Tree-structured discriminant analysis for mixed data

Eslava-Gomez, Guillermina; Cruz, Gonzalo Perez de la

Publication date:
2024

Document Version
Publisher's PDF, also known as Version of record

Tree-structured discriminant analysis for mixed data

Guillermina Eslava-Gómez [*†] Gonzalo Pérez de la Cruz [*]

DTU Compute Research Report. ISSN: 1601-2321
April, 2024

Abstract

Classification of multivariate observations into two or more populations based on a mixture of categorical and continuous variables, is a problem that is often solved by transforming variables to be all either continuous or categorical and then applying a classification method. We deal with the problem of classification of observations into two populations with binary and continuous variables using the ratio of two decomposable tree-structured conditional Gaussian (CG) densities as classification rule, where the tree structure and density for each population are estimated independently. The simplicity of CG densities with tree structure alleviates the problem of the need of large sample sizes, whereas the decomposability property ensures the existence of analytic expressions of the maximum likelihood estimators of the CG distribution and the use of a modified version of the Kruskal's algorithm to find the minimum spanning tree for the structure estimation. Since the selection of features often improves the classification performance of some methods, a stepwise procedure based on the cross-entropy loss is also proposed. We compare the empirical performance of the proposed method with that of other methods, classical and modern, using test error rates for a real data set and for simulated samples of different sizes from a CG density in each population. The empirical performance of the method in the real data was four among various methods. In the simulation, the proposed method was able to recover the structure of the CG densities from which the samples were generated and produced the lowest error rate; it was also observed that the error rates for all the methods were substantially larger than the population Bayes error for small sample sizes. The results suggest that the ratio of two CG densities with a tree structure is a good method, sufficiently fast computationally, worth considering for the classification of observations with mixtures of variables.

Key words. Classification methods, Conditional Gaussian distribution, Decomposable tree graphs, Deep neural networks, LassoNet, Logistic regression, Mixed graphical models, Mixed data.

[*]Departamento de Matemáticas, Facultad de Ciencias, UNAM, Av. Universidad 3000, Circuito Exterior S/N, México. eslava@ciencias.unam.mx and gonzalo.perez@ciencias.unam.mx

[†]Department of Applied Mathematics and Computer Science, Statistics and Data Analysis section, Technical University of Denmark, Kgs. Lyngby 2800, Denmark.

1 Introduction

The use of mixed graphical models (MGMs) for the classification of mixed data based on a set of categorical and continuous variables has been studied in Statistics since the 1970's. There are various other models and methods for classification that have been known for some time, like discriminant analysis in its different forms, parametric, non-parametric, linear and non-linear, and logistic discrimination. More recently, a surge of algorithmic methods has become popular for a data-driven approach to classification, such as random forests, neural networks, and lately, deep neural networks.

When dealing with numerical covariates or features, most classification methods apply. On the other hand, when all covariates are categorical, not all classification methods apply directly, as is the case of linear and quadratic discriminant analysis although they are often nonetheless successfully used. For a set of measurements that consist of categorical and continuous variables, some methods become more difficult to apply, particularly those based on probabilistic graphical models. Although the theory for graphical models for a mixture of variables has been successfully developed, see e.g. Lauritzen (1996, Ch. 6), and some algorithms for model identification and model estimation exist, the use of these models in practical applications is still somewhat limited due to scarce availability of software. Note, however, that some software has been available for some time, see e.g. Højsgaard et al. (2012, Ch. 5) and Scurati (2017).

MGMs, aimed at modelling mixtures of categorical and continuous variables, can be used in the context of classification of multivariate observations. The idea of using these models together with the conditional Gaussian (CG) distribution for classification is not new. The location model introduced by Olkin and Tate (1961), an instance of a homogeneous MGM, was used by Krzanowski (1975, 1980, 1994) for discrimination and classification, though it has not been broadly used due to the large sample size required for its estimation. Edwards (2000, Sec. 4.7) has also showed the application of MGMs for classification using a single MGM in discriminant analysis.

A full MGM has a large number of parameters and demands a large number of observations for its estimation, and selecting a more parsimonious model requires the identification of the graph, which is challenging if not infeasible. For this, some simplifications or modifications of the model have been proposed by restricting the graph structure or modifying the distribution of the variables. One approach is to consider a simplified CG distribution; for example, Lee and Hastie (2015) propose a model based on a homogeneous CG distribution with no interactions other than pairwise interactions, whereas in Cheng et al. (2015) the model includes higher-order interactions. Another approach is to consider MGMs with neighbourhood selection for graph identification and where the conditional distribution of each variable given the rest is a member of the exponential family, Chen et al. (2015) and Yang et al. (2014); or Yang et al. (2018), where the conditional distribution of each variable is modeled with a generalized linear model. A different approach is to consider latent Gaussian copula models, as in Fan et al. (2017) for binary and continuous variables, and recently by Göber et al. (2024) for more general categorical variables and continuous.

In this work, we consider an MGM associated with decomposable trees; this implies

a CG distribution with no interactions other than pairwise interactions, as in Lee and Hastie (2015), but without the restriction of homogeneity and with a subset of all pairwise interactions. An MGM with a decomposable tree graph is a simple decomposable graphical model that requires a much smaller sample size for its estimation and for which there exists an efficient algorithm for identifying the graph structure. Tree-structured models for classification in the pure discrete case have already been used by Chow and Liu (1966); these authors showed that a multinomial distribution with a tree structure could approximate the distribution of a multinomial distribution of discrete variables.

Following the idea of the use of the location model and tree-structured models, we propose using tree-structured MGMs for classification with the ratio of two CG distributions as the classification rule. We call this method tree-structured discriminant analysis (CG-tree). We compare the empirical performance of the proposed method with that of other methods for the case of two populations and a set of binary and continuous variables, using estimated classification error rates. We apply the methods to a real data set and to simulated datasets generated from a heterogeneous CG distribution with a path as the graph structure. We applied linear, nonlinear and algorithmic methods, including, linear, modified linear, naive and quadratic discriminant analysis, logistic regression with and without pairwise interactions, k nearest neighbour, support vector machines, random forests, neural networks and deep neural networks with variable selection in the recent platform LassoNet.

The paper is organized as follows. Section 2 states the classification problem, details the MGM and the CG density, and presents the proposed method. Section 3 specifies the alternative classification methods used. Section 4 presents the empirical results based on estimated error rates in a real and two simulated datasets. Finally, section 5 gives some concluding remarks.

2 Methodology

2.1 Classification problem

We consider the problem of classification between two well-defined populations or classes of observations, Π_1 and Π_2, on the basis of a mixture of p binary and continuous variables measured on a sample of observations from each class. Let $\mathcal{C} \in \{1, 2\}$ be the class variable and $x = (x_1, \ldots, x_p)$ the random vector of p variables of which q are binary $i = (i_1, \ldots, i_q)$ and r are continuous $y = (y_1, \ldots, y_r)$. Let $\pi_1 = P(\mathcal{C} = 1)$ and $\pi_2 = P(\mathcal{C} = 1)$ be the prior probabilities that an observation belongs to class Π_1 and Π_2, and $P(\mathcal{C} = 1|x)$ and $P(\mathcal{C} = 2|x)$ be the posterior probabilities, respectively.

For the classification of the observations we consider the Bayes classification rule with equal misclassification costs. This corresponds to choosing the class with the highest posterior probability $P(\mathcal{C}|x)$, see e.g. Welch (1939). That is, assign an observation to Π_1 if

$$P(\mathcal{C} = 1|x) > P(\mathcal{C} = 2|x). \tag{1}$$

If one assumes that x has a density function $f_{\mathcal{C}}(x|\mathcal{C}) = f_{\mathcal{C}}(x)$ in population $\mathcal{C} = 1, 2$, the Bayes rule (1) is equivalent to assigning an observation to Π_1 if

$$\log\big(f_1(x)/f_2(x)\big) - \log(\pi_2/\pi_1) > 0. \tag{2}$$

This rule is optimal in the sense that minimizes the overall error rate or probability of misclassification

$$P(e) = \pi_1 P(2|1) + \pi_2 P(1|2), \tag{3}$$

where $P(i|j)$ denotes the probability of assigning an observation from population Π_j to Π_i.

2.2 Mixed graphical models

MGMs, as considered in this work, were introduced by Lauritzen and Wermuth (1989). These models are based on the CG distribution and are used to model mixtures of variables, discrete and continuous, by combining hierarchical log-linear models for the discrete variables with Gaussian graphical models for the continuous variables. They are specified as follows.

