A sensitivity study of bias and variance of k-fold cross-validation in prediction error estimation

Juan D. Rodríguez, Aritz Pérez and Jose A. Lozano

Intelligent Systems Group
Department of Computer Science and Artificial Intelligence
University of the Basque Country
Paseo Manuel de Lardizábal 1, 20080. San Sebastian - Donostia, Spain
juandiego.rodriguez@ehu.es, aritz@ehu.es, ja.lozano@ehu.es
http://www.sc.ehu.es/isg

Abstract

In the machine learning field the performance of a classifier is usually measured in terms of prediction error. In most real-world problems, the error can not be exactly calculated and it must be estimated. Therefore, it is important to choose an appropriate estimator of the error.

This paper analyzes the statistical properties (bias and variance) of the k-fold cross-validation classification error estimator (k-cv). Our main contribution is a novel theoretical decomposition of the variance of the k-cv considering its sources of variance: sensitivity to changes in the training set and sensitivity to changes in the folds. The paper also compares the bias and variance of the estimator for different values of k. The empirical study has been performed in artificial domains because they allow the exact computation of the implied quantities and we can specify rigorously the conditions of experimentation. The empirical study has been performed for two different classifiers (naive Bayes and nearest neighbor), different number of folds (2,5,10,n) and sample sizes, and training sets coming from assorted probability distributions.

1 Introduction

Generally, a classifier is induced from a training data using a classifier learning algorithm. Each classifier has an associated prediction error (true error). But usually the true error is unknown (it can not be calculated) and it must be estimated from data (estimated error). An estimator of the error of a classifier is a random variable $\hat{\varepsilon}$ and its quality is usually measured by means of its bias and variance. There are several estimators of the classification error, from the simple Resubstitution (10) and Hold-out (22) to the more complex Bootstrap (12) and Bolstered (3). One of these techniques, and probably the most popular, is k-fold cross-validation (k-cv) (21). In k-cv the dataset is divided into k folds, a classifier is learnt using $k-1$ folds and an error value is calculated by testing the classifier in the left fold. Finally, the k-cv estimation of the error is the average value of the errors committed in each fold. Thus, the k-cv error estimator depends on two factors: the training set and the partition.

This paper presents a statistical analysis of the k-cv error estimator focusing on its bias and variance. Briefly, we can define the bias as a measure of the goodness-of-fit of the estimator and the variance as the estimator variability. If our objective is the error estimation itself, we should choose the less biased classifier but if our objetive is to compare several classifiers, in addition, we should choose the error estimator with the smallest variance.

There are many publications about k-cv but not many of them have considered the influence in the estimation of the different number of folds. Breiman and Spector (7) carried out a feature subset selection experiment in which they compare k-cv for various k and they recommend 10-cv for model selection although 5-cv also works well for model selection and evaluation. Zhang (29) showed that cross-validation will select too many features for any k value. Kohavi (16) made the most complete study on the matter. He found that there is a trade-off between the bias and the variance of the estimator depending on the number of folds and showed that k-cv with moderate k values reduces the variance while increasing the bias and, alternatively, higher k values increases the variance while decreasing the bias. He also found that repeated cross-validation stabilizes the estimate for small values of k. However, Kohavi used real data sets for the experiments so it was impossible for him to know the real error rates.

We propose a novel theoretical decomposition for the variance of k-cv error estimator. The decomposition divides the variance into an irreducible part (independent from the estimator used) and the reducible part (estimator dependent). Then the reducible part is divided taking into account the two sources of variance: sensitivity to changes in the training set and sensitivity to changes in the folds. We also compare the bias and variance of the k-cv estimator for different values of k using the Friedman plus Nemenyi hypothesis tests (8). The study has been performed on artificial domains because they allow the exact computation of the implied quantities and we can specify rigorously the conditions of experimentation.

The rest of the paper is organized as follows. In Section 2 we briefly explain how to estimate the error using k-cv. Section 3 shows the decomposition of the variance. In Section 4 we explain the experimental process and the working out of the experiment. In Section 5 we present the summary of results emphasizing the bias and variance behavior, especially its decomposition. Finally, our conclusions are presented.

2 Error estimation with k-fold cross-validation

2.1 Notation and definitions

A usual approach to *supervised classification* consists of creating a classifier from training data in order to predict the value (*the label*) of a class attribute $C \in \{1, ..., r_c\}$ given the predictive

attributes (*the feature vector*) $\boldsymbol{X} = (X_1, ..., X_d)$ of an unseen unlabeled instance, $\boldsymbol{x} = (x_1, ..., x_d)$. This work is focused on discrete domains $X_i \in \{1, ..., r_i\}$. We suppose that $(\boldsymbol{X}, C)$ is a random vector with a joint *feature-label probability distribution* $p(\boldsymbol{x}, c)$.

A *classifier* ψ is a function that maps $\boldsymbol{X}$ into C:

$$\begin{array}{rccc} \psi: & \{1, ..., r_1\} \times ... \times \{1, ..., r_d\} & \rightarrow & \{1, ..., r\} \\ & \boldsymbol{x} & \mapsto & c \end{array}$$

A classifier is learned from a training set $S_n = \{(\boldsymbol{x}^{(1)}, c^{(1)}), ..., (\boldsymbol{x}^{(n)}, c^{(n)})\}$ with a classifier induction algorithm $A(\cdot)$, $A(S_n)$. Note that the training set S_n can be considered a random variable which depends on the feature-label random variable $(\boldsymbol{X}, C)$. Given an induction algorithm $A(\cdot)$, which is assumed to be a deterministic function of the training set, the classifier obtained from a training set is denoted as $\psi = A(S_n)$.

In the remainder of this paper we will introduce some definitions for a given induction algorithm $A(\cdot)$, and for the sake of brevity we will omit it from the notation. In the performed experimentation (see Section 4) the induction algorithm used should be clear from the context.

The *prediction error* of a classifier ψ is the probability of wrong classification of unlabeled instances $\boldsymbol{x}$ and is denoted as $\epsilon(\psi)$:

$$\epsilon(\psi) = p(\psi(\boldsymbol{X}) \neq C) = E_{\boldsymbol{x}}[1 - p(\psi(\boldsymbol{x})|\boldsymbol{x})] \tag{1}$$

The prediction error random variable will be denoted as ε and it is distributed according to $p(\varepsilon = a) = \sum_{S_n|\epsilon(A(S_n))=a} p(S_n)$. Sometimes it is called *conditional error* (4) because the classifier is trained in a particular data set S_n with n instances. We can also define the expected error of a classifier trained with sample sets of size n ($E_{S_n}[\epsilon(\psi)]$) as the *Expected Prediction Error* (*unconditional error* (4)).

