Journal of Machine Learning Research 21 (2020) 1-33 Submitted 6/18; Revised 3/20; Published 10/20

Beyond Trees: Classification with Sparse Pairwise Dependencies

Yaniv Tenzer YANIV.TENZER@WEIZMANN.AC.IL
Department of Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot, 76100, Israel

Amit Moscovich AMIT@MOSCOVICH.ORG
Program in Applied and Computational Mathematics
Princeton University
Princeton, NJ 08544, USA

Mary Frances Dorn MFDORN@LANL.GOV
Los Alamos National Laboratory
P.O. Box 1663, MS F600
Los Alamos, NM 87545, USA

Boaz Nadler BOAZ.NADLER@WEIZMANN.AC.IL
Department of Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot, 76100, Israel

Clifford Spiegelman
Department of Statistics
Texas A&M University
College Station, TX 77843, USA

Editor: Gal Elidan

Abstract

Several classification methods assume that the underlying distributions follow tree-structured graphical models. Indeed, trees capture statistical dependencies between pairs of variables, which may be crucial to attaining low classification errors. In this setting, the optimal classifier is linear in the log-transformed univariate and bivariate densities that correspond to the tree edges. In practice, observed data may not be well approximated by trees. Yet, motivated by the importance of pairwise dependencies for accurate classification, here we propose to approximate the optimal decision boundary by a sparse linear combination of the univariate and bivariate log-transformed densities. Our proposed approach is semi-parametric in nature: we non-parametrically estimate the univariate and bivariate densities, remove pairs of variables that are nearly independent using the Hilbert-Schmidt independence criterion, and finally construct a linear SVM using the retained log-transformed densities. We demonstrate on synthetic and real data sets, that our classifier, named SLB (sparse log-bivariate density), is competitive with other popular classification methods.

Keywords: binary classification, graphical model, Bayesian network, sparsity, semi-parametric

1. Introduction

Consider a binary classification problem where the vector of explanatory variables $\mathbf{X} = (X_1, \dots, X_d)$ belongs to some d-dimensional space $\Omega \subseteq \mathbb{R}^d$, and the response variable Y takes values in the set $\mathcal{Y} = \{+1, -1\}$. Given a training set of labeled samples $\{\mathbf{x}_\ell, y_\ell\}_{\ell=1}^n$, the goal is to construct a classifier $f : \Omega \to \mathcal{Y}$, with a small misclassification error rate on future unlabeled instances of $\mathbf{X}$.

It is well known that the optimal classifier is a threshold of the likelihood ratio statistic,

$$\mathrm{LR}(\mathbf{x}) = \frac{p(\mathbf{x}|Y = +1)}{p(\mathbf{x}|Y = -1)}. \tag{1}$$

A key challenge in applying this result, is the estimation of the class-conditional densities $p(\mathbf{x}|Y = y)$. Unfortunately, accurate non-parametric density estimation requires a number of labeled samples that is exponential in the dimension d. This is known as the *curse of dimensionality* (see Chapter 2 of Tsybakov (2009)). Hence, in high dimensional settings with a limited number of labeled samples, classification methods based on non-parametric multivariate density estimates may perform poorly and are not widely used in practice.

Instead of a non-parametric estimate of $p(\mathbf{x}|y)$, a popular alternative is to model this class conditional multivariate density with Bayesian Networks (Lauritzen, 1996). Bayesian networks provide a powerful probabilistic representation of multivariate distributions, which in turn often yields accurate classifiers. However, learning an unrestricted Bayesian network may require a large number of labeled samples. Moreover it may be computationally prohibitive (Friedman et al., 1997; Grossman and Domingos, 2004).

To ease the computational burden, many works suggested various simplifying assumptions on the underlying distribution. One popular approach is to restrict attention to specific parametric forms such as the multivariate Gaussian or other distributions, for example, copulas (Lauritzen, 1996; Elidan, 2012). A different approach, most relevant to our work and reviewed in Section 2, is based on the assumption that the multivariate distribution of each class follows a tree-structured Bayesian network (Tan et al., 2011; Friedman et al., 1997). In this case, the tree of each class may be learned by the computationally efficient algorithm of Chow and Liu (1968) or extensions thereof (Lafferty et al., 2012). As we show in Section 3, the tree structure implies that the likelihood ratio statistic of Eq. (1) takes the form of a sparse linear combination of log-transformed univariate and bivariate density terms. Hence, these tree-based approaches require only univariate and bivariate density estimates, and may be applied to high dimensional settings with a limited number of samples. However, as real data sets may not follow a tree distribution, the resulting tree-based classifier may not accurately capture the optimal decision boundary.

Motivated by the success of forest-based approaches, in Section 3 we present a new semi-parametric classifier, denoted as the sparse log-bivariate density classifier (`SLB`). As its name suggests, it is also based on univariate and bivariate densities. However, instead of imposing hard tree constraints, we assume that the decision boundary can be well approximated by a *sparse* linear combination of the logarithms of the univariate and bivariate densities. Concretely, the `SLB` classifier is constructed as follows: first, we run a feature selection procedure, removing pairs of variables that are nearly independent. Next, we non-parametrically estimate for each class, its univariate densities as well as the bivariate densities of those pairs of variables not excluded by the feature selection procedure. Finally, we construct a linear SVM in the space of these log-transformed density estimates.

Our method can be viewed as a generalization of tree-based classifiers. As we prove in Lemma 1, when the conditional distributions of both classes are tree-structured Bayesian networks, the log-likelihood ratio is a *linear* function of the log-transformed univariate and bivariate densities. The vector of coefficients of the resulting linear classifier, whose entries correspond to these log-transformed densities, is sparse. Furthermore, its non-zero entries are integers, determined by the tree structure. Our approach eliminates the tree structure assumption, thus relaxing the integer constraints on the coefficients. Instead, it learns a sparse linear classifier in the space of these log-transformed densities. Our approach also extends the recent work of Fan et al. (2016), which considered a logarithmic transformation of only univariate densities to address the same problem of binary classification.

In Section 4 we present some theoretical properties of our classifier, in particular, its asymptotics as the sample size tends to infinity. In Section 5 we compare our approach to several popular classifiers on both synthetic and real data sets. As our empirical results show, when the underlying distribution is forest structured, the accuracy of our classifier is comparable to methods that explicitly assume a forest structure. However, when the underlying distributions are not forest structured, but instead follow more complicated Bayesian network models, our more flexible method often outperforms the other methods. Furthermore, our experiments highlight the importance of incorporating bivariate features. Finally, as we illustrate with several real data sets, `SLB` is competitive to popular classifiers. We conclude the paper in Section 6 with a discussion and potential extensions.

2. Background

In Section 2.1 we briefly review Bayesian networks and their relation to our approach. In Section 2.2 we describe the Hilbert-Schmidt independence criterion, used in our feature selection step.

2.1 Bayesian Network Classifiers

Bayesian networks provide a general framework to represent and infer about high dimensional densities (Koller and Friedman, 2009). A Bayesian network is described by two components: (i) a directed acyclic graph $\mathcal{G}$ whose nodes correspond to the random variables $X_1, \dots, X_d$. (ii) a set of conditional densities, $\{f(X_i|Par_i)\}$, where Par_i denotes the parents of variable X_i in the graph $\mathcal{G}$. For a root node X_i, without parents, $Par_i = \emptyset$, $f(X_i|Par_i)$ is its marginal density. The structure of the graph $\mathcal{G}$ encodes the statistical dependencies among the d variables X_i. Specifically, given the values of its parents, each random variable X_i is conditionally independent of all other variables X_j which are not its descendants (Lauritzen, 1996); The full density $p(\mathbf{x})$ is the product of all of these conditional densities. It can be shown that this indeed defines a valid joint density, which satisfies the conditional independence properties encoded by $\mathcal{G}$ as outlined above.

One popular application of Bayesian networks is to construct classifiers using the likelihood ratio. Since in practice, neither the graph structure nor the conditional densities are known, an essential part in constructing a Bayesian-network-based classifier is learning, for each class label, its graph $\mathcal{G}$ and the corresponding conditional densities. This problem is known as *structure learning* and has received considerable interest in recent years in the context of Bayesian networks and more general Graphical models (Lee and Hastie, 2015; Yang et al., 2015; Beretta et al., 2018; Yu et al., 2020).

Unfortunately, even with as few as 20 variables, learning a general real-valued graphical model beyond the simple Gaussian Bayesian network can be computationally impractical (Friedman et al., 1997; Grossman and Domingos, 2004). To overcome this computational burden, a common approach is to resort to *tree-structured* networks. Under this assumption, the probability density $p(\mathbf{x})$ factorizes as

$$p_T(\mathbf{x}) = p(x_{m_1}) \prod_{i=2}^{d} p(x_{m_i} | x_{m_{j(i)}}), \tag{2}$$

where $(m_1, \dots, m_d)$ is a permutation of $\{1, \dots, d\}$ and $j(i) \in \{1, \dots, i-1\}$ is the parent of i in the tree. An important property resulting from the tree-structured assumption is that for each class, the corresponding d-dimensional density depends only on its univariate and bivariate marginals. Given n samples, the structure learning task simplifies to estimating the optimal tree structure and its associated univariate and bivariate densities.

For a discrete multivariate distribution, a tree structure may be estimated by the computationally efficient algorithm of Chow and Liu (1968). If the underlying distribution follows a tree structure, then as sample size $n \to \infty$ the estimated tree is consistent (Chow and Wagner, 1973). Tan et al. (2011) extended this approach to more sparse structures, by removing weak edges from the learned tree. The result is a forest, or union of disjoint trees, with the following factorization,

$$p_F(\mathbf{x}) = \prod_{(i,j) \in E_F} \frac{p(x_i, x_j)}{p(x_i) p(x_j)} \prod_{i \in V_F} p(x_i), \tag{3}$$

where E_F and V_F are the edge and vertex sets of the forest. Lafferty et al. (2012) extended the Chow and Liu approach to the case of multivariate continuous data by using kernel density estimates of the univariate and bivariate densities in Eq. (2). In their paper, the authors presented conditions under which this approach is consistent.

Several classifiers have been proposed which are based on specific forest models. Perhaps the simplest one is the popular naive Bayes classifier (Duda and Hart, 1973; Langley et al., 1992). This classifier assumes that the class conditional joint density is simply the product distribution $p_y(\mathbf{x}) = \Pi_i p_y(x_i)$, whose corresponding graph has no edges. A more sophisticated model was considered by Park et al. (2011). They assumed that *both* classes are distributed according to the same a Markov chain. Namely, there exists a permutation $\pi \in \mathrm{Sym}(d)$ such that for all $\mathbf{x}$ and y, $p_y(\mathbf{x}) = p_y(x_{\pi_1}) p_y(x_{\pi_2} | x_{\pi_1}) \dots p_y(x_{\pi_d} | x_{\pi_{d-1}})$.

Finally, Friedman et al. (1997) introduced the tree-augmented Naive-Bayes model (`TAN`). Their approach learns a tree structure over the set of variables for each class, and then apply a likelihood ratio based classifier.

A different approach that makes a tree assumption is discriminative. For example, Tan et al. (2010) proposed to learn trees that are specifically tailored for classification. Their procedure fits each tree distribution to the observed data from that class while simultaneously maximizing the distance between the two distributions. Another example of a discriminative approach is the work of Meshi et al. (2013), where instead of fitting a tree to the observed data, the authors suggest to learn a tree structure that directly minimizes a classification loss function.

2.2 Hilbert-Schmidt Independence Criterion

An important component in our semi-parametric approach is a feature selection step, whereby we remove bivariate densities that correspond to (nearly) independent variables. We perform this step using the Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2005a,b). For completeness we briefly review this method, restricting attention to the case of continuous real-valued random variables.

Let (Z, W) be a pair of real-valued random variables on a compact domain $\Omega_z \times \Omega_w \subset \mathbb{R}^2$ with joint density P_{zw} and marginals P_z, P_w. HSIC is a method to assess if Z and W are independent, i.e., if $P_{z,w} = P_z \times P_w$. The basic idea is that while $\operatorname{Cov}(Z, W) = 0$ does not imply that Z and W are independent, having $\operatorname{Cov}(s(Z), t(W)) = 0$ for all bounded continuous functions s and t does actually imply independence. Unfortunately, going over all bounded continuous functions is not tractable. Instead, Gretton et al. (2004) proposed evaluating $\sup_{s\in\mathcal{F},\, t\in\mathcal{G}} \operatorname{Cov}(s(Z), t(W))$, where $\mathcal{F}, \mathcal{G}$ are universal Reproducing Kernel Hilbert Spaces (RKHS). They proved that if the RKHS is universal and the supremum is zero, then Z and W are independent. Next, in Gretton et al. (2005a), the authors introduced a quantity $\operatorname{HSIC}(Z, W, \mathcal{F}, \mathcal{G})$, which upper bounds the supremum, and still satisfies that $\operatorname{HSIC}(Z, W, \mathcal{F}, \mathcal{G}) = 0$ if and only if Z and W are independent.

Defining $\operatorname{HSIC}(Z, W, \mathcal{F}, \mathcal{G})$ in its most general form requires some background in the theory of RKHS. Here, however, we only present the properties that are essential for the reading of this paper. In particular, any RKHS $\mathcal{F}$ is associated with a kernel $k(\cdot)$ and a mapping function ϕ from $\mathbb{R}$ to $\mathcal{F}$, such that $k(x_1, x_2) = \langle\phi(x_1), \phi(x_2)\rangle_{\mathcal{F}}$. Next, let $\mathcal{F}$, $\mathcal{G}$ be two RKHS with associated kernels $k(\cdot)$, $l(\cdot)$ respectively. Let (Z', W') be an independent copy of (Z, W) with an identical joint distribution. Gretton et al. (2005a) showed that $\operatorname{HSIC}(Z, W, \mathcal{F}, \mathcal{G})$, defined as the Hilbert-Schmidt norm of the cross-covariance operator of Z and W, can be equivalently expressed in terms of the two kernels as follows, assuming all expectations exist,

$$\begin{aligned}\operatorname{HSIC}(Z, W, \mathcal{F}, \mathcal{G}) &= \mathbb{E}_{ZWZ'W'}[k(Z, Z')l(W, W')] \\ &\quad + \mathbb{E}_{ZZ'}[k(Z, Z')] \cdot \mathbb{E}_{WW'}[l(W, W')] \\ &\quad - 2\mathbb{E}_{ZW}\Big[\mathbb{E}_{Z'}[k(Z, Z')]\mathbb{E}_{W'}[l(W, W')]\Big].\end{aligned}$$

Given a sample of n independent pairs $S \equiv \{(z_i, w_i)\}_{i=1}^n$ drawn from the joint distribution P_{zw}, one can estimate the HSIC by the following formula with complexity $O(n^2)$,

$$\widehat{\operatorname{HSIC}}(Z, W, \mathcal{F}, \mathcal{G}) = (n-1) - 2\operatorname{Tr}(KLHL),$$

where $K, L, H \in \mathbb{R}^{n\times n}, K_{ij} \equiv k(z_i, z_j), L_{ij} \equiv l(w_i, w_j), H \equiv I_{n\times n} - (n-1)\mathbf{1}\mathbf{1}^{\mathbf{T}}$, with $\mathbf{1}$ denoting a vector of all ones, and $\operatorname{Tr}(A)$ denotes the trace of a matrix A. Furthermore, as proven in Gretton et al. (2005a), this estimator is nearly unbiased,

$$\mathbb{E}_S[\widehat{\operatorname{HSIC}}(Z, W, \mathcal{F}, \mathcal{G})] = \operatorname{HSIC}(Z, W, \mathcal{F}, \mathcal{G}) + O(n^{-1}),$$

with $\mathbb{E}_S$ denoting expectation over the sample S.

3. The Sparse Log-Bivariate Density Classifier

To motivate the construction of the sparse log-bivariate classifier, let us first consider the optimal log-likelihood ratio based classifier, when the distributions of both classes $y \in \{-1, +1\}$ are forest-structured, possibly with different graphs $G_y = (V_y, E_y)$. This is described in the following lemma, which follows directly from Eq. (3).