Consider a set V of p variables partitioned as $V = \Delta \cup \Gamma$, where $\Delta = \{i_1, \ldots, i_q\}$ is a set of q discrete variables and $\Gamma = \{y_1, \ldots, y_r\}$ a set of r continuous variables. Each discrete variable $i_j \in \Delta$ takes a finite set of categories $\mathcal{I}_{i_j}$, which without loss of generality we assume $\mathcal{I}_{i_j} = \{0, 1\}$, the product space $\mathcal{I} = \prod_{j=1}^{q} \mathcal{I}_{i_j}$ is the table of cells or values that the vector of discrete variables i takes, and $y \in \mathbb{R}^r$. In an MGM, the vector of variables $x = (x_1, \ldots, x_{q+r}) = (i, y) = (i_1, \ldots, i_q, y_1, \ldots, y_r)$ has a CG density f that satisfies the Markov properties (Lauritzen, 1996, p. 32) with respect to an undirected marked graph $G = (V, E)$.

Conditional Gaussian (CG) distribution

The density of a CG distribution is expressed as the product of two densities

$$f(x) = f(i, y) = p(i) f(y|i), \tag{4}$$

where $p(i) = P(x_\Delta = i) > 0$ corresponds to a positive multinomial distribution and $f(y|i)$ to a Gaussian density $N\big(\mu(i), \Sigma(i)\big)$ for each $i \in \mathcal{I}$, $|\mathcal{I}| = 2^q$. The density $f(i, y)$ can be expressed in terms of its canonical $\big(g(i), h(i), K(i)\big)$, its moment characteristics $\big(p(i), \mu(i), \Sigma(i)\big)$, or a mixture of them like $\big(p(i), h(i), K(i)\big)$, see Lauritzen (1996, ch. 6); these are related as:

$$\begin{aligned} \mu(i) &= K(i)^{-1} h(i), \\ \Sigma(i) &= K(i)^{-1}, \\ p(i) &= (2\pi)^{r/2} \det(K(i))^{-1/2} \exp\big\{g(i) + h(i)^t K(i)^{-1} h(i)/2\big\}. \end{aligned} \tag{5}$$

Models for which $\Sigma(i) = \Sigma,\ \forall\ i \in \mathcal{I}$, are called homogeneous, else heterogeneous.

Notice that in general the moment characteristics in the MGM are not independent of each other and are restricted according to the model's associated graph G, see Lauritzen (1996, Theorem 6.11).

Decomposable MGMs

An MGM is decomposable if its associated marked graph is decomposable. An undirected marked graph is decomposable if and only if it is triangulated and does not contain any path between two non-adjacent discrete vertices passing through only continuous vertices (Lauritzen, 1996, p. 11). In the pure discrete or continuous case only the first condition applies.

When the graph G with p vertices is decomposable, there are two properties that ensure certain factorizations of the density: a) there exists a perfect numbering of the cliques of G, $C_1, ..., C_k$, and associated separators $S_1, ..., S_k^*$, $0 \leq k^* < k \leq p$, and b) there is a perfect directed version of G, where the vertices can be chosen such that discrete variables are numbered before the continuous ones, see Lauritzen (1996, p. 18).

Using these properties, the density $f(i, y)$ can be factorized in terms of weak marginal densities involving the variables in the cliques and separators only (Lauritzen, 1996, p. 188), or alternatively, in terms of conditional distributions as in Bayesian Networks as follows.

$$
\begin{aligned}
f(i,y) &= p(i)(2\pi)^{-r/2}\det(\Sigma(i))^{-1/2}\exp\{-(y-\mu(i))^t\Sigma(i)^{-1}(y-\mu(i))/2\} && (6)\\
&= \prod_{j=1}^{k}\frac{f_{[C_j]}(x_{C_j})}{f_{[S_j]}(x_{S_j})} && (7)\\
&= \prod_{j\in V} f(x_j \mid x_{pa_j}), && (8)
\end{aligned}
$$

where the weak marginal $f_{[A]}(x_A)$ of f over the set A is a CG density (Lauritzen, 1996, p. 162) and $f_{[\emptyset]} = 1$; x_{pa_j} denotes the parents of variable x_j defined as the variables, if any, that are connected with an arrow from x_{pa_j} to x_j in the corresponding perfect directed version of G.

Notice that i) $f_{[A]}(x_A)$ in general is not the marginal density $f_A(x_A)$, although they both have the same moment characteristics, ii) when x_j is continuous, $f(x_j \mid x_{pa_j})$ is a Gaussian distribution with the mean and variance depending on the variables in x_{pa_j}, and iii) continuous variables cannot be parents of discrete variables.

The maximum likelihood estimator of the density also factorizes as

$$
\begin{aligned}
\widehat{f}(i,y) &= \prod_{j=1}^{k}\frac{\widehat{f}_{[C_j]}(x_{C_j})}{\widehat{f}_{[S_j]}(x_{S_j})}, && (9)\\
&= \prod_{j\in V}\widehat{f}(x_j \mid x_{pa_j}), && (10)
\end{aligned}
$$

where each estimated factor, $\widehat{f}_{[A]}(x_A)$ or $\widehat{f}(x_j \mid x_{pa_j})$, is based on marginal data only. See Lauritzen (1996, p. 188) and Lindskou et al. (2021). This factorization allows to estimate $f(i, y)$ through its estimated factors only, without calculating the estimated parameters $\big(\widehat{p}(i), \widehat{\mu}(i), \widehat{\Sigma}(i)\big)$.

Decomposable MGMs with tree structure

An MGM with a decomposable tree or forest graph $G_\tau(V, E_\tau)$ is one of the simplest decomposable MGMs. A tree graph is a connected graph without cycles, and a forest is a graph that has no cycles. When the tree (forest) is decomposable, the expression (7) factorizes as

$$f(i,y) \;=\; \prod_{j\in V} f_{[x_j]}(x_j) \prod_{(j,k)\in E_\tau} \frac{f_{[x_j,x_k]}(x_j,x_k)}{f_{[x_j]}(x_j) f_{[x_k]}(x_k)}, \tag{11}$$

where $f_{[x_j,x_k]}(x_j,x_k)$ is a CG distribution involving only two variables x_j and x_k; and the alternative factorization in (8) is such that each x_{pa_j} corresponds to only one variable at most.

In this case, the maximum likelihood estimators of the density factors in (11) exist provided that the sample size for each factor is equal to or larger than two for each involved cell value, see Lauritzen (1996, Proposition 6.20).

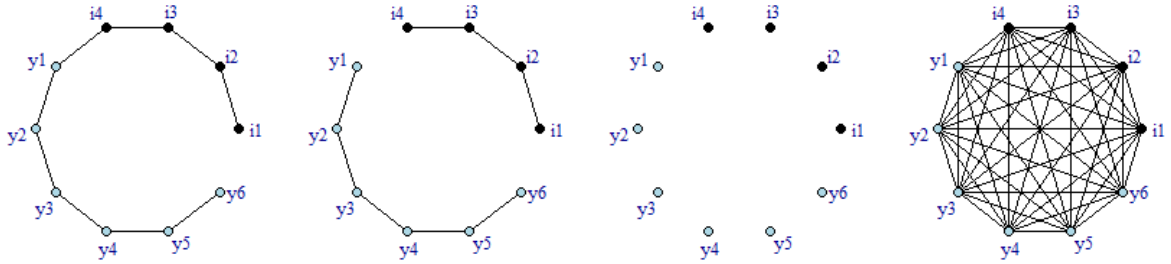
For example, consider 10 variables, $q = 4$ binary and $r = 6$ continuous, $(i,y) = (i_1,\ldots,i_4,y_1,\ldots,y_6)$ with $i \in \mathcal{I} = \{(0,0,0,0),(0,0,0,1),\ldots,(1,1,1,1)\}$, $|\mathcal{I}| = 2^4 = 16$ and $y \in \mathbb{R}^6$. Figure 1 shows four decomposable graphs: 1a) a decomposable tree, 1b) a forest, 1c) the empty graph, and 1d) the complete graph. For each graph, its density $f(i,y)$ can be factorized respectively as below, where b) and c) are homogeneous whereas a) and d) can be either homogeneous or heterogeneous:

a) $f(i,y) = p(i_1)\prod_{j=2}^4 p(i_j \mid i_{j-1}) f(y_1 \mid i_4) \prod_{j=2}^6 f(y_j \mid y_{j-1})$.

b) $f(i,y) = p(i_1)\prod_{j=2}^4 p(i_j \mid i_{j-1}) f(y_1) \prod_{j=2}^6 f(y_j \mid y_{j-1})$.

c) $f(i,y) = \prod_{j=1}^4 p(i_j) \prod_{j=1}^6 f(y_j)$.

d) $f(i,y) = p(i) f(y \mid i)$.



(a) (b) (c) (d)

Figure 1: Some examples of decomposable graphs associated with $f(i,y)$: a) One single path (tree), b) Two unconnected paths (forest), c) Empty graph (forest) and d) Complete graph.

