for each observation gave rise, in many areas of modern sciences, to datasets characterized by a number of features p comparatively larger than the sample size n. Examples of the so called "High-Dimension, Low-Sample Size" (HDLSS, [3]) datasets are very common in a wide range of applications, including genetic studies (DNA microarrays, Deep Sequencing, Micro RNA, CGH -Copy Number Variation, SNPs -Single Nucleotide Polymorphisms, Methalaytion), bioinformatics (fMRI - functional Magnetic Resonance Imaging), neuroimaging (DTI - Diffusion Tensor Imaging, Calcium-Florescence Imaging, EEG & MEG), climatology (spatial and spatio-temporal data), economics and finance (stock markets time series), multimedia data retrieval and social networks (tweets, likes, friendships, interactions, ...).
In such domains, most of the statistical methods originally developed for low dimensional contexts tend to present several limitations, mainly due to the inability of these procedures to both estimate the underlying covariance structure of the HDLSS data and consider their specific characteristics. For these reasons, high-dimensional data have posed both practical and theoretical challenges to standard statistical techniques and have rendered many classification methods impractical [62].

The classification process can be described and performed through a mathemati-

cal function C, called *classifier*. Its traditional task is to assign a new object $\mathbf{x}$ to one of a set of classes by learning from a number of observed attributes related to the object:

$$C : \mathbf{x} \to C(\mathbf{x})$$

In *supervised* classification, the correct output y, i.e. the true class membership of each object $\mathbf{x}$, is known in advance.
In this context, a "classic" supervised classification method is the Linear Discriminant Analysis (LDA), introduced by Fisher in his seminal work [39] of 1936. LDA explicitly attempts to model the difference between the classes by finding the linear combinations of the observed features which best characterize and separate them. Even if it has been originally derived for discrimination purposes, LDA can be also used to address classification issues, i.e. to define a rule for assigning each unit to one of the known groups. In particular, for the two group case, the LDA classifier is given by:

$$\hat{C}_n^{LDA} := \begin{cases} 1 & \text{if } (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T W^{-1} \mathbf{x} > \frac{1}{2}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T W^{-1}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_0) \\ 0 & \text{otherwise} \end{cases}$$

where $\bar{\mathbf{x}}_1$, $\bar{\mathbf{x}}_0$ are the average vectors of class 1 and 0 respectively and W is the within class covariance matrix.

As LDA rests on very strict assumptions which are not always satisfied, many other classification methods have been proposed in the literature, e.g. Quadratic Discriminant Analysis (QDA), kernel discrimination (Knn), Maximum Likelihood Estimation (MLE), decision trees, Random Forest (RF), Neural Networks (NN), Support Vector Machines (SVM) and others.

Any classification algorithm should address two main aims:

- the **accuracy** of the result (in terms of minimization of the misclassification error or, more in general, of a risk function, $P(C(\mathbf{x}) \neq y)$).
- the **generalization** of the result (in terms of predictive performance).

As mentioned before, in presence of high-dimensional data, the use of classification methods originally developed for low dimensional contexts is limited: on one hand, the presence of noisy or irrelevant features can mislead these learning algorithms due to the so-called "*curse of dimensionality*" [7]; on the other, the impossibility to exactly compute some of their discriminant criteria requirements,

makes these procedures unfeasible in high-dimensions. For example, both LDA and QDA need the estimation of the inverse covariance matrix, Σ^{-1}, in order to compute the classification rule; however, being Σ not full-rank when p is larger than n, its inverse cannot be directly calculated.

In order to overcome these problems, a number of proposals to extend Discriminant Analysis to the high-dimensional setting have been put forward. Some of these involve the use of non-sparse classifiers (e.g. [44], [37], [11]); some others imply the positive definite estimation of the within-class covariance matrix, Σ (e.g. [71], [126]); others finally assume sparse (e.g. [38, 116]) decision boundaries or suggest to solve an optimization problem with the addition of an L_1 penalty term to encourage sparsity. In particular, the latter mentioned methods, yielding sparse coefficient vector estimates, perform a process of *variable selection*. Though, it has been demonstrated that dimension reduction procedures which *combine* the input features, rather than *select* a subset of them, are generally more efficient; in fact, feature selection techniques may discard some potentially important variables, e.g. variables that are not predictive if individually considered, could provide significant improvements when taken in conjunction with other features.

Traditionally, variable combination methods involve the projection of the high-dimensional data onto a lower-dimensional subspace so to capture as much data variability as possible. Principal Component Analysis (PCA) is probably the multivariate statistical procedure most broadly used to handle dimension reduction tasks. Although PCA can be successfully used in many applications, its aim does not always coincide with that of a classification task.

A recent approach for dimension reduction is the Random Projection (RP) method. Introduced at the turn of the 21st century [1, 12, 89], the idea is to map at random the original high-dimensional data onto a lower subspace using a matrix with columns of unit length. Specifically, the key point of RPs is that, regardless of the original data dimension, the final solution still preserves almost perfectly the global information. Such result is guaranteed by the Johnson-Lindenstrauss lemma [61]: a subset of n points lying in the Euclidean space of any dimension can be embedded in $d = \mathcal{O}(\log n/2)$ dimensions while approximately preserving the distances between any pair of points.

In [20], Cannings and Samworth introduced the RP method in the context of ensemble classifiers so as to extend to the high-dimensional domain classification methods originally designed for low-dimensional data. The novel idea of the two authors is

to aggregate, using a modified majority voting technique, results of a generic base classifier applied to different training sets, each generated by randomly projecting the feature vector onto a lower-dimensional space (*RP ensemble classifier*).

Although the RP ensemble classifier allows to improve classification accuracy, it may still include redundant information. In addition, differently from other ensemble classifiers (e.g. Random Forest), it does not provide any insight on the actual importance of the input features for classification purposes. In order to account for these aspects, this thesis investigates two new directions of the RP ensemble classifier.

The remainder of this Chapter is organized as follows. Section 2.2 provides an overview of the ensemble methodology and briefly presents the **Random Projection Ensemble classifier**. In Section 2.3, the accuracy of a generic ensemble is described using a novel distribution and some results on such model are given. Sections 2.4 and 2.5 introduce respectively the **ensemble post-pruning** and **variable selection** problems and provide experimental results to illustrate the empirical performances of two new procedures. Conclusions and possible extensions are finally discussed in Section 2.6.

2.2 Ensemble of classifiers

Ensemble classification is a learning paradigm where a finite number of base classifiers are jointly trained and combined (typically through a plurality or majority vote, i.e. the candidate with the majority votes wins) to solve the same problem. This technique is typically used to increase the prediction accuracy in classification beyond the level achieved by any individual classifier.
Generally, an ensemble algorithm is developed in two steps: firstly a collection of base classifiers is trained on the same data (or on some manipulated versions of the same data) and then the individual predictions are combined together.
Ensemble systems usually differ from each other in the number of considered individual classifiers (ensemble size), the procedure used to generate them and the strategy chosen to produce the final decision.

The earliest works on ensembles date back to the late Seventies, when Tukey (1977, [121]) suggested to fit an ensemble of two regression models. Later, in 1979, Dasarathy and Sheela [30] proposed to divide the input space in two or more smaller partitions, where to separately train a single classifier. In 1990, Hansen and Salamon [52] used a plurality consensus scheme to improve the performances of Artificial Neural Networks (ANN). However, the main progress in the field of classifier ensembles was probably achieved in 1995 with Freund and Schapire's seminal paper [42]. The two authors introduced the famous AdaBoost algorithm, the first (and probably still the most used) practical boosting technique[1]. At the same time, in 1996, Breiman [15] laid the foundation of another machine learning ensemble meta-algorithm, called Bagging, *bootstrap aggregating*, whose aim is to improve the classification accuracy by combining predictions from randomly generated training sets.
Since these procedures have been proven to be very effective in solving a wide spectrum of classification problems, research in the ensemble learning context has expanded rapidly over the last couple of decades. As a result, nowadays a vast number of ensemble techniques are available to both enhance the performances of supervised and unsupervised classification methods and improve the quality of clustering algorithms. Some of these techniques center on producing individual learners which disagree in their predictions; in fact, several studies [15, 45, 50, 70, 73, 103] have shown that ensembles of *diverse* base classifiers, i.e. classifiers which return different results with independent errors, achieve better performances than ensembles of identical (and, thus, *redundant*) experts.
According to Ditterich [31], diversity in ensembles can be induced in many different ways:

(i) **manipulating the training set**: each base classifier is trained on a different subset of examples drawn according to a bootstrap scheme (*Bagging*, [15], *Boosting*, [42])[2] or a cross-validation rule (*cross validated committees*, [90]).

(ii) **manipulating the input features**: each base classifier is trained on a different subset of features [26, 122]; this method could be used only if the input

[1]Boosting is a machine learning approach that generates a *strong* classifier in the probably approximately correct (PAC) sense by combining *weak* classifiers.

[2]The difference between these two methods is that while bagging uses an ensemble of independently trained classifiers, boosting creates ensembles by sequentially adding new classifiers in order to mitigate all the previous models lacks.

features are highly correlated.

(iii) **manipulating the output target**: each base classifier is trained on a different partition of the classes labels into two disjoint subsets (*Error-Correcting Output Coding*, [32])

(iv) **injecting randomness**: each base classifier is trained on the same subset of examples, but with different initial weights.

Despite diversity is deemed to be an important factor for the success of ensemble of classifiers, its computation is not straightforward. In fact, although many pairwise measures of diversity are provided by the statistical literature (for an extensive review, see [73]), there is not a unanimous agreement on a single best definition [110]. In addition, in presence of more than two classifiers, not even a single global diversity index exists.

2.2.1 Random projection ensemble classification

In 2017 Cannings and Samworth [20] introduced a novel approach for high-dimensional binary classification based on RPs. The contribution of using RPs in the ensembles context is twofold: on one hand, the required ensemble *diversity* is ensured by the randomness of the projections; on the other, the dimensionality, p, of the dataset (and thus the classification complexity) is reduced while approximately preserving all the pairwise distances between points (Johnson-Lindenstrauss Lemma).
The main idea of the two authors is to generate a classification prediction by averaging over many individual ones and, then, use a data-driven voting threshold α to determine the final assignment. Specifically, each of the averaged B_1 prediction is obtained by applying an arbitrary base classifier on a different low-dimensional random projection of the data, carefully chosen in a set of B_2 possible solutions. The possibility of using any method as base classifier makes such technique a very general and flexible tool.

Let $\hat{C}_n = \hat{C}_{n,\tau_{n,d}}$ be the d-dimensional ($d \leq p$) generic base classifier trained on the data τ , consisting of n pairs in $\mathbb{R}_d \times \{0,1\}$. Let $A_1, A_2, \ldots, A_{B_1}$ be the B_1 independent random projections from $\mathbb{R}_p$ to $\mathbb{R}_d$ (where d is the projecting matrix

rank) chosen from as many non overlapping blocks of B_2 random projections yielding the smallest test error estimates.
Hence, the generic **RP ensemble classifier**, for some $\alpha \in (0,1)$, is given by:

$$\hat{C}_n^{RP} := \begin{cases} 1 & \text{if } \hat{\nu}_n^{B_1}(\mathbf{x}) \geq \alpha \\ 0 & \text{otherwise} \end{cases}$$

where

$$\hat{\nu}_n^{B_1}(\mathbf{x}) := \frac{1}{B_1} \sum_{b_1=1}^{B_1} \mathbb{1}_{\{\hat{C}_n(A_{b_1}\mathbf{x})=1\}}$$

and $\hat{C}_n(A_{b_1}\mathbf{x})$ is the projected data base classifier.

The RPs generating process is not uniquely-defined and, therefore, it could be performed by using different approaches. In their work [20], Cannings and Samworth discuss at first the possibility to simulate $A_1, A_2, \ldots, A_{B_1}$ according to the *Haar* measure. Namely, they suggest to generate a matrix $Q \in \mathbb{R}_{d\times p}$, where the entries are drawn independently from a standard normal distribution, and then use the left singular vectors of the Q^T singular value decomposition so as to derive A^T. Since such a process might be computationally expensive (the computation of the left singular vectors of a $p \times d$ matrix requires $\mathcal{O}(pd^2)$ operations), other alternatives to the Haar method are also mentioned in [20]. One one side, the idea of mapping the training data onto a lower dimensional subspace by employing a random *Gaussian* matrix (which requires only $\mathcal{O}(npd)$ operations) is presented; on the other, the use of projections constrained to be *Axis-Aligned* (i.e. each row of A consists of $p-1$ null components and one non-null component equals to 1) is suggested, especially in ultrahigh dimensional contexts.
For further details and practical considerations concerning the choice of α, d, B_1 and B_2 refer to [20].

Beside its excellent empirical performances, the very interesting aspect of the RP ensemble classifier introduced in [20] rests in its theoretical properties. In their work, in fact, Cannings and Samworth directly derived some important theoretical results on their proposal. As a first step, the authors proved that, as the number of projections increases, the test error of the RP ensemble classifier could be well approximated by its infinite simulation counterpart; in addition, they demonstrated

that the error in this approximation holds uniformly in the Binomial proportion. Then, with no specific assumption on the configuration of the training data, τ, and the distribution of both the test points and the individual projections, they were able to control the test excess risk. Namely, a bound for the difference between the test error of the RP ensemble classifier and the Bayes risk was obtained as the sum of three distinct terms: two of them only depend on the choice of the base classifier and the third one is proved to be even negligible as B_2 increases. Furthermore, a projection A^* yielding to an oracle decision boundary (in $\mathbb{R}^d$) essentially the same as the decision boundary of the Bayes classifier in the original space ($\mathbb{R}^p$) is proven to exist under some limited conditions (i.e. assumption 3 in [20]). Lastly, the theoretical framework in [20] focuses on the demonstration that, by using specific base classifiers (e.g. LDA or Knn), the first two terms of the above-mentioned bound are not affected by the number of input features, p; specifically, it was shown that these terms only depend on the dimension of the subspace (d), the sample size (n) and the number of projections (B_1 and B_2).

All the discussed theoretical results descend from the key assumption of **independent** RPs. Following this assumption, the authors also imply the independence of the base classifiers whose relation with the final ensemble is, thus, described by a Binomial model. In fact, answering to the question of Stander and Dalla Valle on whether it is possible to quantify the classification uncertainty by using the individual proportions $C_n^{A_1}, \ldots, C_n^{A_{B_1}}$, Cannings and Samworth in [20] suggest to employ the Binomial distribution.

2.3 Modeling ensemble accuracy

Let E be an ensemble of B_1 generic base classifiers, $C_i \quad i = 1, \ldots, B_1$, and let

$$D_i = \begin{cases} 1 & \text{if } C_i(\mathbf{x}) = y \\ 0 & \text{if } C_i(\mathbf{x}) \neq y \end{cases},$$

where y is the vector of true memberships.

Assuming

$$\pi_i = P(D_i = 1) = \pi \tag{2.1}$$

to be the individual probability of correctly classifying each observation, then:

- $D_i \sim \text{Ber}(\pi)$;

- the number of accurate predictions is

$$S = \sum_{i=1}^{B_1} D_i \quad ;$$

- the prediction accuracy of the majority vote ensemble E is

$$\widehat{Ac} = P(S \geq j+1) = 1 - F_S(j),$$

where F is the cumulative distribution function of the probability model assumed for S (which will be detailed in the following) and

$$j = \begin{cases} \frac{B_1}{2} & \text{if } B_1 \text{ is even} \\ \frac{B_1-1}{2} & \text{if } B_1 \text{ is odd.} \end{cases}$$

For independent base classifiers, it is straightforward that S follows a Binomial distribution (B), $S \sim \text{Bin}(B_1,\pi)$, and

$$\widehat{Ac} = \sum_{s=j+1}^{B_1} \binom{B_1}{s} \pi^s (1-\pi)^{B_1-s}.$$

In the case of independent and equally accurate base models (with accuracy $\pi > 1/2$), Lam and Suen [74] proved that the majority vote ensemble performs better than any of the individual classifiers that generate it.

However, in spite of the independence of the RPs, the assumption of independent classifiers for the RP ensemble is not realistic as they have been trained on the very same data.
The literature about the sums of non-independent Bernoulli random variables shows different possible strategies for dealing with the "intra-units" association: in 1948 Skellam [106] proposed to model the π parameter of the Binomial distribution with a Beta model of α and β hyperparameters; in 1978 [4] Altham discussed the possibility of extending the Binomial model in two different directions, the Additive Binomial distribution and the Multiplicative Binomial distribution, characterized respectively by an "additive" and a "multiplicative" definition of the interaction among units; in 2010, Diniz *et al.* [33] applied a Bayesian approach to the Correlated Binomial model introduced by Luceno in 1995 [79]; in 2016, Kadane [63] derived the Conway-

Maxwell-Binomial distribution so as to model both positive and negative dependence among the Bernoulli summands.

In presence of positive average pairwise correlation ρ, the Beta-Binomial distribution (BB) could be employed to characterize the accuracy $\hat{Ac}$ as:

$$\widehat{Ac} = \sum_{s=j+1}^{B_1} \binom{B_1}{s} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(s+\alpha)\Gamma(B_1-s+\beta)}{\Gamma(B_1+\alpha+\beta)}$$

where $\alpha = \pi(1-\rho)/\rho$, $\beta = \alpha(1-\pi)/\pi$, $\rho > 0$ so that $Var(\pi) > 0$, and $\Gamma(x)$ is the gamma function [2].
In order to allow for negative correlation, $\rho < 0$, Prentice [92] extended the BBD under the condition that $\rho \geq \max\{-\pi(B_1-\pi-1)^{-1}, -(1-\pi)[B_1-(1-\pi)-1]^{-1}\}$.

In this thesis, the intra-classifiers association in the ensemble context is accounted for the Multiplicative Binomial (MB) distribution, introduced in [4]. Specifically, we refer to a revised version of this distribution proposed by Lovison in 1998 [78], characterized by a more intuitive interpretation of the distribution parameters. Such a distribution is a member of the exponential family and, therefore, it has sufficient statistics and a family of proper conjugate distributions.
Under the assumption of exchangeable classifiers, the MB takes the form:

$$P(S=s) = \frac{\binom{B_1}{s}\psi^s(1-\psi)^{B_1-s}\omega^{(B_1-s)s}}{\sum_{i=0}^{B_1}\binom{B_1}{i}\psi^i(1-\psi)^{B_1-i}\omega^{(B_1-i)i}}.$$

Here:

- ψ, $0 \leq \psi = \pi/\tau_1 \leq 1$ is the **independence marginal probability** parameter (i.e. in the case of *independent* classifiers $\psi = \pi$), where

$$\tau_r(\psi,\omega) = \frac{K_{B_1-r}(\psi,\omega)}{K_{B_1}(\psi,\omega)} \qquad r = 1,\ldots,B_1$$

and

$$K_{B_1-a}(\psi,\omega) = \sum_{i=0}^{B_1-a} \binom{B_1-a}{i} \psi^i(1-\psi)^{B_1-a-i}\omega^{(B_1-a-i)(i+a)};$$

- $\omega > 0$ is the **intra-units association** parameter which governs the dependence between the classifiers: $\omega < 1$ describes positively associated classifiers, $\omega > 1$

a negative global relationship and $\omega = 1$ independent classifiers.
This measure is inversely related to the conditional cross-product ratio (CPR) as

$$\omega_{i,v} = \frac{1}{\sqrt{CPR_{i,v}|rest}},$$
$$CPR_{i,v}|rest = \frac{P(D_i = 1, D_v = 1)P(D_i = 0, D_v = 0)}{P(D_i = 1, D_v = 0)P(D_i = 0, D_v = 1)}, \qquad i, v = 1, \cdots, B_1, \quad i \neq v. \tag{2.2}$$

In [78], Lovison also derived the first two central moments of the MB distribution in a form that facilitates their comparison to the binomial ones:

$$E[S] = B_1\psi\tau_1$$
$$V[S] = B_1\psi\eta$$

where $\eta = \tau_1 - \psi(B_1\tau_1^2 - (B_1 - 1)\tau_2)$.

Following the MB model, in presence of dependent base classifiers (with the same individual probability of success π), the prediction accuracy of the majority vote ensemble is:

$$\widehat{Ac} = 1 - F_s(j) = \sum_{s=j+1}^{B_1} \frac{\binom{B_1}{s}\psi^s(1-\psi)^{B_1-s}\omega^{(B_1-s)s}}{\sum_{i=0}^{B_1}\binom{B_1}{i}\psi^i(1-\psi)^{B_1-i}\omega^{(B_1-i)i}}.$$

In order to investigate the goodness of fit of the Binomial (B), the Beta-Binomial (BB) and the Multiplicative Binomial (MB) distributions to the RP ensemble classifier accuracy, different scenarios were examined.
Specifically, RP ensembles of different sizes, $B_1 = \{5, 25, 100, 300, 500\}$, have been derived, by using to the method described in Section 2.2.1 (with $d = 2$ and $B_2 = 50$), from high-dimensional data generated according to the following model:

$$\mathbf{x}|\{y = 0\} \sim \frac{1}{2}N_p(\boldsymbol{\mu}_1, \Sigma) + \frac{1}{2}N_p(-\boldsymbol{\mu}_1, \Sigma)$$
$$\mathbf{x}|\{y = 1\} \sim \frac{1}{2}N_p(\boldsymbol{\mu}_2, \Sigma) + \frac{1}{2}N_p(-\boldsymbol{\mu}_2, \Sigma)$$

where $p = 100$, $\Sigma = I_{100\times100}$, $\boldsymbol{\mu}_1 = (2, -2, 0, \ldots, 0)^T$ and $\boldsymbol{\mu}_2 = (2, 2, 0, \ldots, 0)^T$.

		B	BB	MB
B_1=5	$M(\lvert Ac - \widehat{Ac}\rvert)$	0.317	0.013	0.008
	10×s.e	0.010	0.010	0.010
	χ^2	48.489	0.050	0.020
B_1=25	$M(\lvert Ac - \widehat{Ac}\rvert)$	0.126	0.017	0.012
	10×s.e	0.020	0.010	0.010
	χ^2	4.091	0.091	0.043
B_1=100	$M(\lvert Ac - \widehat{Ac}\rvert)$	0.044	0.022	0.013
	10×s.e	0.030	0.020	0.010
	χ^2	0.535	0.143	0.059
B_1=300	$M(\lvert Ac - \widehat{Ac}\rvert)$	0.102	0.020	0.018
	10×s.e	0.060	0.020	0.020
	χ^2	2.028	0.141	0.111
B_1=500	$M(\lvert Ac - \widehat{Ac}\rvert)$	0.168	0.021	0.032
	10×s.e	0.070	0.020	0.020
	χ^2	4.427	0.152	0.237

Table 2.1: Averages and standard errors (over 100 simulations) of the absolute differences between the sample accuracy, Ac, of an ensemble of B_1 classifiers and the expected one, $\hat{A}c$, predicted according to the Binomial (B), the Beta Binomial (BB) and the Multiplicative Binomial (MB) distributions. The goodness of fit χ^2 statistic values for each model are also reported.

Results coming from the simulation study confirm that the MB seems to characterize and predict the classification accuracy better than both the B and the BB models. Table 2.1 shows the average and the corresponding standard errors (over 100 simulations) of the absolute differences between the sample accuracy, Ac, of an ensemble of B_1 classifiers and the expected one, $\hat{A}c$, predicted according to the three different distributions. In the same table, the goodness of fit χ^2 statistic values are given.

2.3.1 Limit theorems of MB distribution

As discussed in the previous section, our interest in the MB distribution is closely related to the ensemble classification framework; namely, such distribution is directly employed to model the classification accuracy of the RP ensemble classifier introduced in [20]. Thus, with the intent to better characterize and understand the MB behavior in the ensemble context, some of its limits are investigated. The proof of all these results are given in Appendix A.

Theorem 1. *Let $S \sim MB(\psi, \omega)$, B_1 be the number of trials and k a positive integer:*

- $\forall B_1$*:*

$$S \xrightarrow[\omega \to 0^+]{d} \begin{cases} \delta(0) & \text{if } \psi \to 0 \\ \delta(\mathrm{B}_1) & \text{if } \psi \to 1 \end{cases}$$

- $\forall B_1 = 2k$*:*

$$S \xrightarrow[\omega \to +\infty]{d} \delta\left(\frac{\mathrm{B}_1}{2}\right)$$

- $\forall B_1 = 2k + 1$*:*

$$S \xrightarrow[\omega \to +\infty]{d} \begin{cases} \delta\left(\frac{\mathrm{B}_1 - 1}{2}\right) & \text{if } \psi \to 0 \\ \delta\left(\frac{\mathrm{B}_1 - 1}{2} + 1\right) & \text{if } \psi \to 1 \end{cases}$$

In Theorem 1 the convergence of the MB distribution to the Dirac delta one, δ, when both the parameters ω and ψ diverge, is proven. Such a result is particularly interesting from an ensemble point of view as it allows to identify the characteristics of the ensemble E (in terms of joint probability of success, ψ, and/or level of intra-units association, ω) that yield better performances: in particular, the closer to B_1 is the point mass of δ to which S converges, the larger is the ensemble accuracy predicted by the MB model.

Proposition 1. *Let $S \sim MB(\psi, \omega)$, B_1 be the number of trials,*

$$Z = \frac{S - \mathrm{B}_1 \psi \tau_1}{\sqrt{\mathrm{B}_1 \psi \eta}} \xrightarrow[\mathrm{B}_1 \to +\infty]{d} \mathcal{N}(0, 1)$$

where $\mathcal{N}$ is the Gaussian distribution.

This Proposition shows that, as B_1 increases, the MB asymptotically converges to the Gaussian distribution with mean $B_1 \psi \tau_1$ and variance $B_1 \psi \eta$. In view of this, the asymptotic confidence interval for the parameter, ψ, could be easily derived as

$$P\left\{ \frac{1}{\hat{\tau}_1} \left(\hat{\psi} - z_{\alpha/2} \sqrt{\frac{\hat{\psi}\hat{\eta}}{n}} \right) \leq \psi \leq \frac{1}{\hat{\tau}_1} \left(\hat{\psi} + z_{\alpha/2} \sqrt{\frac{\hat{\psi}\hat{\eta}}{n}} \right) \right\} = 1 - \alpha$$

and, then, compared to the one for the parameter π of the Binomial model:

$$P\left\{\hat{\pi} - z_{\alpha/2}\sqrt{\frac{\hat{\pi}(1-\hat{\pi})}{n}} \leq \pi \leq \hat{\pi} + z_{\alpha/2}\sqrt{\frac{\hat{\pi}(1-\hat{\pi})}{n}}\right\}.$$

Here, $\hat{\tau}_1$, $\hat{\psi}$, $\hat{\eta}$ and $\hat{\pi}$ are the maximum likelihood estimates of τ_1, ψ, η and π.

Theorem 2. *Let $S \sim MB(\psi, \omega)$, B_1 be the number of trials and $\psi = \pi/\tau_1 \geq 1/2$:*

$$\omega > 1 \Leftrightarrow \psi > \pi$$

In other words, the marginal probability of success ψ of a set of B_1 classifiers is larger than the common individual one, π, if and only if the B_1 classifiers are negatively related ($\omega \geq 1$) to each other.

In Theorem 2, the relationship between ψ, ω and π is determined and, then, reshaped in the ensemble context. Figures 2.1 and 2.2 clearly depict this association and illustrate that, for the same $\psi \geq 1/2$ (see, for example $\psi = 0.8$) and for a given size B_1 of the ensemble, as the negative dependence ω among the classifiers increases, the required individual accuracy π decreases.

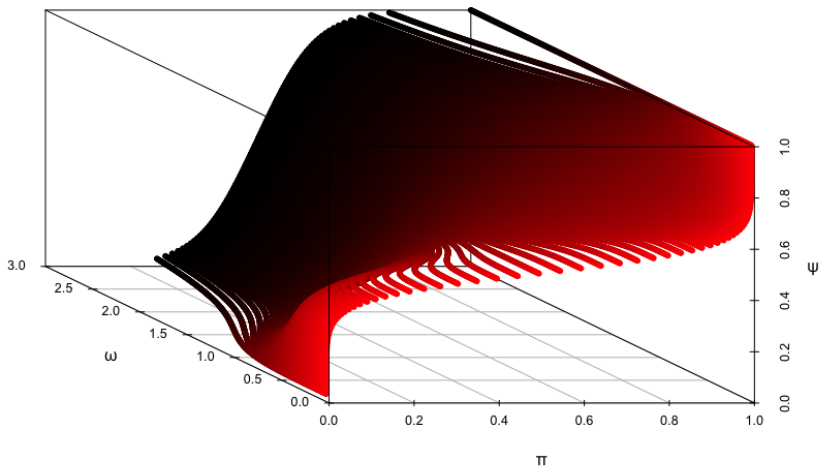


Figure 2.1: Relationship among ω, ψ and π, for $B_1 = 5$.

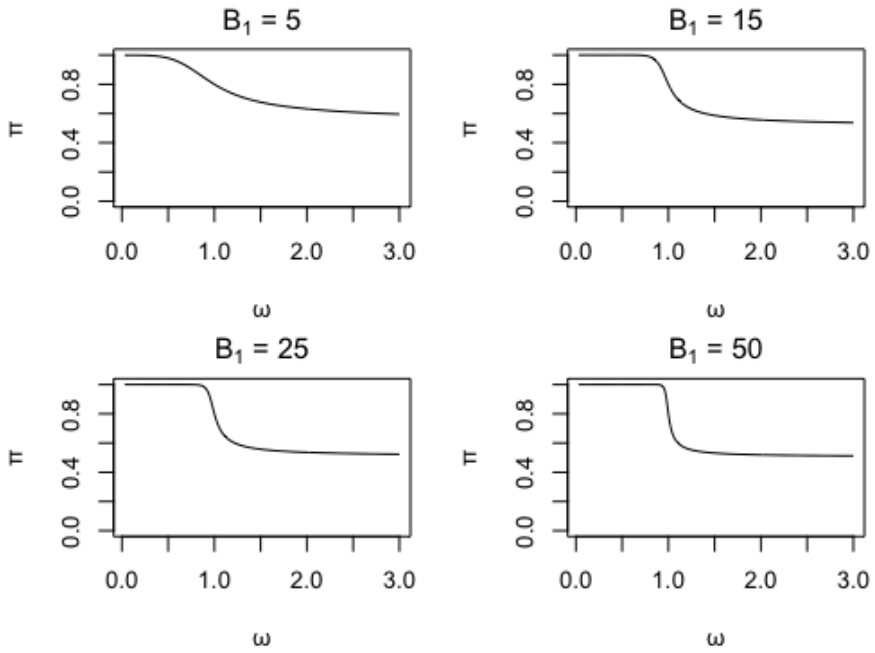


Figure 2.2: Relationship between ω and π for $\psi = 0.8$.

2.4 Ensemble post pruning

Most of the earliest ensemble approaches tend to exploit all the available individual results to produce the final prediction. However, as [119] pointed out, in the late 1990s, many researchers showed that removing some classifiers from an ensemble might determine a positive effect on the classification accuracy. In other words, in presence of *redundant* models, the accuracy of the final ensemble E can be lower than that of one or more of its subsets. See, as an example, the situation below, where $\forall i = 1, 2, 3, E$, $D_i = 1$ if the unit is correctly classified and $D_i = 0$ otherwise:

	D_1	D_2	D_3	D_E
Unit 1	1	1	1	**1**
Unit 2	0	0	1	**0**
Unit 3	1	1	1	**1**
Unit 4	1	0	1	**1**
Unit 5	0	1	0	**0**
Unit 6	1	1	1	**1**
Unit 7	0	0	1	**0**
Unit 8	0	1	0	**0**
Unit 9	1	1	1	**1**
Unit 10	1	0	0	**0**
Accuracy	**0.6**	**0.6**	**0.6**	**0.5**

A detailed review on methods to select classifiers from a given ensemble can be found in [119]. Tsoumakas *et al.* provide a taxonomy of the most important ensemble post pruning strategies, organizing them in four different categories: *Ranking*-based, *Optimization*-based, *Clustering*-based and *Other* methods.

Methods belonging to the *Ranking*-based category try to order individual learners according to a given criterion. Only the learners included in the tail of such distribution are considered in the final ensemble. The main differences among these methods consist in both the measure used to order the model (e.g. *kappa*-pruning, orientation ordering, ...) and the choice of the number of classifiers to retain (i.e. fixed number or dynamic selection).
The first *Ranking*-based strategy originated in 1997 from Margineantu and Dietterich's study on boosting pruning [80]. Later, after Tamon and Xiang (2000, [109]) stated that "*the boosting pruning problem is intractable even to approximate*", many other researchers changed their point of view and focused their attention on pruning ensembles generated by parallel methods (e.g. Bagging). One of the most recent approaches of the ranking category is the Collective-Agreement-based Pruning (CAP), introduced by Rokach in 2009 [93]. This algorithm aims to rank subset of classifiers (rather than individual classifiers) according to both the individual accuracy and the level of redundancy between them.

Optimization-based are guided by a performance measure to be optimized. This class appears to be the most reliable strategy in searching for an appropriate subset from an ensemble. In particular, heuristic methods based on the evaluation of as many different designs as possible seem to always select the best performing model [97]. However, the complexity of such techniques grows exponentially with the number, B_1, of classifiers in the ensemble and thus the problem becomes computationally intractable very quickly. When the ensemble size is large and consequently the search space is enormous, Genetic Algorithms (GA) could be employed to find the best subset of classifiers instead of other stochastic and evolutionary selection techniques, such as greedy hill-climbing [22], artificial immune algorithms [41, 129], case similarity search [27], rough set-based selection [60] and others. GA was firstly introduced by Holland in 1975 [58] as an effective optimization method which tries to find the best solution simulating some of the processes observed in natural selection. The evolution usually begins from a random initial population and then continues by choosing, at each step of the algorithm, the most fit[3] individuals from the current generation (parents). Over the parents a modification of the genetic information (or genome) in terms of mutation, crossover, inversion and selection is performed so as to create a new population. The process terminates when one of the stopping criteria is met, e.g. the maximum number of iteration is reached or the best solution during the evolution process does not change to a better value for a predefined number of generations.
In 2002, Zhou *et al.* firstly presented the GASEN [132] (Genetic Algorithm based Selective ENsemble), a new selective ensemble method which exploits a genetic algorithm to select the most appropriate subset of classifiers. A year later, Zhou and Tang [131] proposed a revised version of the GASEN, called GASEN-b, where a "hard" inclusion (i.e. 0/1), rather than a "soft" one (i.e. weighted), of the classifiers is performed.
As a consequence of the results obtained by Zhou *et al.*, many other authors later used GAs in searching for the best solutions for the classifier ensemble selection (see, among others, [5, 9, 10, 23, 24, 56, 57, 88, 105, 127]).

Clustering-based and dynamic selection methods are employed to simplify the ensemble selection process considerably, even if they do not guarantee the optimality

[3] the fitness function is usually the value of the objective function in the optimization problem being solved.

of the search. The core idea of the clustering-based pruning process is to identify groups of individual classifiers that present similar behavior and then select from each cluster the individual learner prototypes. For the initial phase, different clustering techniques can be used: hierarchical agglomerative clustering [46], k-means clustering based on Euclidean distance [75] and deterministic annealing [6].

Although they differ for the point of view, most of the strategies for ensemble pruning agree in considering *diversity* among individual classifiers as a key issue in building performing ensembles (see, among others,[2, 8, 15, 45, 50, 57, 70, 73, 103]). Experimental studies demonstrate that the ensemble performances might be improved if both the *accuracy* and the *diversity* measures are considered during the classifier selection process. In 2009, Ko *et al.* [66] introduced a compound diversity function for ensemble pruning which exploits both the individual classifiers' accuracy and diversity. In 2014, Bhatnagar *et al.* [8] presented a heuristic algorithm, called ADP, which combines together the individual classification accuracies and the pairwise diversities; such a procedure also eliminates the computational costs of the compound measure introduced by Ko *et al.*, by using an approach based on GA. In their work, Bhatnagar *et al.* asserted that "*ADP algorithm is highly likely to discover optimal ensemble. In case the optimal is missed, the discovered sub-optimal ensemble is empirically found to be close to the optimal ensemble in terms of both accuracy and size*". In 2015, Hernandez *et al.* [57] presented a multi-objective GA as a procedure to select, from all possible combinations of a large number of experts, the configuration of *diverse* base classifiers that provides the best possible *accuracy*.

In the RP ensemble context, if the number of relevant variables is low, the choice of the projection that yields the smallest estimate of the test error in each of B_1 blocks may cause a lack of diversity among the resulting classifiers. Therefore, in order to induce *diversity* and to avoid the selection of B_1 too similar base classifiers, Lu and Xue, in the discussion on the paper by Cannings and Samworth [20], suggest to use a greedy forward strategy that identifies the optimal projection matrices, by penalizing the similarity among them. In a similar spirit, Feng considers the idea of sequentially selecting the RPs so as to make them mutually orthogonal.
Although such techniques may help to generate ensembles of *diverse* base classifiers, they explicitly induce dependence among the projections and, therefore mine the

theoretical framework described 2.2.1.
In reply to these proposals, with the intent of increasing the *diversiy* among the ensemble classifiers, the authors themselves examine in [20] a new extension for the RP ensemble classifier. In particular, they discuss, for each projection, the possibility to randomize the choice of the base models (i.e. LDA, QDA, Knn) with probability 1/3 or, alternatively, to try all the three methods and retain the one that minimizes the leave-one-out error estimate.

In the following section, an innovative ensemble post pruning approach is introduced and applied to the RP ensemble classifier. Specifically, such procedure represents a valid option that allows to identify subsets of *diverse* base classifiers which, if jointly considered, provide *accurate* performances. Furthermore, the proposed algorithm, by performing an *aposteriori* classifier selection, keeps the RP projection matrices mutually independent and, therefore, it is coherent with the theoretical results discussed in [20].

2.4.1 A new proposal for the RP ensemble classifier selection

Motivated by the above-mentioned results from the literature, the idea of using the MBD as the reference model for the ensemble accuracy and the result of Theorem 2, a novel proposal for the selection of the classifiers in the RP ensemble, called Ensemble Selection Algorithm (ESA) is devised.

This technique follows a simple stepwise criterion: starting from a single classifier ensemble E, at each step it adds to the existing ensemble the classifier that is most similar to E in terms of accuracy π (Equation 2.1) and, at the same time, that provides the highest gain in terms of ω (Equation 2.2).
Specifically, the selection algorithm starts by joining the two individual classifiers to which the highest value in the compound matrix H is associated.

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,B_1} \\ & h_{2,2} & \cdots & h_{2,B_1} \\ & & \ddots & \vdots \\ & & & h_{B_1,B_1} \end{bmatrix} = \widetilde{\Pi} + \Omega$$

This hybrid matrix H is derived by adding the similarity matrix $\widetilde{\Pi}$ of the individ-

ual classifier accuracies, π_i, with the matrix of the pairwise dependencies between experts, Ω, measured here in terms of the MBD dependence parameter ω.
Then, following this heuristic principle, at each step of the procedure, a new classifier joins to the existing ensemble E, according to the highest increase in an objective function. In order to identify such classifier, the following steps are carried out:

1. consider a classifier C_i which has not yet been selected;
2. compute
   (a) the difference between the individual accuracy of classifier C_i, π_i, and the average individual accuracy of the existing ensemble, $\bar{\pi}_E$:
   $$\delta_{E,i} = 1 - \frac{|\pi_i - \bar{\pi}_E|}{\max(\pi) - \min(\pi)};$$
   (b) the gain obtained by selecting classifier C_i in terms of ω with respect to the existing ensemble:
   $$\widetilde{\omega}_i = \max\{\omega_{E \cup C_i} - \omega_E, 0\},$$
   where ω_e is the dependence parameter computed for ensemble e;
3. choose the classifier C_i which yields
   $$\max_i \{\delta_{E,i} + \widetilde{\omega}_i\}.$$

However, small differences between the entries of matrix H could not be relevant and a choice based on the highest term may not be optimal.
In order to overcome this potential limit, a multi-start strategy can be pursued; namely, instead of considering the single best value of H, n_{Best} (e.g. 3,5) combinations are taken and carried on.

2.4.2 Empirical analysis

The performances of the ESA have been assessed in terms of classification accuracy on both artificial and real data. For comparison, results from the RP ensemble classifier and those obtained by applying an alternative pruning method (the multi-objective Genetic Algorithm, GA, presented in [57]) are discussed. In particular, the latter procedure was implemented using the `GA` package [101] with fitness function

$$Fitness(\mathbf{x}) = Accuracy(\mathbf{x}) + Diversity(\mathbf{x})$$

where:

- $Accuracy(\mathbf{x}) = \frac{1}{\#(\mathbf{x})} \sum_{i=1}^{\#(\mathbf{x})} \pi_i$;
- $Diversity(\mathbf{x}) = \sum_{i<v}^{\#(\mathbf{x})} DF_{i,v} / \binom{\#(\mathbf{x})}{2}$.

  Here, $DF_{i,v}$ is the pairwise Double Fault measure introduced by Giacinto and Roli in [47], defined as the fraction of misclassifications (n^{00}) made by both the classifiers considered:
  $$DF_{i,v} = \frac{n^{00}}{n}.$$

For each example, an ensemble of size $B_1 = 101$ has been generated using the `RPParallel` function in the `RPEnsemble` package (Cannings and Samworth, 2016, [19]) with a training set of size n_{Tr}, a test set of size $n_{Te} = n - n_{Tr}$ (or, where available, a subsample of size 1000) and blocks of $B_2 = 50$ d-dimensional *Gaussian*-distributed RPs. For the LDA and QDA base classifiers, the training estimator for the test error suggested in [20] was employed; the Knn, instead, was performed by using the leave-one-out based estimate.
The subscript below each method refers to the dimension of the projected data, $d = \{2, 5\}$; the quantity in brackets denotes to the number of classifiers, $\hat{Cl}$, considered in the final ensemble.
Bold results highlight all the situations in which our proposal performs better than all the other competitors.

2.4.2.1 Simulated examples

In this section, the ESA was applied on the four simulated models described in [20] for $\pi_1 = 0.5$, using LDA as base classifier, $n_{Tr} = 200$ and $p = 100$. For each scenario, $N_{reps} = 30$ repetitions were carried out.
Tables 2.2-2.3 show, for all the methods, the averages of both the accuracy rate, $\hat{a} = M(\hat{Ac})$ and, in brackets, the number of selected classifiers, $\hat{cl} = M(\hat{Cl})$. A measure of variability for $\hat{a}$ and $\hat{cl}$ is also provided (i.e. the standard error of the statistic designated by the subscript). In particular, the standard error of $\hat{a}$ in the

tables below is estimated as:

$$\frac{1}{N_{reps}^{1/2}}\left\{\frac{(1-\hat{a})\hat{a}}{n_{Te}}+\frac{n-1}{n_{Te}N_{reps}}\sum_{l=1}^{N_{reps}}(\hat{a}-\hat{A}c_l)^2\right\}^{1/2}.$$

See [20] for further details.

Model 1 – Sparse class boundaries

$$\mathbf{x}|\{y=0\}\sim\frac{1}{2}N_p(\boldsymbol{\mu}_0,\Sigma)+\frac{1}{2}N_p(-\boldsymbol{\mu}_0,\Sigma)$$
$$\mathbf{x}|\{y=1\}\sim\frac{1}{2}N_p(\boldsymbol{\mu}_1,\Sigma)+\frac{1}{2}N_p(-\boldsymbol{\mu}_1,\Sigma)$$

$\Sigma=I_{p\times p}$, $\boldsymbol{\mu}_0=(2,-2,0,\ldots,0)^T$ and $\boldsymbol{\mu}_1=(2,2,0,\ldots,0)^T$

Model 2 – Rotated sparse normal

$$\mathbf{x}|\{y=0\}\sim N_p(\Omega_p\boldsymbol{\mu}_0,\Omega_p\Sigma_0\Omega_p^T)$$
$$\mathbf{x}|\{y=1\}\sim N_p(\Omega_p\boldsymbol{\mu}_1,\Omega_p\Sigma_1\Omega_p^T)$$

Ω_p is a $p\times p$ rotation matrix sampled once according to the Haar measure, $\boldsymbol{\mu}_0=(3,3,3,0,\ldots,0)^T$, $\boldsymbol{\mu}_1=(0,\ldots,0)^T$.
Σ_0 and Σ_1 are block diagonal, with blocks $\Sigma_0^{(1)}$ (3×3 matrix with diagonal entries equal to 2 and off-diagonal entries equal to $\frac{1}{2}$), $\Sigma_1^{(1)}=\Sigma_0^{(1)}-I_{3\times3}$ and $\Sigma_0^{(2)}=\Sigma_1^{(2)}$ ($(p-3)\times(p-3)$ matrix with diagonal entries equal to 1 and off-diagonal entries equal to $\frac{1}{2}$).

- $\underset{3\times3}{\Sigma_0^{(1)}}=\begin{bmatrix}2&0.5&0.5\\0.5&2&0.5\\0.5&0.5&2\end{bmatrix}$

- $\underset{3\times3}{\Sigma_1^{(1)}}=\begin{bmatrix}1&0.5&0.5\\0.5&1&0.5\\0.5&0.5&1\end{bmatrix}$

- $\underset{(p-3)\times(p-3)}{\Sigma_0^{(2)}} = \underset{(p-3)\times(p-3)}{\Sigma_1^{(2)}} = \begin{bmatrix} 1 & 0.5 & \dots & 0.5 \\ 0.5 & 1 & \dots & 0.5 \\ \vdots & & \ddots & \vdots \\ 0.5 & 0.5 & \dots & 1 \end{bmatrix}$

Model 3 – Independent features

$\mathbf{x}|\{y=0\} \sim N_p(\boldsymbol{\mu}, I_{p\times p})$

$\mathbf{x}|\{y=1\}$ is simulated from a distribution of p independent components each with a standard Laplace distribution, $L(0,1)$.

$\boldsymbol{\mu} = (1/\sqrt{p})(1,\dots,1,0\dots,0)^T$ is the mean vector of the normal distribution with $p/2$ non-zero components.

Model 4 – t-distributed features

$$\mathbf{x}|\{y=r\} = \boldsymbol{\mu}_r + Z_r/\sqrt{(U_r/\nu_r)} \qquad \text{r = 0,1}$$

$Z_r \sim N_p(0,\Sigma_r)$ independent of $U_r \sim \chi^2_{\boldsymbol{\nu}_r}$. $\boldsymbol{\mu}_0 = (1,\dots,1,0\dots,0)^T$ with 10 non-zero components, $\boldsymbol{\mu}_1 = 0$, $\nu_0 = 2$, $\nu_1 = 1$, $\Sigma_0 = \Sigma_{j,k}$ where $\Sigma_{j,j} = 1$, $\Sigma_{j,k} = 0.5$ if $\max(j,k) \leq 10$ and $j \neq k$, $\Sigma_{j,k} = 0$ otherwise, and $\Sigma_1 = I_{p\times p}$.

RP Method$_d$	*Results for Model 1*	*Results for Model 2*
LDA$_2$	50.59 $_{0.49}$ (101.00 $_{0.00}$)	93.88 $_{0.25}$ (101.00 $_{0.00}$)
ESA-LDA$_2$	**50.67** $_{0.43}$ (70.30 $_{4.43}$)	92.73 $_{0.32}$ (38.57 $_{4.34}$)
GA-LDA$_2$	50.54 $_{0.43}$ (52.27 $_{0.80}$)	93.43 $_{0.25}$ (45.67 $_{0.95}$)
LDA$_5$	50.27 $_{0.38}$ (101.00 $_{0.00}$)	94.16 $_{0.23}$ (101.00 $_{0.00}$)
ESA-LDA$_5$	49.85 $_{0.38}$ (58.80 $_{4.25}$)	93.30 $_{0.28}$ (27.73 $_{3.83}$)
GA-LDA$_5$	49.92 $_{0.40}$ (51.70 $_{0.64}$)	93.86 $_{0.21}$ (42.97 $_{0.97}$)

Table 2.2: Accuracy rates with standard errors and (number of selected classifiers with standard errors) for Models 1 and 2.

The overall results showed in Tables 2.2 and 2.3 demonstrate that removing redundant classifiers from the RP ensemble (rather than using the entire set) could determine a performance gain. Moreover, in all the situations where it occurs (Model 1, Model 3 and Model 4 with $d = 2$), the ESA tends to be more effective than the

RP Method$_d$	*Results for Model 3*	*Results for Model 4*
LDA_2	50.39 $_{0.86}$ (101.00 $_{0.00}$)	64.26 $_{1.64}$ (101.00 $_{0.00}$)
$ESA\text{-}LDA_2$	**50.47** $_{0.82}$ (66.97 $_{3.98}$)	**64.99** $_{1.35}$ (46.23 $_{6.05}$)
$GA\text{-}LDA_2$	50.38 $_{0.88}$ (55.37 $_{0.98}$)	63.53 $_{1.54}$ (50.73 $_{0.98}$)
LDA_5	52.72 $_{0.70}$ (101.00 $_{0.00}$)	69.68 $_{1.09}$ (101.00 $_{0.00}$)
$ESA\text{-}LDA_5$	52.39 $_{0.69}$ (60.87 $_{4.82}$)	69.35 $_{1.00}$ (49.07 $_{5.51}$)
$GA\text{-}LDA_5$	52.22 $_{0.66}$ (53.07 $_{0.90}$)	68.98 $_{1.07}$ (50.60 $_{0.99}$)

Table 2.3: Accuracy rates with standard errors and (number of selected classifiers with standard errors) for Models 3 and 4.

GA in selecting the smallest subset of base classifiers that provide the best possible accuracy.

2.4.2.2 Real data examples

Seven different high-dimensional datasets available from the UC Irvine (UCI) Machine Learning Repository [76] have been used to evaluate the method performances. In all the real applications, the ESA has been trained for $n_{Best} = 5$ solutions.

Eye state detection dataset

The electroencephalogram eye state dataset provides information about $p = 14$ electroencephalogram measurements on 14980 patients. The task is to use these information to determine whether the eye is either open (class 0, size 8256) or closed (class 1, size 6723).

Ionosphere dataset

The ionosphere dataset contains $p = 32$ high-frequency antenna measurements for 315 observations. Specifically, radar returns from the ionosphere are classified as either suitable for further analysis (class 0, size 225) or not (class 1, size 126) depending on the evidence for free electrons.

Down's syndrome diagnoses in mice

The mice dataset consists of the expression levels of $p = 68$ proteins/protein modifications on 1080 mice. The task is to classify mice as healthy (class 0, size 570) or affected by Down's syndrome (class 1, size 507) on the basis of their protein expression measurements.

Hill-valley dataset

The hill-valley dataset consists of 1212 observations, each of them representing 100 points on a two-dimensional graph. When plotted in sequence, the points create either a hill (a"bump" in the terrain; class 0, size 600) or a valley (a "dip" in the terrain; class 1, size 612). The goal of the analysis is to classify the terrain on the basis of a vector of dimension $p = 100$.

Musk dataset

The musk dataset consists of 6598 molecules classified as *musk* (class 0, size 1016) or *non-musk* (class 1, size 5581), based on $p = 166$ features that describe the exact shape or the conformation of each molecule. The goal is to learn to predict whether new molecules will be musks or non-musks.

Cardiac arrhythmia dataset

The cardiac arrhythmia dataset contains observations on 452 patients. The aim is to distinguish between the *presence* (class 0, size 245) and *absence* (class 1, size 207) of cardiac arrhythmia using results from $p = 190$ electrocardiogram (ECG) measurements.

Human activity recognition dataset

The human activity recognition dataset contains $p = 561$ measurements, recorded from a waist-mounted smartphone with embedded inertial sensors while a subject is performing an activity. The initial dataset has been subsampled in order to include only two of the six original activity: walking and laying. The final dataset consists of 1226 walking" observations (class 0) and 1407 "laying" observations (class 1).

As noticed for the simulation results, for the real data applications too, the classification performances of the ESA are generally in line with those yielded by the other competitors. Moreover, in some real examples (e.g. for the *mice* and the *hill-valley* datasets), the improvement in classification accuracy provided by our proposal is particularly evident.
The inspection of the values in brackets (i.e. the number of classifiers selected for the final ensemble), clearly shows the tendency of the ESA (already mentioned in Section 2.4.2.1 for the simulated examples) to consider *small* subsets of classifiers.

RP Method$_d$	*Results for eye state data*			*Results for ionosphere data*		
	$n_{Tr} = 50$	$n_{Tr} = 200$	$n_{Tr} = 1000$	$n_{Tr} = 50$	$n_{Tr} = 100$	$n_{Tr} = 200$
LDA_2	62.80 (101)	59.20 (101)	62.20 (101)	87.38 (101)	90.04 (101)	90.07 (101)
ESA-LDA_2	60.50 (27)	**62.70** (2)	**63.80** (3)	82.06 (7)	89.24 (49)	89.40 (28)
GA-LDA_2	61.80 (48)	60.20 (47)	62.00 (51)	87.38 (25)	86.45 (33)	88.08 (32)
LDA_5	60.70 (101)	57.60 (101)	63.00 (101)	88.37 (101)	88.45 (101)	90.07 (101)
ESA-LDA_5	58.80 (12)	**58.60** (23)	62.50 (24)	87.71 (10)	86.06 (2)	89.40 (6)
GA-LDA_5	61.10 (32)	57.90 (55)	62.80 (44)	85.38 (26)	87.65 (41)	88.74 (36)
QDA_2	64.80 (101)	66.00 (101)	65.50 (101)	88.70 (101)	94.42 (101)	92.72 (101)
ESA-QDA_2	58.70 (7)	**66.60** (51)	67.00 (23)	**91.69** (6)	92.03 (16)	**94.04** (13)
GA-QDA_2	64.00 (42)	65.90 (57)	68.70 (57)	84.05 (27)	93.23 (32)	91.39 (53)
QDA_5	64.70 (101)	70.80 (101)	74.60 (101)	88.70 (101)	91.63 (101)	96.69 (101)
ESA-QDA_5	63.10 (6)	70.90 (10)	73.70 (11)	86.71 (1)	**92.83** (3)	94.70 (7)
GA-QDA_5	62.60 (28)	71.40 (49)	74.50 (48)	87.38 (33)	91.24 (39)	96.03 (51)
Knn_2	64.10 (101)	66.50 (101)	76.40 (101)	94.35 (101)	93.63 (101)	95.36 (101)
ESA-Knn_2	63.60 (9)	65.80 (64)	76.40 (99)	83.06 (4)	87.65 (4)	94.70 (27)
GA-Knn_2	58.10 (43)	66.60 (55)	77.00 (56)	90.03 (38)	93.23 (21)	94.04 (43)
Knn_5	60.30 (101)	72.50 (101)	88.20 (101)	87.04 (101)	94.82 (101)	92.05 (101)
ESA-Knn_5	**65.30** (7)	69.70 (23)	67.90 (57)	89.04 (5)	90.44 (2)	95.36 (7)
GA-Knn_5	61.50 (37)	72.40 (52)	87.50 (51)	92.05 (46)	89.64 (31)	96.69 (32)

Table 2.4: Accuracy rates and (number of selected classifiers) for the eye state and ionosphere data.

In addition to the discussed outcomes, a further analysis was implemented with the aim to compare the performances of the considered post-pruning approaches (ESA and GA) with those yielded by the new extensions suggested by Cannings and Samworth to increase the RP ensemble diversity. In particular, Table 2.8 contains the accuracy rates for the *mice* and the *hill-valley* datasets obtained by employing the leave-one-out (loo) estimator for all the three base classifiers (i.e. LDA, QDA, Knn) in the RP ensemble and by performing the new procedures introduced in the discussion on [20]. Specifically, with "Random" we denote the authors' proposal of randomly choosing, on each projection, the base classifier; with "All", instead, we refer to the alternative of trying all the base methods on each projection and, than, selecting the most performing one.

Results from this numerical study reveals once again that diversity is a key issue in classifier combination. Moreover, our proposal of *aposteriori* selecting the most diverse and accurate set of the ensemble classifiers according to the MB parameters, seems to provide good results. In fact, the accuracy rates yielded by the ESA are always better than (or comparable with) those achieved by inducing diversity during the RP ensemble generating process.

RP Method$_d$	*Results for mice data*			*Results for hill-valley data*		
	n_{Tr} = 200	n_{Tr}= 500	n_{Tr}= 1000	n_{Tr} = 100	n_{Tr}= 200	n_{Tr}= 500
LDA_2	71.49 (101)	70.54 (101)	64.94 (101)	56.10 (101)	62.90 (101)	63.34 (101)
ESA-LDA_2	**73.55** (19)	**76.60** (8)	**67.53** (8)	55.40 (4)	**71.00** (1)	**64.89** (4)
GA-LDA_2	72.41 (61)	68.80 (41)	71.43 (39)	53.50 (47)	59.80 (53)	64.04 (34)
LDA_5	74.23 (101)	76.26 (101)	72.73 (101)	60.40 (101)	80.40 (101)	69.94 (101)
ESA-LDA_5	**81.98** (6)	**80.59** (8)	72.73 (14)	**63.90** (11)	**80.90** (68)	**77.95** (27)
GA-LDA_5	74.34 (50)	78.34 (43)	77.92 (46)	60.30 (50)	80.70 (100)	82.58 (53)
QDA_2	74.68 (101)	75.91 (101)	72.73 (101)	57.90 (101)	58.90 (101)	59.55 (101)
ESA-QDA_2	**78.56** (17)	**83.71** (11)	**81.82** (11)	57.90 (101)	**59.90** (16)	**61.24** (11)
GA-QDA_2	77.31 (52)	76.78 (51)	75.32 (41)	60.70 (55)	59.80 (52)	59.55 (39)
QDA_5	82.21 (101)	83.88 (101)	81.82 (101)	66.40 (101)	67.40 (101)	62.08 (101)
ESA-QDA_5	**85.86** (14)	**86.48** (11)	85.71 (44)	66.10 (100)	**67.40** (101)	**64.47** (8)
GA-QDA_5	83.58 (42)	84.23 (48)	87.01 (55)	65.20 (44)	66.80 (60)	60.11 (40)
Knn_2	85.40 (101)	93.59 (101)	97.40 (101)	52.20 (101)	56.00 (101)	70.93 (101)
ESA-Knn_2	**86.32** (59)	93.24 (82)	97.40 (79)	53.30 (16)	**56.30** (22)	**83.57** (16)
GA-Knn_2	84.38 (29)	92.89 (50)	98.70 (51)	53.90 (49)	54.70 (49)	74.58 (49)
Knn_5	86.55 (101)	97.05 (101)	98.70 (101)	50.20 (101)	52.80 (101)	64.75 (101)
ESA-Knn_5	86.20 (6)	**98.27** (15)	**100.00** (6)	50.50 (10)	**54.50** (12)	**69.52** (17)
GA-Knn_5	86.32 (48)	97.40 (36)	100.00 (51)	51.30 (49)	51.50 (54)	66.29 (40)

Table 2.5: Accuracy rates and (number of selected classifiers) for the mice and hill-valley data.

2.5 Variable Importance in ensembles

As discussed in the previous section, ensemble of classifiers proved to be a very useful tool for excellently solving many classification problems. In particular, by combining the predictions of several (potentially weak) base classifiers, ensembles allow to better improve both the generalizability and the robustness of the final estimates. Hovewer, these notable performances carry a remarkable drawback that strongly affects ensemble algorithms. Namely, methods in this class could be considered as "black-boxes" which take in input and give out just predictions, without worrying too much about the underlying mechanism. In this sense, one of the main shortcomings of ensembles is the fact that, differently from the single classifier, they loose connection with the original variables and, therefore, do not provide any insight about the feature importance in the classification process.

Among the proposed ensembles of classifiers, the Random Forest procedure represents one of the most commonly used. The RF algorithm was firstly introduced by Breiman in 2001 [16] as an ensemble learning technique which combines the predictions of B_1 weak learners (classification or regression trees) in order to boost their individual performances. In order to help the interpretation of the final outcome and to overcome the ensemble limits above-discussed, the possibility of efficiently

RP Method$_d$	*Results for musk data*			*Results for cardiac data*		
	$n_{Tr} = 100$	$n_{Tr} = 200$	$n_{Tr} = 500$	$n_{Tr} = 50$	$n_{Tr} = 100$	$n_{Tr} = 200$
LDA_2	83.00 (101)	83.10 (101)	83.30 (101)	62.94 (101)	72.59 (101)	77.38 (101)
ESA-LDA_2	77.90 (4)	**85.70** (18)	**86.80** (44)	**63.68** (11)	67.05 (8)	75.00 (26)
GA-LDA_2	83.00 (50)	83.10 (50)	83.30 (50)	62.19 (43)	72.16 (50)	73.41 (55)
LDA_5	74.00 (101)	81.90 (101)	88.80 (101)	63.18 (101)	71.88 (101)	76.59 (101)
ESA-LDA_5	**83.60** (6)	**83.80** (32)	88.70 (6)	**64.43** (11)	70.17 (10)	**78.17** (37)
GA-LDA_5	73.00 (39)	82.60 (39)	88.80 (47)	62.69 (36)	71.02 (42)	75.79 (51)
QDA_2	83.00 (101)	83.10 (101)	87.90 (101)	61.94 (101)	70.74 (101)	77.38 (101)
ESA-QDA_2	78.70 (51)	82.40 (37)	87.50 (50)	61.69 (69)	**70.74** (35)	**78.17** (41)
GA-QDA_2	78.40 (51)	83.10 (50)	87.50 (65)	59.20 (48)	69.60 (46)	77.38 (50)
QDA_5	83.70 (101)	88.60 (101)	90.00 (101)	60.20 (101)	72.73 (101)	78.57 (101)
ESA-QDA_5	81.90 (1)	85.90 (5)	89.10 (14)	59.95 (16)	68.47 (17)	78.17 (13)
GA-QDA_5	82.40 (21)	85.60 (36)	88.90 (32)	57.71 (41)	66.19 (50)	77.38 (42)
Knn_2	84.00 (101)	85.20 (101)	90.00 (101)	59.70 (101)	73.30 (101)	73.02 (101)
ESA-Knn_2	83.10 (7)	81.40 (99)	87.60 (6)	58.71 (14)	73.01 (25)	**73.02** (101
GA-Knn_2	78.50 (54)	84.00 (56)	89.10 (45)	62.19 (44)	69.32 (51)	69.44 (55)
Knn_5	86.20 (101)	86.00 (101)	89.10 (101)	66.42 (101)	70.74 (101)	76.59 (101)
ESA-Knn_5	85.00 (5)	**87.50** (25)	88.00 (5)	63.93 (26)	67.33 (21)	78.18 (75)
GA-Knn_5	86.00 (39)	85.20 (53)	89.40 (60)	63.93 (43)	69.60 (43)	77.78 (56)

Table 2.6: Accuracy rates and (number of selected classifiers) for the musk and cardiac arrhythmia data.

ranking the input features according to their importance was considered since the first formulation of the algorithm. In particular, in RFs, the strength of a generic u-th feature can be measured by averaging, over all the trees in the forest, the difference between the initial Out-Of-Bag (OOB) error and the OOB error computed after permuting the values for the u-th variable in the OOB sample. The final score is than obtained by normalizing these differences with their standard deviations.

Inspired both by the RF process for variable ranking and the work of Montanari and Lizzani [85] on projection pursuits, the main idea in this work is to use the information provided by the RP ensemble classifier so as to mitigate the typical lack of interpretability which characterizes of ensembles.

2.5.1 Variable ranking for the RP ensemble

A still open issue in [20] is "*to understand the properties of the variable ranking induced by the RP ensemble classifier*". In fact, despite such classifier highly improves the classification accuracy, it does not allow to identify the variables with the highest discriminative power, as a single classifier does.
In the discussion on the paper by Cannings and Samworth [20], several contributors mention the potential use of sparse RPs (e.g. Axis-Aligned Random Projections,

RP Method$_d$	*Results for human activity recognition data*		
	$n_{Tr} = 50$	$n_{Tr} = 200$	$n_{Tr} = 1000$
LDA_2	99.80 (101)	100.00 (101)	99.90 (101)
ESA-LDA_2	98.80 (1*)	99.50 (1**)	**100.00** (1)
GA-LDA_2	99.80 (50)	100.00 (20)	99.80 (13)
LDA_5	99.80 (101)	100.00 (101)	100.00 (101)
ESA-LDA_5	99.40 (1*)	99.50 (1*)	**100.00** (1**)
GA-LDA_5	99.80 (50)	100.00 (50)	99.60 (4)
QDA_2	99.80 (101)	100.00 (101)	100.00 (101)
ESA-QDA_2	98.00 (1*)	99.40 (1**)	99.70 (1)
GA-QDA_2	99.80 (50)	99.90 (15)	99.90 (14)
QDA_5	99.90 (101)	99.90 (101)	100.00 (101)
ESA-QDA_5	99.00 (1*)	99.40 (1*)	99.60 (1)
GA-QDA_5	99.80 (50)	99.90 (50)	100.00 (50)
Knn_2	99.80 (101)	99.90 (101)	100.00 (101)
ESA-Knn_2	99.10 (1*)	99.50 (1**)	99.90 (1)
GA-Knn_2	99.90 (50)	100.00 (12)	99.60 (19)
Knn_5	99.70 (101)	99.90 (101)	99.90 (101)
ESA-Knn_5	98.50 (1*)	99.50 (1*)	99.50 (1**)
GA-Knn_5	99.70 (50)	99.90 (50)	100.00 (50)

* means that all the π_i are equal and, thus, the ESA does not start.
** means that the H matrix does not contain $n_{Best} = 5$ different values and, thus, only smaller solutions of n_{Best} (corresponding to the number of distinct $h_{i,v}$, $i \neq v$) are explored.

Table 2.7: Accuracy rates and (number of selected classifiers) for the human activity recognition data.

AA-RP) to measure the importance of each input variable. Gataric, for example, numerically demonstrates that performing a majority vote scheme across the B_1 projections

$$\hat{a}_u^* = \frac{1}{B_1} \sum_{i=1}^{B_1} \mathbb{1}_{\{(A_i^T A_i)_{u,u}=1\}} \qquad u = 1, \cdots, p \tag{2.3}$$

could provide a good estimation of the classification power for each feature u.

In this work, in the same spirit, a specific coefficient, called *Variable Importance in Projection* (VIP), is introduced so as to evaluate the importance of each input variable.
Following Montanari and Lizzani (2001), for the u-th variable the *Importance Coefficient* (CI) is defined as

$$CI_{ui} = \sum_{q=1}^{d} \frac{|a_{uqi}| s_u}{\sqrt{\sum_{z=1}^{p} (a_{uzi} s_u)^2}} \qquad i = 1, \cdots, B_1$$

RP Method$_d$	*Results for mice data*			*Results for hill-valley data*		
	n_{Tr} = 200	n_{Tr}= 500	n_{Tr}= 1000	n_{Tr} = 100	n_{Tr}= 200	n_{Tr}= 500
LDA_2	70.35 (101)	71.75 (101)	66.23 (101)	56.10 (101)	63.20 (101)	63.76 (101)
ESA-LDA_2	**73.43** (8)	**77.12** (4)	63.64 (1)	**58.90** (3)	**67.10** (15)	63.06 (10)
GA-LDA_2	69.78 (54)	74.87 (41)	67.53 (47)	53.50 (47)	60.80 (49)	64.47 (49)
LDA_5	73.55 (101)	73.48 (101)	72.73 (101)	60.40 (101)	69.60 (101)	68.82 (101)
ESA-LDA_5	**78.45** (16)	**80.24** (6)	**81.82** (9)	**64.10** (6)	**72.60** (16)	**71.91** (9)
GA-LDA_5	74.66 (54)	76.95 (45)	74.03 (53)	59.60 (42)	66.30 (50)	68.82 (43)
QDA_2	74.68 (101)	75.91 (101)	72.73 (101)	53.30 (101)	59.70 (101)	59.55 (101)
ESA-QDA_2	**78.56** (17)	**83.71** (11)	**81.82** (11)	**54.50**(1)	**59.70** (91)	**62.36** (9)
GA-QDA_2	77.31 (52)	76.78 (51)	75.32 (41)	53.40 (57)	57.00 (41)	59.41 (46)
QDA_5	82.21 (101)	83.88 (101)	81.82 (101)	60.20 (101)	59.20 (101)	61.80 (101)
ESA-QDA_5	**85.86** (14)	**86.48** (11)	85.71 (44)	56.00 (1)	**59.60** (9)	**64.04** (17)
GA-QDA_5	83.58 (42)	84.23 (48)	87.01 (55)	56.90 (47)	58.20 (43)	60.39 (17)
Knn$_2$	85.40 (101)	93.59 (101)	97.40 (101)	52.20 (101)	56.00 (101)	70.93 (101)
ESA-Knn$_2$	**86.32** (59)	93.24 (82)	97.40 (79)	53.30 (16)	**56.30** (22)	**83.57** (16)
GA-Knn$_2$	84.38 (29)	92.89 (50)	98.70 (51)	53.90 (49)	54.70 (49)	74.58 (49)
Knn$_5$	86.55 (101)	97.05 (101)	98.70 (101)	50.20 (101)	52.8 (101)	64.75 (101)
ESA-Knn$_5$	86.20 (6)	**98.27** (15)	**100.00** (6)	50.50 (10)	**54.50** (12)	**69.52** (17)
GA-Knn$_5$	86.32 (48)	97.40 (36)	100.00 (51)	51.30 (49)	51.50 (54)	66.29 (40)
Random$_2$	84.49 (101)	90.64 (101)	94.80 (101)	52.00 (101)	55.20 (101)	66.29 (101)
All$_2$	84.15 (101)	94.28 (101)	100.00 (101)	53.70 (101)	55.60 (101)	73.59 (101)
Random$_5$	84.72 (101)	97.05 (101)	100.00 (101)	60.70 (101)	68.40 (101)	66.43 (101)
All$_5$	87.00 (101)	97.40 (101)	98.70 (101)	59.80 (101)	70.30 (101)	72.89 (101)

Table 2.8: Accuracy rates and (number of selected classifiers) for the mice and hill-valley data obtained by using the loo estimator.

where a_{uqi} indicates the attribute u coefficient in the q-th vector of the d-dimensional random projection solution and s_u the variability (i.e. the standard deviation) of each attribute.

The Variable Importance in Projection for feature u is then obtained as

$$VIP_u = \underset{i=1,\ldots,B_1}{\text{median}}\ CI_{ui}. \tag{2.4}$$

The median is used here so as to mitigate the effects on the VIP of potential not-so-good projections. By computing the VIP it is possible to rank the input features and highlight the most relevant ones for classification purposes.

The number of variables to be kept is decided by the user; a possible strategy is to explore all the solutions and, then, retain only the first h variables that minimize the test error estimate.

2.5.2 Empirical analysis

Performances of the VIP criterion have been evaluated in both simulated and real data applications.

As a first step, for each simulated scenario, the capability of the measure in 2.4 to recognize the actual important variables was tested and, then, compared to the one described in 2.3. Secondly, both the VIP (RP-VIP) and the proposal by Gataric (AA-RP) were applied within the RP ensemble classifier framework with the specific aim to address classification issues. In this case, the input variables of each dataset have been initially ranked according to the two discussed criteria, each computed on $B_1 = 101$ d-dimensional *Gaussian*-distributed RP matrices selected within blocks of $B_2 = 50$ possible solutions; then, three base classifiers (LDA, QDA, Knn) were performed on 100 different training sets, by using, for each method, only the first h variables yielding the largest estimate of the training accuracy.
In addition to the accuracy rates provided by the RP-VIP and the AA-RP ensemble classifiers, results from the RP ensemble classifier in [20] and the "standard" classification (i.e. by applying the base classifier in the original space) are reported.
The subscript below each method still refers to the dimension of the projected data, $d = \{2, 5\}$.