Lemma 1 *If both classes are forest-structured then their log likelihood ratio is*

$$\log\left(\frac{p_{+1}(\mathbf{x})}{p_{-1}(\mathbf{x})}\right) = \sum_{(i,j)\in E_{+1}} \log p_{+1}(x_i, x_j) + \sum_{i\in V_{+1}} (1 - \mathit{deg}_{+1}(i)) \cdot \log p_{+1}(x_i)$$
$$- \sum_{(i,j)\in E_{-1}} \log p_{-1}(x_i, x_j) - \sum_{i\in V_{-1}} (1 - \mathit{deg}_{-1}(i)) \cdot \log p_{-1}(x_i),$$

where $\mathit{deg}_y(i)$ is the degree of x_i in the graph of class y.

An immediate corollary of Lemma 1 is that if both classes follow a forest-structured graphical model then the optimal decision boundary is a *linear* combination of their univariate and bivariate log-transformed densities. Specifically, let $T_o : \mathbb{R}^d \to \mathbb{R}^{d(d+1)}$ be the oracle transformation that maps an input vector $\mathbf{x} \in \mathbb{R}^d$ to a vector containing all of $\mathbf{x}$'s univariate and bivariate log density terms for both classes,

$$T_o(\mathbf{x}) := \{\log p_y(x_i, x_j) \mid y \in \{-1, +1\} \;\; i, j \in \{1, \ldots, d\}\},$$

where $p_y(x_i, x_i)$ is an alias for $p_y(x_i)$. In this notation, Lemma 1 can be restated as

$$\log \mathrm{LR}(\mathbf{x}) = \mathbf{w}_o^{\mathrm{T}} T_o(\mathbf{x}),$$

where the coefficients of $\mathbf{w}_o$ depend on the graph structures of both classes. It follows that the Bayes classifier has the form

$$\hat{y}(\mathbf{x}) = \mathrm{sign}(\mathbf{w}_o^{\mathrm{T}} T_o(\mathbf{x}) - b_o). \tag{4}$$

Under the forest model, the coefficients in $\mathbf{w}_o$ that correspond to the bivariate densities are all either $0, +1$ or -1, depending on the existence of edges in the respective graphical models. Similarly, the coefficients that multiply the univariate densities are integers that depend on the degrees of the corresponding variables in the graphs. The intercept b_o depends on the class imbalance and misclassification costs. Note that by the forest assumption, there are only $O(d)$ non-zero coefficients in the vector $\mathbf{w}_o$. As its length is $O(d^2)$, it is thus *sparse*.

To derive the SLB classifier, we relax the forest-structure constraints, and simply assume that the optimal decision boundary can be well approximated by a sparse linear combination of the univariate and bivariate log-densities,

$$\mathbf{w}^{\mathrm{T}} T_o(\mathbf{x}) - b.$$

To construct such a sparse vector $\mathbf{w}$, we first exclude bivariate densities that correspond to nearly independent pairs of variables. Specifically, we measure the strength of dependence between all pairs of variables using their empirical HSIC measure, as defined in Section 2.2,

Algorithm 1 The Sparse Log-Bivariate Density Classifier (SLB)

```
Input: Sample $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$ of feature vectors $\mathbf{x} \in \mathbb{R}^d$ and labels $y \in \{-1, +1\}$.
Step 1: For each class $y \in \{+1, -1\}$, estimate statistical dependencies $\widehat{\mathrm{HSIC}}$ between all pairs of variables $i, j \in \{1, \dots, d\}$ and $y \in \{-1, +1\}$ and filter out weakly-dependent pairs of variables.
Step 2: Compute all univariate density estimates $\widehat{p}_y(x_i)$, and the bivariate density estimates $\widehat{p}_y(x_i, x_j)$ for the variable pairs that were not filtered in the previous step.
Step 3: Fit a linear SVM $(\widehat{\mathbf{w}}, \widehat{b})$ to the transformed samples $\{(\widehat{T}(\mathbf{x}_i), y_i)\}$, where $\widehat{T}$ is the log-density transformation defined in Eq. (5).
Output: Classifier given by $\mathbf{x} \mapsto \mathrm{sign}(\widehat{\mathbf{w}}^{\mathrm{T}} \widehat{T}(\mathbf{x}) - \widehat{b})$.
```

and filter out bivariate densities with estimated HSIC values smaller than a given threshold λ. We then fit a linear SVM to learn the coefficients of all the univariate log-densities and bivariate log-densities that correspond to the remaining pairs. Since T_o is typically unknown, we replace it by a vector of estimated densities:

$$\widehat{T}(\mathbf{x}) := \left(\log \widehat{p}_y(x_i, x_j)\right)_{y \in \{-1,+1\} \;\; i,j \in \{1,\dots,d\}} \tag{5}$$

Lastly, setting λ in a principled manner can be done in several ways. One approach is to run a cross-validation procedure and choose the threshold whose corresponding classifier has the smallest misclassification error. Another option is to use a multiple hypothesis testing procedure on all pairwise $\widehat{\mathrm{HSIC}}$ values. The `SLB` classifier is outlined in Algorithm 1.

4. Theoretical Analysis

In this section we study the theoretical properties of the linear SVM solution $(\widehat{\mathbf{w}}, \widehat{b})$ based on the estimated densities $\widehat{p}_y(x_i, x_j)$ given a training set of n i.i.d. samples $D = \{(\mathbf{x}_\ell, y_\ell)\}_{\ell=1}^n$. Concretely we prove that under certain conditions, the risk of the linear SVM solution $(\widehat{\mathbf{w}}, \widehat{b})$ based on the estimated densities $\widehat{p}_y(x_i, x_j)$ converges to that of the optimal SVM classifier based on the exact densities $p_y(x_i, x_j)$.

To simplify the theoretical analysis we make the following two assumptions: (i) the input to the SVM classifier consists of all the univariate and bivariate log-transformed densities, without the HSIC-based filtering step. This can be viewed as an analysis of the large-sample regime where the HSIC is able to detect all of the dependent pairs. (ii) the n training samples were split into two disjoint sets D_0 and D_1 of sizes n_0 and n_1, respectively, with $n_0 + n_1 = n$. The set D_0 is used to estimate the densities $\widehat{p}_y(x_i)$ and $\widehat{p}_y(x_i, x_j)$ for each class $y \in \{-1, +1\}$ and the set D_1 to construct the SVM classifier. Specifically, given the estimated densities, we apply the feature map $\widehat{T} : \mathbb{R}^d \to \mathbb{R}^{d(d+1)}$ of Eq. (5) to the n_1 samples in D_1 and construct a linear classifier using the transformed samples $\{(\widehat{T}(\mathbf{x}_\ell), y_\ell)\}_{\ell=n_0+1}^n$. We add a feature that is identically equal to 1 to the feature map T, so that the intercept b can be subsumed into $\mathbf{w}$. The class label predicted by a linear classifier $\mathbf{w}$ applied to a feature-mapped sample is thus equal to $\mathrm{sign}(\mathbf{w}^{\mathrm{T}} T(\mathbf{x}))$.

In the rest of this section, we define several risk functions and then formulate `SLB` as an empirical risk minimizer. The classification error rate, or 0-1 risk, of the classifier $(\mathbf{w}, T)$ is

$$R_{01}(\mathbf{w}, T) := \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \left[\mathbb{1}(y \neq \operatorname{sign}(\mathbf{w}^{\mathrm{T}} T(\mathbf{x})))\right].$$

From Eq. (4) it follows that $\mathbf{w}_o$ minimizes $R_{01}(\mathbf{w}, T_o)$. The empirical 0-1 risk is the average loss over the n_1 samples in D_1,

$$\widehat{R}_{01}(\mathbf{w}, T) := \frac{1}{n_1} \sum_{\ell=n_0+1}^{n_1} \left[\mathbb{1}(y_\ell \neq \operatorname{sign}(\mathbf{w}^{\mathrm{T}} T(\mathbf{x}_\ell)))\right].$$

Minimizing this risk with respect to $\mathbf{w}$ is computationally difficult (Ben-David et al., 2003). To circumvent this difficulty, one may consider instead the risk and empirical risk with respect to the hinge loss $\phi(z) := \max\{0, 1-z\}$,

$$R_\phi(\mathbf{w}, T) := \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \left[\phi(y\mathbf{w}^{\mathrm{T}} T(\mathbf{x}))\right], \qquad \widehat{R}_\phi(\mathbf{w}, T) := \frac{1}{n_1} \sum_{\ell=n_0+1}^{n_1} \phi(y_\ell \mathbf{w}^{\mathrm{T}} T(\mathbf{x}_\ell)).$$

The hinge loss is a convex function and therefore can be minimized efficiently. Since the hinge-loss bounds the 0-1 loss from above, minimizing the hinge loss is a way to control for the misclassification error. A common formulation of the SVM classifier minimizes the empirical hinge loss with an additional Tikhonov regularization term. For the transformed data set $\{(\widehat{T}(\mathbf{x}_\ell), y_\ell)\}_{\ell=n_0+1}^{n}$ this is

$$\widehat{\mathbf{w}}_{\text{Tikhonov}} := \operatorname*{argmin}_{\mathbf{w}} \left(\widehat{R}_\phi(\mathbf{w}, \widehat{T}) + \lambda\|\mathbf{w}\|^2\right), \tag{6}$$

where $\lambda > 0$ controls the regularization strength (see for example Section 15.2 of Ben-David and Shalev-Shwartz (2014)). For `SLB`, we consider instead the SVM solution with Ivanov regularization,

$$\widehat{\mathbf{w}} := \operatorname*{argmin}_{\mathbf{w}:\|\mathbf{w}\|_2 \leq B} \widehat{R}_\phi(\mathbf{w}, \widehat{T}), \tag{7}$$

where the radius B controls the complexity of the hypothesis class. Both the Tikhonov form in Eq. (6) and the Ivanov form in Eq. (7) share the same regularization path (Oneto et al., 2015). Namely, for every choice of Tikhonov regularization parameter λ there is an Ivanov regularizer B_λ which yields the same solution. Finally, we denote by $\mathbf{w}_\phi$ the SVM population minimizer using the oracle feature map T_o

$$\mathbf{w}_\phi := \operatorname*{argmin}_{\mathbf{w}:\|\mathbf{w}\|_2 \leq B} R_\phi(\mathbf{w}, T_o). \tag{8}$$

In the rest of this section, we prove that as the sample size tends to infinity, the risk of the `SLB` classifier $R_\phi(\widehat{\mathbf{w}}, \widehat{T})$ converges to the oracle risk $R_\phi(\mathbf{w}_\phi, T_o)$ and give high probability bounds on their difference.

4.1 Convergence of the Data-dependent Feature Map

We begin by studying the convergence rate of the estimated feature map $\widehat{T}$ to the oracle map T_o. For simplicity, we use the bivariate notation $\widehat{p}_y(x_i, x_i)$ to also denote univariate density estimates $\widehat{p}_y(x_i)$. For our theoretical analysis we need 3 assumptions on the class conditional densities and their estimation procedures:

Assumption 1 (bounded density) *There are constants* $p_{\min},\ p_{\max} > 0$ *such that for all* $\mathbf{x} \in \Omega$ *and all* y, i, j *we have that* $p_{\min} \leq p_y(x_i, x_j) \leq p_{\max}$.

We make this assumption for the sake of simplicity. See Remark 2 at the end of the next section for a discussion on how it may be relaxed.

Assumption 2 (density estimator consistency) *There is a function* $U(n_0)$ *that satisfies* $U(n_0) \xrightarrow{n_0 \to \infty} 0$ *such that the following uniform bound holds with high probability,*

$$\sup_{\mathbf{x} \in \Omega, y, i, j} |\widehat{p}_y(x_i, x_j) - p_y(x_i, x_j)| < U(n_0).$$

This assumption holds under various conditions on the underlying density and method of density estimation. For example, for estimating $\beta-$Hölder bivariate densities, the minimax bound $U(n) = n^{-\beta/(2\beta+2)}$ holds (up to log factors) with high probability, both for kernel density estimation (Liu et al., 2011; Jiang, 2017) and for k-nearest-neighbor density estimation (Dasgupta and Kpotufe, 2014).

Assumption 3 *The following expectation is finite,*

$$E(n_0, d) := \mathop{\mathbb{E}}_{\mathbf{x}} \|\widehat{T}(\mathbf{x}) - T_o(\mathbf{x})\|_2 = \mathop{\mathbb{E}}_{\mathbf{x}} \sqrt{\textstyle\sum_{y,i,j} (\log \widehat{p}_y(x_i, x_j) - \log p_y(x_i, x_j))^2}$$

and $E(n_0, d) \xrightarrow{n_0 \to \infty} 0$.

Remark 1 *A potential problem with non-parametric kernel density estimators is that* $\widehat{p}_y(x_i, x_j)$ *may be zero (or even negative) for some values of* (x_i, x_j). *In that case,* $\log \widehat{p}_y(x_i, x_j)$ *is undefined. However, the uniform consistency in Assumption 2 combined with the lower bound* $p_y(x_i, x_j) \geq p_{\min}$ *in Assumption 1 means that the probability of this event vanishes as* $n \to \infty$. *To avoid the issue altogether, one may construct an adjusted density estimator of the form* $\widetilde{p}_y(x_i, x_j) := \max\{\widehat{p}_y(x_i, x_j), c\}$ *for some* $c \in (0, p_{\min})$. *It is easy to check that Assumptions 1, 2 and 3 still hold for the adjusted estimator* $\widetilde{p}$.

The following lemma, proved in the appendix quantifies the convergence rate of the feature map $\widehat{T}$ to the oracle map.

Lemma 2 *Under Assumption 1,*

$$\|T_o(\mathbf{x})\|_2 < \sqrt{d(d+1)}L, \quad \textit{where} \quad L := \max\{|\log p_{\min}|, |\log p_{\max}|\}.$$

Under assumptions 1 and 2, the following bound holds w.h.p.

$$\sup_{\mathbf{x} \in \Omega} \|\widehat{T}(\mathbf{x}) - T_o(\mathbf{x})\|_2 < \frac{2}{p_{\min}} \sqrt{d(d+1)} U(n_0).$$

Corollary 3 *Under assumptions 1 and 2, w.h.p.*

$$\sup_{\mathbf{x} \in \Omega} \|\widehat{T}(\mathbf{x})\|_2 < \sqrt{d(d+1)} \left(L + \frac{2}{p_{\min}} U(n_0) \right).$$

4.2 Convergence of the SVM Risk to the Oracle Risk

In this section we present our main theoretical result. The proofs appear in the appendix. We begin with a technical lemma that bounds the difference in risks of using the exact log densities T_o compared to using the estimated log densities $\widehat{T}$.

Lemma 4 *Under Assumption 3, for any $\mathbf{w} \in \mathbb{R}^{d(d+1)}$,*

$$|R_\phi(\mathbf{w}, \widehat{T}) - R_\phi(\mathbf{w}, T_o)| \leq \|\mathbf{w}\|_2 E(n_0, d).$$

This holds for the empirical risk as well,

$$|\widehat{R}_\phi(\mathbf{w}, \widehat{T}) - \widehat{R}_\phi(\mathbf{w}, T_o)| \leq \|\mathbf{w}\|_2 E(n_0, d).$$

We now show that for the hinge loss, the population risk of the estimated $\widehat{\mathbf{w}}$ of Eq. (7), converges in probability to the population risk of the oracle SVM solution $\mathbf{w}_\phi$, defined in Eq. (8). In other words, $R_\phi(\widehat{\mathbf{w}}, \widehat{T})$ converges to $R_\phi(\mathbf{w}_\phi, T_o)$.