2.3 Discriminant analysis and graphical models

Considering two populations Π_1 and Π_2, and a mixture of binary and continuous variables, $x = (i,y) = (i_1,\ldots,i_q,y_1,\ldots,y_r)$, $i_j \in \{0,1\}, j:1,\ldots,q$, and $y \in \mathbb{R}^r$. We assume that x

follows an MGM with density $f_{G_\mathcal{C}}(i,y)$ and associated graph $G_\mathcal{C} = (V, E_\mathcal{C})$ in population $\Pi_\mathcal{C}$, $\mathcal{C} \in \{1,2\}$. The graph $G_\mathcal{C} = (V, E_\mathcal{C})$ imposes some restrictions on the parameters $\big(p_\mathcal{C}(i), \mu_\mathcal{C}(i), \Sigma_\mathcal{C}(i)\big)$, for $i \in \mathcal{I}$ and $\mathcal{C} \in \{1,2\}$.

Using the Bayes rule in (2) with the densities $f_{G_1}(i,y)$ and $f_{G_2}(i,y)$ is a method that allows a reduction in the number of parameters by restricting the graph structures or by imposing equalities on the parameters. In fact, Krzanowski (1975) proposed this method with a common complete graph $G_1 = G_2 = G$ and with $\Sigma_1(i) = \Sigma_2(i) = \Sigma$ using a smoothed method for parameter estimation. Later, Krzanowski (1994) considered the case with $\Sigma_\mathcal{C}(i) = \Sigma_\mathcal{C}$, $\mathcal{C} = 1, 2$, and $\Sigma_1 \neq \Sigma_2$. In these cases, the structure of the graph is already identified as the complete graph, but the number of parameters is so large that their estimation demands a large number of observations. Perez-de-la-Cruz and Eslava (2016, 2019) proposed a discriminant analysis rule based on tree-structured graphical models in the pure continuous and discrete case, respectively.

The use of MGMs with a graph other than the complete or the empty graph, on the one hand, diminishes the number of parameters to be estimated; on the other, it imposes the problem of the identification or estimation of the graph which in general is not an easy task.

In practice, both the structure of $G_\mathcal{C}$ and the density $f_{G_\mathcal{C}}(i,y)$, for each $\mathcal{C} \in \{1,2\}$, should be estimated or learned. When restricting the structure of the graph to decomposable trees or forests, the structure of each $G_{\tau_\mathcal{C}}$ can be learned using the factorization in (11) and a modified version of Kruskal's algorithm for finding the maximum weight spanning decomposable tree or forest (Edwards et al., 2010). Then, each density $f_{\tau_\mathcal{C}}(i,y)$ can be estimated using the maximum likelihood estimators of the factors in (10). The availability of these methods to estimate the structures and densities is attractive from the computational point of view.

Tree-structured discriminant analysis

We propose a discriminant analysis rule as the Bayes rule given in (2) with two tree-structured decomposable CG densities, $f_{\tau_1}(i,y)$ and $f_{\tau_2}(i,y)$. That is, assign an observation $x = (i,y)$ to population $\mathcal{C} = 1$ if

$$\log\big(f_{\tau_1}(i,y)/f_{\tau_2}(i,y)\big) > \log\big(\pi_2/\pi_1\big), \tag{12}$$

for $i \in \mathcal{I}$ with $\mid \mathcal{I} \mid = 2^q$ and $y \in \mathbb{R}^r$.

The estimation of expression (12) is done by estimating each of the two densities separately to obtain:

$$\log\big(\widehat{f}_{\tau_1}(i,y)/\widehat{f}_{\tau_2}(i,y)\big) > \log\big(\widehat{\pi}_2/\widehat{\pi}_1\big), \tag{13}$$

where for each density, both the graph structure and the density are estimated as follows. The graph structure of $G_{\tau_\mathcal{C}}$ is learned using function *minForest* in the package *gRapHD* (Abreu et al., 2010) to obtain the decomposable tree (forest) that maximizes (minimizes) the likelihood (BIC). Then, each estimated density $\widehat{f}_{\tau_\mathcal{C}}(i,y)$ is computed with function

bn.fit in the package *bnlearn* (Scutari, 2017) using the maximum likelihood estimated factors in (10). The estimated population sizes $\widehat{\pi}_1$ and $\widehat{\pi}_2$ are the relative sample size of each population.

Notice that there are two options for estimating each decomposable graph: a tree, which is learned by maximizing the likelihood, or a forest, which is learned by minimizing the BIC. Here we use either two trees or two forests in the expression (13) corresponding to two estimated rules denoted as CG-tree and CG-forest.

The proposed rule in (12) considers the use of the p variables $x = (x_1, \ldots, x_p)$, however selecting some of the variables might improve the predictive accuracy. Since for any non-empty subset of the p variables, x^*, estimating $G_{\tau_1^*}$ and $G_{\tau_2^*}$, as well as the corresponding $f_{\tau_1^*}(x^*)$ and $f_{\tau_2^*}(x^*)$, is computationally fast and $\widehat{f}_{\tau_{\mathcal{C}}^*}(x^*)$ can be used to estimate $P(\mathcal{C}|x^*)$ for each observation in the sample as:

$$\widehat{P}(\mathcal{C}|x^*) \;\; = \frac{\widehat{f}_{\tau_{\mathcal{C}}^*}(x^*)\widehat{\pi}_{\mathcal{C}}}{\widehat{f}_{\tau_1^*}(x^*)\widehat{\pi}_1 + \widehat{f}_{\tau_2^*}(x^*)\widehat{\pi}_2}, \tag{14}$$

a backward step procedure based on the cross-entropy loss is also implemented and applied for variable selection. The selected models are denoted as CG-tree-step when applied to a CG-tree model, and CG-forest-step when applied to a CG-forest.

The performance of the proposed rule in (12) is compared with that of some alternative classification methods using simulated and real data.

3 Alternative classification methods

We give a brief note about each of the different classification methods used for comparing the performance of the tree-structured discriminant analysis.

Discrimination assuming normal populations. If one assumes that x is a vector of continuous variables with a multivariate Gaussian density on each population, $N\big(\mu_1, \Sigma_1\big)$ and $N\big(\mu_2, \Sigma_2\big)$, the rule in expression (2) is the quadratic discriminant function or rule (QDA). Assuming $\Sigma_1 = \Sigma_2 = \Sigma$, the quadratic reduces to a linear function (LDA). In addition to these two, we tried the LDA2 which corresponds to LDA based on the enlarged set of variables $\{x_1, \ldots, x_p, x_1x_2, \ldots, x_{p-1}x_p\}$, and a penalized version of LDA, LDA-pen, as in Witten and Tibshirani (2011).

Logistic discrimination. Logistic regression can be used for classification when variables are both binary and continuous. It estimates the posterior probabilities $P(\mathcal{C} = 1|x) = \exp(\beta_0 + \beta_1 x_1 + ... + \beta_p x_p)/[1 + \exp(\beta_0 + \beta_1 x_1 + ... + \beta_p x_p)]$ and $P(\mathcal{C} = 2|x) = 1 - P(\mathcal{C} = 1|x)$ for the Bayes discriminant rule (1). In this study we used logistic regression model in five forms: with main effects only, $\{x_1, \ldots, x_p\}$, and adding pairwise interactions, $\{x_1, \ldots, x_p, x_1x_2, \ldots, x_{p-1}x_p\}$, denoted as LOG and LOG2, respectively. In each case we used its regularized version with the l_1 metric corresponding to lasso (LOG-lasso and LOG2-lasso). We also applied a selection of variables using the backward step method with the BIC criterion, LOG2-step. The model LOG2 was applied but produced unstable results due to the large number of parameters and small data sets: the estimated

coefficients of the model were numerically too large, and these produced results with a large difference between test and training errors (low ability to generalize).

Naive Classifier. The naive classifier assumes independence among the p variables $(i_1, \ldots, i_q, y_1, \ldots, y_r)$ in each population distinguishing between categorical and continuous variables. This classifier can be obtained considering a CG density with the empty graph on each population, e.g. Figure 1c).

Support vector machines. Support vector machines (SVM) is a model that predicts the class identity of the observations by finding linear or non-linear boundaries in the feature space using kernels; this prediction is not based on class probabilities, see for example Cortes and Vapnik (1995). We used SVM using the standarized variables and reported results for the linear (SVM-lin) and the polynomial of degree 2 (SVM-poly2) kernels.

k-nearest neighbour. This nonparametric method estimates the posterior probability of membership of each observation to each of the populations, $P(\mathcal{C} = c|x)$. The observation is assigned to the class or population with the highest estimated posterior probability. These probabilities are estimated using the proportions associated with each population when considering the k nearest neighbours, $1 \leq k < n$, which are obtained using a distance measure on the predictor's values between the new and each of the n observations in the training sample. The larger the k, the smaller the error is at a price of a higher variance. We applied this method using the Euclidean distance and the standardized variables, and a tuned value of k (k-nn). We also applied k-nn with the Gower distance though it did not improve the results given when using the Euclidean distance.