The minimum theoretical prediction error is given by the *Bayes classifier*, ψ_B, which is defined as:

$$\psi_B(\boldsymbol{x}) = argmax_c\{p(c|\boldsymbol{x})\} = c_B(\boldsymbol{x})$$

We define the *Bayes error* as the prediction error of the Bayes classifier:

$$\epsilon(\psi_B) = E_{\boldsymbol{x}}[1 - p(c_B(\boldsymbol{x})|\boldsymbol{x})] = \sum_{\boldsymbol{x}}(1 - p(c_B(\boldsymbol{x})|\boldsymbol{x})) \cdot p(\boldsymbol{x})$$

Note that this value is the highest lower bound of the error of every classifier. It is important to note that Bayes error does not depend on training data or sample size, since the Bayes classifier depends only on the feature-label probability distribution of the domain.

Nevertheless, in most real world problems, the feature-label probability distribution is unknown. So both the Bayes classifier and the highest lower bound of the prediction error are unknown. Moreover, the prediction error of a classifier ψ is also unknown (can not be computed) and, thus, it must be estimated. Two of the most popular prediction error estimators are the k-fold cross-validation and m times k-fold cross-validation (introduced in the next Section). In order to analyze the estimated error, it is necessary to consider the concepts of bias and variance of the estimator used. Let ε be the real error of the classifier and $\hat{\varepsilon}$ the estimation of the error. The **bias** of an error estimator is defined as the real error value minus the expected estimated error value ($\varepsilon - E[\hat{\varepsilon}]$). An estimator is said to be **unbiased** if it has zero bias. The **variance** of an error estimator is the variance of the error estimate ($E[(\hat{\varepsilon} - E[\hat{\varepsilon}])^2]$). For example, in Figure 1 we have two estimators, $\hat{\varepsilon}_1$ with a high bias and low variance, and $\hat{\varepsilon}_2$ with low bias but a higher variance. Intuitively, the bias measures the average precision of the error estimation while the variance measures the variability of the estimations of the error.

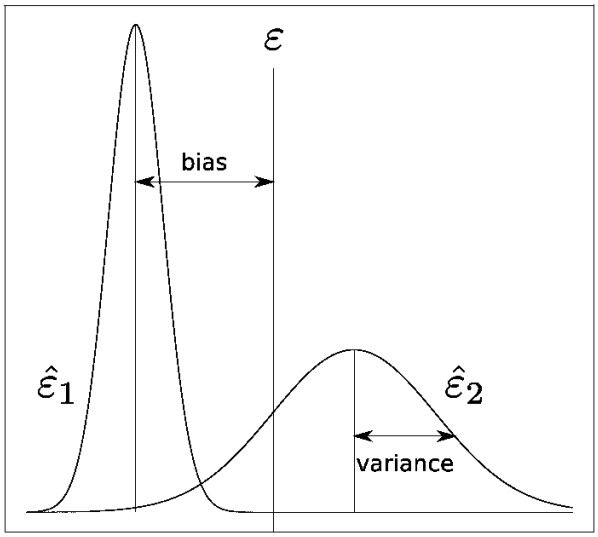


Figure 1: Bias and variance of an estimator

2.2 k-fold cross-validation

In k-cv a data set S_n is partitioned into k folds of similar size $P = \{P_1, ..., P_k\}$ when possible. Let $T_i = S_n \setminus P_i$ be the complement data set of P_i. Then, the algorithm $A(\cdot)$ induces a classifier from T_i, $\psi_i = A(T_i)$, and estimates its prediction error with P_i. The k-cv prediction error estimator of $\psi = A(S_n)$ is defined as follows (21):

$$\hat{\epsilon}_k(S_n, P) = \frac{1}{n} \sum_{i=1}^{k} \sum_{(\boldsymbol{x},c) \in S} 1(c, \psi_i(\boldsymbol{x})) \tag{2}$$

where $1(i, j) = 1$ iff $i = j$ and zero otherwise. So the k-cv error estimator is the average of the errors committed by the classifiers ψ_i in their corresponding partitions P_i. The estimated error can be considered a random variable which depends on the training set S_n and the partition P. The k-cv proccess is graphically represented in Figure 2.

Generally, an estimator is a *randomized error estimator* if there are internal factors that affect its outcome. On the other hand, if the error estimator is a deterministic function, it is a *non-randomized error estimator* and its variance due to internal factors is zero. For example k-cv with $k < n$ is a randomized error estimator because it depends on the partition P used, and k-cv with $k = n$ is deterministic because there is only one possible partition of the data.

A k-cv error estimator is an unbiased estimator of the error ϵ on data sets of $n - n/k$ size (2), but it is biased for ϵ on data sets of size n because only a subset of the instances with size $n - n/k$ is used for training. This is called *the surrogate problem* (5). Intuitively, this characteristic will cause k-cv to be a pessimistic estimator. On the other hand,as regards the variance, it is known that there is no unbiased estimator of the variance $Var[\hat{\epsilon}_k(S_n, P)]$ of k-cv (1).

The *repeated m times k-cv* (m-k-cv) consists of estimating the error as the average of m k-cv estimations with different random partitions $\boldsymbol{P} = \{P^{(1)}, \ldots, P^{(m)}\}$:

$$\hat{\epsilon}_{k,m}(S_n, \boldsymbol{P}) = \frac{1}{m} \sum_{i=1}^{m} \hat{\epsilon}_k(S_n, P^{(i)})$$