Theorem 5 *Let $\widehat{T}$ be the feature map constructed using n_0 samples and let $\widehat{\mathbf{w}}$ be the SVM classifier of* `SLB`*, trained using n_1 samples. Then, under Assumptions 1,2 and 3, the following bound holds w.h.p.*

$$R_\phi(\widehat{\mathbf{w}}, \widehat{T}) - R_\phi(\mathbf{w}_\phi, T_o) < B\left(E(n_0, d) + \sqrt{d(d+1)\tfrac{\ln n_1}{n_1}}\left(2L + \tfrac{2}{p_{\min}} U(n_0)\right)\right).$$

The terms $E(n_0, d)$ and $U(n_0)$ are due to errors in the univariate and bivariate density estimates, whereas the term $\sqrt{\ln n_1 / n_1}$ is an SVM generalization term.

Remark 2 *The lower bound on the density $p_y(x_i, x_j) \geq p_{\min} > 0$ in Assumption 1 may be relaxed by dividing the (possibly infinite) support of the distribution to the following domain and its complement,*

$$\Omega(p_{\min}) := \left\{\mathbf{x} \middle| \forall y, i, j : p_y(x_i, x_j) \geq p_{\min}\right\}. \tag{9}$$

Our theoretical analysis holds on the domain $\Omega(p_{\min})$. The remaining part contributes at most a factor of $B \cdot \Pr[\mathbf{x} \notin \Omega(p_{\min})]$ to the excess SVM risk in Theorem 5. Taking $p_{\min} \to 0$ at an appropriate rate as the number of samples tends to infinity yields risk-consistency. Further distributional assumptions, such as sub-Gaussianity may be used to tighten the bounds on the excess risk.

5. Experimental Results

We compare the predictive accuracy of `SLB` to several widely used classifiers both on synthetic data and on various public data sets. Our method may be tuned to improve performance on a specific domain by considering such things as different kernels and bandwidths in the density estimation step, as well as different misclassification costs and penalty terms for the norm of the vector of coefficients in the construction of the SVM classifier. However, we refrain from doing so and instead demonstrate SLB's competitiveness across a broad range of data sets. The seven classification methods we tested appear below. All compared methods were implemented in the `R` programming language. When required, univariate or bivariate densities were estimated by the `ks` package.

- **SLB**: Our sparse log-bivariate density classifier. We estimated all pairwise HSIC values using the `dHSIC` package with a Gaussian kernel. The SVM classifier was constructed by the `e1071` package with a default parameter $\lambda = 1/2$.
- **LU**: Log-univariate density classifier. This classifier trains an SVM model using only the log univariate densities of the d variables of each class, as its input features. This is similar to the approach of Fan et al. (2016), that trained a logistic regression model rather than an SVM, on the same features.
- **5NN**: 5 nearest-neighbor classifier, as implemented in the `FNN` package.
- **SVM**: Support vector machine classifier with the radial basis function kernel, as implemented in the `e1071` package. Note that by default $\lambda = \frac{1}{2}$.
- **NB**: Naive Bayes classifier. Here we used our own implementation.
- **TAN**: Tree-augmented naive Bayes. We used the `mpmi` package to estimate the mutual information between each pair of variables, and the `optrees` package to estimate the maximum spanning tree of each class.
- **RF**: Random forest classifier as implemented in the `randomForest` package, with 50 trees and the default number of $\sqrt{d}$ randomly selected variables at each split.

5.1 Synthetic Data Experiments

5.1.1 Forest Structured Bayesian Networks

We begin with the simple setting where the underlying distribution of each class is a forest structured Bayesian network with $d = 20$ variables. We consider the following configurations that differ in the number of training samples, class imbalance proportion and complexity of the Conditional Probability Distributions (CPDs) that parametrize the Bayesian network. Concretely, our simulations follow a 5×2^2 factorial design:

- **Sample size of the training data:** $n = 200, 400, 600, 800, 1000$.
- **Class imbalance:** either balanced 50%-50% or imbalanced 75%-25%.
- **Complexity of the Conditional Probability Distributions:** Let Z be the single parent of X in the forest. We consider the following two cases for $P(X|\mathbf{Z} = z)$:
  (i) a simple setting where $X|\mathbf{Z} = z \sim N(z, 1)$.
  (ii) a more complex setting in which

  $$X|\mathbf{Z} = z \sim \begin{cases} t + z & \text{w.p. } \frac{1}{2} \\ \frac{1}{2}N(z+1,1) + \frac{1}{2}N(z-1,1) & \text{w.p. } \frac{1}{2}, \end{cases}$$

  where t is a student-t distribution with 5 degrees of freedom.
  In both cases, root variables follow a standard Gaussian distribution.

Marginals	Class labels	n	SLB	LU	TAN	NB	RF	SVM	5-NN
Normal	Balanced	200	15.4 ± 2.98	21.2 ± 3.64	$\mathbf{8.17 \pm 3.78}$	23 ± 3.75	15 ± 2.54	11.9 ± 2.36	13.8 ± 2.65
		400	9.37 ± 2.34	19.4 ± 3.5	$\mathbf{6.68 \pm 3.88}$	21.8 ± 3.66	11.6 ± 2.08	8.07 ± 2.02	11.4 ± 2.56
		600	6.91 ± 1.76	17.6 ± 3.45	$\mathbf{4.98 \pm 2.64}$	20.3 ± 3.36	9.73 ± 1.55	6.34 ± 1.42	9.91 ± 2.1
		800	$\mathbf{5.82 \pm 1.82}$	18.3 ± 3.19	$\mathbf{5.56 \pm 4.02}$	21 ± 3.14	8.87 ± 1.67	$\mathbf{5.46 \pm 1.54}$	8.89 ± 2.22
		1000	$\mathbf{5.11 \pm 1.36}$	17.3 ± 3.46	$\mathbf{4.66 \pm 2.87}$	20.7 ± 3.44	8.23 ± 1.63	$\mathbf{4.84 \pm 1.16}$	8.37 ± 1.93
	Unbalanced	200	20.4 ± 4.7	24.6 ± 4.57	$\mathbf{11.4 \pm 5.51}$	26.1 ± 4.41	30.3 ± 4.6	25.8 ± 4.83	24.3 ± 3.89
		400	13.2 ± 3.71	22.7 ± 4.88	$\mathbf{7.37 \pm 4.32}$	25.1 ± 4.29	25.5 ± 4.08	18.7 ± 3.43	19.2 ± 3.72
		600	9.48 ± 2.98	21.5 ± 4.04	$\mathbf{6.52 \pm 3.96}$	23.7 ± 4.25	22 ± 3.65	14.7 ± 2.79	17.2 ± 3.71
		800	7.43 ± 2	21.3 ± 3.99	$\mathbf{6.13 \pm 4.05}$	23.4 ± 3.74	20.3 ± 2.71	12.7 ± 2.43	15.9 ± 2.95
		1000	$\mathbf{7.04 \pm 2.12}$	20.5 ± 3.6	$\mathbf{6.56 \pm 4.54}$	22.6 ± 3.86	18.9 ± 2.61	11.6 ± 2.23	15 ± 3.18
Complex	Balanced	200	33.3 ± 3.23	33.3 ± 3.23	35.5 ± 3.16	33.1 ± 3.38	$\mathbf{30.8 \pm 2.89}$	37.2 ± 2.92	38.2 ± 2.38
		400	31.1 ± 3.71	30.7 ± 3.97	32.8 ± 3.85	30.8 ± 3.66	$\mathbf{26.8 \pm 2.87}$	33.9 ± 3.18	36.1 ± 2.95
		600	29.4 ± 3.34	29.5 ± 3.46	30.8 ± 4.93	29.7 ± 3.37	$\mathbf{25.5 \pm 2.49}$	31.7 ± 3.19	34.8 ± 2.91
		800	27.7 ± 3.13	28.1 ± 3.32	28.4 ± 4.83	27.9 ± 3.34	$\mathbf{24.2 \pm 2.28}$	29.3 ± 2.62	33.5 ± 2.64
		1000	27.6 ± 3.33	28.1 ± 3.65	27.8 ± 4.79	28.1 ± 3.68	$\mathbf{23.8 \pm 2.41}$	28.7 ± 2.61	33.1 ± 2.65
	Unbalanced	200	$\mathbf{37.1 \pm 3.72}$	$\mathbf{37.1 \pm 3.86}$	$\mathbf{38 \pm 3.41}$	37.9 ± 3.35	43.2 ± 3.13	48.3 ± 1.57	43.8 ± 1.75
		400	34.9 ± 3.71	35.3 ± 3.87	$\mathbf{33.3 \pm 4.05}$	35.6 ± 3.6	39.7 ± 3.12	45.9 ± 2.69	41.3 ± 2.48
		600	$\mathbf{32.7 \pm 3.66}$	$\mathbf{33 \pm 3.72}$	$\mathbf{31.2 \pm 4.65}$	33.6 ± 3.58	37.6 ± 3.07	43.7 ± 2.93	39.9 ± 2.26
		800	$\mathbf{32.2 \pm 3.59}$	33.2 ± 3.75	$\mathbf{30.5 \pm 4.71}$	33.5 ± 3.65	36.4 ± 3.13	42.5 ± 2.99	39.3 ± 2.06
		1000	31.6 ± 3.68	32.9 ± 3.93	$\mathbf{29 \pm 5.04}$	33.1 ± 3.7	35.7 ± 3.32	41.4 ± 3.03	38.7 ± 2.68

Table 1: Misclassification error rates for the various classifiers, in the forest setting, averaged across 100 replicate data sets generated at each factor level combination. The classifier with the best performance, as well as classifiers that achieve comparable error rates are printed in boldface (see main text for details).

For each combination of factor levels, 100 Bayesian networks were generated. A training data set drawn from each model was used to construct the different classifiers. Finally, we evaluate the predictive accuracy on an independent test set of 1000 i.i.d. samples generated from the same model. In each test set, half of the observations were generated from each class. Average misclassification error rates from the average 100 replicates are shown in Appendix B, Figures 1–4. The full results appear in Table 1. In addition, in each setting, we compare the classifier with the best performance to each of the other classifiers, using a two-sided Wilcoxon signed-rank test. Since there are 6 comparisons per configuration setting, we use Bonferroni-correction with $\alpha = 0.05/6$. The classifier with the best performance, as well as classifiers that were found to be statistically compatible (i.e., the null hypothesis is not rejected) are printed in the table in boldface. As shown, under the Gaussian setting, `TAN` either achieves the minimum error rate or it is comparable to the best performing classifier. This is somewhat to be expected since the `TAN` classifier is based on the Chow-Liu algorithm (Chow and Liu, 1968), which is consistent under the *Gaussian-tree* setting. In our *Gaussian-forest* setting, we expect the tree constructed by `TAN` to include all true forest edges and a few additional spurious ones that make the forest a connected graph. Nevertheless, in all the 5×2^2 configurations, as the sample size increases, the gap between `SLB` and `TAN` decreases. It is also apparent that both `SLB` and `TAN` models outperform all other classifiers under the Gaussian-imbalanced configuration.

Marginals	Common structure	Class labels	n	SLB-	SLB	LU	TAN	NB	RF	SVM	5-NN
Normal	None	Balanced	200	12.3 ± 3.23	**11.7 ± 2.92**	15.2 ± 4.33	16.8 ± 6.81	28.7 ± 6.45	15 ± 3.68	17.8 ± 2.66	17.1 ± 3.09
			400	7.53 ± 2.1	**7.3 ± 2.04**	13.2 ± 3.83	12 ± 6.83	26.4 ± 6.13	10.3 ± 2.82	12.5 ± 2.74	13 ± 2.76
			600	5.4 ± 1.53	**5.24 ± 1.46**	12.9 ± 3.54	12.9 ± 7.59	26.7 ± 5.72	8.92 ± 2.17	10.4 ± 2.46	10.9 ± 2.3
			800	**4.24 ± 1.31**	**4.21 ± 1.32**	11.7 ± 3.3	9.96 ± 5.22	24.4 ± 6.18	7.47 ± 2.24	8.83 ± 2.49	9.26 ± 2.09
			1000	**3.73 ± 1.22**	**3.67 ± 1.24**	11.7 ± 3.79	10.8 ± 7.81	24.8 ± 6.82	6.5 ± 1.91	7.75 ± 2.33	8.89 ± 2.47
		Unbalanced	200	**15.1 ± 4.52**	**15.1 ± 3.73**	17.1 ± 4.27	**16.2 ± 8.36**	29.6 ± 6.7	25.7 ± 6.07	30.1 ± 5.86	26.1 ± 5.24
			400	10.4 ± 2.61	**8.87 ± 2.51**	14.9 ± 3.95	14.4 ± 8.41	27.8 ± 6.93	20.3 ± 5.39	25.3 ± 4.21	20.5 ± 4.43
			600	7.69 ± 2.12	**6.31 ± 1.72**	13.3 ± 3.94	12.7 ± 7.66	25.8 ± 5.67	16.9 ± 4.44	20.9 ± 3.81	17.6 ± 4.01
			800	6.21 ± 1.78	**4.99 ± 1.77**	13.3 ± 4.02	10.5 ± 6.41	26.1 ± 5.41	16 ± 4.21	19 ± 2.93	16.2 ± 3.31
			1000	5.29 ± 1.54	**.4.11 ± 1.18**	13 ± 4.09	10.7 ± 6.8	25 ± 5.01	13.9 ± 3.87	18.2 ± 3.45	14.4 ± 3.44
	$\frac{1}{3}$ Shared	Balanced	200	14.8 ± 3.71	**14.3 ± 3.59**	17.9 ± 4.57	18.7 ± 8.01	31.2 ± 6.77	17.6 ± 4.31	20.3 ± 3.71	21.4 ± 4.26
			400	**9.56 ± 2.59**	**9.42 ± 2.53**	15.5 ± 4.93	16.5 ± 7.97	29 ± 7.08	12.6 ± 3.42	14.8 ± 3.03	15.9 ± 3.52
			600	**7.02 ± 1.79**	**6.92 ± 1.74**	14.6 ± 4.24	13.8 ± 7.56	27.7 ± 6.55	10.5 ± 2.62	11.7 ± 3.05	13.7 ± 3.13
			800	**5.91 ± 1.73**	**5.82 ± 1.69**	13.7 ± 4.21	14.1 ± 7.37	26.7 ± 6.21	9.28 ± 2.8	10.7 ± 2.45	12.5 ± 2.98
			1000	**5.59 ± 1.85**	**5.57 ± 1.9**	14.2 ± 4.03	14.4 ± 7.91	27.6 ± 6.39	8.78 ± 2.77	9.61 ± 2.56	12.1 ± 3.15
		Unbalanced	200	19 ± 5.44	**17.7 ± 5.18**	21.4 ± 5.66	**21.2 ± 9.18**	32.9 ± 6.1	29.5 ± 6.45	35.2 ± 6.87	31 ± 4.89
			400	12 ± 3.52	**11.6 ± 3.33**	18.8 ± 5.04	19.6 ± 9.23	31.2 ± 6.95	24 ± 5.6	28.3 ± 5.26	25.6 ± 4.79
			600	9.16 ± 2.87	**8.97 ± 2.82**	18.1 ± 5.73	17.1 ± 8.19	29.7 ± 6.59	21.1 ± 5.89	24.7 ± 5.29	23 ± 4.79
			800	7.96 ± 2.24	**7.8 ± 2.16**	18 ± 5.4	17.1 ± 8.3	30.2 ± 5.85	20 ± 5.04	22.9 ± 4.03	21.3 ± 3.95
			1000	**6.77 ± 2.2**	**6.67 ± 2.21**	17.7 ± 5.45	16 ± 8.13	29.8 ± 6.65	18.4 ± 5.49	21 ± 4.28	20 ± 4.73
Complex	None	Balanced	200	14 ± 3.29	**13.5 ± 3.13**	17.9 ± 4.71	**16.9 ± 7.79**	30.1 ± 6.54	16.6 ± 3.78	18.7 ± 3.33	18.9 ± 3.47
			400	8.54 ± 1.84	**8.32 ± 1.86**	15.3 ± 3.71	13.6 ± 6.91	27.7 ± 5.61	11.8 ± 2.69	13.3 ± 2.67	14.6 ± 2.45
			600	**6.37 ± 1.65**	**6.24 ± 1.69**	15.1 ± 3.93	13.2 ± 7.49	27.1 ± 5.46	10 ± 2.58	10.4 ± 2.57	12.2 ± 2.73
			800	**5.17 ± 1.44**	**5.13 ± 1.42**	14.5 ± 3.63	13.4 ± 7.52	27 ± 5.52	8.86 ± 2.12	9.01 ± 2.37	11.1 ± 2.24
			1000	**4.55 ± 1.15**	**4.48 ± 1.13**	13.5 ± 3.96	11.2 ± 7.45	25.8 ± 6.1	7.76 ± 1.95	8.08 ± 2.31	10.3 ± 2.13
		Unbalanced	200	18.3 ± 5	**17.1 ± 4.98**	21 ± 4.73	**18.4 ± 7.8**	31.4 ± 5.6	29.4 ± 5.07	32.4 ± 5.79	29.3 ± 4.43
			400	10.9 ± 2.76	**10.6 ± 2.83**	18.7 ± 4.88	14.8 ± 7.45	29.5 ± 5.15	23.6 ± 5.19	25.8 ± 4.17	23.4 ± 3.91
			600	**8.18 ± 2.29**	**8 ± 2.34**	18.1 ± 5.08	13 ± 8.36	29 ± 5.8	20.6 ± 4.82	23.2 ± 4.31	20.6 ± 4.09
			800	7.04 ± 1.8	**6.89 ± 1.82**	18.3 ± 4.5	15 ± 8.81	28.8 ± 5.6	19.8 ± 4.2	20.9 ± 3.38	19.5 ± 3.5
			1000	**5.77 ± 1.62**	**5.69 ± 1.61**	17.4 ± 4.74	13.1 ± 7.98	27.7 ± 5.63	17.8 ± 4.18	18.9 ± 2.84	18.1 ± 3.81
	$\frac{1}{3}$ Shared	Balanced	200	17.3 ± 4.02	**16.5 ± 3.86**	20.9 ± 5.39	19.7 ± 7.28	32.8 ± 6.65	19.4 ± 4.26	21.1 ± 3.89	22.5 ± 3.8
			400	11.5 ± 2.94	**11.1 ± 2.92**	19 ± 4.63	18.3 ± 9	30.9 ± 5.29	14.6 ± 3.48	15.6 ± 3.53	18.3 ± 3.73
			600	**8.64 ± 2.6**	**8.58 ± 2.57**	16.7 ± 4.61	15.8 ± 8.03	28.6 ± 6.21	11.8 ± 3.06	13 ± 3.11	16.1 ± 3.51
			800	**7.43 ± 2.22**	**7.36 ± 2.17**	17.1 ± 4.77	15.3 ± 7.53	29.6 ± 6.41	11 ± 2.96	11.2 ± 2.79	14.7 ± 3.04
			1000	**6.38 ± 1.73**	**6.34 ± 1.7**	15.9 ± 4.18	14.9 ± 8.07	28 ± 5.81	9.83 ± 2.42	10.1 ± 2.57	13.5 ± 3.02
		Unbalanced	200	20.8 ± 5.64	**19.7 ± 5.58**	24.1 ± 6.42	**23.3 ± 8.8**	35.1 ± 5.91	32.6 ± 6.77	37.1 ± 7.2	32.9 ± 5
			400	14.2 ± 3.76	**13.7 ± 3.59**	21.6 ± 5.17	19.9 ± 9.47	32.5 ± 6	26.8 ± 5.86	29.7 ± 5.18	27.4 ± 4.23
			600	11.5 ± 2.95	**11.2 ± 2.78**	21.6 ± 5.24	19.6 ± 8.94	32.5 ± 5.23	24.4 ± 4.92	26.5 ± 4.72	25.5 ± 4.09
			800	**9.22 ± 2.59**	**9.18 ± 2.63**	20.5 ± 5.99	17.8 ± 9.27	30.8 ± 6.51	22 ± 5.65	23.6 ± 4.14	23.1 ± 4.47
			1000	**8.26 ± 2.73**	**8.18 ± 2.69**	20 ± 5.79	17.2 ± 8.01	30.8 ± 5.53	20.6 ± 5.7	21.7 ± 4.25	21.9 ± 4.49