Random forests. This methodology combines decision trees with the bootstrap method. It is used as a classifier, often successfully and with computational efficiency. It can be applied to data with both binary and continuous variables and can additionally provide a measure of the importance of each of the variables in terms of classification performance. Random forests are not invariant to transformations of the covariates. We applied the method tuning the number of fitted trees and the number of variables allowed to participate on each split, using the Gini index, and letting the trees to grow to the maximum possible considering that the minimum number of observations at each terminal node was one (Rand-forest).

Deep Neural networks and LassoNet. Neural networks including Deep neural networks (DNN) are powerful black-box predictors that can perform very well in different fields, like computer vision, image recognition, and language modelling, where large data sets are generally available, Agarwal et al. (2021). They can also be used for small-medium tabular data, though the large number of parameters involved in the model will often overfit the training set, and this makes it a challenge to learn a reasonable architecture and a good fit when the number of observations is small. Variable selection methodology for DNN has already been suggested and implemented, for example, LassoNet by Lemhadri et al. (2021), LocalGLMnet by Richman and Wuthrich (2023), and Weight predictor network with feature selection by Margeloiu et al. (2023). Their implementation is based on DNN algorithms already implemented. We fitted or learned DNN using Keras, and DNN with variable selection using LassoNet.

Table 1 lists the methods applied to the real and simulated data.

Classification Method	Name	Classification Method	Name
Methods with no interactions		*Methods with pairwise interactions*	
Linear discriminant analysis	LDA	Tree-structured discriminant	CG-tree
Penalized LDA	LDA-pen	Forest-structured discriminant	CG-forest
Naive	Naive	Step reduced Tree-structured disc.	CG-tree-step
Logistic regression	LOG	Step reduced Forest-structured disc.	CG-forest -step
Logistic lasso	LOG-lasso	Step reduced Logistic with pairwise int.	LOG2-step
SVM with linear kernel	SVM-lin	Logistic lasso with pairwise int.	LOG2-lasso
		Quadratic discriminant analysis	QDA
		LDA with pairwise interactions	LDA2
		SVM with polynomial kernel	SVM-poly2
Algorithmic methods		*Deep neural networks*	
K nearest neighbour	k-nn	Deep neural networks	DNN
Random forests	Rand-forest	DNN with variable selection	LassoNet

Table 1: Classification methods used to compare the performance of the tree-structure discriminant analysis.

4 Empirical comparison

We considered a real data set and two simulated data to compare the empirical performance of the proposed rule with that of other classification methods.

Assessment of the classification methods. Two aspects determining how well a method will perform are its ability to produce a small test error and to generalize, this is to produce a small difference between the test and the training error.

Preprocessing of the data. Most of the methods used in this work are scale-dependent, and no single transformation works well with all of them. For the methods: k-nn, DNN and LassoNet, the data were standardized to have mean zero and variance one on each variable. Software used for applying methods like penalized LDA, SVM and logistic regression with lasso, do internally standardize all the variables. For the real data set, variables four to six were transformed with the logarithm due to the large difference in variances among the continuous variables.

Misclassification errors. For the real data set, the test and training errors were calculated by randomly splitting the data into two parts, 90% for training and 10% for testing the models, repeating the splitting B times. For the simulated data, B samples of size $n \in \{50, 100, 1000\}$, were simulated for training and of size $n = 1000$ for testing the methods. For all the methods $B = 1000$, except for DNN and LassoNet, where $B = 50$ and $B = 100$, respectively. Additional computational details are given in Appendix C.

4.1 Real data

The data set has been previously analyzed by Krzanowski (1975) in the context of classification and discrimination of mixtures of variables, and earlier in Armitage et al. (1969) for the prognosis in advanced breast cancer. The data comprises 186 individuals who underwent ablative surgery for advanced breast cancer between 1958 and 1965 at Guy's Hospital, London. In 99 cases the treatment was considered to be either successful or intermediate, and in 87 as a failure. The former set will be treated as population Π_1 and the latter as Π_2. The dataset has six continuous and three binary variables.

The results of classifying the observations are presented in Figure 2. The test and training error rates for each and for both populations are given in Table 5 in the Appendix B. Figure 2 shows test and training errors in two groups, group one shows methods with no interactions (linear), and group two comprises methods with interactions, nonparametric or algorithmic, and deep learning methods (nonlinear). They show the following.

a) The proposed method, CG-tree-step, had an error of 33.2%, 2.3 percentage points higher than the lowest and 9.1 lower than the highest. The CG-forest-step, not shown in the figure, had 1.2 percentage points higher than CG-tree-step.

b) Methods with no interaction terms had errors that range from 37.8 for the logistic regression to 42.3 for the penalized linear discriminant rule. The lowest value was similar to the four largest in the nonlinear methods (k-nn, Rand-forest, QDA and CG-tree).

c) Nonlinear methods had errors that varied from 30.9 for the step reduced logistic regression up to 38.6 for the CG-tree rule.

d) Considering nonlinear methods, those with variable selection performed better than without (LOG2-step, LOG2-lasso, CG-tree-step, CG-forest-step).

e) Linear methods had, on average, an error four points larger than nonlinear methods, 39.3 vs 35.1; their lowest error rate was seven points higher than the lowest obtained by the nonlinear methods, 37.8 vs 30.9.

f) The two computer-intensive methods, DNN without and with variable selection (LassoNet), did not perform the best. We note that relatively little hyperparameter tuning was performed on these methods, and a more exhaustive hyperparameter optimization might have improved their performance.

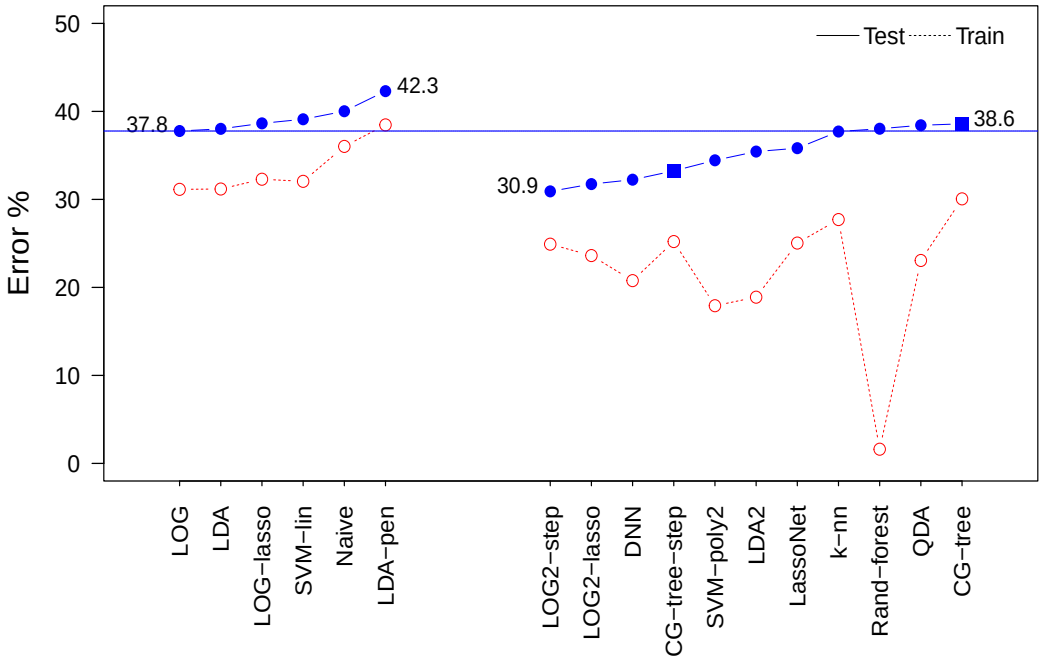


Figure 2: Breast cancer dataset with $n_{success} = 99$ and $n_{failure} = 87$. Estimated test and training error rates. Values averaged across 1000 random training-test data splits, except for DDN and LassoNet with 50 and 100 data splits, respectively. The data splits were done within each group in proportions $(9/10, 1/10)$. Labels of the horizontal axis appear in Table 1

4.2 Numerical simulations

We generated samples from a heterogeneous CG density under two settings for each population Π_1 and Π_2.

The CG-density has a single path as in Figure 1a) as associated graph and satisfies the Markov properties according to its graph, where the binary variables follow a multinomial distribution $p(i) = p(i_1, i_2)p(i_2, i_3)p(i_3, i_4)/(p(i_2)p(i_3))$, and the continuous a Gaussian distribution $N\big(\mu(i), K(i)^{-1}\big)$ within each cell i with a tridiagonal concentration matrix $K(i)$ as in (15) in Appendix A.

In one setting, the CG densities for the two populations have equal marginal means, equal marginal variances and different covariances, such that the difference between the distribution of the two populations lies on their covariances, and a second setting where both distributions have different marginal means and covariances. Specifically, the two settings are as follows.