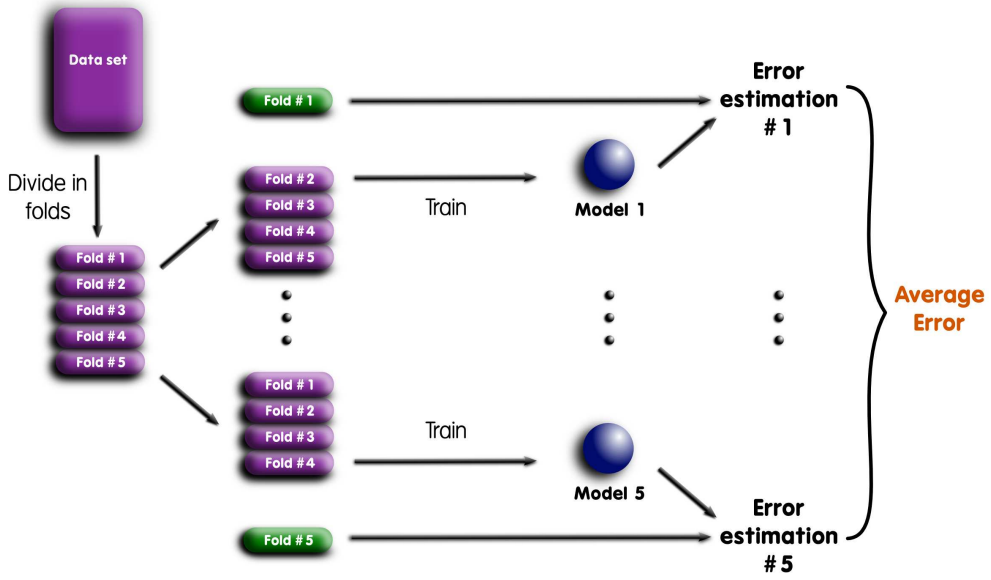


Figure 2: k-fold cross validation error estimator

It is supposed that the repeated version stabilizes the error estimation and so it reduces the variance of the k-cv estimator, especially for small samples (16).

As can be deduced from the previous definitions, when a classifier induction algorithm $A(\cdot)$ is fixed, the k-cv and m-k-cv estimators have two sources of variance (when $k < n$). One comes from the training sets S_n used for the training-test process and the other comes from the partition P (or partitions $\boldsymbol{P}$) of S_n because it affects the internal training-test partitions. So the k-cv and m-k-cv estimators are sensitive to changes in both the training set and the partitions. But, what part of the total variance depends on the estimator used and what part is independent? How are the different sources of variance defined and how are they related with the total variance? What is their relative importance for determining the total variance? So as to answer these interesting questions, the next section provides a novel decomposition of the variance.

3 Decomposition of the Variance of the k-cv estimator

In order to analyze the behavior of the variance of the cross-validation, we use the following random variables. All of these variables are defined given a classifier induction algorithm $A(\cdot)$ wich is omitted from the expressions. The true prediction error random variable, ε, measures the prediction error of a classifier induced with $A(\cdot)$, and it follows the distribution $p(\varepsilon = a) = \sum_{S_n|\epsilon(A(S_n))=a} p(S_n)$ (see Eq. 1). The estimated error random variable $\hat{\varepsilon}_k$, measures the estimated prediction error of a classifier induced with $A(\cdot)$ by means of the k-cv procedure and it follows the distribution $p(\hat{\varepsilon}_k = a) = \sum_{S_n,P|\hat{\epsilon}_k(S_n,P)=a} p(S_n, P)$ (see Eq. 2). Note that $p(S_n, P) = p(S_n)p(P)$ due to the independence of S_n and P. The deviation of the error random variable, δ_k, measures the deviation $\Delta_k(S_n, P) = \epsilon(S_n) - \hat{\epsilon}_k(S_n, P)$ and follows the distribution $p(\delta_k = a) = \sum_{S_n,P|\Delta_k(S_n,P)=a} p(S_n, P)$.

The estimated error $\hat{\varepsilon_k}$ can be written as $\hat{\varepsilon}_k = \varepsilon - \delta_k$. Thus, its variance can be decomposed into three terms:

$$Var[\hat{\varepsilon}_k] = Var[\varepsilon] + Var[\delta_k] - 2Cov[\varepsilon, \delta_k] \tag{3}$$

If we assume the independence between ε and δ_k, the variance of $\hat{\varepsilon}_k$ can be decomposed into two terms because $Cov[\varepsilon, \delta_k] = 0$ (this is not a strong assumption as in the domains used for the experimentation the covariance is a small fraction of the total variance, less than 5% in most cases):

$$Var[\hat{\varepsilon}_k] \simeq Var[\varepsilon] + Var[\delta_k] \tag{4}$$

Now we can study the variance of the estimation as the variance of the real error (with respect to S_n) plus the variance of the deviation of the error. The variance of the real error ε depends only on the training set used and it is independent from the estimator. We call it *irreducible variance* because it is common to all the estimators. So, in order to study the properties of the k-cv and m-k-cv estimators it is desirable to subtract it from the total variance, $Var[\hat{\varepsilon}_k] - Var[\varepsilon] = Var[\delta_k]$. The variance of δ_k is the variance of the precision of the estimation. It is the part of the total variance associated with the estimator used and we call it *reducible variance*. It depends on both the training set S and the partition P used.

The variance of δ_k can be decomposed exactly into two terms (see Appendix) taking into account its sources of the variance:

$$Var[\delta_k] = TS + PS \tag{5}$$

where TS and PS are the variances due to changes in the training set and to changes in the partitions respectively. Note that the variance can be understood as a sensitivity measure. The definition of both terms is as follows:

$$TS = 1/2(Var_S[E_P[\delta_k]] + E_P[Var_S[\delta_k]]) \tag{6}$$
$$PS = 1/2(Var_P[E_S[\delta_k]] + E_S[Var_P[\delta_k]]) \tag{7}$$

All the terms in P (E_P and Var_P) can be interpreted as follows: without loss of generality, we

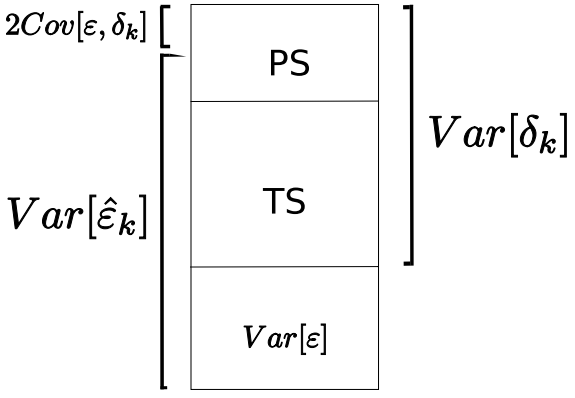


Figure 3: Variance decomposition

can assume a complete order in the probability space $(\mathbf{X}, C)$, so given a training set S_n, it can be

considered ordered. Now we can understand a partition P as a permutation of $\frac{n}{k}$ 1's, $\frac{n}{k}$ 2's,...,$\frac{n}{k}$ k's. Therefore, given a partition P, it is possible to consider expectation or variance values over this partition as it univocally describes a partition for each S_n.