Table 2: Misclassification error rates for the various classifiers, in the general Bayesian network setting, averaged across 100 replicate data sets generated at each factor level combination. The classifier with the best performance, as well as classifiers that achieve comparable error rates are printed in boldface.

Marginals	Class labels	n	SLB (HSIC)	SLB (Pearson)
Gaussian	Balanced	200	0.12 ± 0.03	0.12 ± 0.03
		400	0.07 ± 0.02	0.08 ± 0.02
		600	0.05 ± 0.01	0.06 ± 0.02
		800	$\mathbf{0.04 \pm 0.01}$	0.05 ± 0.02
		1000	0.04 ± 0.01	0.05 ± 0.01
	Unbalanced	200	0.16 ± 0.05	0.16 ± 0.04
		400	0.10 ± 0.03	0.11 ± 0.03
		600	0.07 ± 0.02	0.08 ± 0.02
		800	0.06 ± 0.02	0.07 ± 0.02
		1000	0.06 ± 0.02	0.06 ± 0.02
Complex	Balanced	200	0.14 ± 0.03	0.15 ± 0.03
		400	0.08 ± 0.02	0.10 ± 0.02
		600	0.06 ± 0.02	0.08 ± 0.02
		800	0.05 ± 0.01	0.07 ± 0.01
		1000	$\mathbf{0.05 \pm 0.01}$	0.06 ± 0.01
	Unbalanced	200	0.19 ± 0.05	0.19 ± 0.04
		400	0.11 ± 0.03	0.12 ± 0.03
		600	0.08 ± 0.02	0.10 ± 0.03
		800	0.07 ± 0.02	0.08 ± 0.01
		1000	0.06 ± 0.01	0.07 ± 0.01

Table 3: Misclassification error rates of the `SLB(HSIC)` and `SLB(Pearson)` classifiers, in the general Bayesian network setting, averaged across 100 replicate data sets generated at each factor level combination. Results that are significantly better are printed in boldface (see main text for details).

5.1.2 General Bayesian Networks

Next, we consider a more challenging setting where each class follows a Bayesian network in which all non-root nodes have three parents. For a r.v. X, at a non-root node, let $Par_X \equiv \{Z_1, Z_2, Z_3\}$ be its three parents. We consider the following two settings for $P(X|Par_X)$:

(i) a simple setting where $X|\mathbf{Par_X} \sim N(\sum_{Z_i \in \mathbf{Par_X}} z_i, 1)$.

(ii) a setting in which

$$X|\mathbf{Par_X} \sim \begin{cases} t + \sum_{Z_i \in \mathbf{Par_X}} z_i & \text{w.p. } \frac{1}{2} \\ \frac{1}{2}N(z_1 + z_2, 1) + \frac{1}{2}N(z_2 + z_3, 1) & \text{w.p. } \frac{1}{2}. \end{cases}$$

where t is a variable distributed as Student's t-distribution with 5 degrees of freedom. As before, we set $d = 20$ and the root variable follows a standard Gaussian distribution. To accommodate further variety, we extend our factorial design and allow for partial common structure between the Bayesian networks of the two classes. In the simpler setting, the two networks are constructed independently of each other. In the more challenging setting, roughly a third of the nodes share the same parent sets and associated conditional probability distributions. Overall our simulations now follow a 5×2^3 factorial design. We repeat the same protocol as in the forest setting, with 100 Bayesian networks generated for each configuration. Classifiers are evaluated on an independent test set of 1000 samples. As before, in these test sets, half of the observations were generated from each class.

To highlight the importance of removing bivariate densities corresponding to nearly independent variables, we also consider a variant of SLB in which we skip the first step of Algorithm 1. The resulting classifier denoted SLB-, constructs a linear SVM using *all* log bivariate densities.

Average misclassification error rates from the 100 replicates are shown in Appendix B, Figures 5–12. The full results, including standard deviations, are displayed in Table 2. Again, we compare the classifier with the best performance to each of the other classifiers, using a two-sided Wilcoxon signed-rank test, at the Bonferroni-corrected $\alpha = 0.05/7$ level (since now we have a total of 8 classifiers). The classifier with the best performance, as well as classifiers that were found to be statistically compatible are printed in boldface.

As can be seen, in all configurations, SLB achieves the lowest misclassification error rate, whereas TAN attains compatible results in only five (out of 40) configurations, all five having Gaussian distributions. SLB is significantly more accurate than SLB-, particularly for small sample sizes. As sample size increases, the estimated bivariate densities of independent variables become closer to the product of the univariate densities. In this case, their logarithm is approximately the sum of the log of the corresponding univariate densities. As the latter are already incorporated into the SVM model, the effect of removing such pairwise densities is negligible, and SLB and SLB- achieve similar error rates.

Lastly, in this work we used the Hilbert-Schmidt independence criterion to decide if two variables are nearly independent. Given a d-dimensional sample of size n, computing the HSIC for all pairs of variables takes $O(d^2n^2)$ time (Gretton et al., 2004). This can be computationally expensive for large data sets. Therefore, one may prefer other measures of independence that are faster to compute, even if they do not enjoy the nice theoretical properties of HSIC. For example, Pearson's correlation coefficient, Spearman's ρ, etc. To quantify the potential impact of this change, we consider a variant of SLB that uses Pearson's correlation coefficient for measuring pairwise variable dependence. We focus on the setting of independent Bayesian networks and compare the performance of the two variants, denoted by SLB(HSIC) and SLB(Pearson). Full results, including standard deviations, are displayed in Table 3. We use a two-sided Wilcoxon signed-rank test at significance level $\alpha = 0.05$ to compare the performance. In cases where a statistically significant difference was found, the superior result is shown in boldface. As can be seen, under most configurations, the two variants show compatible results. Nevertheless, in two configurations (out of 20), SLB(HSIC) performs better.

5.2 Real Data Experiments

We next evaluate the various classifiers on 16 real data sets, publicly available at the UCI Machine Learning and Kaggle[1] Databases (Dheeru and Karra Taniskidou, 2017). The data sets we consider differ in sample size and dimensionality. In all data sets, instances with missing values were removed. In data sets where features are not commensurate, such as

[1] www.kaggle.com

Data set	d	n	(n_1, n_2)	SLB	LU	TAN	NB	RF	SVM (RBF)	5NN
Blood	4	748	$(570, 178)$	$\mathbf{40 \pm 6.3}$	49.7 ± 1.7	42.6 ± 2.9	$\mathbf{36.8 \pm 6.7}$	$\mathbf{40.1 \pm 3.7}$	44.1 ± 3.2	$\mathbf{37.4 \pm 3.9}$
Climate model sim. crashes	18	540	$(46, 494)$	$\mathbf{25.2 \pm 9.8}$	50 ± 1.3	43.8 ± 2.7	47.8 ± 3	46.7 ± 4.9	42.4 ± 4.8	$\mathbf{36.2 \pm 7.9}$
Hill-valley with out noise	100	606	$(305, 301)$	$\mathbf{38.9 \pm 7.1}$	51.3 ± 6.1	46.7 ± 3	43.9 ± 3.1	46.6 ± 3.1	47.2 ± 2.2	47.2 ± 6.6
Hill-valley with noise	100	606	$(307, 299)$	$\mathbf{37.7 \pm 2.3}$	51.8 ± 6.4	47.3 ± 61	44.3 ± 4.6	47.9 ± 3.6	47 ± 5.8	48.4 ± 2.4
Ionosphere	32	351	$(126, 225)$	$\mathbf{7.5 \pm 5}$	45.4 ± 5.4	$\mathbf{7.6 \pm 4.1}$	10.2 ± 4.3	$\mathbf{7.9 \pm 4.4}$	$\mathbf{8 \pm 4.3}$	22.1 ± 7.4
Liver	6	345	$(145, 200)$	$\mathbf{30.8 \pm 2.7}$	48.6 ± 2.8	35.5 ± 5.2	38.1 ± 4.08	$\mathbf{29.9 \pm 7.3}$	$\mathbf{33.2 \pm 4.7}$	40.6 ± 3.7
Ozone	71	1847	$(1719, 128)$	32.3 ± 2.8	48.8 ± 1.1	42.5 ± 3.9	$\mathbf{24.9 \pm 3.3}$	40.1 ± 5.1	49.6 ± 0.8	41.4 ± 3.7
Parkinson	22	195	$(48, 147)$	18.2 ± 7.1	54.7 ± 3.9	21.3 ± 5.5	20.6 ± 7.5	$\mathbf{13.1 \pm 1.7}$	27.6 ± 6.7	$\mathbf{13.6 \pm 6.3}$
Pima	8	768	$(500, 268)$	$\mathbf{28.6 \pm 3.8}$	49.2 ± 1.5	34.5 ± 6.6	$\mathbf{27.4 \pm 4.9}$	$\mathbf{26.8 \pm 2.4}$	29 ± 1.4	31.2 ± 2.6
Pulsar stars	8	17898	$(16259, 1639)$	$\mathbf{7.1 \pm 1.3}$	49.6 ± 2.6	8.6 ± 0.8	8.31 ± 0.9	8.2 ± 1.2	9.3 ± 1.2	8.8 ± 1.2
Ringnorm	20	7400	$(3664, 3736)$	$\mathbf{1.4 \pm 0.3}$	31.8 ± 1.6	2.7 ± 0.9	$\mathbf{1.4 \pm 0.3}$	3.8 ± 0.5	1.4 ± 0.3	31.7 ± 0.8
Sonar	60	208	$(111, 97)$	$\mathbf{18.1 \pm 6.5}$	52.9 ± 5.1	$\mathbf{20.5 \pm 4.9}$	26.2 ± 9.2	$\mathbf{19.6 \pm 9.8}$	$\mathbf{20.1 \pm 5.9}$	22.9 ± 2.7
Spectf	44	80	$(40, 40)$	$\mathbf{21.2 \pm 7.1}$	45 ± 21.8	$\mathbf{26.2 \pm 20.4}$	$\mathbf{22.5 \pm 7.1}$	$\mathbf{21.2 \pm 13.7}$	$\mathbf{20 \pm 11.2}$	27.5 ± 14.4
WI prog	32	198	$(151, 47)$	$\mathbf{35.7 \pm 13.5}$	52.7 ± 11.6	$\mathbf{42.9 \pm 6.9}$	$\mathbf{39.8 \pm 9.6}$	42 ± 4.9	44 ± 4.1	$\mathbf{40.3 \pm 6.3}$
WI diag	30	569	$(357, 212)$	4.52 ± 2.2	47.9 ± 4.7	11.5 ± 2.1	6.3 ± 3	4.8 ± 2.3	$\mathbf{2.9 \pm 2.4}$	4 ± 2.6
Vertebral	6	310	$(210, 100)$	23.2 ± 5.3	50 ± 1.2	$\mathbf{18.8 \pm 4.9}$	22.5 ± 3.8	$\mathbf{18.8 \pm 7.3}$	$\mathbf{19.2 \pm 7.9}$	$\mathbf{22.1 \pm 7}$

Table 4: Summary of balanced error rates on data sets from the UCI ML and Kaggle Repository. n is the total number of labeled samples, n_1 and n_2 are the number of samples in the ‘positive’ and ‘negative’ classes respectively. The classifier with the best performance, as well as classifiers that achieve comparable error rates are printed in boldface (see main text for details).

the Ozone data set in which one feature is the temperature and another is the wind speed, they were standardized prior to applying 5NN and SVM (RBF), both of which are known to be sensitive to the scale of the data.

The misclassification error was estimated by 5-fold cross-validation, with the folds sampled in a stratified manner so that they have approximately the same proportions of class labels as the full data set. For each data set, the various classifiers were learned on the same training sets and their performance was evaluated on the same test sets. We evaluated each classifier by its Balanced Error Rate (BER),

$$\mathrm{BER} = 1 - \frac{1}{2}(\mathrm{Sensitivity} + \mathrm{Specificity}).$$

Sensitivity is the true positive rate, equal to the proportion of predicted positives from the positive set. Specificity is the true negative rate, equal to the proportion of predicted negatives from the negative set. Note that the misclassification error rates for the simulated data sets in the previous subsection were estimated on a test set with an equal number of samples in each class, and hence are equivalent to the BER.