I. The two CG densities f_{τ_1} and f_{τ_2} have equal marginal means, equal marginal variances and different marginal covariances. That is, $E_1(x_j) = E_2(x_j)$ and $V_1(x_j) = V_2(x_j)$, $j = 1, ..., 10$, and $Cor_1(x_i, x_{i+1}) > 0$, $\forall\ i \in \{1, \ldots, 9\}$ and $Cor_2(x_i, x_{i+1}) < 0$, $\forall\ i \in \{1, \ldots, 9\}$.

II. The two CG densities f_{τ_1} and f_{τ_2} have different marginal: means, variances and covariances.

The specifications of the parameter values for CG densities involved in the simulations are given in Table 2 in Appendix A.

An approximation to the value of the Bayes error in (3) was computed using 50,000 generated observations per population in each setting. For setting I, this value was 20.1% and for setting II, 20.2%.

The results of the simulation study are shown in Figure 3. Test and training error rates for both settings appear in Tables 6 and 7 in the Appendix B.

I. *Two Populations where both have equal marginal means and marginal variances.* In this setting, methods that include only main effects and no interactions will not discriminate between the two populations. The test error rate in all cases is around 50% as expected.

Considering the nonlinear methods we observe the following.

a) Error rates vary considerably according to the sample size. For sample size 50, there is an average difference of 12.6 points between the test and the population Bayes error. This average difference decreases to 7.8 and 1.7 for sample sizes 100 and 1000, respectively.

b) The proposed method based on the CG densities produced the lowest error rates. The differences between these errors and the Bayes error are 7.5, 4.0, and 0.1 for sample sizes 50, 100 and 1000, respectively. No noticeable difference was observed considering rules with trees or forests, or when considering the stepwise procedure for variable selection.

c) The second-best results were obtained by the quadratic discriminant and the logistic lasso rules.

d) Random forest and k-nn produced the largest test errors for all sample sizes.

e) DNN produced test errors that were among the 4 largest, and with variable selection using LassoNet, the error decreased, e.g. 2.8 percentage points for sample size 50. LassoNet has test errors similar to those for regularized regression.

II. *Two populations with different means and marginal means.* Here, we observe the following.

a) The linear methods with no interactions were able to discriminate between the two populations. Their performance was similar, with test errors that, on average, differed from the population error by 11.0, 9.4 and 7.6 points for sample size 50, 100 and 1000, respectively.

b) The performance of the nonlinear methods was similar to that of those where the two populations had equal marginal means and marginal variances, though the error rates differed less among the former, 28.1–34.8 for sample size 50, than among the latter, 27.6–40.0. Their performance, on average, differed from the population error by 10.4, 6.8 and 1.6 points for sample size 50, 100 and 1000, respectively.

c) CG-forest with and without variable selection, together with LOG2-lasso and QDA performed better than all the linear methods for the same sample size, and were the best in both simulation settings.

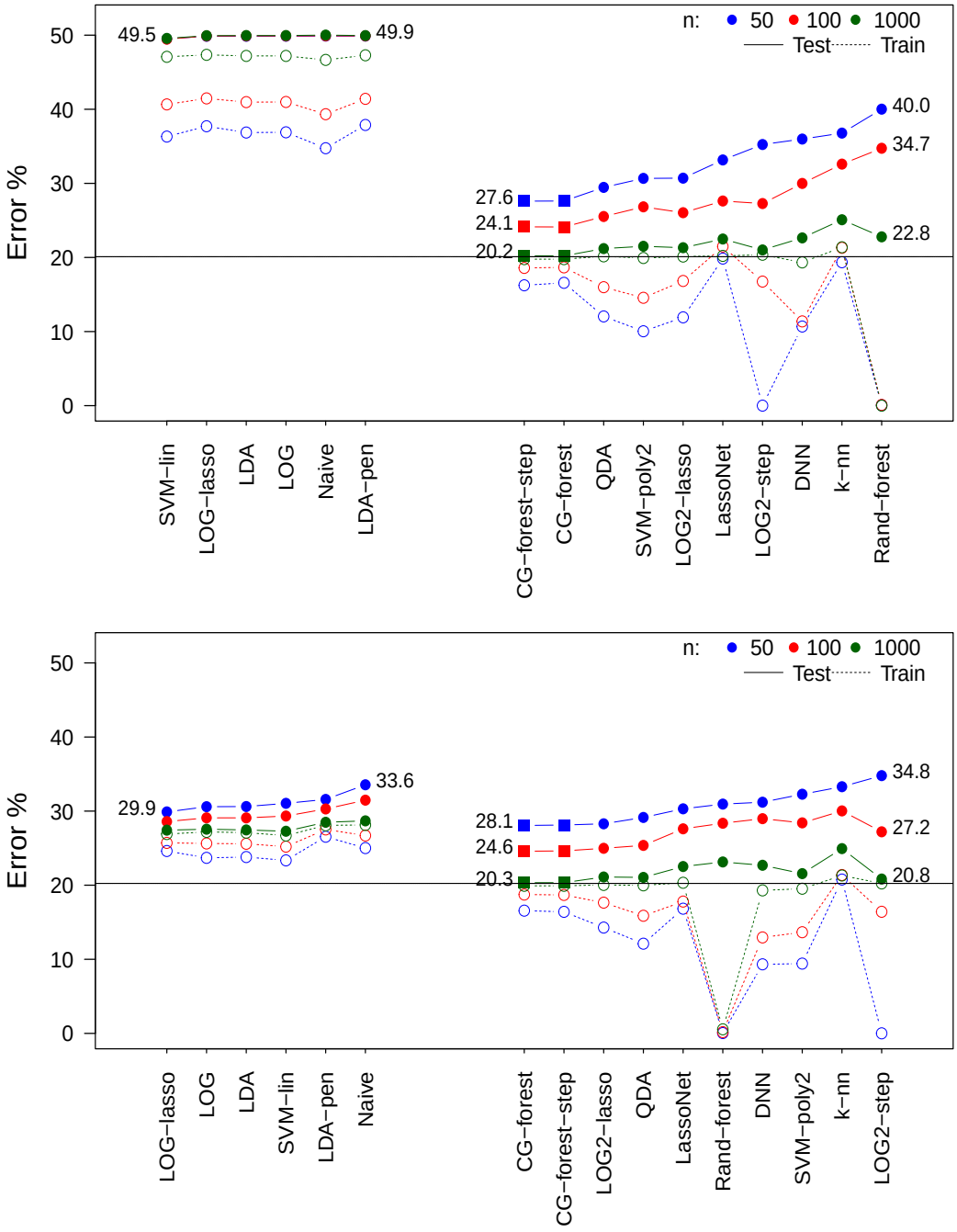


Figure 3: Simulated data. Estimated test and training error rates. Test errors averaged across 1000 test sets of size 1000, except for DDN and LassoNet with 50 and 100 test sets, respectively. Training sets of size 50 (blue), 100 (red) and 1000 (green) on each of the two populations. a) Top panel, setting where the data were generated from two CG densities with equal marginal means and marginal variances, $E_1(x_j) = E_2(x_j)$ and $V_1(x_j) = V_2(x_j)$, $j \in \{1, \ldots, 10\}$. b) Bottom panel, setting where the data were generated from two CG densities with different marginal means. Labels of the horizontal axis appear in Table 1

5 Concluding remarks

The simulation study is based on samples from density functions where interactions among variables are pairwise and not higher, and where the performance of the methods was largely due to the sample sizes.

For large sample sizes, in both simulation settings, all nonlinear methods performed equally well, except k-nn; whereas linear methods performed poorly in one setting and did not detect any difference between the two populations when these differed in their interactions and not in their means.

For the small sample size, in both settings, the proposed method based on CG tree models performed the best, though their errors were seven points higher than the Bayes error. This suggests that even when these models were parsimonious and in most of the cases the graph structure was recovered, the error associated with the estimation of the parameters diminished the performance of the estimated rule.

However, when interactions among three or more variables are present, the performance of the methods might be more variable. For small sample sizes, parsimonious methods are expected to perform better. For large sample sizes, methods that are able to capture high nonlinearities are expected to perform better than those that are limited to at most pairwise interactions. In the case, when the number of observations is large and interactions of high order are present, regularized logistic regression including products among three or more variables might perform well, and deep neural networks might outperform most methods.

Preprocessing the data, like using transformations or standardizing the variables, as well as using variable selection or regularization improved the classification performance in most of the cases. This supports the suggestion of using them when classifying observations and more strongly when the data sets are small.

The proposed method with variable selection and logistic regression with interactions and lasso had a good performance in the simulation and the real data sets. They are straightforward to apply and not highly computer-intensive. This makes them worth considering for classification, especially for small sample sizes where parsimonious methods with variable selection or regularization might perform better. They are also worth considering when computational resources are limited or simply as an additional alternative in order to compare the magnitude of error rates with those given by alternative classification methods.