2.5.2.1 Simulated examples

In this section, the VIP criterion (2.4, RP-VIP) and the proposal discussed by Gataric (2.3, AA-RP) for variable ranking have been tested and compared in a Monte Carlo simulation study, focusing on their capability of recovering the actually important features, p^*.
In particular, four different simulation settings, inspired to the synthetic data examples described in [81], were considered.
Samples of size $n_{Tr} \in 50, 100$ have been simulated for a $p = 100$-dimensional feature vector, where only the first $p^* = 4$ variables contain useful information for classification purposes. The **relevant** features were generated from the following distribution,

$$\mathbf{x}_{[1:4]}|\{y = 0\} \sim N_4(\boldsymbol{\mu}_0, I_2)$$
$$\mathbf{x}_{[1:4]}|\{y = 1\} \sim N_4(\boldsymbol{\mu}_1, I_2)$$

where $\boldsymbol{\mu}_0 = (-2, -2, -2, -2)^T$ and $\boldsymbol{\mu}_1 = (2, 2, 2, 2)^T$. The remaining 96 variables were created according to the model

$$\mathbf{x}_{[5:100]} = \mathbf{x}_{[1:4]}\beta + \varepsilon,$$

where $\varepsilon \sim N(0, \Omega)$. Different settings for β and Ω define different scenarios: in Model 1, the **irrelevant** variables have been simulated independently of the relevant ones; in Models 2 to 4, in addition to the relevant and irrelevant variables, an increasing number of **redundant** features has been included in the data generating process. The values of the parameters for all the models are reported below. Each non-null entry of β was randomly sampled with replacement from the sequence $seq = (0.00, 0.05, 0.10, 0.15, 0.20, 0.25)$; every element of seq has the same probability of being chosen, 0.08, except for the 0.00 that is selected with probability 0.6.

Model 1

$$\beta = \mathbf{0}_{96} \qquad \Omega = I_{96}$$

Model 2

$$\beta = \left[\begin{array}{cccc|c} \beta_{1,1} & \beta_{1,2} & \dots & \beta_{1,24} & \\ \vdots & & & \vdots & \mathbf{0}_{72} \\ \beta_{4,1} & \beta_{4,2} & \dots & \beta_{4,24} & \end{array}\right] \qquad \Omega = I_{96}$$

Model 3

$$\beta = \left[\begin{array}{cccccc|c} \beta_{1,1} & \beta_{1,2} & \dots & \beta_{1,24} & \dots & \beta_{1,48} & \\ \vdots & & & & & \vdots & \mathbf{0}_{48} \\ \beta_{4,1} & \beta_{4,2} & \dots & \beta_{4,24} & \dots & \beta_{4,48} & \end{array}\right] \qquad \Omega = \begin{bmatrix} I_{24} & & \\ & 0.5I_{24} & \\ & & I_{48} \end{bmatrix}$$

Model 4

$$\beta = \begin{bmatrix} \beta_{1,1} & \beta_{1,2} & \dots & \beta_{1,24} & \dots & \beta_{1,48} & \dots & \beta_{1,96} \\ \vdots & & & & & & & \vdots \\ \beta_{4,1} & \beta_{4,2} & \dots & \beta_{4,24} & \dots & \beta_{4,48} & \dots & \beta_{4,96} \end{bmatrix} \qquad \Omega = \begin{bmatrix} I_{24} & & \\ & 0.5I_{48} & \\ & & I_{24} \end{bmatrix}$$

Table 2.9 shows the number of relevant variables detected from the two approaches as the first p^* ones over 100 simulations. In Tables 2.10-2.11, for all the discussed methods, the number of variables h used for the final computation, the accuracy rates and the standard errors of 100 repetitions are compared.

Model	*d*	*Method*	*Results for RP-VIP*				*Results for AA-RP*					
			$n_{Tr} = 50$		$n_{Tr} = 100$		$n_{Tr} = 50$			$n_{Tr} = 100$		
			1-3	4	1-3	4	0	1-3	4	0	1-3	4
1	2											
		LDA		100%		100%		30%	70%		27%	73%
		QDA		100%		100%		25%	75%		23%	77%
		Knn		100%		100%		17%	83%		24%	76%
1	5											
		LDA		100%		100%		51%	49%		60%	40%
		QDA		100%		100%		37%	63%		53%	47%
		Knn		100%		100%		65%	35%		55%	45%
2	2											
		LDA		100%		100%		31%	69%		24%	76%
		QDA		100%		100%		34%	66%		21%	79%
		Knn		100%		100%		17%	83%		12%	88%
2	5											
		LDA		100%		100%		38%	62%		62%	38%
		QDA		100%		100%		29%	71%		43%	57%
		Knn	1%	99%		100%		59%	41%		56%	44%
3	2											
		LDA		100%		100%		59%	41%		32%	68%
		QDA	1%	99%		100%		55%	45%		31%	69%
		Knn	2%	98%		100%		46%	54%		22%	78%
3	5											
		LDA		100%		100%		67%	33%		67%	33%
		QDA		100%		100%		66%	34%		51%	49%
		Knn		100%		100%		69%	31%		63%	37%
4	2											
		LDA	2%	98%	1%	99%		81%	19%		43%	57%
		QDA	1%	99%	1%	99%		83%	17%		47%	53%
		Knn	2%	98%	2%	98%		76%	24%		40%	60%
4	5											
		LDA	1%	99%		100%	1%	94%	6%		78%	22%
		QDA	1%	99%		100%		96%	4%		74%	26%
		Knn	2%	98%		100%	1%	91%	9%		79%	21%

Table 2.9: Number of relevant variables detected as first p^* ones on 100 repetitions of Models 1—4.

As can be noticed from Table 2.9, the VIP measure is very good in performing its tasks: namely, it is capable to almost perfectly identify the actually relevant information sources among the others. On the contrary, the procedure introduced by Gataric does not perform excellently as well: in fact, even if from the hardest scenario (Model 4) to the simplest one (Model 1) the number of relevant variables correctly recognized by the method increases, the obtained results never equal those

provided by the VIP.

Method	*Results for Model 1*				*Results for Model 2*			
	$n_{Tr} = 50$		$n_{Tr} = 100$		$n_{Tr} = 50$		$n_{Tr} = 100$	
	h		h		h		h	
LDA_2		97.94 $_{0.24}$		98.74 $_{0.10}$		98.35 $_{0.20}$		98.71 $_{0.11}$
VIP-LDA_2	3	98.98 $_{0.13}$	8	98.80 $_{0.15}$	3	98.91 $_{0.14}$	7	98.98 $_{0.11}$
AA-LDA_2	3	99.08 $_{0.13}$	4	99.19 $_{0.08}$	3	98.97 $_{0.14}$	4	99.16 $_{0.09}$
LDA_5		98.57 $_{0.19}$		98.59 $_{0.12}$		98.50 $_{0.20}$		98.60 $_{0.12}$
VIP-LDA_5	3	98.96 $_{0.14}$	8	98.94 $_{0.13}$	3	98.94 $_{0.15}$	7	99.04 $_{0.10}$
AA-LDA_5	3	98.86 $_{0.17}$	3	99.23 $_{0.07}$	4	99.00 $_{0.15}$	5	99.17 $_{0.08}$
LDA		—†		66.87 $_{0.67}$		—†		68.94 $_{0.67}$
QDA_2		98.00 $_{0.22}$		98.81 $_{0.11}$		98.21 $_{0.20}$		98.76 $_{0.11}$
VIP-QDA_2	3	98.66 $_{0.18}$	4	98.81 $_{0.16}$	3	98.75 $_{0.17}$	4	98.90 $_{0.12}$
AA-QDA_2	2	98.87 $_{0.15}$	3	99.05 $_{0.09}$	2	98.84 $_{0.15}$	3	99.03 $_{0.10}$
QDA_5		98.58 $_{0.17}$		98.77 $_{0.12}$		98.70 $_{0.18}$		98.59 $_{0.13}$
VIP-QDA_5	3	98.55 $_{0.17}$	4	98.83 $_{0.15}$	3	98.72 $_{0.17}$	4	98.86 $_{0.12}$
AA-QDA_5	2	98.81 $_{0.16}$	3	99.09 $_{0.09}$	2	98.78 $_{0.15}$	3	99.10 $_{0.09}$
QDA		—†		—†		—†		—†
Knn$_2$		98.10 $_{0.19}$		98.80 $_{0.11}$		98.34 $_{0.21}$		98.65 $_{0.11}$
VIP-Knn$_2$	3	98.73 $_{0.15}$	2	98.84 $_{0.11}$	3	98.78 $_{0.15}$	4	98.82 $_{0.11}$
AA-Knn$_2$	3	99.10 $_{0.14}$	2	98.88 $_{0.10}$	3	99.04 $_{0.13}$	4	98.89 $_{0.10}$
Knn$_5$		98.11 $_{0.22}$		98.83 $_{0.10}$		98.10 $_{0.22}$		98.72 $_{0.11}$
VIP-Knn$_5$	5	98.72 $_{0.15}$	2	98.81 $_{0.10}$	3	98.67 $_{0.16}$	3	98.94 $_{0.10}$
AA-Knn$_5$	5	98.91 $_{0.15}$	2	98.94 $_{0.11}$	3	99.05 $_{0.14}$	3	98.90 $_{0.10}$
Knn		98.75 $_{0.15}$		98.58 $_{0.10}$		98.82 $_{0.15}$		98.83 $_{0.11}$

Table 2.10: Number of selected variables h, accuracy rates and standard errors for Models 1 and 2.

Although the performances for the VIP and the AA criteria discussed for Table 2.9 are noticeably different, as can be seen in Tables 2.10-2.11, the classification accuracies yielded by the two methods are comparable at all. In fact, even not being able to perfectly distinguish the relevant information from the rest, the AA-RP classifier attains competitive results in terms of unit allocation. A possible explanation for this apparently contradictory outcome could be found in the use, by the AA-RP ensemble classifier, of sets of both relevant and/or redundant variables when it tackles classification problems.

2.5.2.2 Real data examples

Performances of the VIP criterion have been evaluated in real data applications, too. The RP-VIP classification accuracy has been tested on the same datasets discussed in Section 2.4.2.2; each dataset was split into training and test sets of size respectively n_{Tr} and $n_{Te} = n - n_{Tr}$ (or, where available, a subsample of size 1000). Although the main aim of ranking the input features in terms of their discriminative power rests in a better understanding of the classification problem, results presented in Tables 2.12-2.15 clearly show that, in RP ensemble context, this procedure allows

Method	*Results for Model 3*				*Results for Model 4*			
	$n_{Tr} = 50$		$n_{Tr} = 100$		$n_{Tr} = 50$		$n_{Tr} = 100$	
	h		h		h		h	
LDA_2		98.52 $_{0.16}$		98.76 $_{0.11}$		98.91 $_{0.15}$		98.83 $_{0.11}$
VIP-LDA_2	3	98.96 $_{0.13}$	9	99.01 $_{0.11}$	3	99.00 $_{0.13}$	7	99.03 $_{0.12}$
AA-LDA_2	3	99.07 $_{0.12}$	3	99.21 $_{0.08}$	3	98.98 $_{0.14}$	4	99.15 $_{0.09}$
LDA_5		98.84 $_{0.14}$		98.73 $_{0.11}$		99.09 $_{0.13}$		99.01 $_{0.10}$
VIP-LDA_5	3	98.99 $_{0.12}$	7	98.99 $_{0.11}$	3	98.97 $_{0.13}$	7	99.07 $_{0.09}$
AA-LDA_5	3	99.02 $_{0.15}$	4	99.12 $_{0.10}$	4	98.67 $_{0.17}$	4	99.05 $_{0.09}$
LDA		—†		67.33 $_{0.75}$		—†		65.99 $_{0.80}$
QDA_2		98.33 $_{0.18}$		98.73 $_{0.11}$		98.90 $_{0.15}$		98.84 $_{0.10}$
VIP-QDA_2	3	98.53 $_{0.16}$	5	98.88 $_{0.13}$	3	98.74 $_{0.15}$	5	98.84 $_{0.14}$
AA-QDA_2	2	98.69 $_{0.15}$	3	98.99 $_{0.10}$	3	98.87 $_{0.15}$	3	99.05 $_{0.09}$
QDA_5		98.80 $_{0.12}$		98.70 $_{0.11}$		99.01 $_{0.13}$		98.94 $_{0.10}$
VIP-QDA_5	3	98.41 $_{0.22}$	4	98.87 $_{0.13}$	3	98.57 $_{0.17}$	5	98.73 $_{0.18}$
AA-QDA_5	3	98.75 $_{0.15}$	3	99.10 $_{0.09}$	2	98.71 $_{0.16}$	3	98.91 $_{0.14}$
QDA		—†		—†		—†		—†
Knn_2		98.29 $_{0.18}$		98.70 $_{0.12}$		98.69 $_{0.18}$		98.82 $_{0.10}$
VIP-Knn_2	2	98.77 $_{0.13}$	3	98.85 $_{0.10}$	3	98.69 $_{0.15}$	3	98.88 $_{0.10}$
AA-Knn_2	2	99.01 $_{0.13}$	3	98.88 $_{0.10}$	3	98.96 $_{0.15}$	3	98.94 $_{0.10}$
Knn_5		98.81 $_{0.14}$		98.72 $_{0.11}$		99.11 $_{0.13}$		98.88 $_{0.10}$
VIP-Knn_5	2	98.68 $_{0.15}$	3	98.89 $_{0.10}$	2	98.81 $_{0.15}$	2	98.89 $_{0.10}$
AA-Knn_5	2	99.02 $_{0.13}$	3	98.89 $_{0.10}$	2	98.73 $_{0.18}$	2	98.82 $_{0.11}$
Knn		98.89 $_{0.13}$		98.86 $_{0.10}$		98.89 $_{0.14}$		98.96 $_{0.09}$

Table 2.11: Number of selected variables h, accuracy rates and standard errors for Models 3 and 4.

to still preserve the classification accuracy.

The variable selection process for both the RP-VIP and the AA-RP ensemble classifiers is illustrated in the plots of Appendix B.1. In particular, as can be noticed in Figures B.7-B.84, the variable ranking induced by the two procedures is notably different and the solution provided by the VIP seems to be much more stable than that obtained by performing a majority vote across the axis-aligned projections. From the plots, in fact, it is evident that while the accuracy of the RP-VIP tends to increase only until all the (potentially) most important variables have been considered, the one of the AA-RP presents alternate peaks, especially when used in conjunction with LDA or QDA. In addition, in some cases of the mice, hill-valley, cardiac and human activity recognition datasets, the proposal of Gataric in Equation 2.3 cannot even be performed: namely, being some axis-aligned projected data collinear, both the linear and quadratic discriminant analyses become totally unfeasible (see Section 2.1).

Method	*Results for eye state data, $p = 14$*						*Results for ionosphere data, $p = 32$*					
	$n_{Tr} = 50$		$n_{Tr} = 200$		$n_{Tr} = 1000$		$n_{Tr} = 50$		$n_{Tr} = 100$		$n_{Tr} = 200$	
	h		h		h		h		h		h	
LDA_2		58.22 $_{0.30}$		61.05 $_{0.23}$		62.96 $_{0.24}$		86.40 $_{0.49}$		89.16 $_{0.31}$		90.26 $_{0.30}$
VIP-LDA_2	13	57.52 $_{0.35}$	13	60.99 $_{0.22}$	14	63.06 $_{0.19}$	26	76.14 $_{0.41}$	31	81.48 $_{0.32}$	29	84.61 $_{0.33}$
AA-LDA_2	2	53.02 $_{0.43}$	2	58.27 $_{0.26}$	6	59.23 $_{0.23}$	31	76.40 $_{0.44}$	20	81.75 $_{0.29}$	2	82.02 $_{0.31}$
LDA_5		58.23 $_{0.34}$		61.26 $_{0.24}$		63.74 $_{0.16}$		86.80 $_{0.37}$		89.06 $_{0.28}$		89.79 $_{0.30}$
VIP-LDA_5	14	57.37 $_{0.36}$	10	57.50 $_{0.23}$	14	63.06 $_{0.19}$	26	77.69 $_{0.40}$	32	81.73 $_{0.33}$	32	84.42 $_{0.35}$
AA-LDA_5	2	55.28 $_{0.45}$	3	54.64 $_{0.34}$	2	58.82 $_{0.22}$	2	82.25 $_{0.27}$	3	82.54 $_{0.25}$	27	84.60 $_{0.30}$
LDA		57.37 $_{0.36}$		60.74 $_{0.23}$		63.06 $_{0.19}$		76.28 $_{0.44}$		81.73 $_{0.33}$		84.42 $_{0.35}$
QDA_2		59.28 $_{0.34}$		63.83 $_{0.39}$		64.51 $_{0.74}$		90.08 $_{0.40}$		92.81 $_{0.23}$		93.80 $_{0.23}$
VIP-QDA_2	12	60.78 $_{0.32}$	14	70.90 $_{0.37}$	14	71.58 $_{0.99}$	10	84.86 $_{0.42}$	15	88.22 $_{0.28}$	30	86.56 $_{0.35}$
AA-QDA_2	9	58.23 $_{0.33}$	7	62.90 $_{0.37}$	6	58.64 $_{0.56}$	11	83.43 $_{0.51}$	7	90.08 $_{0.31}$	2	85.21 $_{0.29}$
QDA_5		60.98 $_{0.36}$		67.92 $_{0.40}$		69.84 $_{0.85}$		89.73 $_{0.48}$		93.61 $_{0.22}$		94.65 $_{0.22}$
VIP-QDA_5	14	60.24 $_{0.36}$	12	69.73 $_{0.39}$	13	71.06 $_{1.01}$	13	80.56 $_{0.65}$	19	85.60 $_{0.36}$	31	86.25 $_{0.35}$
AA-QDA_5	8	59.90 $_{0.34}$	2	59.96 $_{0.46}$	3	62.80 $_{0.88}$	3	86.40 $_{0.30}$	3	84.90 $_{0.29}$	3	89.67 $_{0.31}$
QDA		60.24 $_{0.36}$		70.90 $_{0.36}$		71.58 $_{0.99}$		—†		80.62 $_{0.61}$		85.93 $_{0.37}$
Knn_2		60.16 $_{0.34}$		68.53 $_{0.25}$		75.60 $_{0.19}$		88.41 $_{0.44}$		92.90 $_{0.26}$		94.26 $_{0.23}$
VIP-Knn_2	5	59.11 $_{0.29}$	10	69.21 $_{0.22}$	5	75.39 $_{0.14}$	23	79.21 $_{0.68}$	8	81.43 $_{0.38}$	6	84.93 $_{0.35}$
AA-Knn_2	5	59.33 $_{0.33}$	10	68.67 $_{0.24}$	5	77.28 $_{0.15}$	23	69.27 $_{0.71}$	8	85.28 $_{0.34}$	6	88.19 $_{0.31}$
Knn_5		60.39 $_{0.34}$		72.90 $_{0.24}$		86.22 $_{0.15}$		87.15 $_{0.45}$		92.63 $_{0.27}$		94.50 $_{0.23}$
VIP-Knn_5	11	55.01 $_{0.31}$	14	70.96 $_{0.24}$	2	57.91 $_{0.16}$	9	77.97 $_{0.54}$	16	81.86 $_{0.40}$	3	79.85 $_{0.41}$
AA-Knn_5	11	59.51 $_{0.32}$	14	70.96 $_{0.24}$	2	63.70 $_{0.18}$	9	80.06 $_{0.53}$	16	82.47 $_{0.41}$	3	89.32 $_{0.29}$
Knn		59.18 $_{0.28}$		70.95 $_{0.24}$		85.60 $_{0.12}$		78.27 $_{0.74}$		81.93 $_{0.48}$		83.65 $_{0.38}$

Table 2.12: Number of selected variables h, accuracy rates and standard errors for the eye state and ionosphere data.

2.6 Discussion and extensions

The RP ensemble classifier introduced by Cannings and Samworth in [20] seems to be a promising and very general tool for solving binary classification tasks. In particular, their idea to use RPs in the ensemble context successfully introduces *diversity* into the classification solution and, thus, significantly improves the ensemble accuracy.

In this work, two different directions for the RP ensemble classifier are investigated: on one hand, an Ensemble Selection Algorithm (ESA) is introduced with the aim of identifying the most accurate subset of classifiers, by combining the original idea of using the Multiplicative Binomial distribution (MB) as the reference model to describe and predict the ensemble accuracy with an important result on such distribution. On the other, inspired by the Random Forest (RF) process for feature selection, the RP ensemble classifier is adjusted so as to provide a variable ranking through a specific coefficient called Variable Importance in Projection (VIP). The innovative contribution of these two solutions does not rely on the algorithmic procedure, rather on the introduction of novel criteria that enhance the results, in terms of both accuracy and understanding.

Results of applying both the ESA and the VIP criterion in simulated and real data

Method	*Results for mice data,* $p = 68$						*Results for hill-valley data,* $p = 100$					
	$n_{Tr} = 200$		$n_{Tr} = 500$		$n_{Tr} = 1000$		$n_{Tr} = 100$		$n_{Tr} = 200$		$n_{Tr} = 500$	
	h		h		h		h		h		h	
LDA_2		68.86 $_{0.31}$		70.59 $_{0.30}$		70.22 $_{0.59}$		56.16 $_{0.47}$		58.46 $_{0.55}$		61.55 $_{0.57}$
VIP-LDA_2	61	93.65 $_{0.13}$	56	95.77 $_{0.10}$	67	96.78 $_{0.21}$	95	62.52 $_{0.48}$	29	65.41 $_{0.50}$	90	66.71 $_{0.43}$
AA-LDA_2	25	92.84 $_{0.11}$	28	93.54 $_{0.11}$	7	92.00 $_{0.26}$	21	62.36 $_{0.51}$	20	61.94 $_{0.49}$	6	61.84 $_{0.39}$
LDA_5		74.46 $_{0.32}$		76.42 $_{0.28}$		77.03 $_{0.49}$		62.65 $_{0.84}$		65.55 $_{0.91}$		68.38 $_{0.97}$
VIP-LDA_5	58	93.87 $_{0.13}$	58	95.96 $_{0.10}$	66	96.78 $_{0.21}$	99	62.27 $_{0.50}$	55	65.34 $_{0.50}$	7	65.56 $_{0.45}$
AA-LDA_5	61	92.83 $_{0.15}$	27	93.86 $_{0.10}$	7	90.14 $_{0.30}$	27	63.71 $_{0.47}$	97	63.57 $_{0.48}$	23	64.93 $_{0.44}$
LDA		93.49 $_{0.14}$		95.98 $_{0.10}$		96.76 $_{0.21}$		62.36 $_{0.51}$		63.34 $_{0.45}$		66.19 $_{0.38}$
QDA_2		73.64 $_{0.31}$		75.55 $_{0.27}$		75.17 $_{0.53}$		53.50 $_{0.30}$		55.24 $_{0.32}$		58.22 $_{0.35}$
VIP-QDA_2	33	93.22 $_{0.18}$	38	98.08 $_{0.07}$	50	99.13 $_{0.11}$	12	54.64 $_{0.32}$	7	56.25 $_{0.303}$	10	56.36 $_{0.31}$
AA-QDA_2		—†		—†		—†	2	50.16 $_{0.21}$	2	51.42 $_{0.24}$	3	53.09 $_{0.28}$
QDA_5		81.17 $_{0.29}$		83.59 $_{0.23}$		84.30 $_{0.44}$		55.09 $_{0.42}$		57.81 $_{0.44}$		60.84 $_{0.49}$
VIP-QDA_5	27	91.36 $_{0.20}$	37	97.82 $_{0.08}$	52	99.16 $_{0.11}$	11	51.64 $_{0.23}$	29	52.35 $_{0.24}$	18	54.97 $_{0.27}$
AA-QDA_5		—†		—†		—†		—†		—†	5	53.22 $_{0.27}$
QDA		—†		—†		—†		—†		—†		—†
Knn$_2$		87.72 $_{0.36}$		92.50 $_{0.23}$		96.73 $_{0.22}$		52.11 $_{0.25}$		57.18 $_{0.34}$		72.69 $_{0.39}$
VIP-Knn$_2$	7	73.45 $_{0.23}$	5	78.28 $_{0.23}$	3	76.65 $_{0.48}$	3	50.24 $_{0.24}$	12	51.13 $_{0.23}$	4	53.68 $_{0.22}$
AA-Knn$_2$	7	81.13 $_{0.22}$	5	91.01 $_{0.15}$	3	83.53 $_{0.43}$	3	50.39 $_{0.23}$	12	51.01 $_{0.24}$	4	53.68 $_{0.22}$
Knn$_5$		87.94 $_{0.28}$		92.42 $_{0.10}$		99.49 $_{0.09}$		50.82 $_{0.23}$		52.90 $_{0.24}$		64.19 $_{0.23}$
VIP-Knn$_5$	5	71.53 $_{0.22}$	4	73.44 $_{0.22}$	8	92.79 $_{0.33}$	57	50.52 $_{0.24}$	8	50.94 $_{0.23}$	38	55.36 $_{0.24}$
AA-Knn$_5$	5	82.72 $_{0.20}$	4	88.18 $_{0.17}$	8	97.95 $_{0.19}$	57	50.53 $_{0.24}$	8	51.86 $_{0.22}$	38	55.36 $_{0.24}$
Knn		80.25 $_{0.23}$		92.97 $_{0.09}$		99.12 $_{0.12}$		50.62 $_{0.24}$		51.41 $_{0.23}$		53.34 $_{0.21}$

Table 2.13: Number of selected variables h, accuracy rates and standard errors for the mice and hill-valley data.

demonstrate that our proposals successfully control the misclassification rate by using a very small number of individual classifiers and by ranking the features in terms of their discriminative power.

Although preliminary results are good, we are almost certain that a further research could provide even additional enhancements. In particular, for the ESA, we are aware that the forward search-based procedure is quite complex - $\mathcal{O}(B_1^3)$ - and does not guarantee the optimality of the combination found.

Method	*Results for musk data, $p = 166$*						*Results for cardiac data, $p = 190$*					
	$n_{Tr} = 100$		$n_{Tr} = 200$		$n_{Tr} = 500$		$n_{Tr} = 50$		$n_{Tr} = 100$		$n_{Tr} = 200$	
	h		h		h		h		h		h	
LDA_2		84.29 $_{0.20}$		85.27 $_{0.16}$		85.79 $_{0.18}$		65.67 $_{0.45}$		68.95 $_{0.36}$		70.87 $_{0.36}$
VIP-LDA_2	62	76.30 $_{0.41}$	107	82.52 $_{0.23}$	157	91.11 $_{0.11}$	43	55.31 $_{0.48}$	83	57.36 $_{0.42}$	166	57.63 $_{0.62}$
AA-LDA_2	2	83.97 $_{0.21}$	2	84.62 $_{0.12}$	5	84.86 $_{0.11}$		—†		—†		—†
LDA_5		84.87 $_{0.32}$		87.80 $_{0.18}$		89.85 $_{0.12}$		66.71 $_{0.47}$		70.06 $_{0.33}$		72.29 $_{0.35}$
VIP-LDA_5	50	78.82 $_{0.37}$	118	81.70 $_{0.24}$	153	91.18 $_{0.11}$	37	57.10 $_{0.50}$	81	57.55 $_{0.41}$	179	57.89 $_{0.60}$
AA-LDA_5	24	83.68 $_{0.25}$	121	84.25 $_{0.23}$	2	84.65 $_{0.11}$		—†		—†		—†
LDA		—†		74.66 $_{0.41}$		90.93 $_{0.11}$		—†	190	—†		53.97 $_{0.25}$
QDA_2		84.98 $_{0.28}$		86.73 $_{0.20}$		87.94 $_{0.18}$		67.11 $_{0.39}$		70.15 $_{0.33}$		71.22 $_{0.36}$
VIP-QDA_2	15	82.06 $_{0.24}$	24	84.72 $_{0.17}$	67	86.46 $_{0.14}$	9	60.02 $_{0.39}$	31	60.23 $_{0.46}$	66	66.93 $_{0.44}$
AA-QDA_2	2	83.65 $_{0.21}$	2	84.06 $_{0.14}$	2	84.76 $_{0.11}$		—†		—†		—†
QDA_5		87.77 $_{0.25}$		89.28 $_{0.20}$		91.24 $_{0.13}$		63.23 $_{0.58}$		70.16 $_{0.34}$		72.75 $_{0.34}$
VIP-QDA_5	12	79.47 $_{0.51}$	31	85.55 $_{0.13}$	58	87.36 $_{0.15}$	15	56.83 $_{0.41}$	43	55.89 $_{0.61}$	69	65.64 $_{0.46}$
AA-QDA_5	7	81.74 $_{0.26}$	2	84.90 $_{0.13}$	2	84.65 $_{0.11}$		—†		—†		—†
QDA		—†		—†		—†		—†	190	—†		—†
Knn$_2$		86.56 $_{0.30}$		88.74 $_{0.23}$		91.17 $_{0.12}$		66.27 $_{0.47}$		69.70 $_{0.34}$		71.98 $_{0.33}$
VIP-Knn$_2$	6	83.07 $_{0.27}$	2	84.80 $_{0.17}$	3	84.99 $_{0.12}$	5	58.20 $_{0.40}$	8	59.39 $_{0.36}$	2	53.95 $_{0.36}$
AA-Knn$_2$	6	84.27 $_{0.24}$	2	87.75 $_{0.17}$	3	90.40 $_{0.13}$	5	58.96 $_{0.38}$	8	68.58 $_{0.31}$	2	59.65 $_{0.37}$
Knn$_5$		87.65 $_{0.31}$		89.87 $_{0.22}$		91.91 $_{0.13}$		65.81 $_{0.49}$		69.71 $_{0.35}$		72.46 $_{0.34}$
VIP-Knn$_5$	45	85.12 $_{0.21}$	4	85.82 $_{0.26}$	3	85.65 $_{0.12}$	43	59.70 $_{0.39}$	6	58.90 $_{0.37}$	14	60.25 $_{0.34}$
AA-Knn$_5$	45	86.29 $_{0.26}$	4	86.61 $_{0.20}$	3	90.26 $_{0.12}$	43	61.45 $_{0.46}$	6	58.54 $_{0.37}$	14	67.13 $_{0.36}$
Knn		85.96 $_{0.25}$		88.89 $_{0.15}$		91.83 $_{0.13}$		59.79 $_{0.38}$		61.80 $_{0.35}$		64.75 $_{0.34}$

Table 2.14: Number of selected variables h, accuracy rates and standard errors for the musk and cardiac arrhythmia data.

Method	*Results for human activity recognition data, $p = 561$*					
	$n_{Tr} = 50$		$n_{Tr} = 200$		$n_{Tr} = 1000$	
	h		h		h	
LDA_2		99.85 $_{0.01}$		99.92 $_{0.01}$		100.00 $_{0.00}$
VIP-LDA_2	2	100.00 $_{0.00}$	2	100.00 $_{0.00}$	2	100.00 $_{0.00}$
AA-LDA_2	2	99.33 $_{0.07}$	2	100.00 $_{0.00}$	3	100.00 $_{0.00}$
LDA_5		99.82 $_{0.02}$		99.90 $_{0.01}$		100.00 $_{0.00}$
VIP-LDA_5	2	100.00 $_{0.00}$	2	100.00 $_{0.00}$	2	100.00 $_{0.00}$
AA-LDA_5	3	99.33 $_{0.06}$	2	99.94 $_{0.01}$	2	100.00 $_{0.00}$
LDA		—†		—†		100.00 $_{0.00}$
QDA_2		99.80 $_{0.02}$		99.91 $_{0.01}$		100.00 $_{0.00}$
VIP-QDA_2	2	99.96 $_{0.01}$	2	99.98 $_{0.00}$	2	99.99 $_{0.00}$
AA-QDA_2		—†		—†		—†
QDA_5		99.83 $_{0.02}$		99.88 $_{0.01}$		99.99 $_{0.00}$
VIP-QDA_5	2	99.96 $_{0.01}$	2	99.98 $_{0.00}$	2	100.00 $_{0.00}$
AA-QDA_5		—†		—†		—†
QDA		—†		—†		—†
Knn$_2$		99.81 $_{0.02}$		99.91 $_{0.01}$		100.00 $_{0.00}$
VIP-Knn$_2$	2	99.47 $_{0.04}$	2	100.00 $_{0.00}$	2	100.00 $_{0.00}$
AA-Knn$_2$	2	99.46 $_{0.04}$	2	99.83 $_{0.02}$	2	99.98 $_{0.01}$
Knn$_5$		99.80 $_{0.02}$		99.90 $_{0.01}$		100.00 $_{0.00}$
VIP-Knn$_5$	2	100.00 $_{0.00}$	2	100.00 $_{0.00}$	2	100.00 $_{0.00}$
AA-Knn$_5$	2	98.63 $_{0.05}$	2	98.72 $_{0.04}$	2	100.00 $_{0.00}$
Knn		99.74 $_{0.02}$		99.87 $_{0.01}$		99.98 $_{0.00}$

Table 2.15: Number of selected variables h, accuracy rates and standard errors for the human activity recognition data.

Chapter 3

One-class classification

3.1 Introduction

As widely discussed in 2.1, the typical problem of classification is to assign a new object to one of a set of classes which are known in advance. But how can this procedure be performed if the information on only one of the classes (the *target class*) is available? How can a boundary around this class be defined so as to recognize as much of the target objects as possible while minimizing the chance of error? Circumstances of this kind characterize various contexts including machine fault and fraud detection, food authentication and medical or machine diagnostics.

In order to better explain the peculiarities of these situations, let's consider two simple examples.

Firstly, imagine that you are holding a beef burger in you hand, wandering if you are going to eat wealthy enough. Would you be able to recognize whether the meet in your burger is "100% *pure* beef" as claimed?
Then, change scenario and suppose that you are an art curator who has been asked to evaluate a painting in terms of its state of preservation. Could you give an advice on ways to restore or maintain the artwork in good conditions, on the basis of the specific "ingredients" it shall be composed of? Although it may seem so, the answer to these questions is not trivial. Surely, you are supposed to know the color, the shape, the smell and the flavor of a *true* beef burger; similarly, depending both on the the painting technique adopted and the historic period the piece of art dates back, its *pure* chemical composition is presumed to be given. However, the ways

both the burger and the painting might be contaminated are countless and mostly unpredictable.
By their nature, these issues could be read as typical **one-class classification** problems [86] and they are usually addressed by either resorting to distance-based or to density-based methods.

In this work, a new statistical approach for one-class classification based on Gini's definition of **transvariation probability** between a group and a constant is proposed. In particular, we refer to the concept of transvariation and some of its related measures, firstly introduced in an univariate context by Gini in 1916 [48] and, subsequently, extended to the multivariate case and to a model-based formulation by Gini and Livada [49] and Dagum [28].

The remainder of this chapter is organized as follow. Section 2 formalizes the one-class classification problem and provides a detailed taxonomy of the existing methods; then, in the same section, the definition of transvariation probability between a group and a constant, both in the univariate and the multivariate contexts, is presented. In Section 3, a novel **transvariation-based one-class classification algorithm** is introduced and some technical aspects, including dimension reduction or variable selection procedures, are discussed. In Section 4, the methodology is tested and its performances are evaluated in both simulated and real data. A final discussion on the obtained results and possible extensions is included in Section 5.

3.2 Theoretical background

3.2.1 What is one-class classification?

In order to fully understand what one-class classification is and why it is different from other well known classification tasks, let's consider, as an example, the study on physical measurements (i.e. weight and height) of a set of individuals shown in Figure 3.1.
In subfigure (a), the typical one-class classification problem is presented. Particularly, with the aim to describe the observed individuals (target class) and to detect which (new) observations resemble them in terms of weight and height, a clear boundary around this set is defined and, on the basis of it, each new unit (e.g. the triangle) classified. Moving then to subfigure (b), the world of standard classifica-

tion (namely, the Quadratic Discriminant Analysis, QDA) is described. In this case, differently from the one-class, the set of individuals is divided in a given number of classes (e.g two: adult and children) according to their physical characteristics and the goal is to learn how to assign each (new) unit to the most likely class, while minimizing the error. Finally, subfigure (c) show the outlier detection issue and it points out how this problem might be similar, even if not identical to the one-class classification one. In fact, while the training set for the outlier detection is naturally polluted by deviant observations, that used to train the one-class classifier does not include any outlier and all the anomalies should be recognized only in new observations.

The classic one-class classification methodologies always identify two distinct elements:

- a **distance** (or **resemblance**, or probability) measure of a new object $\mathbf{z}$ to the target class $\chi \in \mathbb{R}_{n \times p}$;
- a **threshold**, $\mathbf{t}$, for this measure.

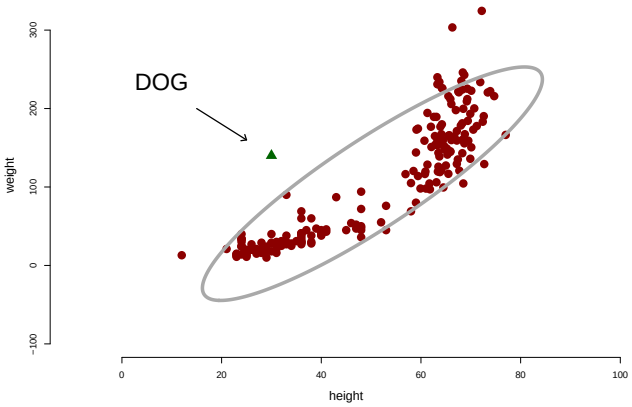
In particular, a new object $\mathbf{z}$ is classified as a target class one only if the distance measure $d(\mathbf{z})$ is smaller than a given threshold $\mathbf{t}_d$, $d(\mathbf{z}) < \mathbf{t}_d$, or, equivalently, if the resemblance measure $f(\mathbf{z})$ is bigger than the threshold $\mathbf{t}_f$, $f(\mathbf{z}) > \mathbf{t}_f$.

In the one-class classification framework, there is no way to a priori assess the false positive rate as no examples from the outlier class are, in principle, available. Therefore, in this case, only the number of objects of the target class that are wrongly attributed to the outlier group (false negatives) can be controlled.

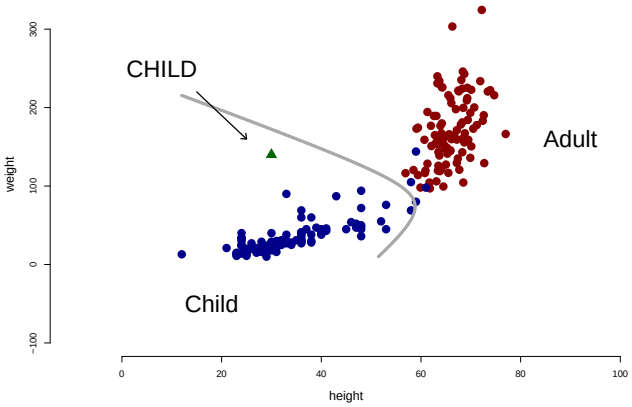
3.2.2 Taxonomy of one-class classifiers and methods comparison

In the context of one-class classification, different algorithms, methodologies and procedures have been proposed. According to the internal model used as classifier, all these techniques could be grouped in three different categories: *density* methods estimate the probability density function in the complete feature space, *boundary* methods aim to define the best boundary[1] around the training data and *reconstruction* methods assume a data generating process and evaluate the fit of each

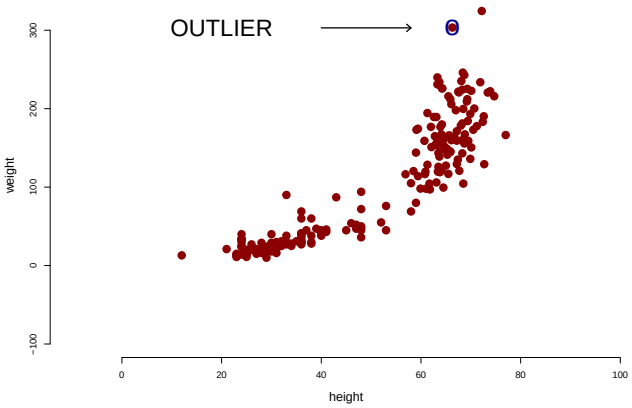
[1] the boundary which maximizes the probability of accepting a target object while minimizing the chance of error



(a)
One-class classification



(b)
QDA



(c)
Outlier detection

Figure 3.1: Different classification algorithms performed on the same sample of 198 human weights and heights.

observation with respect to that model.

Although the statistical literature provides several approaches to address the one-class problem, no method has been shown to consistently outperform (or underperform) the others. In particular, the comparison between the different one-class classifiers is typically based on the following criteria:

- **Behavior with respect to the outliers**: one-class classification methods should be able to accept as many objects from the target class as possible, while rejecting all the observations which might contaminate the training set. In this sense, the preference is for all the models that, with the aim to improve the classification rule, recognize and, then, use noise objects to give a more precise description of the target data.

- **Number of parameters** to be estimated or set by the user: the number of free parameters that should be decided beforehand, along with their initial values, have a strong effect on the final performances of any one-class classifier. Since no clear rule is provided, any method might completely fail when the user decision is not correct. Therefore, models with a small number of free parameters should be preferred.

- **Computational and storage requirements**: although the computational power and the storage capacity accommodated by the new computing devices is constantly increasing, several aspects that limit the applicability of some methods to real contexts still exist (e.g. the implementation of adaptive models to new settings could be computationally intractable). Procedures computationally easy and low-demanding in terms of storage space are the favored.

- **Accuracy rates**: this requirement represents the most important aspect in evaluating a one-class classifier, even though the less trivial to measure, as it requires the *true* class label for each object to be computed. In particular, while in a simulation study the true membership of the data is given, in real applications such information is not known *apriori* and, thus, it should be derived from past observations (which are not necessarily similar to new ones). According to this criterion, the method which obtains the best trade-off between the fraction of target objects s (*sensitivity*, $1 - e_I$) and the fraction of

outliers s^- (*specificity*, $1-e_{II}$)[2] correctly recognized is to be preferred. Usually, as only characteristics of the target class χ are given, s is fixed so as

$$s := \begin{cases} \dfrac{\#(d(\mathbf{x}) < \mathbf{t}_d)}{n} & \text{if } d \text{ is a distance measure} \\ \dfrac{\#(f(\mathbf{x}) > \mathbf{t}_f)}{n} & \text{if } f \text{ is a resemblance measure} \end{cases}, \mathbf{x} \in \chi \text{ and } |\chi| = n$$

and, then, methods are compared in terms of s^-.

When all the possible combinations of s and s^- for varying $\mathbf{t}$ values are computed, the Receiver-Operating Characteristic (ROC) curve [83] can be obtained. Specifically, such a curve allows to define a complete sensitivity/specificity report for each model and, therefore, it helps in comparing different methods in terms of classification performances. Figure 3.2 shows the ROC space, defined by s (axis of ordinates) and s^- (axis of abscissae). The *perfect classification* representing 100% sensitivity and 100% specificity is given by the point with coordinates (1,1) in the ROC space; the random guess, instead, is described by the dashed diagonal line drawn from the top-left to the bottom-right corner of the plot. Points below the diagonal represent a classification that is significantly better than a random result, whilst points above the line describe poor results (worse than those obtained by chance).

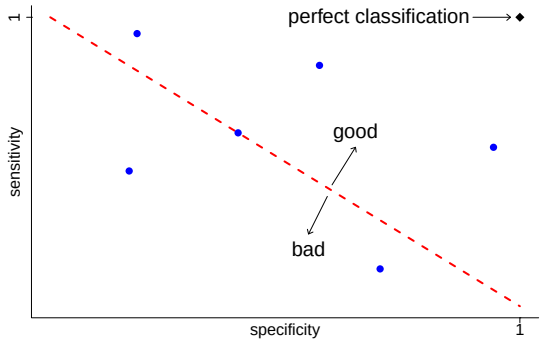


Figure 3.2: The ROC space and plots of some prediction examples.

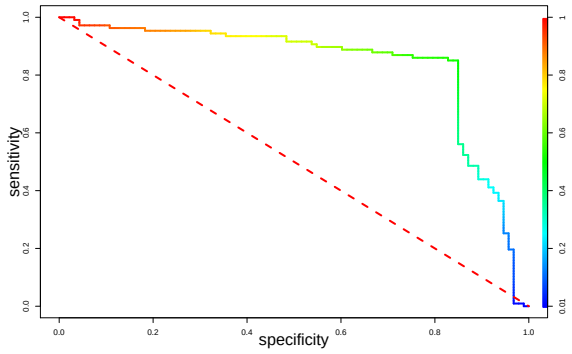


Figure 3.3: ROC curve of a simple artificial prediction data.

3.2.2.1 Density methods

The most straightforward approach for one-class classification is probably the idea of modeling the training set by using a probability density function (pdf), e.g. a

[2] e_I and e_{II} are respectively the Type I and Type II errors.

Gaussian or a Poisson distribution. The aim of density-based methods, in fact, is to estimate the density of the target class χ, $f(\mathbf{x})$ with $\mathbf{x} \in \chi$, and to set a threshold, $\mathbf{t}_f$, on the resulting densities.
These techniques usually work very well, especially when the sample size is sufficiently large and the model assumed to describe the target distribution is appropriate. However, since such a choice is not trivial and it requires a large number of training objects to overcome the *curse of dimensionality* [7], their actual implementation could be limited. In this work, four different density estimation methods for the target class are considered: Gaussian and mixture of Gaussian models, the Kernel Density Estimation (KDE), the K-nearest-neighbors (Knn) estimation and histograms.

Gaussian and mixture of Gaussians

These methods assume that the target class χ could be well described using the p-dimensional Gaussian distribution:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\},$$

where $\boldsymbol{\mu}$ is the mean vector, Σ is the covariance matrix and p is the dimension of the feature space, or, alternatively, by resorting to a mixture of K Gaussian p-dimensional distributions $f_k(\mathbf{x}) = f_k(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k)$

$$f(\mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K} \pi_k f_k(\mathbf{x}),$$

where π_k is the prior probability that $\mathbf{x}$ is generated from the k-th component of the mixture and K is the total number of mixing components.
The mixture model is more flexible than the single Gaussian distribution and presents a better fit. However, as drawbacks, it requires more training data to be estimated, as the number of free parameters of the mixture, $n_{FreeMixG}$, is larger comparatively to that of the simple distribution, n_{FreeG}:

$$n_{FreeMixG} = \left(p + \frac{p(p+1)}{2} + 1\right) K \geq n_{FreeG} = p + \frac{p(p-1)}{2}.$$

Notice that $n_{FreeMixG}$ is often reduced by assuming just diagonal covariance matrices, i.e. $\Sigma_k = \mathrm{diag}(\sigma_k)$.

These models are characterized by both a learning and a classification process computationally inexpensive; here, the only computational effort consists in the inversion of the covariance matrix Σ. When the inverse of Σ cannot be calculated, e.g. when the data have singular directions or they are badly scaled, it should be approximated by using the pseudo-inverse matrix $\Sigma^+ = \Sigma^T(\Sigma^T\Sigma)^{-1}$ [108] or by applying a regularization constraint to Σ (e.g. by adding a user-defined constant λ to the diagonal, i.e. $\Sigma' = \Sigma + \lambda I$).
The storage space required for the learning phase is relatively large since all the training data should be used and retained; however, it could be significantly reduced by incrementally evaluating the model parameters. The storage requirements for classification, instead, are negligible.
The presence of outliers in the training set could seriously affect the model performances; the final classification accuracy, in fact, strongly depends on how well the assumed distribution (i.e. the Gaussian model) fits the target class.

Kernel density estimator

The Kernel density estimation method is essentially a data-interpolation technique that does not make any strong assumption about the shape of the data distribution. In this case, in fact, the density of the target class is directly derived from the data and it is given by:

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \varphi_H\left(\mathbf{x} - \mathbf{x}_i\right). \tag{3.1}$$

Here, n is the training set size, H is the positive definite bandwidth matrix and $\varphi_H(.) = |H|^{-1/2}\varphi(H^{-1/2}\mathbf{x})$ is the *kernel* function in the p-dimensional space.
The choice of $\varphi(.)$ in 3.1 is not crucial to the performances of f. Among all the possible alternatives, a popular solution for $\varphi(.)$ is the normal kernel,

$$\varphi(\mathbf{x}) = (2\pi)^{-p/2}|H|^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T|H|^{-1}(\mathbf{x}-\boldsymbol{\mu})\right),$$

where H plays the role of the covariance matrix. In this particular case, the Kernel density estimator could be considered as a natural extension of the Gaussian method that has been previously described.
The Kernel density model is extremely flexible and it allows to approximate arbitrary distributions for which the parametric form is unknown. In addition, it presents few free parameters, corresponding to the number of different positive entries of H. The choice of H, controlling the amount and orientation of smoothing induced, plays an

important role on the classifier performances and, therefore, it should be carefully specified. A common method to choose the optimal H is to use the bandwidth that minimizes the Approximated Mean Integrated Squared Error[3]. Based on this approach, several bandwidth selection techniques (differing from the method used to estimate the AMISE) have been proposed in the literature: among the others, the plug-in and the cross validation selectors are the the most commonly employed procedures.
For the training phase, the Kernel density estimator computational costs are limited; the testing phase, instead, is very expensive and it requires a storage space that may become even prohibitive when the number of observations, n, is large.

Knn estimation

A well-known nonparametric method for classification is the K nearest neighbors density estimator, or Knn. It is a special type of the kernel density estimation method with a local choice of the bandwidth.
In this case, the local density of a generic observation $\mathbf{x}$ is estimated by:

$$f(\mathbf{x}) = \frac{K}{nV_p r^p}, \tag{3.2}$$

where:

- n is the cardinality of the training set;
- V_p is the volume of the Euclidean p-dimensional unit ball centered in $\mathbf{x}$;
- r^p is the Euclidean distance between $\mathbf{x}$ and its K-th closest neighbor.

Equation 3.2 can be directly derived from Equation 3.1 by choosing H to be identity matrix and a kernel that is a uniform density on the p-dimensional Euclidean unit ball [59].

The Knn method needs to keep all the observation vectors during the testing phase and, therefore, it requires a very large storage space. Furthermore, the produced estimates are not true probability densities as the integral taken over all the sample space diverges.
No free parameters have to be set or estimated by the Knn method and only the number of neighbors K included in the area should be provided in advance. The

[3]It is the asymptotic approximation of the Mean Integrated Squared Error. The MISE cannot be directly used since it does not have a tractable closed form.

choice of K strongly affects the smoothing of the estimates and, consequently, the method robustness. Existing theoretical results suggest that, if $n \to \infty$ and the Euclidean or the Mahalanobis distances are used for classification, K should vary with n such that $\frac{K}{n} \to 0$. However, since no general guideline is provided for all the other scenarios, the optimal value for K is typically chosen by minimizing the cross-validation error.

Histograms

Histogram analysis is an extremely common way to perform kernel density estimation, as it could be used even in presence of symbolic data. Generally, histograms are obtained by dividing the complete feature space into non-overlapping and consecutive intervals (called *bins*) and then by counting the number of observation vectors falling in each of them. The number of bins, K, should be provided in advance and this value affects the smoothness of the estimates: a large value for K may produce a very spike density estimation, whilst a small K could provide over-smoothed estimates.

By their nature, histograms are quite resistant to noise and mislabelling errors and they are pretty inexpensive in terms of both computational and storage requirements. However, they need a large training set to overcome the curse of dimensionality and, thus, provide accurate estimates; moreover they are not smooth and, as discussed, they tend to be very sensitive to the correct choice of K.

3.2.2.2 Boundary methods

Although the density method performances are generally pretty good, when the available amount of data is limited, the kernel function produces unreliable estimates. In situations characterized by a large number of variables, p, and/or a small sample size, n, in fact, a boundary approach appears more appropriate. Methods in this category only imply the definition of the tightest boundary around the target set. The classification issue is performed by evaluating the distance of a given object from the target class and, then, by comparing it to a threshold t_d. In particular, t_d is directly derived on the distance measures and it is adjusted to ensure a predefined sensitivity, s:

$$\mathbf{t}_d : \frac{\#(d(\mathbf{x}) < \mathbf{t}_d)}{n} = s.$$

The main drawback of the boundary methods relies on their inherent sensitivity to scaling of the features, mostly due to their use of a distance measure between the

observations.
For this category of methods, the K-centers algorithm, two techniques derived from the Support Vector Classifier (ν Support Vector Classification, ν-SVC and Support Vector Data Description, SVDD) and the class of depth-based approaches are discussed in the following.

K-centers

K-centers is probably the simplest boundary method that has been proposed so far. Namely, it covers the training data with K small hyperspheres of equal radii whose centers, $\boldsymbol{\mu}_k$, are placed on the target class so as to minimize $\varepsilon_{K-\text{centers}}$ error, i.e. the maximum distance of the minimum distances between the data and the centers:

$$\varepsilon_{K-centers} = \max_i \left(\min_k ||\mathbf{x}_i - \mu_{\mathbf{k}}||^{\mathbf{2}} \right).$$

Starting from either a single or a multiple random initialization, the K-centers method uses a forward search strategy to determine the oprtimal radius of the hyperspheres. For this reason, this approach is strongly sensitive to the presence of outlier observations in the training set.
Once the K balls have been placed and the centers decided, the distances between each observation $\mathbf{x}$ and the centers $\boldsymbol{\mu}_k$ could be computed as:

$$d_{K-\text{centers}}(\mathbf{x}) = \min_k ||\mathbf{x} - \boldsymbol{\mu}_k||^2.$$

Then, the classification issue could be addressed by comparing each distance to a threshold t_d: if $d_{K-\text{centers}}(\mathbf{x}) > \mathbf{t}_d$, $\mathbf{x}$ is deemed not to belong to the target class; otherwise, $\mathbf{x}$ is considered as a target object. For the K-centers method, only K parameters (corresponding to the number of balls) have to be defined,

$$n_{FreeK-centers} = K$$

and the computational cost is very low. However, as a drawback, it may require a large memory space, due to its necessity of storing all the observation vectors during the learning phase.

ν Support Vector Classification (ν-SVC) and Support Vector Data Description (SVDD)

The the ν Support Vector Classification (ν-SVC) is a method for one-class classification proposed by proposed by Schölkopf *et al.* in 1999 as a variant of the conventional Support Vector Machine (SVM) introduced by Vapnik in [124]. In their work [100], the authors suggest to use an hyperplane, $\mathbf{w}$, in order to separate the training set from the origin with a maximum margin. In particular, the minimization problem that should be solved in order to find $\mathbf{w}$ is:

$$\min_{\mathbf{w},\rho,\xi} \frac{1}{2}||\mathbf{w}||^2 - \rho + \frac{1}{\nu n}\sum_i \xi_i \tag{3.3}$$

subject to the constraints $\mathbf{w} \cdot \mathbf{x}_i \geq \rho - \xi_i$, $\forall i$ $\xi_i \geq 0$, where ρ is the margin which separates $\mathbf{x}_i$ from the origin with error ξ and $\nu \in (0, 1)$ is a user defined parameter indicating the fraction of the data that should be separated (comparable to C in SVVD).
Given that the training data are preprocessed to have unit norm, the ν-SVC has been proven to provide good results. In this particular case, the optimization problem in 3.3 could be rewritten as:

$$\min \; R'^2 + C' \sum_{i=1}^{n} \xi'_i$$

subject to the constraint $||\mathbf{x}'_i - \mathbf{a}'||^2 \leq R'^2 + \xi'_i$, $\forall i$.

A recent alternative to ν-SVC, also inspired to SVM, is the Support Vector Data Description (SVDD), introduced by Tax and Duin in 2004 in [112]. This approach aims to find the smallest closed hypersphere (in terms of volume), rather than an hyperplane, with the highest density of training data. Specifically, such a sphere is identified so as to minimize the error function

$$\varepsilon_{SVDD} = R^2 \; , \tag{3.4}$$

subject to the constraint $||\mathbf{x}_i - \mathbf{a}||^2 \leq R^2$, $i = 1, \cdots, n$. Here, $\mathbf{a}$ is the center of the hypersphere, R is its radius and $\mathbf{x}_i$ represents the i-th sample vector from the training set.
Although the error function in 3.4 guarantees the identification of the smallest closed

sphere, it is very strict and, therefore, it does not allow the presence of training objects with large distances from the ball center, $\mathbf{a}$. In order to overcome this limit, the minimization problem could be rewritten with the inclusion of a penalization term, in analogy with 3.3:

$$\varepsilon_{SVDD} = R^2 + C\sum_{i=1}^{n}\xi_i \ , \ i = 1, \cdots, n \ , \tag{3.5}$$

subject to the constraint $||\mathbf{x}_i - \mathbf{a}||^2 \leq R^2 + \xi_i$, $\forall i$. In particular, the penalty term is composed by a set of slack variables, $\xi_i \geq 0$, and a given parameter, C, controlling the number of training vectors not covered by the sphere. Using the Lagrange multipliers, $\alpha_i \geq 0$ and $\gamma_i \geq 0$, $\forall i$, equation 3.5 and its constraint could be incorporated in

$$\varepsilon_{SVDD} = R^2 + C\sum_i \xi_i - \sum_i \alpha_i \left\{R^2 + \xi_i - (||\mathbf{x}_i||^2 - 2\mathbf{a}\mathbf{x}_i + ||\mathbf{a}||^2)\right\} - \sum_i \gamma_i\xi_i \tag{3.6}$$

This equation should be minimized with respect to $R, \mathbf{a}, \xi_i$ and maximized with respect to α_i, γ_i. By setting the partial derivatives of 3.6 to zero and substituting the resulting constraints to the same equation, the SVVD error function results as:

$$\varepsilon_{SVDD} = \sum_i \alpha_i(\mathbf{x}_i \cdot \mathbf{x}_i) - \sum_{i,j}\alpha_i\alpha_j(\mathbf{x}_i \cdot \mathbf{x}_j) \tag{3.7}$$

subject to the constraints $\sum_i \alpha_i = 1$ and $0 \leq \alpha_i \leq C$.
With the aim to provide further flexibility, Vapnik proposed to expand 3.7 by using a kernel function, $\varphi(\mathbf{x}_i \cdot \mathbf{x}_j)$ instead of a simple inner product $(\mathbf{x}_i \cdot \mathbf{x}_j)$. The use of $\varphi(.)$ allows to map the training vectors onto a higher dimensional feature space and, thus, to produce an accurate description of the target class.

In order to perform classification, the distance $d(\mathbf{x}, \mathbf{a})$ between the observation vector $\mathbf{x}$ and the center of the sphere, $\mathbf{a}$, is computed and, then, compared to the radius R. Only if such a distance is smaller (or equal) than the radius, $\mathbf{x}$ is accepted as a target :

$$d(\mathbf{x}, \mathbf{a}) = ||\mathbf{x} - \mathbf{a}||^2 = (\mathbf{x} \cdot \mathbf{x}) - 2\sum_i \alpha_i(\mathbf{x} \cdot \mathbf{x}_i) + \sum_{i,j}\alpha_i\alpha_j(\mathbf{x}_i \cdot \mathbf{x}_j) \leq R^2 \ ,$$

where $\mathbf{a} = \sum_i \alpha_i\mathbf{x}_i$ and $R^2 = (\mathbf{x}_k \cdot \mathbf{x}_k) - 2\sum_i \alpha_i(\mathbf{x}_k \cdot \mathbf{x}_i) + \sum_{i,j}\alpha_i\alpha_j(\mathbf{x}_i \cdot \mathbf{x}_j)$ ($\mathbf{x}_k$ are the *support vectors* which have $\alpha_k < C$).

Since both the the ν-SVC and the SVDD use a regularization parameter (ρ or C) to control the noisy or mislabelled vectors that should be excluded from the description, they are quite robust to the outliers. Furthermore, the classification issue is computationally simple and it does not require a large storage memory.
However, as the size of the training set, n increases, the applicability of the two methods could be seriously precluded. In these situations, in fact, the algorithm complexity could become prohibitive being n equal to the number of both the parameters to estimate and the objects to store during the learning phase.

Depth-based approaches

The concept of location depth was firstly introduced in 1975 by Tukey [120] as a graphical tool for visualizing bivariate data sets, and has since been extended to the multivariate case [34]. Different depth measures with different characteristics have been proposed [77], but all of them have the same purpose: to determine how deep (or central) a given observaton is.
Statistical depth functions provide center-outward ordering of multi-dimensional data and, therefore, can be exploited to measure the "extremeness" or "outlyingness" of a data point with respect to a given data set. In this sense, these functions could be successfully used to answer the one-class classification issue: all the observations that significantly deviate from the data cloud are indeed expected to be more likely characterized by small depth values than large ones.
One-class classification methods (and, more in general, outlier detection methods) based on statistical depths have gained increasing attention in the literature thanks to their appealing features [25, 29, 98]. Depth-based methods, in fact, are completely data-driven and avoid strong distributional assumption; in addition, for a low dimensional input space, they provide intuitive visualization of the data set by finding peeling and depth contours (e.g. bagplot, convex hull, ...).

3.2.2.3 Reconstruction methods

The main idea of any reconstruction method is to make an assumption about the data generating process and, then, describe objects by using their *reconstruction error* $\varepsilon_{reconstr}$, i.e. the difference between the fitted and the observed values. In particular, since the underlying model is supposed to well represent the target class, $\varepsilon_{reconstr}$ could be considered as measure of distance from x to this set.

In this work, well-known reconstruction methodologies are discussed, including K-means, Learning Vector Quantization (LVQ), Self-Organizing Maps (SOM), Principal Component Analysis (PCA) and mixture of PCAs and a network model (Autoencoder).

K-means, LVQ and SOM

All these methods are based on the idea that if the data reflect an underlying group structure, they can be well represented by using a set of K prototype vectors μ_k, with K decided beforehand. The position of each prototype ("placing") is directly learned from the training set. For the K-means algorithm [13] the best placing is obtained by minimizing the total mean squared error between the training samples and their representative prototypes, i.e. the trace of the pooled within cluster covariance matrix:

$$\varepsilon_{K-means} = \sum_{i=1}^{n} \left(\min_k ||\mathbf{x} - \boldsymbol{\mu}_k||^2 \right).$$

The optimal solution for $\varepsilon_{K-\text{means}}$ can be found by employing either a batch or an on-line routine. Batch algorithms start with a random placement of the prototypes; then, at each step, they assign $\mathbf{x}$ to the closest prototype, i.e. the group k for which

$$d_{K-\text{means}}(\mathbf{x}) = \arg\min_k ||\mathbf{x} - \boldsymbol{\mu}_k||,$$

finally, they update the prototype to the mean of the new set S_k (the set of objects which include $\mathbf{x}$),

$$\boldsymbol{\mu}_k = \frac{1}{n_k} \sum_{i \in S_k} \mathbf{x}_i.$$

Such a procedure is repeated until the convergence is met, i.e. until the prototype places are stable. On-line techniques, on the contrary, consider each observation vector sequentially and use it to update the position of its nearest prototype (competitive learning):

$$\mu_{\mathbf{k}}(\tau + \mathbf{1}) = \boldsymbol{\mu}_{\mathbf{k}}(\tau) + \eta(\tau)(\mathbf{x_i} - \boldsymbol{\mu}_{\mathbf{k}}),$$

where $0 < \eta(\tau) < 1$ is the learning rate.

The LVQ procedure [21] is so similar to the K-means one that it can be even considered as its supervised version (a label y_i is provided for each training object $\mathbf{x}_i$). In particular, LVQ derives the best placing for $\boldsymbol{\mu}_k$ by minimizing the misclassification error and, at each step until convergence, it updates only the nearest prototype to

the training object $\mathbf{x}_i$.

Lastly, the SOM algorithm [68] usually learns the best prototype placing by using a competitive learning routine. Namely, SOM starts by setting a uniform lattice of prototypes on a p_{SOM}-dimensional plane; then, until stability, the nearest prototype of each observation vector $\mathbf{x}_i$ is identified and all the prototypes in its neighborhood updated according to:

$$\boldsymbol{\mu}_k(\tau+1) = \boldsymbol{\mu}_k(\tau) + \eta(\tau) f_{wind}(|\mathbf{x}_i - \boldsymbol{\mu}_k|)(\mathbf{x}_i - \boldsymbol{\mu}_k),$$

where $f_{wind}(|\mathbf{x}_i - \boldsymbol{\mu}_k|)$ is a window function that is equal to 1 when $\mathbf{x}_i = \boldsymbol{\mu}_k$ and decreases as $|\mathbf{x}_i - \boldsymbol{\mu}_k|$ increases.

K-means, LVQ and SOM use the euclidean distance for the definition of $\varepsilon_{reconstr}$ and, therefore, they are sensitive to scaling of the features. Furthermore, their performances strongly depend on the correct choice of their specific parameters: the number K of clusters for K-means, the learning rate η for LVQ and the topological assumptions determining the neighborhood for SOM.
For each method, the number of free parameters is equal to the dimension of $\boldsymbol{\mu}$:

$$n_{FreeK-means} = n_{FreeLQV} = pk,$$

$$n_{FreeSOM} = pk^{p_{SOM}}.$$

All these techniques are computationally low expensive and they require small memory spaces, especially when they use on-line learning procedures.

PCA and mixture of PCAs

Principal Component Analysis is a statistical procedure that could be employed as a one-class classifier when p is large and a clear linear subspace is present. Its original aim is to find the linear combinations of the input features which explain (as best as possible) the internal variance and covariance structure of the data. Specifically, PCA maps each data vector $\mathbf{x}_i$ on the orthonormal subspace spanned by the eigenvectors $\mathbf{e}_i = (e_{i1}, \cdots, e_{ip})$, $i = 1, \cdots, p$ of Σ, decreasingly ordered according to the corresponding eigenvalues λ_i.

$$X' = AX\ , \qquad A = [\mathbf{e}_1, \cdots, \mathbf{e}_p]. \tag{3.8}$$

Usually, not the whole set of $\mathbf{e}_i$, $i = 1, \cdots, d$, is used to define the final transformation of X into X', but only the first q eigenvectors explaining a certain fraction (e.g. the $70 - 80\%$) of the data variability are retained. In this sense, PCA could be considered as a dimensionality reduction technique.
In order to implement one-class classification, the reconstruction error may be computed as the Mahalanobis distance from each original object to its mapped version

$$\varepsilon_{PCA} = \sum_{i=1}^{p} \frac{\mathbf{x}'_i}{\lambda_i} ;$$

then, the empirical distribution of this error could be used to identify the optimal threshold for classification.
According to Pearson [91], an intersting property of PCA is that the projection defined by 3.8 minimizes ε_{PCA}.
PCA is particularly sensitive to both noise and outlier observations as they directly affect the variance and covariance structure of the data. In addition, since the number of free parameters is quite large

$$n_{FreePCA} = \frac{p(p-1)}{2},$$

a substantial effective sample size is required.
In the case of not mean-centered data, the mean vector has to be estimated and, thus, another p free parameters should be added to $n_{FreePCA}$. Obviously, if only q components are retained,

$$\varepsilon_{PCA} = \sum_{i=1}^{q} \frac{\mathbf{x}'_i}{\lambda_i} \quad \text{and} \quad n_{FreePCA} = \frac{q(q-1)}{2}.$$

Since PCA only defines a *linear* projection of the data, its application is quite limited. Several *non-linear* extensions have been proposed in the statistical literature: among other, curves ([54] and [114]), multi-layer auto-associative neural networks ([69]), kernel-function approach ([125]) and generative topographic mapping, or GTM, ([14]) represent just some examples.

In 1999, Tipping and Bishop ([117] and [118]) firstly attempted to model the nonlinear structure of the data by using a mixture of K local linear sub-models. Specifically, they reformulated the PCA within a maximum-likelihood framework

based on a specific version of the Gaussian latent variable model. Here, the marginal probability of a given object x is

$$f_{MixPCA}(\mathbf{x}) = \pi_k \sum_{k=1}^{K} \left((2\pi)^{-p/2} |C_k|^{-1/2} \exp\left\{ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T C_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right\} \right),$$

where $C_k = \sigma^2 I + A_k A_k^T$ is the covariance matrix in the A_k subspace. The Mixture of Principal Component Analyzers is built in a probabilistic structure, and, therefore, all the model parameters $\boldsymbol{\mu}_k$, C_k and A_k could be estimated through the EM algorithm. The number of free parameters is significantly large

$$n_{FreeMixPCA} = pq + 1 - \frac{p(p-1)}{2},$$

even if it could be controlled by the choice of q.
Similarly to PCA, also Mixture of PCA is very sensitive to scaling of the features.

Autoencoders

The autoencoder is a particular type of neural network algorithms whose aim is to approximately reproduce as output only the input objects that resemble the training data.
Specifically, an autoencoder is composed of an input and an output layers of the same dimension p and one (ore more) internal hidden layer(s), constrained to have a dimension $q < p$, where q is the number of hidden units. This "undercomplete" representation learning process acts as an information compressor and it forces the model to capture only the most relevant features of the training data. The algorithm structure presents two different phases, each defined by a transition function:

1. during the *encoding* phase the input $\mathbf{x} \in \mathbb{R}^p$ is mapped into a code (or image) $\mathbf{r} = \sigma(A\mathbf{x} + \mathbf{b}) \in \mathbb{R}^q$, where σ is an element-wise activation function (sigmoid function or rectified linear unit), A is a weight matrix and $\mathbf{b}$ is a bias vector;

2. during the *decoding* phase $\mathbf{r}$ is reconstructed into $\mathbf{x}' = \sigma'(A'r + \mathbf{b}') \in \mathbb{R}^p$, where σ', A' and $\mathbf{b}'$ could differ from the corresponding objects of the previous phase.

The parameters of this model are optimized so as to minimize the average reconstruction error, computed as the difference between the network output, $\mathbf{x}'$, and the network input, $\mathbf{x}$:

$$\varepsilon_{Autoenc} = ||\mathbf{x}' - \mathbf{x}||^2.$$

The autoencoder with only one hidden layer and q linear transformation units projects $\mathbf{x}$ onto the q-dimensional subspace spanned by the first q principal components. Therefore, this method, too, could be viewed as a dimension reduction technique.
By their specific nature, autoencoders are very flexible and they allow a variety of functional mappings to be represented. However, their shortcoming is the need to set several parameters by the user: the number of hidden layers, the number of hidden units at each layer, the type of transformation function, the learning rate and the stopping criterion. The number of free parameters can be very large, even if it can be controlled by the choice of q. In the case of just one hidden layer:

$$n_{FreeAutoenc} = (2p + 1)q + p.$$

As the set of weights, A, has to be estimated using the complete training set, both the computational complexity and the storage requirements of the learning phase could be very high (approximately of order $\mathcal{O}(N_A^3)$, N_A being the number of weights in the network). Instead, the computational complexity and the storage requirements of the classification phase are moderate.

3.2.3 What is transvariation probability?

The transvariation concept has proved to be very useful in the standard classification context as a measure of group separability, especially when the assumptions that justify the optimality of Fisher's linear discriminant function are not met [84]. Its applicability can be even extended to the one-class domain, as the definition of transvariation probability seems to perfectly fit the idea of resemblance between an object and a group.
Moreover, the transvariation probability concept we refer to can be also viewed as a *data depth* measure, i.e. a measure of how deeply a generic observation lies in the data cloud [120].

According to Gini [48],

Definition 3. *A group g and a constant c are said to transvariate on a variable x, with respect to its mean value m_x if the sign of some of the n differences $x_i - c$ is opposite to that of $m_x - c$.*

In this definition, the constant c can be seen as the observed value of a *degener-*

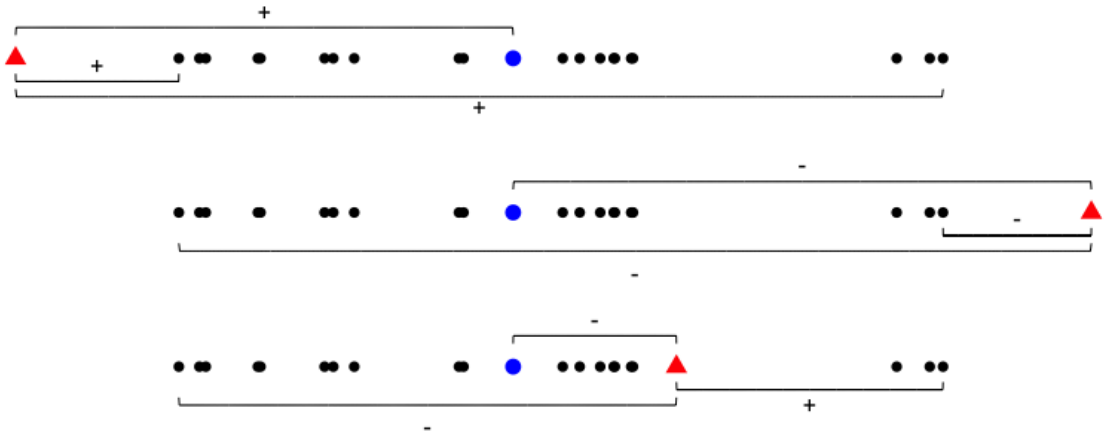


Figure 3.4: Two examples of no transvariation (first two rows) and a case of transvariation (third row) between a given unit (red triangle) and the group median (blue circle).

ated group, that is a group made of a single unit. Hence, by following this approach, the application of such definition to the one-class domain is straightforward: c can be considered as the single unit whose resemblance with respect to the target class (namely, with its median m_x) shall be evaluated.
In order to fully understand what transvariation means, consider as an example, the three different scenarios depicted in Figure 3.4. In the first two, no transvariation occurs between constant c (red triangle) and the mean value m_x (blue circle) as all the differences $x_i - c$, where x_i is any other group observation (black points), have the same sign pattern. In the third case, on the contrary, there is evidence that c transvariates with respect to m_x, as there are three points whose differences with c have opposite sign with respect to that of $m_x - c$.