The empirical mean and standard deviation of the BER taken across the 5 cross-validation folds are given in Table 4. For each data set, the classifier with the smallest error rate and classifiers that are not significantly different from it, according to a two-sided Wilcoxon signed-rank test, appear in bold. Note that the tests were performed at Bonferroni-corrected $\alpha = 0.05/6$ level.

As shown, in 13 out of 16 data sets, `SLB` either achieves the minimum misclassification rate or is on par with the best performing model. Comparing `SLB` to `LU` and `NB` which only uses the univariate log density features, highlights the importance of including at least some of the bivariate features. Finally, note that in 15 out of 16 data sets, `SLB` achieves balanced error rates that are either better or compatible with those of the `TAN` classifier.

These results demonstrate the potential benefit of `SLB` over the `TAN` classifier that relies on a tree assumption, which may be violated in practice.

6. Discussion

In this work, we propose the sparse log-bivariate density classifier (`SLB`), a semiparametric procedure that generalizes tree and forest-based classification approaches. As in forest-based methods, `SLB` requires the estimation of only univariate and bivariate densities. However, it is more flexible and does not require learning the structure of the class conditional distributions.

In recent years, the winners of many data modeling competitions made use of extensive feature engineering, whereby many new features are generated from the original features in the data set (for example, Koren (2009); Narayanan et al. (2011)). Our work suggests that the estimated log densities $\log \widehat{p}_y(x_i)$ and $\log \widehat{p}_y(x_i, x_j)$ are interesting features to consider for a wide range of classification problems. Of course, one may choose to also keep the original features as in Fan et al. (2016) or to combine the log density features with other non-linear features.

At test time, `SLB` requires the evaluation of the estimated log densities $\widehat{T}(\mathbf{x})$ at new instances $\mathbf{x}$. Using a standard kernel density estimator, this requires to have at test time the original training data. Nevertheless, since the computation involves only 1-dimensional and 2-dimensional density estimates, there are efficient methods to approximate these estimated densities without access to the original data (Ram et al., 2009).

`SLB` uses the HSIC measure to rank the bivariate densities and applies a cross-validation procedure to select only some of them. A different approach is to directly incorporate a sparsity-inducing penalty term, into the SVM loss function, similar to the approaches presented in Neumann et al. (2005) and Zhu et al. (2004).

Acknowledgments

This work was supported in part by a Texas A&M-Weizmann research grant from Paul and Tina Gardner. Parts of this work were done while AM was at the Weizmann Institute of Science and at Tel-Aviv University. BN is incumbent of the William Petschek professorial chair in mathematics. Part of this work was done while BN was on sabbatical at the Institute for Advanced Study at Princeton. BN would like to thank the IAS and the Charles Simonyi endowment for their generous support.

Professor Clifford H. Spiegelman passed away during the final review process of this article. We would like to dedicate this paper to his memory.

Appendix A. Proofs

Proof of Lemma 2. The first part follows immediately from Assumption 1, since

$$\|T_o(\mathbf{x})\|_2^2 = \sum_{y,i,j} \log^2 p_y(x_i, x_j) < d(d+1) \max\{|\log p_{\min}|, |\log p_{\max}|\}^2.$$

We now turn to the second part of the lemma.

$$\sup_{\mathbf{x}\in\Omega} \|\widehat{T}(\mathbf{x}) - T_o(\mathbf{x})\|_2^2 = \sup_{\mathbf{x}\in\Omega} \sum_{y,i,j} (\log \widehat{p}_y(x_i, x_j) - \log p_y(x_i, x_j))^2.$$

By the mean value theorem, for any $a, b > 0$ there exists some $\xi \in [a, b]$ such that $\log b - \log a = (b - a)/\xi$. It follows that

$$\frac{|b-a|}{\max\{a,b\}} \leq |\log b - \log a| \leq \frac{|b-a|}{\min\{a,b\}}. \tag{10}$$

Hence,

$$\sup_{\mathbf{x}\in\Omega} \|\widehat{T}(\mathbf{x}) - T_o(\mathbf{x})\|_2^2 \leq \sup_{\mathbf{x}\in\Omega} \sum_{y,i,j} \left(\frac{\widehat{p}_y(x_i, x_j) - p_y(x_i, x_j)}{\min\{\widehat{p}_y(x_i, x_j), p_y(x_i, x_j)\}} \right)^2.$$

By Assumptions 1 and 2, w.h.p. $\min\{\widehat{p}_y(x_i, x_j), p_y(x_i, x_j)\} > p_{\min}/2$, therefore

$$\begin{aligned}
&\sup_{\mathbf{x}\in\Omega} \|\widehat{T}(\mathbf{x}) - T_o(\mathbf{x})\|_2^2 \\
&\quad \leq \frac{4}{p_{\min}^2} \sup_{\mathbf{x}\in\Omega} \sum_{y,i,j} \left(\widehat{p}_y(x_i, x_j) - p_y(x_i, x_j)\right)^2 \\
&\quad \leq \frac{4}{p_{\min}^2} \sum_{y,i,j} \sup_{\mathbf{x}\in\Omega} \left(\widehat{p}_y(x_i, x_j) - p_y(x_i, x_j)\right)^2 \\
&\quad < \frac{4}{p_{\min}^2} d(d+1) U^2(n_0).
\end{aligned}$$

■

Proof of Lemma 4. By definition,

$$|R_\phi(\mathbf{w}, \widehat{T}) - R_\phi(\mathbf{w}, T_o)| = \left| \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \left[\phi(y\mathbf{w}^{\mathrm{T}} \widehat{T}(\mathbf{x})) - \phi(y\mathbf{w}^{\mathrm{T}} T_o(\mathbf{x})) \right] \right|.$$

Using the triangle inequality we have

$$|R_\phi(\mathbf{w}, \widehat{T}) - R_\phi(\mathbf{w}, T_o)| \leq \mathbb{E} \left| \phi(y\mathbf{w}^{\mathrm{T}} \widehat{T}(\mathbf{x})) - \phi(y\mathbf{w}^{\mathrm{T}} T_o(\mathbf{x})) \right|.$$

Since the hinge-loss ϕ is 1-Lipschitz and $|y| = 1$, we can simplify the bound further to

$$|R_\phi(\mathbf{w}, \widehat{T}) - R_\phi(\mathbf{w}, T_o)| \leq \mathbb{E} \left| y\mathbf{w}^{\mathrm{T}} (\widehat{T}(\mathbf{x}) - T_o(\mathbf{x})) \right| = \mathbb{E} \left| \mathbf{w}^{\mathrm{T}} (\widehat{T}(\mathbf{x}) - T_o(\mathbf{x})) \right|.$$

By the Cauchy-Schwarz inequality and Assumption 3,

$$|R_\phi(\mathbf{w}, \widehat{T}) - R_\phi(\mathbf{w}, T_o)| \leq \|\mathbf{w}\|_2 \, \mathbb{E} \, \|\widehat{T}(\mathbf{x}) - T_o(\mathbf{x})\|_2 = \|\mathbf{w}\|_2 E(n_0, d).$$

The bound for the empirical risk is proved in the same manner. ■

Proof of Theorem 5. We decompose the difference of risks into 4 terms,

$$\begin{aligned} & R_\phi(\widehat{\mathbf{w}}, \widehat{T}) - R_\phi(\mathbf{w}_\phi, T_o) \\ & \quad = \left(R_\phi(\widehat{\mathbf{w}}, \widehat{T}) - \widehat{R}_\phi(\widehat{\mathbf{w}}, \widehat{T})\right) + \left(\widehat{R}_\phi(\widehat{\mathbf{w}}, \widehat{T}) - \widehat{R}_\phi(\mathbf{w}_\phi, \widehat{T})\right) \\ & \quad\quad + \left(\widehat{R}_\phi(\mathbf{w}_\phi, \widehat{T}) - \widehat{R}_\phi(\mathbf{w}_\phi, T_o)\right) + \left(\widehat{R}_\phi(\mathbf{w}_\phi, T_o) - R_\phi(\mathbf{w}_\phi, T_o)\right). \end{aligned} \tag{11}$$

We now bound each of these terms separately. To bound the first and fourth terms we apply a generalization bound for the soft-margin SVM. First, recall that by Lemma 2, $\|T_o(\mathbf{x})\|_2 < \sqrt{d(d+1)}L$. It follows from Theorem 26.12 of Ben-David and Shalev-Shwartz (2014), that the following bound is satisfied w.h.p. over $D_1 \sim \mathcal{D}^{n_1}$.

$$\sup_{\mathbf{w}:\|\mathbf{w}\| \leq B} |R_\phi(\mathbf{w}, T_o) - \widehat{R}_\phi(\mathbf{w}, T_o)| < \sqrt{d(d+1)} L B \sqrt{\frac{\ln n_1}{n_1}}. \tag{12}$$

In particular, this gives a high probability bound on the fourth term of Eq. (11). Using Corollary 3 we can similarly bound the first term of Eq. (11). w.h.p over $D \sim \mathcal{D}^{n_0+n_1}$,

$$|R_\phi(\widehat{\mathbf{w}}, \widehat{T}) - \widehat{R}_\phi(\widehat{\mathbf{w}}, \widehat{T})| < \sqrt{d(d+1)} \left(L + \frac{2}{p_{\min}} U(n_0)\right) B \sqrt{\frac{\ln n_1}{n_1}}. \tag{13}$$

To bound the second term of Eq. (11), note that $\widehat{\mathbf{w}}$ is the minimizer of $\widehat{R}_\phi(\mathbf{w}, \widehat{T})$, hence

$$\widehat{R}(\widehat{\mathbf{w}}, \widehat{T}) - \widehat{R}(\mathbf{w}_\phi, \widehat{T}) \leq 0.$$

The proof concludes by bounding the third term of Eq. (11) using Lemma 4,

$$|\widehat{R}_\phi(\mathbf{w}_\phi, \widehat{T}) - \widehat{R}_\phi(\mathbf{w}_\phi, T_o)| = BE(n_0, d).$$

■

Appendix B. Simulation Results

In this section, we present the simulation results (in the form of box plots) of the various experiments described in Section 5.

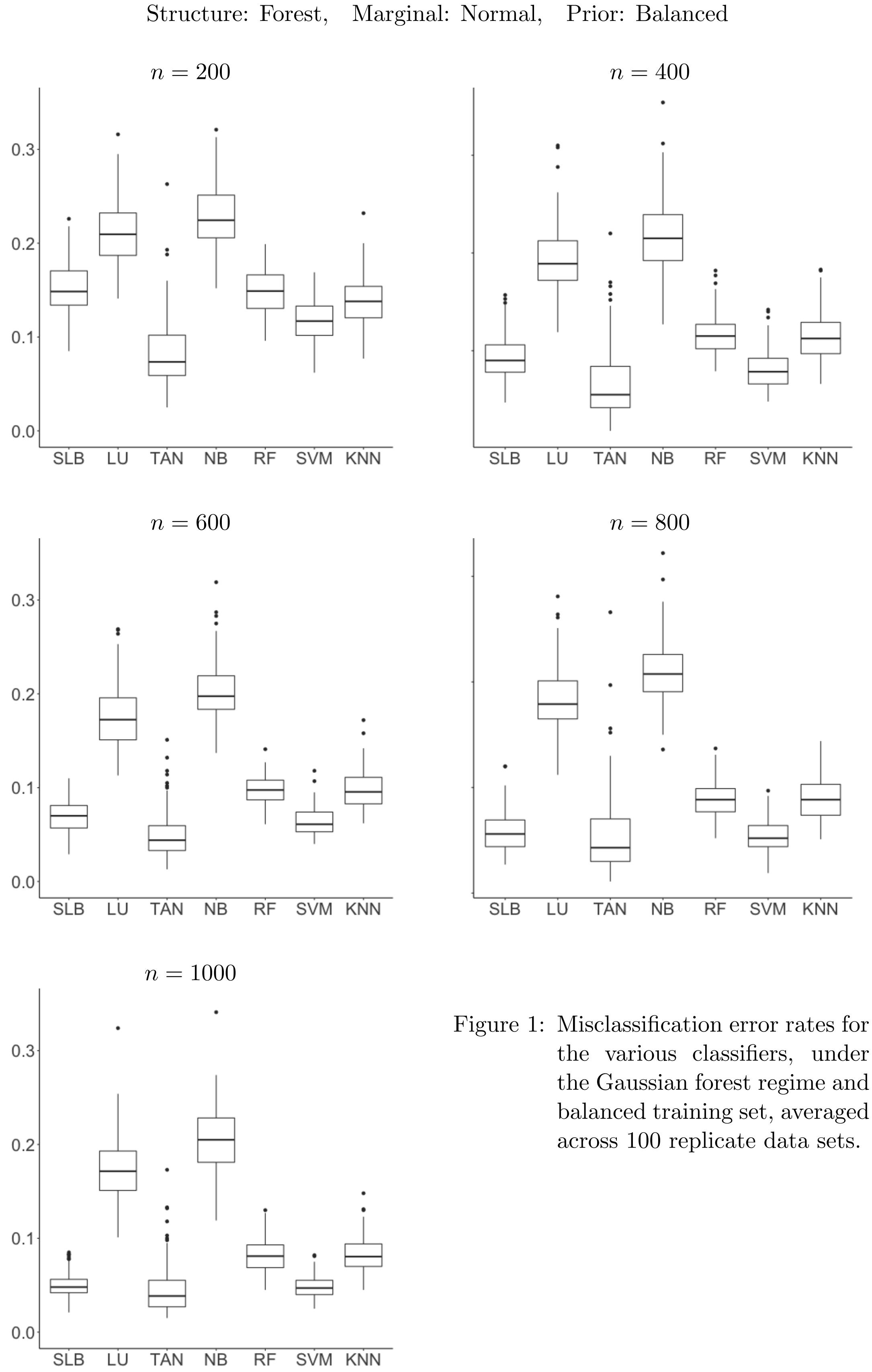


Figure 1: Misclassification error rates for the various classifiers, under the Gaussian forest regime and balanced training set, averaged across 100 replicate data sets.