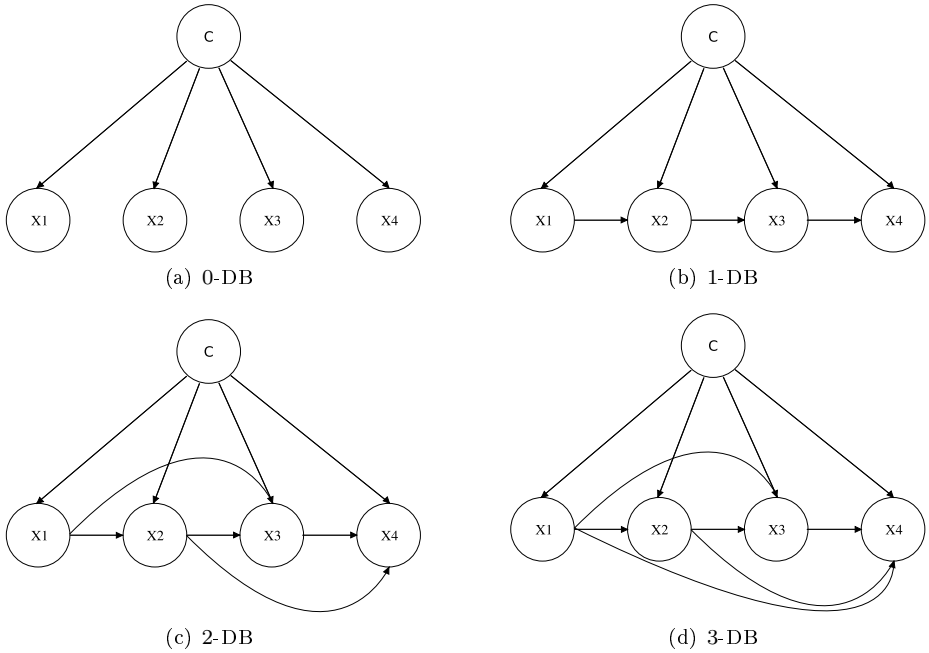
A representation of the overall decomposition can be seen in Figure 3.

4 Experimental study

In this section we empirically study the statistical properties of the k-cv estimator (bias and variance) and we analyze the variance using the decomposition proposed in the previous section. First we present the artificial domains (Subsection 4.1) and the classifiers (Subsection 4.2) that we have used and subsequently, the empirical process (Subsection 4.3) and the obtained results (Subsection 4.4).

4.1 Artificial domains

We use artificial data sets because it allows us to calculate the real error rate instead of using the empirical one. For this purpose, we sample the data sets from artificial *feature-label probability distributions* represented as Bayesian networks (24). We use *K-dependence Bayesian classifier* (K-DB) (25) structures that allow each predictive variable X_i to have a maximum of K dependencies with other predictive variables, with the exception of dependences with variable C. A particular kind of K-DB is found when $K = 0$ (naive Bayes (19; 23)) and $K = 1$ (forest augmented naive Bayes, *FAN* (20)). We use K-DB structures because it allows us to specify the number of dependencies between the attributes and so we can control the complexity of the probability distribution. The chosen K values are $\{0, 1, 2, 3\}$. (see figure 4)



(a) 0-DB (b) 1-DB

(c) 2-DB (d) 3-DB

Figure 4: K-DB

4.2 Naive Bayes and K-NN classifiers

The experimentation includes the study of the k-cv estimator for two different classifiers: naive Bayes (NB) (19; 23) and nearest neighbor (NN) (9). We have decided to use these classifiers due to their opposite nature from the point of view of the number of parameters required for each model. The bias and variance of the k-cv estimator should change depending on the classifier used due to its particular sensitivity to changes in the training set. The sensitivity of a classifier is usually related with the number of parameters that it needs: the sensitivity increases as the number of parameters increases. After introducing both paradigms, we briefly analyze the number of parameters required by them in order to establish their relative sensitivities.

The NB classifier can be considered as a Bayesian network with a special graph topology. It assumes that the predictor variables are conditionally independent given the class, being the class the only parent of each predictor variable. In order to obtain the *a posteriory* probability distribution of the class given the predictors, $p(c|\boldsymbol{x})$, it uses the Bayes rule:

$$p(c|\boldsymbol{x}) = \frac{p(c,\boldsymbol{x})}{p(\boldsymbol{x})} \propto p(c,\boldsymbol{x})$$

The factorization of the joint probability is very simple because of its independence assumption:

$$p(c|\boldsymbol{x}) = p(c)\prod_{i=1}^{n} p(x_i|c)$$

Generally, NB classifies a new case $\boldsymbol{x}$ using the *a posteriori* distribution together with the *winner-takes-all* rule:

$$c^* = argmax_c\{p(c|\boldsymbol{x})\} = argmax_c\{p(c,\boldsymbol{x})\}$$

The NB classifier requires $r - 1 + \sum_{i=1}^{d}(r_i - 1)\cdot r$ parameters, where r is the cardinality of the class variable, r_i is the cardinality of the predictive variable X_i and d is the number of predictive attributes. The low number of parameters needed by NB is due to the strong conditional independence of each pair of predictive variables given the class variable. It should be noted that the number of parameters needed is independent of the number of instances n in the training set.

The NN classifier is based on a distance measure. In order to classify a new instance, it computes the distances to every case in the training set and, then, it selects the class which belongs to the nearest case. The NN classifier requires $n \cdot d$ parameters so, considering that in our experiments $n >> d$, the number of parameters of NN is higher than the number of parameters of NB. NN is known as a lazy classifier because it does not construct a model of the data from the training set and it needs to store all the available data (if a case condenses or selection technique is not performed).

It is generally accepted that the error estimation of a classifier has higher variance and lower bias as the number of required parameters increases, or equivalently, as the sensitivity to the changes in the training sets increases (11).

4.3 The empirical process

We consider domains with 10 predictive attributes and 1 class attribute. The predictive attributes are binary and the class attribute cardinality ranges from 2 to 5. In order to obtain assorted distributions with different dependencies and complexity degrees the following procedure (shown in Figure 5) has been carried out. For each K and class cardinality we generate 10 random distributions encoded with the previously described Bayesian networks. Then, for each generated Bayesian classifier, we sample 10 data sets of each sample size. The selected sample sizes are 1%, 5%, 10% and 25% of the total size of the probability space.