Deep neural networks are computer-intensive methods that are powerful in tasks other than tabular data of small size. They did not perform the best in any of the simulated settings nor in the real data set; it might be that the small data sets were insufficient for a more efficient training of the networks, or that more expertise from the user was required. For high-dimensional small data sets it has been noticed that they do not perform well, Margeloiu et al. (2023).

In ongoing work we are studying the performance of the methods with samples from

populations with interactions among three or more variables; in particular, for analizing whether the proposed method and logistic lasso outperform methods with a larger number of parameters when the sample size is small.

Further research on the use of MGMs for classification is worth pursuing. For instance, some modifications to the ratio of two CG densities are the following. a) Restricting the densities in both populations to have the same decomposable tree graph, as the TAN structure in Friedman et al. (1997, 1998). b) The use of CG densities with decomposable models with a graph structure more complex than the trees. In this case, a fast algorithm for the identification of the structure should be implemented since the R package *gRapHD* is no longer maintained. c) The use of CG-densities where the interactions of the variables are limited up to two or three variables, using for example the R packages *mgm* (Haslbeck and Waldorp, 2020) or *hume* (Göber et al., 2024).

Acknowledgements Guillermina Eslava gratefully acknowledges the hospitality of the Department of Applied Mathematics and Computer Science, Technical University of Denmark. This work was done partly while she was on Sabbatical leave and partly on a six-month leave from the Faculty of Sciences at the National Autonomous University of Mexico (UNAM). She gratefully acknowledges the receipt of a grant from the program PASPA from DGAPA, UNAM, for six months of Sabbatical leave. Gonzalo Perez gratefully acknowledges that this work was supported by UNAM-PAPIIT IA101224.

References

Abreu, G., Edwards, D. and Labouriau, R. (2010) High-Dimensional Graphical Model Search with the gRapHD R Package. *Journal of Statistical Software,* 37(1), 1–18

Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R., and Hinton, G. E. (2021) Neural additive models: Interpretable machine learning with neural nets. *Advances in Neural Information Processing Systems*, 34.

Armitage, P., McPherson, C.K. and Copas, J.C. (1969). Statistical Studies of Prognosis in Advanced Breast Cancer, *Journal of Chronic Diseases*, 22, 5, 343-60.

Chen, S., Witten, D.M., and . Shojaie, A.S. (2015) Selection and estimation for mixed graphical models. *Biometrika*, 102, 1, 47-64

Cheng, J., Li, T., Levina, E. and Zhu, J. (2017) High-dimensional mixed graphical models. *J Comput Graph Stat*, 26(2), 367–378

Chow, C. and Liu, C. (1966) An approach to structure adaptation in pattern recognition. *IEEE Trans. Syst. Sci. Cybern.* 2, 73–80.

Cortes, C. and Vapnik, V. (1995) Support-Vector Networks. *Machine Learning*, 20, 273-297.

Edwards, D. (2000) *Introduction to Graphical Modelling.* Springer-Verlag, New York.

Edwards, D., Abreu, G. and Labouriau, R. (2010) Selecting high-dimensional mixed graphical models using minimal AIC or BIC forests. *BMC Bioinformatics* 11, 18.

Fan, J., Liu, H., Ning, Y. and Zou, H. (2017) High dimensional semiparametric latent graphical model for mixed data. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 79(2), 405–421.

Friedman, J., Tibshirani, R. and Hastie, T. (2010) Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33(1), 1-22.

Friedman, N., Geiger, D. and Goldszmidt, M. (1997) Bayesian network classifiers. *Mach Learn* 29, 131–163

Friedman, N., Goldszmidt, M. and Lee, T. (1998) Bayesian Network Classification with Continuous Attributes: Getting the Best of Both Discretization and Parametric Fitting. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998)*, 179–187.

Göber, K., Miloschewski, A., Drton, M. and Mukherjee, S. (2024) High-Dimensional Undirected Graphical Models for Arbitrary Mixed Data. arXiv:2211.11700v2 [stat.ML]

Haslbeck, J.M.B. and Waldorp, L.J. (2020) mgm: Estimating Time-Varying Mixed Graphical Models in High-Dimensional Data. *Journal of Statistical Software*, 93, 8, 1–46.

Højsgaard, S., Lauritzen, S.L. and Edwards, D. (2012) *Graphical Models with R.* Springer, New York

Krzanowski, W.J. (1975) Discrimination and classification using both binary and continuous variables, *J Amer. Statist. Assoc.* 70, 782-790.

Krzanowski, W. J. (1980) Mixtures of continuous and categorical variables in discriminant analysis. *Biometrics*, 36, 493-499.

Krzanowski, W. J. (1994) Quadratic location discriminant functions for mixed categorical and continuous data. *Statistics and Probability Letters,* 19, 2, 91-95

Lauritzen, S.L. (1996) *Graphical Models.* Clarendon Press, Oxford

Lauritzen, S. L. and Wermuth, N. (1989). Graphical models for associations between variables, some of which are qualitative and some quantitative. *Ann. Statist.* 17, 31-57.

Lee, J.D. and Hastie, T.J. (2015) Learning the structure of mixed graphical models. *J Comput Graph Stat* 24(1), 230–253

Lemhadri, I., Ruan, F., Abraham, L. and Tibshirani, R. (2021) LassoNet: a neural network with feature sparsity. *J Mach Learn Res* 22, 1–29

Liaw, A. and Wiener, M. (2002) Classification and Regression by randomForest. *R News* 2(3), 18–22

Lindskou, M., Tvedebrink, T., Eriksen, P.S. and Morling, N. (2021) Detecting Outliers in High-dimensional Data with Mixed Variable Types using Conditional Gaussian Regression Models. arXiv:2103.02366v3 [math.ST]

Lisha, L. and Jamieson, K. (2018) Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research* 18, 1–52.

Margeloiu, A., Simidjievski, N., Lio, P. and Jamnik, M. (2023) Weight predictor network with feature selection for small sample tabular biomedical data. *AAAI Conference on Artificial Intelligence*, 2023.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A. and Leisch F. (2020) e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien

Olkin, I. and R.F. Tate (1961) Multivariate correlation models with mixed discrete and continuous variables, *Ann. Math. Statist.* 32, 442-465. [Correction: ibid. 36, 343-344.1]

Perez-de-la-Cruz G, Eslava-Gomez G (2016) Discriminant analysis with Gaussian graphical tree models. *Adv Stat Anal.* 100, 161–187

Perez-de-la-Cruz G, Eslava-Gomez G (2019) Discriminant analysis for discrete variables derived from a tree-structured graphical model. *Advances in Data Analysis and Classification.* 7, 1–22.

R Core Team (2023) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

Richman, R. and Wuthrich, M.V. (2023) LASSO regularization within the LocalGLMnet architecture. *Advances in Data Analysis and Classification*, 17, 951—981.

Scutari, M. (2017) Bayesian Network Constraint-Based Structure Learning Algorithms: Parallel and Optimized Implementations in the bnlearn R Package. *Journal of Statistical Software*, 77(2), 1–20

Venables, W.N. and Ripley, B.D. (2002) *Modern Applied Statistics with S*. Springer, New York

Welch, B.L. (1939) Note on discriminant functions. *Biometrika*, 31, 218–220

Witten, D.M. and Tibshirani, R. (2011) Penalized classification using Fisher's linear discriminant. *J R Stat Soc Series B Stat Methodol.* 73(5), 753–772.

Yang, Y., Baker, Y., Ravimumar, P., Allen, G. and Liu, Z. (2014) Mixed Graphical Models via Exponential Families. *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, in *Proceedings of Machine Learning Research* 33, 1042–1050.

Yang, Z., Ning, Y. and Liu, H. (2018). On semiparametric exponential family graphical models. *J. Mach. Learn. Res.* 19, 1-59.