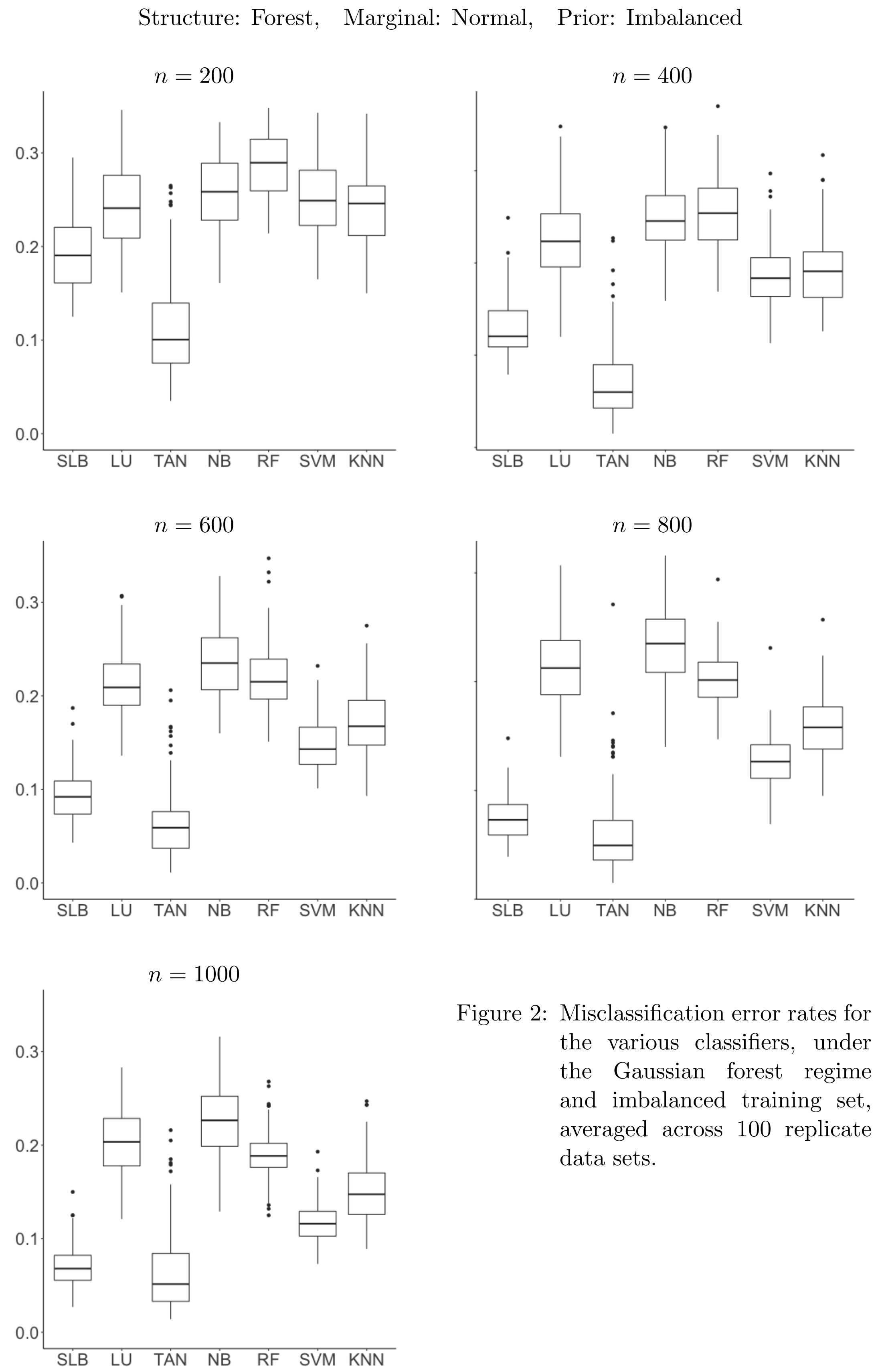


Figure 2: Misclassification error rates for the various classifiers, under the Gaussian forest regime and imbalanced training set, averaged across 100 replicate data sets.

Structure: Forest, Marginal: Complex, Prior: Balanced

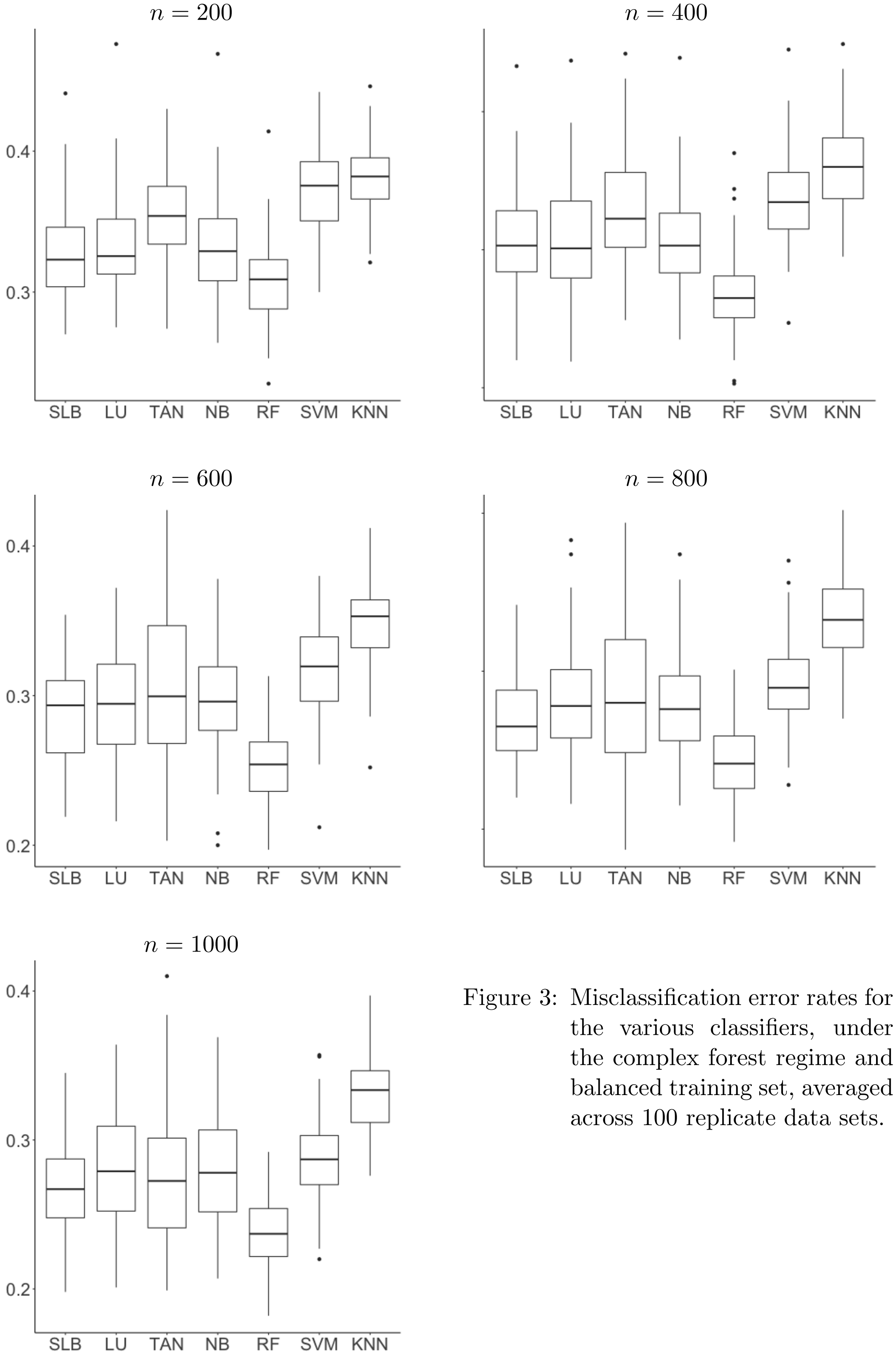
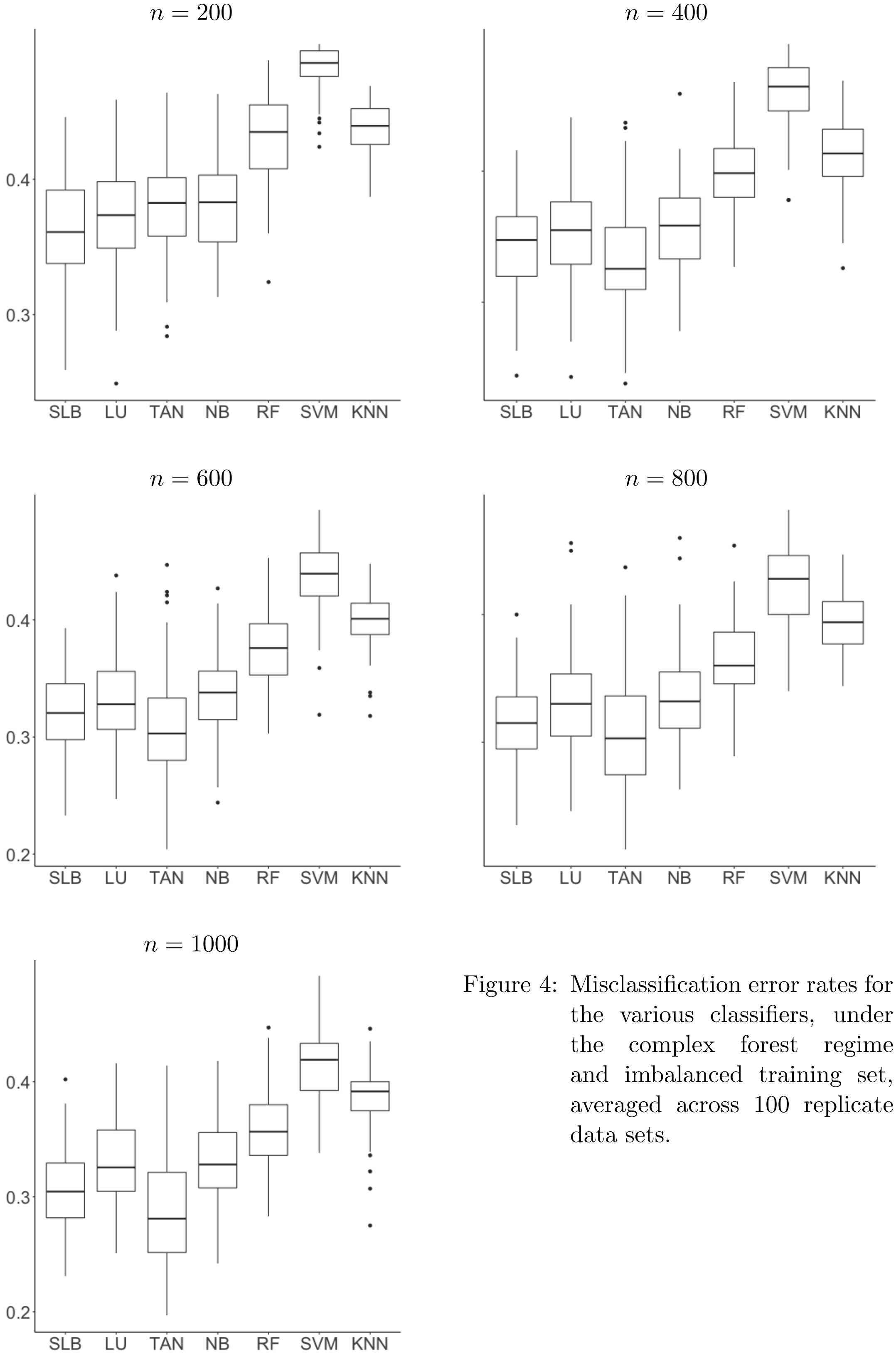


Figure 3: Misclassification error rates for the various classifiers, under the complex forest regime and balanced training set, averaged across 100 replicate data sets.

Structure: Forest,   Marginal: Complex,   Prior: Imbalanced



$n = 200$

$n = 400$

$n = 600$

$n = 800$

$n = 1000$

Figure 4: Misclassification error rates for the various classifiers, under the complex forest regime and imbalanced training set, averaged across 100 replicate data sets.

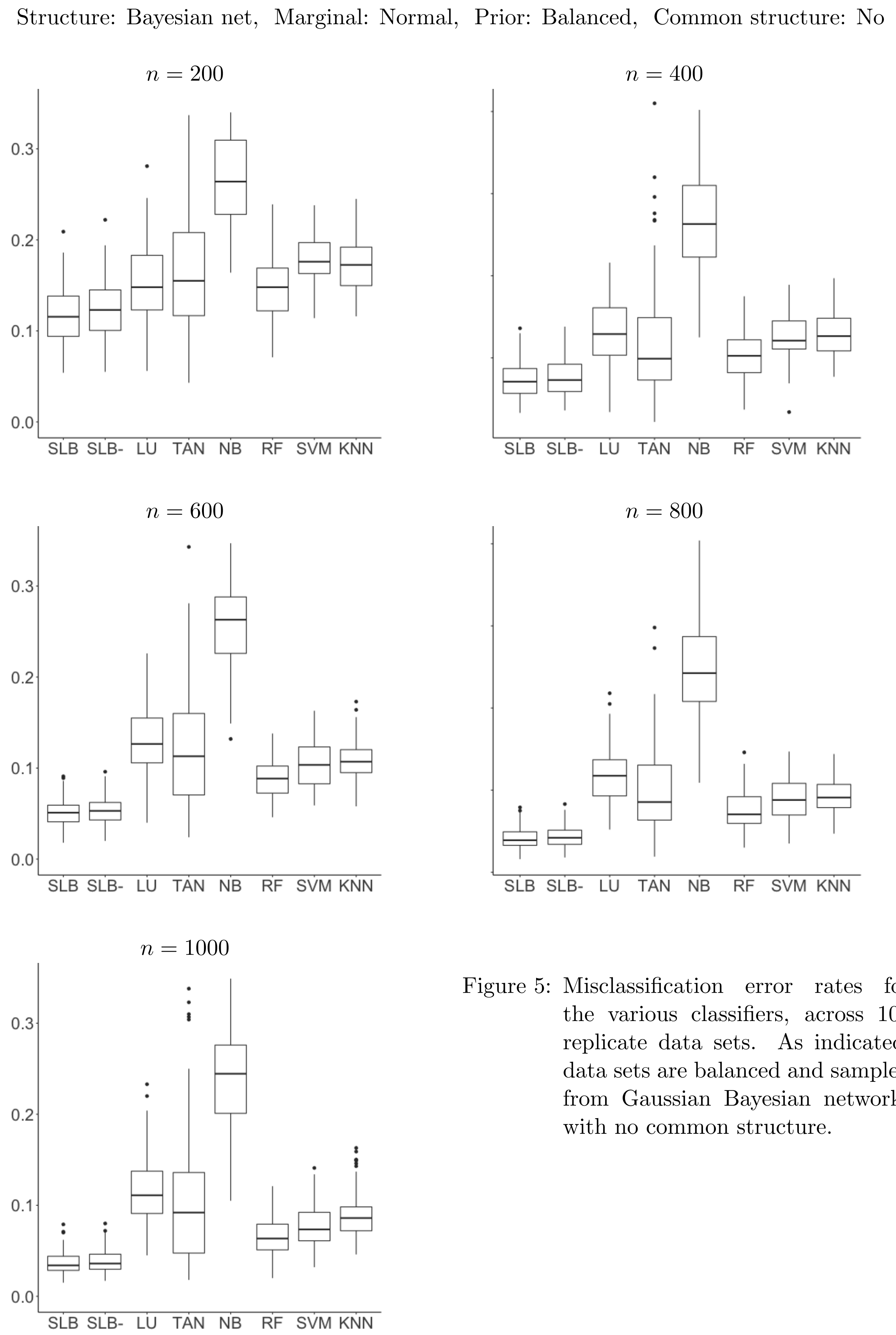


Figure 5: Misclassification error rates for the various classifiers, across 100 replicate data sets. As indicated, data sets are balanced and sampled from Gaussian Bayesian networks with no common structure.