The data sets generated are 6400 (4 different K values, 4 different class cardinalities, 10 distributions for each class cardinality and K value, 4 different sample sizes and 10 sets sampled

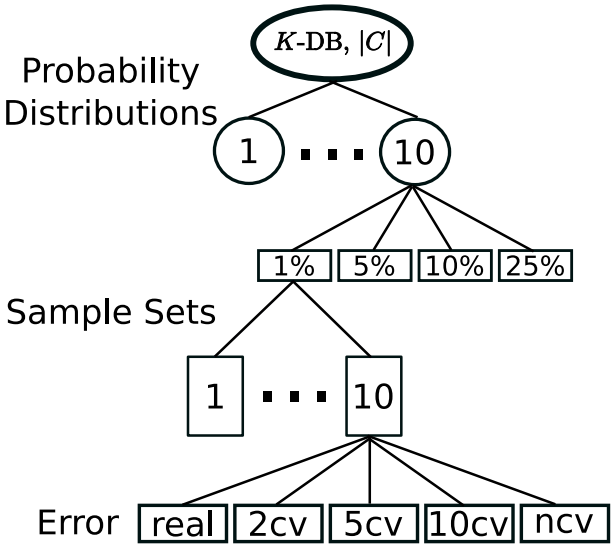


Figure 5: Experimental design.

from each distribution and sample size). For each dataset and each classifier (NB and NN) we estimate 10 times the $PE_k(S_n, P)$ for 10 different random data partitions P, and 10 times the $PE_{(m=10),k}(S_n, \boldsymbol{P})$ for 10 different sets of partitions $\boldsymbol{P}$. The considered k values for the cross-validation are $k = \{2, 5, 10, n\}$. We use the k-cv error estimator provided by the *WEKA* library (28). The random generated Bayesian networks have been obtained using the *BNGenerator* software (14). This software first generates a directed acyclic graph with N nodes uniformly distributed in the space of graphs under cosideration and then, for the generated graph, the conditional probability distributions. The structure is constructed by means of a Markov Chain Monte Carlo (MCMC) method and the probabilities are sampled from a *Dirichlet distribution* (13).

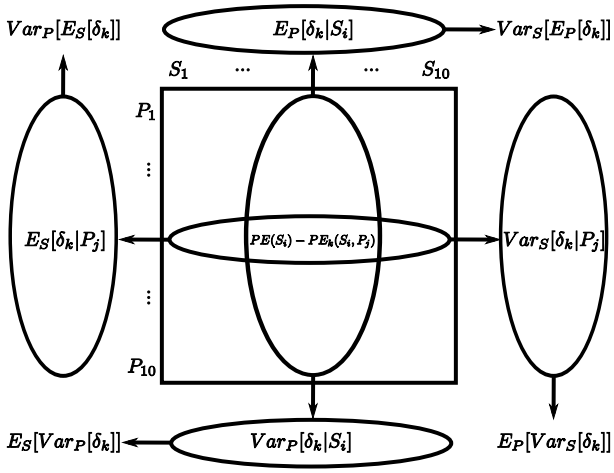


Figure 6: Computation of the implied quantities.

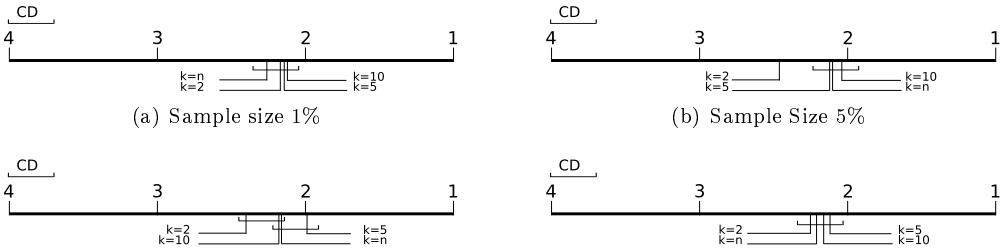
Another point of interest is how we compute TS and PS. In Figure 6 we show the computation

of these quantities.

4.4 Experimental results

This section has been divided into three paragraphs. First, in order to measure the influence of the different sources of variance of the k-cv error estimator, we empirically analyze the decomposition of the variance given in Eq. 3. Then, we study the behavior of the bias and the variance of k-cv for different k values and sample sizes using the Friedman plus Nemenyi statistical test (8). Finally, we make a brief comparison of the NB and NN classifiers using the Wilcoxon test (8).

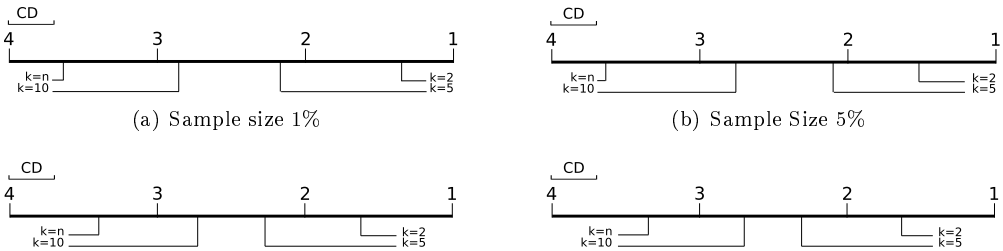
Decomposition of the variance We begin the variance analysis starting out from the decomposition of the variance of the deviation of the error δ_k (Eq. 5). In Figures 9, 10, 11 and 12 we present the results of the proposed decomposition (see Figure 3). Each bar of the figures represents the total variance of the estimator, the lowest part of the bar (the darkest one) is the variance of the true error ϵ (irreducible variance), the rest of the bar (reducible variance) is the variance of the deviation of the error, δ_k (Eq. 5), and is divided into two terms, the variance due to changes in the partitions, partition sensitivity PS (Eq. 7), and the variance due to changes in the training set, training sensitivity TS (Eq. 6). Note that PS is zero for $k = n$.