The probability that an event fulfills Definition 3 is known as *transvariability*, τ. For the discrete case, τ is simply the number of transvariations over the number of possible differences,

$$\tau = \frac{s_x + \frac{s'_x}{2}}{n}, \tag{3.9}$$

where:

- s_x is the number of units for which $(x_i - c)(m_x - c) < 0$;

- s'_x is the number of units for which $(x_i - c)(m_x - c) = 0$;
- n is the number of differences $(x_i - c)$.

If we assume m_x to be the median (as Gini did), the maximum of τ is $\frac{1}{2}$.
The definition of transvariation probability with respect to an average value (the median) is the ratio between the *transvariability* and its maximum (τ_M)

$$tp_c = \frac{\tau}{\tau_M} \qquad 0 \leq tp_c \leq 1.$$

Here, values closer to 1 reflect an higher resemblance of c with the target class.
The **discrete** definition of transvariation probability is:

$$tp_c = \frac{\tau}{(1/2)} = 2\,\frac{s_x + \frac{s'_x}{2}}{n}.$$

When the probability density function of the target class is known or can be estimated, a density version of transvariation probability can be used. In analogy with the discrete case, transvariability can be defined as:

$$\tau = \min[F(c), 1 - F(c)], \tag{3.10}$$

where $F(c)$ is the cumulative distribution function evaluated in c. Assuming m_x to be the median, its maximum is still $\frac{1}{2}$. Thus, the **density** version of transvariation probability is given by:

$$tp_c = \frac{\tau}{(1/2)} = 2 \cdot \begin{cases} F(c) & m_x \geq c \\ 1 - F(c) & m_x < c \end{cases}.$$

Transvariation probability allows for extensions to more than one variable. Specifically, in the multivariate discrete case, the definition of transvariability τ, coherently to 3.9, corresponds to the *joint* probability that an event fulfills the Definition 3:

$$\tau = \frac{s_{\mathbf{x}} + \frac{s'_{\mathbf{x}}}{2}}{n}, \tag{3.11}$$

where

- $s_{\mathbf{x}}$ is the number of units for which $(\mathbf{x}_i - \mathbf{c})(m_{\mathbf{x}} - \mathbf{c}) < 0$ for all the variables;
- $s'_{\mathbf{x}}$ is the number of units for which $(\mathbf{x}_i - \mathbf{c})(m_{\mathbf{x}} - \mathbf{c}) = 0$ for all the variables;

- n is the number of differences $(\mathbf{x}_i - \mathbf{c})$.

If we assume

$$m_{\mathbf{x}} = (m_1, \ldots, m_p)$$

be the multivariate *spatial* median or *mediancentre*[4] (i.e. $m_{\mathbf{x}}$ is the vector that minimizes $\sum_n d(\mathbf{x}, m_{\mathbf{x}})$, where $d(\mathbf{x}, m_{\mathbf{x}})$ is the distance between $\mathbf{x}$ and $m_{\mathbf{x}}$), the maximum τ_M is no longer $\frac{1}{2}$ but it needs to be estimated. In particular, τ_M can be computed as τ in 3.11 on the translated data $\mathbf{y} = \mathbf{x} - (m_{\mathbf{x}} - \mathbf{c})$. Therefore, the **multidimensional discrete** definition of **transvariation probability** is

$$tp_{\mathbf{c}} = \frac{s_{\mathbf{x}} + \frac{s'_{\mathbf{x}}}{2}}{s_{\mathbf{y}} + \frac{s'_{\mathbf{y}}}{2}}. \tag{3.12}$$

Extending, in the same way, 3.10 to the multidimensional case and considering that τ_M is no longer $\frac{1}{2}$, the **multidimensional density** definition of **transvariation probability** is

$$tp_{\mathbf{c}} = \frac{\int_{a_{\mathbf{x}_1}}^{b_{\mathbf{x}_1}} \cdots \int_{a_{\mathbf{x}_p}}^{b_{\mathbf{x}_p}} f(\mathbf{x})\, d\mathbf{x}}{\int_{a_{M\mathbf{x}_1}}^{b_{M\mathbf{x}_1}} \cdots \int_{a_{M\mathbf{x}_p}}^{b_{M\mathbf{x}_p}} f(\mathbf{x})\, d\mathbf{x}}$$

where, for $u = 1, \ldots, p$:

- $f(\mathbf{x})$ is the probability density function of the target class;
- $a_{\mathbf{x}_u} = \begin{cases} c_u & \text{if } c_u \geq m_u \\ -\infty & \text{if } c_u < m_u \end{cases}$;
- $b_{\mathbf{x}_u} = \begin{cases} +\infty & \text{if } c_u \geq m_u \\ c_u & \text{if } c_u < m_u \end{cases}$;
- $a_{M\mathbf{x}_u} = \begin{cases} m_u & \text{if } c_u \geq m_u \\ -\infty & \text{if } c_u < m_j \end{cases}$;
- $b_{M\mathbf{x}_u} = \begin{cases} +\infty & \text{if } c_u \geq m_u \\ m_u & \text{if } c_u < m_u \end{cases}$.

Obviously, when the variables involved in the computation can be assumed to be independent, the multivariate transvariation probability reduces to the product of

[4]Since there is more than one definition of the multivariate median in the literature, other alternatives could be considered.

the simple univariate ones:

$$tp_c = \prod_h tp_{c_{1,u}} \quad u = 1, \ldots, p,$$

where $tp_{c_{1,u}}$ is the univariate marginal transvariation probability corresponding to the u-th variable.

3.3 Transvariation based One-Class Classifier (TOCC)

3.3.1 The proposal

As stated in the introduction, the goal of any one-class classifier is to define a learning rule that accepts as many target objects as possible and rejects all those significantly deviating from this class. In particular, during the training phase, the one-class classifier uses the available information on the target class in order to build the classification model, i.e. so as to derive a frontier around this set. In this thesis, a new one-class classification method based on both the discrete and density definitions of transvariation probability is introduced. We shall refer to a Discrete version of the TOCC (D TOCC) if the transvariation probability is computed according to 3.12; similarly, we would refer to the Density-Based algorithm (DB TOCC) when considering the continuous version of the transvariation probability.
The classification rule of the TOCC is carried out through the following steps:

1. Set a value, s, as the expected sensitivity of the one-class classifier;
2. For each unit $\mathbf{c}$ compute its transvariation probability $tp_{\mathbf{c}}$ with respect to the target group median, $m_{\mathbf{x}}$;
3. Use the $s-th$ percentile of the (increasing) ordered distribution of transvariation probabilities as a threshold, $\mathbf{t}$, for the one-class classifier

For a new test sample $\mathbf{x}$, the transvariation probability of $\mathbf{z}$, $tp_{\mathbf{z}}$,with respect to $m_{\mathbf{x}}$ is computed. Then, the decision whether $\mathbf{z}$ belongs to the target set or not is based on threshold t and can be summarized as follows:

$$\begin{cases} tp_{\mathbf{z}} \geq \mathbf{t} & \mathbf{z} \in \text{ target class} \\ tp_{\mathbf{z}} < \mathbf{t} & \mathbf{z} \in \text{ outlier class} \end{cases}.$$

Let's go back to the example described in Section 3.2.1 and visualize how the TOCC works in practice. In Figure 3.5, observations are plotted in different colors according to the level of their transvariation probabilities, $tp_{\mathbf{c}}$, with respect to the target group median, $m_{\mathbf{x}}$ (green rhombus). As expected, moving away from $m_{\mathbf{x}}$, the intensity of the transvariation probability decreases. In particular, setting s equals to 0.90, all the objects with a value of $tp_{\mathbf{c}}$ smaller than the threshold, $\mathbf{t}$, are classified as (false) negative (blue circles).

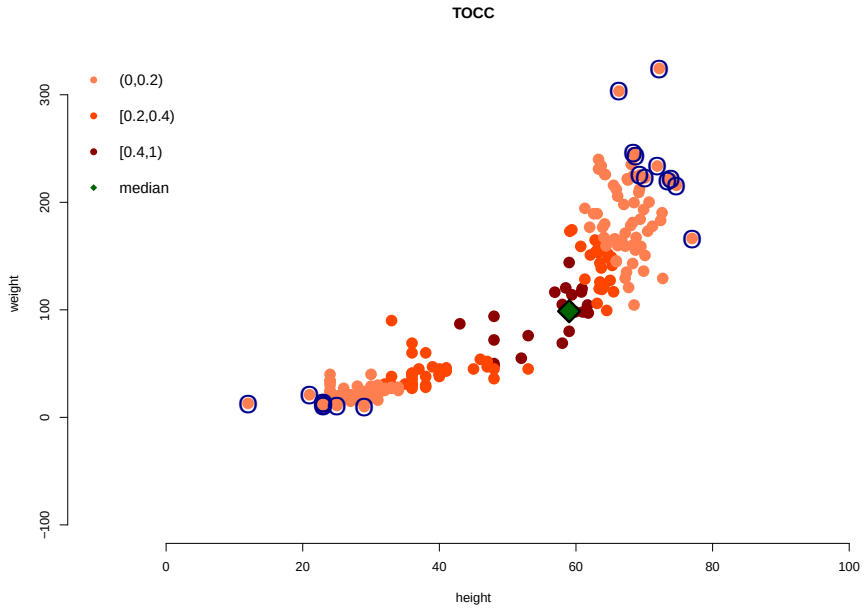


Figure 3.5: Level of transvariation probability between each observation and the target group median (green rhombus). Blue circles are the target objects (about the 10% *of the whole target set) wrongly classified.*

3.3.2 A modified version of the TOCC

With the aim to better improve the approach described in the previous section and inspired by those algorithms that use a set of prototypes to represent the input data (e.g. K-means, SOM, ...), a modified version of the Discrete TOCC is introduced. Basically, the proposal extends the D TOCC procedure, by combining it with the clustering information on the target class provided by a clustering algorithm (the Partitioning Around Medoids, PAM). The main peculiarity of the PAM D TOCC is that, by analyzing each cluster separately, it returns a set of thresholds, rather than a single one. In so doing, it is capable to detect those deviating observations

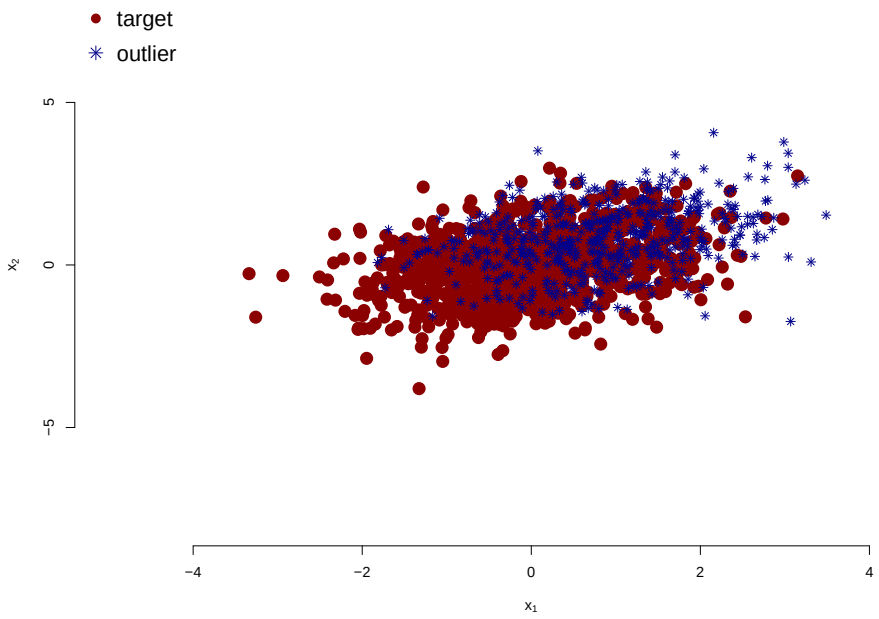


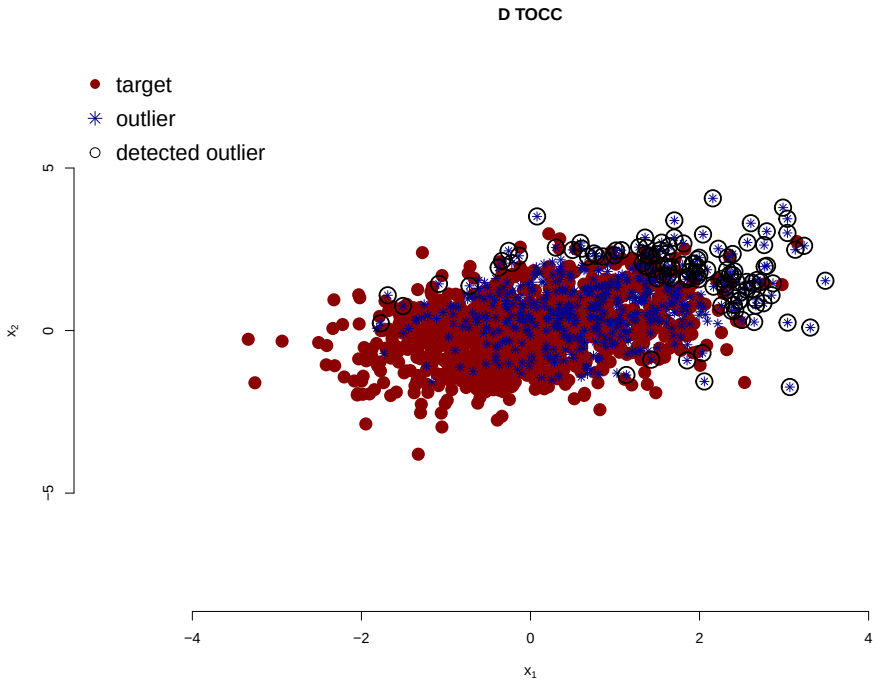
Figure 3.6: True class membership of a simulated example.

that do not necessarily lie on the external border of the target class, but that are scattered within the set.
For a better understanding, consider the example in Figure 3.6, where the observations deviating from the target class (black points) are plotted as (yellow) triangles. As can be easily noticed, the *non-target* objects are not well separated from the *target* ones and all of them are confused in the same points cloud.

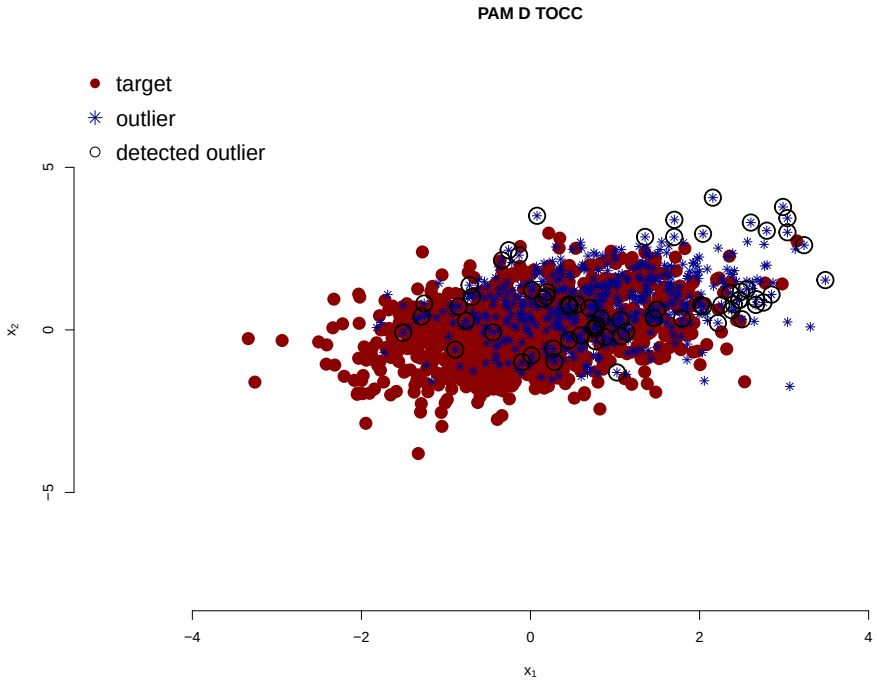
In Figure 3.7, the two different solutions yielded by the "standard" D TOCC (a) and the PAM D TOCC (b), respectively, are depicted. In particular, subfigure (a) shows how the D TOCC, by its nature, is able to recognize as outliers only the deviating points placed on the target class perimeter. For this reason, this procedure is particularly appropriate in presence of outlier objects that are "distant" from the target class or when there is no evidence of strong overlap between the two "classes". In all the other situations, the PAM D TOCC should be preferred: as clearly illustrated in subfigure (b), this algorithm is able to detect non-target objects that deviate from the target class along different directions.

The following steps outline the PAM D TOCC two-phases process:

Phase I:



(a)



(b)

Figure 3.7: Class memberhip of the same simulated example in Figure 3.6 predicted by the (a) D TOCC and the (b) PAM D TOCC.

(a) run the PAM algorithm[5] on the target class and store the resulting information on both the cluster membership and the prototype vectors,

Phase II: for each cluster k,

(a) set a value, s, as the expected sensitivity of the one-class classifier[6];

(b) For each unit $\mathbf{c}$ in the k-th cluster compute its transvariation probability $tp_{\mathbf{c}}$ with respect to the group prototype, ${}_k m_{\mathbf{x}}$. Notice that, in this case, the formula described in 3.12 should be used for both the univariate and the multivariate computations. In fact, since $m_{\mathbf{x}}$ is no longer the median, but the cluster centroid, there is no guarantee that τ_M is equal to $\frac{1}{2}$;

(c) use the $s-th$ percentile of the (increasing) ordered distribution of transvariation probabilities as a threshold, ${}_k\mathbf{t}$, for the one-class classifier.

For a new sample $\mathbf{z}$, the cluster membership, i, should be predicted (for example by assigning the object to the group described by the nearest prototype, ${}_i m_{\mathbf{z}}$[7]) and, then, its transvariation probability, $tp_{\mathbf{z}}$, with respect to ${}_i m_{\mathbf{x}}$, computed. The final decision to accept or reject $\mathbf{z}$ as a target object results from the rule described in 3.3.1, considering ${}_i\mathbf{t}$ as the threshold.

3.3.3 One-class classification in high-dimensional contexts

When dealing with one-class classification issues (or, more in general, with any classification task), in high-dimensional contexts, preliminary dimension reduction or variable selection procedures may be required. In particular, such a preprocess is essential in order to both avoid the effects of the *curse of dimensionality* and reduce the computational costs that might result by the presence of too many features.

3.3.3.1 Dimension reduction

For dimension reduction, the classical Principal Component Analysis (PCA) or its sparse version (sPCA, introduced in [133]) proved to produce good results in the

[5] The number of groups K is chosen beforehand.
[6] Generally, this value is set equal for all clusters.
[7] $\mathbf{x}$ is assigned to group i if

$$d(\mathbf{x}, {}_i m_{\mathbf{x}}) < d(\mathbf{x}, {}_k m_{\mathbf{x}}) \quad k = 1, \ldots, K.$$

one-class framework, given that only the low-variance projections are retained [113]. Such directions, in fact, by producing the tightest description of the target class, turned out to be the most informative ones for the one-class classification problem. In addition to PCA, the Random Projection (RP) method represents a valid alternative for reducing the data dimensionality[8]. Similarly to the ensemble approach introduced by Cannings and Samworth [20] for supervised classification, the identification of the B_1 most interesting projection from a one-class point of view, too, is possible. In particular, since, in this context, no information on the outlier class is available and the objective is to identify those directions yielding the compact representations of the target set, a new one-class specific criterion is required. Coherently with the definition of transvariation probability presented in Section 3.2.3, a possible choice for our procedure is to select, within B_2 different solutions, the RP that minimizes the Meadian Absolute Value (MAD) of the projected data. Such a choice, in fact, provides the most compact version of the projected target set with respect to its median. The classification results obtained by performing one-class classification on the selected projections are, then, aggregated by using a majority vote so as to derive the final unit allocation.

3.3.3.2 Variable selection

Alternatively to dimension reduction, it is reasonable to help classification methods work more efficiently by finding the subset of variables that actually carries the *relevant information* about the observations and by discarding the non-informative ones. The model-based *varSel* algorithm introduced in [99] deals with this issue and it uses Gaussian Mixtures to identify the most suitable variables for classification (and clustering) purposes. In this sense, others approaches to variable selection can be also found in [102] [87] and [82].
In addition to those, the Variable Importance in Projection (VIP) criterion introduced in Section 2.5.1 could be used in the one-class context, too, with the aim to rank the input features and, thus, to remove those that are deemed not to improve the classification performances.

3.4 Empirical analysis

In this section, the specificity rates s^- (i.e. the fraction of outliers correctly recognized) of the TOCC in both simulated and real experiments are discussed. For each

[8] A detailed review of RPs is discussed in Section 2.1

example, the Discrete (D TOCC), the Density-Based (DB TOCC) and the PAM-based (PAM D TOCC) versions of the procedure have been implemented and, where needed, the dimension reduction techniques described in 3.3.3.1 were considered.
In the DB TOCC, since the true shape of the target class distribution was not known, a Gaussian mixture model (see 3.4) has been fit using the `Mclust` function[9]:

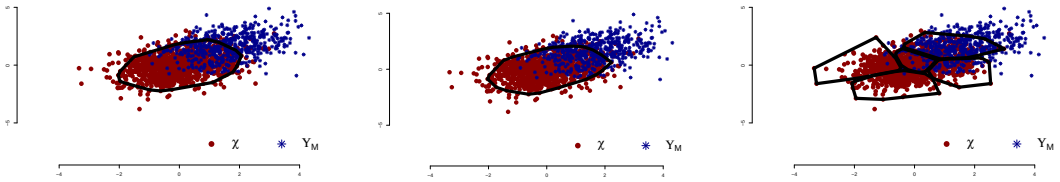
$$f(\mathbf{x}) = \sum_{k=1}^{K} \pi_k f_k(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k).$$

In the PAM D TOCC, $K = 5$ clusters were considered.
For comparison, results of applying six different one-class classification methods representing the state of the art, are presented. In particular, these methods include the Gaussian model (Gauss, implemented using the `mahalanobis` function), the Mixture of Gaussians approach (Mix-Gauss, implemented using the `mclust` package; here, the optimal number of components, ranging from 1 to 9, wass chosen so as to maximize the BIC), the kernel density estimation (KDE, implemented using the `ks` package with the normal kernel an the unconstrained plug-in bandwidth matrix selector), the K-means algorithm (KM, implemented using the `kmeans` function with $K = 5$ clusters), the 2-dimensional self organizing map (SOM, implemented using the `kohonen` package with a 5×5 grid and a learning rate $\alpha = (0.5, 0.3)$) and the support vector data description (SVDD, implemented using the `svdd` package [107], with a cost parameter for the postive examples $C = 0.1$).

3.4.1 Simulated examples

A wide simulation study has been conducted in order to evaluate the performances of the method proposed in Section 3.3.1. In each of the simulation settings described below, the target (χ, red points) and non-target (Υ, blue stars) data have been generated according to different p-dimensional distributions, where p was chosen equal to 1 and 2 so as to visualize (see, for example Figure 3.8) how the different versions of the TOCC act on the boundary definition.

[9]The number of mixing components (ranging from 1 to 9) and the model shape were chosen so as to maximize the BIC.



D TOCC *DB TOCC* *PAM D TOCC*

Figure 3.8: Examples of the boundary definition.

Let

$$\boldsymbol{\mu}_0 = 0_p \ , \qquad \Sigma_0 = \begin{cases} 1 & p = 1 \\ \begin{bmatrix} 1 & 0.35 \\ 0.35 & 1 \end{bmatrix} & p = 2 \end{cases}$$

be the mean vector and the covariance structure of the target data. Let

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0 + \delta$$

be the mean vector of the non-target data, where, for a given value of δ, all the variables are equally shifted. The magnitude of the shift $\boldsymbol{\delta}$ is described by the noncentrality parameter

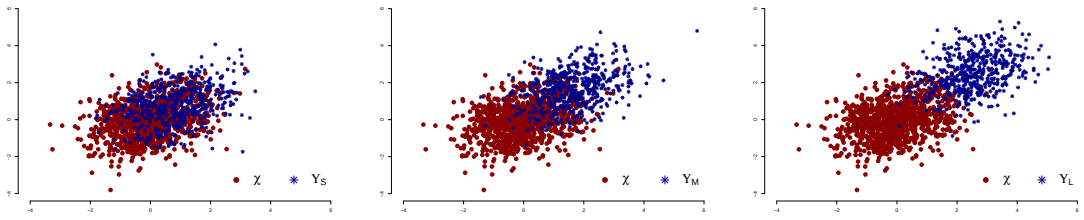
$$\lambda = \sqrt{\boldsymbol{\delta}' \Sigma_0^{-1} \boldsymbol{\delta}}.$$

For each scenario, different sizes of the target class, $n \in \{100, 200, 500\}$, and different magnitudes of the shift ($\lambda_S = 1$, Small shift; $\lambda_M = 2$, Medium shift; $\lambda_L = 3$, Large shift) were considered. The number of the deviating observations generated is always one half the number of the target ones.

Simulation setup

- **Model 1**: Gaussian

$$\begin{aligned} \chi &\sim N_p(\boldsymbol{\mu}_0, \Sigma_0) \\ \Upsilon_h &\sim N_p(\boldsymbol{\mu}_1, \Sigma_0) \qquad p = 1, 2 \end{aligned}$$

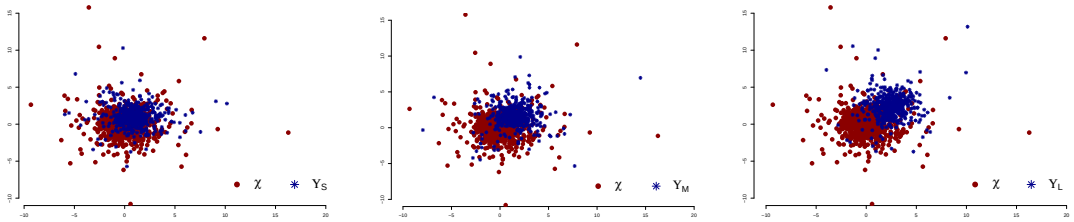


Small shift (S) *Medium shift (M)* *Large shift (L)*

Figure 3.9: Examples of the gaussian dataset.

- **Model 2**: t

$$\chi \sim t_p(\mathrm{df} = 3, \boldsymbol{\mu}_0, I_p)$$
$$\Upsilon_h \sim t_p(\mathrm{df} = 3, \boldsymbol{\mu}_1, I_p), \qquad p = 1, 2$$

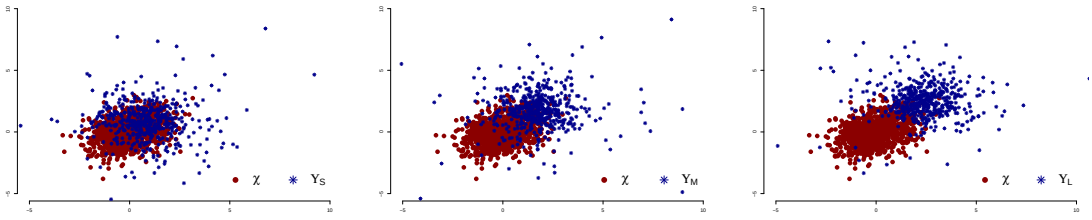


Small shift (S) *Medium shift (M)* *Large shift (L)*

Figure 3.10: Examples of the t dataset.

- **Model 3**: Gaussian/t

$$\chi \sim N_p(\boldsymbol{\mu}_0, \Sigma_0)$$
$$\Upsilon_h \sim t_p(\mathrm{df} = 3, \boldsymbol{\mu}_1, I_p), \qquad p = 1, 2$$



Small shift (S) *Medium shift (M)* *Large shift (L)*

Figure 3.11: Examples of the Gaussian/t dataset.

- **Model 4**: Gaussian/Uniform

$$\chi \sim N_p(\boldsymbol{\mu}_0, \Sigma_0)$$
$$\Upsilon \sim U_d(\min = \boldsymbol{\mu}_0 - 3 * \text{diag}(\Sigma_0), \max = \boldsymbol{\mu}_0 + 3 * \text{diag}(\Sigma_0)) \qquad p = 1, 2$$

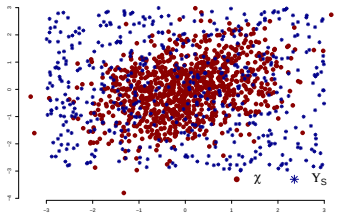
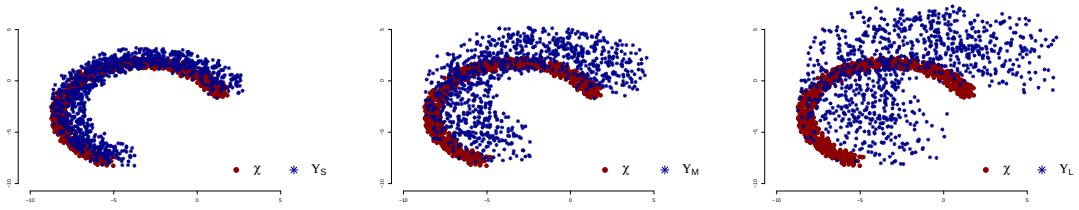


Figure 3.12: Example of the Gaussian/Uniform dataset.

- **Model 5**: Banana-shaped

  χ and Υ were generated according to a bivariate ($p = 2$) banana-shaped distribution with different types of angles: 1 for the target class, 2 for the small shift, 4 for the medium shift and 6 for the largest one.
  The function used to simulate the banana-shaped data is reported in Appendix D.



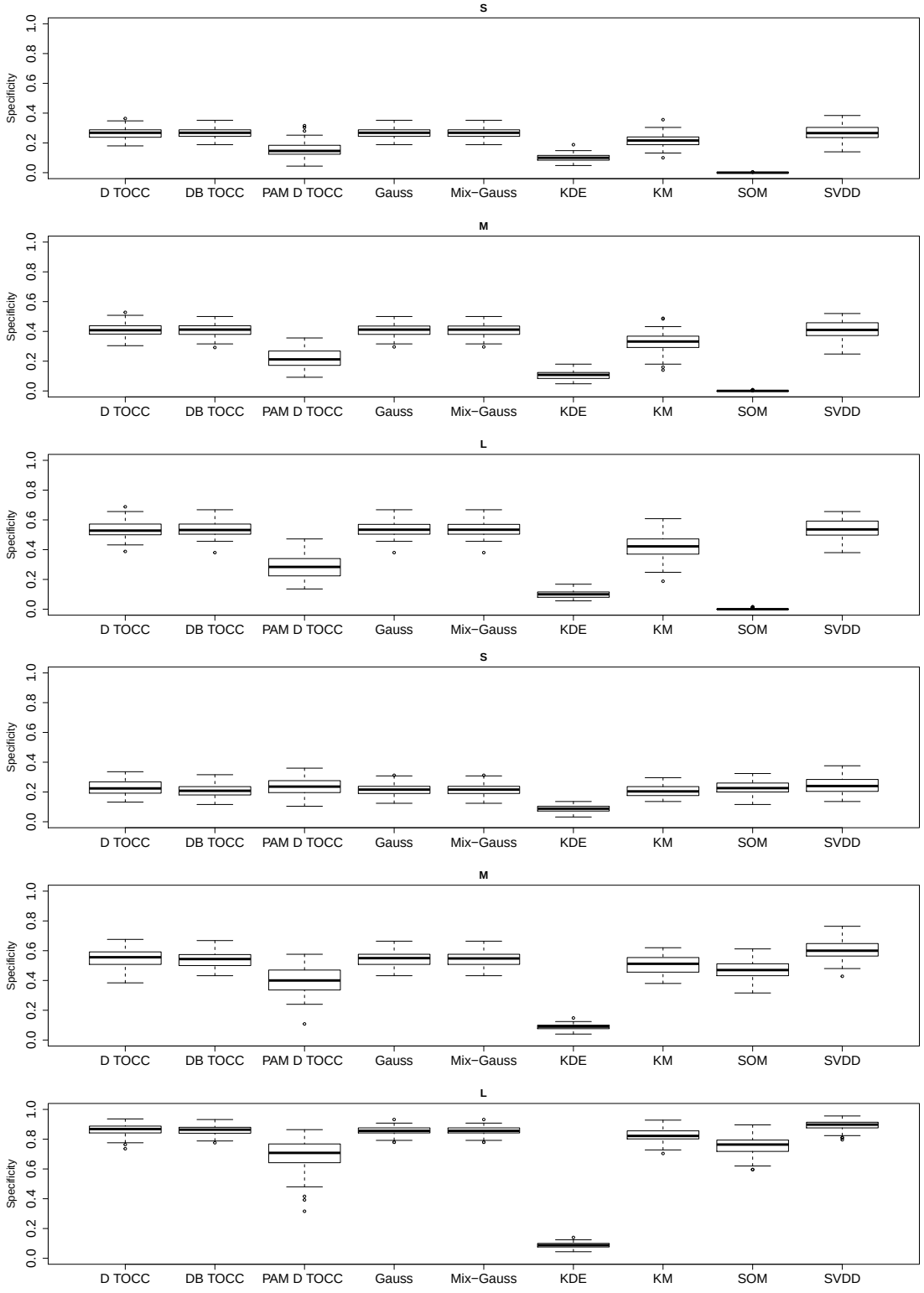
Small shift (S) *Medium shift (M)* *Large shift (L)*

Figure 3.13: Examples of the banana-shaped dataset.

Figures 3.14-3.18 and Figures C.1-C.10 (in Appendix C) contain the boxplots of the specificity rates corresponding to a sensitivity level $s \geq 0.9$, deriving from 100 simulations. Tables C.4-C.5 (in Appendix C) summarize, for each scenario, their average value and the corresponding standard deviations.

Results coming from this study clearly show the general effectiveness of the transvariation-based one-class classifier (TOCC) we introduced. In particular, for all the simulated models, both the discrete and the density versions of the algorithm attain specificity rates (for a sensitivity level $s \geq 0.9$) that are always better than or, in the worse cases, comparable with those from the state-of-the-art methods. These promising outcomes allow to efficiently use the proposed algorithms in a wide variety of problems.

A separate evaluation should be carried out for the PAM D TOCC since, as clearly depicted in the boxplots in Figures 3.14-3.18 and in Figures C.1-C.10, it works notably well in some specific scenarios. In particular, the performances of this classifier strongly depend on both the characteristics of the target set and the behavior of the non-target observations. Namely, the most effective results could be noticed in the multivariate case and/or when the target class do not present an elliptical shape (Model 5). Competitive outcomes are also evident in presence of a strong overlap between the target and non-target classes, i.e. when the outlier set is generated according to a small shift from the target one. In all these situations, in fact, being the PAM D TOCC a very flexible procedure, it seems able to both fit well the data and perfectly identify those deviating observations that are scattered within the target set and do not limitedly lie on its external perimeter.



$p = 1$

$p = 2$

Figure 3.14: Specificity for $s \geq 0.9$ sensitivity level for small (S), medium (M) and large (L) shifts for Model 1, $n = 500$.

$p = 1$

$p = 2$

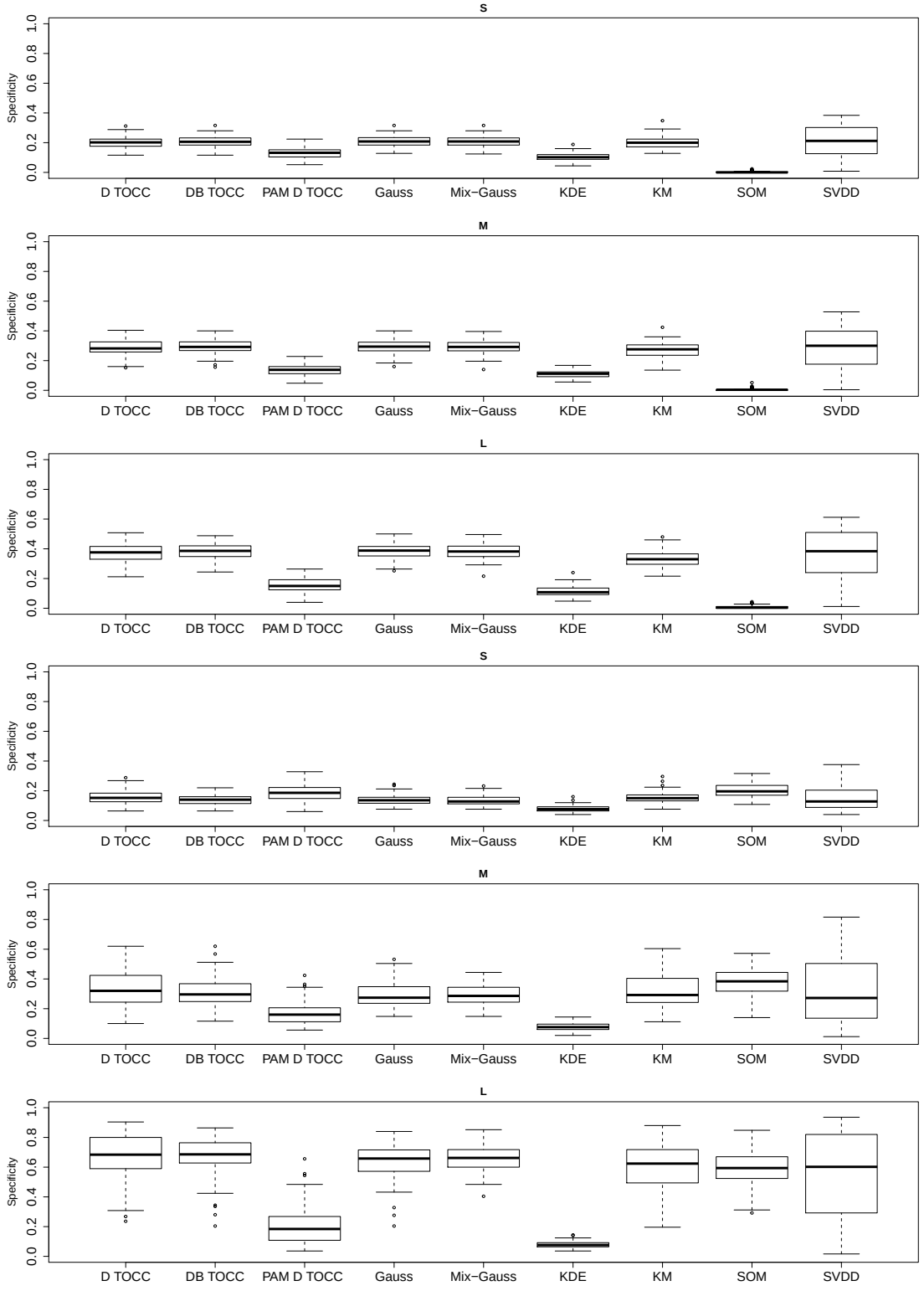
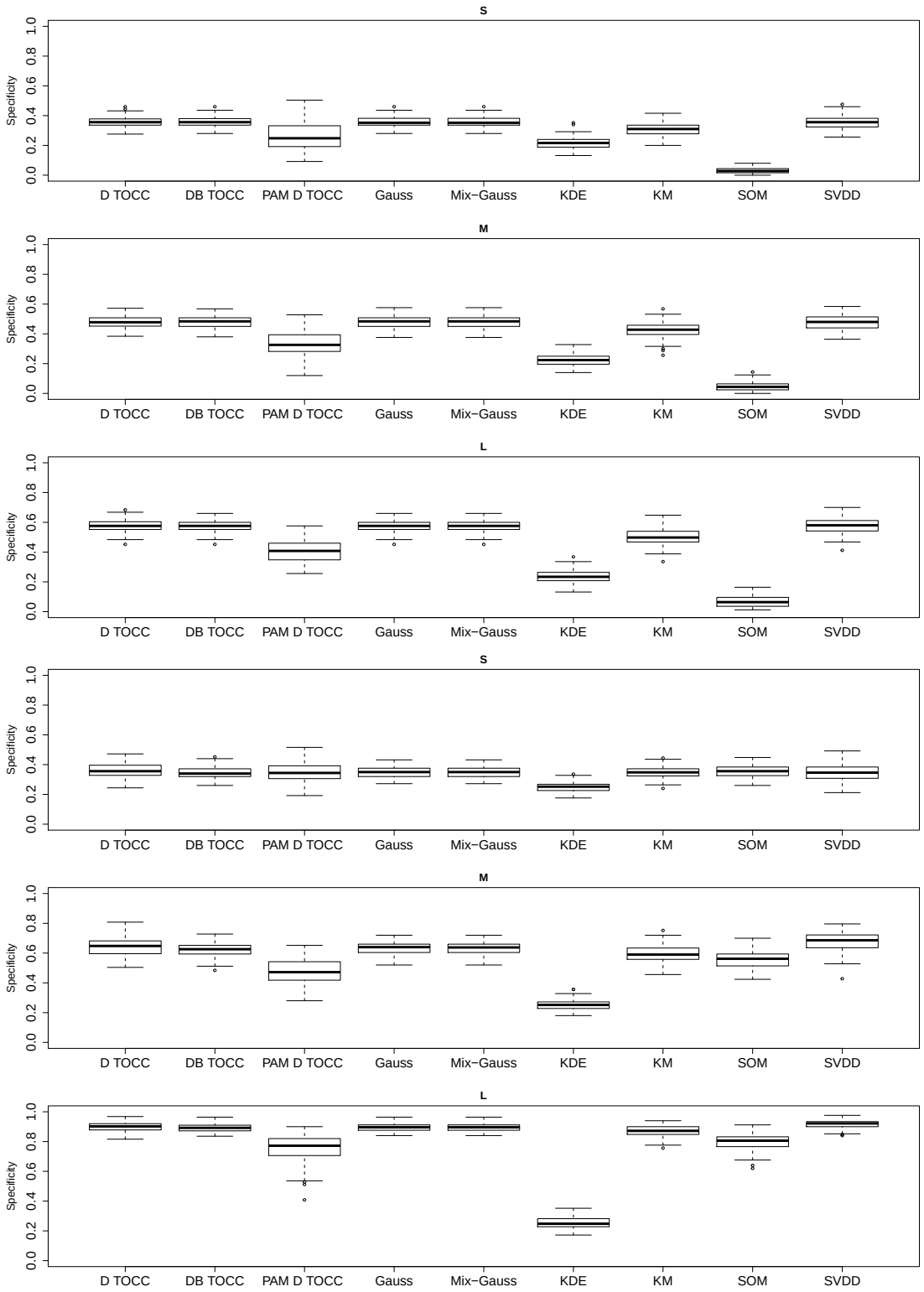


Figure 3.15: Specificity for $s \geq 0.9$ sensitivity level for small (S), medium (M) and large (L) shifts for Model 2, $n = 500$.

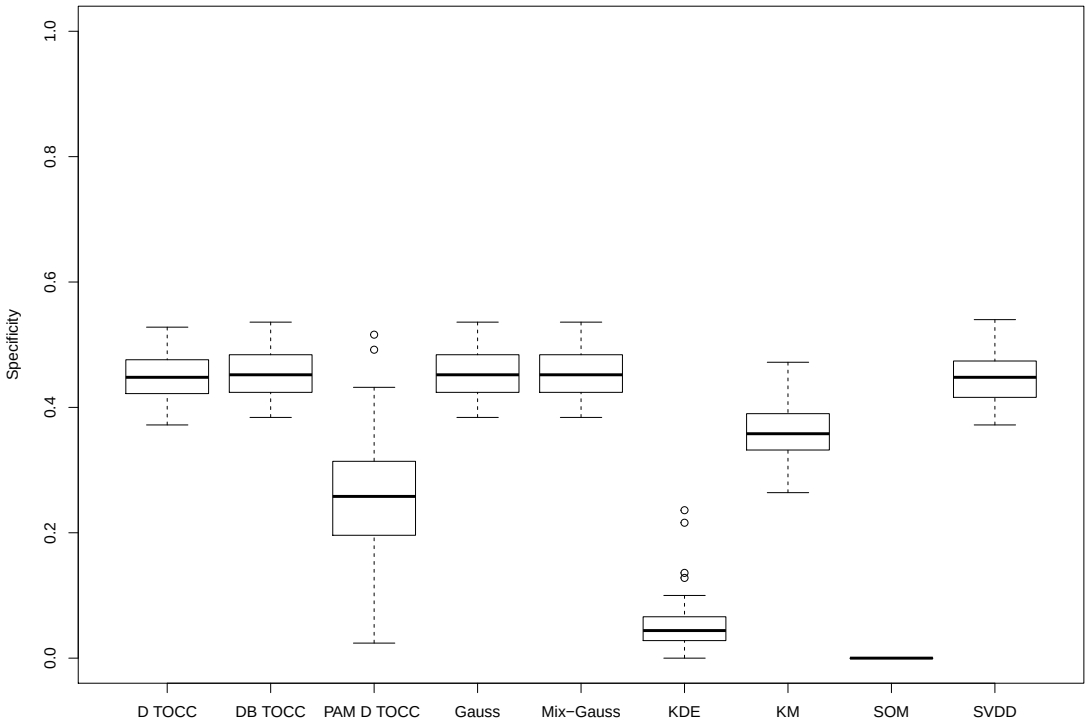


$p = 1$

$p = 2$

Figure 3.16: Specificity for $s \geq 0.9$ sensitivity level for small (S), medium (M) and large (L) shifts for Model 3, $n = 500$.

$p = 1$



$p = 2$

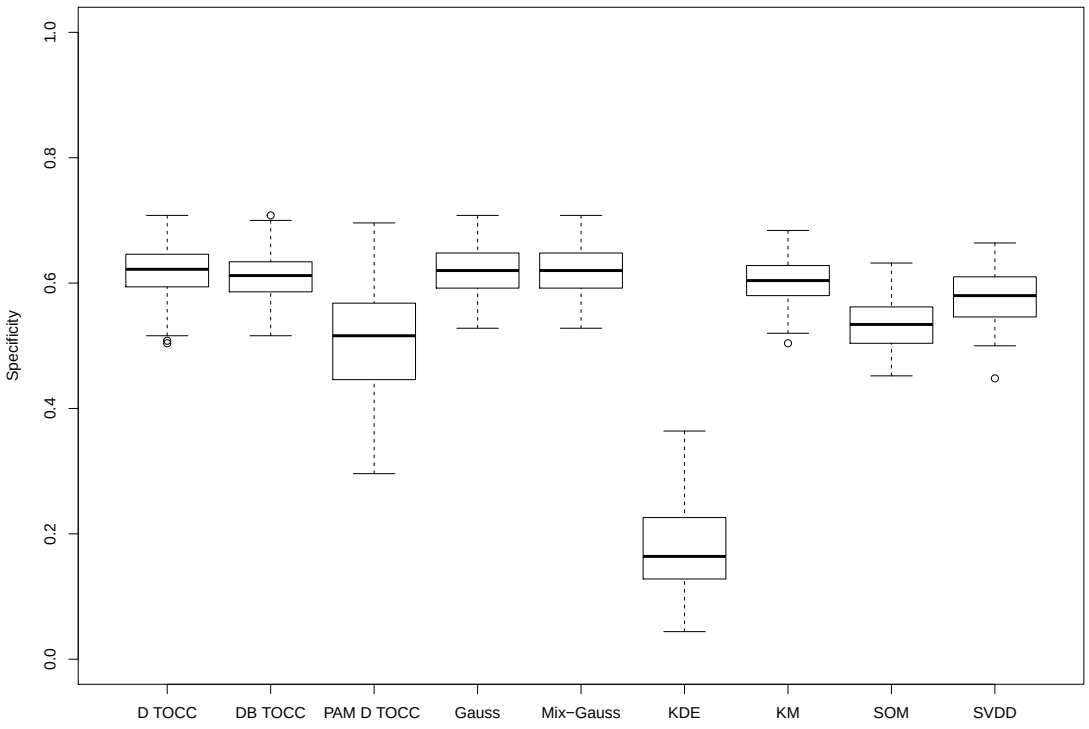


Figure 3.17: Specificity for $s \geq 0.9$ sensitivity level for Model 4, $n = 500$.

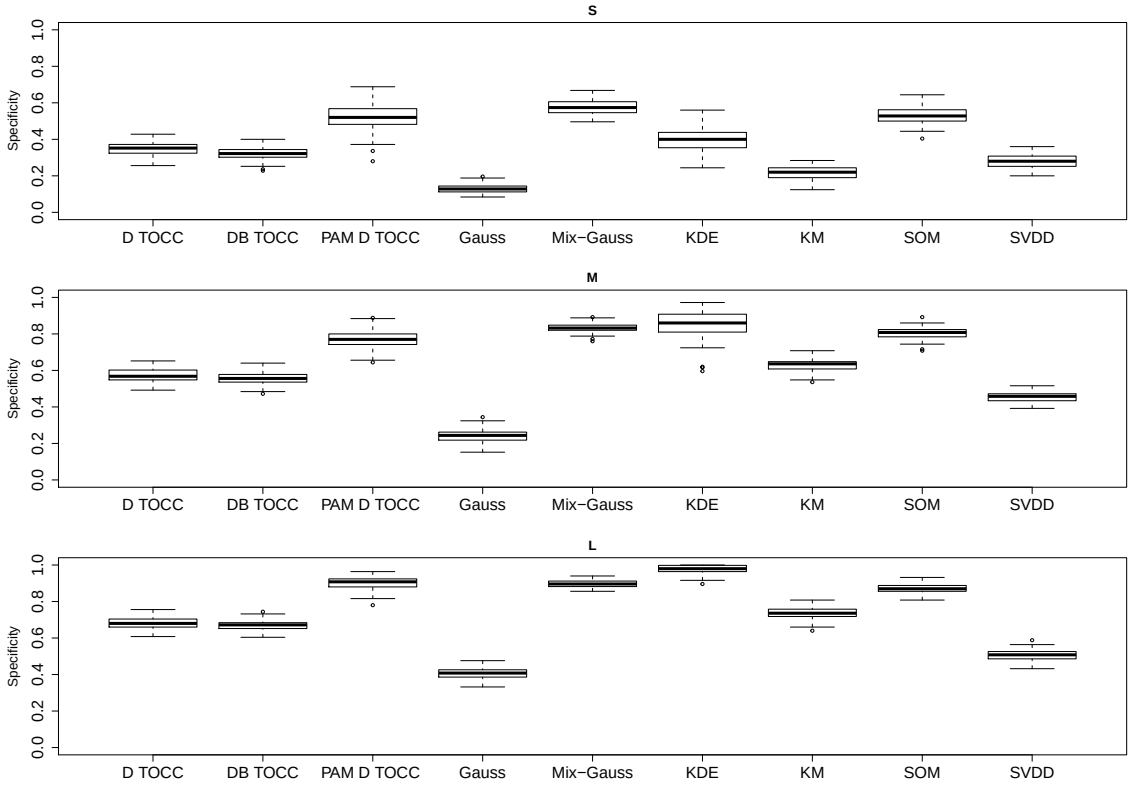


Figure 3.18: Specificity for $s \geq 0.9$ sensitivity level for small (S), medium (M) and large (L) shifts for Model 5, $n = 500$.

A further analysis of the plots reveals that the sample size, n, only affects the variability of the performances achieved by all the one-class classifiers taken into account: namely, as n increases, all the algorithms provide more stable result. Moreover, as expected, the yielded specificity rates tend to be better when the non-target objects are placed far away from the target class. Among the state-of-the art methods cosidered, the KDE represents the only exception on this and its contrary behavior is probably due to a wrong specification of the bandwidth matrix H for the outlier class: expressively, being H estimated only on the target set, the kernel $\varphi_H(.)$ is likely to produce incorrect estimates for the observations that differ too much from this class.
A special mention should be made for the results of Model 5. The non-convexity of the banana-shaped data, in fact, appears very hard to be detected by all the methods, especially by the less flexible ones. In situations like this, as clearly illustrated in Figure 3.18 and Figures C.9-C.10, the most adaptive procedures (i.e. PAM D TOCC, Mix-Gauss, KDE and SOM) seem to handle the "non-typicality" of the target class distribution more appropriately.

3.4.2 Real data examples

In this section, the classifiers above have been tested and compared on two sets of near infrared spectroscopic food data and a dataset containing measurements on waste treatment plants. Since they all present a large number of input features, p, the dimension reduction and variable selection procedures discussed in Section 3.3.3 have proved necessary. For the RP method discussed in Section 3.3.3.1, the best $B_1 = 101$ RPs were considered, each carefully chosen within $B_2 = 50$ possible solutions.
A brief description of the data used is given below. Then, Tables 3.1-3.3 show, for each example, the specificity rates corresponding to a sensitivity level $s \geq 0.9$.
The subscript below each dimension reduction or variable selection method refers to the feature space dimension used for the analysis.

3.4.2.1 Honey data

This dataset, originally described in [35] and [65], contains $n = 314$ honey samples. For experimental purposes, an alteration of half of the samples (157) was performed in laboratory using three different adulterants: fructose-glucose mixture (fg), beet invert syrup (bi) and high fructose corn syrup (hfcs). The spectra of these samples

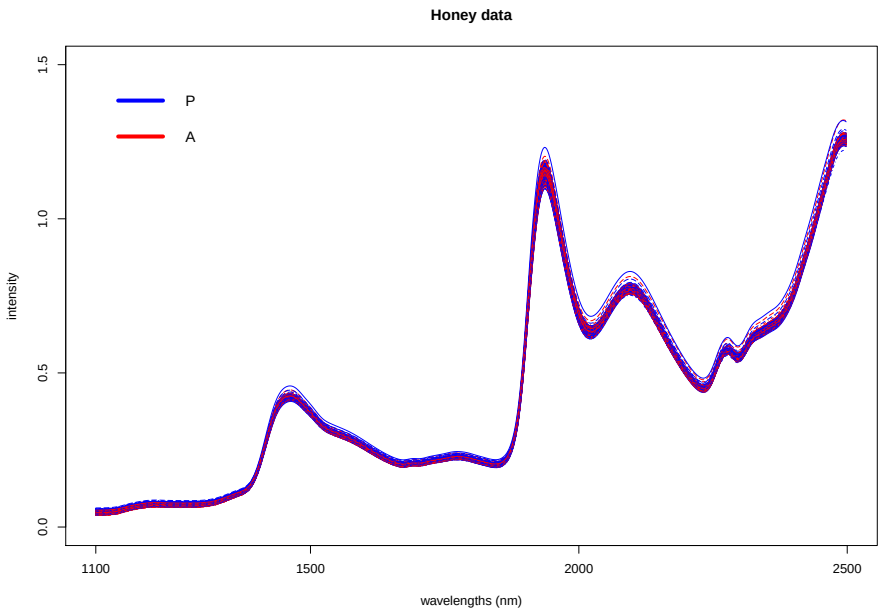


Figure 3.19: Honey data spectrum.

were recorded over $p = 700$ wavelengths (see Figure 3.19).

3.4.2.2 Oil data

The dataset [36] contains $n = 92$ Greek olive oil samples whereof 46 "pure" and 46 "adulterated". Particularly, the samples labelled "P" have not been adulterated, while the samples labelled "A" have been altered with the 5% of sunflower oil. The spectra of these samples are recorded over $p = 1050$ wavelengths (see Figure 3.20).

3.4.2.3 Waste treatment plant data

This dataset comes from the daily measures of sensors in a urban waste water treatment plant and it is available from the UC Irvine (UCI) Machine Learning Repository [76]. Here, the objective is to classify the operational state of the plant in order to predict faults through the state variables of the plant at each of the stages of the treatment process. It contains $n = 527$ observations on $p = 38$ continuous variables.
This dataset represents a difficult classification task since no method turned out to be able to identify the days in which the plant wrongly operated.

The analysis of the real data results confirms the general good performances

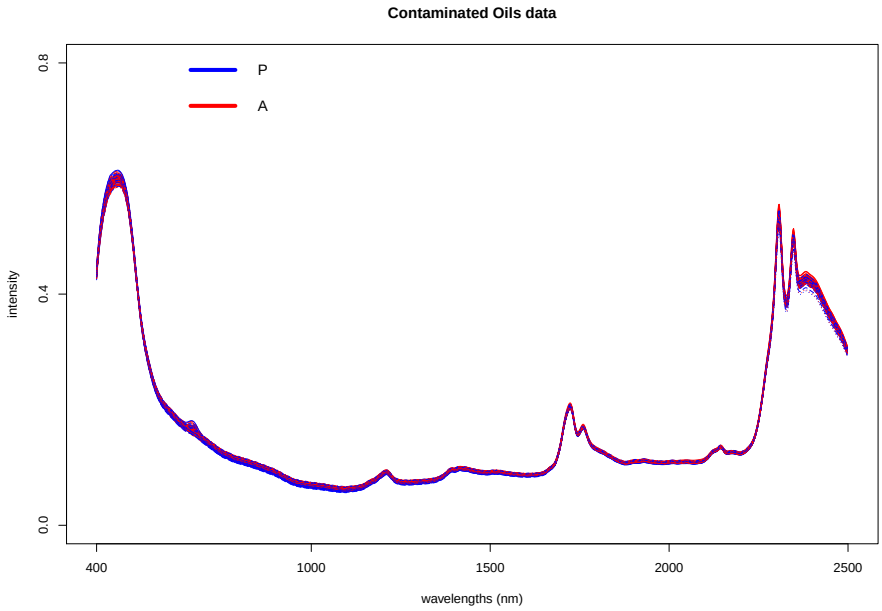


Figure 3.20: Oil data spectrum.

Method	PCA_2	$SPCA_2$	RP_2	$VARSEL_4$	VIP_4
D TOCC	1.00	0.92	0.24	0.77	0.23
DB TOCC	1.00	0.92	0.19	0.55	0.20
PAM D TOCC	1.00	0.96	0.45	0.88	0.20
Gauss	1.00	0.92	0.26	0.76	0.24
Mix-Gauss	1.00	0.92	0.31	0.83	0.34
KDE	1.00	0.99	0.09	0.23	0.21
KM	0.93	0.91	0.26	0.53	0.19
SOM	0.99	0.93	0.53	0.71	0.26
SVDD	0.76	0.92	0.19	0.32	0.16

Table 3.1: Specificity for $s \geq 0.9$ sensitivity level for Honey data.

Method	PCA_2	$SPCA_2$	RP_2	$VARSEL_4$	VIP_4
D TOCC	1.00	1.00	0.98	1.00	0.26
DB TOCC	1.00	1.00	0.91	0.96	0.28
PAM D TOCC	1.00	1.00	1.00	1.00	0.93
Gauss	—†	1.00	0.89	1.00	0.98
Mix-Gauss	1.00	1.00	0.89	1.00	0.98
KDE	1.00	0.24	0.07	0.15	0.09
KM	1.00	1.00	0.96	0.89	0.24
SOM	1.00	1.00	1.00	0.98	0.89
SVDD	1.00	1.00	0.78	0.74	0.07

Table 3.2: Specificity for $s \geq 0.9$ sensitivity level for Oil data.

Method	PCA_2	$SPCA_2$	RP_2	$VARSEL_4$	VIP_4
D TOCC	0.34	0.21	0.37	0.62	0.49
DB TOCC	0.25	0.21	0.35	0.36	0.34
PAM D TOCC	0.35	0.23	0.13	0.79	0.61
Gauss	0.31	0.20	0.35	0.31	0.22
Mix-Gauss	0.34	0.19	0.37	0.33	0.24
KDE	0.30	0.19	0.19	0.00	0.00
KM	0.34	0.19	0.32	0.25	0.25
SOM	0.42	0.24	0.36	0.40	0.44
SVDD	0.23	0.15	0.29	0.24	0.24

Table 3.3: Specificity for $s \geq 0.9$ sensitivity level for Waste treatment plant data.

attained by the TOCC, even in situations where the dimension reduction procedure implemented seems not appropriate. In particular, the adaptive version of the proposed algorithms (i.e. the PAM D TOCC) still provide specificity rates that are quite always higher than those yielded by the most flexible approaches (Mix-Gauss, KDE, SOM and SVDD).

For both the Honey and the Oil datasets, the benefit of employing dimension reduction procedures, rather than the feature selection ones, is evident: as illustrated in the first two columns of Tables 3.1 and 3.2, in fact, PCA and sPCA enable all the considered methods to excellently work. On the contrary, for the Plant data (Table 3.3), the best performances are achieved when the feature selection methods are applied: the PAM D TOCC, for example, used in conjunction with the VarSel algorithm outperforms all the other techniques, by correctly identifying the 79% of the days in which the plant wrongly operated.

For the majority of the one-class classifiers implemented on the Oil and the Plant data, the RP ensemble approach described in Section 3.3.3.1 provides specificity rates that are comparable with those obtained by performing PCA or sPCA.

As regards the VIP criterion for variable selection, it is competitive with the VarSel procedure only on the Waste treatment plant data data: the presence of highly correlated variables in both the infrared spectroscopic food data analyzed, in fact, severely affects its capability to identify the very relevant input features.

3.5 Discussion and extensions

In this work, new directions for the one class classification issue are introduced. In particular, transvariation probability (tp) has been firstly suggested as a measure of resemblance between an observation and a set of well-known objects (*target* class). The proposal performances, evaluated in terms of specificity, i.e. the proportion of actual negatives that are correctly predicted, on both real and simulated one-class

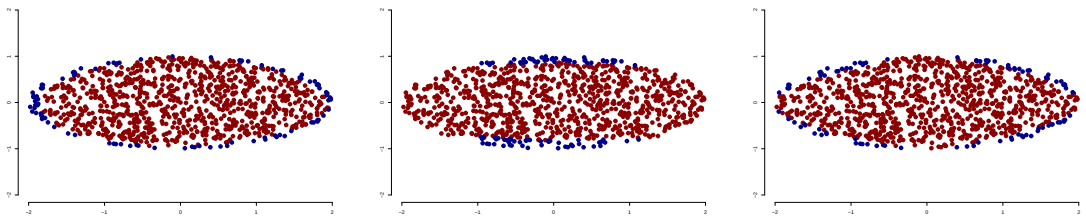
datasets, demonstrate that the use of *tp* as a tool in the construction of a one-class classifier allows to outperform several state-of-the-art methods.

Although they exploit the same measure of resemblance, the D TOCC and the DB TOCC catch different aspects of the target class boundaries. In particular, the density approach appears to give a good approximation of the so called **density** (or s-upper) **level set**, $L(s)$:

$$L(s) := \{\mathbf{x} \in \mathbb{R}^p : f(\mathbf{x}) \geq s\}$$

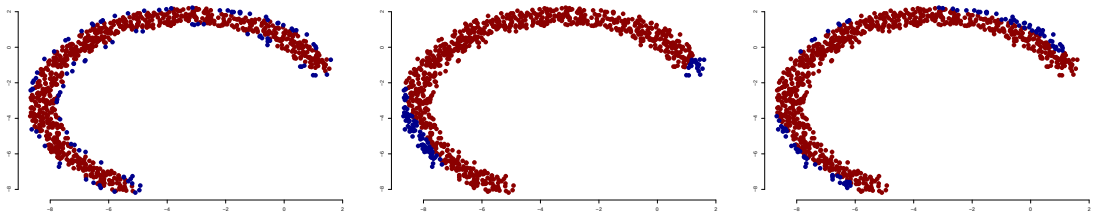
where $f(\mathbf{x})$ is a generic probability density function.

As Figures 3.21-3.22 show, in fact, the DB TOCC is able to "peel" the target set around its whole perimeter and, thus, to approximately reproduce its $L(0.9)$.



(a) *(b)* *(c)*

Figure 3.21: Density level set (a) approximation for the ellipse-shaped data performed by the D TOCC (b) and the DB TOCC (c) for s=0.9.



(a) *(b)* *(c)*

Figure 3.22: Density level set (a) approximation for the banana-shaped data performed by the D TOCC (b) and the DB TOCC (c) for s=0.9.

In the one-class context, especially for the two sets of spectroscopic food data, the VIP criterion does not perform as well as in other situations (see, for example,

the results for the Waste treatment plant data or those discussed in 2.5.2 for supervised classification). This is due to the presence of highly associated input features that pollutes the capability of the VIP to detect those actually relevant (by its nature, the VIP tends to assume approximately the same value for the very correlated variables). Thus, with the aim to identify the *relevant* features for one-class classification purposes (and to exclude the redundant ones) a specific correction procedure is advisable, so as to mitigate the correlation effect. In particular, a possible strategy is to consider the variables with the highest VIP value whilst discarding those who have an average absolute correlation with the variables already included larger than a given threshold, κ. From our empirical experience, a reasonable interval for κ would be $0.4-0.7$, depending on the average degree of the association in the original data: the strongest, the lower is the threshold. However, a more formal approach could be to rephrase the problem as an optimization one and to solve it with an iterative numerical procedure:

$$\max \quad \text{VIP}_u - \kappa|\bar{\rho}|$$

where $\bar{\rho}$ is the average correlation between the variable u and those already selected.

Tables 3.4-3.6 show, for each dataset, the specificity rates corresponding to a sensitivity level $s \geq 0.9$ for the *adjusted for correlation* VIP, κ-VIP.

Method	κ-VIP$_4$, $\kappa = 0.5$	VIP$_4$
D TOCC	0.43	0.23
DB TOCC	0.15	0.20
PAM D TOCC	0.73	0.20
Gauss	0.12	0.24
Mix-Gauss	0.14	0.34
KDE	1.00	1.00
KM	0.21	0.19
SOM	0.41	0.26
SVDD	0.11	0.16

Table 3.4: Specificity for $s \geq 0.9$ sensitivity level for Honey data for the κ-VIP with $\kappa = 0.5$.

The comparison of the performances achieved by all the discussed one-class classifiers shows the quite general improvement determined by the use of the κ-VIP, rather than its original version, to identify the four most relevant variables of each dataset. This attitude is particularly evident for the two spectroscopic data, since, for these sets, the average correlation between the input features is strong (0.63 for the Honey and 0.72 for the Oil data). The fraction of altered honey samples correctly recognized by the PAM D TOCC, for example, is boosted from 0.20 to 0.73

Method	κ-VIP$_4$, $\kappa = 0.4$	VIP$_4$
D TOCC	0.98	0.26
DB TOCC	0.96	0.28
PAM D TOCC	1.00	0.93
Gauss	1.00	0.98
Mix-Gauss	1.00	0.98
KDE	0.09	0.11
KM	0.57	0.24
SOM	0.96	0.89
SVDD	0.26	0.07

Table 3.5: Specificity for $s \geq 0.9$ sensitivity level for Oil data for the κ-VIP with $\kappa = 0.4$.

Method	κ-VIP$_4$, $\kappa = 0.6$	VIP$_4$
D TOCC	0.70	0.49
DB TOCC	0.48	0.34
PAM D TOCC	0.90	0.61
Gauss	0.41	0.22
Mix-Gauss	0.48	0.24
KDE	0.00	0.00
KM	0.25	0.25
SOM	0.38	0.44
SVDD	0.23	0.24

Table 3.6: Specificity for $s \geq 0.9$ sensitivity level for Waste treatment plant data for the κ-VIP with $\kappa = 0.6$.

by employing a threshold, κ, equals to 0.5 to discard very associated variables. An analogous result is attained for the Oil data, too: in this case, the performance of the D TOCC goes from 0.26 to 0.98 if each of the four features is sequentially identified so as to present an average absolute correlation with those already selected lower than $\kappa = 0.4$.