Structure: Bayesian net, Marginal: Normal, Prior: Imbalanced, Common structure: No

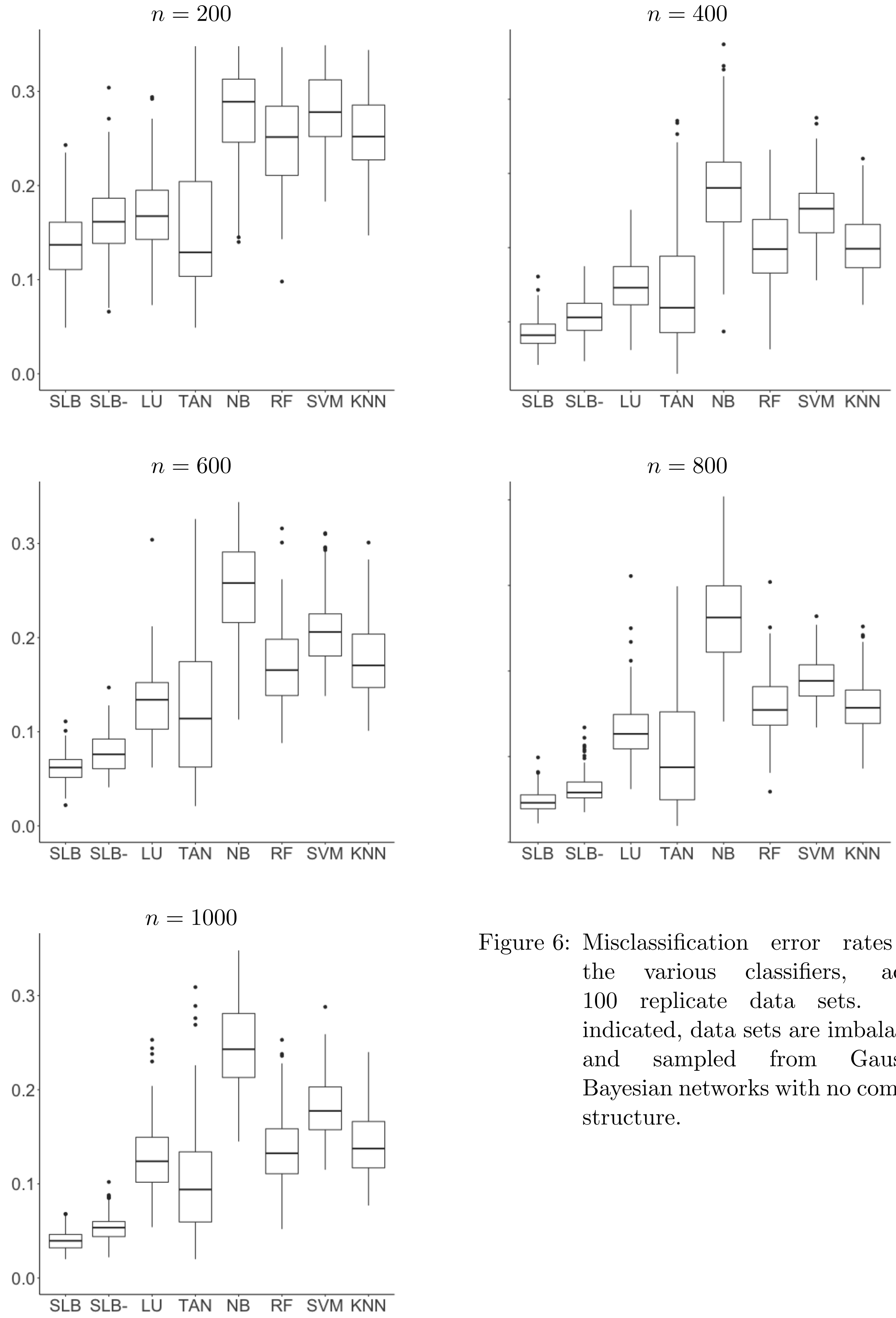
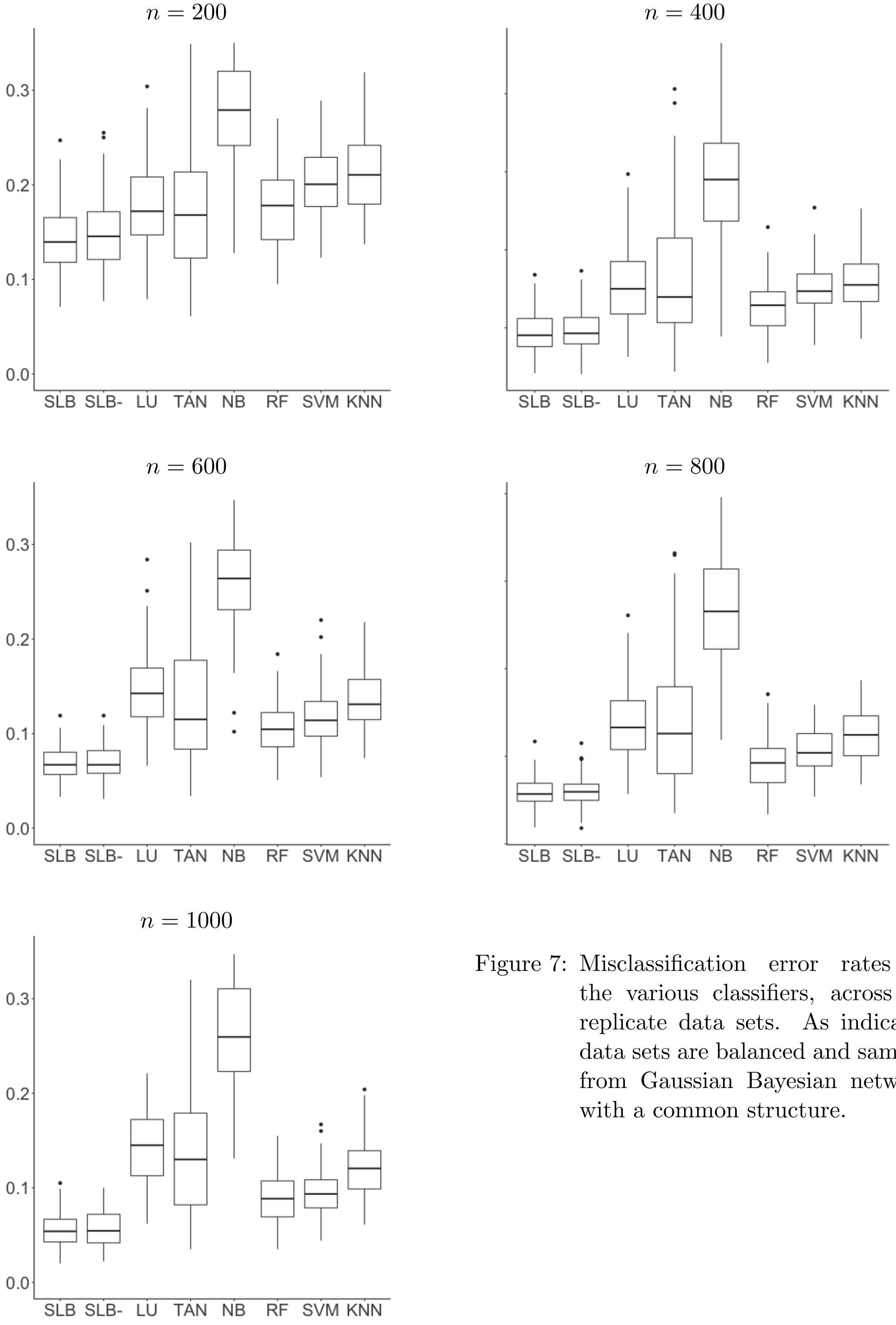


Figure 6: Misclassification error rates for the various classifiers, across 100 replicate data sets. As indicated, data sets are imbalanced and sampled from Gaussian Bayesian networks with no common structure.

Structure: Bayesian net, Marginal: Normal, Prior: Balanced, Common structure: Yes



$n = 200$

$n = 400$

$n = 600$

$n = 800$

$n = 1000$

Figure 7: Misclassification error rates for the various classifiers, across 100 replicate data sets. As indicated, data sets are balanced and sampled from Gaussian Bayesian networks with a common structure.

Structure: Bayesian net, Marginal: Normal, Prior: Imbalanced, Common structure: Yes

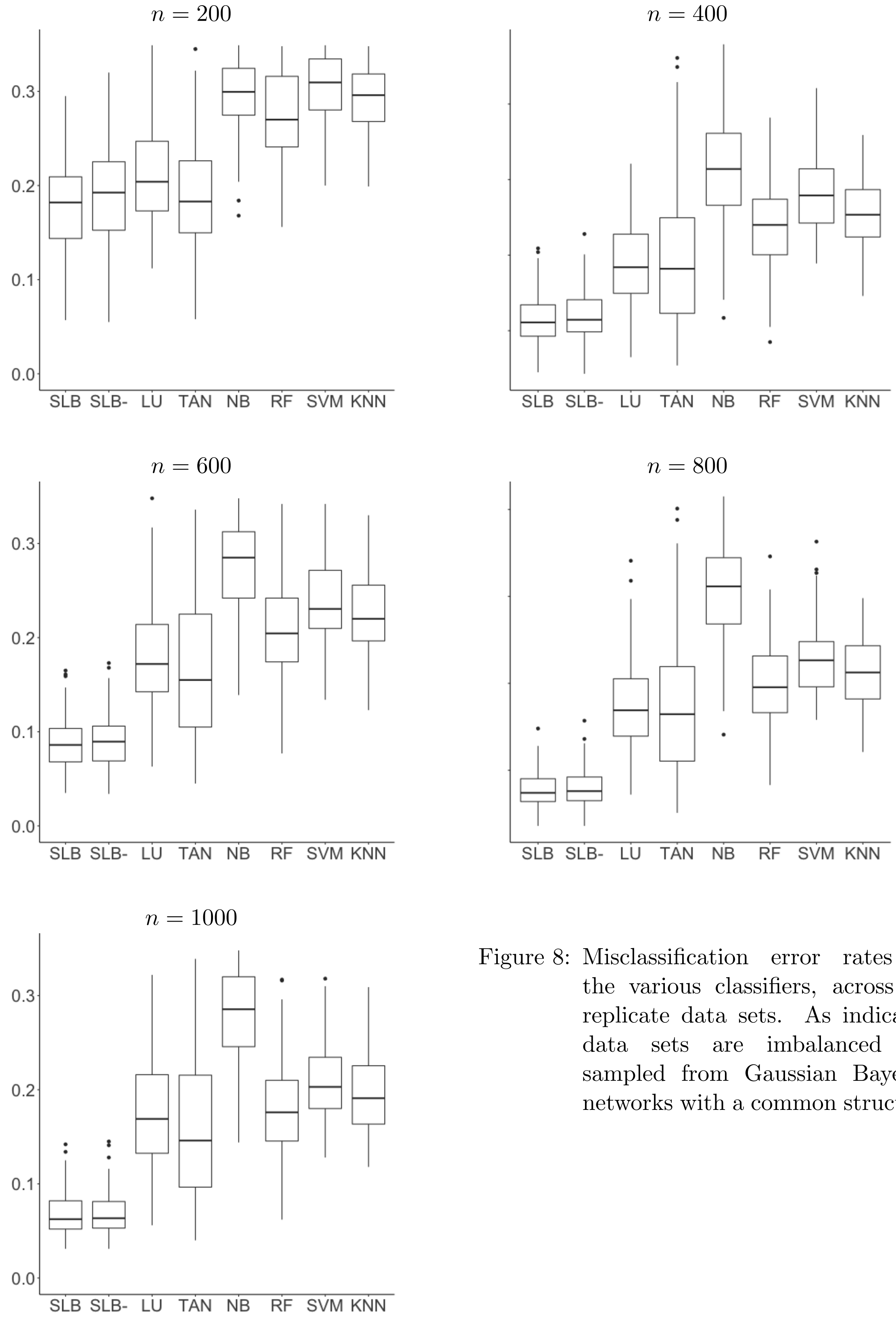
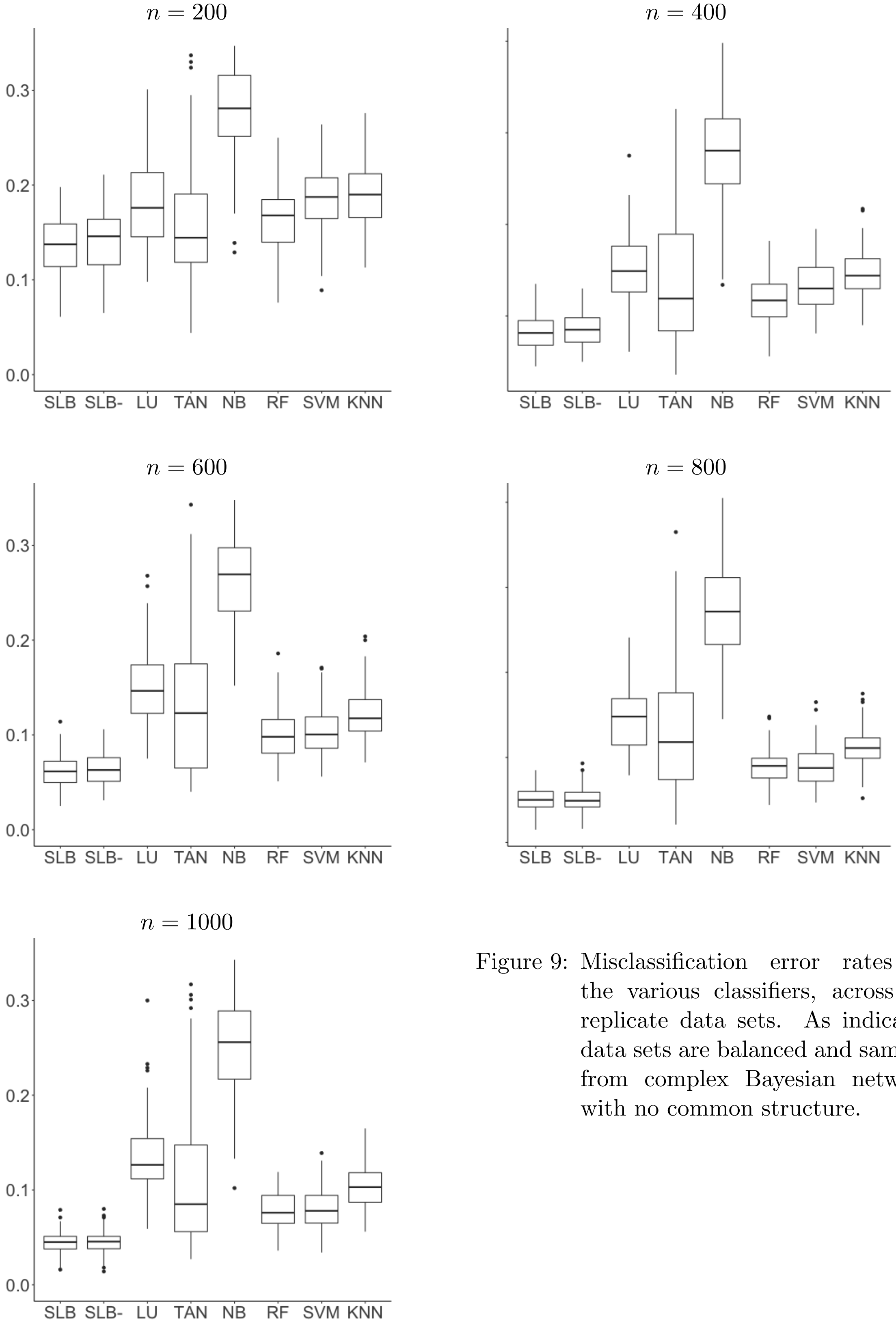


Figure 8: Misclassification error rates for the various classifiers, across 100 replicate data sets. As indicated, data sets are imbalanced and sampled from Gaussian Bayesian networks with a common structure.

Structure: Bayesian net, Marginal: Complex, Prior: Balanced, Common structure: No



$n = 200$

$n = 400$

$n = 600$

$n = 800$

$n = 1000$

Figure 9: Misclassification error rates for the various classifiers, across 100 replicate data sets. As indicated, data sets are balanced and sampled from complex Bayesian networks with no common structure.