A Specifications of the CG densities in the simulation study

The parameters $(p(i), \mu(i), \Sigma(i))$ in the CG density were specified as follows. The inverse of the covariance matrix, $K(i)$, was a banded matrix:

$$K(i) = K(i_4) = \frac{1}{1-\rho^2}\begin{pmatrix} a_{i_4} & -\rho & 0 & 0 & 0 & 0 \\ -\rho & 1+\rho^2 & -\rho & 0 & 0 & 0 \\ 0 & -\rho & 1+\rho^2 & -\rho & 0 & 0 \\ 0 & 0 & -\rho & 1+\rho^2 & -\rho & 0 \\ 0 & 0 & 0 & -\rho & 1+\rho^2 & -\rho \\ 0 & 0 & 0 & 0 & -\rho & 1 \end{pmatrix}, \tag{15}$$

$|K(i)| = \frac{a_{i_4}-\rho^2}{(1-\rho^2)^6}$, $|\Sigma(i)| = |K^{-1}(i)| = \frac{(1-\rho^2)^6}{a_{i_4}-\rho^2} \to a_{i_4} > \rho^2$.

$$\Sigma(i) = K^{-1}(i) = \frac{1-\rho^2}{a_{i_4}-\rho^2}\times$$

$$\begin{pmatrix} 1 & \rho & \rho^2 & \rho^3 & \rho^4 & \rho^5 \\ \rho & a_{i_4} & a_{i_4}\rho & a_{i_4}\rho^2 & a_{i_4}\rho^3 & a_{i_4}\rho^4 \\ \rho^2 & a_{i_4}\rho & (a_{i_4}-1)\rho^2+a_{i_4} & (a_{i_4}-1)\rho^3+a_{i_4}\rho & (a_{i_4}-1)\rho^4+a_{i_4}\rho^2 & (a_{i_4}-1)\rho^5+a_{i_4}\rho^3 \\ \rho^3 & a_{i_4}\rho^2 & (a_{i_4}-1)\rho^3+a_{i_4}\rho & (a_{i_4}-1)(\rho^4+\rho^2)+a_{i_4} & (a_{i_4}-1)(\rho^5+\rho^3)+a_{i_4}\rho & (a_{i_4}-1)(\rho^6+\rho^4)+a_{i_4}\rho^2 \\ \rho^4 & a_{i_4}\rho^3 & (a_{i_4}-1)\rho^4+a_{i_4}\rho^2 & (a_{i_4}-1)(\rho^5+\rho^3)+a_{i_4}\rho & (a_{i_4}-1)(\rho^6+\rho^4+\rho^2)+a_{i_4} & (a_{i_4}-1)(\rho^7+\rho^5+\rho^3)+a_{i_4}\rho \\ \rho^5 & a_{i_4}\rho^4 & (a_{i_4}-1)\rho^5+a_{i_4}\rho^3 & (a_{i_4}-1)(\rho^6+\rho^4)+a_{i_4}\rho^2 & (a_{i_4}-1)(\rho^7+\rho^5+\rho^3)+a_{i_4}\rho & (a_{i_4}-1)(\rho^8+\rho^6+\rho^4+\rho^2)+a_{i_4} \end{pmatrix}, \tag{16}$$

with a_{i_4} taking values depending on the value of the corresponding cell $i \in \mathcal{I}$. For the specific case of a single path graph as in Fig. 1a), (15) and (16) depend on the value of

$i_4 \in \{0, 1\}$ only. Table 2 shows the values of the parameters used in the simulations.

	Population		Variable
	Π_1	Π_2	i_4
Equal marginal means and marginal variances in both distributions			
$p(1)$	.5	.5	
$p(0,0)$	.3	.2	
$p(1,1)$	.3	.2	
$K(i)$	(15) with $(\rho, a_{i_4}) = (0.2, 2.5)$	(15) with $(\rho, a_{i_4}) = (-0.2, 1.0)$	0
	(15) with $(\rho, a_{i_4}) = (0.2, 1.0)$	(15) with $(\rho, a_{i_4}) = (-0.2, 2.5)$	1
$\mu(i)$	$-0.5(1, \rho, \rho^2, \rho^3, \rho^4, \rho^5)$	$0.5(1, -\rho, \rho^2, -\rho^3, \rho^4, -\rho^5)$	0
	$0.5(1, \rho, \rho^2, \rho^3, \rho^4, \rho^5)$	$-0.5(1, -\rho, \rho^2, -\rho^3, \rho^4, -\rho^5)$	1
Different distributions in both populations			
$p(1)$	.5	.4	
$p(0,0)$	.3	.312	
$p(1,1)$	.3	.112	
$K(i)$	(15) with $(\rho, a_{i_4}) = (0.2, 2.0)$	(15) with $(\rho, a_{i_4}) = (-0.2, 1.0)$	0
	(15) with $(\rho, a_{i_4}) = (0.2, 1.0)$	(15) with $(\rho, a_{i_4}) = (-0.2, 2.0)$	1
$\mu(i)$	$(1, \rho, \rho^2, \rho^3, \rho^4, \rho^5)$	$(0, 0, 0, 0, 0)$	0
	$(0, 0, 0, 0, 0)$	$-(1, -\rho, \rho^2, -\rho^3, \rho^4, -\rho^5)$	1

Table 2: Heterogenous CG distribution. a) Top panel, setting where the two CG densities have equal marginal means and marginal variances, $E_1(x_j) = E_2(x_j)$ and $V_1(x_j) = V_2(x_j)$, $j \in \{1, \ldots, 10\}$, and $Cor_1(x_i, x_{i+1}) > 0$ and $Cor_2(x_i, x_{i+1}) < 0$, $\forall\ i \in \{1, \ldots, 9\}$. Estimated Bayes error rate of 20.1%. b) Bottom panel, setting where the data were generated from two CG densities with different marginal means. Estimated Bayes error rate of 20.2%. $p(1) = p(i_j = 1)$, $p(l, k) = p(i_j = l, i_{j+1} = k)$, $j \in \{1, \ldots, 3\}$, $l, k \in \{0, 1\}$, $p(1) + p(0) = 1$, $p(1, 0) = p(0, 1) = p(1) - p(1, 1)$.

i	$p(i)$		$N\big(\mu(i), \Sigma(i)(\rho)\big)$	
$i_1 i_2 i_3 i_4$	Π_1	Π_2	Π_1	Π_2
0000	.108	.032	$\mu(i) = -.5(1, \rho, \rho^2, \rho^3, \rho^4, \rho^5)$	$\mu(i) = .5(1, \rho, \rho^2, \rho^3, \rho^4, \rho^5)$
0010	.048	.048	$\Sigma(i)$ with $(\rho = .2, a_{i_4} = 2.5)$	$\Sigma(i)$ with $(\rho = -.2, a_{i_4} = 1)$
0100	.048	.072		
0110	.048	.072		
1000	.072	.048		
1010	.032	.108		
1100	.072	.048		
1110	.072	048		
0001	.072	.048	$\mu(i) = .5(1, \rho, \rho^2, \rho^3, \rho^4, \rho^5)$	$\mu(i) = -.5(1, \rho, \rho^2, \rho^3, \rho^4, \rho^5)$
0011	.072	.048	$\Sigma(i)$ with $(\rho = .2, a_{i_4} = 1)$	$\Sigma(i)$ with $(\rho = -.2, a_{i_4} = 2.5)$
0101	.032	.108		
0111	.072	.048		
1001	.048	.072		
1011	.048	.072		
1101	.048	.072		
1111	.108	.032		

$.3^3/.5^2 = .108; .3 \times .2^2/.5^2 = .048;\ .2 \times .3^2/.5^2 = .072; .2^3/.5^2 = .032$

Table 3: Heterogenous case where the two CG densities $f_1(x)$ and $f_2(x)$ have equal marginal means and marginal variances, $E_1(x_j) = E_2(x_j)$ and $V_1(x_j) = V_2(x_j)$, $j \in \{1, \ldots, 10\}$, and $Cor_1(x_i, x_{i+1}) > 0$ and $Cor_2(x_i, x_{i+1}) < 0$, $\forall\, i \in \{1, \ldots, 9\}$. $K(i)$ depends on a single parameter ρ and is given in (15).

i	$p(i) = \frac{p(i_1,i_2)p(i_2,i_3)p(i_3,i_4)}{p(i_2)p(i_3)}$		$N\big(\mu(i), \Sigma(\rho)\big)$	
$i_1 i_2 i_3 i_4$	Π_1	Π_2	Π_1	Π_2
0000	.108	.1898	$\mu(i) = (1, \rho, \rho^2, \rho^3, \rho^4, \rho^5)$	$\mu(i) = (0, 0, 0, 0, 0, 0)$
0010	.048	.1617	$\Sigma(i)$ with $(\rho = .2, a_{i_4} = 2)$	$\Sigma(i)$ with $(\rho = -.2, a_{i_4} = 1)$
0100	.048	.1617		
0110	.048	.0581		
1000	.072	.1752		
1010	.032	.1493		
1100	.072	.0629		
1110	.072	0226		
0001	.072	.1752	$\mu(i) = (0, 0, 0, 0, 0, 0)$	$\mu(i) = -(1, \rho, \rho^2, \rho^3, \rho^4, \rho^5)$
0011	.072	.0629	$\Sigma(i)$ with $(\rho = .2, a_{i_4} = 1)$	$\Sigma(i)$ with $(\rho = -.2, a_{i_4} = 2)$
0101	.032	.1493		
0111	.072	.0226		
1001	.048	.1617		
1011	.048	.0581		
1101	.048	.0581		
1111	.108	.0088		

$.3^3/.5^2 = .108; .3 \times .2^2/.5^2 = .048;\ .2 \times .3^2/.5^2 = .072; .2^3/.5^2 = .032$

$.312^3/.4^2 = .1898; .312 \times .288^2/.4^2 = .1617;\ .288^2 \times .112/.4^2 = .0581; .288 \times .312^2/.4^2 = .1752; .288^3/.4^2 = .1493;$

$.112 \times .288 \times .312/.4^2 = .0629; .112^2 \times .288/.4^2 = .0226; .112^3/.4^2 = .0088$

Table 4: Heterogenous case with two different CG densities $f_1(x)$ and $f_2(x)$ where the marginal means and variances are different on each population. $K(i)$ depends on a single parameter ρ and is given in (15).