(a) Sample size 1% (b) Sample Size 5%

(c) Sample Size 10% (d) Sample Size 25%

Figure 7: Nemenyi tests of Variance on k-cv

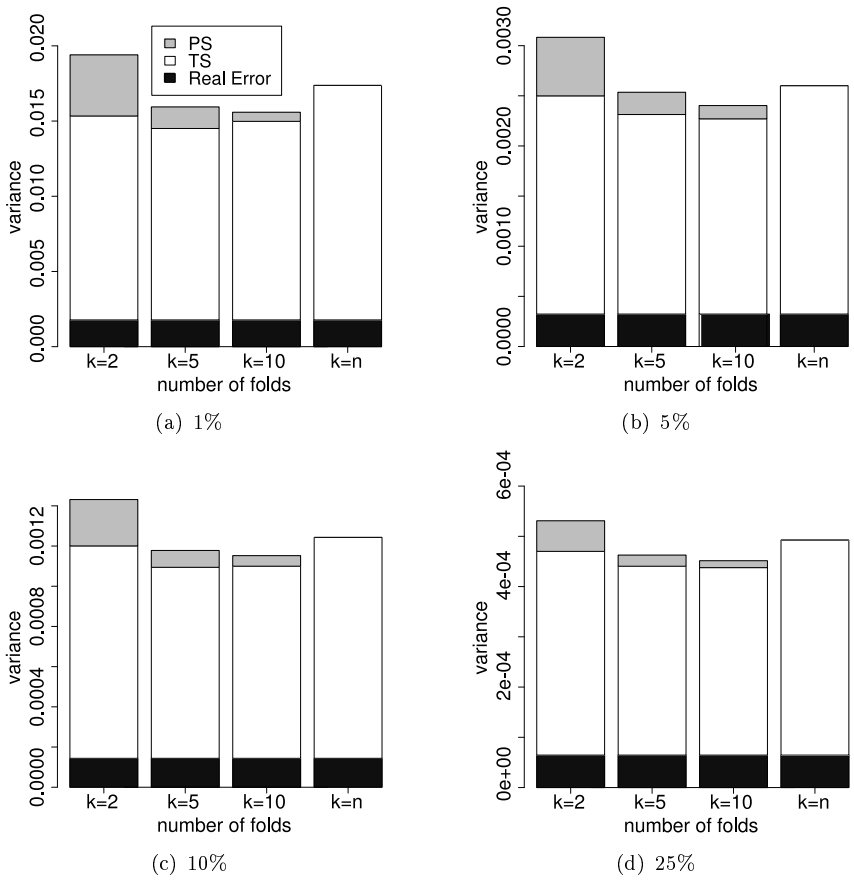


(a) Sample size 1% (b) Sample Size 5%

(c) Sample Size 10% (d) Sample Size 25%

Figure 8: Nemenyi tests of Variance on repeated k-cv

The training sensitivity TS dominates the total variance because it is clearly bigger than the partition sensitivity PS. In no-repeated k-cv the training sensitivity TS is 2-4 times bigger with $k = 2$, 4-9 times bigger with $k = 5$ and 5-12 times bigger with $k = 10$. In repeated k-cv the differences are even greater, the training sensitivity TS is 11-33 times bigger with $k = 2$, 21-80 times bigger with $k = 5$ and 28-143 times bigger with $k = 10$.



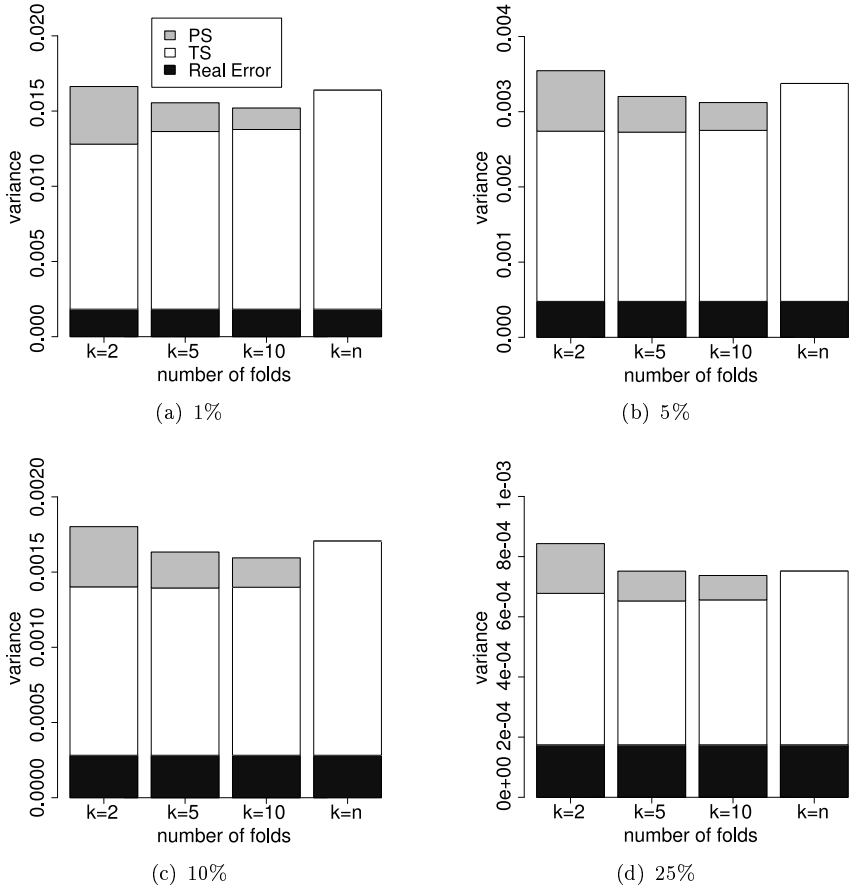
(a) 1% (b) 5%

(c) 10% (d) 25%

Figure 9: Variance Decomposition on k-cv with NB.

In the analysis of the decomposition for different values of k, the partition sensitivity PS decreases with higher values of k. Moreover, in the case of no-repeated, the differences between the values of the total variance ($k = 2, 5, 10$) are mainly due to PS (as the values of TS are very similar). Training sensitivity TS does not have a clear behavior in no-repeated k-cv, but in repeated k-cv it increases with higher k values. Finally, it is important to note that the ratio between PS and TS seems to be independent of the size of the training set (this relation seems to be kept for each k).

Comparison of bias and variance for different k values In addition to the previous analysis, we have also compared the bias and the total variance of the estimators for the different values of k. In order to do that, we have carried out statistical tests, a paired Friedman test plus the Nemenyi post-hoc test when the null hypothesis is rejected (8) based on 320 paired estimated errors (2 classifiers, 4 K values, 4 class cardinalities, 10 distributions for each cardinality and K value). The significance of this test is $0,01$ (see Figures 7, 8, 17 and 18). Each point of this figures represents the bias or variance of the classifier as the average over the 400 data sets of each sample size. There are more specific figures considering different K-DBs and different class cardinality in the appendix.