Appendix A

A.1 Proof of Theorem 1

Proof. **Case 1**: $\forall B_1$:

$$\begin{aligned}\lim_{\omega\to 0^+}\tau_j &= \lim_{\omega\to 0^+}\frac{\sum_{i=0}^{B_1-j}\binom{n-j}{i}\psi^i(1-\psi)^{B_1-j-i}\omega^{(B_1-j-i)(i+j)}}{\sum_{i=0}^{B_1}\binom{B_1}{i}\psi^i(1-\psi)^{B_1-i}\omega^{(B_1-i)i}}\\ &= \frac{\lim_{\omega\to 0^+}\sum_{i=0}^{B_1-j}\binom{B_1-j}{i}\psi^i(1-\psi)^{B_1-j-i}\omega^{(B_1-j-i)(i+j)}}{\lim_{\omega\to 0^+}\sum_{i=0}^{B_1}\binom{B_1}{i}\psi^i(1-\psi)^{n-i}\omega^{(n-i)i}}\\ &= \frac{\psi^{B_1-j}}{\psi^{B_1}+(1-\psi)^{B_1}}, \qquad j\le B_1, \qquad \tau_j=\mathcal{O}(\omega^{B_1-1}).\end{aligned}$$

Therefore,

$$\begin{aligned}\lim_{\omega\to 0^+}E[S] &= \lim_{\omega\to 0^+}B_1\psi\tau_1 = B_1\psi\lim_{\omega\to 0^+}\tau_1 = \frac{B_1\psi^{B_1}}{\psi^{B_1}+(1-\psi)^{B_1}}\\ \lim_{\omega\to 0^+}V[S] &= \lim_{\omega\to 0^+}B_1\psi\eta = \lim_{\omega\to 0^+}B_1\psi[\tau_1-\psi(B_1\tau_1^2-(B_1-1)\tau_2)] = B_1\psi[\lim_{\omega\to 0^+}\tau_1-\psi(B_1\lim_{\omega\to 0^+}\tau_1^2-(B_1-1)\lim_{\omega\to 0^+}\tau_2)]\\ &= B_1\psi\left[\frac{\psi^{B_1-1}}{\psi^{B_1}+(1-\psi)^{B_1}}-\psi B_1\left(\frac{\psi^{B_1-1}}{\psi^{B_1}+(1-\psi)^{B_1}}\right)^2+\psi(B_1-1)\frac{\psi^{B_1-2}}{\psi^{B_1}+(1-\psi)^{B_1}}\right]\\ &= B_1\psi\left[\frac{\psi^{B_1-1}}{\psi^{B_1}+(1-\psi)^{B_1}}-\psi B_1\left(\frac{\psi^{B_1-1}}{\psi^{B_1}+(1-\psi)^{B_1}}\right)^2+\psi B_1\frac{\psi^{B_1-2}}{\psi^{B_1}+(1-\psi)^{B_1}}-\psi\frac{\psi^{B_1-2}}{\psi^{B_1}+(1-\psi)^{B_1}}\right]\\ &= B_1\psi\left[\frac{\psi^{B_1-1}}{\psi^{B_1}+(1-\psi)^{B_1}}-\psi B_1\left(\frac{\psi^{B_1-1}}{\psi^{B_1}+(1-\psi)^{B_1}}\right)^2+\psi B_1\frac{\psi^{B_1-2}}{\psi^{B_1}+(1-\psi)^{B_1}}-\frac{\psi^{B_1-1}}{\psi^{B_1}+(1-\psi)^{B_1}}\right]\\ &= B_1\psi\left[\frac{B_1\psi^{B_1-1}}{\psi^{B_1}+(1-\psi)^{B_1}}-\psi B_1\frac{\psi^{2B_1-2}}{[\psi^{B_1}+(1-\psi)^{B_1}]^2}\right]\\ &= B_1\psi\left[\frac{B_1\psi^{B_1-1}}{\psi^{B_1}+(1-\psi)^{B_1}}-\frac{B_1\psi^{2B_1-1}}{[\psi^{B_1}+(1-\psi)^{B_1}]^2}\right]\\ &= \frac{{B_1}^2\psi^{B_1}}{\psi^{B_1}+(1-\psi)^{B_1}}-\frac{{B_1}^2\psi^{2B_1}}{[\psi^{B_1}+(1-\psi)^{B_1}]^2}\end{aligned}$$

It follows that:

$$\lim_{\psi\to 0^+}\left(\lim_{\omega\to 0^+} E[S]\right)=\lim_{\psi\to 0^+}\left[\frac{B_1\psi^{B_1}}{\psi^{B_1}+(1-\psi)^{B_1}}\right]=0,\quad \frac{B_1\psi^{B_1}}{\psi^{B_1}+(1-\psi)^{B_1}}=\mathcal{O}(\psi^{B_1-1})$$
$$\lim_{\psi\to 0^+}\left(\lim_{\omega\to 0^+} V[S]\right)=\lim_{\psi\to 0^+}\left[\frac{{B_1}^2\psi^{B_1}}{\psi^{B_1}+(1-\psi)^{B_1}}-\frac{{B_1}^2\psi^{2B_1}}{[\psi^{B_1}+(1-\psi)^{B_1}]^2}\right]=0,\quad \frac{{B_1}^2\psi^{B_1}}{\psi^{B_1}+(1-\psi)^{B_1}}-\frac{{B_1}^2\psi^{2B_1}}{[\psi^{B_1}+(1-\psi)^{B_1}]^2}=\mathcal{O}(\psi^{B_1-1})$$
$$\lim_{\psi\to 1^-}\left(\lim_{\omega\to 0^+} E[S]\right)=\lim_{\psi\to 1^-}\left[\frac{B_1\psi^{B_1}}{\psi^{B_1}+(1-\psi)^{B_1}}\right]=B_1,\quad \frac{B_1\psi^{B_1}}{\psi^{B_1}+(1-\psi)^{B_1}}=\mathcal{O}(\psi-1)$$
$$\lim_{\psi\to 1^-}\left(\lim_{\omega\to 0^+} V[S]\right)=\lim_{\psi\to 1^-}\left[\frac{{B_1}^2\psi^{B_1}}{\psi^{B_1}+(1-\psi)^{B_1}}-\frac{{B_1}^2\psi^{2B_1}}{[\psi^{B_1}+(1-\psi)^{B_1}]^2}\right]=0,\quad \frac{{B_1}^2\psi^{B_1}}{\psi^{B_1}+(1-\psi)^{B_1}}-\frac{{B_1}^2\psi^{2B_1}}{[\psi^{B_1}+(1-\psi)^{B_1}]^2}=\mathcal{O}(\psi^{B_1})$$

Case 2.1: $n=2k$:

$$\begin{aligned}\lim_{\omega\to+\infty}\tau_j&=\lim_{\omega\to+\infty}\frac{\sum_{i=0}^{B_1-j}\binom{B_1-j}{i}\psi^i(1-\psi)^{B_1-j-i}\omega^{(B_1-j-i)(i+j)}}{\sum_{i=0}^{B_1}\binom{B_1}{i}\psi^i(1-\psi)^{B_1-i}\omega^{(B_1-i)i}}\\&=\frac{\lim_{\omega\to+\infty}\sum_{i=0}^{B_1-j}\binom{B_1-j}{i}\psi^i(1-\psi)^{B_1-j-i}\omega^{(B_1-j-i)(i+j)}}{\lim_{\omega\to+\infty}\sum_{i=0}^{B_1}\binom{B_1}{i}\psi^i(1-\psi)^{B_1-i}\omega^{(B_1-i)i}}\\&=\frac{1}{\psi^j}\frac{\binom{B_1-j}{\frac{B_1}{2}-j}}{\binom{B_1}{\frac{B_1}{2}}},\qquad j\le\frac{B_1}{2},\qquad \tau_j=\mathcal{O}\left(\frac{1}{\omega}\right).\end{aligned}$$

Therefore,

$$\lim_{\omega\to+\infty}E[S]=\lim_{\omega\to+\infty}B_1\psi\tau_1=B_1\psi\lim_{\omega\to+\infty}\tau_1=B_1\psi\frac{1}{2\psi}=\frac{B_1}{2}$$
$$\lim_{\omega\to+\infty}V[S]=\lim_{\omega\to+\infty}B_1\psi\eta=\lim_{\omega\to+\infty}B_1\psi[\tau_1-\psi(B_1\tau_1^2-(B_11)\tau_2)]=B_1\psi[\lim_{\omega\to+\infty}\tau_1-\psi(B_1\lim_{\omega\to+\infty}\tau_1^2-(B_1-1)\lim_{\omega\to+\infty}\tau_2)]$$
$$\begin{aligned}&=B_1\psi\left[\frac{1}{2\psi}-\psi B_1\left(\frac{1}{2\psi}\right)^2+\psi(B_1-1)\frac{B_1-2}{4(B_1-1)\psi^2}\right]\\&=B_1\psi\left[\frac{1}{2\psi}-\psi B_1\frac{1}{4\psi^2}+\frac{B_1-2}{4\psi}\right]\\&=B_1\psi\left[\frac{1}{2\psi}-\frac{B_1}{4\psi}+\frac{B_1-2}{4\psi}\right]\\&=B_1\psi\left[\frac{2-B_1+B_1-2}{4\psi}\right]=0\end{aligned}$$

Case 2.2: $n=2k+1$:

$$\begin{aligned}\lim_{\omega\to+\infty}\tau_j&=\lim_{\omega\to+\infty}\frac{\sum_{i=0}^{B_1-j}\binom{B_1-j}{i}\psi^i(1-\psi)^{B_1-j-i}\omega^{(B_1-j-i)(i+j)}}{\sum_{i=0}^{B_1}\binom{B_1}{i}\psi^i(1-\psi)^{B_1-i}\omega^{(B_1-i)i}}\\&=\frac{\lim_{\omega\to+\infty}\sum_{i=0}^{B_1-j}\binom{B_1-j}{i}\psi^i(1-\psi)^{B_1-j-i}\omega^{(B_1-j-i)(i+j)}}{\lim_{\omega\to+\infty}\sum_{i=0}^{B_1}\binom{B_1}{i}\psi^i(1-\psi)^{B_1-i}\omega^{(B_1-i)i}}\\&=\frac{1}{\psi^j}\frac{R^{-1}+\psi}{\frac{\binom{B_1}{\frac{B_1-1}{2}}}{\binom{B_1-j}{\frac{B_1-1}{2}-j}}R^{-1}},\qquad j\le\frac{B_1-1}{2},\qquad \tau_j=\mathcal{O}\left(\frac{1}{\omega^2}\right)\end{aligned}$$

where $R = \frac{\binom{B_1-j}{\frac{B_1-1}{2}-j+1}}{\binom{B_1-j}{\frac{B_1-1}{2}-j}} - 1$

Therefore,

$$\lim_{\omega\to+\infty} E[S] = \lim_{\omega\to+\infty} B_1\psi\tau_1 = B_1\psi \lim_{\omega\to+\infty} \tau_1 = B_1\psi\frac{\frac{B_1-1}{2}+\psi}{B_1\psi} = \frac{B_1-1}{2}+\psi$$

$$\lim_{\omega\to+\infty} V[S] = \lim_{\omega\to+\infty} B_1\psi\eta = \lim_{\omega\to+\infty} B_1\psi[\tau_1 - \psi(B_1\tau_1^2 - (B_1-1)\tau_2)] = B_1\psi[\lim_{\omega\to+\infty}\tau_1 - \psi(B_1\lim_{\omega\to+\infty}\tau_1^2 - (B_1-1)\lim_{\omega\to+\infty}\tau_2)]$$

$$= B_1\psi\left[\frac{\frac{B_1-1}{2}+\psi}{B_1\psi} - B_1\psi\left(\frac{\frac{B_1-1}{2}+\psi}{B_1\psi}\right)^2 + (B_1-1)\psi\left(\frac{\frac{B_1-3}{4}+\psi}{B_1\psi^2}\right)\right]$$

$$= B_1\psi\left[\frac{\frac{B_1-1}{2}+\psi}{B_1\psi} - \frac{(\frac{B_1-1}{2}+\psi)^2}{B_1\psi} + \frac{(B_1-1)\left(\frac{B_1-3}{4}+\psi\right)}{B_1\psi}\right]$$

$$= B_1\psi\left[\frac{\frac{B_1-1}{2}+\psi - \frac{(B_1-1)^2}{4} - \psi^2 - (B_1-1)\psi + \frac{(B_1-1)(B_1-3)}{4} + (B_1-1)\psi}{B_1\psi}\right]$$

$$= B_1\psi\left[\frac{2B_1 - 2 + 4\psi - {B_1}^2 - 1 + 2B_1 - 4\psi^2 + {B_1}^2 - 3B_1 - B_1 + 3}{4B_1\psi}\right]$$

$$= B_1\psi\left[\frac{4\psi - 4\psi^2}{4B_1\psi}\right] = \psi(1-\psi)$$

It follows that:

$$\lim_{\psi\to0^+}\left(\lim_{\omega\to+\infty} E[S]\right) = \lim_{\psi\to0^+}\left[\frac{B_1-1}{2}+\psi\right] = \frac{B_1-1}{2}, \quad \frac{B_1-1}{2}+\psi = \mathcal{O}(\psi)$$

$$\lim_{\psi\to0^+}\left(\lim_{\omega\to+\infty} V[S]\right) = \lim_{\psi\to0^+}[\psi(1-\psi)] = 0, \quad \psi(1-\psi) = \mathcal{O}(\psi^2)$$

$$\lim_{\psi\to1^-}\left(\lim_{\omega\to+\infty} E[S]\right) = \lim_{\psi\to1^-}\left[\frac{B_1-1}{2}+\psi\right] = \frac{B_1-1}{2}, \quad \frac{B_1-1}{2}+\psi = \mathcal{O}(\psi-1)$$

$$\lim_{\psi\to1^-}\left(\lim_{\omega\to+\infty} V[S]\right) = \lim_{\psi\to1^-}[\psi(1-\psi)] = 0, \quad \psi(1-\psi) = \mathcal{O}(\psi-1)$$

Combining these results, it is straightforward to notice that, in all the cases where the limit of the variance is equal to 0, the random variable S degenerates to the limit of its expectation, L_E, with probability 1. Formally,

$$P(S=s) = \begin{cases} 1 & \text{if } s = L_E \\ 0 & \text{otherwise} \end{cases} \quad \Longrightarrow \quad S \xrightarrow[\substack{\omega\to0^+ \vee \omega\to+\infty \\ \psi\to0^+ \vee \psi\to1^-}]{} \delta[L_E]$$

where δ is the Dirac-Delta function $\delta_{x_0}[\phi] = \phi(x_0)$.

□

A.2 Proof of Proposition 1

Proof. Because of the symmetry of the joint distribution of $(C_1, ..., C_{B_1})$, it is always possible to write

$$E[S] = B_1 E[C_1]$$

and

$$V[S] = B_1 V[C_1] + B_1(B_1 - 1)Cov[C_1, C_2]$$

Therefore, the Central Limit Theorem for dependent random variables can be applied to S, provided that the overall mean and variance behave 'sensibly'. Specifically, we refer to the central limit theorem for dependent classes of random variables derived by Kaminski in [64]:

Theorem. *Let $\{X_i\}_{i\geq 1}$ be a sequence of identically distributed random variables such that $E|X_1|^{2+\epsilon} < +\infty$ for some $\epsilon > 0$. Let $V[X_1] = \sigma^2$ and ϵ_1 be a positive number such that $\epsilon_1 < \frac{\epsilon}{2(1+\epsilon)}$. Denote by $S = \sum_{i=1}^{n} X_i$ the partial sum. Suppose that for sufficiently large k, the inequality*

$$\sup\left\{ |P\left(\bigcap_{i=1}^{j}\{X_{v_i} \leq x_{v_i}\}\right) - \prod_{i=1}^{j} P(X_{v_i} \leq x_{v_i}) : (x_{v_1}, \ldots, x_{v_j}) \in \mathbb{R}^j| \right\} \leq (1-k^{-\epsilon_1})^{k-k^{\epsilon_1}-j} \tag{A.1}$$

holds, where $v_1, \ldots, v_j$ is any choice of indices such that $k^{\epsilon_1} < v_1 < \cdots < v_j \leq k$. Then:

$$\frac{Y_n - E[Y_n]}{\sigma\sqrt{n}} \xrightarrow{d} N(0,1) \qquad as\ n \to \infty.$$

It is important to underline that the left-hand side of condition A.1 is only on the tail $X_{k^{\epsilon_1}}, X_{k^{\epsilon_1}+1}, \ldots, X_k$ and it reflects the degree of dependence among $X_{v_1}, \ldots, X_{v_j}$ (i.e. if $X_{v_1}, \ldots, X_{v_j}$ are independent, the left-hand side of inequality A.1 is 0, otherwise it is a real positive number). Then, it is easy to see that, for fixed k, the right-hand side of A.1 tends to become larger as j increases.

In our case, a different number B_1 of Bernoulli variables $C_1, ..., C_{B_1}$ is required so as to satisfy inequality A.1, depending on their average degree of dependence, ω. Condition A.1 surely holds for any ω if $n \to \infty$. □

A.3 Proof of Theorem 2

Proof.

$$\pi(\psi, \omega) = \psi\tau_1$$

Now,

$$\tau_1 = \frac{\sum_{i=0}^{B_1-1} \binom{B_1-1}{i} \psi^i (1-\psi)^{B_1-1-i} \omega^{(B_1-1-i)(i+1)}}{\sum_{i=0}^{B_1} \binom{B_1}{i} \psi^i (1-\psi)^{B_1-i} \omega^{(B_1-i)i}} \leq 1 \iff$$

$$D_{B_1} = \sum_{i=0}^{B_1-1} \binom{B_1-1}{i} \psi^i (1-\psi)^{B_1-1-i} \omega^{(B_1-1-i)(i+1)} - \sum_{i=0}^{B_1} \binom{B_1}{i} \psi^i (1-\psi)^{B_1-i} \omega^{(B_1-i)i} \leq 0 \tag{A.2}$$

The difference D_{B_1} can be factored as:

$$D_{B_1} = \begin{cases} \Delta(\psi-1)(2\psi-1)(\omega-1) & \text{if } B_1 \text{ is even} \\ \Delta(\psi-1)(2\psi-1)(\omega-1)(\omega+1) & \text{if } B_1 \text{ is odd} \end{cases} \tag{A.3}$$

where Δ is a positive polynomial (only numeric proofs are possible and they are given in Tables A.1-A.4 and in Figure A.2).
By expressions A.2-A.3, it follows that:

$$\tau_1 \leq 1 \iff D_{B_1} \leq 0 \iff \begin{cases} 0 \leq \psi \leq \frac{1}{2} \wedge 0 \leq \omega \leq 1 \\ \frac{1}{2} \leq \psi < 1 \wedge \omega \geq 1 \end{cases}$$

This result is also shown in Figure A.2, where red and black points correspond respectively to $\tau_1 \leq 1$ and $\tau_1 > 1$, for different sample size B_1. □

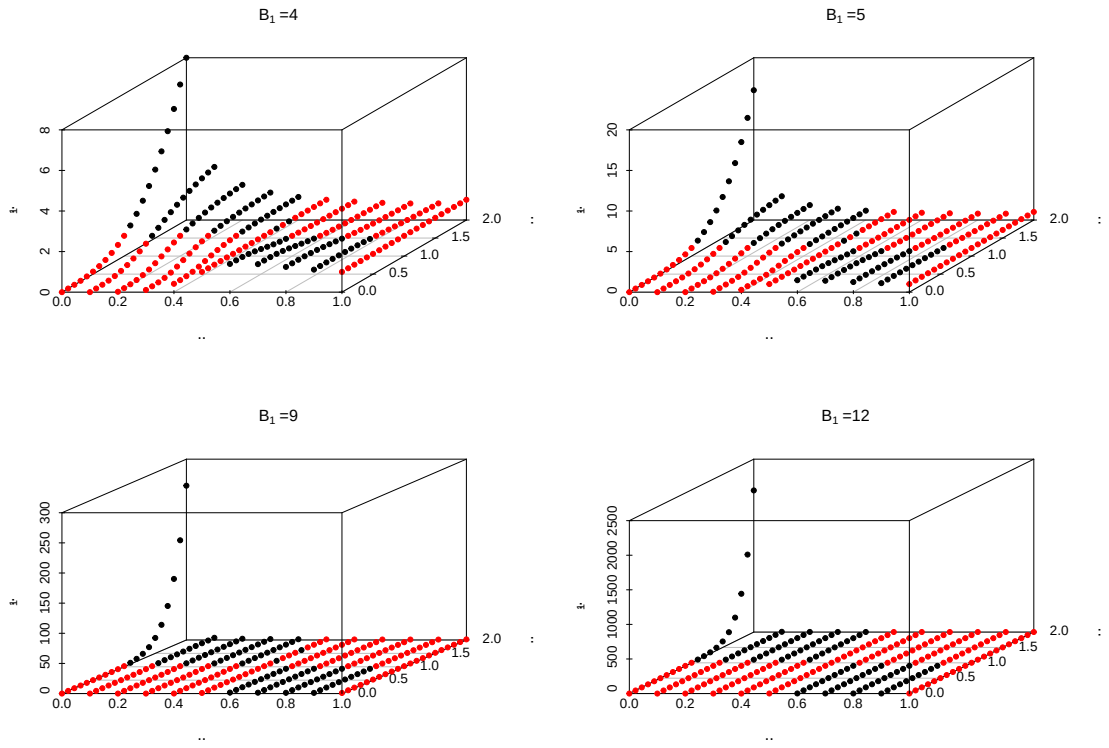


Figure A.1: Values of τ_1 for $\psi \in [0,1]$, $\omega \in [0,2]$. Red and black points correspond respectively to $\tau_1 \leq 1$ and $\tau_1 > 1$.

ω \ ψ	0.	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.
0.	1.00	0.910	0.840	0.790	0.760	0.750	0.760	0.790	0.840	0.910	1.00
0.1	1.11	1.01	0.933	0.878	0.844	0.833	0.844	0.878	0.933	1.01	1.11
0.2	1.24	1.13	1.05	0.985	0.948	0.936	0.948	0.985	1.05	1.13	1.24
0.3	1.39	1.27	1.18	1.12	1.08	1.06	1.08	1.12	1.18	1.27	1.39
0.4	1.56	1.44	1.34	1.27	1.23	1.22	1.23	1.27	1.34	1.44	1.56
0.5	1.75	1.63	1.53	1.46	1.42	1.41	1.42	1.46	1.53	1.63	1.75
0.6	1.96	1.84	1.75	1.68	1.65	1.63	1.65	1.68	1.75	1.84	1.96
0.7	2.19	2.09	2.00	1.95	1.91	1.90	1.91	1.95	2.00	2.09	2.19
0.8	2.44	2.36	2.30	2.25	2.22	2.21	2.22	2.25	2.30	2.36	2.44
0.9	2.71	2.66	2.63	2.60	2.58	2.58	2.58	2.60	2.63	2.66	2.71
1.	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
1.1	3.31	3.37	3.42	3.45	3.47	3.48	3.47	3.45	3.42	3.37	3.31
1.2	3.64	3.78	3.89	3.96	4.01	4.03	4.01	3.96	3.89	3.78	3.64
1.3	3.99	4.22	4.41	4.54	4.61	4.64	4.61	4.54	4.41	4.22	3.99
1.4	4.36	4.71	4.98	5.17	5.29	5.33	5.29	5.17	4.98	4.71	4.36
1.5	4.75	5.23	5.61	5.88	6.04	6.09	6.04	5.88	5.61	5.23	4.75
1.6	5.16	5.80	6.30	6.66	6.87	6.94	6.87	6.66	6.30	5.80	5.16
1.7	5.59	6.41	7.05	7.51	7.79	7.88	7.79	7.51	7.05	6.41	5.59
1.8	6.04	7.07	7.87	8.45	8.79	8.90	8.79	8.45	7.87	7.07	6.04
1.9	6.51	7.78	8.76	9.46	9.89	10.0	9.89	9.46	8.76	7.78	6.51
2.	7.00	8.53	9.72	10.6	11.1	11.3	11.1	10.6	9.72	8.53	7.00

Table A.1: Values of Δ for $\psi \in [0,1]$, $\omega \in [0,2]$ and $B_1 = 4$.

ω \ ψ	0.	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.
0.	1.00	0.820	0.680	0.580	0.520	0.500	0.520	0.580	0.680	0.820	1.00
0.1	1.01	0.828	0.687	0.586	0.525	0.505	0.525	0.586	0.687	0.828	1.01
0.2	1.04	0.853	0.708	0.605	0.542	0.522	0.542	0.605	0.708	0.853	1.04
0.3	1.09	0.897	0.746	0.639	0.575	0.553	0.575	0.639	0.746	0.897	1.09
0.4	1.16	0.960	0.805	0.694	0.628	0.606	0.628	0.694	0.805	0.960	1.16
0.5	1.25	1.05	0.890	0.777	0.710	0.688	0.710	0.778	0.890	1.05	1.25
0.6	1.36	1.16	1.01	0.898	0.832	0.810	0.832	0.898	1.01	1.16	1.36
0.7	1.49	1.31	1.17	1.07	1.01	0.985	1.01	1.07	1.17	1.31	1.49
0.8	1.64	1.49	1.38	1.30	1.25	1.23	1.25	1.30	1.38	1.49	1.64
0.9	1.81	1.72	1.65	1.60	1.57	1.56	1.57	1.60	1.65	1.72	1.81
1.	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1.1	2.21	2.34	2.44	2.51	2.55	2.57	2.55	2.51	2.44	2.34	2.21
1.2	2.44	2.75	2.99	3.16	3.26	3.29	3.26	3.16	2.99	2.75	2.44
1.3	2.69	3.23	3.66	3.96	4.14	4.20	4.14	3.96	3.66	3.23	2.69
1.4	2.96	3.81	4.47	4.94	5.23	5.32	5.23	4.94	4.47	3.81	2.96
1.5	3.25	4.49	5.45	6.14	6.55	6.69	6.55	6.14	5.45	4.49	3.25
1.6	3.56	5.28	6.62	7.57	8.14	8.33	8.14	7.57	6.62	5.28	3.56
1.7	3.89	6.20	7.99	9.27	10.0	10.3	10.0	9.27	7.99	6.20	3.89
1.8	4.24	7.26	9.60	11.3	12.3	12.6	12.3	11.3	9.60	7.26	4.24
1.9	4.61	8.47	11.5	13.6	14.9	15.3	14.9	13.6	11.5	8.47	4.61
2.	5.00	9.86	13.6	16.3	18.0	18.5	18.0	16.3	13.6	9.86	5.00

Table A.2: Values of Δ for $\psi \in [0,1]$, $\omega \in [0,2]$ and $B_1 = 5$.

ω \ ψ	0.	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.
0.	1.00	0.538	0.280	0.144	0.0807	0.0625	0.0807	0.144	0.280	0.538	1.00
0.1	1.01	0.544	0.282	0.145	0.0815	0.0631	0.0815	0.145	0.282	0.544	1.01
0.2	1.04	0.561	0.291	0.150	0.0841	0.0651	0.0841	0.150	0.291	0.561	1.04
0.3	1.10	0.591	0.307	0.158	0.0887	0.0687	0.0887	0.158	0.307	0.591	1.10
0.4	1.19	0.641	0.333	0.172	0.0963	0.0747	0.0963	0.172	0.333	0.641	1.19
0.5	1.33	0.717	0.374	0.194	0.109	0.0850	0.109	0.194	0.374	0.717	1.33
0.6	1.54	0.839	0.444	0.234	0.135	0.106	0.135	0.234	0.444	0.839	1.54
0.7	1.85	1.04	0.578	0.323	0.200	0.163	0.200	0.323	0.578	1.04	1.85
0.8	2.31	1.42	0.878	0.563	0.401	0.352	0.401	0.563	0.878	1.42	2.31
0.9	3.00	2.20	1.67	1.33	1.14	1.08	1.14	1.33	1.67	2.20	3.00
1.	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
1.1	5.45	8.48	11.3	13.5	15.0	15.5	15.0	13.5	11.3	8.48	5.45
1.2	7.50	20.2	34.3	46.9	55.4	58.5	55.4	46.9	34.3	20.2	7.50
1.3	10.4	51.5	106.	158.	195.	209.	195.	158.	106.	51.5	10.4
1.4	14.3	134.	318.	508.	648.	700.	648.	508.	318.	134.	14.3
1.5	19.7	349.	926.	1.55×10^3	2.03×10^3	2.20×10^3	2.03×10^3	1.55×10^3	926.	349.	19.7
1.6	26.9	892.	2.58×10^3	4.50×10^3	5.99×10^3	6.54×10^3	5.99×10^3	4.50×10^3	2.58×10^3	892.	26.9
1.7	36.4	2.22×10^3	6.91×10^3	1.24×10^4	1.67×10^4	1.84×10^4	1.67×10^4	1.24×10^4	6.91×10^3	2.22×10^3	36.4
1.8	48.7	5.36×10^3	1.77×10^4	3.26×10^4	4.45×10^4	4.89×10^4	4.45×10^4	3.26×10^4	1.77×10^4	5.36×10^3	48.7
1.9	64.7	1.26×10^4	4.35×10^4	8.19×10^4	1.13×10^5	1.24×10^5	1.13×10^5	8.19×10^4	4.35×10^4	1.26×10^4	64.7
2.	85.0	2.86×10^4	1.03×10^5	1.97×10^5	2.74×10^5	3.03×10^5	2.74×10^5	1.97×10^5	1.03×10^5	2.86×10^4	85.0

Table A.3: Values of Δ for $\psi \in [0,1]$, $\omega \in [0,2]$ and $B_1 = 9$.

ω \ ψ	0.	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.
0.	1.00	0.392	0.143	0.0494	0.0179	0.0107	0.0179	0.0494	0.143	0.392	1.00
0.1	1.11	0.436	0.159	0.0549	0.0199	0.0119	0.0199	0.0549	0.159	0.436	1.11
0.2	1.25	0.490	0.179	0.0618	0.0224	0.0134	0.0224	0.0618	0.179	0.490	1.25
0.3	1.43	0.560	0.205	0.0706	0.0256	0.0153	0.0256	0.0706	0.205	0.560	1.43
0.4	1.67	0.654	0.239	0.0824	0.0299	0.0179	0.0299	0.0824	0.239	0.654	1.67
0.5	2.00	0.785	0.287	0.0990	0.0360	0.0216	0.0360	0.0990	0.287	0.785	2.00
0.6	2.49	0.981	0.360	0.125	0.0459	0.0277	0.0459	0.125	0.360	0.981	2.49
0.7	3.27	1.31	0.494	0.178	0.0680	0.0423	0.0680	0.178	0.494	1.31	3.27
0.8	4.57	1.98	0.827	0.341	0.157	0.111	0.157	0.341	0.827	1.98	4.57
0.9	6.86	3.81	2.16	1.31	0.907	0.791	0.907	1.31	2.16	3.81	6.86
1.	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
1.1	18.5	51.4	92.9	133.	162.	173.	162.	133.	92.9	51.4	18.5
1.2	32.2	335.	941.	1.67×10^3	2.27×10^3	2.49×10^3	2.27×10^3	1.67×10^3	941.	335.	32.2
1.3	56.4	2.52×10^3	9.54×10^3	1.94×10^4	2.81×10^4	3.15×10^4	2.81×10^4	1.94×10^4	9.54×10^3	2.52×10^3	56.4
1.4	98.7	1.93×10^4	9.01×10^4	2.01×10^5	3.03×10^5	3.44×10^5	3.03×10^5	2.01×10^5	9.01×10^4	1.93×10^4	98.7
1.5	171.	1.43×10^5	7.74×10^5	1.84×10^6	2.86×10^6	3.28×10^6	2.86×10^6	1.84×10^6	7.74×10^5	1.43×10^5	171.
1.6	292.	9.88×10^5	6.02×10^6	1.50×10^7	2.39×10^7	2.76×10^7	2.39×10^7	1.50×10^7	6.02×10^6	9.88×10^5	292.
1.7	488.	6.36×10^6	4.24×10^7	1.10×10^8	1.78×10^8	2.07×10^8	1.78×10^8	1.10×10^8	4.24×10^7	6.36×10^6	488.
1.8	802.	3.80×10^7	2.73×10^8	7.29×10^8	1.20×10^9	1.39×10^9	1.20×10^9	7.29×10^8	2.73×10^8	3.80×10^7	802.
1.9	1.29×10^3	2.11×10^8	1.61×10^9	4.40×10^9	7.30×10^9	8.54×10^9	7.30×10^9	4.40×10^9	1.61×10^9	2.11×10^8	1.29×10^3
2.	2.05×10^3	1.09×10^9	8.73×10^9	2.44×10^{10}	4.09×10^{10}	4.80×10^{10}	4.09×10^{10}	2.44×10^{10}	8.73×10^9	1.09×10^9	2.05×10^3

Table A.4: Values of Δ for $\psi \in [0,1]$, $\omega \in [0,2]$ and $B_1 = 12$.

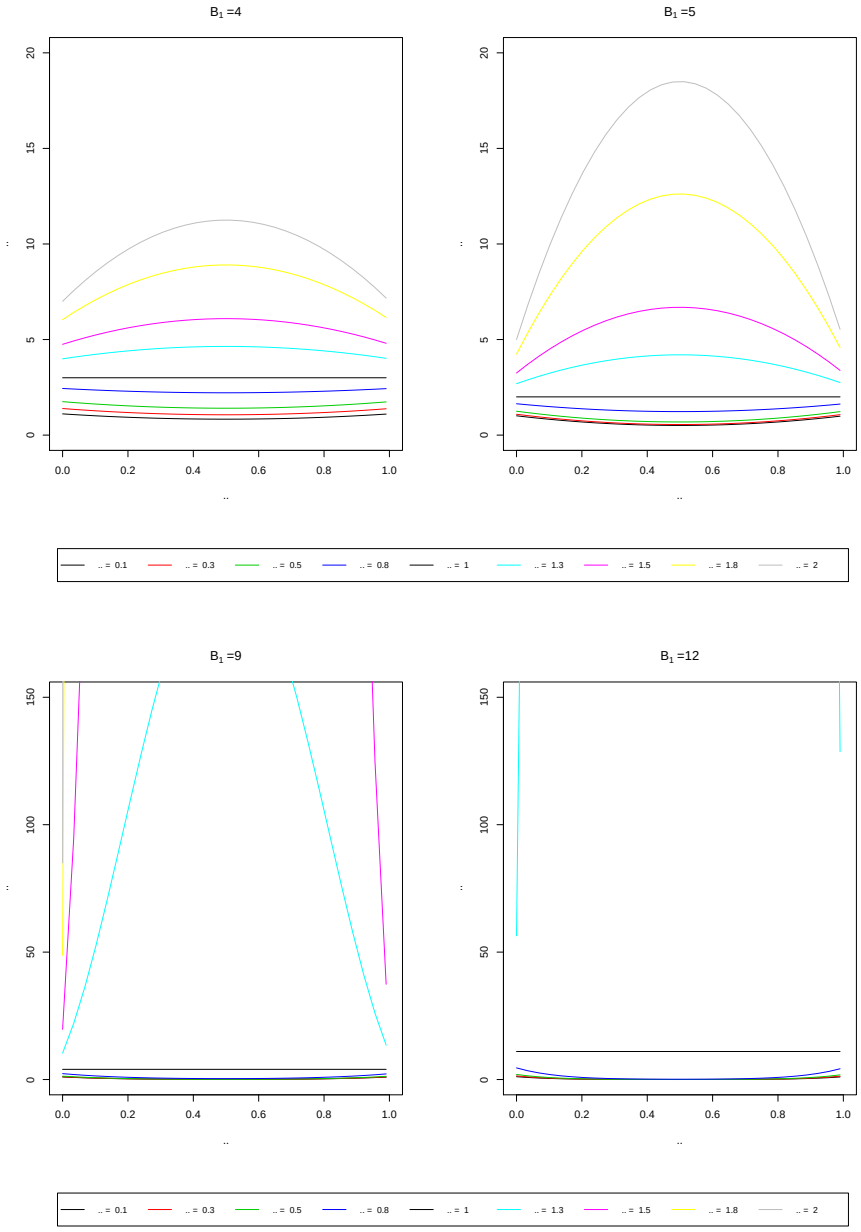


Figure A.2: Values of Δ *for* $\psi \in [0, 1]$ *according to different values of* ω *and* B_1.

Appendix B

B.1 RP-VIP and AA-RP ensemble classifiers variable selection in real data applications

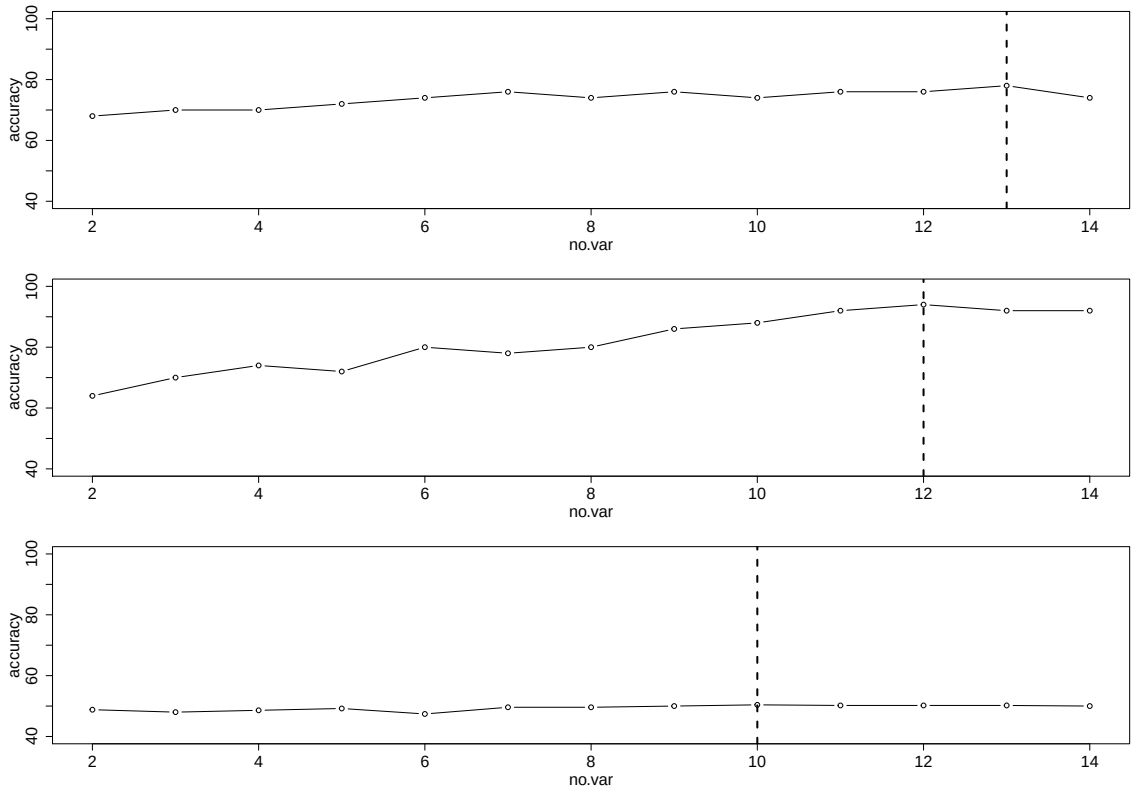


Figure B.1: Accuracy rate (_____) and optimal number of variables, h, (.....) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the eye state data with $n = 50$ and $d = 2$.

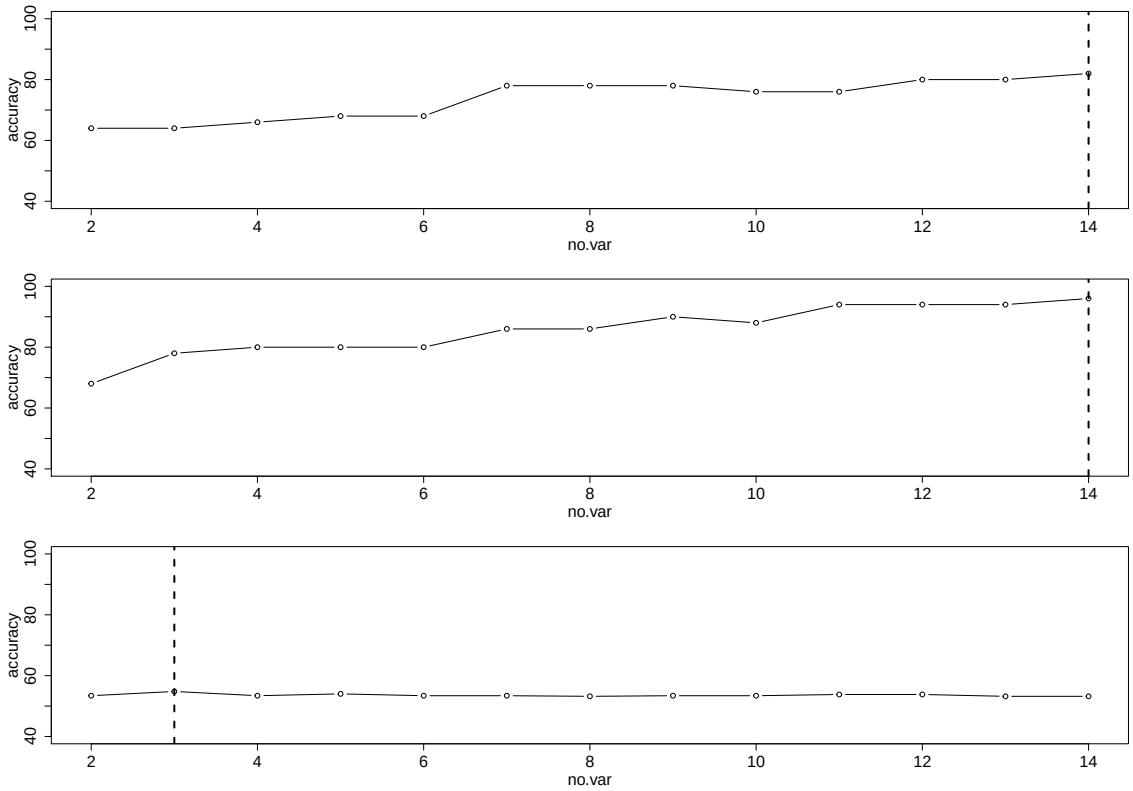


Figure B.2: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-K nn (third row) for the eye state data with $n = 50$ and $d = 5$.

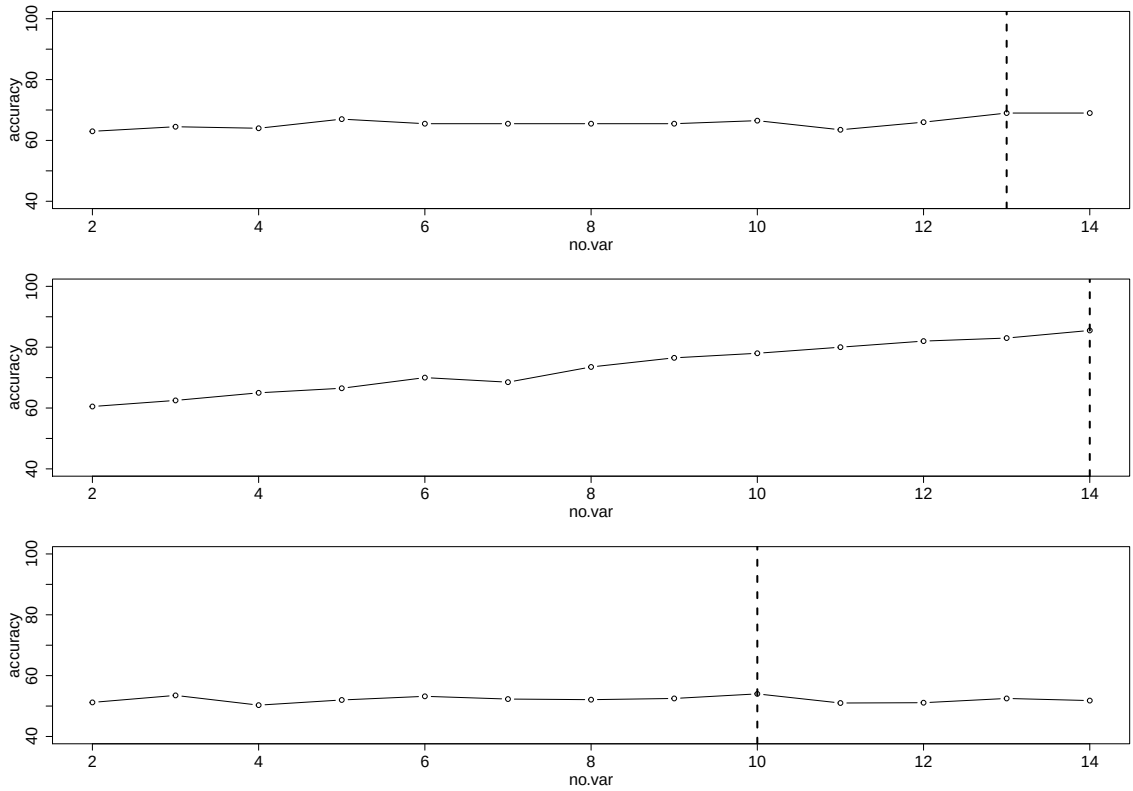


Figure B.3: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the eye state data with $n = 200$ *and* $d = 2$.

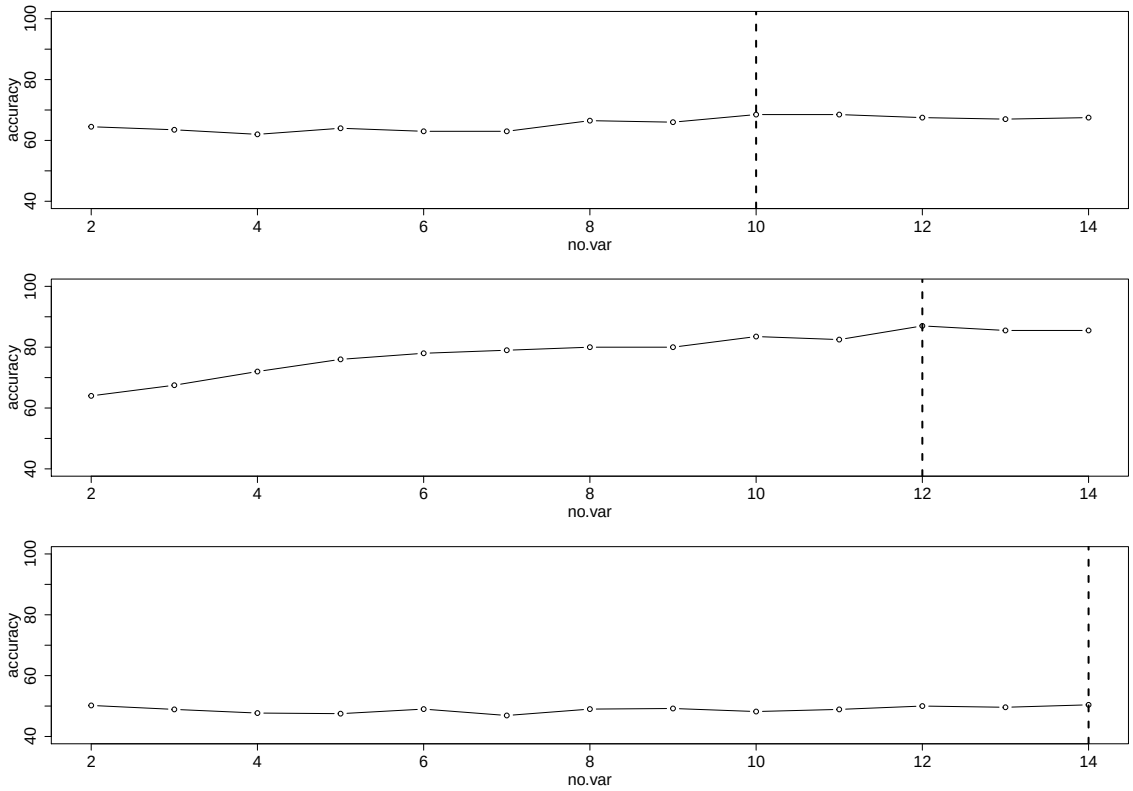


Figure B.4: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-K nn (third row) for the eye state data with $n = 200$ and $d = 5$.

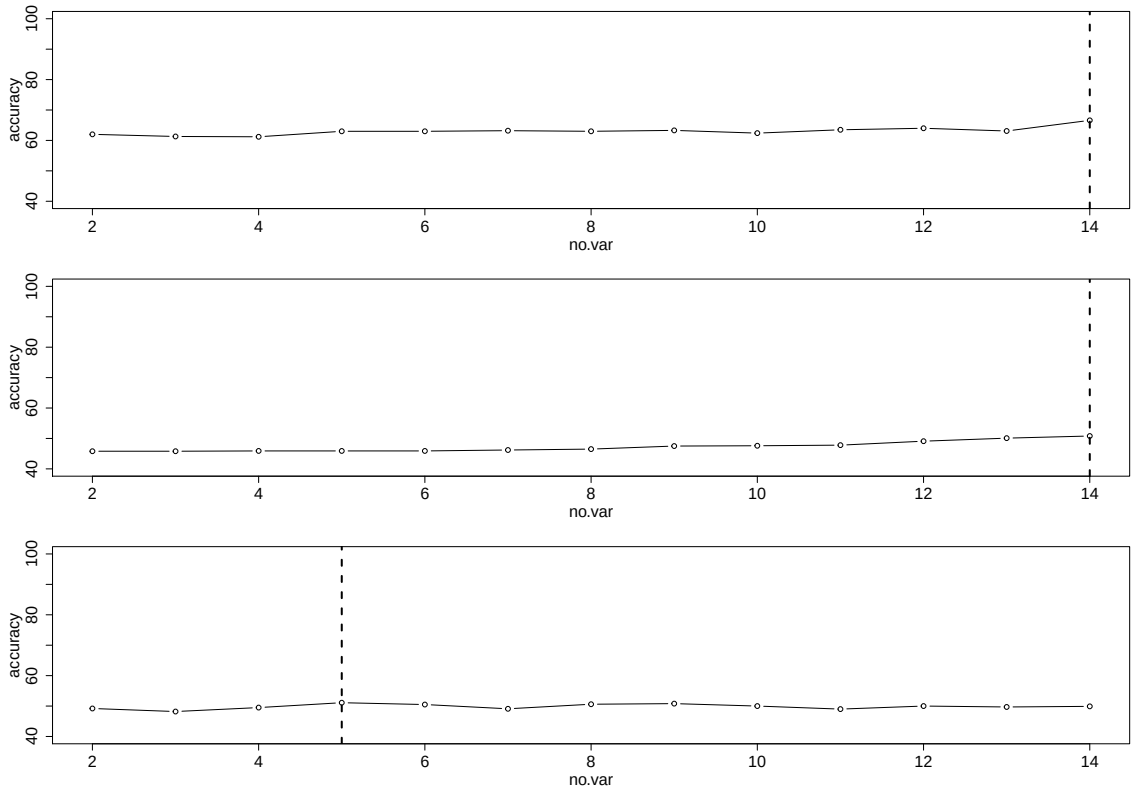


Figure B.5: Accuracy rate (_____) and optimal number of variables, h, (.....) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-K nn (third row) for the eye state data with $n = 1000$ and $d = 2$.

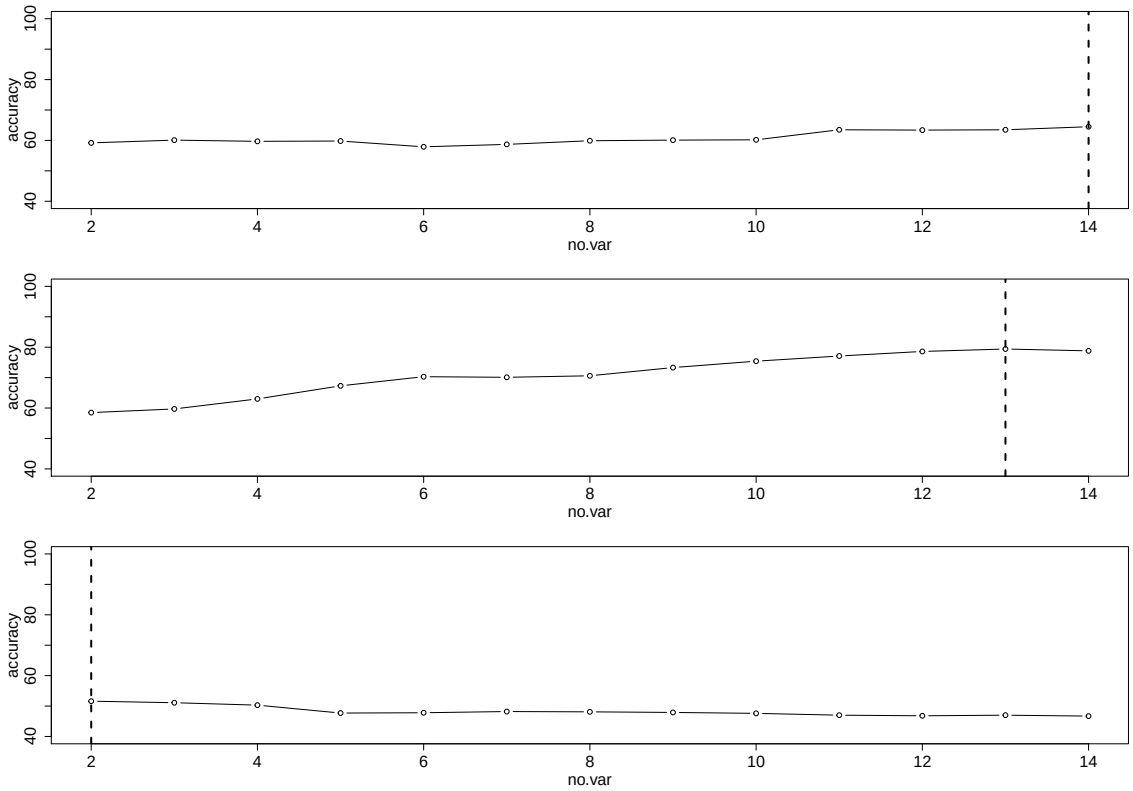


Figure B.6: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-K nn (third row) for the eye state data with $n = 1000$ and $d = 5$.

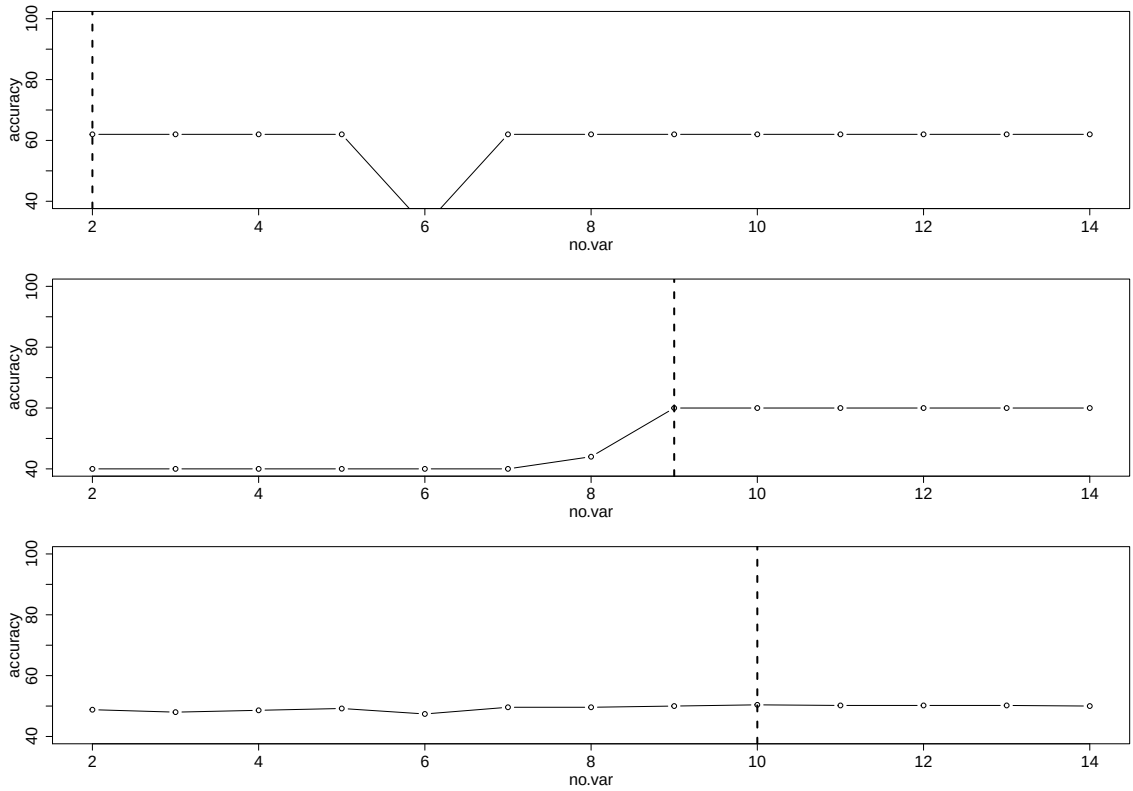


Figure B.7: Accuracy rate (_____) and optimal number of variables, h, (.....) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the eye state data with $n = 50$ and $d = 2$.

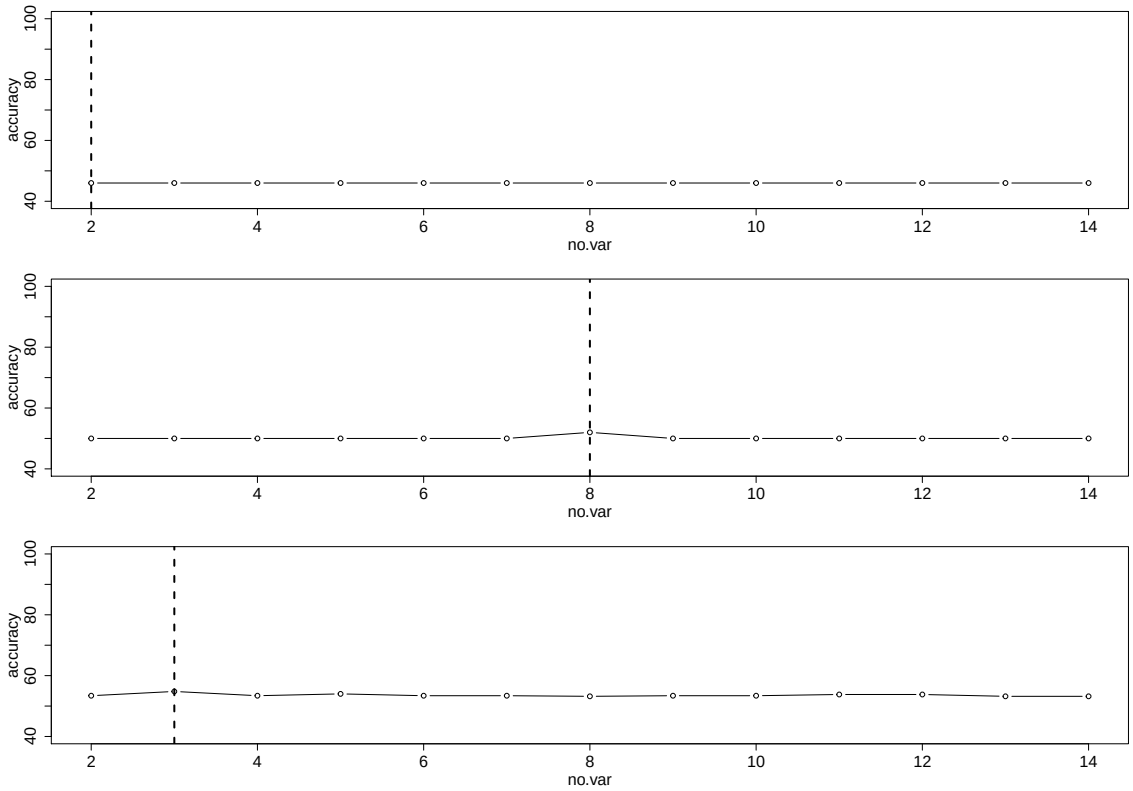


Figure B.8: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the eye state data with $n = 50$ and $d = 5$.

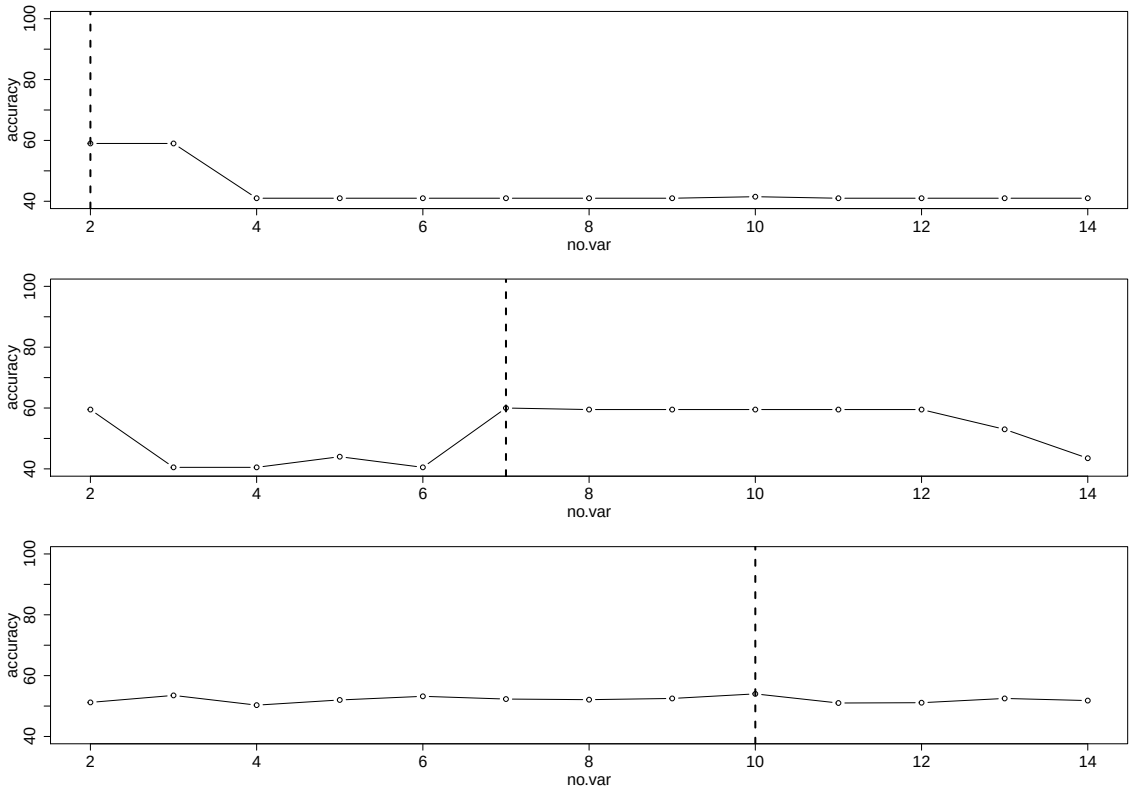


Figure B.9: Accuracy rate (_____) and optimal number of variables, h, (.....) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the eye state data with $n = 200$ *and* $d = 2$.

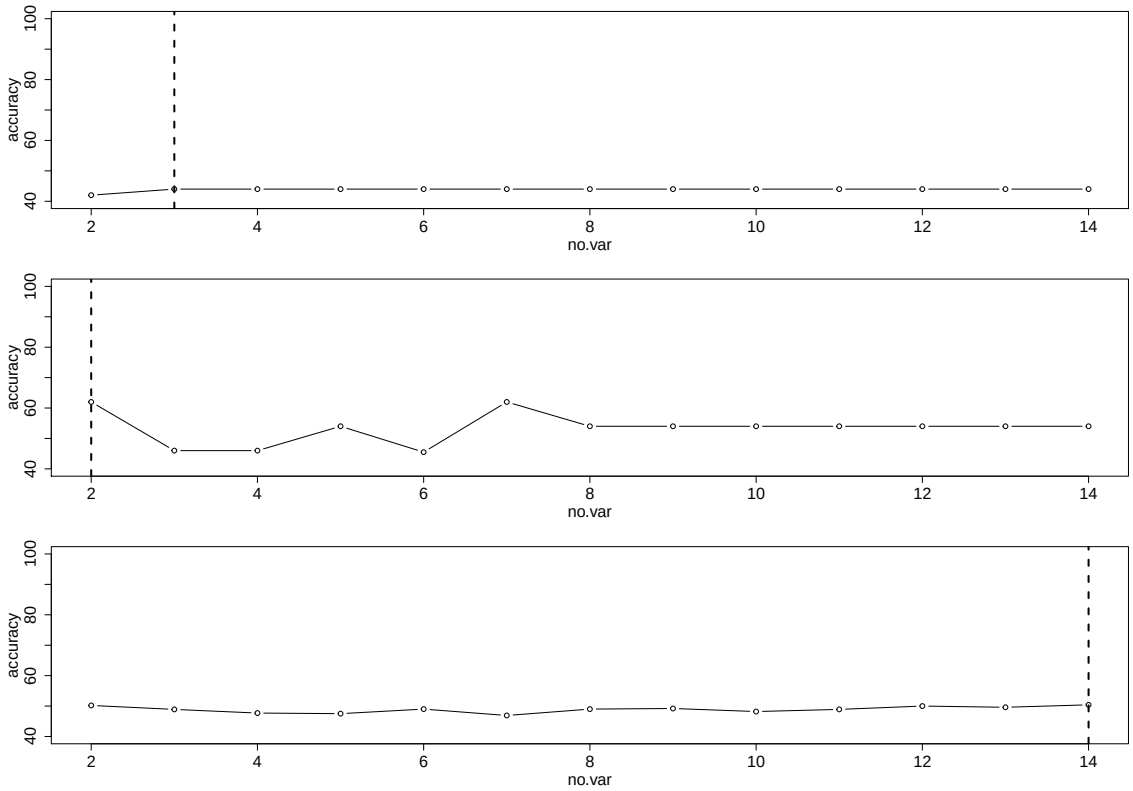


Figure B.10: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and VIP-Knn (third row) for the eye state data with $n = 200$ and $d = 5$.

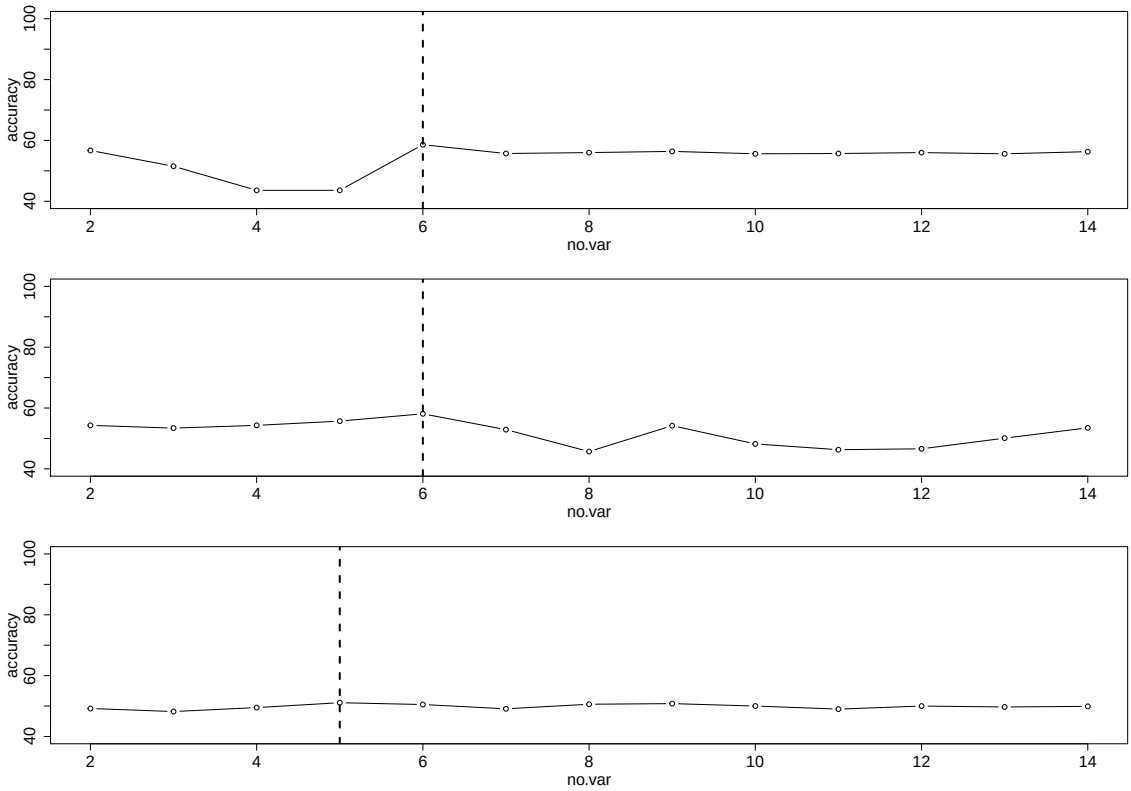


Figure B.11: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and VIP-Knn (third row) for the eye state data with $n = 1000$ *and* $d = 2$.

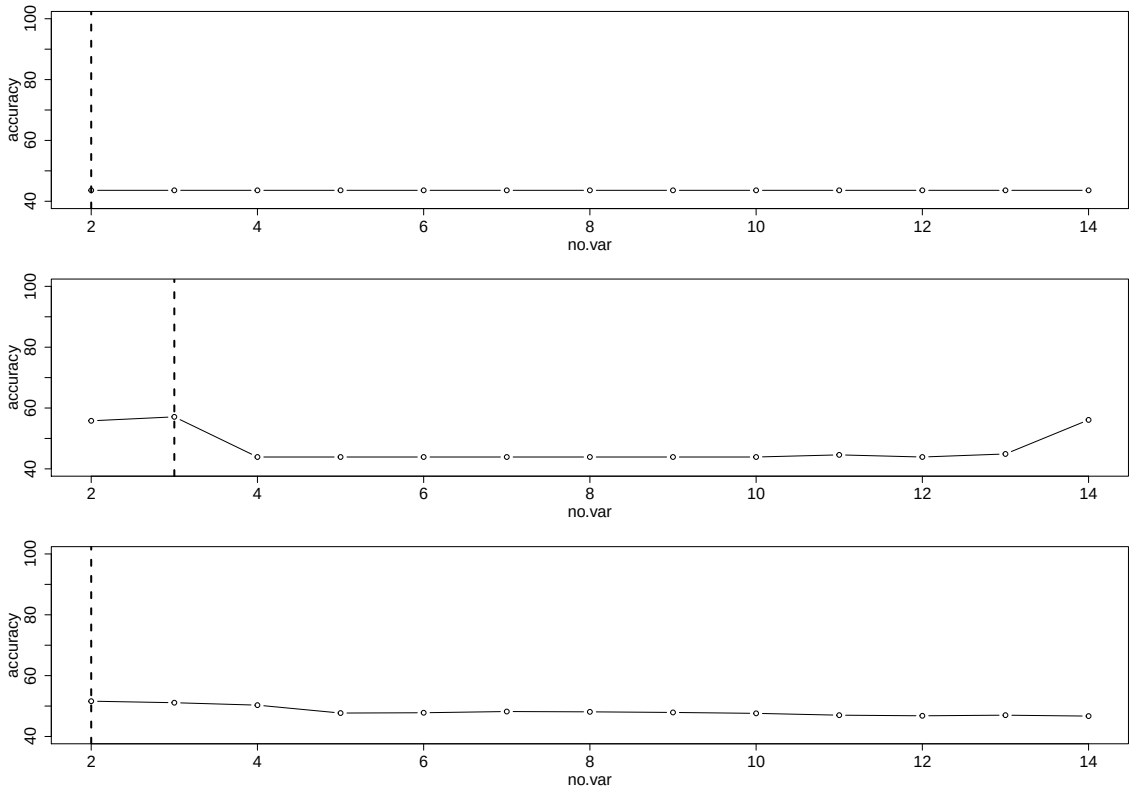


Figure B.12: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and VIP-Knn (third row) for the eye state data with $n = 1000$ and $d = 5$.

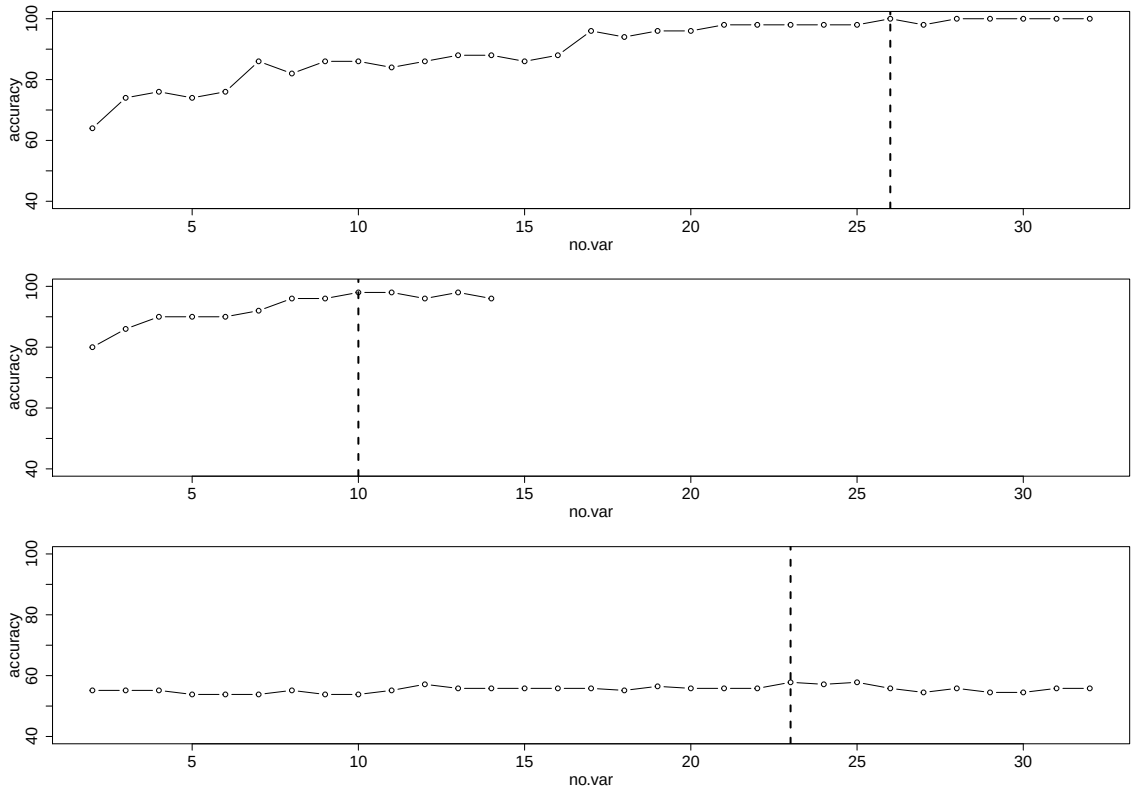


Figure B.13: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the ionosphere data with $n = 50$ and $d = 2$.

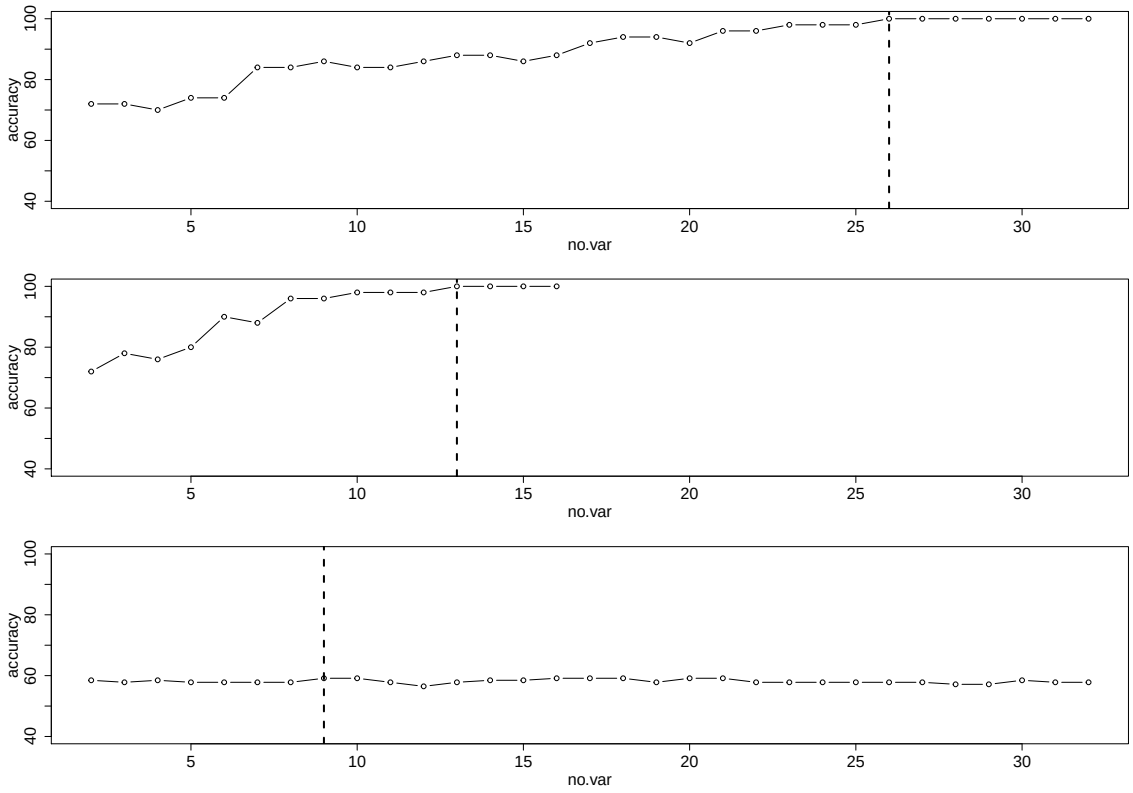


Figure B.14: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the ionosphere data with $n = 50$ and $d = 5$.