Structure: Bayesian net, Marginal: Complex, Prior: Imbalanced, Common structure: No

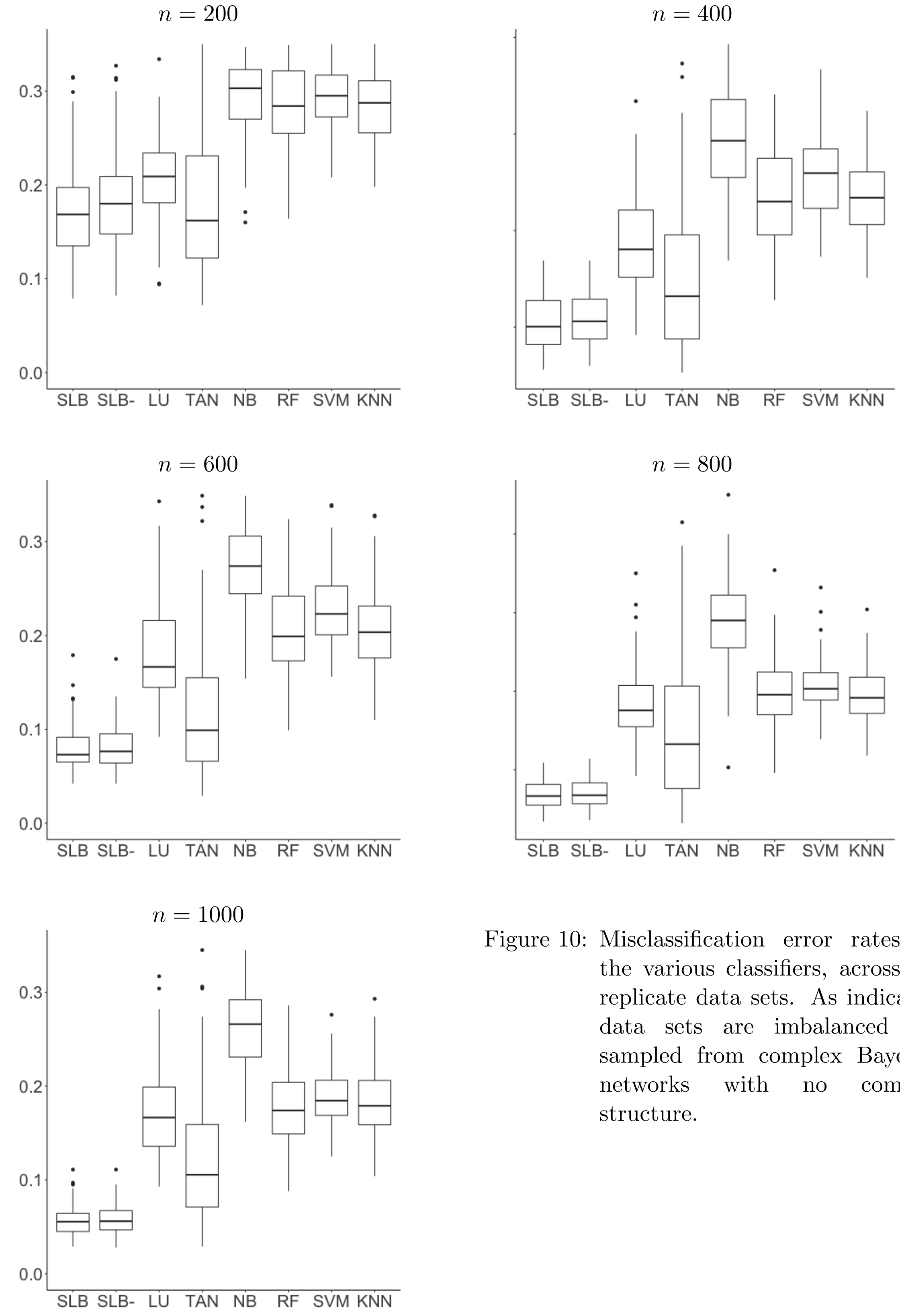
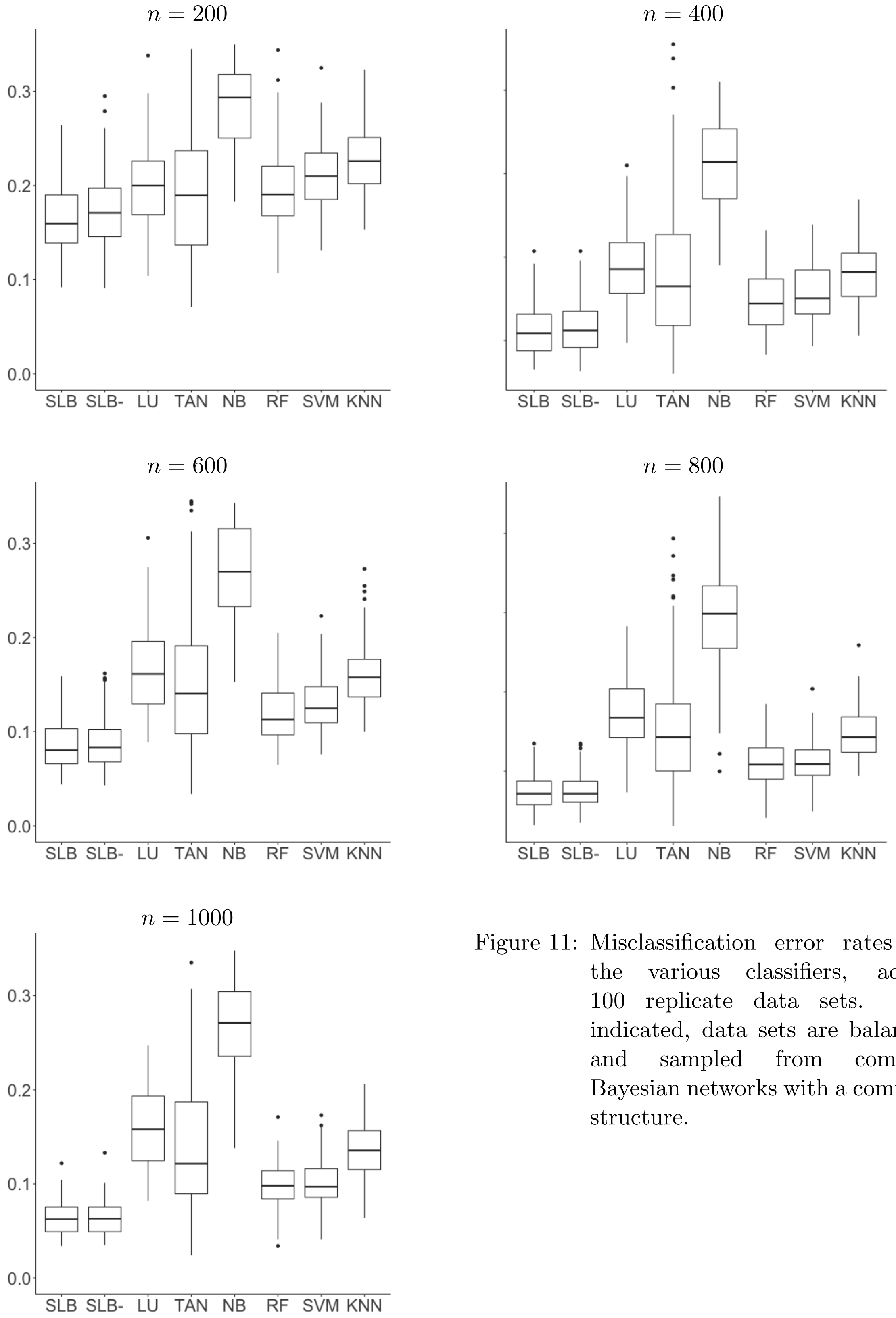


Figure 10: Misclassification error rates for the various classifiers, across 100 replicate data sets. As indicated, data sets are imbalanced and sampled from complex Bayesian networks with no common structure.

Structure: Bayesian net, Marginal: Complex, Prior: Balanced, Common structure: Yes


$n = 200$

$n = 400$

$n = 600$

$n = 800$

$n = 1000$

Figure 11: Misclassification error rates for the various classifiers, across 100 replicate data sets. As indicated, data sets are balanced and sampled from complex Bayesian networks with a common structure.

Structure: Bayesian net, Marginal: Complex, Prior: Imbalanced, Common structure: Yes

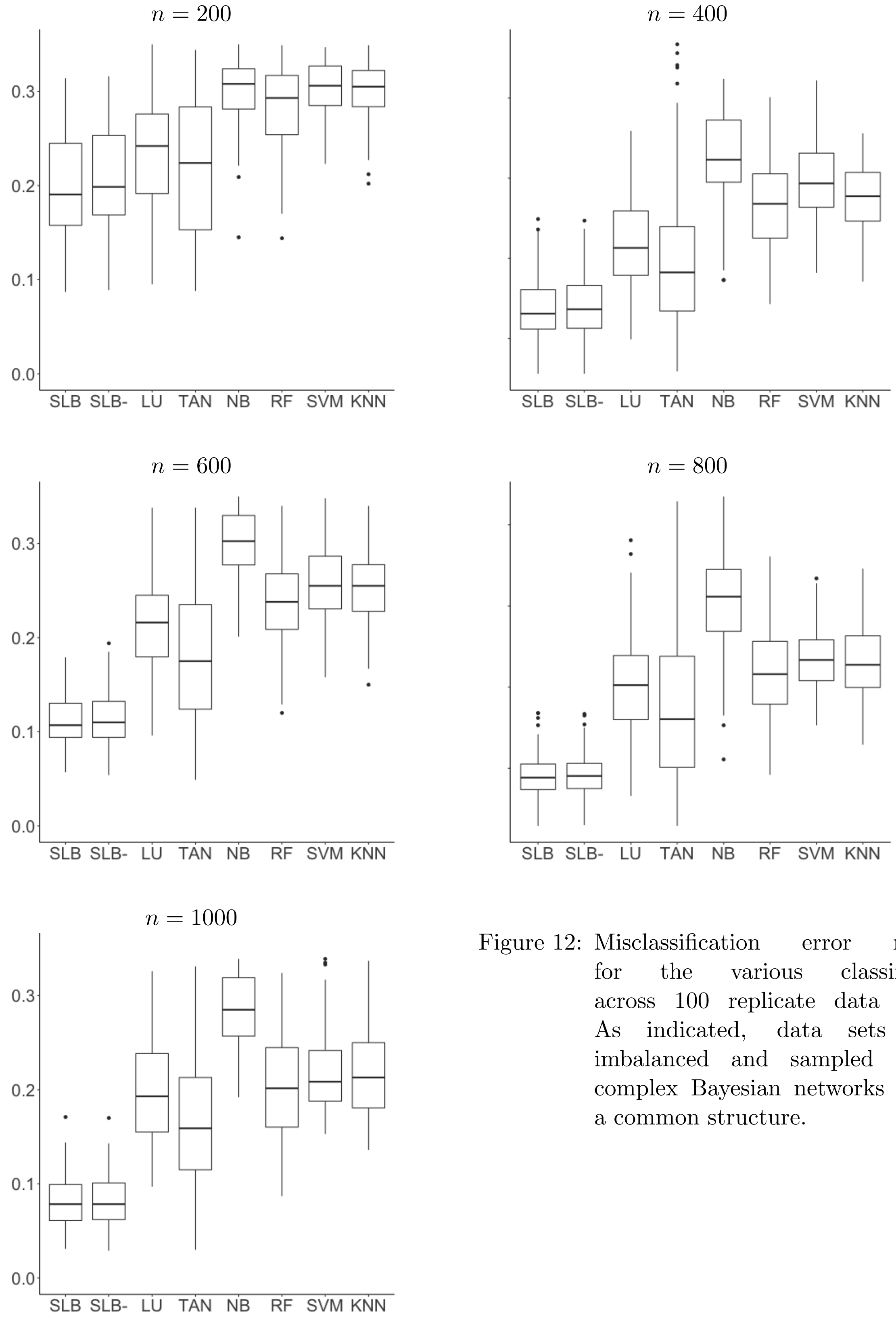


Figure 12: Misclassification error rates for the various classifiers, across 100 replicate data sets. As indicated, data sets are imbalanced and sampled from complex Bayesian networks with a common structure.

References

Shai Ben-David and Shai Shalev-Shwartz. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge university press, 2014.

Shai Ben-David, Nadav Eiron, and Philip M. Long. On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences*, 66(3):496–514, 2003.

Stefano Beretta, Mauro Castelli, Ivo Gonçalves, Roberto Henriques, and Daniele Ramazzotti. Learning the structure of Bayesian networks: A quantitative assessment of the effect of different algorithmic schemes. *Complexity*, 2018, 2018.

C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

C. K. Chow and T. J. Wagner. Consistency of an estimate of tree-dependent probability distributions (corresp.). *IEEE Transactions on Information Theory*, 19(3):369–371, 1973.

Sanjoy Dasgupta and Samory Kpotufe. Optimal rates for k-nn density and mode estimation. In *Neural Information Processing Systems (NIPS)*, pages 2555–2563, 2014.

Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017.

Richard O Duda and Peter E Hart. Pattern recognition and scene analysis, 1973.

Gal Elidan. Copula Network Classifiers (CNCs). In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 22, pages 346–354, 2012.

Jianqing Fan, Yang Feng, Jiancheng Jiang, and Xin Tong. Feature augmentation via nonparametrics and selection (FANS) in high-dimensional classification. *Journal of the American Statistical Association*, 111(513):275–287, 2016.

Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2):131–163, 1997.

Arthur Gretton, Alexander Smola, Olivier Bousquet, Ralf Herbrich, Bernhard Schölkopf, and NK Logothetis. Behaviour and convergence of the constrained covariance. Technical report, Max Planck Institute for Biological Cybernetics, 2004.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *International Conference on Algorithmic Learning Theory (ALT)*, pages 63–77. Springer, 2005a.

Arthur Gretton, Ralf Herbrich, Alexander Smola, Olivier Bousquet, and Bernhard Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6: 2075–2129, 2005b.

Daniel Grossman and Pedro Domingos. Learning Bayesian network classifiers by maximizing conditional likelihood. In *International Conference on Machine Learning (ICML)*, page 46. ACM, 2004.

Heinrich Jiang. Uniform convergence rates for kernel density estimation. In *International Conference on Machine Learning (ICML)*, volume 70, pages 1694–1703, 2017.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Yehuda Koren. The BellKor solution to the Netflix grand prize. *Netflix prize documentation*, 2009.

John Lafferty, Han Liu, and Larry Wasserman. Sparse nonparametric graphical models. *Statistical Science*, 27(4):519–537, 2012.

Pat Langley, Wayne Iba, and Kevin Thompson. An analysis of Bayesian classifiers. In *National Conference on Artificial Intelligence (AAAI)*, volume 90, pages 223–228, 1992.

Steffen L. Lauritzen. *Graphical Models*. Clarendon Press, 1996.

Jason D. Lee and Trevor J. Hastie. Learning the structure of mixed graphical models. *Journal of Computational and Graphical Statistics*, 24(1):230–253, 2015.

Han Liu, Min Xu, Haijie Gu, Anupam Gupta, John Lafferty, and Larry Wasserman. Forest density estimation. *Journal of Machine Learning Research*, 12:907–951, 2011.

Ofer Meshi, Elad Eban, Gal Elidan, and Amir Globerson. Learning max-margin tree predictors. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 411–420, 2013.

Arvind Narayanan, Elaine Shi, and Benjamin I. P. Rubinstein. Link prediction by de-anonymization: How we won the Kaggle social network challenge. In *International Joint Conference on Neural Networks (IJCNN)*, 2011.

Julia Neumann, Christoph Schnörr, and Gabriele Steidl. Combined SVM-based feature selection and classification. *Machine Learning*, 61(1-3):129–150, 2005.

Luca Oneto, Sandro Ridella, and Davide Anguita. Tikhonov, Ivanov and Morozov regularization for support vector machine learning. *Machine Learning*, 103(1):103–136, 2015.

Eun Sug Park, Clifford Spiegelman, and Jeongyoun Ahn. A nonparametric approach based on a Markov like property for classification. *Chemometrics and Intelligent Laboratory Systems*, 108 (2):87–92, 2011.

Parikshit Ram, Dongryeol Lee, William March, and Alexander G. Gray. Linear-time algorithms for pairwise statistical problems. In *Neural Information Processing Systems (NIPS)*, pages 1527–1535. 2009.

Vincent Y. F. Tan, Sujay Sanghavi, John W. Fisher, and Alan S. Willsky. Learning graphical models for hypothesis testing and classification. *IEEE Transactions on Signal Processing*, 58 (11):5481–5495, 2010.

Vincent Y. F. Tan, Animashree Anandkumar, and Alan S. Willsky. Learning high-dimensional Markov forest distributions: Analysis of error rates. *Journal of Machine Learning Research*, 12: 1617–1653, 2011.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

Eunho Yang, Pradeep Ravikumar, Genevera I. Allen, and Zhandong Liu. Graphical models via univariate exponential family distributions. *Journal of Machine Learning Research*, 16(115): 3813–3847, 2015.

Ming Yu, Varun Gupta, and Mladen Kolar. Simultaneous inference for pairwise graphical models with generalized score matching. *Journal of Machine Learning Research*, 21(91):1–51, 2020.

Ji Zhu, Saharon Rosset, Robert Tibshirani, and Trevor J Hastie. 1-norm support vector machines. In *Neural Information Processing Systems (NIPS)*, pages 49–56, 2004.