B Error rates for the real and simulated data sets

Method	*Error rates %*					
	Training set			*Test set*		
Group	Π_1	Π_2	Global	Π_1	Π_2	Global
Methods with no interactions						
Linear discriminant analysis	24.6	38.7	31.2	30.3	46.8	38.0
Penalized LDA	31.2	46.7	38.5	34.5	51.0	42.3
Naive	23.1	50.8	36.0	26.4	55.4	40.0
Logistic regression	25.3	37.8	31.1	30.8	45.6	37.8
Logistic lasso	22.9	43.0	32.3	28.4	50.2	38.6
SVM with linear kernel	23.8	41.5	32.1	29.9	49.4	39.1
Methods with pairwise interactions						
Tree-structured discriminant	19.6	41.9	30.1	26.7	52.0	38.6
Step reduced Tree-structured discriminant	17.9	33.5	25.2	24.6	43.0	33.2
LDA with pairwise interactions	13.6	24.9	18.9	29.7	41.9	35.4
Step reduced Logistic with pairwise interactions	19.7	30.9	24.9	24.4	38.2	**30.9**
Logistic lasso with pairwise interactions	19.0	28.9	23.6	24.6	39.8	31.7
Quadratic discriminant analysis	14.3	33.1	23.1	30.1	47.8	38.4
Algorithmic methods						
SVM with polynomial kernel	11.1	25.7	17.9	25.6	44.4	34.4
K nearest neighbour	20.4	36.1	27.7	29.0	47.5	37.7
Random forests	0.1	3.3	1.6	27.2	50.2	38.0
Deep neural networks						
Deep neural networks	15.0	27.3	20.8	27.1	38.0	32.2
DNN with variable selection	18.6	32.3	25.0	29.0	43.5	35.8

Table 5: Breast cancer dataset with $n_{success} = 99$ and $n_{failure} = 87$. Estimated test and training error rates. Values averaged across 1000 random training-test data splits, except for DDN and LassoNet where there were 50 and 100 data splits, respectively. The data splits were done within each group in proportions $(9/10, 1/10)$. Labels of the horizontal axis appear in Table 1

Method	*Error rates %*					
	Training set			*Test set*		
Group sample size	50	100	1000	50	100	1000
Methods with no interactions						
Linear discriminant analysis	36.9	41.0	47.2	49.9	49.9	50.0
Penalized LDA	37.9	41.4	47.3	49.9	49.9	49.9
Naive	34.7	39.3	46.7	49.9	49.9	50.0
Logistic regression	36.9	41.0	47.2	49.9	49.9	50.0
Logistic lasso	37.7	41.5	47.3	49.9	49.9	49.9
SVM with linear kernel	36.3	40.7	47.1	49.5	49.5	49.6
Methods with pairwise interactions						
Forest-structured discriminant	16.6	18.7	19.8	27.6	24.1	20.2
Step reduced Forest-structured discriminant	16.2	18.6	19.8	27.6	24.1	20.2
Step reduced Logistic with pairwise interactions	0.0	16.7	20.4	35.2	27.3	21.0
Logistic lasso with pairwise interactions	11.9	16.8	20.1	30.7	26.0	21.3
Quadratic discriminant analysis	12.0	16.0	20.1	29.5	25.5	21.2
Algorithmic methods						
SVM with polynomial kernel	10.0	14.6	19.9	30.7	26.8	21.5
K nearest neighbour	19.4	21.3	21.4	36.8	32.6	25.1
Random forests	0.0	0.1	0.0	40.0	34.7	22.8
Deep neural networks						
Deep neural networks	10.7	11.4	19.3	36.0	30.0	22.6
DNN with variable selection	19.9	21.5	20.2	33.2	27.6	22.5

Table 6: Simulated data. Estimated test and training error rates. Test errors averaged across 1000 test sets of size 1000, except for DDN and LassoNet with 50 and 100 test sets, respectively. Training errors averaged across training sets of size 50, 100 and 1000 on each of the two populations. Setting where the data were generated from two CG densities with equal marginal means and marginal variances, $E_1(x_j) = E_2(x_j)$ and $V_1(x_j) = V_2(x_j)$, $j \in \{1, \ldots, 10\}$, and $Cor_1(x_i, x_{i+1}) > 0$ and $Cor_2(x_i, x_{i+1}) < 0$, $\forall\ i \in \{1, \ldots, 9\}$.

Method	*Error rates %*					
	Training set			*Test set*		
Group sample size	50	100	1000	50	100	1000
Methods with no interactions						
Linear discriminant analysis	23.8	25.6	27.1	30.6	29.1	27.5
Penalized LDA	26.5	27.5	28.1	31.6	30.3	28.5
Naive	25.0	26.7	28.1	33.6	31.5	28.7
Logistic regression	23.7	25.6	27.2	30.6	29.1	27.6
Logistic lasso	24.6	25.7	26.9	29.9	28.6	27.4
SVM with linear kernel	23.3	25.2	26.7	31.0	29.3	27.3
Methods with pairwise interactions						
Forest-structured discriminant	16.6	18.7	19.9	28.1	24.6	20.3
Step reduced Forest-structured discriminant	16.4	18.7	19.9	28.1	24.6	20.3
Step reduced Logistic with pairwise interactions	0.0	16.4	20.2	34.8	27.2	20.8
Logistic lasso with pairwise interactions	14.3	17.6	20.0	28.3	25.0	21.1
Quadratic discriminant analysis	12.1	15.9	20.0	29.1	25.4	21.1
Algorithmic methods						
SVM with polynomial kernel	9.4	13.7	19.5	32.3	28.4	21.6
K nearest neighbour	20.8	21.4	21.3	33.3	30.0	24.9
Random forests	0.0	0.2	0.6	30.9	28.4	23.1
Deep neural networks						
Deep neural networks	9.3	12.9	19.3	31.2	29.0	22.7
DNN with variable selection	16.8	17.8	20.3	30.3	27.6	22.5

Table 7: Simulated data. Estimated test and training error rates. Test errors averaged across 1000 test sets of size 1000, except for DDN and LassoNet with 50 and 100 test sets, respectively. Training errors averaged across training sets of size 50, 100 and 1000 on each of the two populations. Setting where the data were generated from two CG densities with different marginal means.

C Computational details

The computation of most of the errors was done with R (R Core Team, 2023), using the following functions and packages. For logistic discrimination, *glm* and *step* with the BIC criterion; and *cv.glmnet* in *glmnet* package (Friedman et al., 2010) for lasso with λ tuned over a grid of 100 values. For discrimination assuming normal populations, *lda* and *qda*, both functions in the *MASS* package (Venables and Ripley, 2002); and *PenalizedLDA.cv* in the *penalizedLDA* package (Witten and Tibshirani, 2011) with $\lambda \in \{0.001, 0.005, 0.010, 0.030, ..., 0.490\}$. For the Naive classifier and SVM, functions *naiveBayes* and *svm* in *e1071* package (Meyer et al., 2020), tuning the hyperparameters cost $\in \{0.01, 0.10, 1, 10, 100\}$ and additionally for the quadratic polynomial kernel $\gamma \in \{.00001, .0001, .001, .01, .03, 0.0667, .1, .5\}$. For k-nn the function *knn* in *class* package (Venables and Ripley, 2002), with $k \in \{1, 2, ..., 20\}$. For Random forest, package *randomForest* (Liaw and Wiener, 2002) with ntree $\in \{100, 500, 1000\}$ and mtry $\in \{1, 2, ..., 10\}$. The hyperparameters of all the previous methods were tuned by tenfold cross-validation, except for Random forest where the out-of-bag error was used.

The training of the deep neural networks was done with *Keras* with most of the hyperparameters with their default values for a one-hidden-layer feed-forward neural network with ReLU activation function, Adam optimizer and the following values: the number of neurons within $\{7, 8, ..., 11\}$ and $\{1, 2, ..., 15\}$ for the real and simulated data, respectively, a learning rate value between 0.0001 and .02, dropout $\in \{0, .1, .2\}$ and a regularization value $l_1 \in \{0, .001, .005, .01, .025, .05, .1\}$. The tuning process was done using the *Hyperband* Tuner in Keras (Lisha and Jamieson, 2018).

The training of deep neural network with variable selection was done using the platform *LassoNet* with default values for most of the hyperparameters except for the following: one hidden layer with the number of neurons within $\{7, 8, ..., 11\}$ and $\{4, 6, ..., 12\}$ for the real and simulated data, respectively; with M $\in \{1, 10\}$ and dropout $\in \{0, .2, .4\}$. The tuning of the hyperparameters was done by fivefold cross-validation using function *LassoNetClassifierCV*.