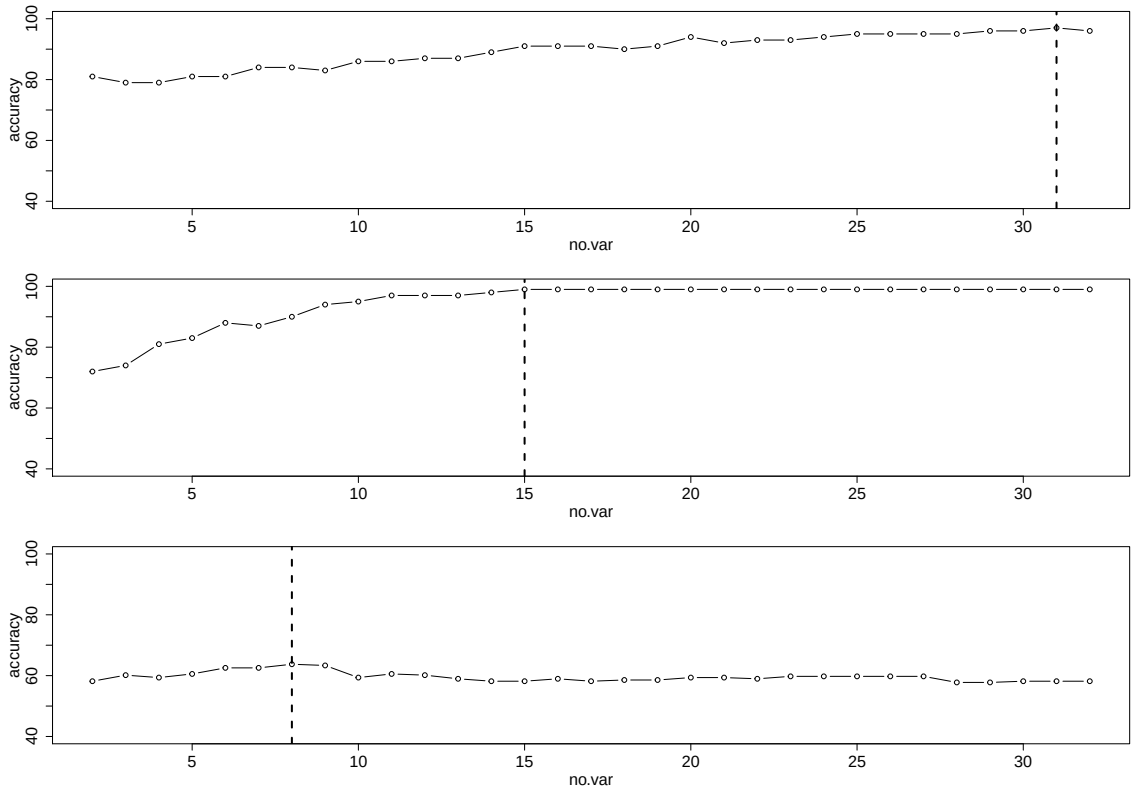


Figure B.15: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the ionosphere data with $n = 100$ and $d = 2$.

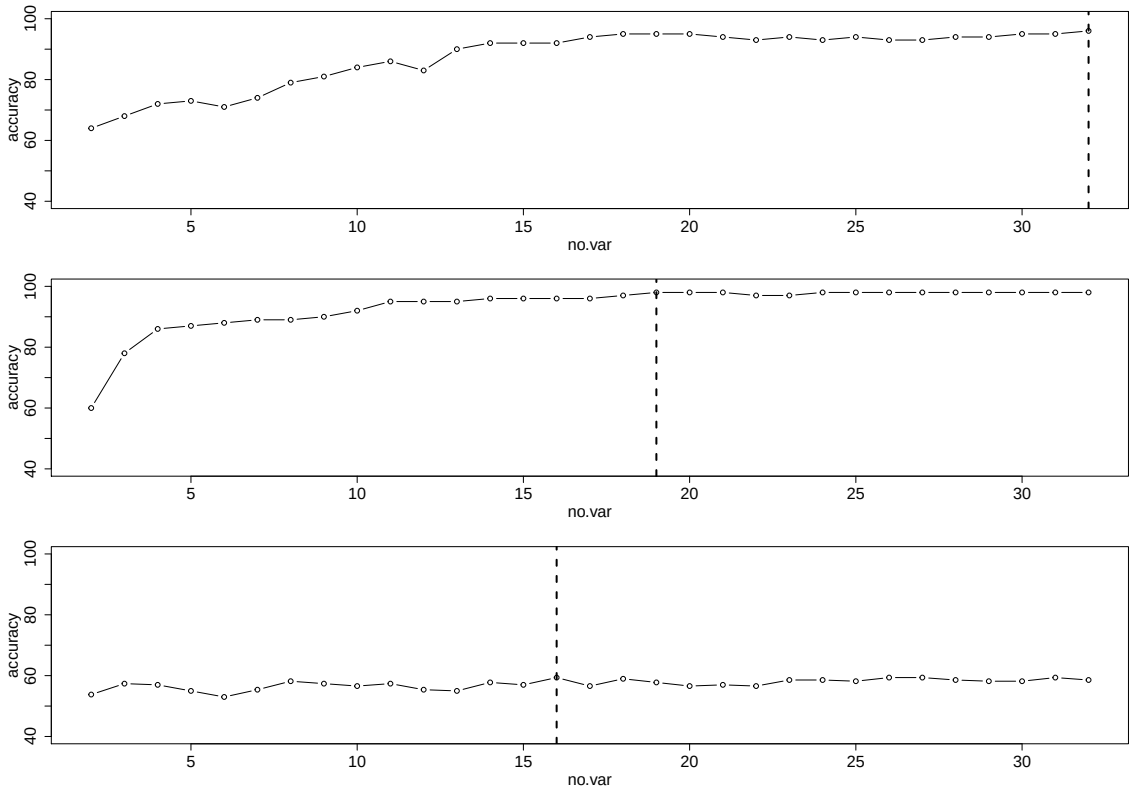


Figure B.16: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the ionosphere data with $n = 100$ and $d = 5$.

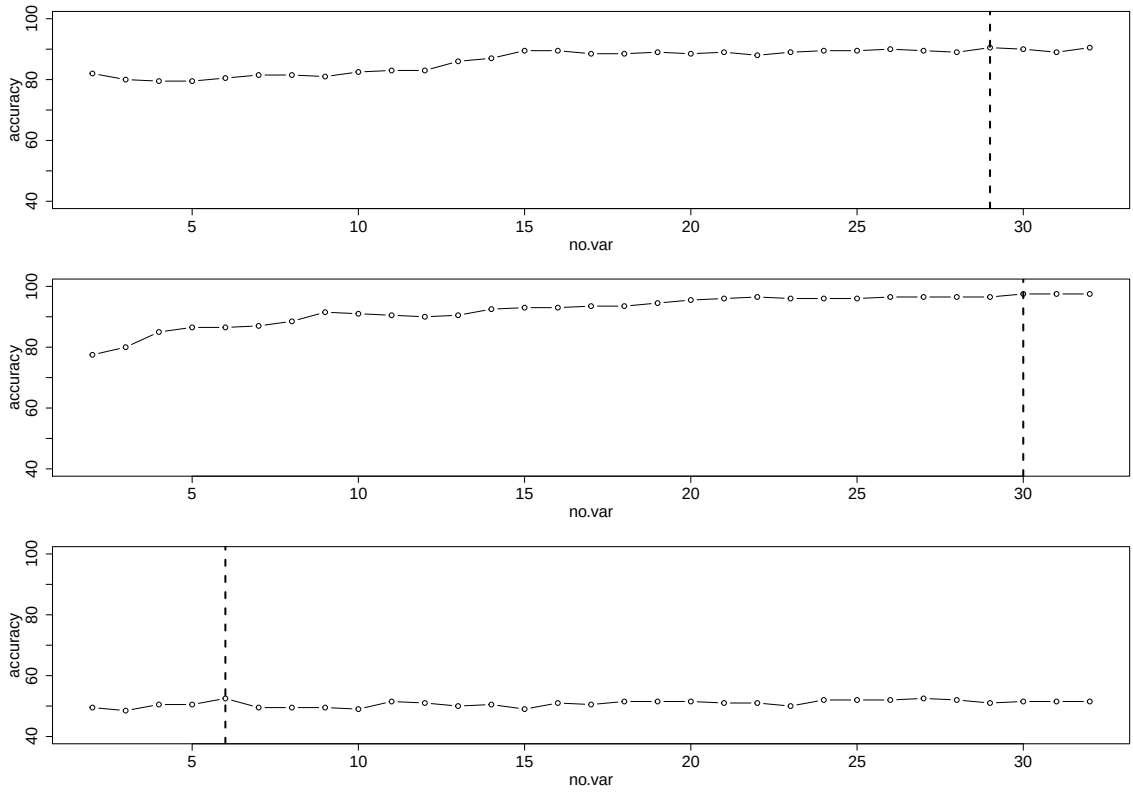


Figure B.17: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the ionosphere data with $n = 200$ and $d = 2$.

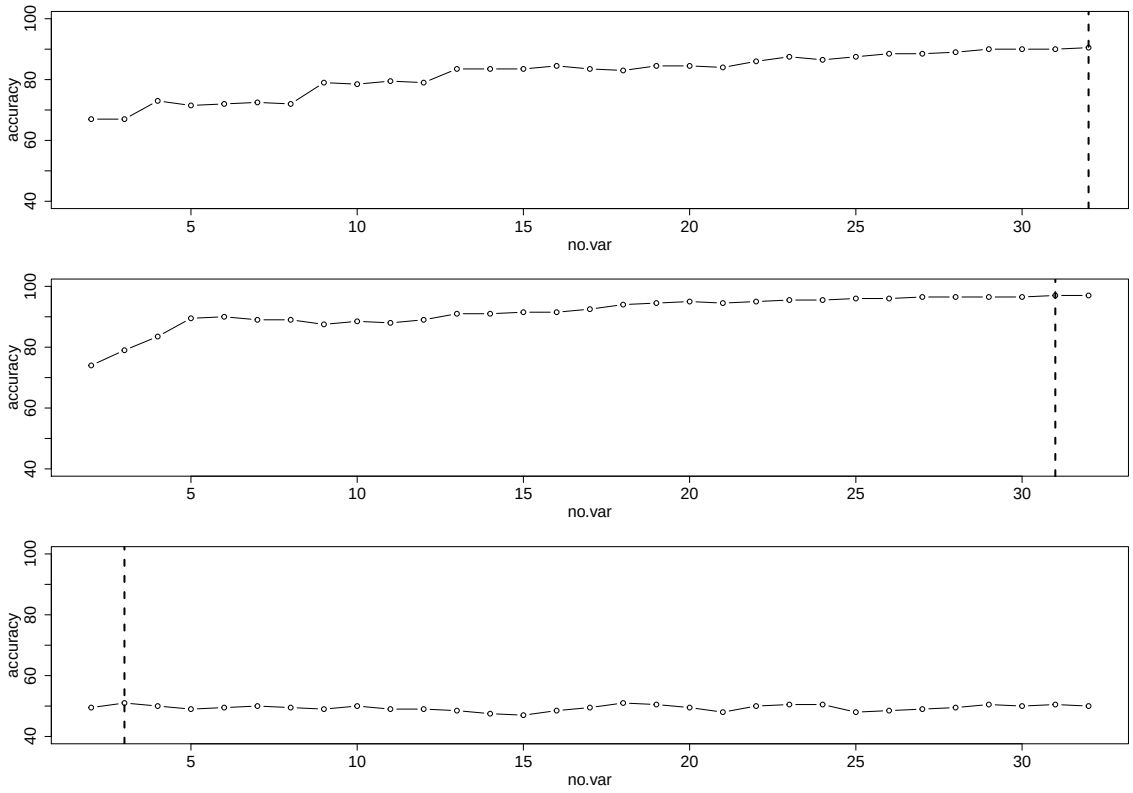


Figure B.18: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the ionosphere data with $n = 200$ and $d = 5$.

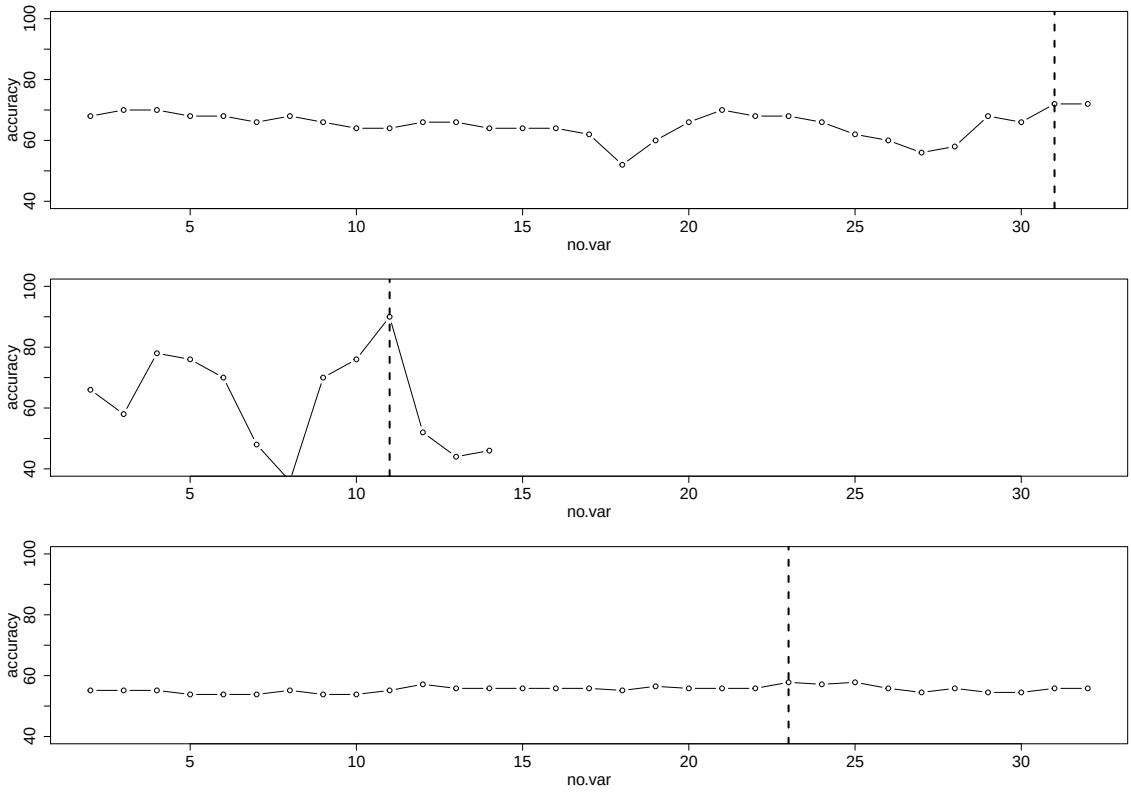


Figure B.19: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the ionosphere data with $n = 50$ and $d = 2$.

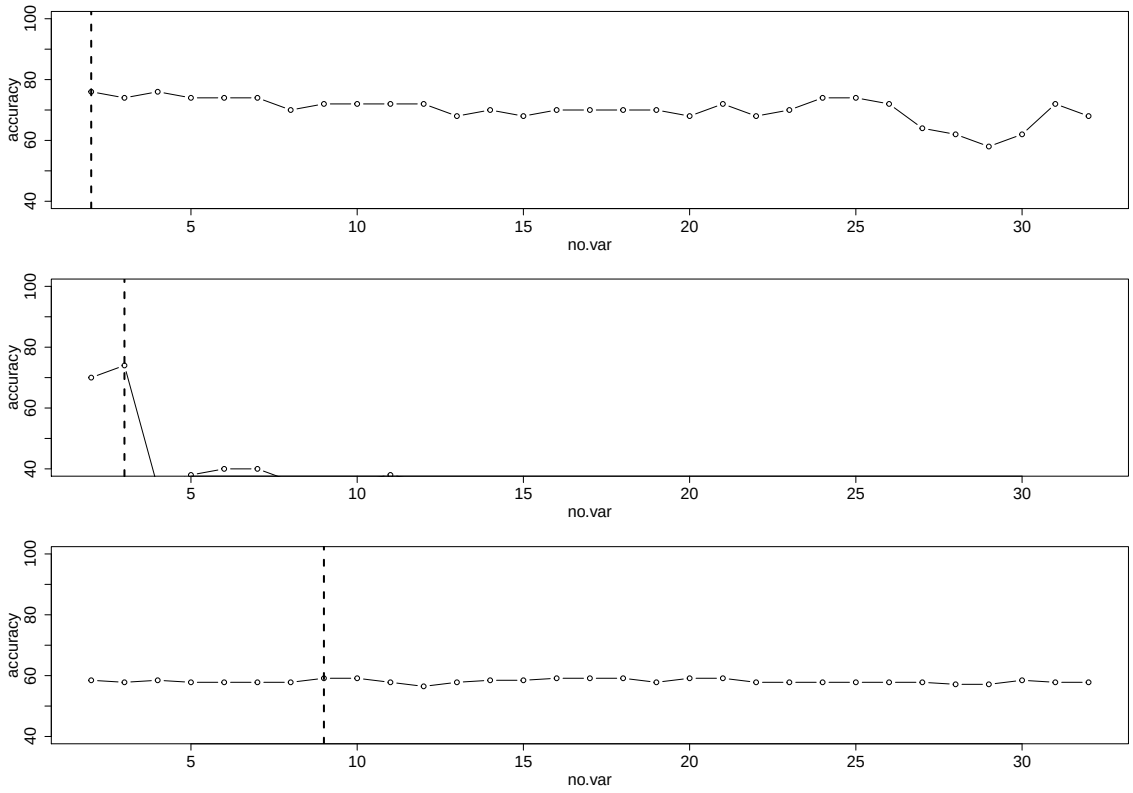


Figure B.20: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the ionosphere data with $n = 50$ and $d = 5$.

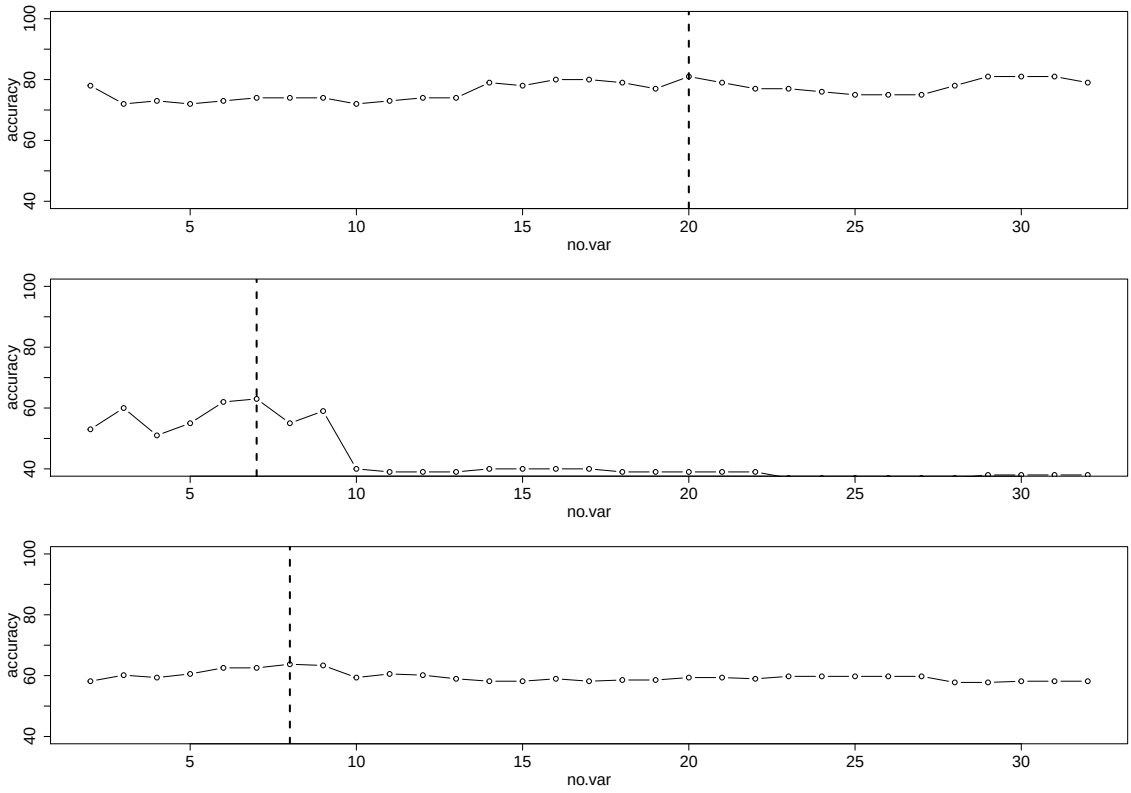


Figure B.21: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the ionosphere data with $n = 100$ and $d = 2$.

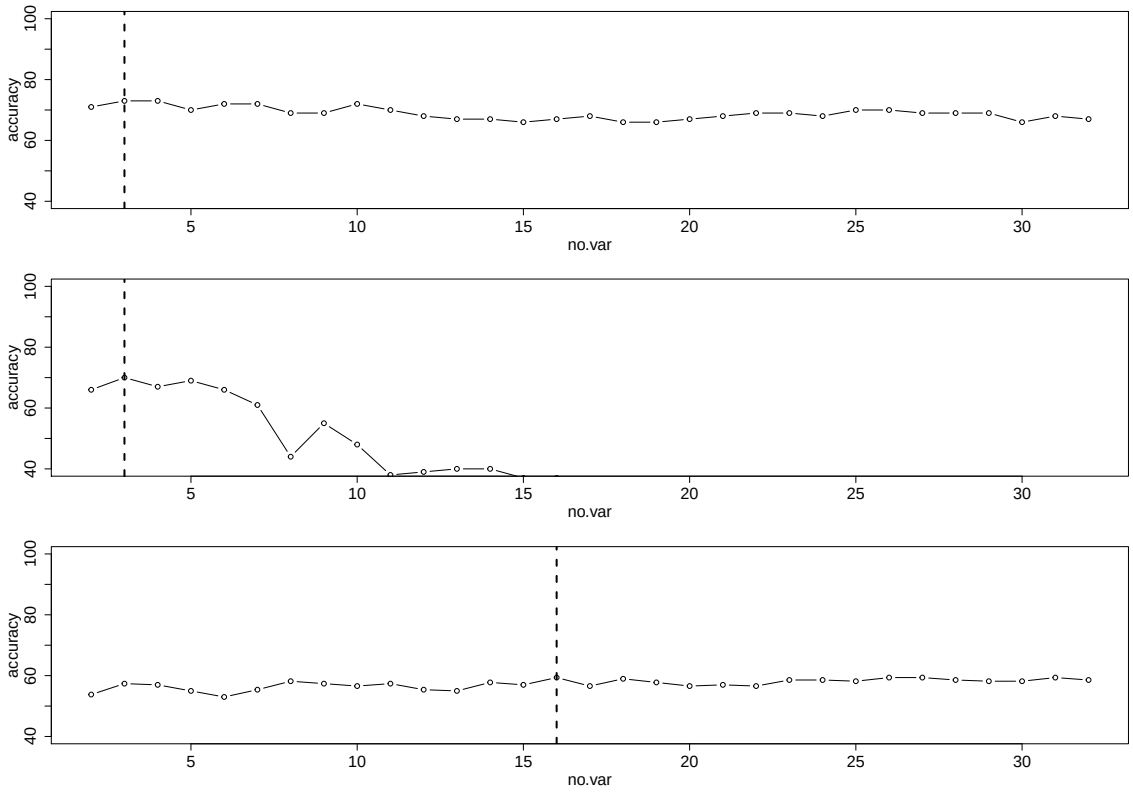


Figure B.22: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the ionosphere data with $n = 100$ and $d = 5$.

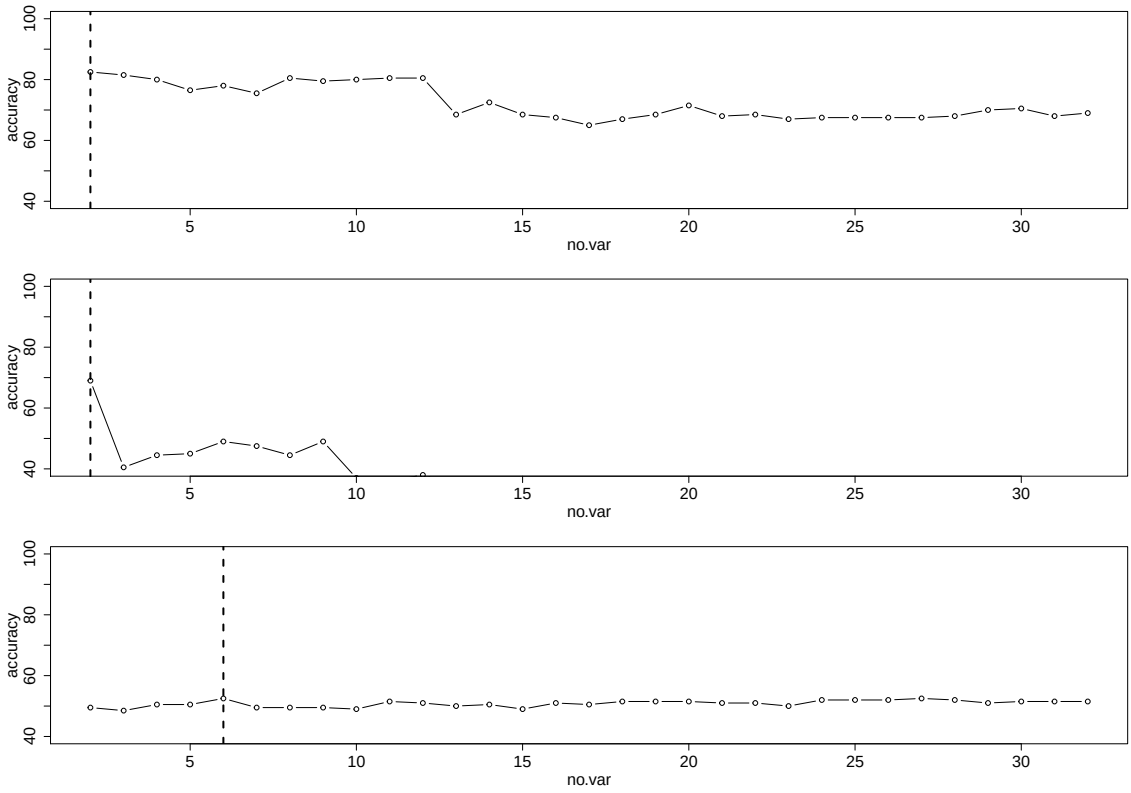


Figure B.23: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the ionosphere data with $n = 200$ and $d = 2$.

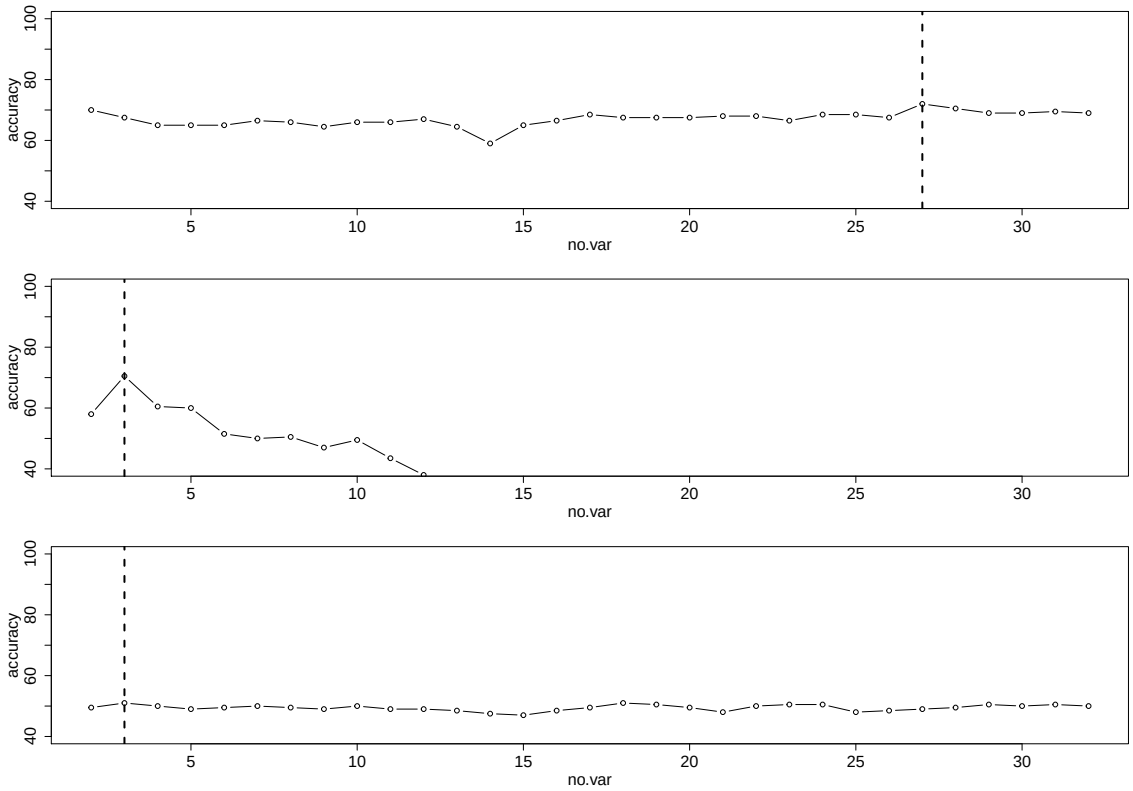


Figure B.24: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the ionosphere data with $n = 200$ *and* $d = 5$.

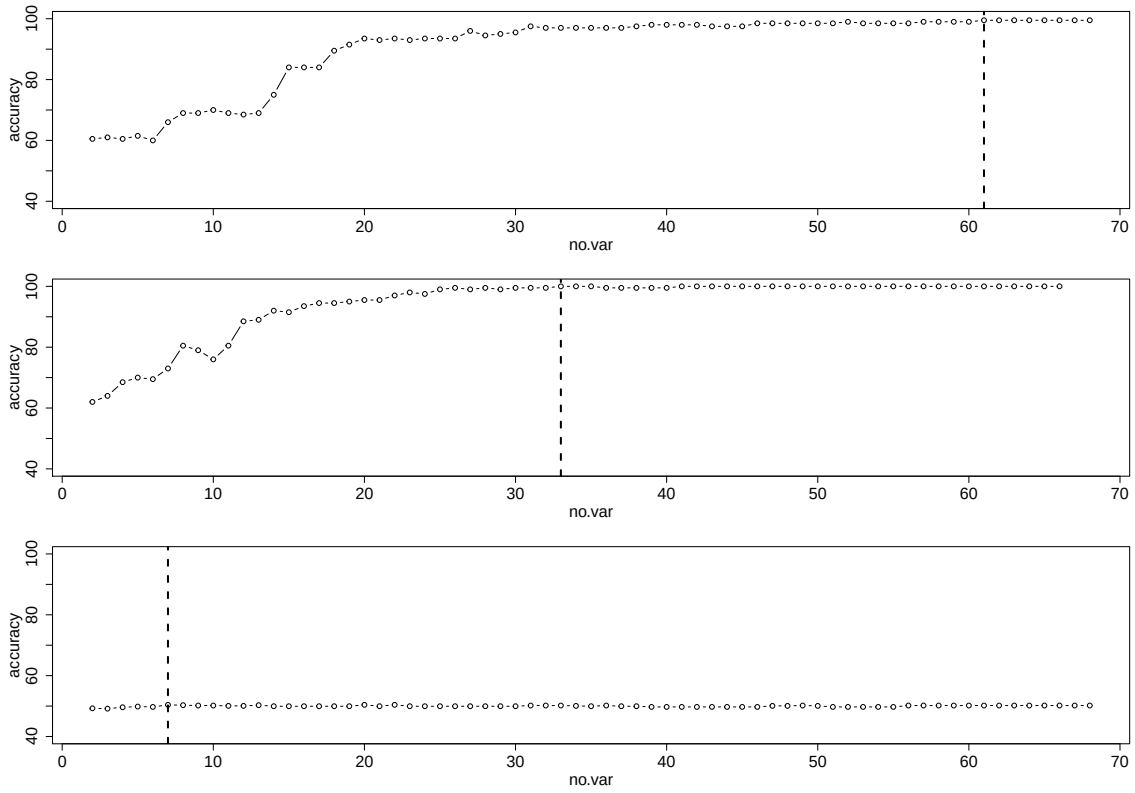


Figure B.25: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the mice data with $n = 200$ *and* $d = 2$.

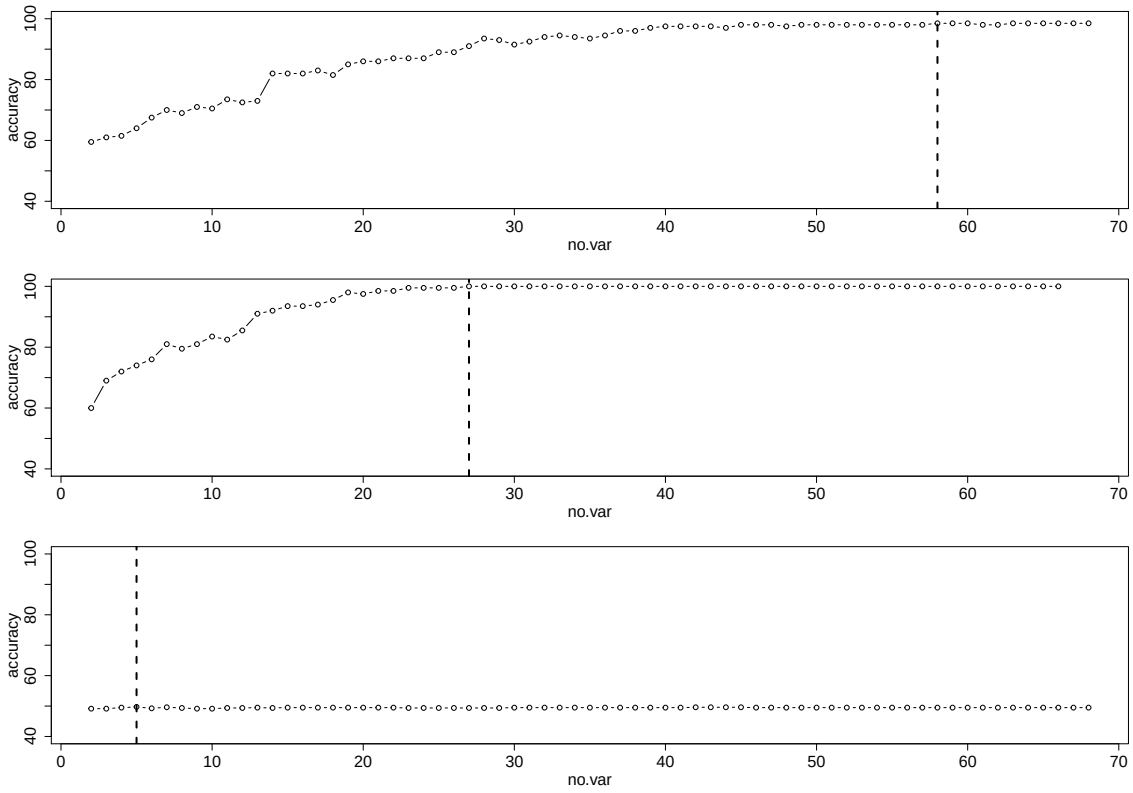


Figure B.26: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the mice data with $n = 200$ and $d = 5$.

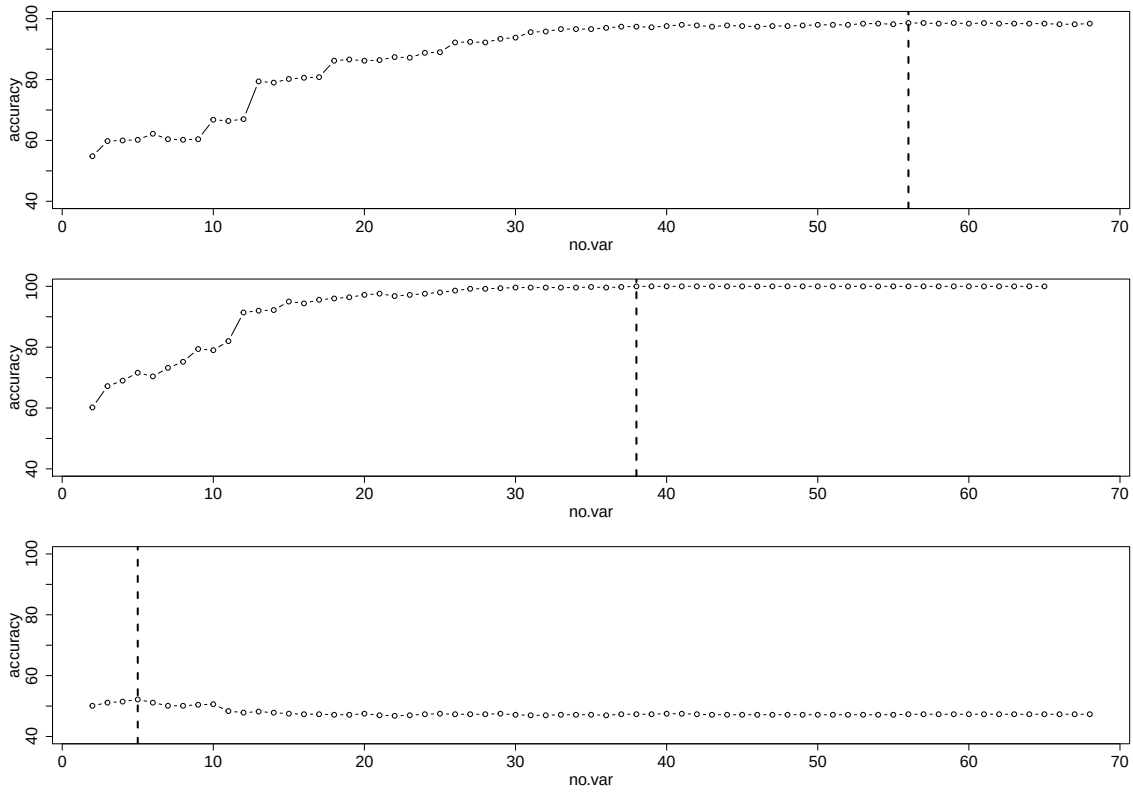


Figure B.27: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the mice data with $n = 500$ *and* $d = 5$.

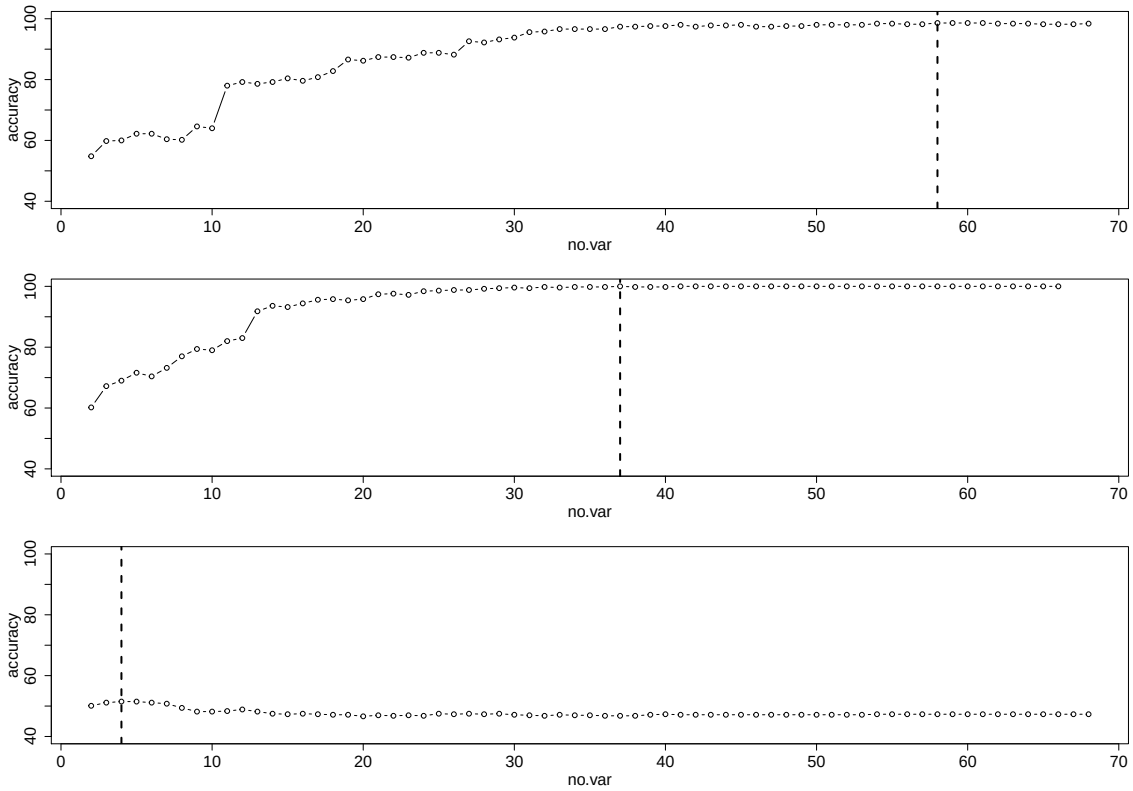


Figure B.28: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the mice data with $n = 500$ and $d = 5$.

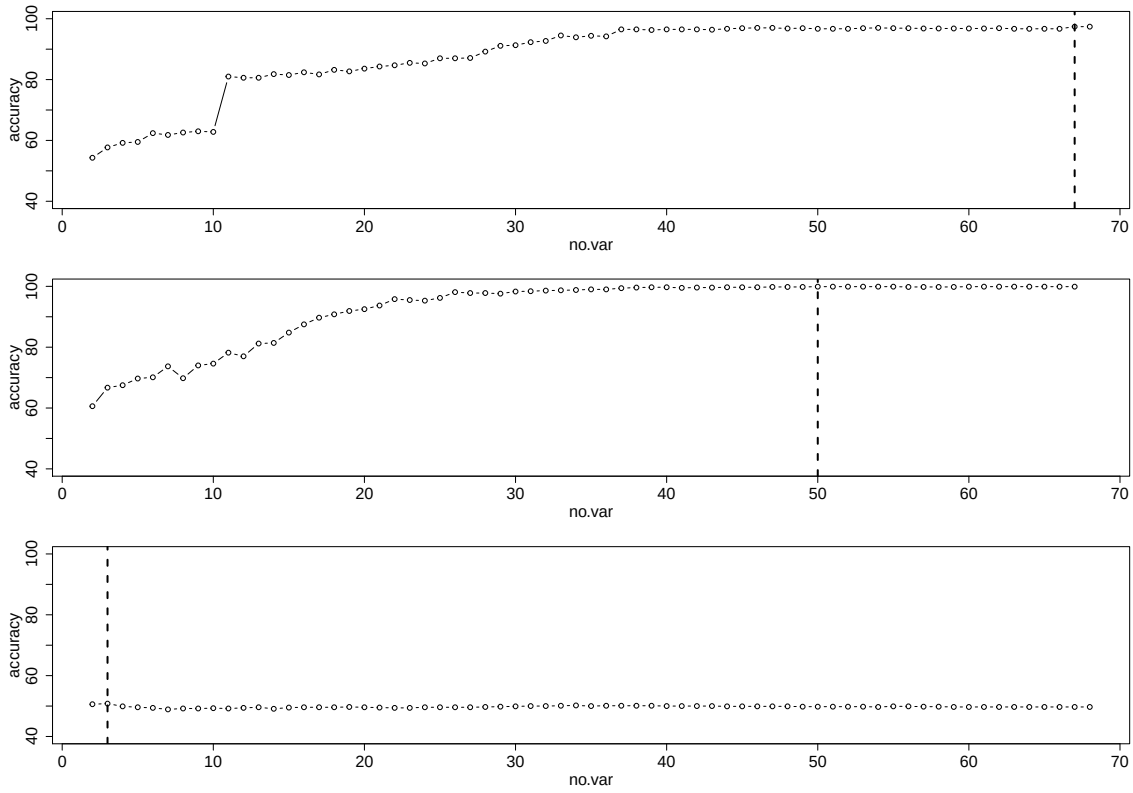


Figure B.29: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the mice data with $n = 1000$ *and* $d = 2$.

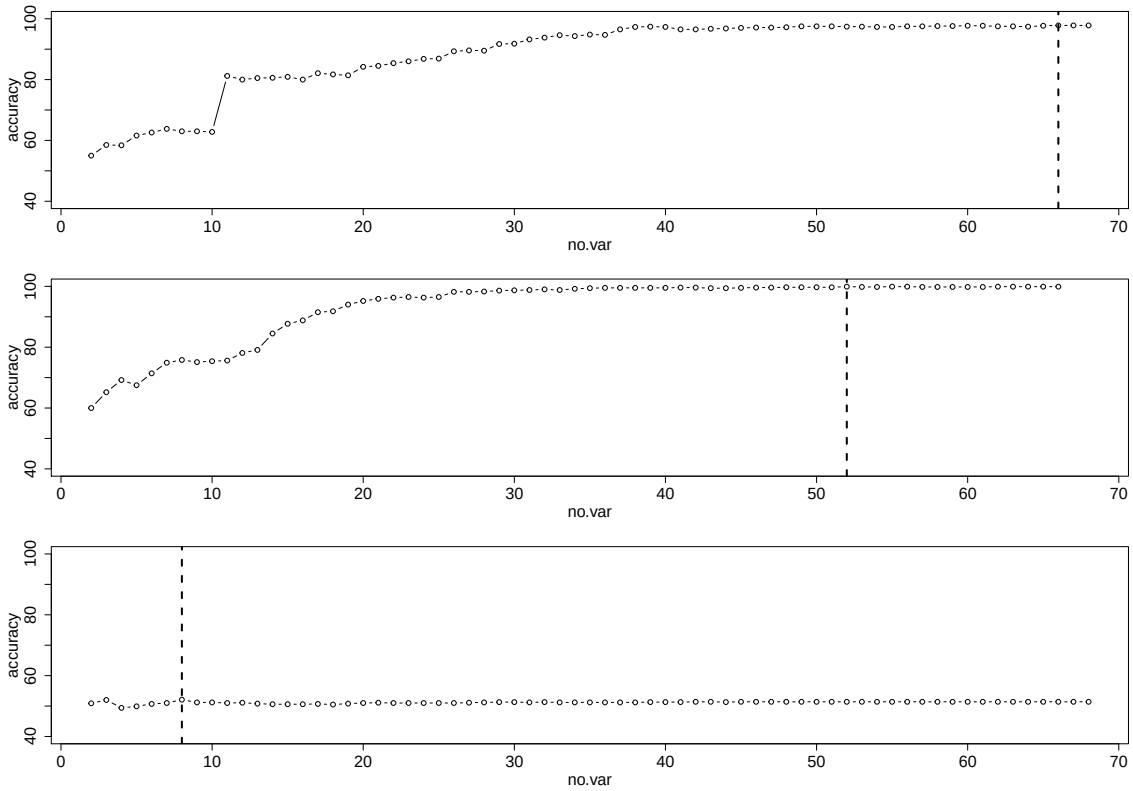


Figure B.30: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the mice data with $n = 1000$ *and* $d = 5$.

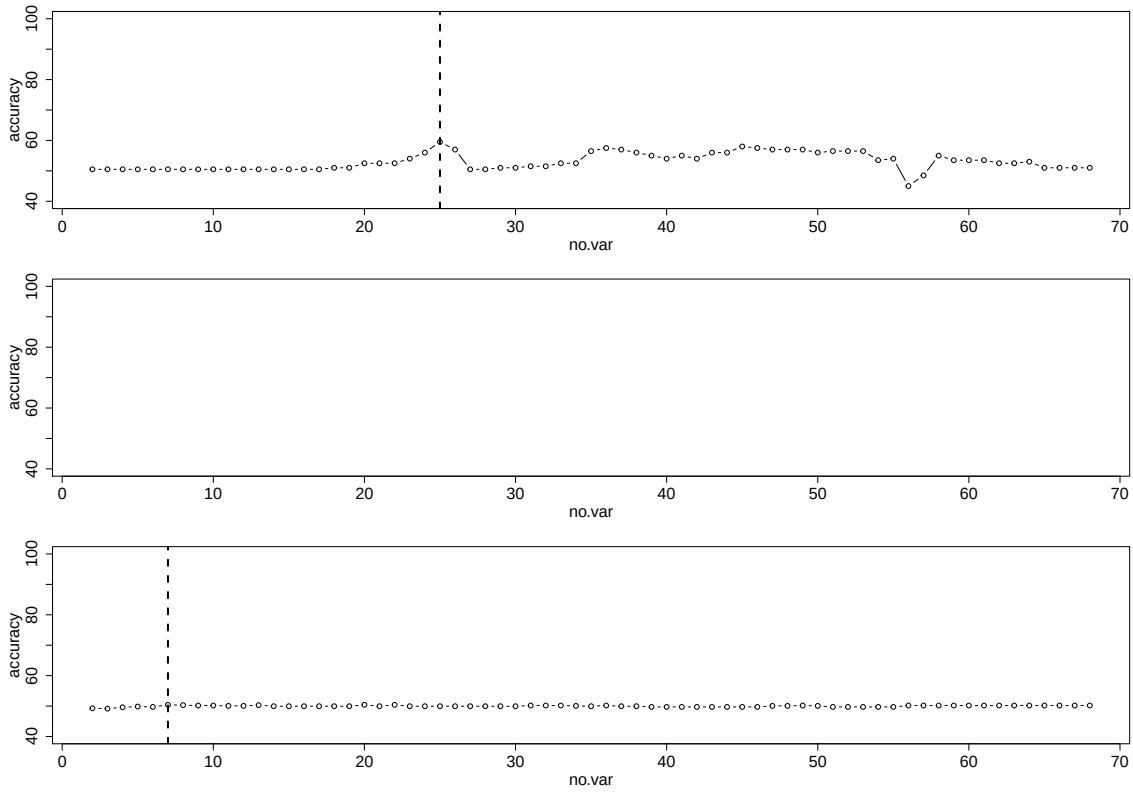


Figure B.31: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the mice data with $n = 200$ and $d = 2$.

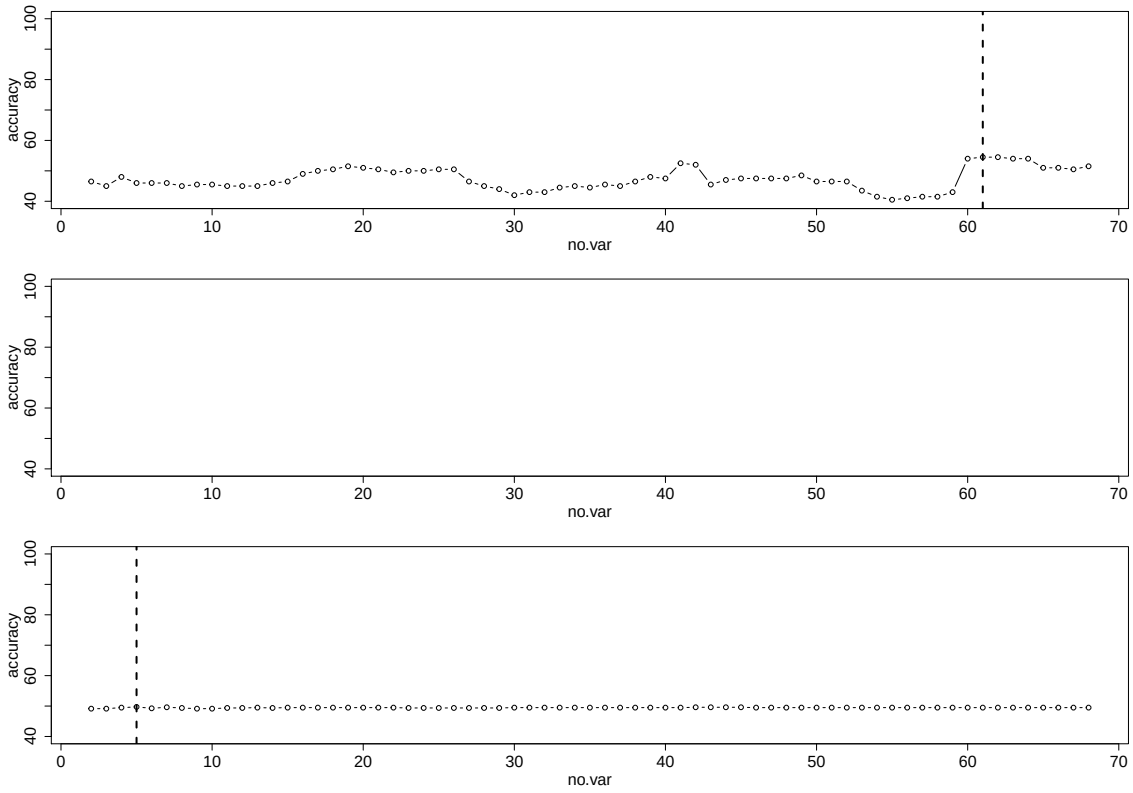


Figure B.32: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the mice data with $n = 200$ and $d = 5$.

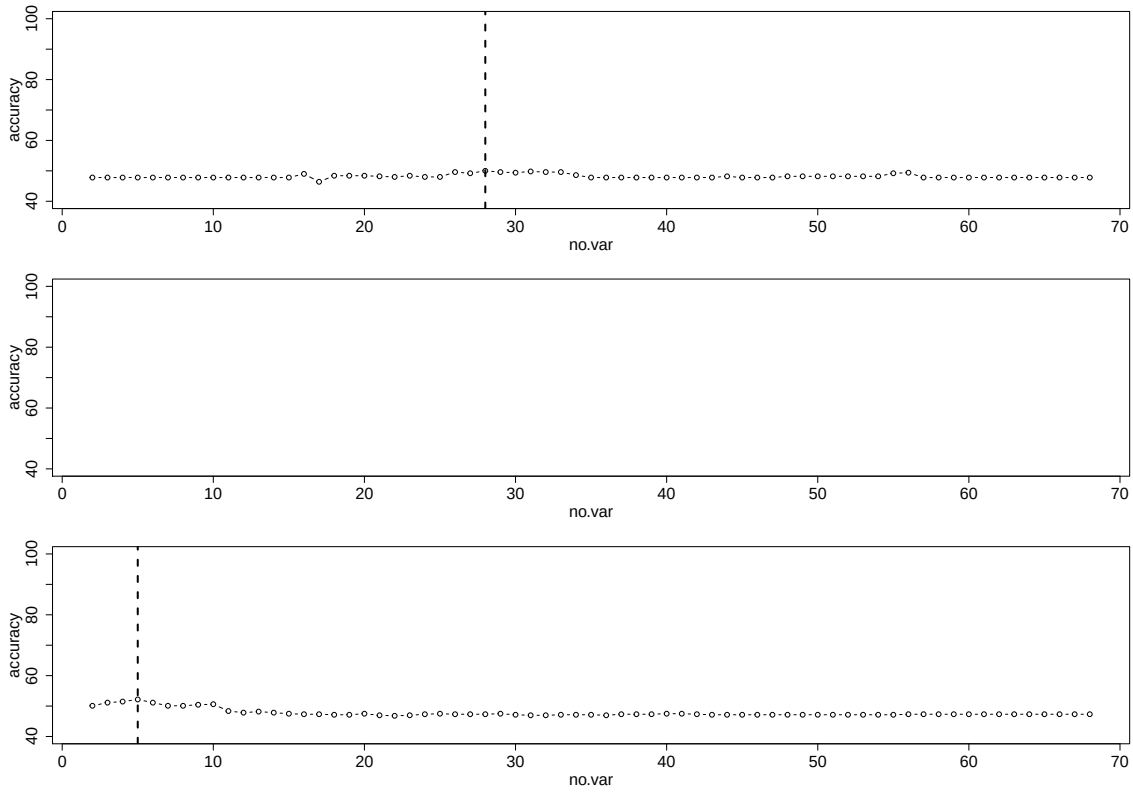


Figure B.33: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the mice data with $n = 500$ and $d = 5$.

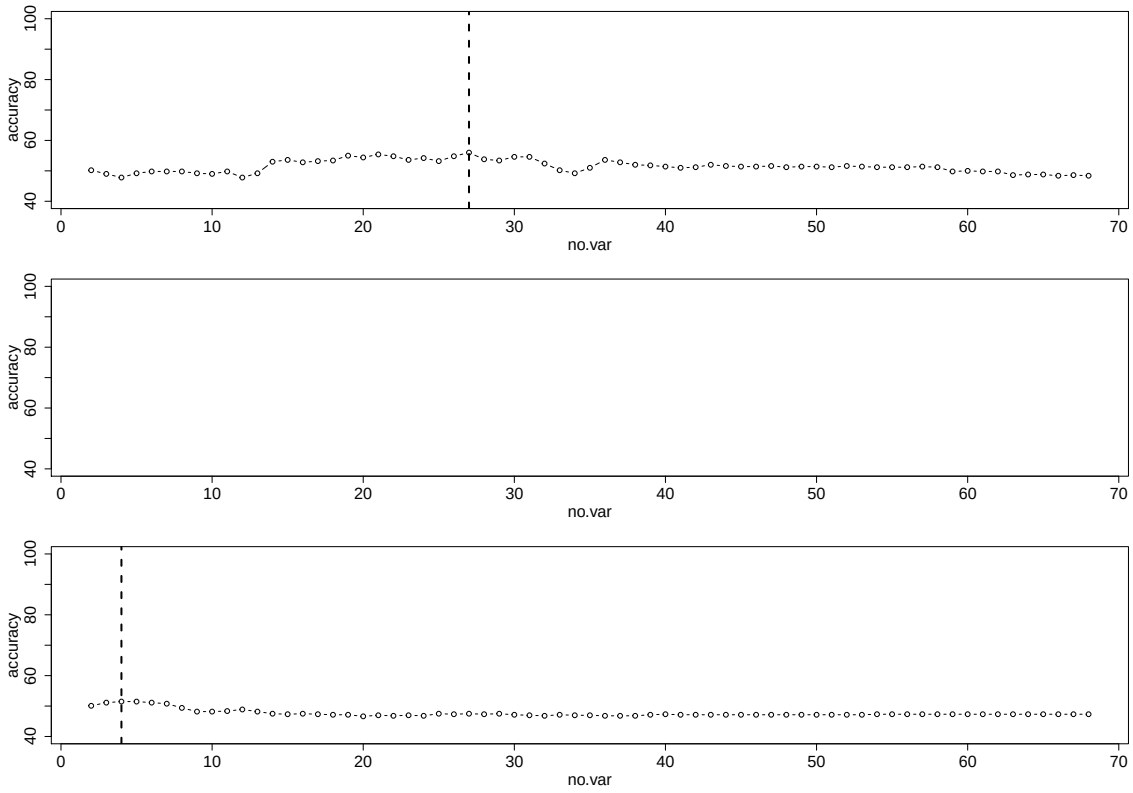


Figure B.34: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the mice data with $n = 500$ and $d = 5$.

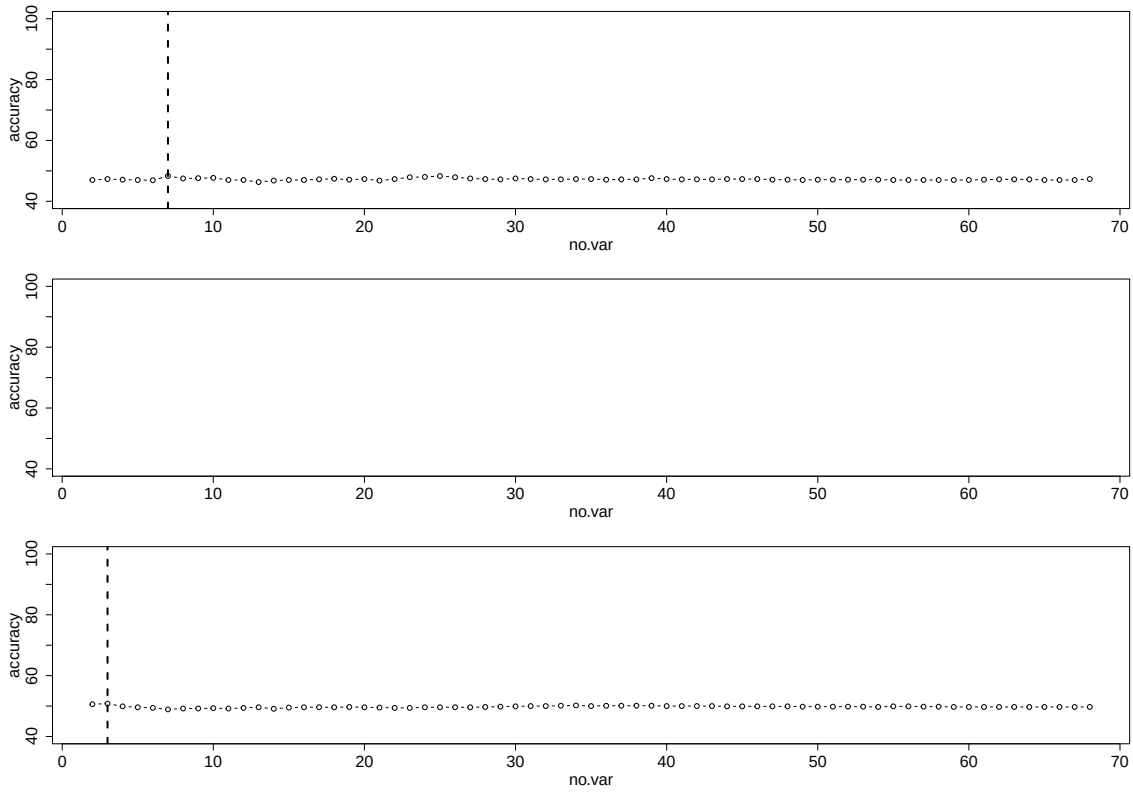


Figure B.35: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the mice data with $n = 1000$ *and* $d = 2$.

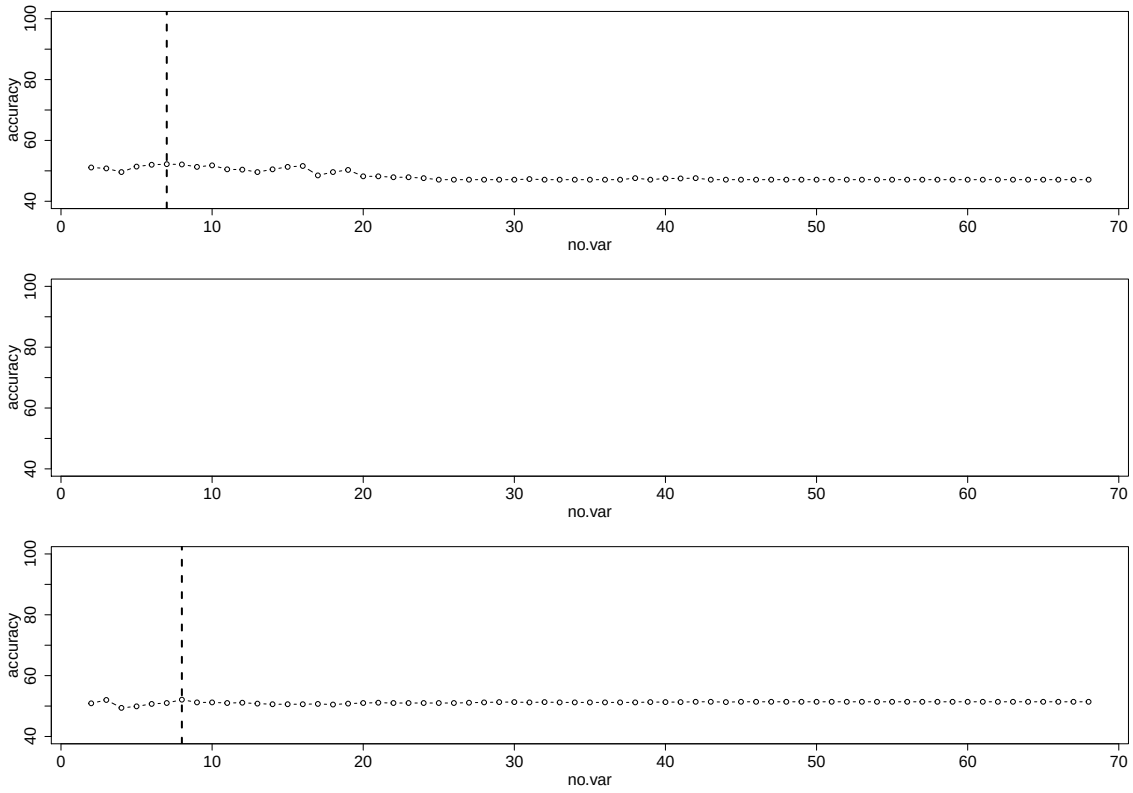


Figure B.36: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the mice data with $n = 1000$ *and* $d = 5$.

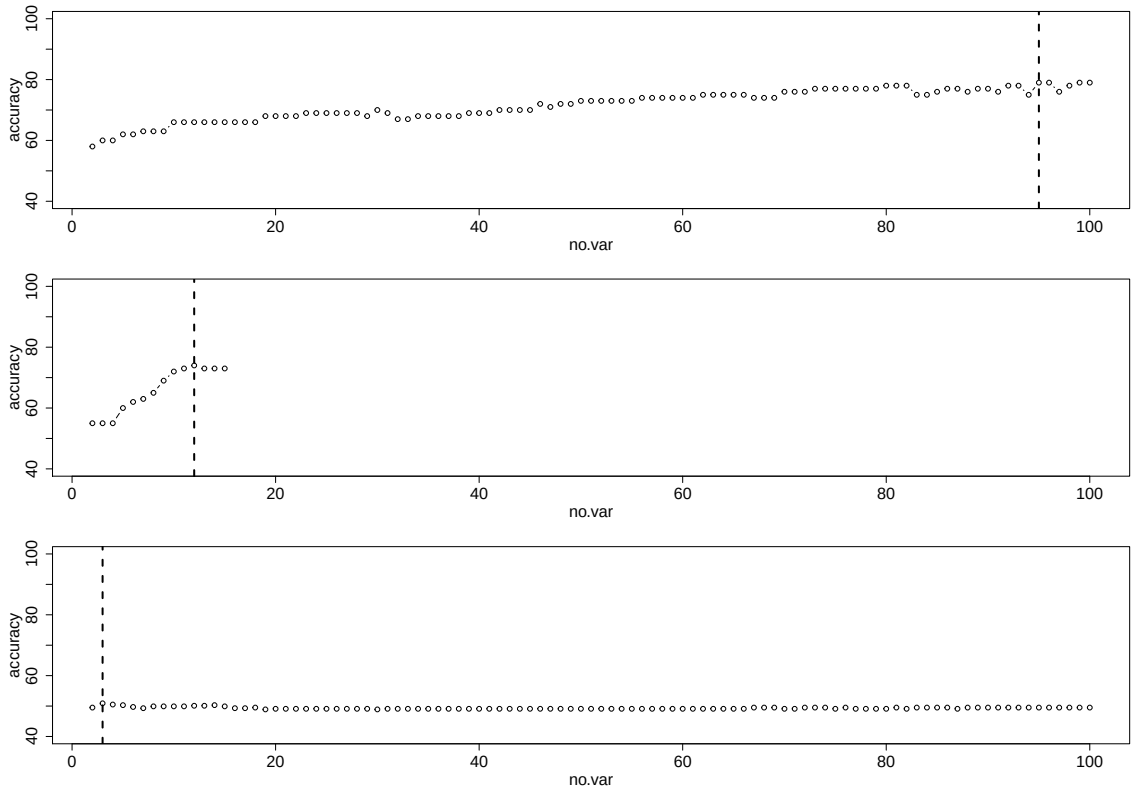


Figure B.37: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the hill-valley data with $n = 100$ *and* $d = 2$.

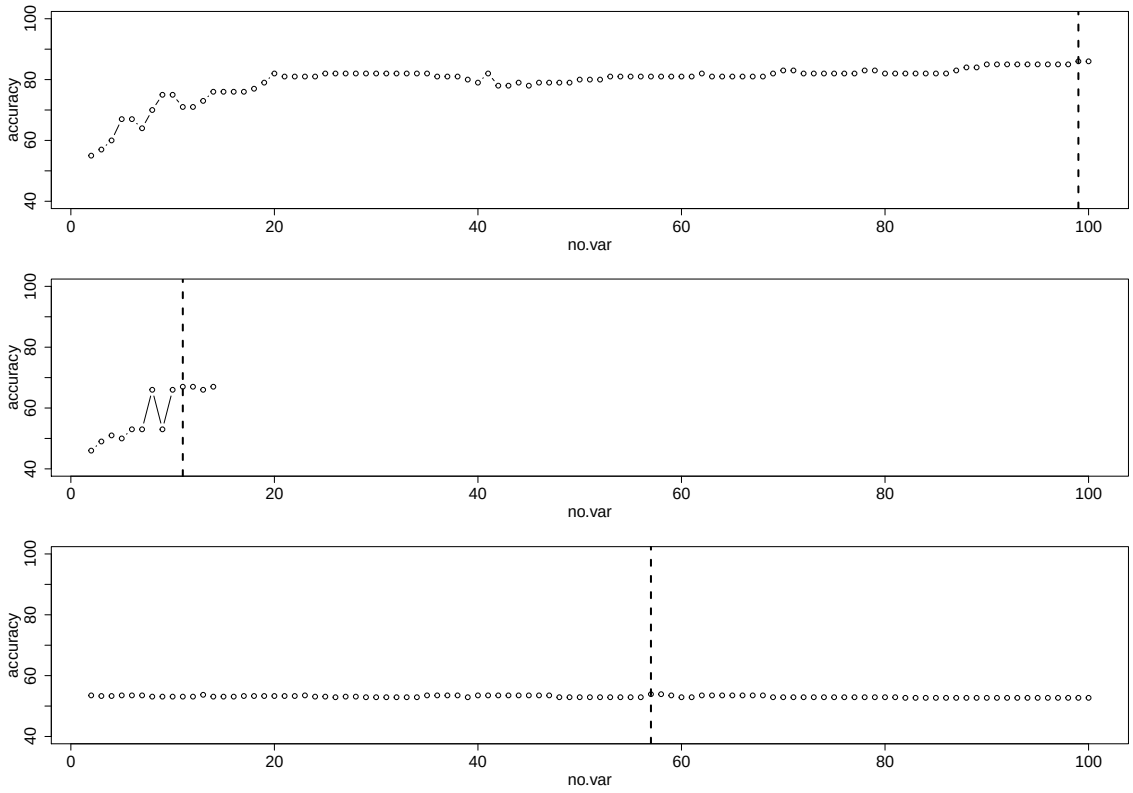


Figure B.38: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the hill-valley data with $n = 100$ and $d = 5$.