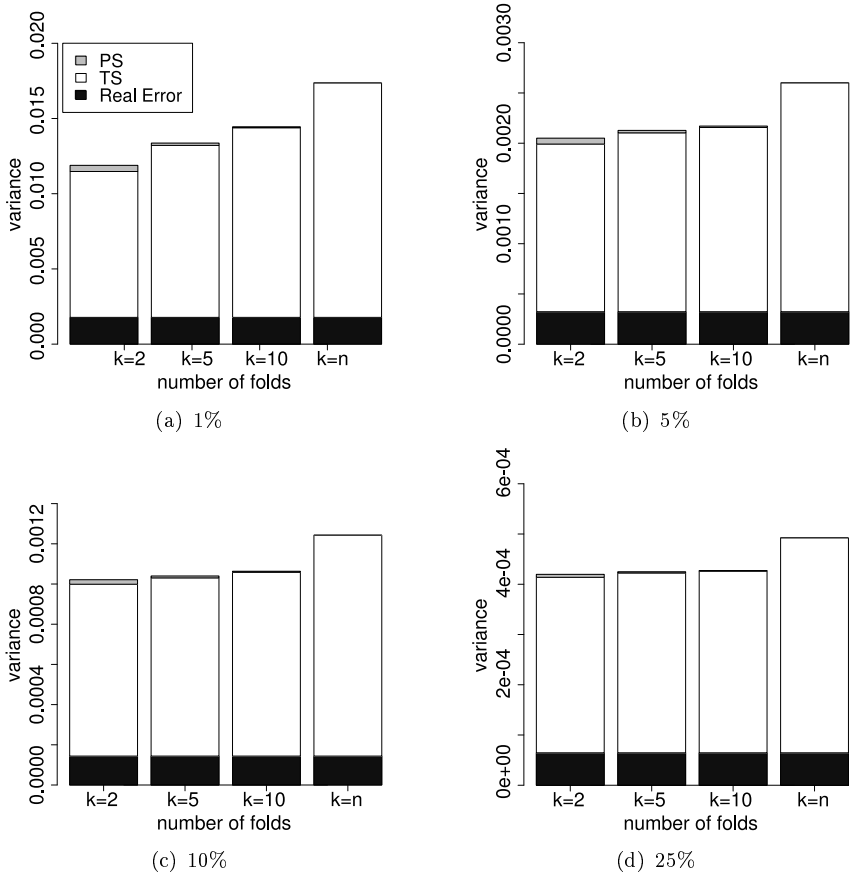
(a) 1%

(b) 5%

(c) 10%

(d) 25%

Figure 10: Variance Decomposition on k-cv with NN.



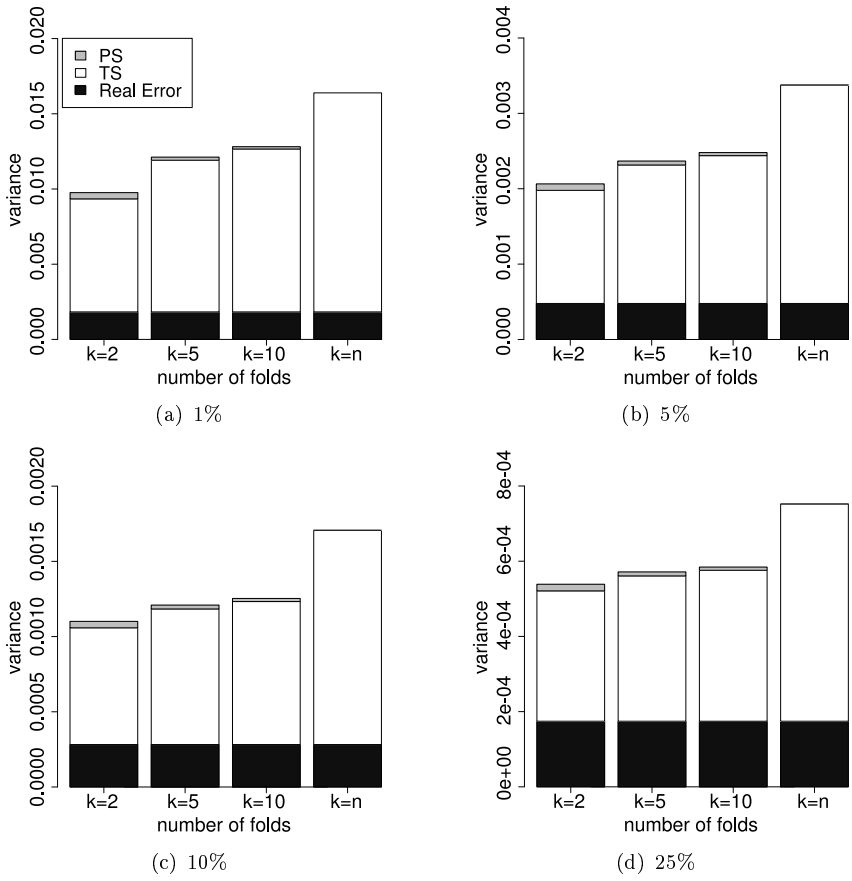
(a) 1%

(b) 5%

(c) 10%

(d) 25%

Figure 11: Variance Decomposition on repeated k-cv with NB.



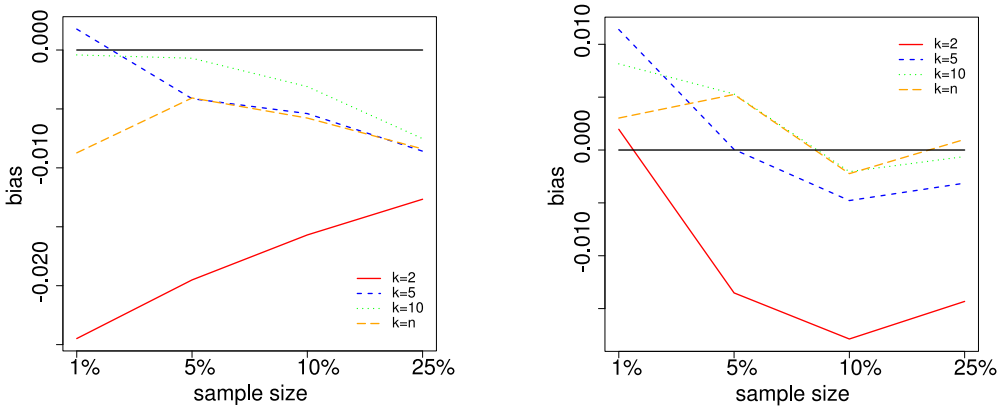
(a) 1%

(b) 5%

(c) 10%

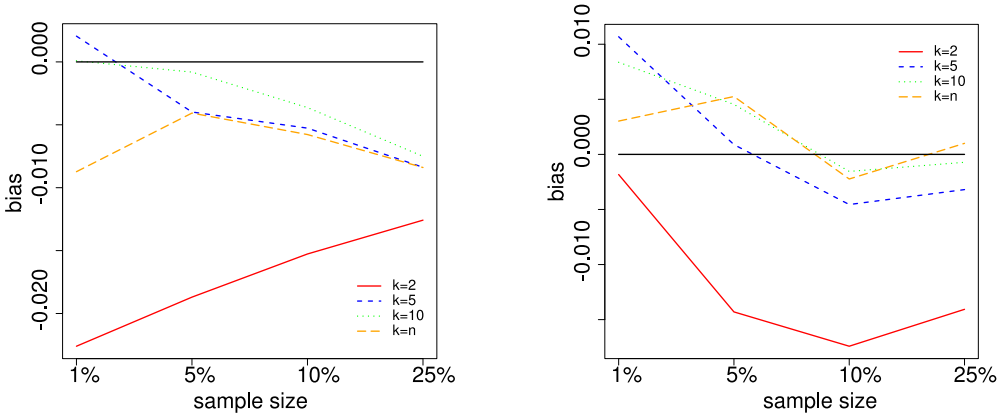
(d) 25%

Figure 12: Variance Decomposition on repeated k-cv with NN.