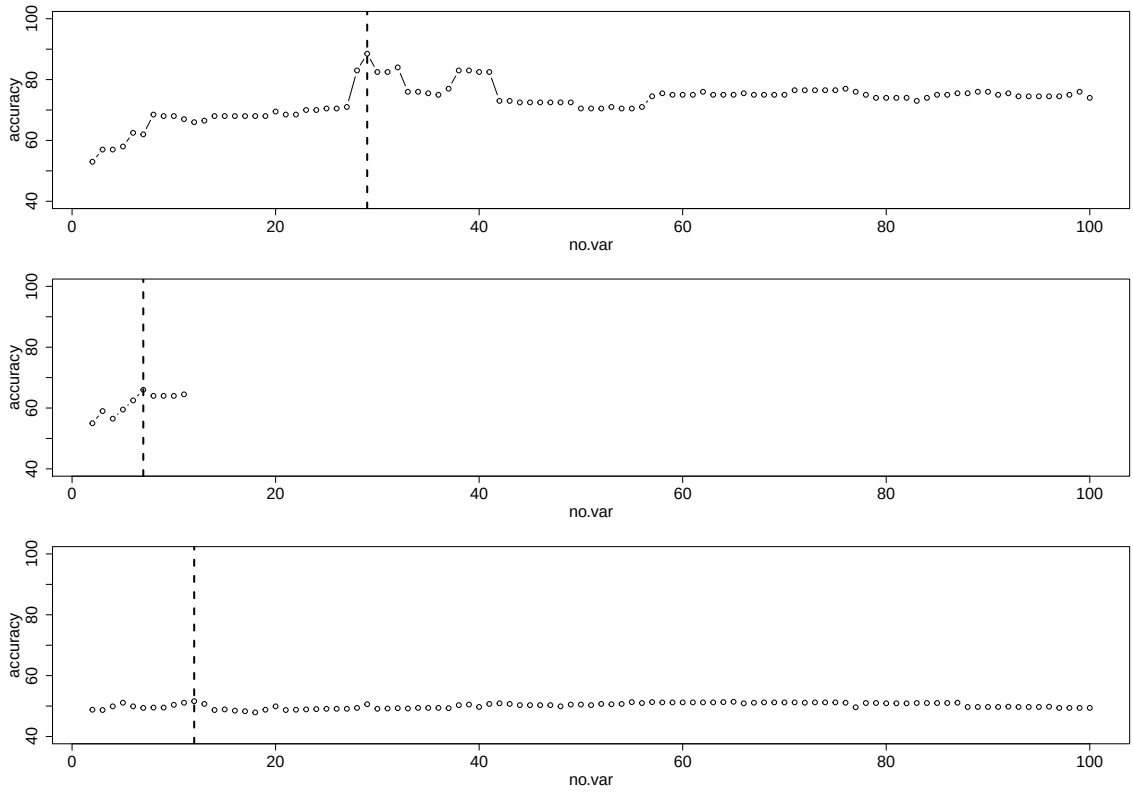


Figure B.39: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the hill-valley data with $n = 200$ *and* $d = 2$.

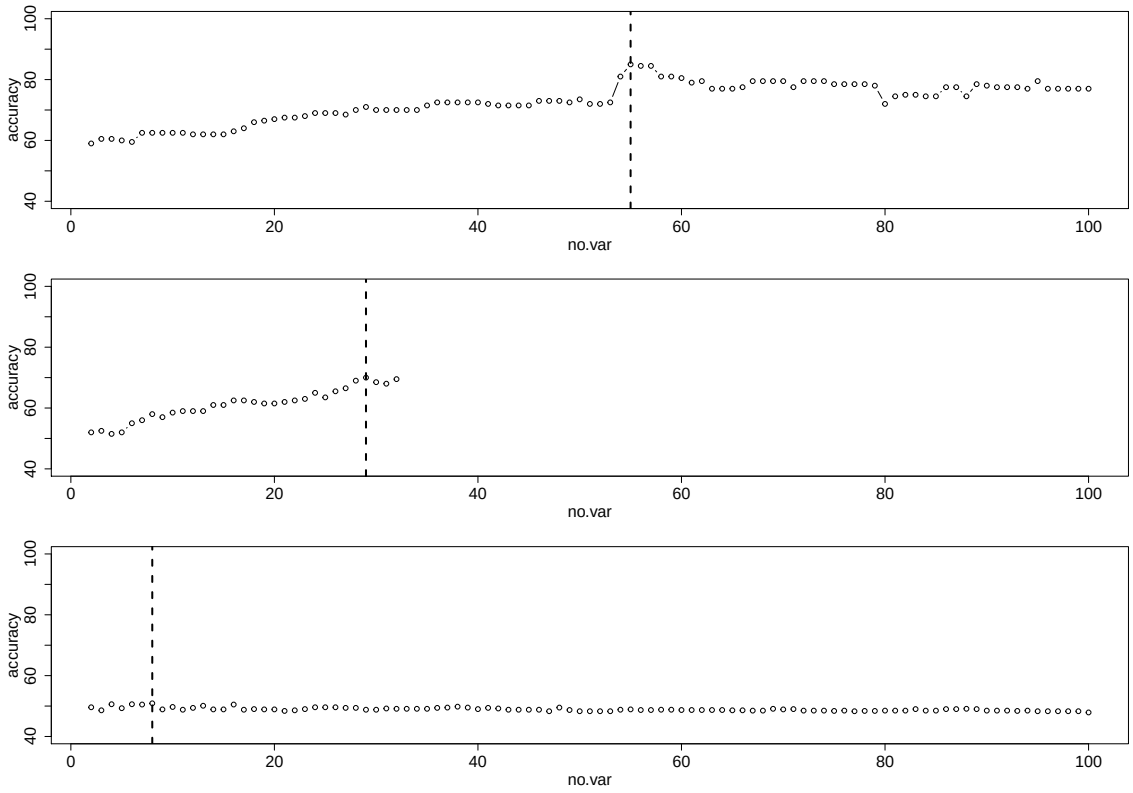


Figure B.40: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the hill-valley data with $n = 200$ and $d = 5$.

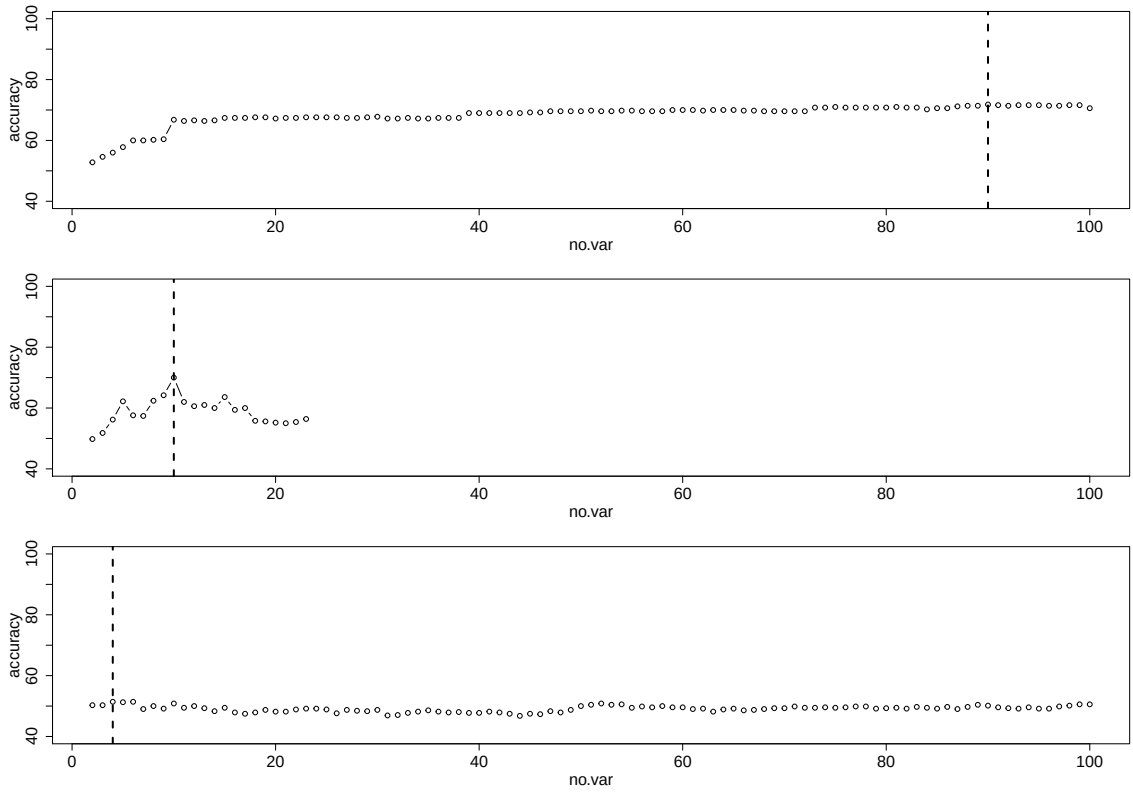


Figure B.41: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the hill-valley data with $n = 500$ and $d = 2$.

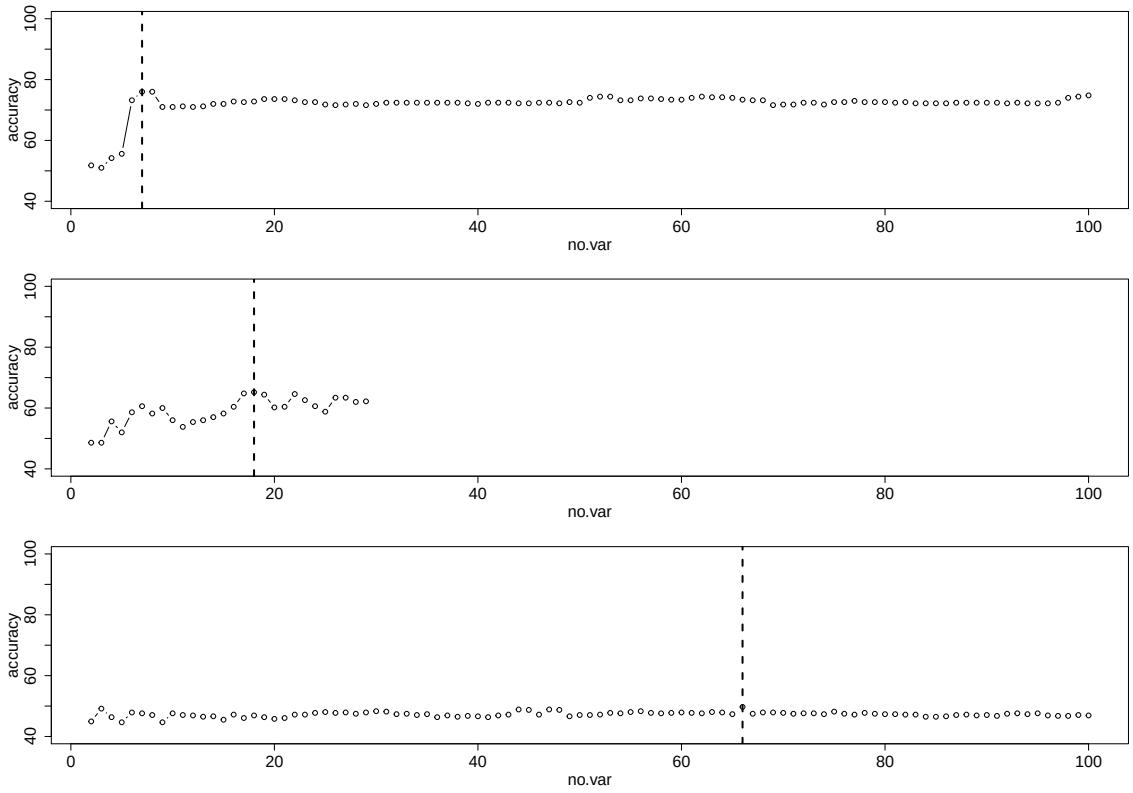


Figure B.42: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the hill-valley data with $n = 500$ and $d = 5$.

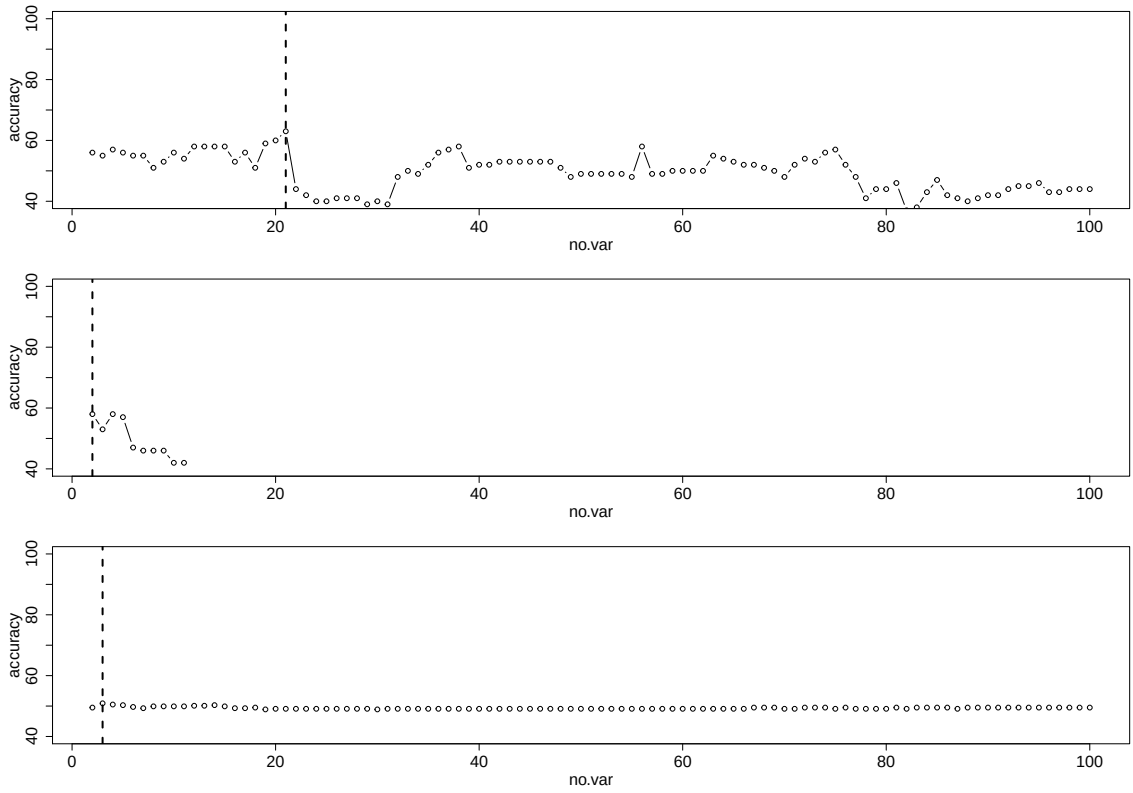


Figure B.43: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the hill-valley data with $n = 100$ and $d = 2$.

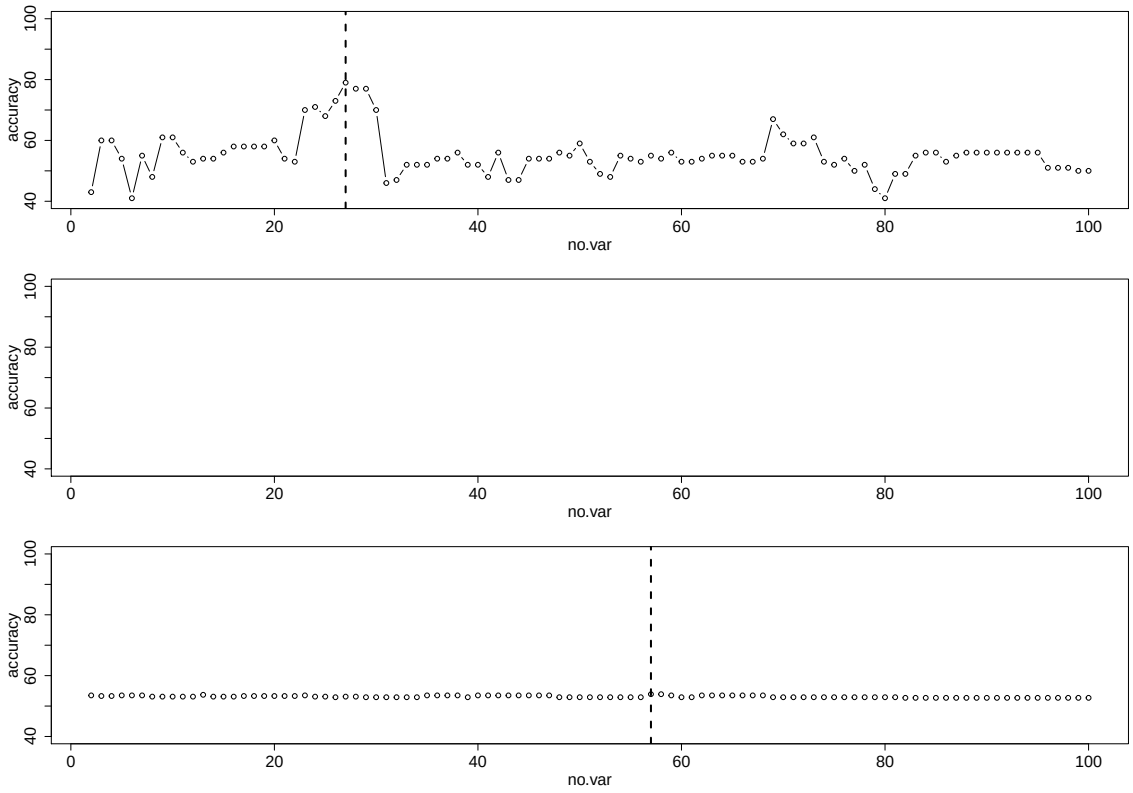


Figure B.44: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the hill-valley data with $n = 100$ and $d = 5$.

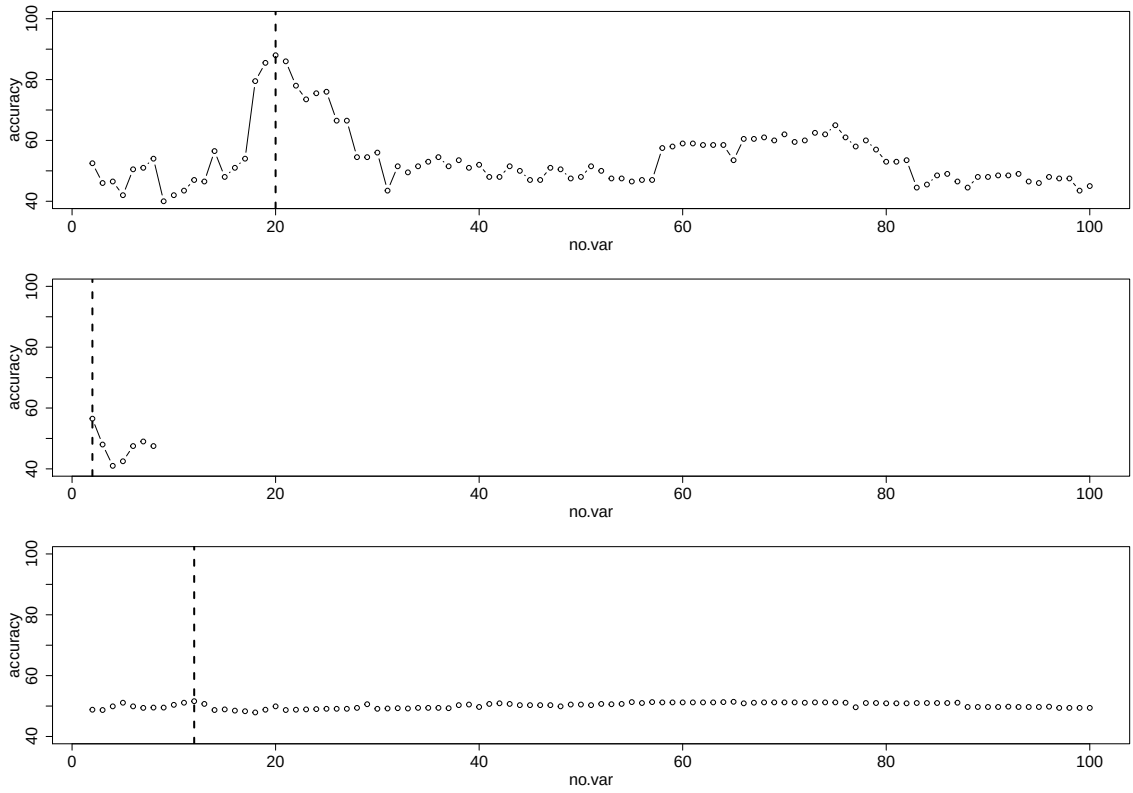


Figure B.45: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the hill-valley data with $n = 200$ and $d = 2$.

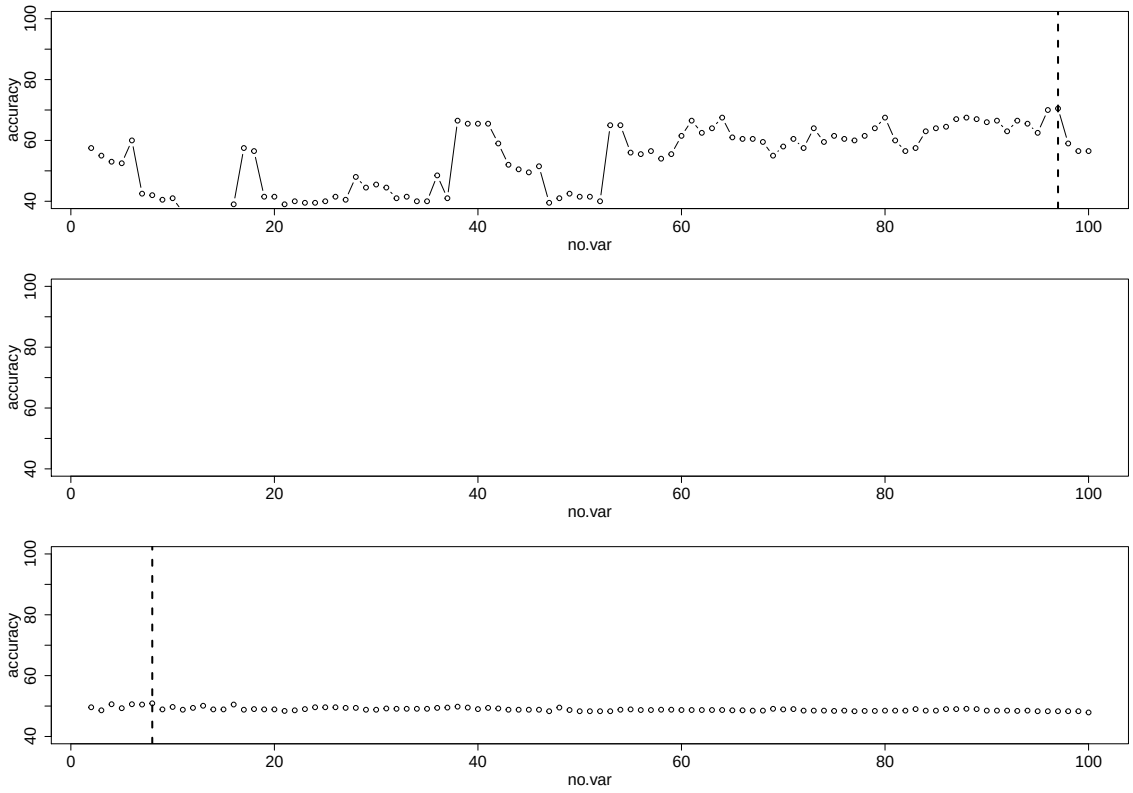


Figure B.46: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the hill-valley data with $n = 200$ and $d = 5$.

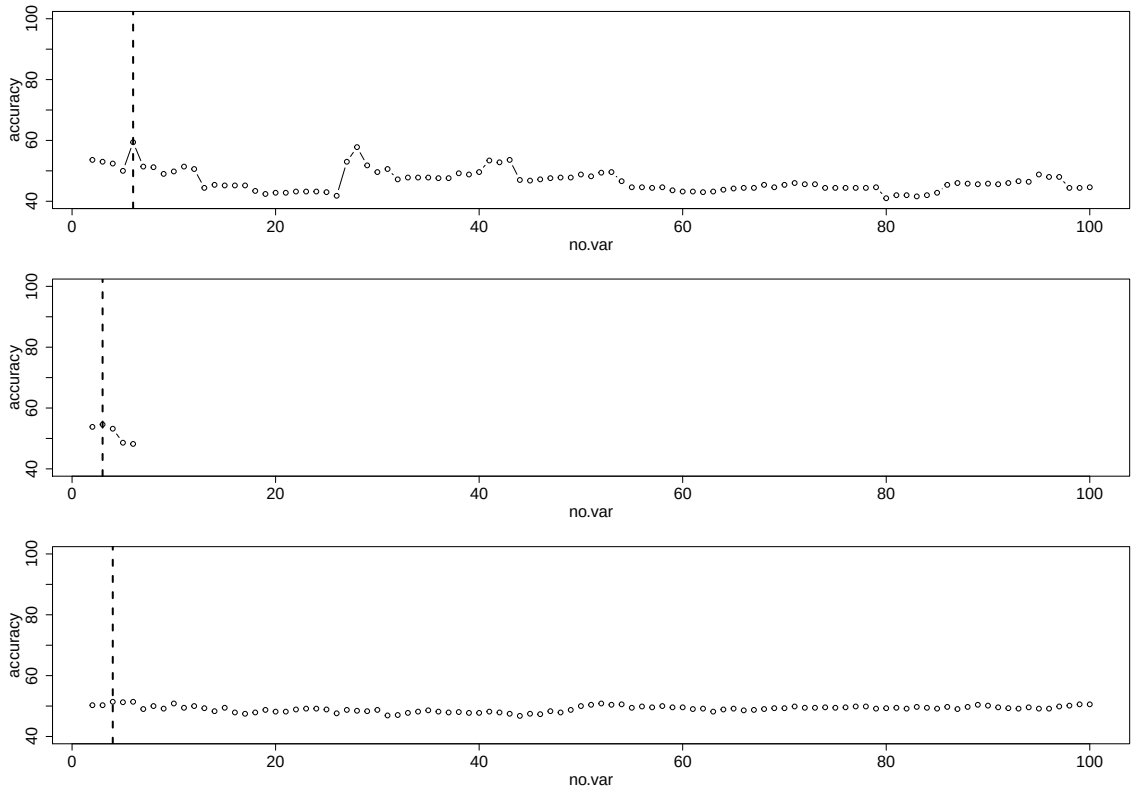


Figure B.47: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the hill-valley data with $n = 500$ and $d = 2$.

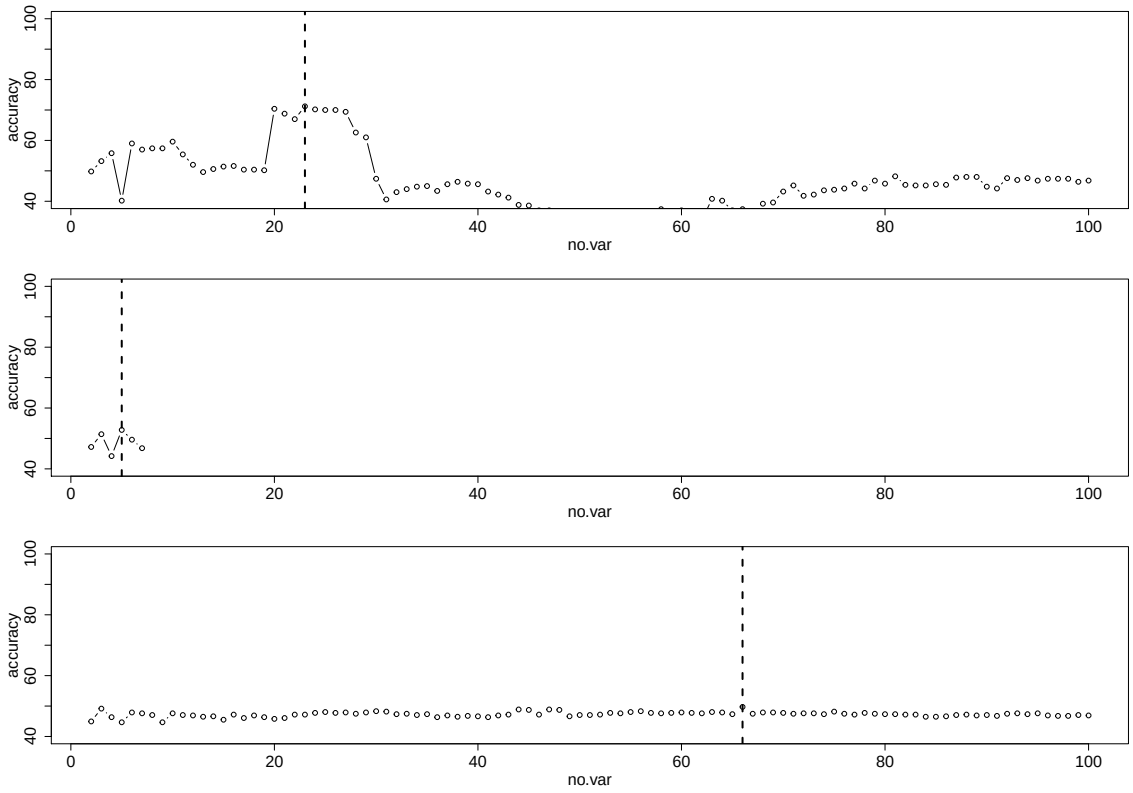


Figure B.48: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the hill-valley data with $n = 500$ and $d = 5$.

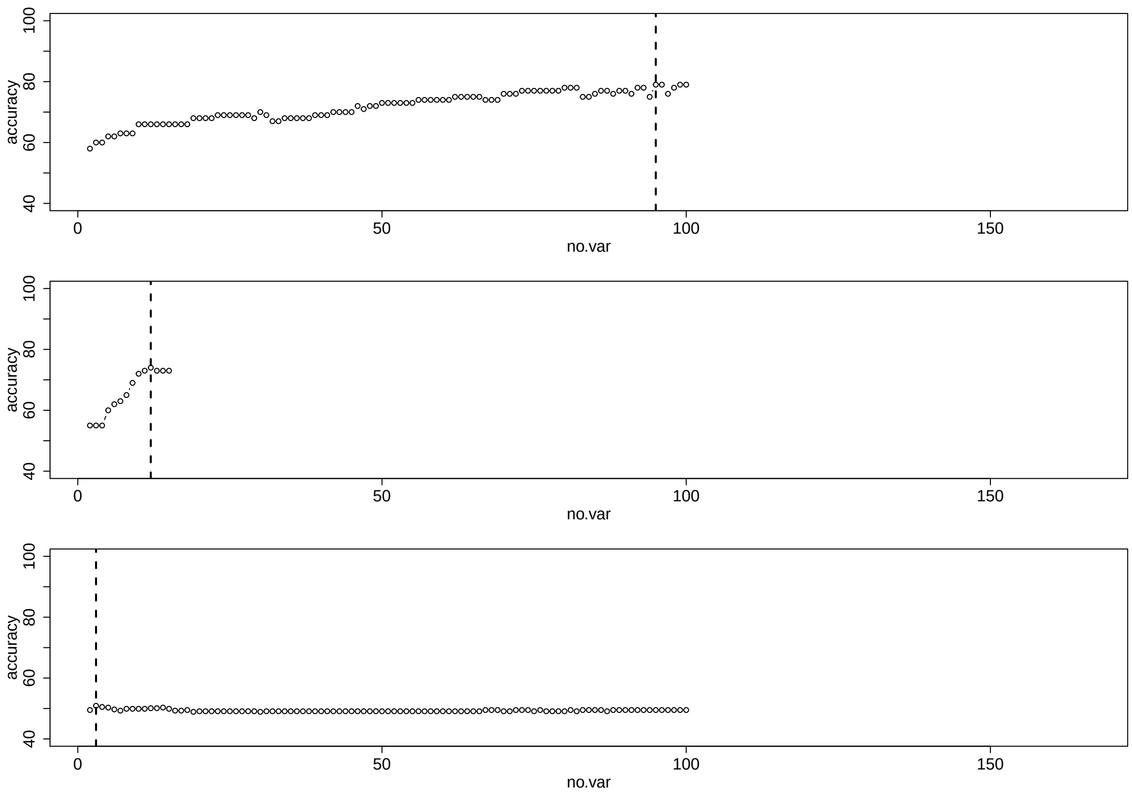


Figure B.49: Accuracy rate (————) and optimal number of variables, h, (- - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the musk data with $n = 100$ and $d = 5$.

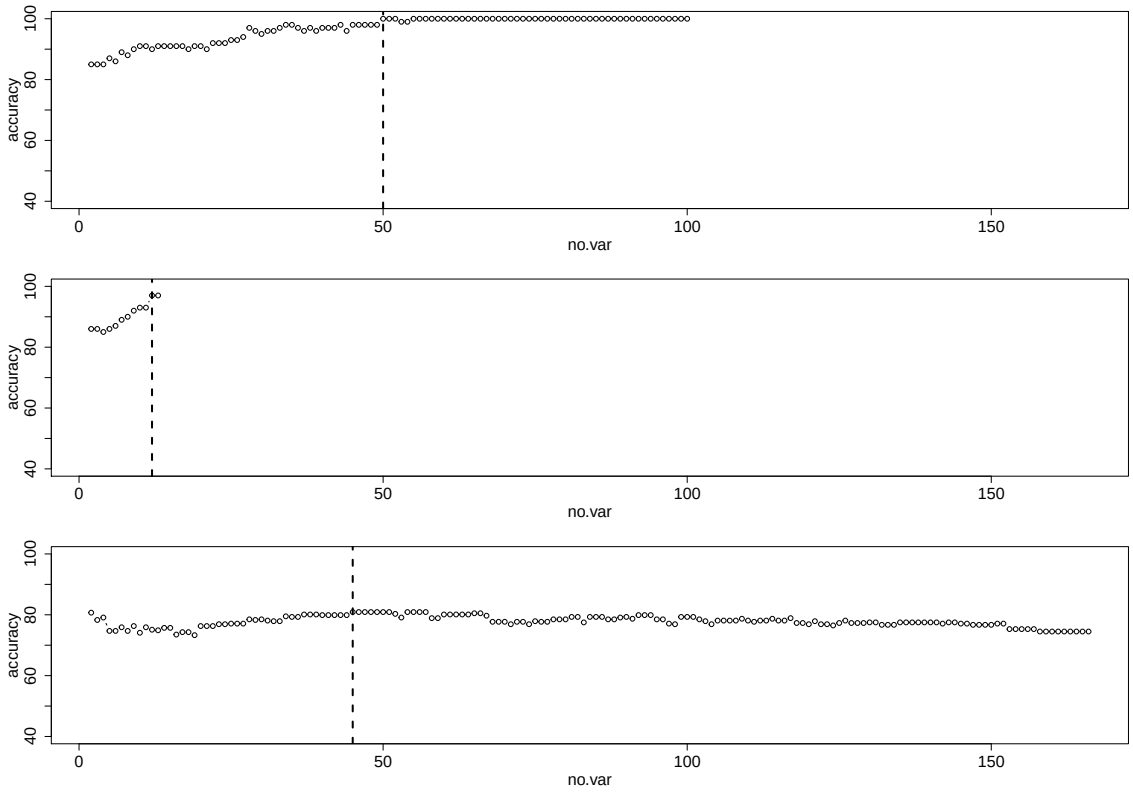


Figure B.50: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the musk data with $n = 100$ and $d = 5$.

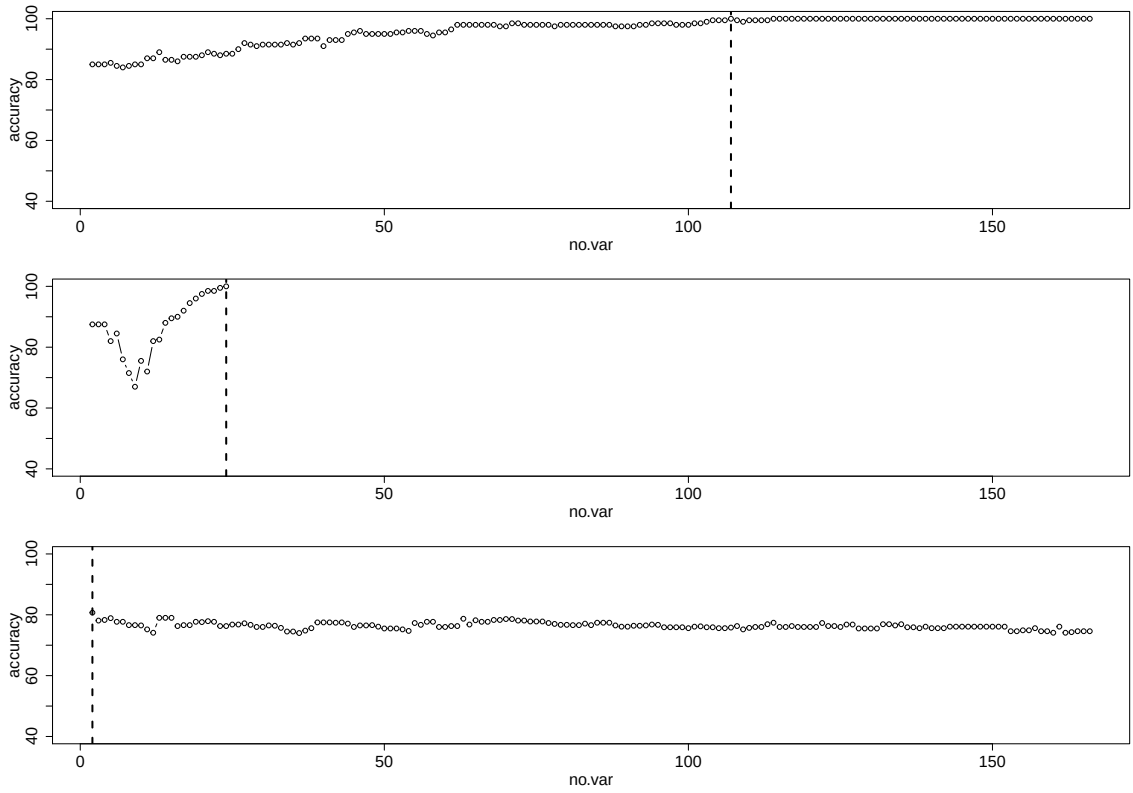


Figure B.51: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the musk data with $n = 200$ *and* $d = 2$.

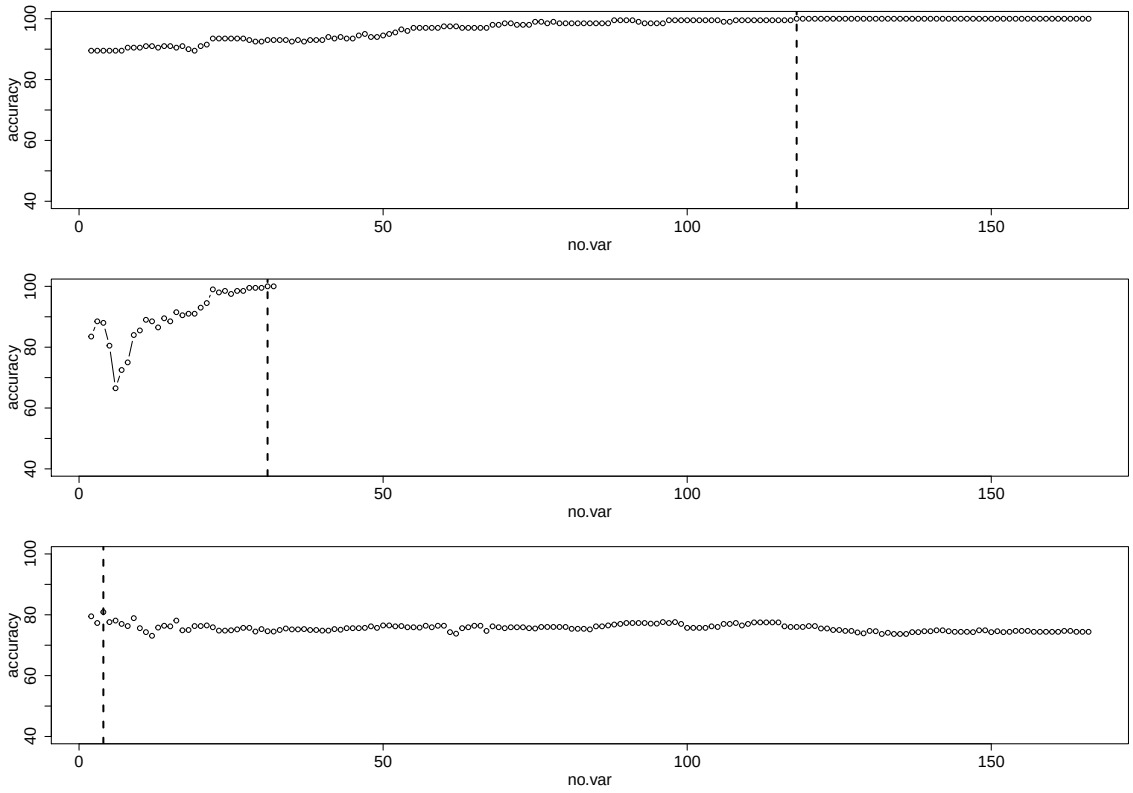


Figure B.52: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the musk data with $n = 200$ and $d = 5$.

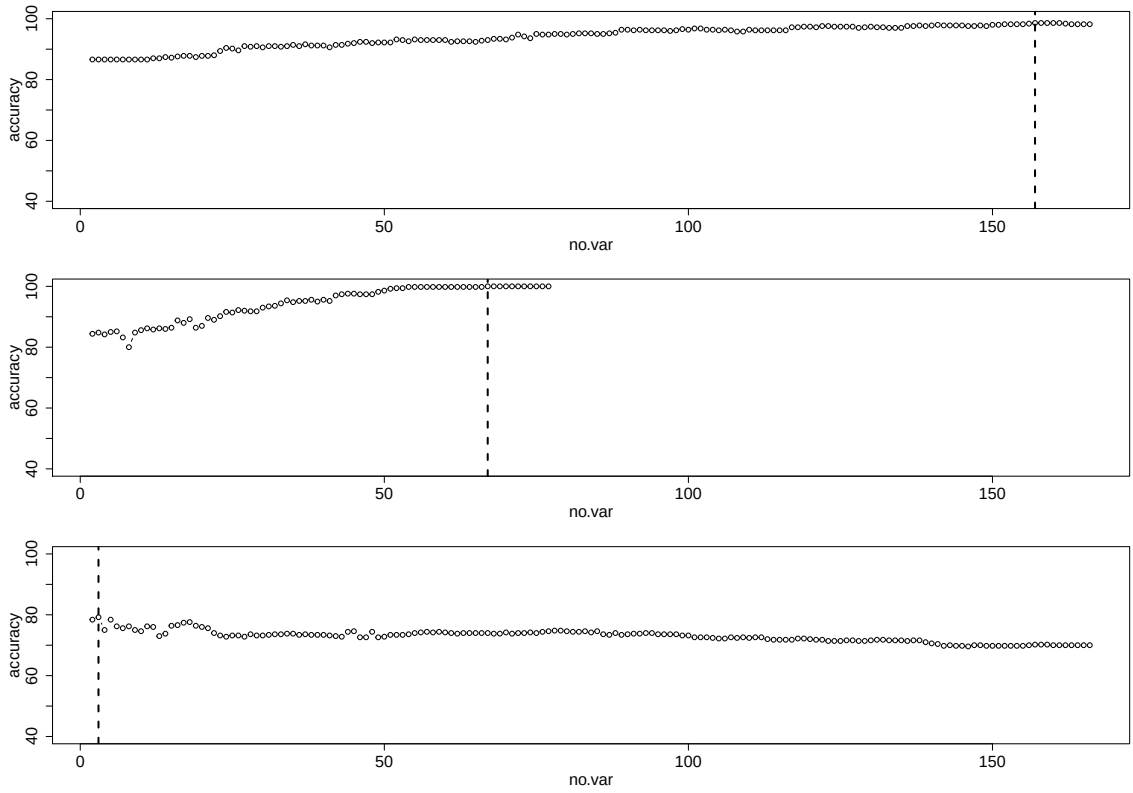


Figure B.53: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the musk data with $n = 500$ and $d = 2$.

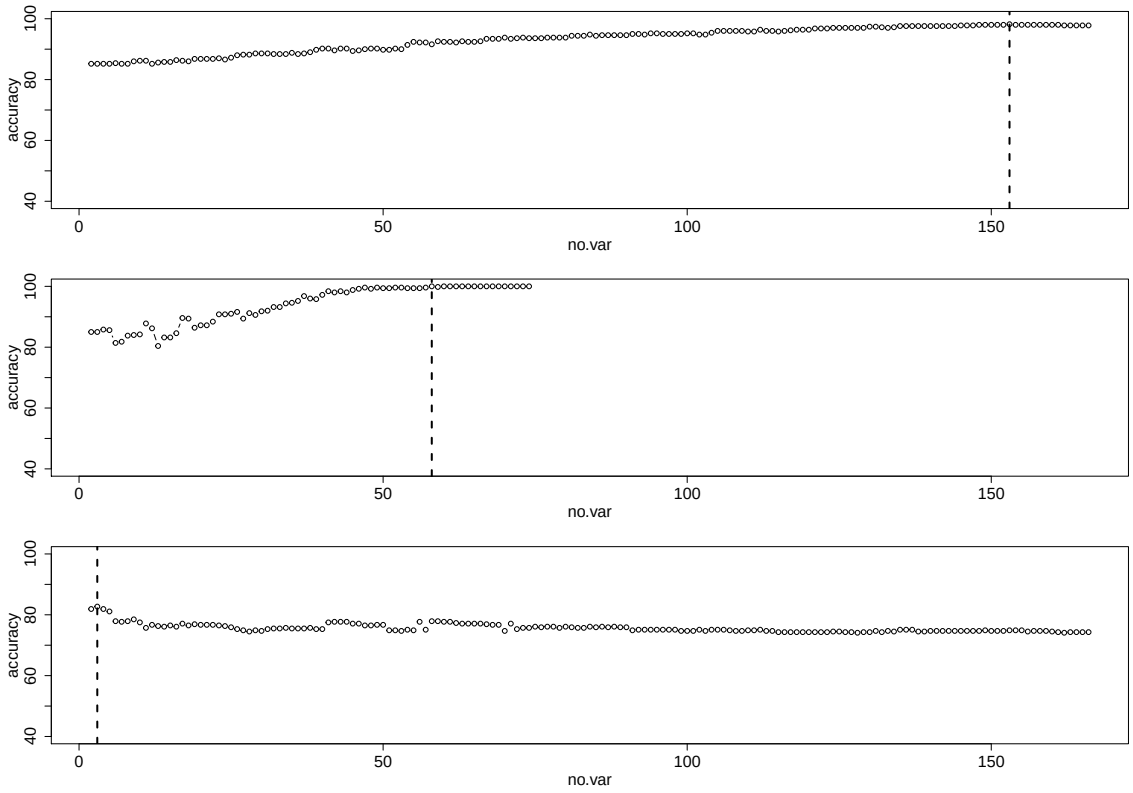


Figure B.54: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the musk data with $n = 500$ and $d = 5$.

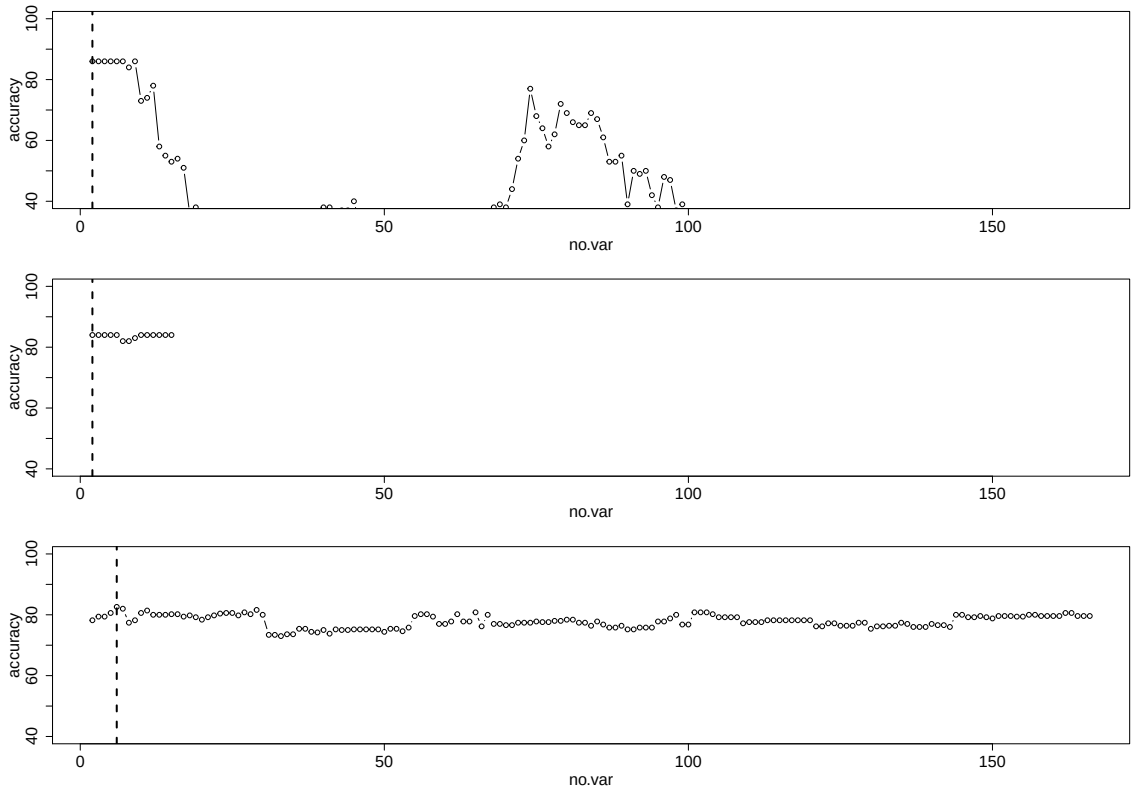


Figure B.55: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the musk data with $n = 100$ and $d = 5$.

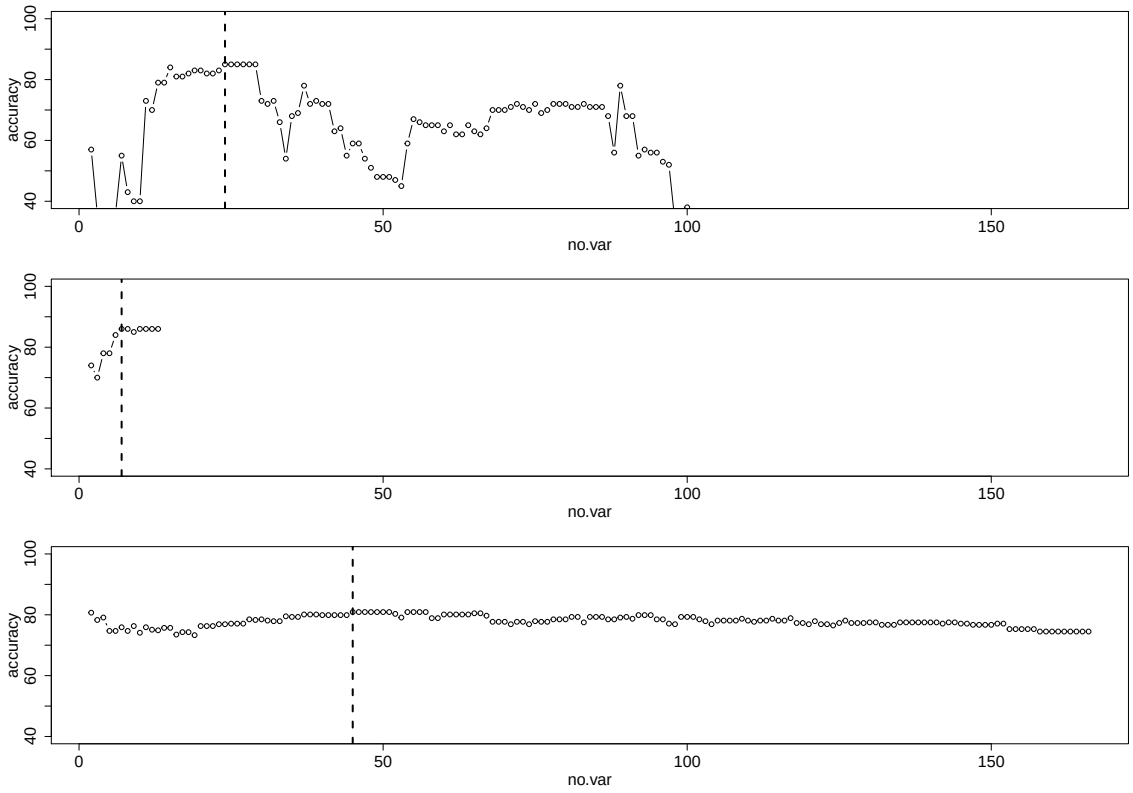


Figure B.56: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the musk data with $n = 100$ and $d = 5$.

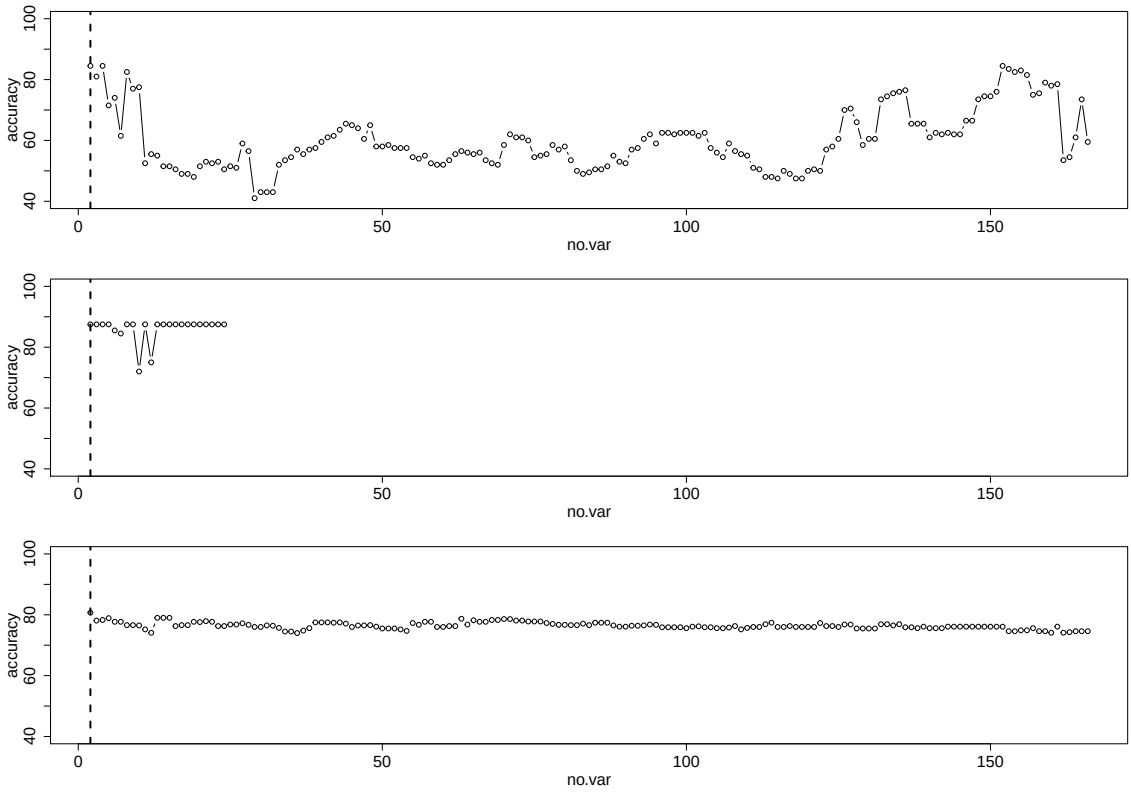


Figure B.57: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the musk data with $n = 200$ and $d = 2$.

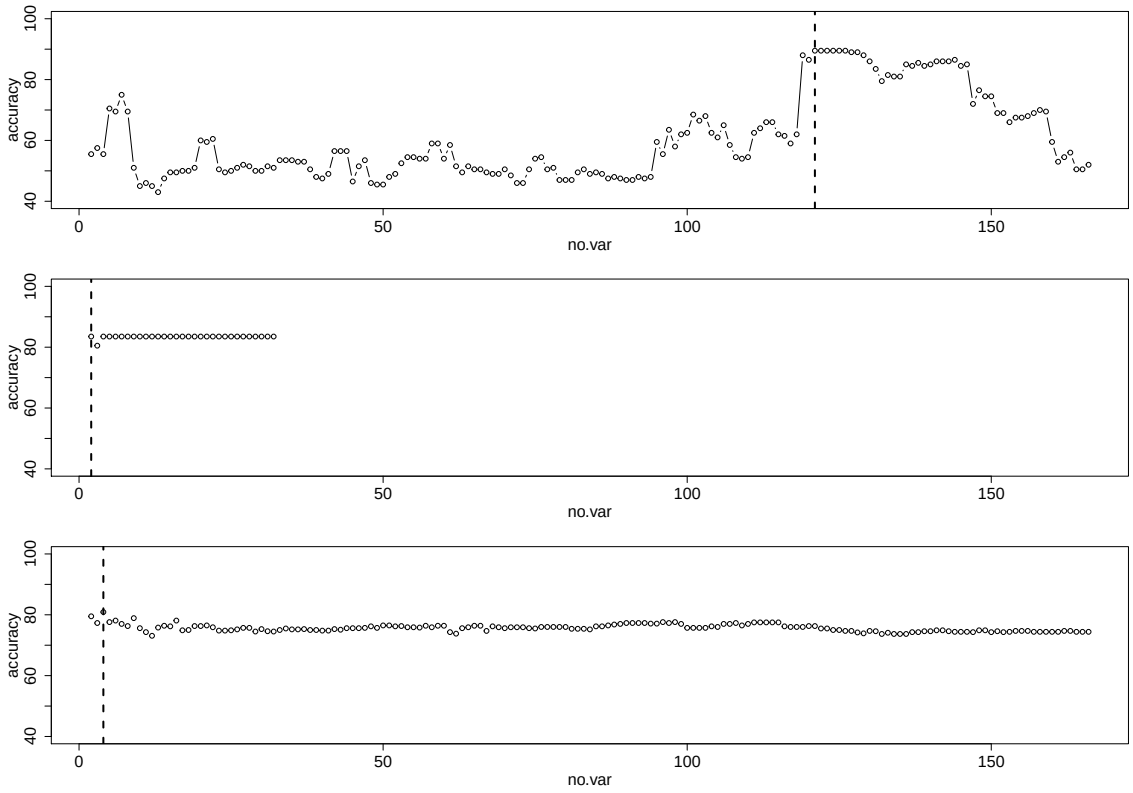


Figure B.58: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the musk data with $n = 200$ and $d = 5$.

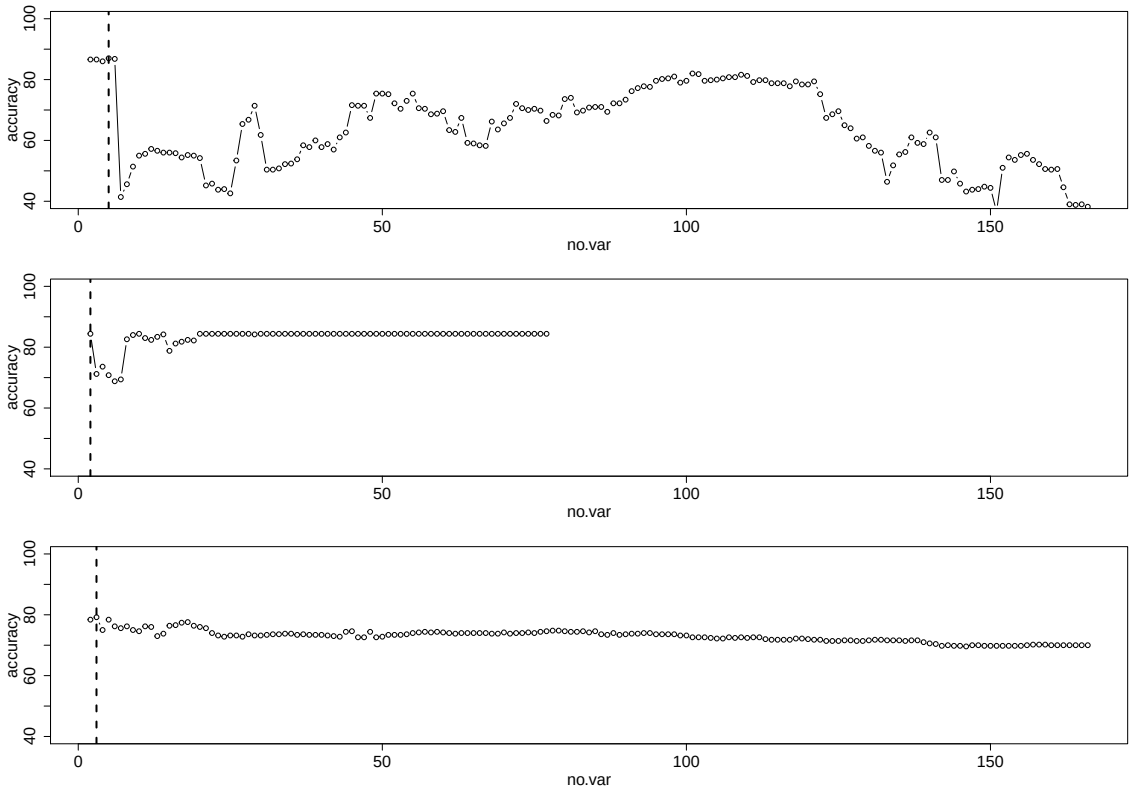


Figure B.59: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the musk data with $n = 500$ and $d = 2$.

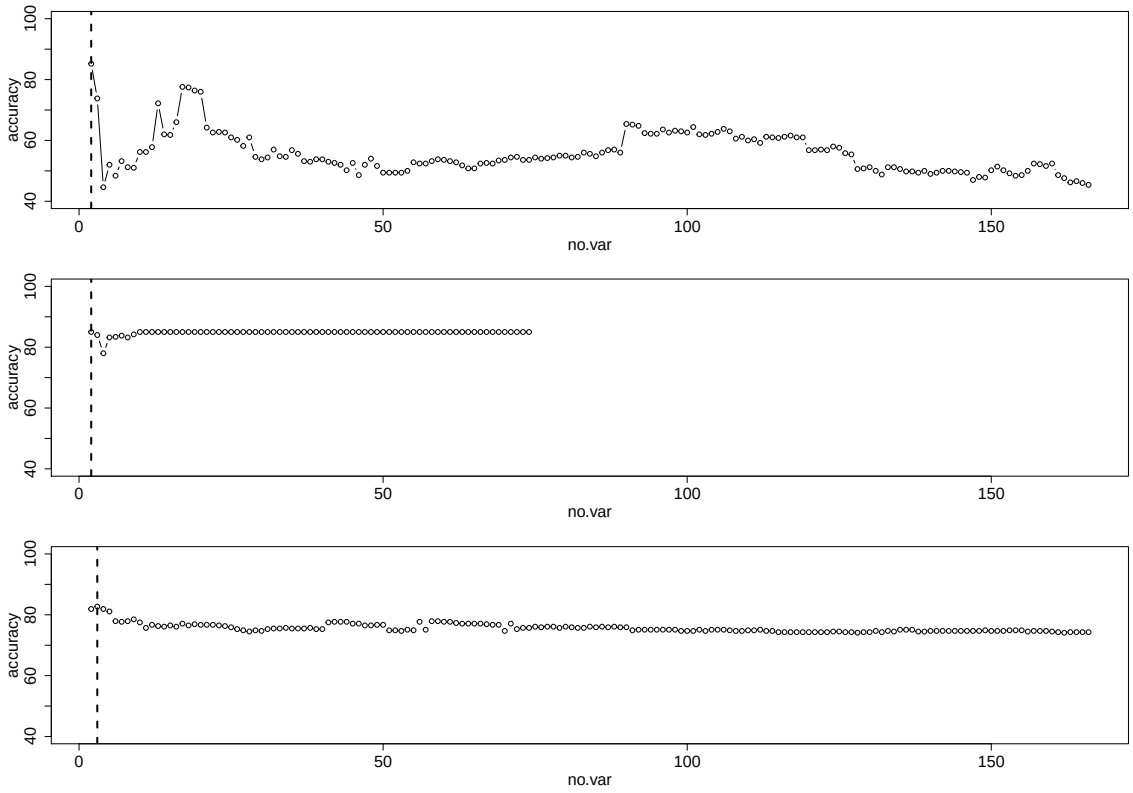


Figure B.60: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the musk data with $n = 500$ and $d = 5$.

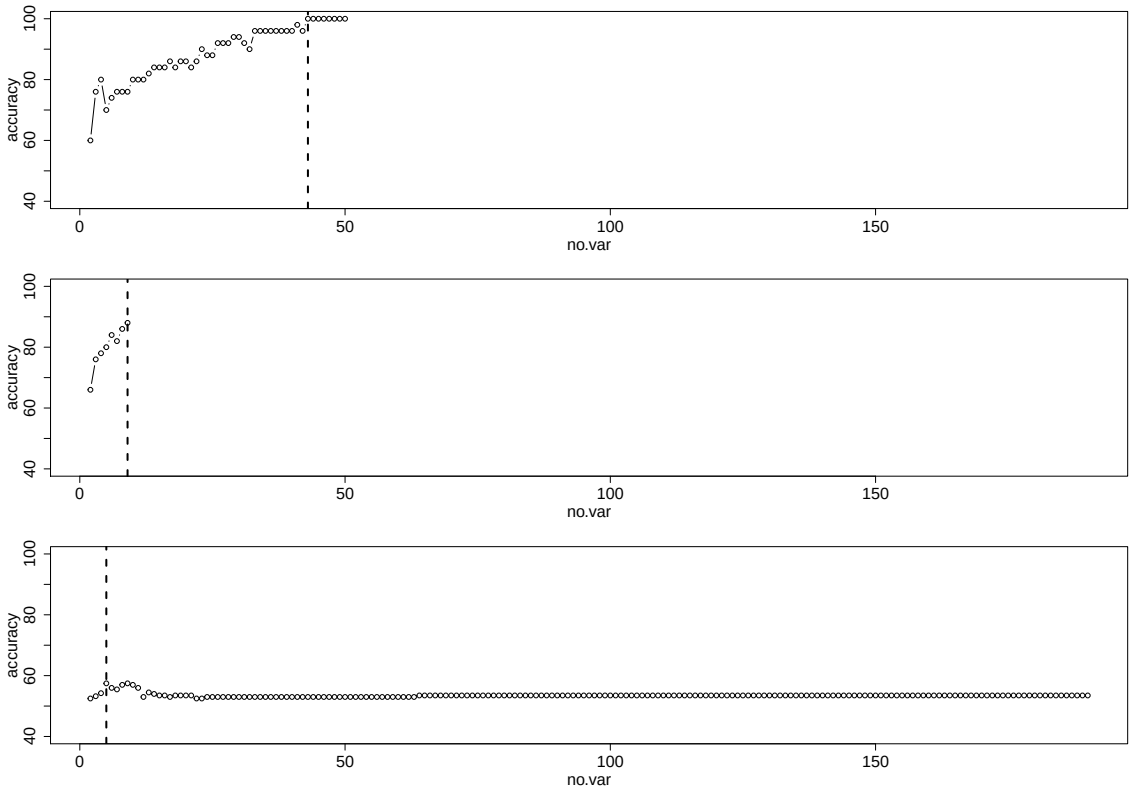


Figure B.61: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the cardiac arrhythmia data with $n = 50$ and $d = 2$.

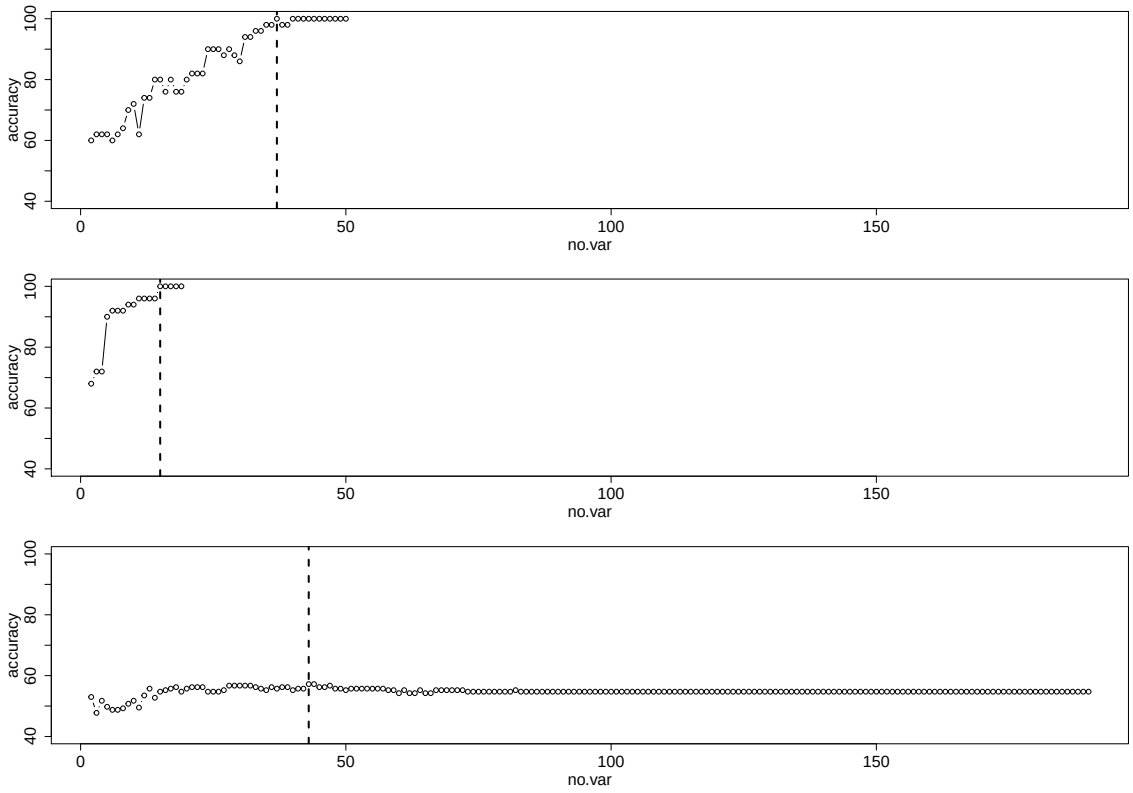


Figure B.62: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the cardiac arrhythmia data with $n = 50$ and $d = 5$.

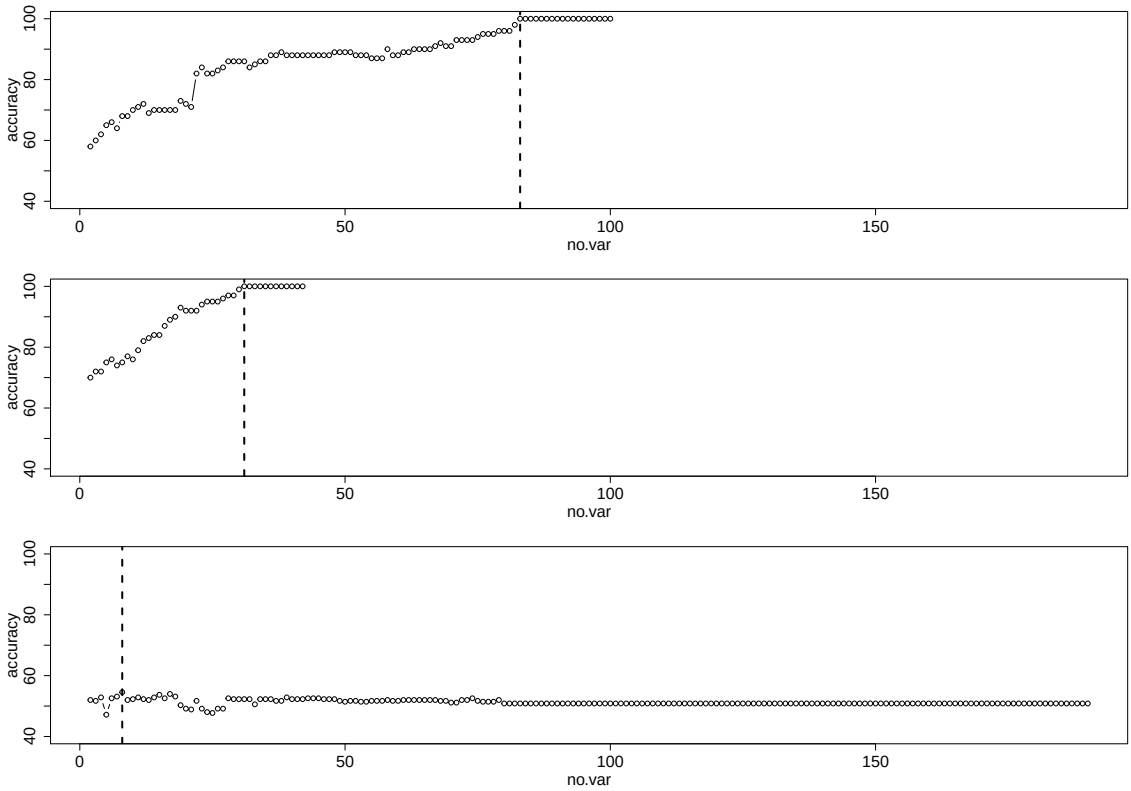


Figure B.63: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the cardiac arrhythmia data with $n = 100$ and $d = 2$.

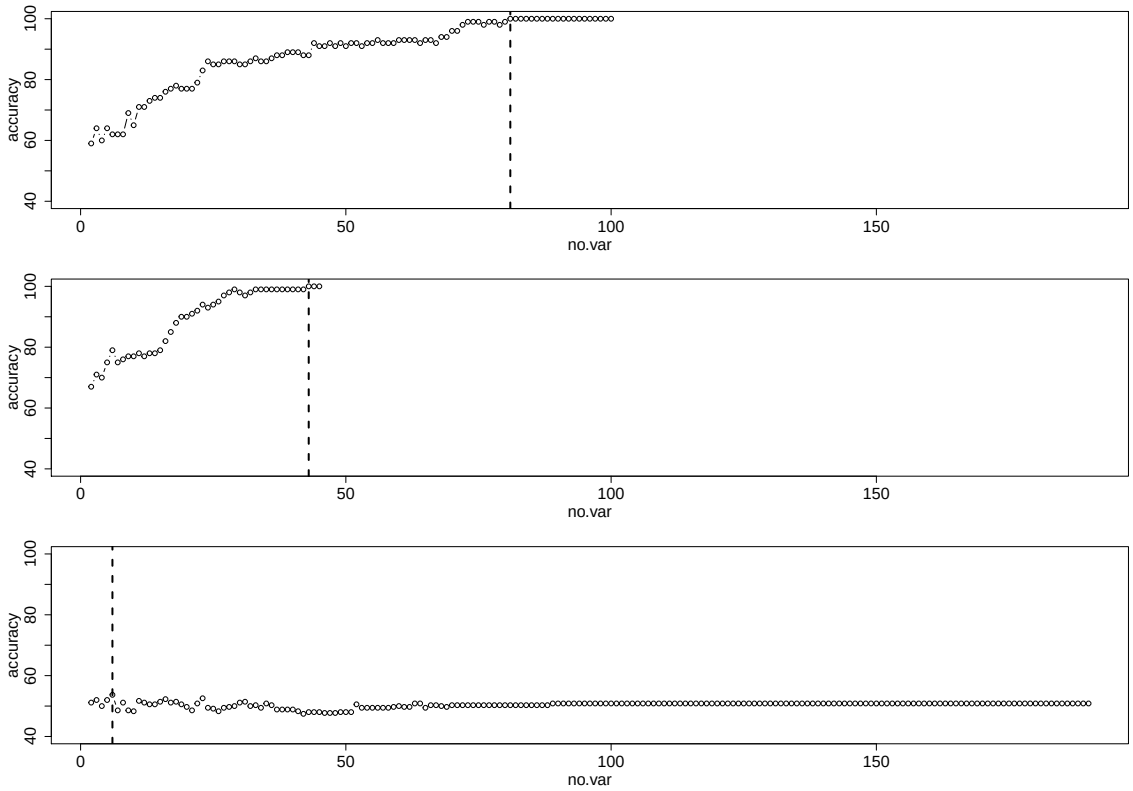


Figure B.64: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the cardiac arrhythmia data with $n = 100$ *and* $d = 5$.

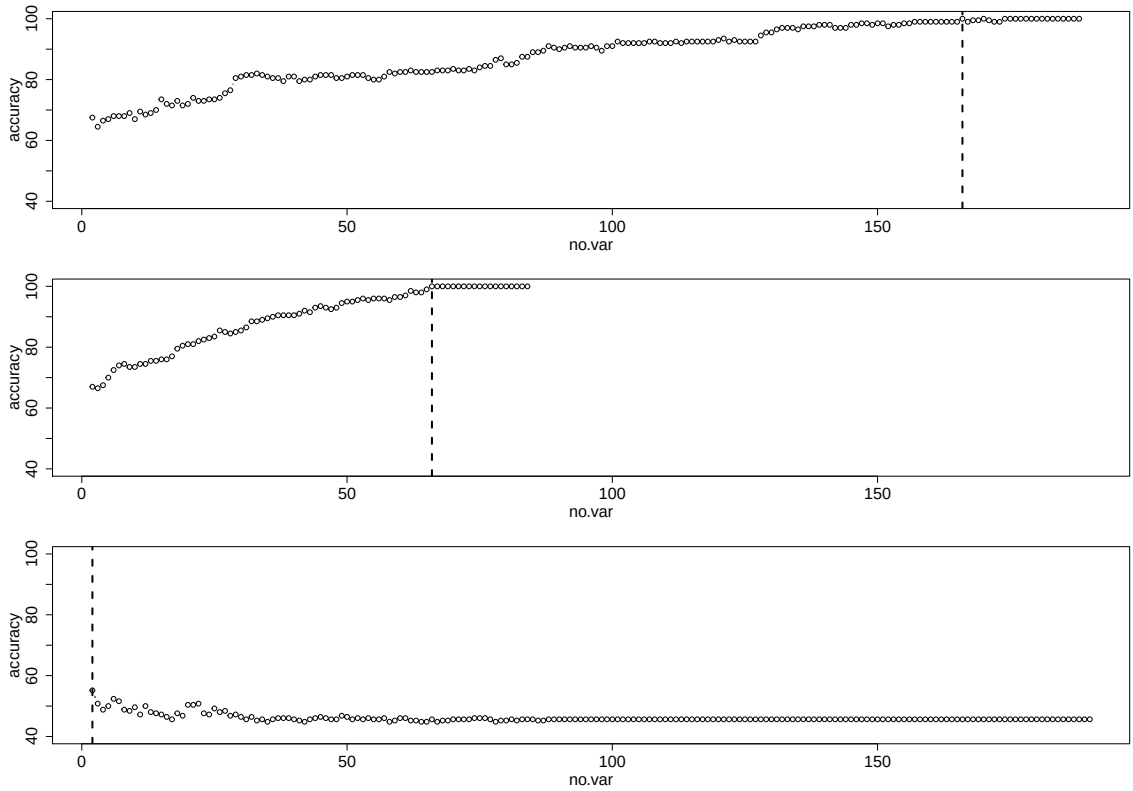


Figure B.65: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the cardiac arrhythmia data with $n = 200$ and $d = 2$.

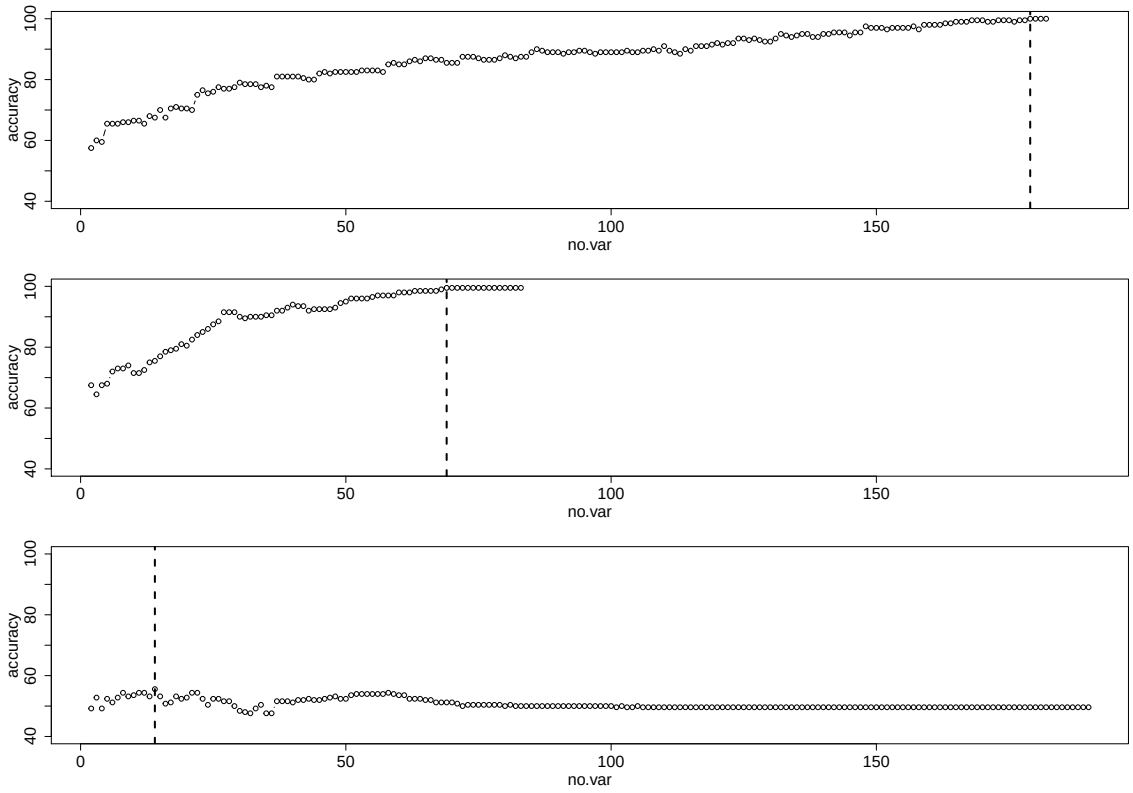


Figure B.66: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the cardiac arrhythmia data with $n = 200$ and $d = 5$.

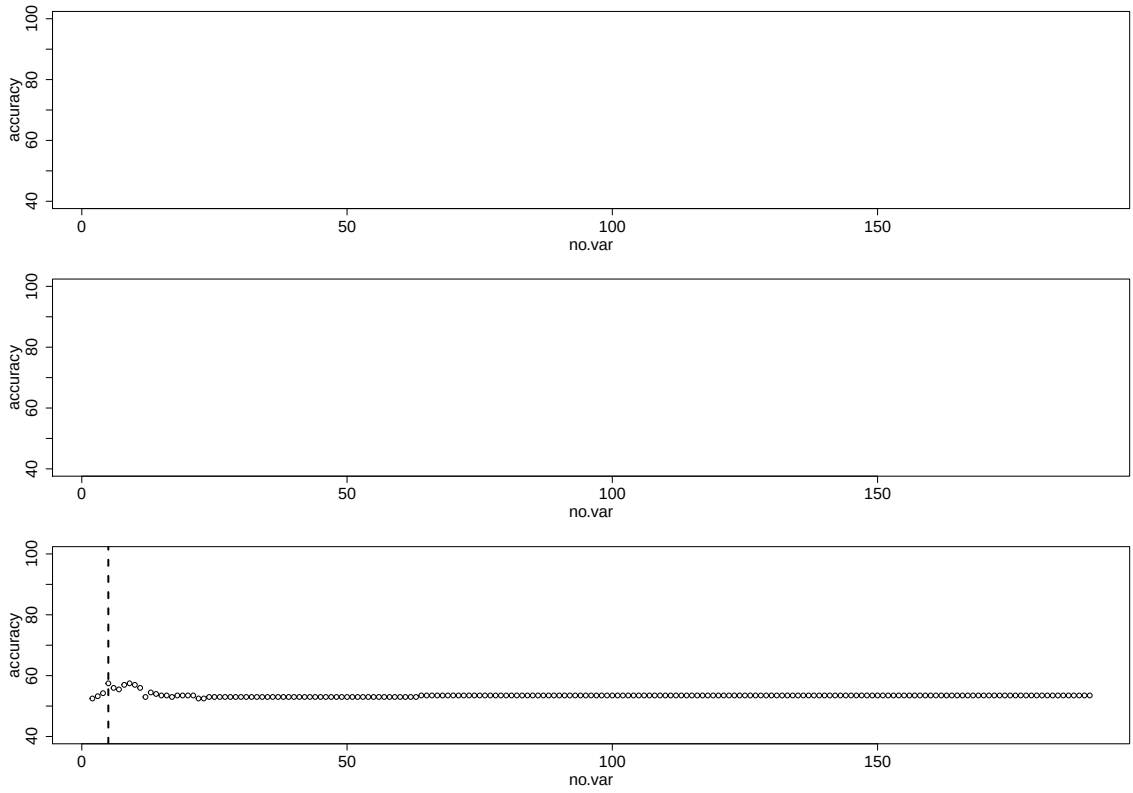


Figure B.67: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the cardiac arrhythmia data with $n = 50$ and $d = 2$.

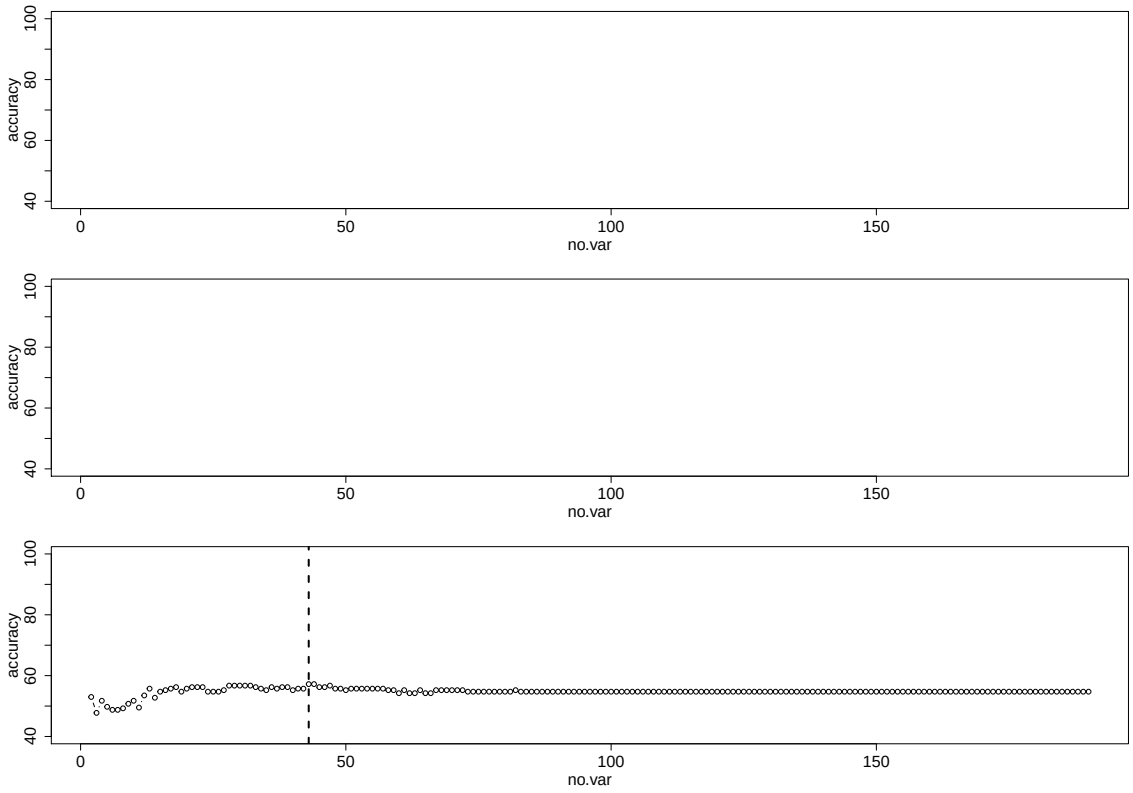


Figure B.68: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the cardiac arrhythmia data with $n = 50$ and $d = 5$.

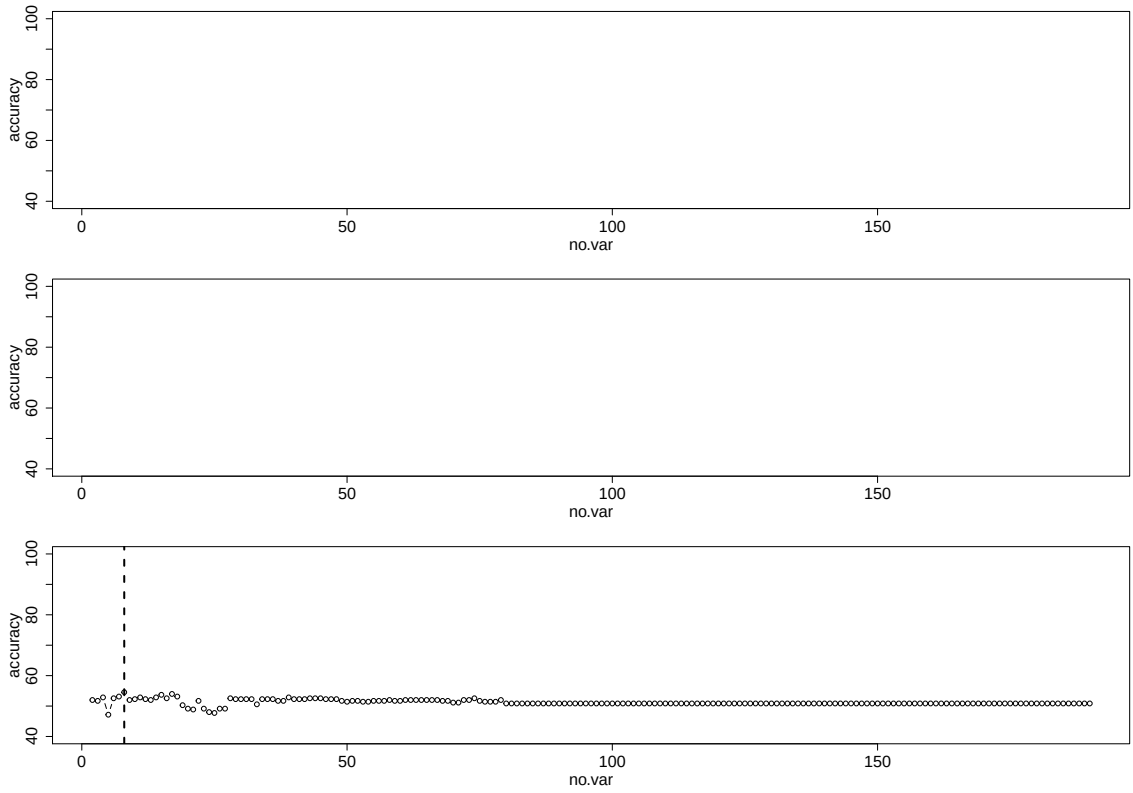


Figure B.69: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the cardiac arrhythmia data with $n = 100$ *and* $d = 2$.

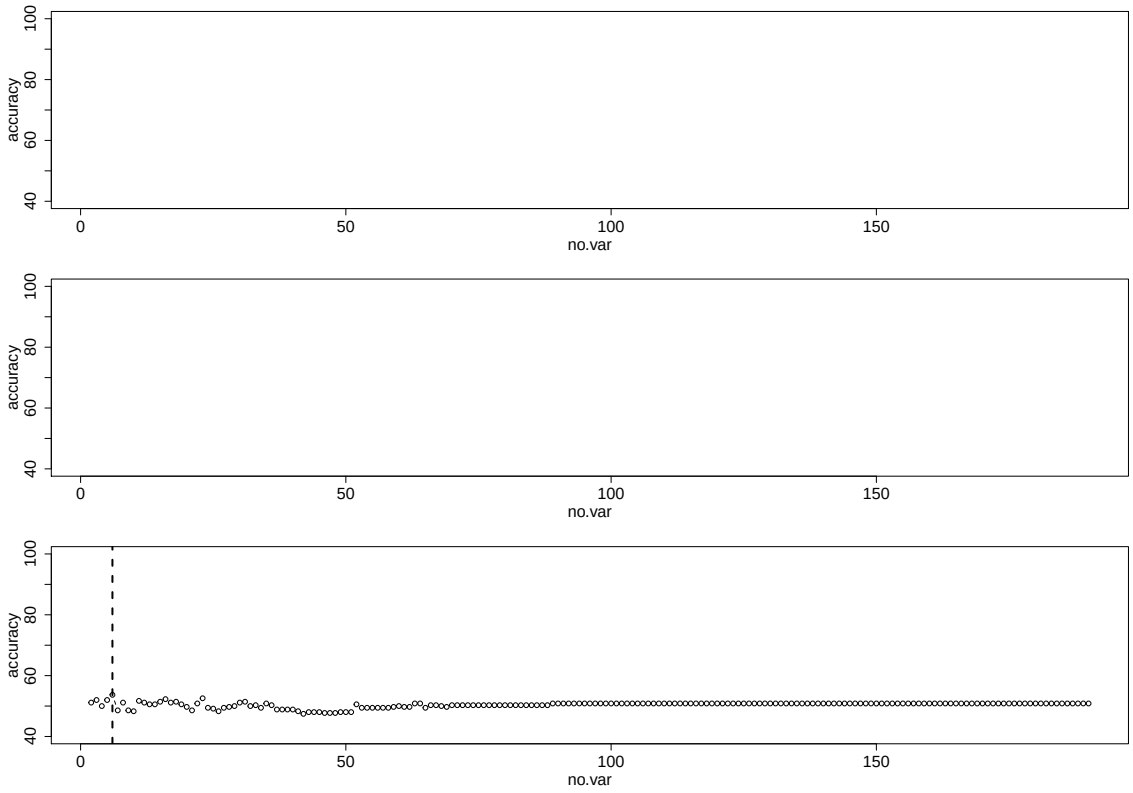


Figure B.70: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the cardiac arrhythmia data with $n = 100$ and $d = 5$.

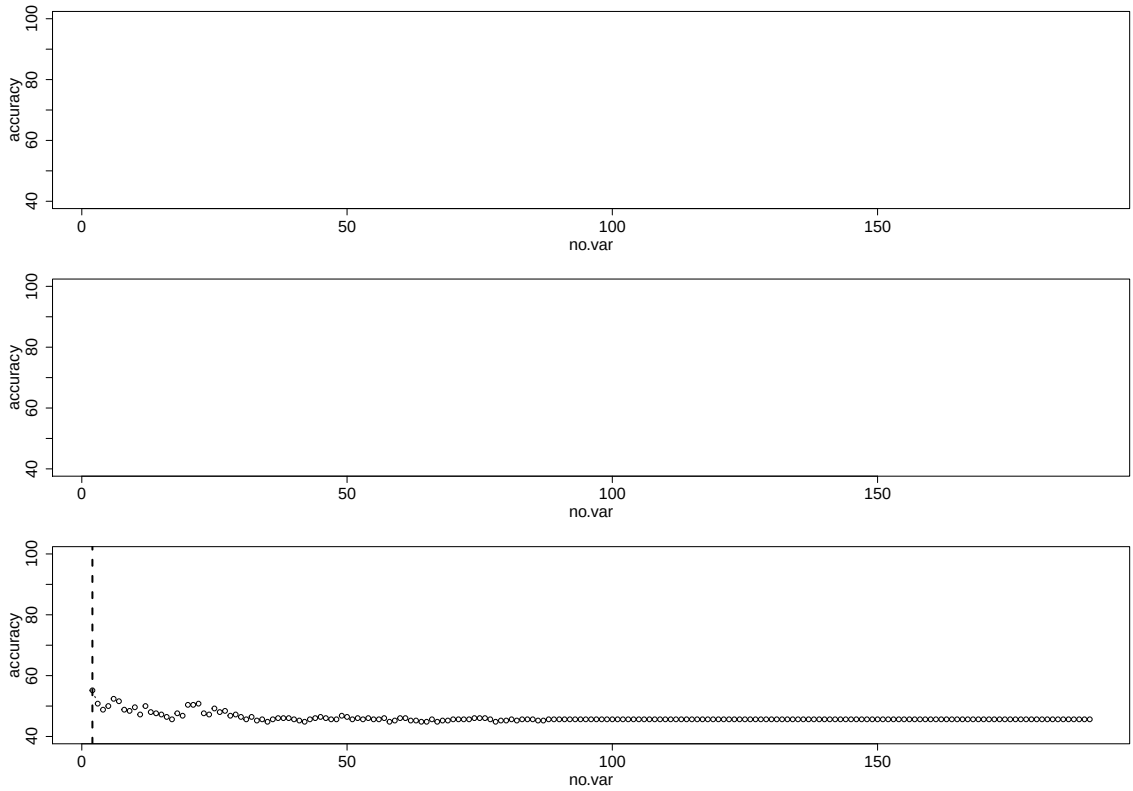


Figure B.71: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the cardiac arrhythmia data with $n = 200$ and $d = 2$.

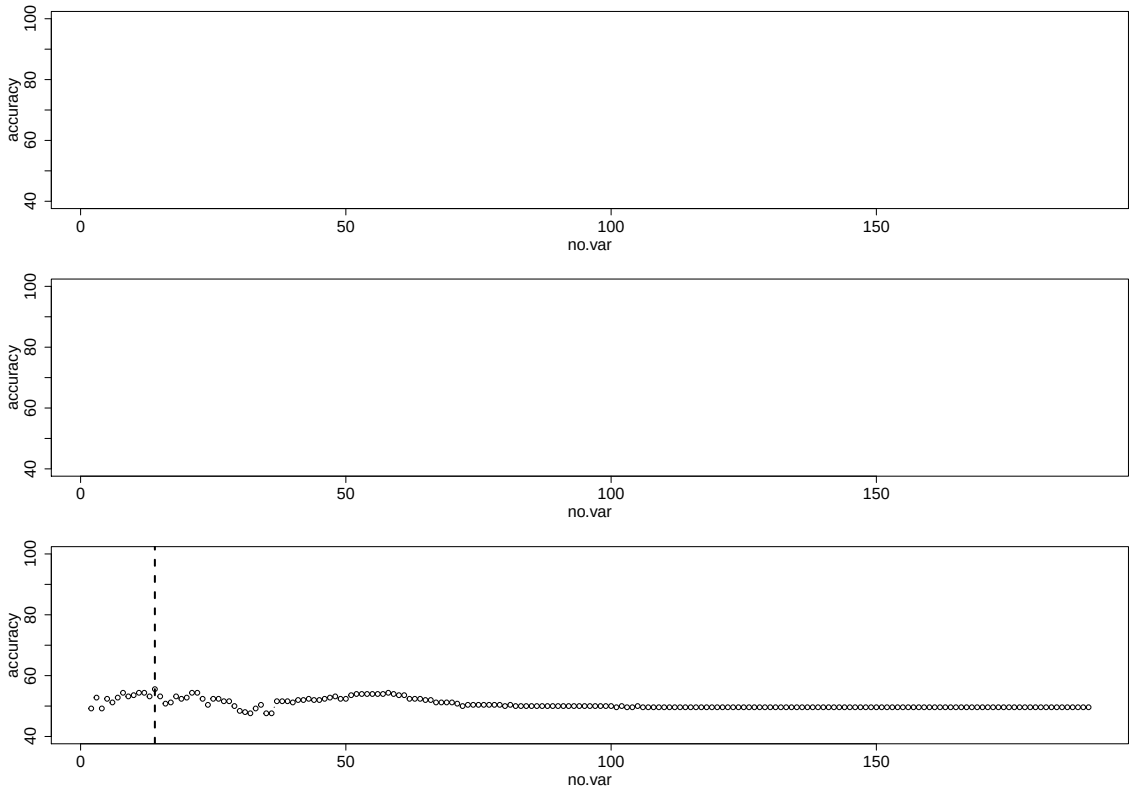


Figure B.72: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the cardiac arrhythmia data with $n = 200$ and $d = 5$.

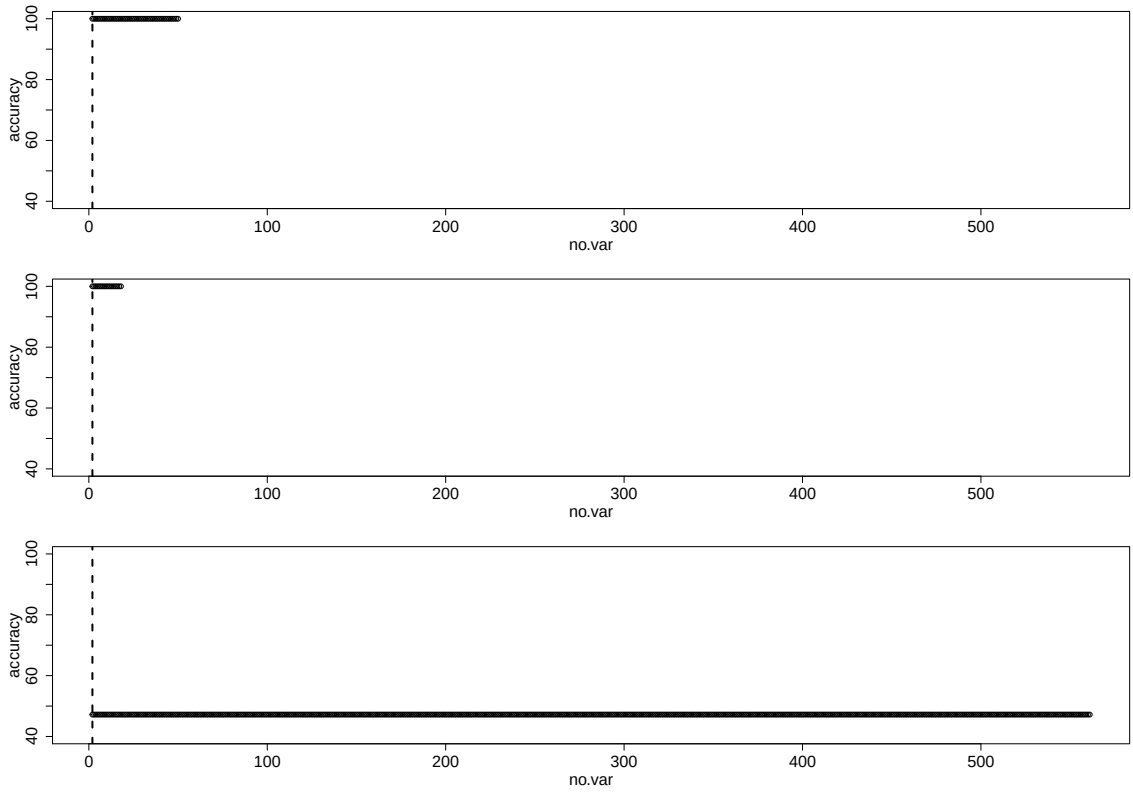


Figure B.73: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the human activity recognition data with $n = 50$ and $d = 2$.

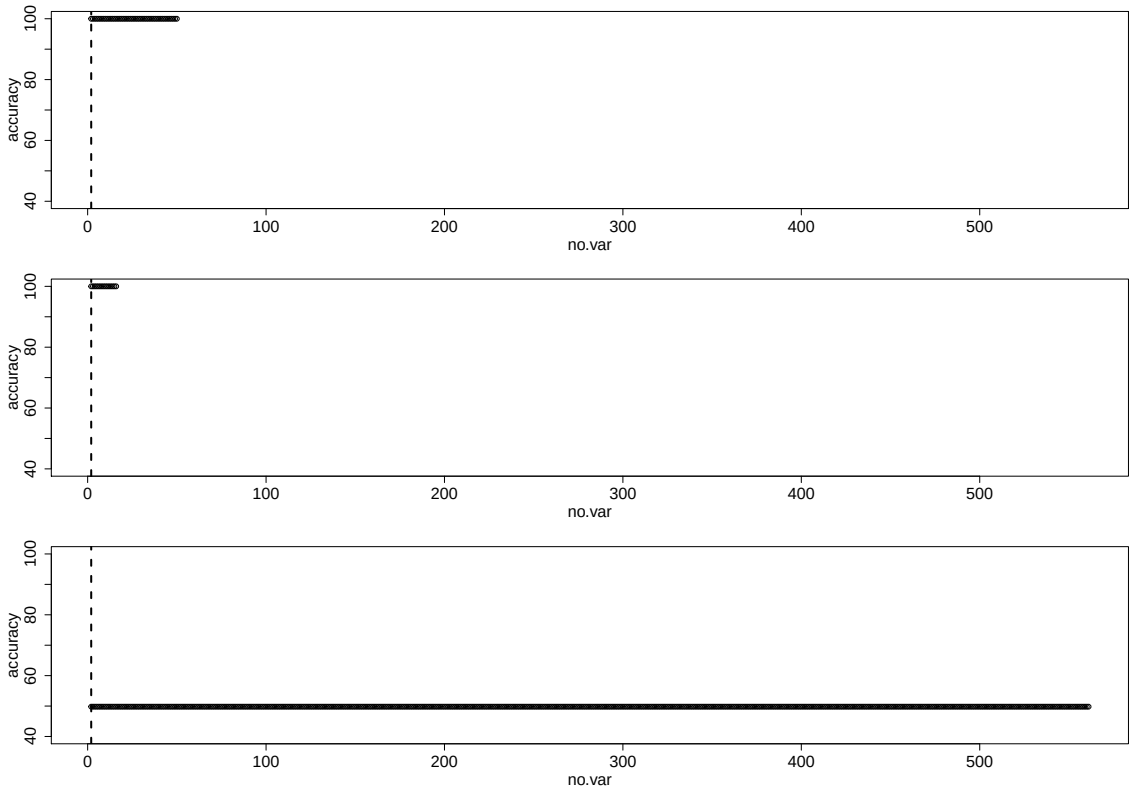


Figure B.74: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the human activity recognition data with $n = 50$ and $d = 5$.

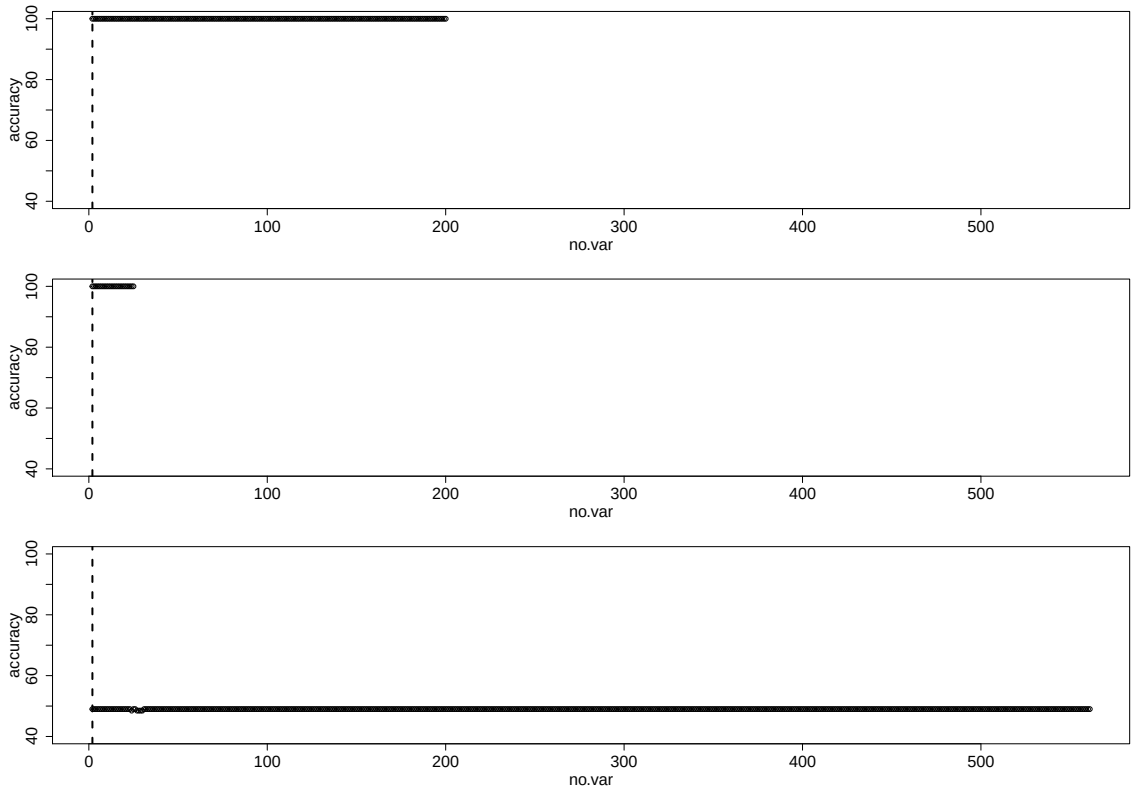


Figure B.75: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the human activity recognition with $n = 200$ *and* $d = 2$.

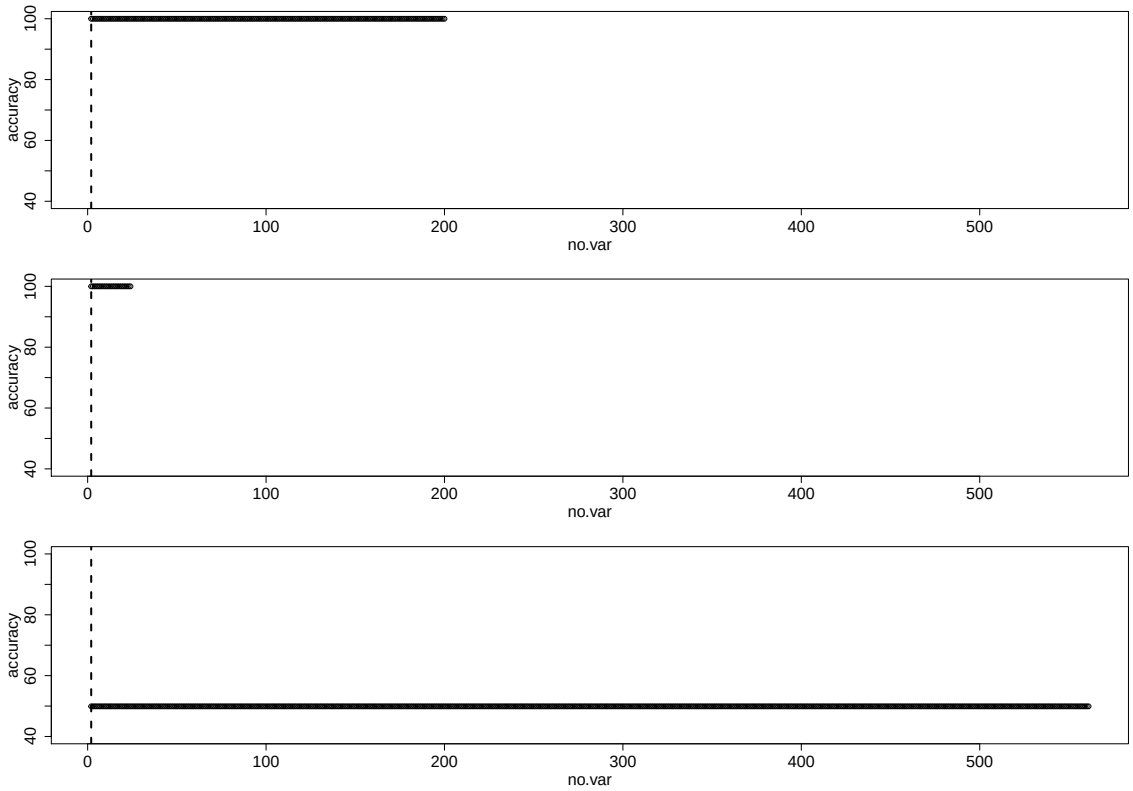


Figure B.76: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the human activity recognition data with $n = 200$ and $d = 5$.

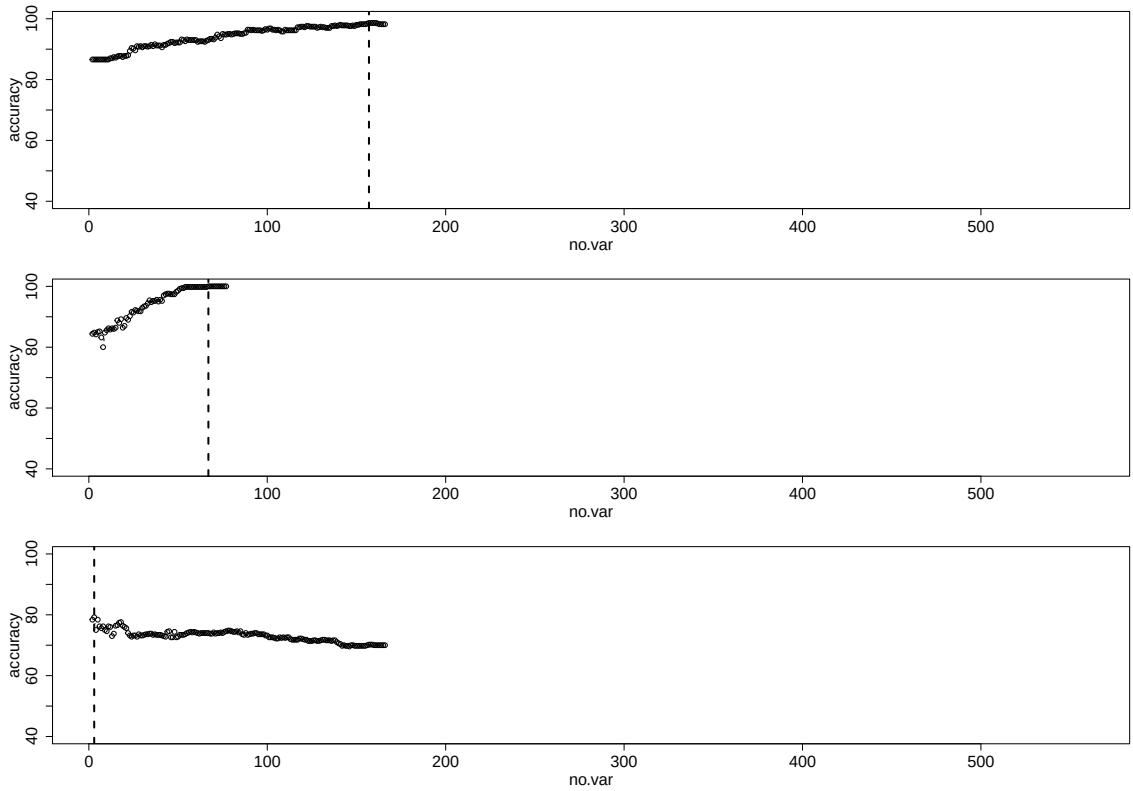


Figure B.77: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the human activity recognition data with $n = 1000$ *and* $d = 2$.

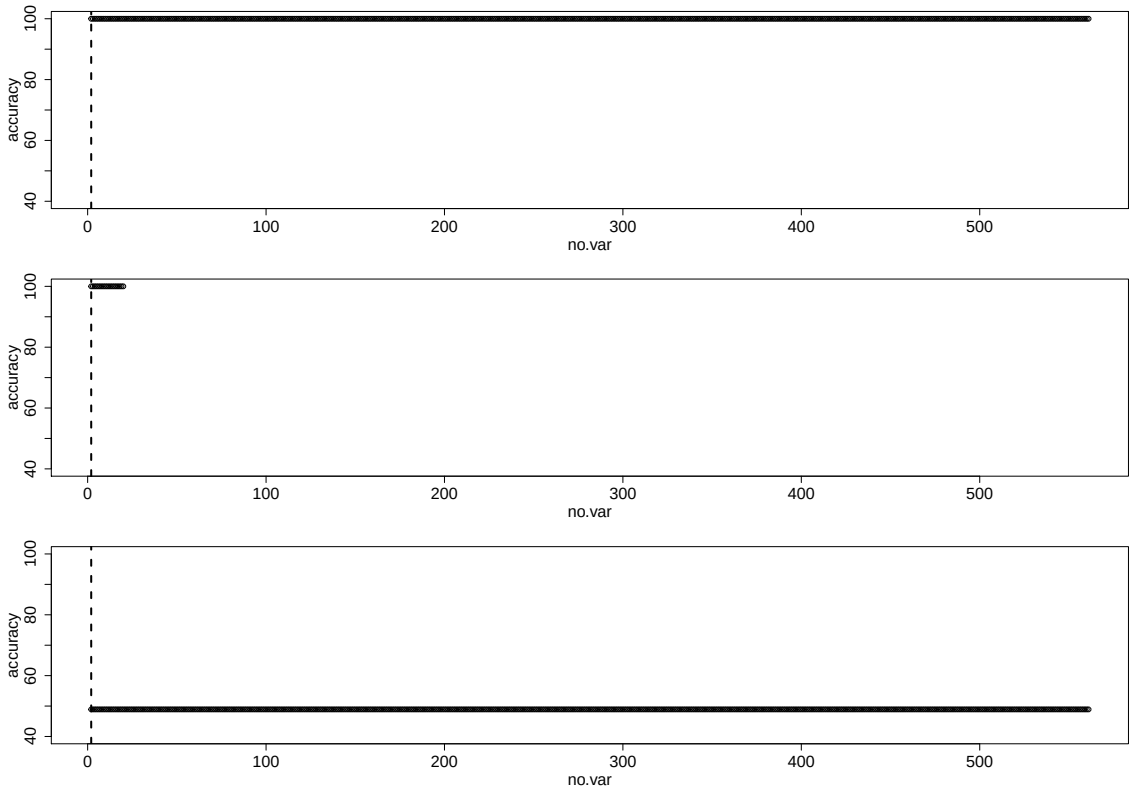


Figure B.78: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the VIP-LDA (first row), VIP-QDA (second row) and VIP-Knn (third row) for the human activity recognition data with $n = 1000$ *and* $d = 5$.

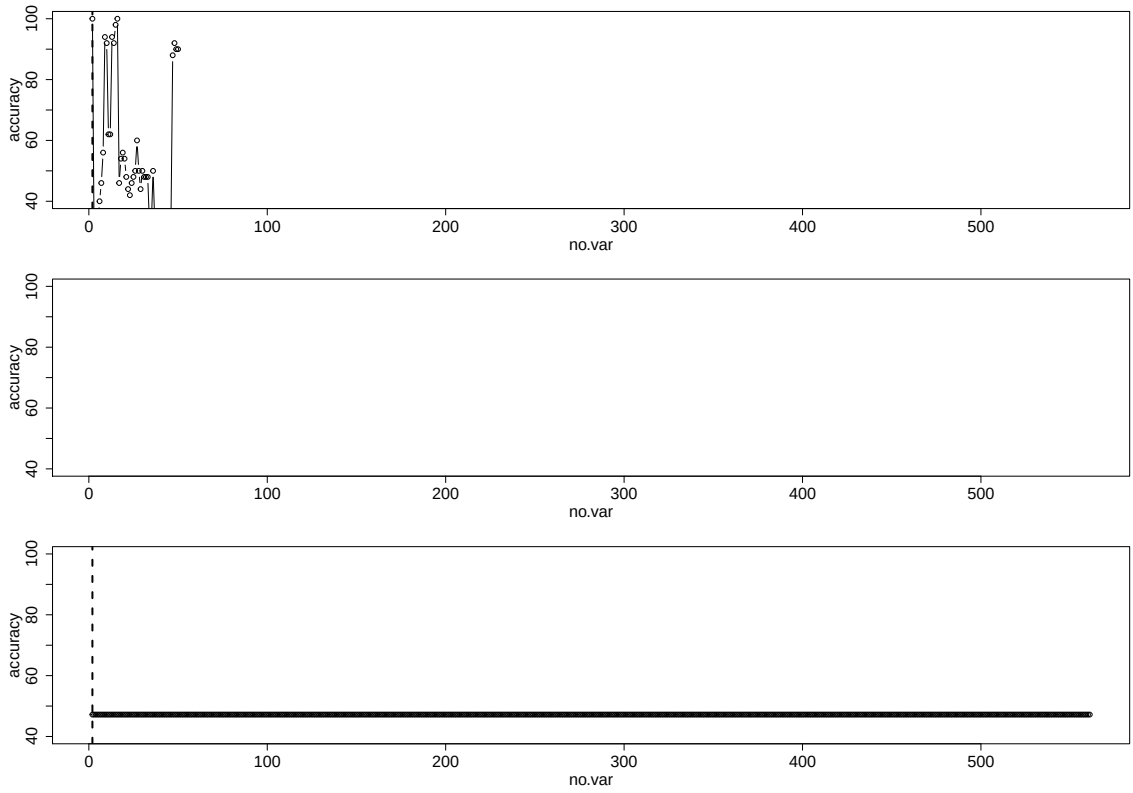


Figure B.79: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the human activity recognition data with $n = 50$ and $d = 2$.

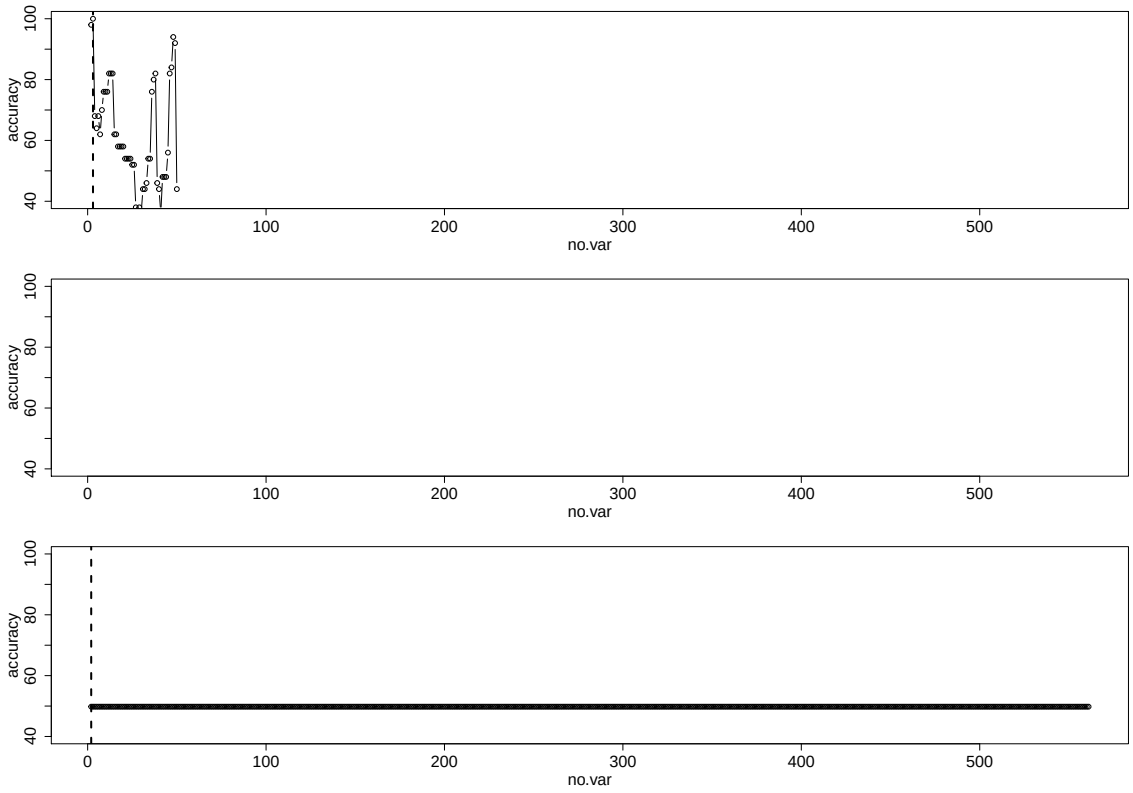


Figure B.80: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the human activity recognition data with $n = 50$ and $d = 5$.

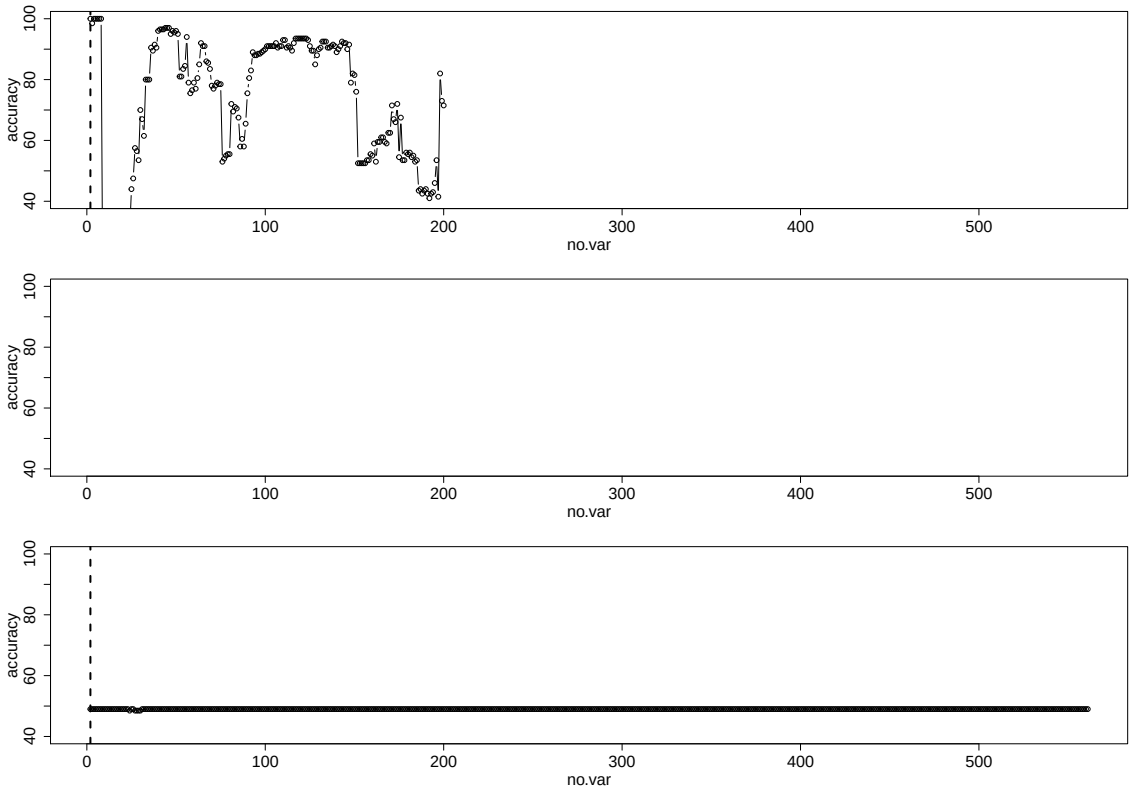


Figure B.81: Accuracy rate (_____) and optimal number of variables, h, (- - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the human activity recognition with $n = 200$ and $d = 2$.

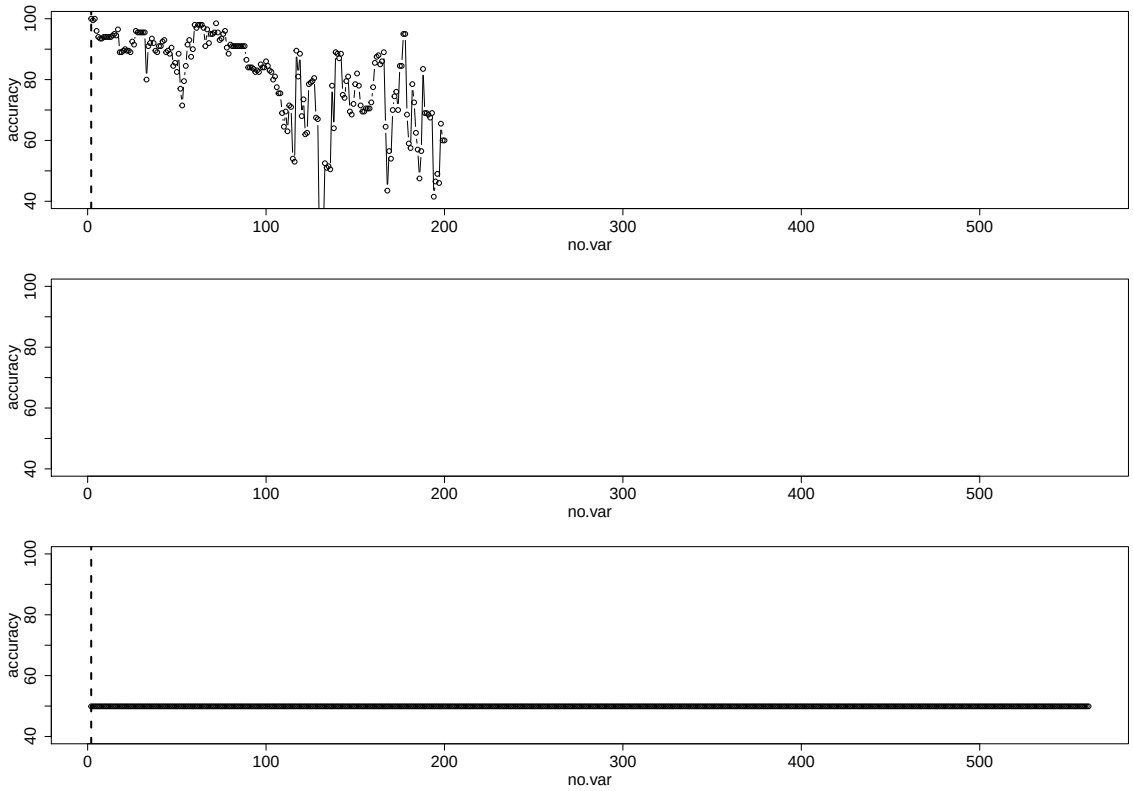


Figure B.82: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the human activity recognition data with $n = 200$ and $d = 5$.

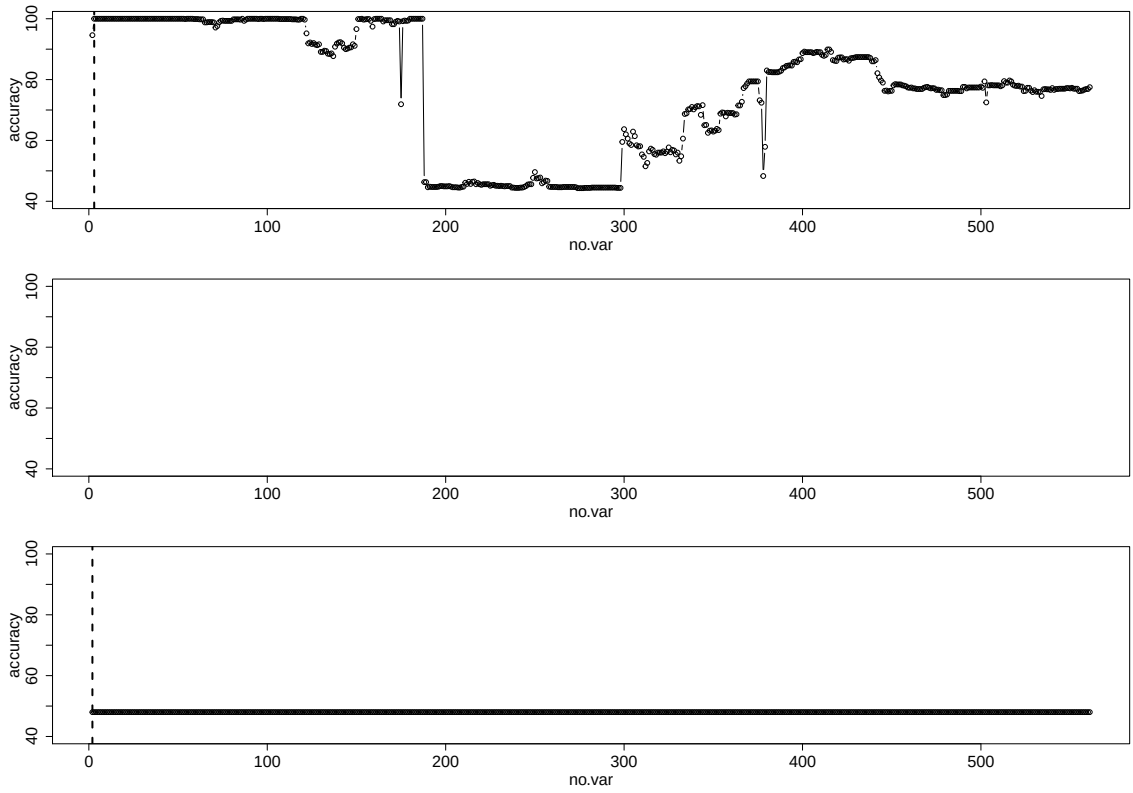


Figure B.83: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the human activity recognition data with $n = 1000$ *and* $d = 2$.

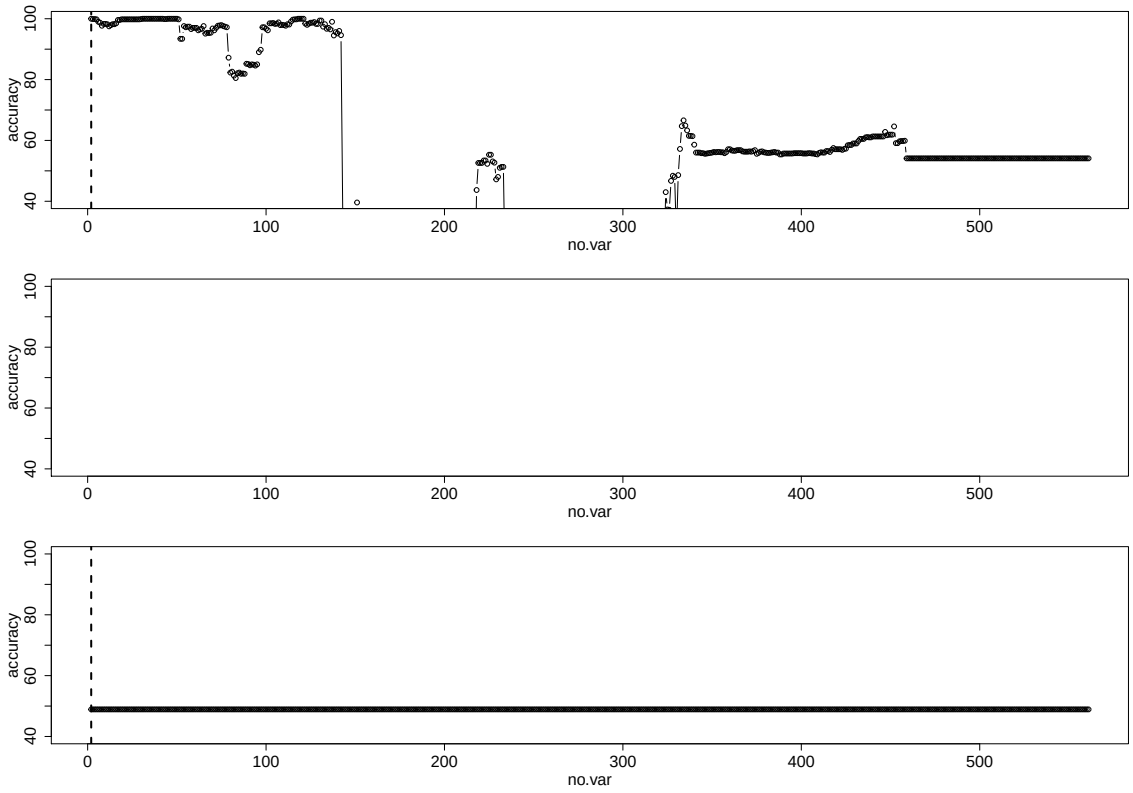
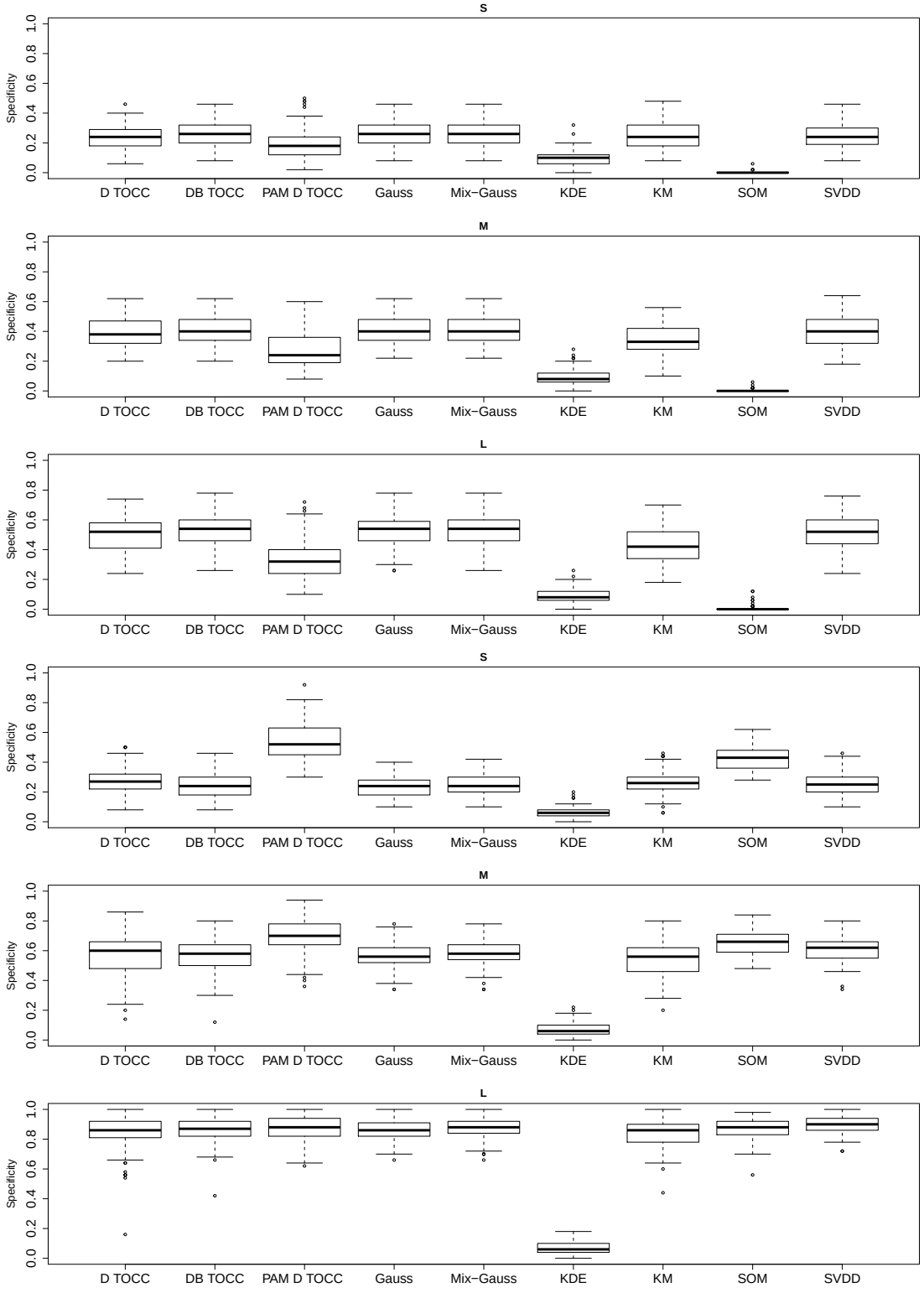


Figure B.84: Accuracy rate (_____) and optimal number of variables, h, (- - - - - - - - -) for the AA-LDA (first row), AA-QDA (second row) and AA-Knn (third row) for the human activity recognition data with $n = 1000$ *and* $d = 5$.

Appendix C

C.1 Simulation results for One-class classification



$p = 1$

$p = 2$

Figure C.1: Specificity for $s \geq 0.9$ sensitivity level for small (S), medium (M) and large (L) shifts for Model 1, $n = 100$.

$p = 1$

$p = 2$

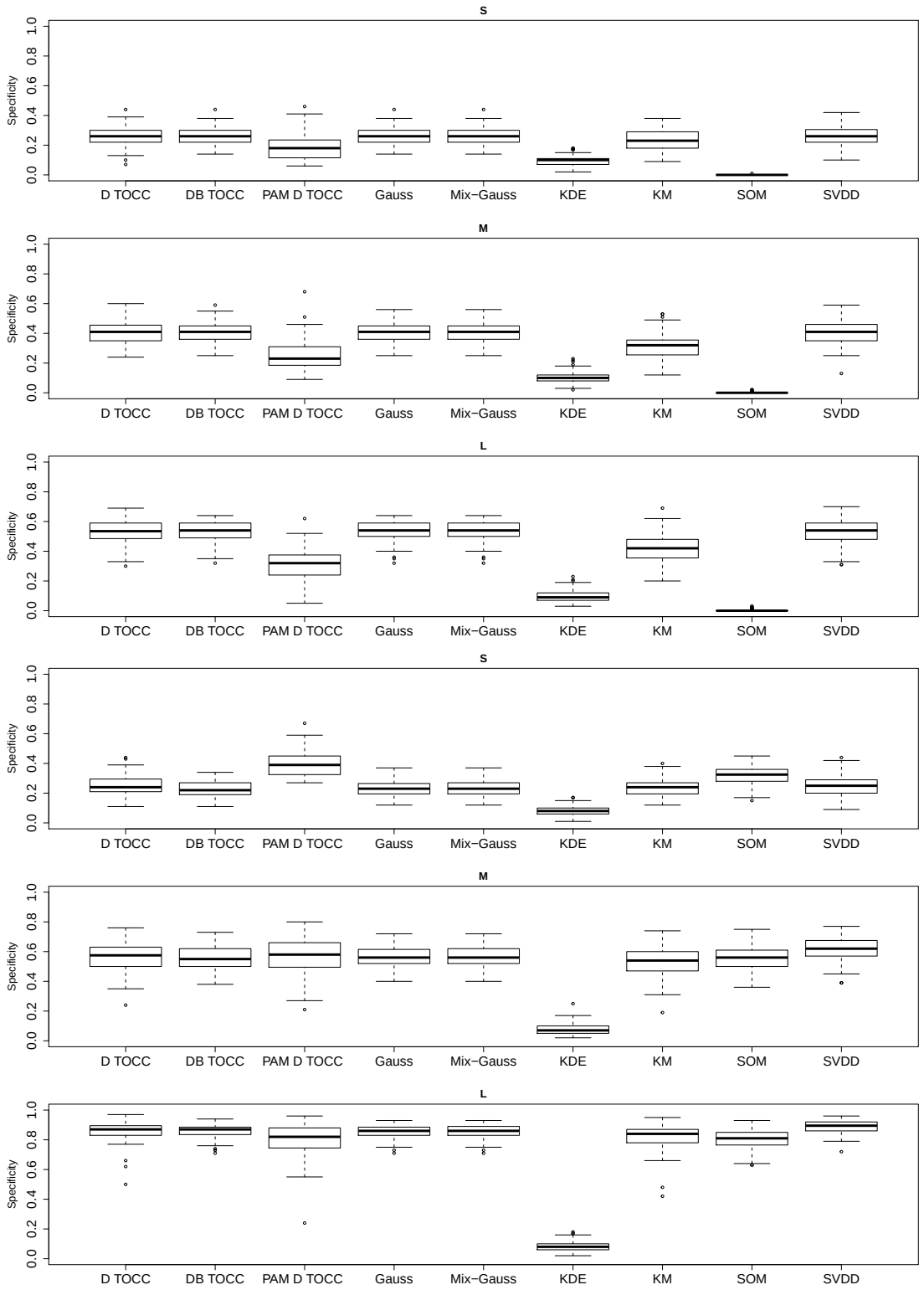
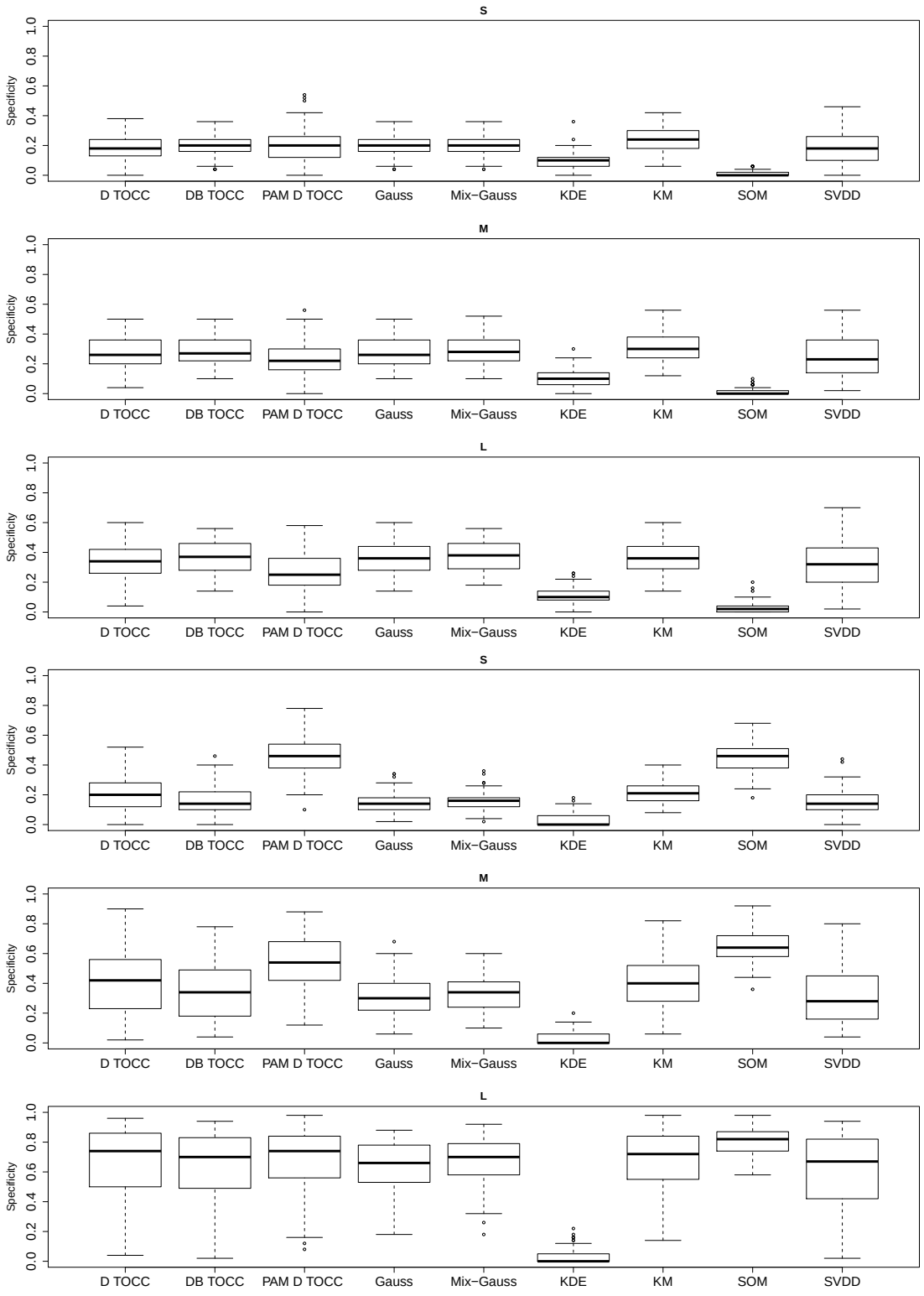


Figure C.2: Specificity for $s \geq 0.9$ sensitivity level for small (S), medium (M) and large (L) shifts for Model 1, $n = 200$.



$p = 1$

$p = 2$

Figure C.3: Specificity for $s \geq 0.9$ sensitivity level for small (S), medium (M) and large (L) shifts for Model 2, $n = 100$.

$p = 1$

$p = 2$

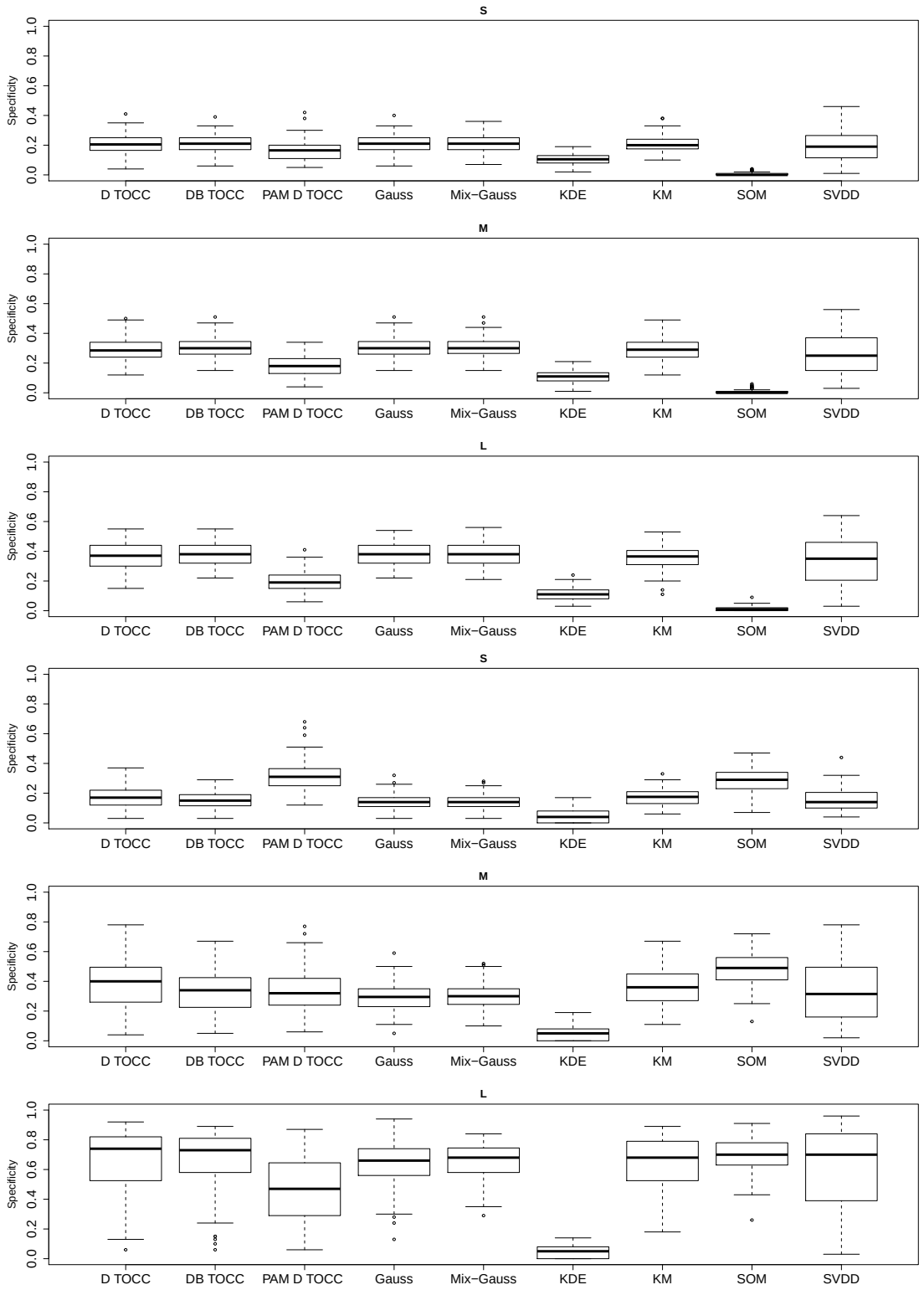
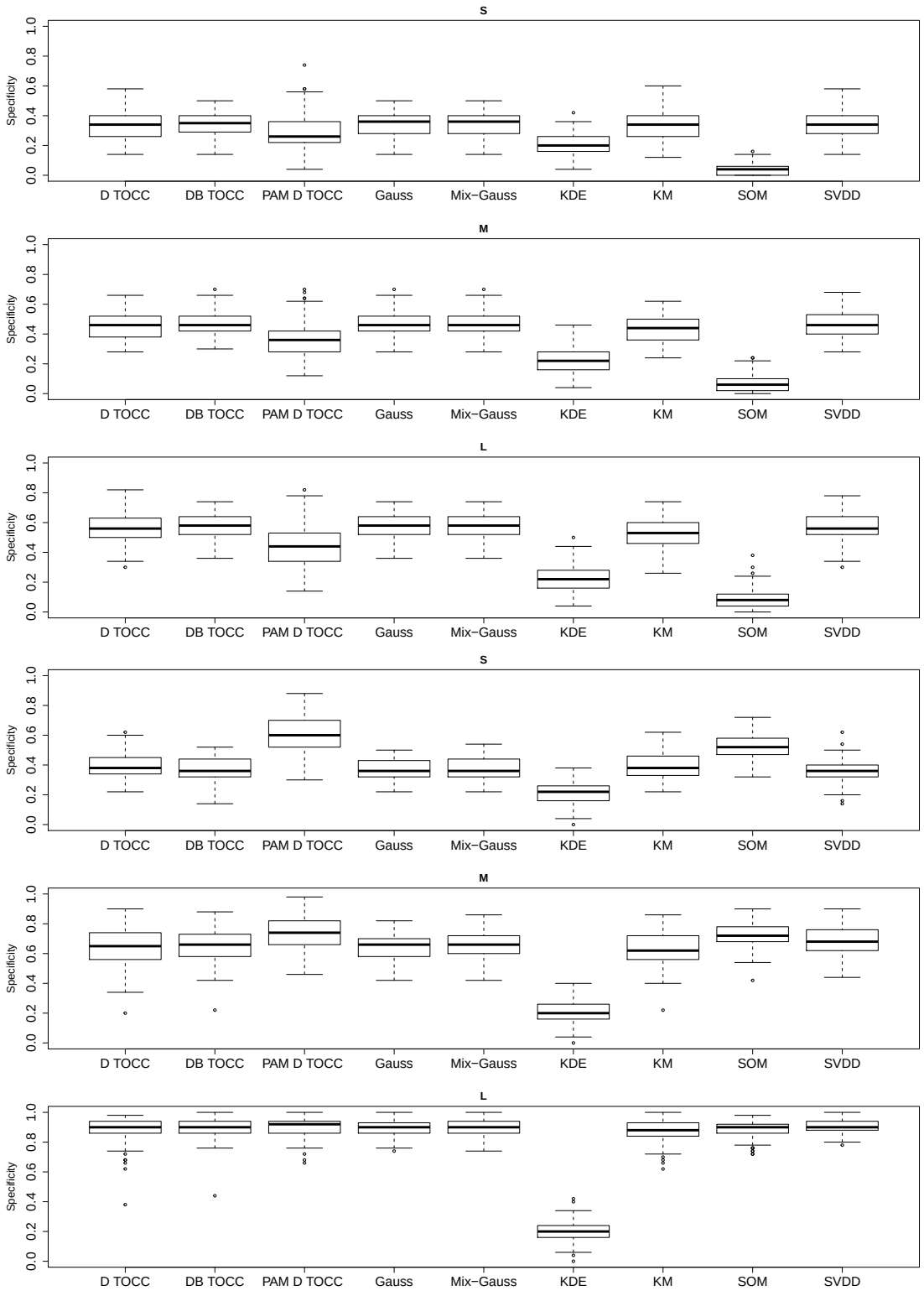


Figure C.4: Specificity for $s \geq 0.9$ sensitivity level for small (S), medium (M) and large (L) shifts for Model 2, $n = 200$.



$p = 1$

$p = 2$

Figure C.5: Specificity for $s \geq 0.9$ sensitivity level for small (S), medium (M) and large (L) shifts for Model 3, $n = 100$.

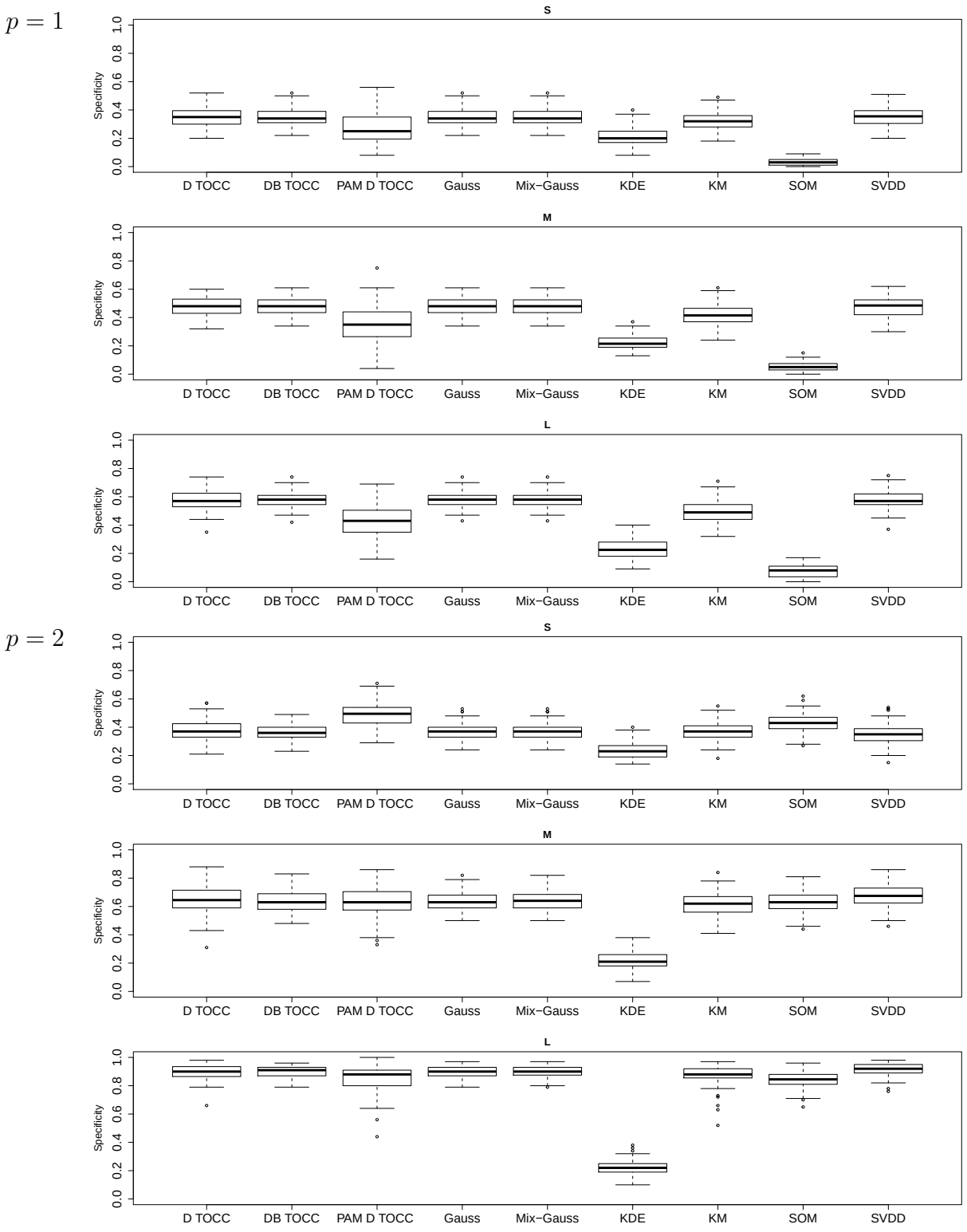


Figure C.6: Specificity for $s \geq 0.9$ sensitivity level for small (S), medium (M) and large (L) shifts for Model 3, $n = 200$.

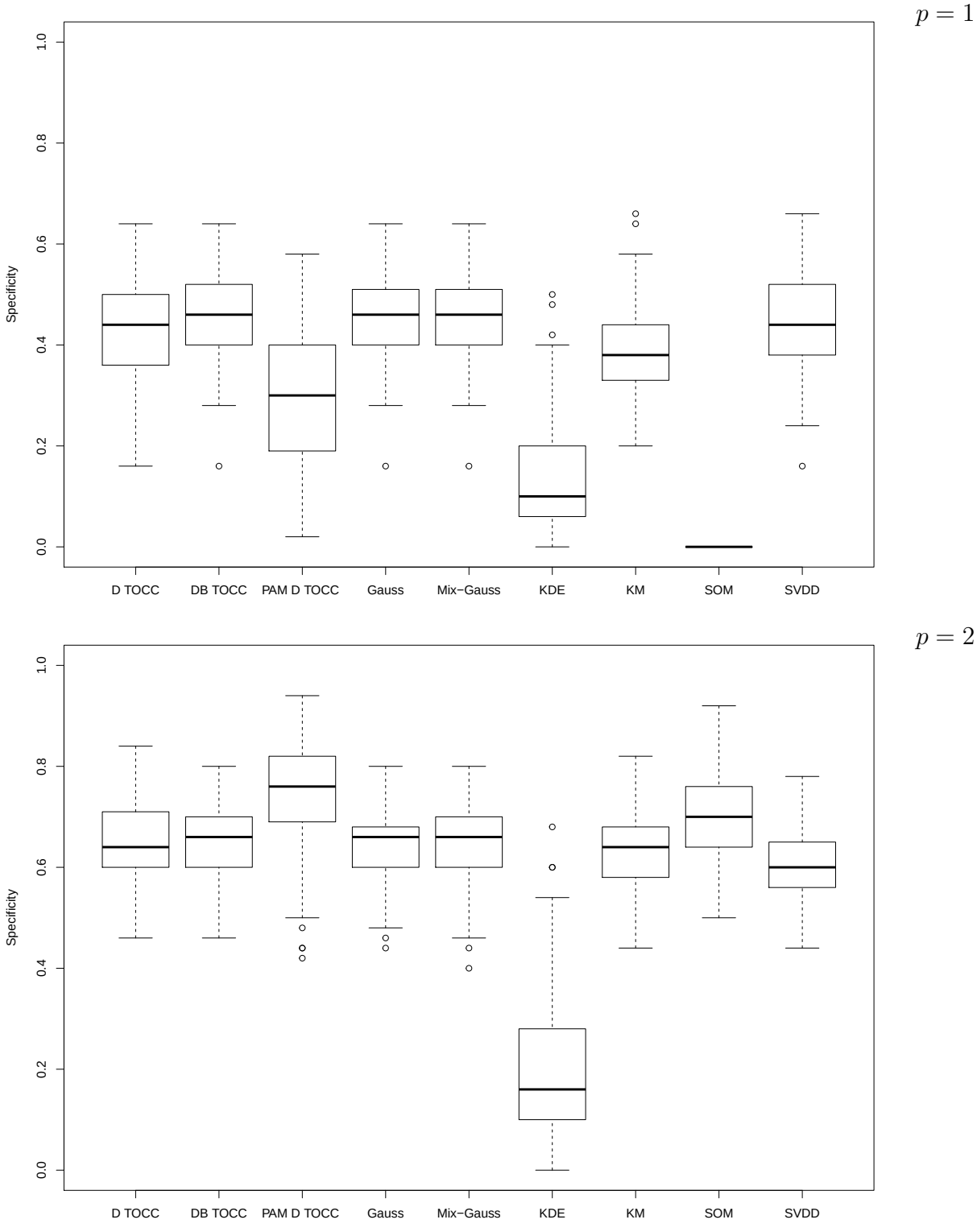
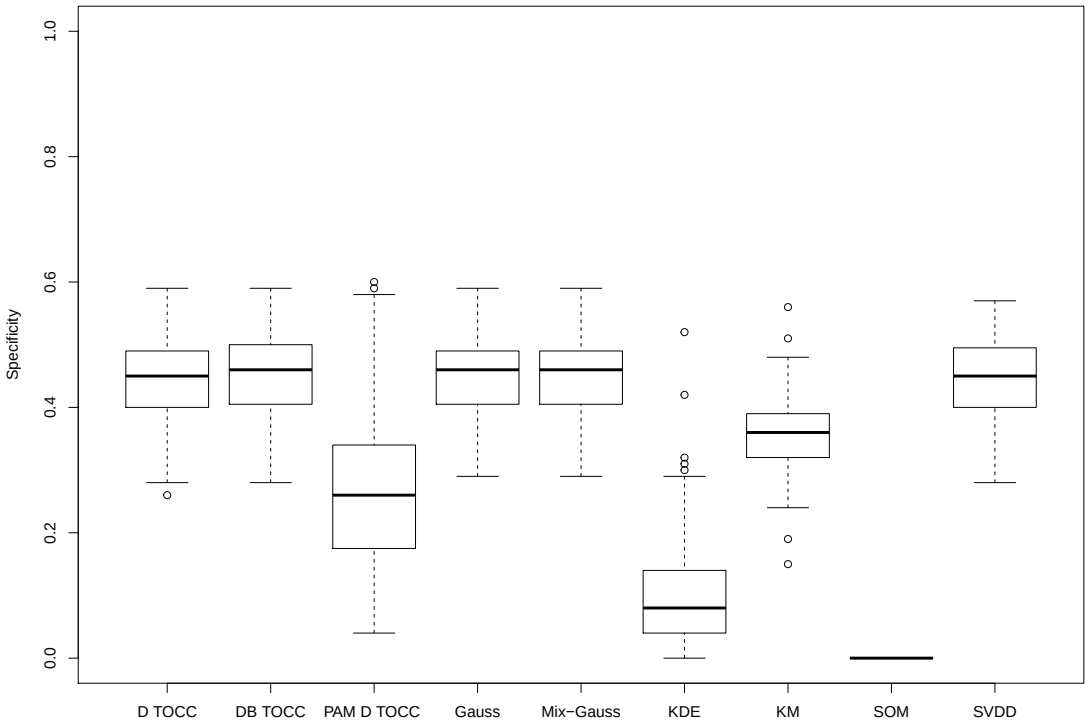


Figure C.7: Specificity for $s \geq 0.9$ sensitivity level for Model 4, $n = 100$.

$p = 1$



$p = 2$

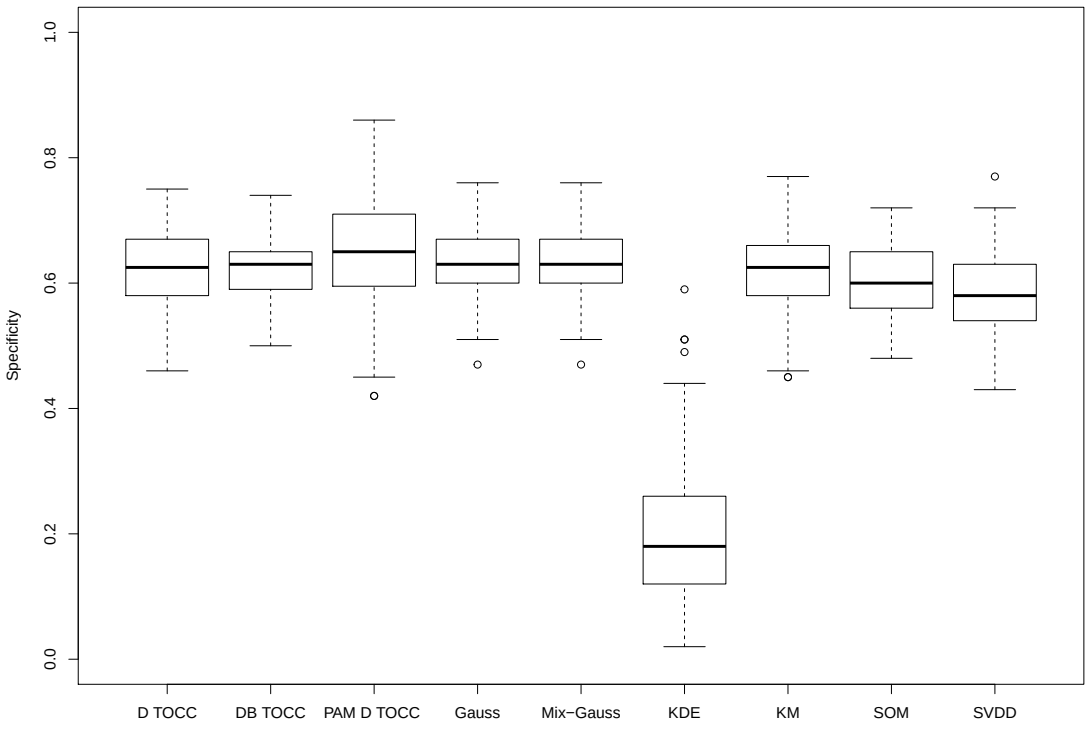


Figure C.8: Specificity for $s \geq 0.9$ sensitivity level for Model 4, $n = 200$.

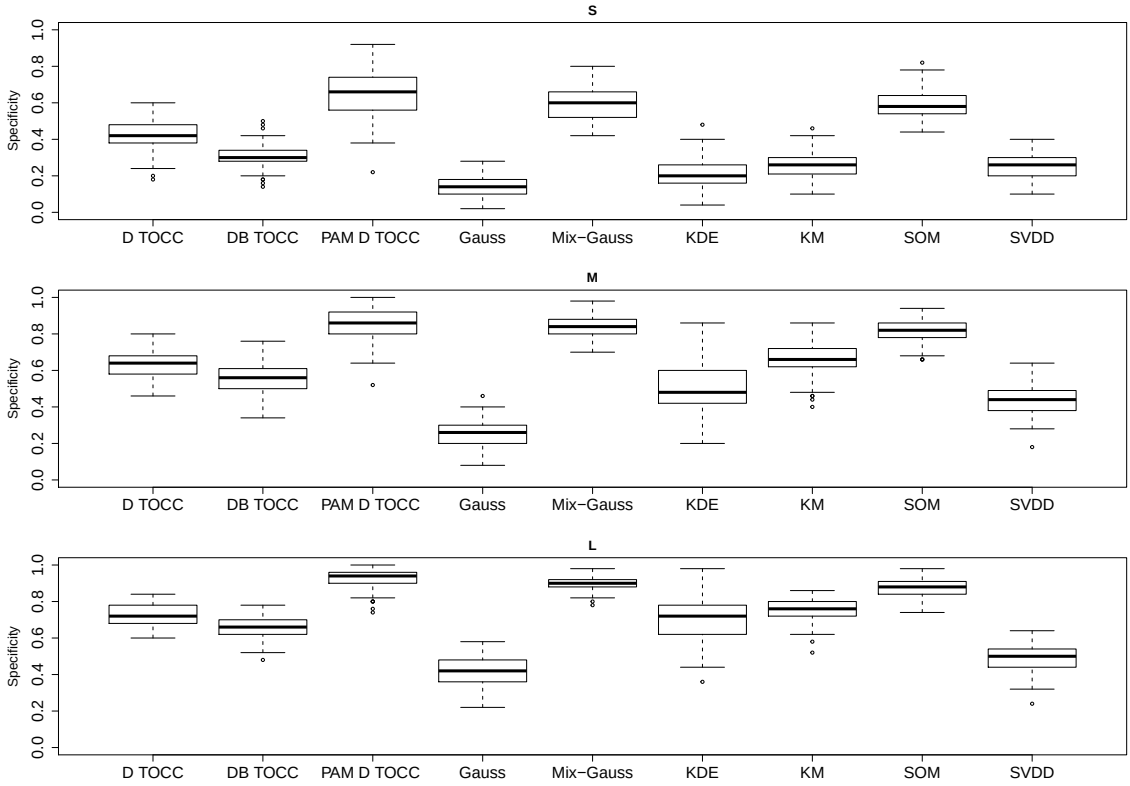


Figure C.9: Specificity for $s \geq 0.9$ sensitivity level for for small (S), medium (M) and large (L) shifts Model 5, $n = 100$

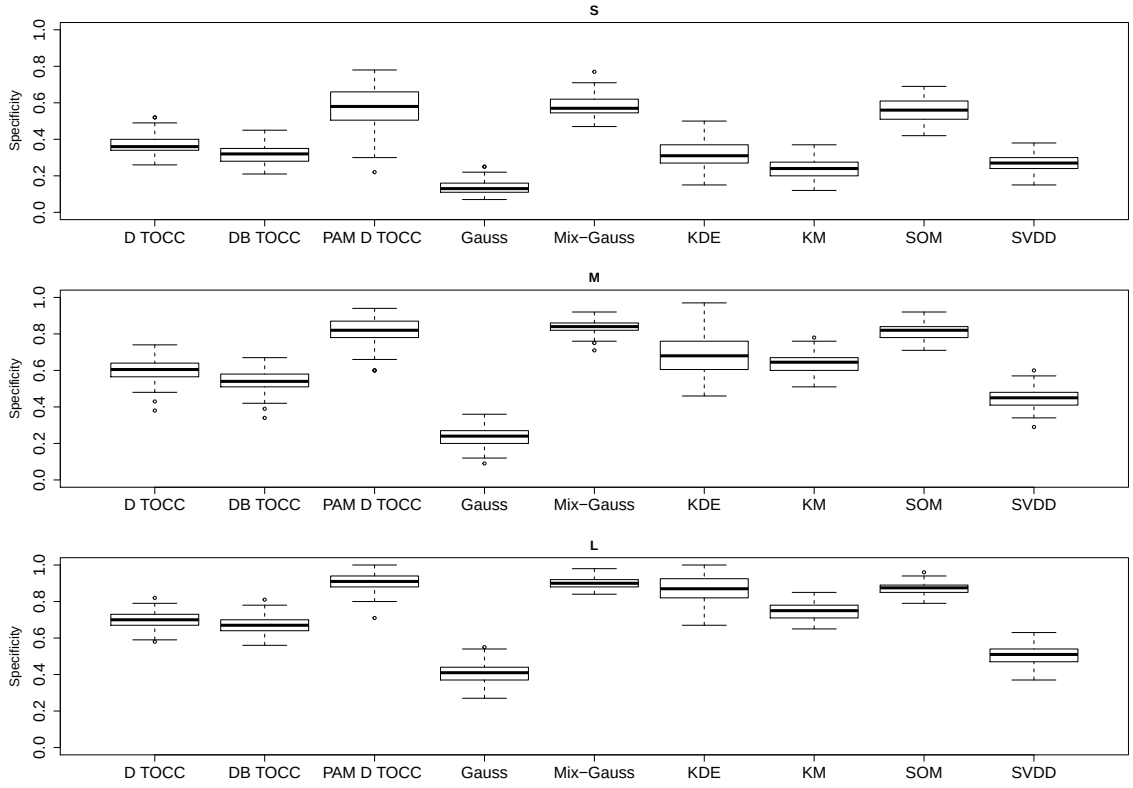


Figure C.10: Specificity for $s \geq 0.9$ sensitivity level for small (S), medium (M) and large (L) shifts for Model 5, $n = 200$.

Method	*shift*	Results for $d = 1$			Results for $d = 2$		
		n = 100	n = 200	n= 500	n = 100	n = 200	n = 500
D TOCC	S	$0.24_{0.08}$	$0.26_{0.06}$	$0.27_{0.04}$	$0.27_{0.09}$	$0.25_{0.07}$	$0.23_{0.05}$
	M	$0.38_{0.10}$	$0.40_{0.08}$	$0.41_{0.04}$	$0.56_{0.14}$	$0.57_{0.10}$	$0.55_{0.06}$
	L	$0.49_{0.12}$	$0.53_{0.08}$	$0.54_{0.05}$	$0.84_{0.12}$	$0.86_{0.07}$	$0.86_{0.04}$
DB TOCC	S	$0.25_{0.08}$	$0.26_{0.06}$	$0.27_{0.03}$	$0.25_{0.08}$	$0.23_{0.05}$	$0.21_{0.04}$
	M	$0.40_{0.09}$	$0.41_{0.07}$	$0.41_{0.04}$	$0.57_{0.11}$	$0.56_{0.07}$	$0.54_{0.05}$
	L	$0.52_{0.10}$	$0.53_{0.07}$	$0.54_{0.05}$	$0.86_{0.09}$	$0.86_{0.05}$	$0.86_{0.03}$
PAM D TOCC	S	$0.20_{0.10}$	$0.18_{0.08}$	$0.15_{0.05}$	$0.54_{0.13}$	$0.40_{0.08}$	$0.24_{0.05}$
	M	$0.27_{0.12}$	$0.25_{0.10}$	$0.22_{0.06}$	$0.70_{0.12}$	$0.57_{0.12}$	$0.40_{0.09}$
	L	$0.53_{0.14}$	$0.31_{0.11}$	$0.29_{0.07}$	$0.87_{0.09}$	$0.80_{0.11}$	$0.69_{0.10}$
Gaussian	S	$0.25_{0.08}$	$0.26_{0.06}$	$0.27_{0.03}$	$0.24_{0.07}$	$0.23_{0.05}$	$0.22_{0.04}$
	M	$0.40_{0.09}$	$0.41_{0.07}$	$0.41_{0.04}$	$0.57_{0.09}$	$0.56_{0.07}$	$0.54_{0.05}$
	L	$0.52_{0.10}$	$0.53_{0.07}$	$0.54_{0.05}$	$0.86_{0.07}$	$0.86_{0.04}$	$0.86_{0.03}$
Mix-Gauss	S	$0.26_{0.08}$	$0.26_{0.06}$	$0.27_{0.03}$	$0.25_{0.08}$	$0.23_{0.05}$	$0.22_{0.04}$
	M	$0.40_{0.09}$	$0.41_{0.07}$	$0.41_{0.04}$	$0.58_{0.09}$	$0.56_{0.07}$	$0.54_{0.05}$
	L	$0.52_{0.10}$	$0.53_{0.07}$	$0.54_{0.05}$	$0.87_{0.07}$	$0.86_{0.04}$	$0.86_{0.03}$
KDE	S	$0.10_{0.05}$	$0.10_{0.03}$	$0.10_{0.02}$	$0.06_{0.04}$	$0.08_{0.03}$	$0.09_{0.02}$
	M	$0.10_{0.05}$	$0.10_{0.04}$	$0.10_{0.03}$	$0.07_{0.05}$	$0.08_{0.04}$	$0.09_{0.02}$
	L	$0.09_{0.05}$	$0.10_{0.04}$	$0.10_{0.02}$	$0.07_{0.04}$	$0.08_{0.04}$	$0.09_{0.02}$
KM	S	$0.25_{0.09}$	$0.23_{0.07}$	$0.22_{0.04}$	$0.26_{0.08}$	$0.24_{0.06}$	$0.21_{0.04}$
	M	$0.34_{0.10}$	$0.31_{0.09}$	$0.33_{0.06}$	$0.55_{0.11}$	$0.53_{0.10}$	$0.51_{0.06}$
	L	$0.43_{0.11}$	$0.42_{0.10}$	$0.42_{0.07}$	$0.83_{0.10}$	$0.82_{0.08}$	$0.82_{0.04}$
SOM	S	$0.00_{0.01}$	$0.00_{0.00}$	$0.00_{0.00}$	$0.43_{0.09}$	$0.32_{0.06}$	$0.23_{0.04}$
	M	$0.00_{0.01}$	$0.00_{0.00}$	$0.00_{0.00}$	$0.66_{0.08}$	$0.55_{0.08}$	$0.47_{0.06}$
	L	$0.01_{0.02}$	$0.00_{0.01}$	$0.00_{0.00}$	$0.87_{0.07}$	$0.80_{0.07}$	$0.76_{0.06}$
SVDD	S	$0.25_{0.08}$	$0.26_{0.07}$	$0.27_{0.05}$	$0.26_{0.08}$	$0.25_{0.07}$	$0.24_{0.05}$
	M	$0.39_{0.10}$	$0.40_{0.08}$	$0.41_{0.06}$	$0.61_{0.09}$	$0.61_{0.08}$	$0.61_{0.06}$
	L	$0.51_{0.12}$	$0.53_{0.08}$	$0.54_{0.06}$	$0.90_{0.06}$	$0.89_{0.04}$	$0.89_{0.03}$

Table C.1: Specificity for $s \geq 0.9$ sensitivity level for Model 1.

Method	*shift*	Results for $d = 1$			Results for $d = 2$		
		n = 100	n = 200	n= 500	n = 100	n = 200	n = 500
	S	$0.19_{0.08}$	$0.20_{0.06}$	$0.20_{0.04}$	$0.21_{0.11}$	$0.17_{0.07}$	$0.16_{0.04}$
D TOCC	M	$0.27_{0.11}$	$0.29_{0.08}$	$0.29_{0.05}$	$0.41_{0.21}$	$0.38_{0.16}$	$0.33_{0.12}$
	L	$0.34_{0.12}$	$0.37_{0.09}$	$0.37_{0.06}$	$0.65_{0.27}$	$0.67_{0.21}$	$0.68_{0.15}$
	S	$0.20_{0.07}$	$0.21_{0.06}$	$0.21_{0.03}$	$0.16_{0.08}$	$0.15_{0.06}$	$0.14_{0.04}$
DB TOCC	M	$0.29_{0.10}$	$0.31_{0.07}$	$0.29_{0.05}$	$0.34_{0.19}$	$0.33_{0.15}$	$0.32_{0.10}$
	L	$0.37_{0.10}$	$0.38_{0.08}$	$0.38_{0.05}$	$0.62_{0.26}$	$0.66_{0.20}$	$0.67_{0.13}$
	S	$0.20_{0.11}$	$0.16_{0.07}$	$0.13_{0.03}$	$0.46_{0.13}$	$0.32_{0.10}$	$0.19_{0.05}$
PAM D TOCC	M	$0.24_{0.11}$	$0.18_{0.07}$	$0.14_{0.04}$	$0.54_{0.17}$	$0.34_{0.14}$	$0.17_{0.07}$
	L	$0.26_{0.12}$	$0.19_{0.07}$	$0.16_{0.05}$	$0.67_{0.23}$	$0.47_{0.22}$	$0.21_{0.13}$
	S	$0.20_{0.07}$	$0.21_{0.06}$	$0.21_{0.03}$	$0.15_{0.06}$	$0.14_{0.05}$	$0.14_{0.03}$
Gaussian	M	$0.28_{0.10}$	$0.31_{0.07}$	$0.29_{0.04}$	$0.31_{0.12}$	$0.30_{0.10}$	$0.29_{0.08}$
	L	$0.36_{0.10}$	$0.38_{0.08}$	$0.38_{0.05}$	$0.64_{0.17}$	$0.63_{0.15}$	$0.64_{0.12}$
	S	$0.20_{0.07}$	$0.21_{0.06}$	$0.21_{0.03}$	$0.16_{0.06}$	$0.14_{0.05}$	$0.14_{0.03}$
Mix-Gauss	M	$0.29_{0.10}$	$0.31_{0.06}$	$0.29_{0.04}$	$0.33_{0.12}$	$0.30_{0.09}$	$0.29_{0.07}$
	L	$0.37_{0.10}$	$0.39_{0.07}$	$0.38_{0.05}$	$0.68_{0.15}$	$0.65_{0.12}$	$0.65_{0.08}$
	S	$0.10_{0.05}$	$0.10_{0.03}$	$0.11_{0.03}$	$0.03_{0.04}$	$0.05_{0.04}$	$0.08_{0.02}$
KDE	M	$0.10_{0.05}$	$0.11_{0.02}$	$0.11_{0.03}$	$0.03_{0.05}$	$0.05_{0.05}$	$0.08_{0.02}$
	L	$0.11_{0.05}$	$0.11_{0.03}$	$0.11_{0.03}$	$0.03_{0.05}$	$0.05_{0.04}$	$0.08_{0.02}$
	S	$0.23_{0.08}$	$0.21_{0.06}$	$0.20_{0.04}$	$0.21_{0.07}$	$0.17_{0.05}$	$0.15_{0.04}$
KM	M	$0.31_{0.09}$	$0.29_{0.07}$	$0.27_{0.05}$	$0.40_{0.15}$	$0.36_{0.12}$	$0.32_{0.11}$
	L	$0.36_{0.10}$	$0.36_{0.08}$	$0.33_{0.05}$	$0.67_{0.20}$	$0.65_{0.18}$	$0.60_{0.16}$
	S	$0.01_{0.02}$	$0.01_{0.01}$	$0.00_{0.00}$	$0.44_{0.10}$	$0.29_{0.07}$	$0.20_{0.04}$
SOM	M	$0.01_{0.02}$	$0.01_{0.01}$	$0.00_{0.01}$	$0.65_{0.10}$	$0.49_{0.11}$	$0.38_{0.08}$
	L	$0.02_{0.04}$	$0.01_{0.02}$	$0.01_{0.01}$	$0.80_{0.09}$	$0.70_{0.10}$	$0.59_{0.11}$
	S	$0.18_{0.10}$	$0.19_{0.10}$	$0.20_{0.11}$	$0.16_{0.09}$	$0.16_{0.08}$	$0.15_{0.08}$
SVDD	M	$0.25_{0.14}$	$0.26_{0.14}$	$0.28_{0.15}$	$0.32_{0.19}$	$0.34_{0.20}$	$0.32_{0.21}$
	L	$0.32_{0.16}$	$0.33_{0.16}$	$0.36_{0.18}$	$0.60_{0.25}$	$0.61_{0.27}$	$0.55_{0.29}$

Table C.2: Specificity for $s \geq 0.9$ sensitivity level for Model 2.

Method	*shift*	Results for $d = 1$			Results for $d = 2$		
		n = 100	n = 200	n= 500	n = 100	n = 200	n = 500
D TOCC	S	$0.33_{0.09}$	$0.35_{0.06}$	$0.36_{0.04}$	$0.39_{0.09}$	$0.38_{0.07}$	$0.36_{0.05}$
	M	$0.46_{0.09}$	$0.48_{0.07}$	$0.48_{0.04}$	$0.64_{0.14}$	$0.65_{0.10}$	$0.64_{0.06}$
	L	$0.56_{0.10}$	$0.58_{0.06}$	$0.58_{0.04}$	$0.87_{0.09}$	$0.90_{0.05}$	$0.90_{0.03}$
DB TOCC	S	$0.35_{0.08}$	$0.35_{0.06}$	$0.36_{0.03}$	$0.37_{0.08}$	$0.36_{0.06}$	$0.35_{0.04}$
	M	$0.47_{0.08}$	$0.48_{0.06}$	$0.48_{0.04}$	$0.65_{0.11}$	$0.63_{0.08}$	$0.62_{0.05}$
	L	$0.58_{0.08}$	$0.58_{0.06}$	$0.58_{0.04}$	$0.89_{0.07}$	$0.90_{0.04}$	$0.89_{0.03}$
PAM D TOCC	S	$0.29_{0.12}$	$0.27_{0.11}$	$0.26_{0.09}$	$0.61_{0.12}$	$0.49_{0.08}$	$0.35_{0.06}$
	M	$0.36_{0.12}$	$0.35_{0.12}$	$0.34_{0.08}$	$0.74_{0.11}$	$0.63_{0.10}$	$0.48_{0.08}$
	L	$0.44_{0.14}$	$0.45_{0.11}$	$0.41_{0.08}$	$0.90_{0.07}$	$0.85_{0.09}$	$0.75_{0.09}$
Gaussian	S	$0.35_{0.08}$	$0.35_{0.06}$	$0.36_{0.03}$	$0.37_{0.07}$	$0.37_{0.06}$	$0.35_{0.04}$
	M	$0.47_{0.09}$	$0.48_{0.06}$	$0.48_{0.04}$	$0.65_{0.09}$	$0.64_{0.07}$	$0.63_{0.05}$
	L	$0.58_{0.08}$	$0.58_{0.06}$	$0.58_{0.04}$	$0.89_{0.05}$	$0.90_{0.04}$	$0.90_{0.03}$
Mix-Gauss	S	$0.35_{0.08}$	$0.35_{0.06}$	$0.36_{0.03}$	$0.38_{0.07}$	$0.37_{0.06}$	$0.35_{0.04}$
	M	$0.47_{0.09}$	$0.48_{0.06}$	$0.48_{0.04}$	$0.66_{0.10}$	$0.64_{0.07}$	$0.63_{0.05}$
	L	$0.58_{0.08}$	$0.58_{0.06}$	$0.58_{0.04}$	$0.89_{0.05}$	$0.90_{0.04}$	$0.90_{0.03}$
KDE	S	$0.21_{0.07}$	$0.21_{0.06}$	$0.22_{0.04}$	$0.21_{0.07}$	$0.23_{0.06}$	$0.25_{0.03}$
	M	$0.22_{0.08}$	$0.22_{0.05}$	$0.22_{0.04}$	$0.21_{0.07}$	$0.22_{0.06}$	$0.25_{0.03}$
	L	$0.22_{0.08}$	$0.23_{0.07}$	$0.24_{0.05}$	$0.20_{0.07}$	$0.22_{0.05}$	$0.25_{0.04}$
KM	S	$0.34_{0.10}$	$0.32_{0.06}$	$0.31_{0.05}$	$0.39_{0.08}$	$0.37_{0.07}$	$0.35_{0.04}$
	M	$0.43_{0.09}$	$0.42_{0.07}$	$0.42_{0.05}$	$0.63_{0.12}$	$0.61_{0.09}$	$0.60_{0.06}$
	L	$0.52_{0.10}$	$0.50_{0.07}$	$0.50_{0.05}$	$0.87_{0.07}$	$0.87_{0.07}$	$0.87_{0.04}$
SOM	S	$0.04_{0.04}$	$0.03_{0.02}$	$0.03_{0.02}$	$0.52_{0.08}$	$0.43_{0.06}$	$0.35_{0.04}$
	M	$0.07_{0.05}$	$0.05_{0.03}$	$0.05_{0.03}$	$0.72_{0.08}$	$0.63_{0.07}$	$0.56_{0.06}$
	L	$0.09_{0.07}$	$0.08_{0.04}$	$0.07_{0.04}$	$0.89_{0.06}$	$0.84_{0.06}$	$0.80_{0.06}$
SVDD	S	$0.34_{0.08}$	$0.35_{0.06}$	$0.36_{0.04}$	$0.36_{0.08}$	$0.35_{0.07}$	$0.35_{0.05}$
	M	$0.46_{0.09}$	$0.48_{0.07}$	$0.48_{0.06}$	$0.68_{0.09}$	$0.68_{0.08}$	$0.68_{0.06}$
	L	$0.57_{0.10}$	$0.58_{0.06}$	$0.58_{0.05}$	$0.91_{0.05}$	$0.92_{0.04}$	$0.82_{0.03}$

Table C.3: Specificity for $s \geq 0.9$ sensitivity level for Model 3.

Method	Results for $d = 1$			Results for $d = 2$		
	n = 100	n = 200	n= 500	n = 100	n = 200	n = 500
D TOCC	$0.44_{0.09}$	$0.44_{0.07}$	$0.45_{0.04}$	$0.65_{0.08}$	$0.62_{0.06}$	$0.62_{0.04}$
DB TOCC	$0.45_{0.09}$	$0.45_{0.06}$	$0.45_{0.04}$	$0.64_{0.08}$	$0.62_{0.06}$	$0.61_{0.04}$
PAM D TOCC	$0.29_{0.14}$	$0.27_{0.13}$	$0.26_{0.09}$	$0.75_{0.12}$	$0.65_{0.09}$	$0.51_{0.09}$
Gaussian	$0.45_{0.09}$	$0.45_{0.06}$	$0.45_{0.04}$	$0.65_{0.07}$	$0.63_{0.05}$	$0.62_{0.04}$
Mix-Gauss	$0.45_{0.09}$	$0.45_{0.06}$	$0.45_{0.04}$	$0.65_{0.07}$	$0.63_{0.05}$	$0.62_{0.04}$
KDE	$0.13_{0.11}$	$0.10_{0.09}$	$0.05_{0.04}$	$0.20_{0.14}$	$0.20_{0.11}$	$0.17_{0.07}$
KM	$0.39_{0.09}$	$0.36_{0.06}$	$0.36_{0.04}$	$0.64_{0.08}$	$0.62_{0.06}$	$0.60_{0.04}$
SOM	$0.00_{0.00}$	$0.00_{0.00}$	$0.00_{0.00}$	$0.70_{0.08}$	$0.60_{0.06}$	$0.53_{0.04}$
SVDD	$0.45_{0.09}$	$0.45_{0.07}$	$0.45_{0.04}$	$0.60_{0.07}$	$0.58_{0.07}$	$0.58_{0.04}$

Table C.4: Specificity for $s \geq 0.9$ sensitivity level for Model 4.

Method	*shift*	Results for $d = 2$		
		n = 100	n = 200	n= 500
D TOCC	S	$0.42_{0.08}$	$0.37_{0.05}$	$0.35_{0.03}$
	M	$0.63_{0.07}$	$0.60_{0.06}$	$0.57_{0.04}$
	L	$0.72_{0.06}$	$0.70_{0.05}$	$0.68_{0.03}$
DB TOCC	S	$0.31_{0.07}$	$0.32_{0.05}$	$0.32_{0.03}$
	M	$0.55_{0.08}$	$0.55_{0.06}$	$0.55_{0.03}$
	L	$0.66_{0.07}$	$0.67_{0.05}$	$0.67_{0.03}$
PAM D TOCC	S	$0.65_{0.14}$	$0.57_{0.11}$	$0.52_{0.07}$
	M	$0.86_{0.09}$	$0.81_{0.07}$	$0.77_{0.05}$
	L	$0.93_{0.06}$	$0.91_{0.05}$	$0.90_{0.03}$
Gaussian	S	$0.14_{0.06}$	$0.13_{0.04}$	$0.13_{0.02}$
	M	$0.25_{0.07}$	$0.24_{0.05}$	$0.24_{0.03}$
	L	$0.41_{0.08}$	$0.40_{0.06}$	$0.41_{0.03}$
Mix-Gauss	S	$0.59_{0.09}$	$0.58_{0.06}$	$0.58_{0.04}$
	M	$0.84_{0.06}$	$0.84_{0.04}$	$0.83_{0.02}$
	L	$0.90_{0.04}$	$0.90_{0.03}$	$0.90_{0.02}$
KDE	S	$0.32_{0.07}$	$0.32_{0.07}$	$0.40_{0.07}$
	M	$0.69_{0.11}$	$0.69_{0.11}$	$0.85_{0.07}$
	L	$0.87_{0.07}$	$0.87_{0.07}$	$0.97_{0.03}$
KM	S	$0.26_{0.07}$	$0.24_{0.05}$	$0.22_{0.03}$
	M	$0.66_{0.09}$	$0.64_{0.05}$	$0.63_{0.03}$
	L	$0.75_{0.06}$	$0.75_{0.04}$	$0.74_{0.03}$
SOM	S	$0.59_{0.08}$	$0.56_{0.06}$	$0.53_{0.05}$
	M	$0.82_{0.06}$	$0.81_{0.04}$	$0.80_{0.03}$
	L	$0.87_{0.05}$	$0.87_{0.03}$	$0.87_{0.03}$
SVDD	S	$0.25_{0.06}$	$0.27_{0.05}$	$0.28_{0.04}$
	M	$0.44_{0.08}$	$0.45_{0.05}$	$0.45_{0.03}$
	L	$0.49_{0.07}$	$0.50_{0.05}$	$0.51_{0.03}$

Table C.5: Specificity for $s \geq 0.9$ sensitivity level for Model 5.

Appendix D

D.1 R Functions

```
banana2d=function(N,s){
  # N is the number of sample units to generate
  # s is the banana angle

  r=5
  p=c(0.5, 0.5)
  #N=c(50,50)
  #s=1

  domaina=0.125*pi+runif(N[1])*1.25*pi
  B=cbind(runif(N[1]),runif(N[1]))*s
  A=cbind(r*sin(domaina),r*cos(domaina))
  a=A+B

  domainb=0.375*pi-runif(N[2])*1.25*pi
  B2=cbind(runif(N[2]),runif(N[2]))*s
  A2=cbind(r*sin(domainb),r*cos(domainb))
  C2=cbind(rep(-0.75*r,times=N[2]),rep(-0.75*r,times=N[2]))
  a2=A2+B2+C2

  aa=rbind(a,a2)

  return(aa)
}
```

Bibliography

[1] Dimitris Achlioptas. "Database-friendly random projections: Johnson-Lindenstrauss with binary coins". In: *Journal of computer and System Sciences* 66.4 (2003), pp. 671–687.

[2] Hongshik Ahn, Hojin Moon, Melissa J Fazzari, Noha Lim, James J Chen, and Ralph L Kodell. "Classification by ensembles from random partitions of high-dimensional data". In: *Computational Statistics & Data Analysis* 51.12 (2007), pp. 6166–6179.

[3] Jeongyoun Ahn. "High dimension, low sample size data analysis". PhD thesis. University of North Carolina at Chapel Hill, 2006.

[4] Patricia M. E. Altham. "Two Generalizations of the Binomial Distribution". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 27.2 (1978), pp. 162–167. ISSN: 00359254, 14679876.

[5] Carlos Alberto de Araújo Padilha, Dante Augusto Couto Barone, and Adrião Duarte Dória Neto. "A multi-level approach using genetic algorithms in an ensemble of Least Squares Support Vector Machines". In: *Knowledge-Based Systems* 106 (2016), pp. 85–95.

[6] Bart Bakker and Tom Heskes. "Clustering ensembles of neural network models". In: *Neural networks* 16.2 (2003), pp. 261–269.

[7] Richard E Bellman. *Adaptive control processes: a guided tour*. Princeton university press, 2015.

[8] Vasudha Bhatnagar, Manju Bhardwaj, Shivam Sharma, and Sufyan Haroon. "Accuracy–diversity based pruning of classifier ensembles". In: *Progress in Artificial Intelligence* 2.2-3 (2014), pp. 97–111.

[9] Urvesh Bhowan, Mark Johnston, Mengjie Zhang, and Xin Yao. "Evolving diverse ensembles using genetic programming for classification with unbalanced data". In: *IEEE Transactions on Evolutionary Computation* 17.3 (2013), pp. 368–386.

[10] Urvesh Bhowan, Mark Johnston, Mengjie Zhang, and Xin Yao. "Reusing genetic programming for ensemble selection in classification of unbalanced data". In: *IEEE Transactions on Evolutionary Computation* 18.6 (2014), pp. 893–908.

[11] Peter J Bickel and Elizaveta Levina. "Some theory for Fisher's linear discriminant function,'naive Bayes', and some alternatives when there are many more variables than observations". In: *Bernoulli* (2004), pp. 989–1010.

[12] Ella Bingham and Heikki Mannila. "Random projection in dimensionality reduction: applications to image and text data". In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2001, pp. 245–250.

[13] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[14] Christopher M Bishop, Markus Svensén, and Christopher KI Williams. "GTM: The generative topographic mapping". In: *Neural computation* 10.1 (1998), pp. 215–234.

[15] Leo Breiman. "Bagging predictors". In: *Machine learning* 24.2 (1996), pp. 123–140.

[16] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[17] Leo Breiman, A Cutler, A Liaw, and M Wiener. *Package randomForest*. 2015.

[18] Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

[19] Timothy I. Cannings and Richard J. Samworth. *Package RPEnsemble*. 2016.

[20] Timothy I Cannings and Richard J Samworth. "Random-projection ensemble classification". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79.4 (2017), pp. 959–1035.

[21] Gail A Carpenter, Stephen Grossberg, and David B Rosen. "ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition". In: *Neural networks* 4.4 (1991), pp. 493–504.

[22] Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. "Ensemble selection from libraries of models". In: *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 18.

[23] George DC Cavalcanti, Luiz S Oliveira, Thiago JM Moura, and Guilherme V Carvalho. "Combining diversity measures for ensemble pruning". In: *Pattern Recognition Letters* 74 (2016), pp. 38–45.

[24] Huanhuan Chen, Peter Tiňo, and Xin Yao. "Predictive ensemble pruning by expectation propagation". In: *IEEE Transactions on Knowledge and Data Engineering* 21.7 (2009), pp. 999–1013.

[25] Yixin Chen, Xin Dang, Hanxiang Peng, and Henry L Bart. "Outlier detection with the kernelized spatial depth function". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.2 (2009), pp. 288–305.

[26] Kevin J Cherkauer. "Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks". In: *Working notes of the AAAI workshop on integrating multiple learned models*. Citeseer. 1996, pp. 15–21.

[27] Maurice Coyle and Barry Smyth. "On the use of selective ensembles for relevance classification in case-based web search". In: *European Conference on Case-Based Reasoning*. Springer. 2006, pp. 370–384.

[28] Camillo Dagum. "Transvariazione fra più di due distribuzioni". In: *Gini, C.(ed.) Memorie di metodologia statistica* 2 (1959).

[29] Xin Dang and Robert Serfling. "Nonparametric depth-based multivariate outlier identifiers, and robustness properties". In: *submitted for journal publication* (2006).

[30] Belur V Dasarathy and Belur V Sheela. "A composite classifier system design: concepts and methodology". In: *Proceedings of the IEEE* 67.5 (1979), pp. 708–713.

[31] Thomas G Dietterich. "Ensemble methods in machine learning". In: *International workshop on multiple classifier systems*. Springer. 2000, pp. 1–15.

[32] Thomas G. Dietterich and Ghulum Bakiri. "Solving multiclass learning problems via error-correcting output codes". In: *Journal of artificial intelligence research* 2 (1995), pp. 263–286.

[33] Carlos AR Diniz, Marcelo H Tutia, and Jose G Leite. "Bayesian analysis of a correlated binomial model". In: *Brazilian Journal of Probability and Statistics* (2010), pp. 68–77.

[34] David L Donoho and Miriam Gasko. "Breakdown properties of location estimates based on halfspace depth and projected outlyingness". In: *The Annals of Statistics* (1992), pp. 1803–1827.

[35] Gerard Downey, Vanessa Fouratier, and J Daniel Kelly. "Detection of honey adulteration by addition of fructose and glucose using near infrared transflectance spectroscopy". In: *Journal of Near Infrared Spectroscopy* 11.6 (2004), pp. 447–456.

[36] Gerard Downey, Peter McIntyre, and Antony N Davies. "Detecting and quantifying sunflower oil adulteration in extra virgin olive oils from the Eastern Mediterranean by visible and near-infrared spectroscopy". In: *Journal of Agricultural and Food chemistry* 50.20 (2002), pp. 5520–5525.

[37] Sandrine Dudoit, Jane Fridlyand, and Terence P Speed. "Comparison of discrimination methods for the classification of tumors using gene expression data". In: *Journal of the American statistical association* 97.457 (2002), pp. 77–87.

[38] Jianqing Fan and Yingying Fan. "High dimensional classification using features annealed independence rules". In: *Annals of statistics* 36.6 (2008), p. 2605.

[39] Ronald A Fisher. "The use of multiple measurements in taxonomic problems". In: *Annals of eugenics* 7.2 (1936), pp. 179–188.

[40] Francesca Fortunato. "Limit theorems for the Multiplicative Binomial Distribution (MBD)". In: *arXiv.org* (2017).

[41] Fabricio Olivetti de França, Fernando J Von Zuben, and Leandro Nunes de Castro. "An artificial immune network for multimodal function optimization on dynamic environments". In: *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. ACM. 2005, pp. 289–296.

[42] Yoav Freund and Robert E Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". In: *European conference on computational learning theory*. Springer. 1995, pp. 23–37.

[43] Jerome Friedman. *The elements of statistical learning*. Springer-Verlag. 2013.

[44] Jerome H Friedman. "Regularized discriminant analysis". In: *Journal of the American statistical association* 84.405 (1989), pp. 165–175.

[45] Stuart Geman, Elie Bienenstock, and René Doursat. "Neural networks and the bias/variance dilemma". In: *Neural computation* 4.1 (1992), pp. 1–58.

[46] Giorgio Giacinto and Fabio Roli. "An approach to the automatic design of multiple classifier systems". In: *Pattern recognition letters* 22.1 (2001), pp. 25–33.

[47] Giorgio Giacinto and Fabio Roli. "Design of effective neural network ensembles for image classification purposes". In: *Image and Vision Computing* 19.9 (2001), pp. 699–707.

[48] Corrado Gini. *Il Concetto di "transvariazione" e le sue prime applicazioni*. Athenaeum, 1916.

[49] Corrado Gini and Gregorio Livada. *Transvariazione a più dimensioni*. Paneto & Petrelli, 1943.

[50] Pablo M Granitto, Pablo F Verdes, and H Alejandro Ceccatto. "Neural network ensembles: evaluation of aggregation algorithms". In: *Artificial Intelligence* 163.2 (2005), pp. 139–162.

[51] Isabelle Guyon and André Elisseeff. "An introduction to variable and feature selection". In: *Journal of machine learning research* 3.Mar (2003), pp. 1157–1182.

[52] Lars Kai Hansen and Peter Salamon. "Neural network ensembles". In: *IEEE transactions on pattern analysis and machine intelligence* 12.10 (1990), pp. 993–1001.

[53] Trevor Hastie, Andreas Buja, and Robert Tibshirani. "Penalized discriminant analysis". In: *The Annals of Statistics* (1995), pp. 73–102.

[54] Trevor Hastie and Werner Stuetzle. "Principal curves". In: *Journal of the American Statistical Association* 84.406 (1989), pp. 502–516.

[55] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

[56] Daniel Hernández-Lobato, José Miguel Hernández-Lobato, Rubén Ruiz-Torrubiano, and Ángel Valle. "Pruning adaptive boosting ensembles by means of a genetic algorithm". In: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer. 2006, pp. 322–329.

[57] Leidys Cabrera Hernández, Alejandro Morales Hernández, Gladys M Casas Cardoso, and Yailen Martínez Jiménez. "Genetic algorithms with diversity measures to build classifier systems". In: *Investigación Operacional* 36.3 (2015), pp. 206–225.

[58] John H Holland. "Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence". In: *Ann Arbor, MI: University of Michigan Press* (1975).

[59] Sigve Hovda. "Using pseudometrics in kernel density estimation". In: *Journal of Nonparametric Statistics* 26.4 (2014), pp. 669–696.

[60] Qinghua Hu, Daren Yu, Zongxia Xie, and Xiaodong Li. "EROS: Ensemble rough subspaces". In: *Pattern recognition* 40.12 (2007), pp. 3728–3739.

[61] William B Johnson and Joram Lindenstrauss. "Extensions of Lipschitz mappings into a Hilbert space". In: *Contemporary mathematics* 26.189-206 (1984), p. 1.

[62] Iain M Johnstone and D Michael Titterington. *Statistical challenges of high-dimensional data*. 2009.

[63] Joseph B Kadane et al. "Sums of Possibly Associated Bernoulli Variables: The Conway–Maxwell-Binomial Distribution". In: *Bayesian Analysis* 11.2 (2016), pp. 403–420.

[64] Mark Kaminski. "Central limit theorem for certain classes of dependent random variables". In: *Theory of Probability & Its Applications* 51.2 (2007), pp. 335–342.

[65] J Daniel Kelly, Cristina Petisco, and Gerard Downey. "Potential of near infrared transflectance spectroscopy to detect adulteration of Irish honey by beet invert syrup and high fructose corn syrup". In: *Journal of Near infrared spectroscopy* 14.2 (2006), pp. 139–146.

[66] Albert Hung-Ren Ko, Robert Sabourin, and Alceu de Souza Britto. "Compound diversity functions for ensemble selection". In: *International Journal of Pattern Recognition and Artificial Intelligence* 23.04 (2009), pp. 659–686.

[67] Ron Kohavi and George H John. "Wrappers for feature subset selection". In: *Artificial intelligence* 97.1-2 (1997), pp. 273–324.

[68] Teuvo Kohonen. "The self-organizing map". In: *Neurocomputing* 21.1-3 (1998), pp. 1–6.

[69] Mark A Kramer. "Nonlinear principal component analysis using autoassociative neural networks". In: *AIChE journal* 37.2 (1991), pp. 233–243.

[70] Anders Krogh, Jesper Vedelsby, et al. "Neural network ensembles, cross validation, and active learning". In: *Advances in neural information processing systems* 7 (1995), pp. 231–238.

[71] Wojtek Krzanowski, Philip Jonathan, WV McCarthy, and MR Thomas. "Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data". In: *Applied statistics* (1995), pp. 101–115.

[72] Ludmila I Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2004.

[73] Ludmila I Kuncheva and Christopher J Whitaker. "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy". In: *Machine learning* 51.2 (2003), pp. 181–207.

[74] Louisa Lam and Ching Y Suen. "Optimal combinations of pattern classifiers". In: *Pattern Recognition Letters* 16.9 (1995), pp. 945–954.

[75] Aleksandar Lazarevic and Zoran Obradovic. "Effective pruning of neural network classifier ensembles". In: *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on*. Vol. 2. IEEE. 2001, pp. 796–801.

[76] Moshe Lichman. *UCI Machine Learning Repository*. 2013.

[77] Regina Y Liu, Jesse M Parelius, Kesar Singh, et al. "Multivariate analysis by data depth: descriptive statistics, graphics and inference,(with discussion and a rejoinder by Liu and Singh)". In: *The annals of statistics* 27.3 (1999), pp. 783–858.

[78] Gianfranco Lovison. "An alternative representation of Altham's multiplicative-binomial distribution". In: *Statistics & Probability Letters* 36.4 (1998), pp. 415–420.

[79] Alberto Luceño. "A family of partially correlated Poisson models for overdispersion". In: *Computational statistics & data analysis* 20.5 (1995), pp. 511–520.

[80] Dragos D Margineantu and Thomas G Dietterich. "Pruning adaptive boosting". In: *ICML*. Vol. 97. 1997, pp. 211–218.

[81] Cathy Maugis, Gilles Celeux, and Marie-Laure Martin-Magniette. "Variable selection for clustering with Gaussian mixture models". In: *Biometrics* 65.3 (2009), pp. 701–709.

[82] Geoffrey McLachlan, Kim-Anh Do, and Christophe Ambroise. *Analyzing microarray gene expression data*. Vol. 422. John Wiley & Sons, 2005.

[83] Charles E Metz. "Basic principles of ROC analysis". In: *Seminars in nuclear medicine*. Vol. 8. 4. Elsevier. 1978, pp. 283–298.

[84] Angela Montanari. "Linear discriminant analysis and transvariation". In: *Journal of Classification* 21.1 (2004), pp. 71–88.

[85] Angela Montanari and Laura Lizzani. "A projection pursuit approach to variable selection". In: *Computational statistics & data analysis* 35.4 (2001), pp. 463–473.

[86] Mary M Moya and Don R Hush. "Network constraints and multi-objective optimization for one-class classification". In: *Neural Networks* 9.3 (1996), pp. 463–474.

[87] Thomas Brendan Murphy, Nema Dean, and Adrian E Raftery. "Variable selection and updating in model-based discriminant analysis for high dimensional data with food authenticity applications". In: *The annals of applied statistics* 4.1 (2010), p. 396.

[88] Carlos Padilha, Adrião D Dória Neto, and Jorge D Melo. "RSGALS-SVM: random subspace method applied to a LS-SVM ensemble optimized by genetic algorithm". In: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer. 2012, pp. 253–260.

[89] Christos H Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. "Latent semantic indexing: A probabilistic analysis". In: *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*. ACM. 1998, pp. 159–168.

[90] Bambang Parmanto, Paul W Munro, and Howard R Doyle. "Improving committee diagnosis with resampling techniques". In: *Advances in neural information processing systems* (1996), pp. 882–888.

[91] Karl Pearson. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572.

[92] Ross L Prentice. "Binary Regression Using an Extended Beta-Binomial Distribution, with Discussion of Correlation Induced by Covariate Measurement Errors". In: *Journal of the American Statistical Association* 81.394 (1986), pp. 321–327. eprint: http://www.tandfonline.com/doi/pdf/10.1080/01621459.1986.10478275.

[93] Lior Rokach. "Collective-agreement-based pruning of ensembles". In: *Computational Statistics & Data Analysis* 53.4 (2009), pp. 1015–1026.

[94] Lior Rokach. "Ensemble-based classifiers". In: *Artificial Intelligence Review* 33.1-2 (2010), pp. 1–39.

[95] Lior Rokach. "Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography". In: *Computational Statistics & Data Analysis* 53.12 (2009), pp. 4046–4072.

[96] Dymitr Ruta and Bogdan Gabrys. "Analysis of the correlation between majority voting error and the diversity measures in multiple classifier systems". In: *Proceedings of the 2nd International Workshop on Multiple Classifier Systems, Cambridge, UK*. Vol. Lecture Notes in Computer Science. LNCS 2096. Springer-Verlag, 2001, pp. 399–408.

[97] Dymitr Ruta and Bogdan Gabrys. "Classifier selection for majority voting". In: *Information fusion* 6.1 (2005), pp. 63–81.

[98] Ida Ruts and Peter J Rousseeuw. "Computing depth contours of bivariate point clouds". In: *Computational Statistics & Data Analysis* 23.1 (1996), pp. 153–168.

[99] Matteo Sartori. "Model-based classification methods for food authentication". MA thesis. University of Bologna, 2014.

[100] Bernhard Schölkopf, Peter L Bartlett, Alex J Smola, and Robert C Williamson. "Shrinking the tube: a new support vector regression algorithm". In: *Advances in neural information processing systems*. 1999, pp. 330–336.

[101] Luca Scrucca. *Package GA*. 2016.

[102] Luca Scrucca and Adrian E Raftery. "clustvarsel: A package implementing variable selection for model-based clustering in R". In: *arXiv preprint arXiv:1411.0606* (2014).

[103] Hyung Wong Shin and So Young Sohn. "Selected tree classifier combination based on both accuracy and error diversity". In: *Pattern recognition* 38.2 (2005), pp. 191–197.

[104] Catherine A Shipp and Ludmila I Kuncheva. "Relationships between combination methods and measures of diversity in combining classifiers". In: *Information fusion* 3.2 (2002), pp. 135–148.

[105] Palanisamy Shunmugapriya and S Kanmani. "Optimization of stacking ensemble configurations through Artificial Bee Colony algorithm". In: *Swarm and Evolutionary Computation* 12 (2013), pp. 24–32.

[106] John Gordon Skellam. "A probability distribution derived from the binomial distribution by regarding the probability of success as variable between the sets of trials". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 10.2 (1948), pp. 257–261.

[107] Funk Spencer. *svdd package*. GitHub.

[108] Gilbert Strang. *Linear algebra and its applications*. Ed. by Academic. 3rd. Brooks Cole, 1988.

[109] Christino Tamon and Jie Xiang. "On the boosting pruning problem". In: *European Conference on Machine Learning*. Springer. 2000, pp. 404–412.

[110] KE Tang, Ponnuthurai N. Suganthan, and Xin Yao. "An analysis of diversity measures". In: *Machine Learning* 65.1 (2006), pp. 247–271. ISSN: 1573-0565.

[111] David Martinus Johannes Tax. "One-class classification". PhD thesis. Delft University of Technology, 2001.

[112] David MJ Tax and Robert PW Duin. "Support vector data description". In: *Machine learning* 54.1 (2004), pp. 45–66.

[113] David MJ Tax and Klaus-Robert Müller. "Feature extraction for one-class classification". In: *Lecture notes in computer science* (2003), pp. 342–349.

[114] Robert Tibshirani. "Principal curves revisited". In: *Statistics and computing* 2.4 (1992), pp. 183–190.

[115] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), pp. 267–288.

[116] Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. "Class prediction by nearest shrunken centroids, with applications to DNA microarrays". In: *Statistical Science* (2003), pp. 104–117.

[117] Michael E Tipping and Christopher M Bishop. "Mixtures of probabilistic principal component analyzers". In: *Neural computation* 11.2 (1999), pp. 443–482.

[118] Michael E Tipping and Christopher M Bishop. "Probabilistic principal component analysis". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622.

[119] Grigorios Tsoumakas, Ioannis Partalas, and Ioannis Vlahavas. "An ensemble pruning primer". In: *Applications of supervised and unsupervised ensemble methods*. Springer, 2009, pp. 1–13.

[120] J. W. TUKEY. "Mathematics and the Picturing of Data". In: *Proceedings of the International Congress of Mathematicians, Vancouver, 1975* 2 (1975), pp. 523–531.

[121] John W Tukey. *Exploratory data analysis*. Reading, Mass., 1977.

[122] Kagan Tumer and Joydeep Ghosh. "Error correlation and error reduction in ensemble classifiers". In: *Connection science* 8.3-4 (1996), pp. 385–404.

[123] Eugene Tuv, Alexander Borisov, George Runger, and Kari Torkkola. "Feature selection with ensembles, artificial variables, and redundancy elimination". In: *Journal of Machine Learning Research* 10.Jul (2009), pp. 1341–1366.

[124] Vladimir Vapnik, Steven E Golowich, and Alex J Smola. "Support vector method for function approximation, regression estimation and signal processing". In: *Advances in neural information processing systems*. 1997, pp. 281–287.

[125] Andrew R Webb. "An approach to non-linear principal components analysis using radially symmetric kernel functions". In: *Statistics and computing* 6.2 (1996), pp. 159–168.

[126] Ping Xu, Guy N Brock, and Rudolph S Parrish. "Modified linear discriminant analysis approaches for classification of high-dimensional microarray data". In: *Computational Statistics & Data Analysis* 53.5 (2009), pp. 1674–1687.

[127] Xiaowei Xue, Min Yao, Zhaohui Wu, and Jianhua Yang. "Genetic ensemble of extreme learning machine". In: *Neurocomputing* 129 (2014), pp. 175–184.

[128] Lue Ping Zhao and Ross L Prentice. "Correlated binary regression using a quadratic exponential model". In: *Biometrika* 77.3 (1990), pp. 642–648.

[129] Yan-fei Zhong, Liang-pei Zhang, Jian-ya Gong, and Ping-xiang Li. "Remote sensing image classification based on artificial immune system". In: *JOURNAL OF REMOTE SENSING-BEIJING-* 9.4 (2005), p. 374.

[130] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[131] Zhi-Hua Zhou and Wei Tang. "Selective ensemble of decision trees". In: *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*. Springer. 2003, pp. 476–483.

[132] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. "Ensembling neural networks: many could be better than all". In: *Artificial intelligence* 137.1-2 (2002), pp. 239–263.

[133] Hui Zou, Trevor Hastie, and Robert Tibshirani. "Sparse principal component analysis". In: *Journal of computational and graphical statistics* 15.2 (2006), pp. 265–286.