(a) naive Bayes (b) nearest neighbor

Figure 13: Bias on k-cv



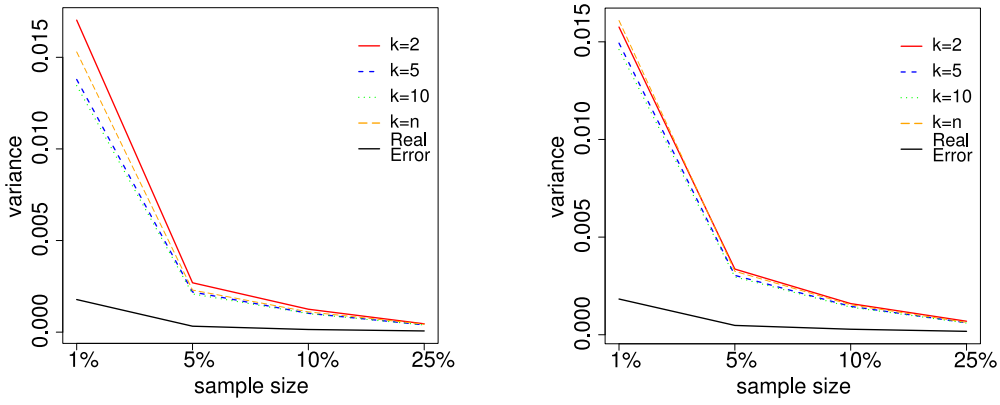
(a) naive Bayes (b) nearest neighbor

Figure 14: Bias on repeated k-cv

The first evidence is that in all cases the variance of the estimator decreases with the sample size (4) (see Figures 15 and 16). Besides, the variance of the estimator is lower on repeated k-cv than in no-repeated k-cv.

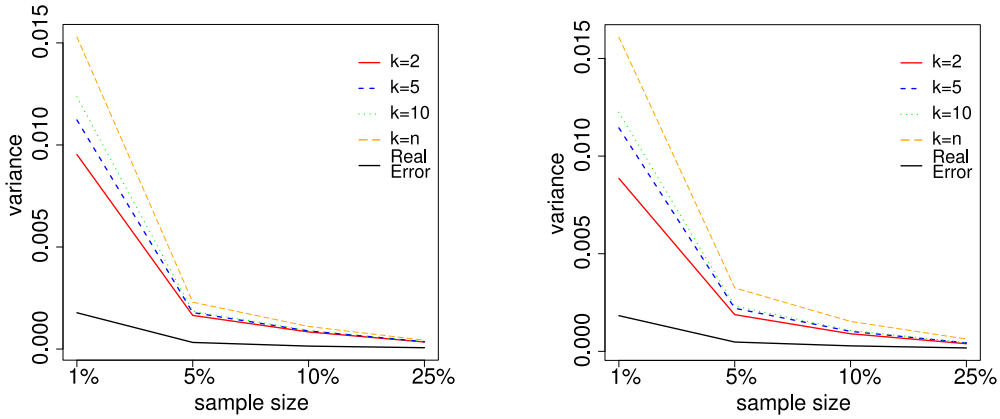
But there are differences among repeated and no-repeated k-cv if we focus on the variance for different k-values. In no-repeated there are no significative differences between different number of folds because the total variance for different k values is very similar (see Figure 7). Repeated k-cv stabilizes the variance in such a way that significative differences appear (see Figure 8) and a k ranking from lowest to highest variance arises: $k = 2, 5, 10, n$.

On the other hand, if we focus on the bias we realize that $k = 2$ has the larger bias for both classifiers (NB and NN) because we use only $n/2$ samples for learning. There is no significative difference between the bias for the remaining studied k values but they all are significatively less biased than $k = 2$ except for sample size of 1% (See figure 7). The bias is nearly zero for all sample sizes, specially for sample size $> 5\%$. $k = 2$ is significatively the most biased k value except



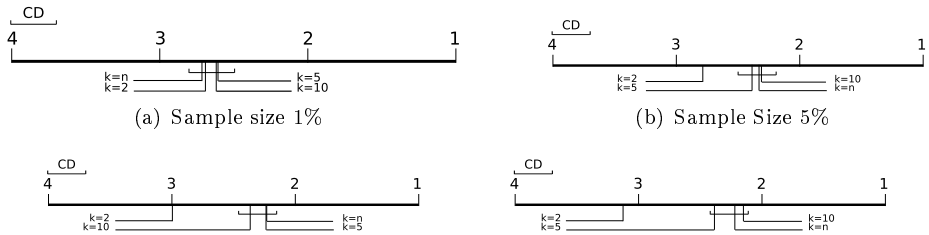
(a) naive Bayes (b) nearest neighbor

Figure 15: Variance on k-cv



(a) naive Bayes (b) nearest neighbor

Figure 16: Variance on repeated k-cv



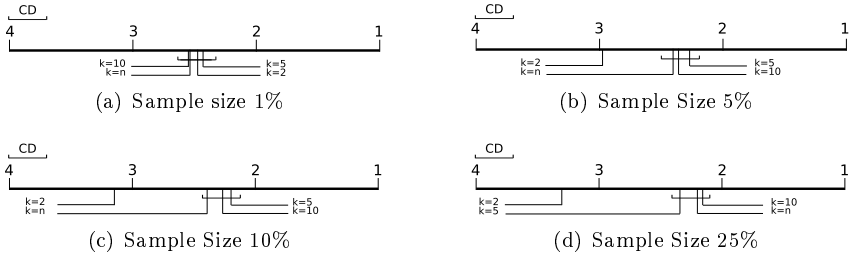
(a) Sample size 1%

(b) Sample Size 5%

(c) Sample Size 10%

(d) Sample Size 25%

Figure 17: Nemenyi tests of Bias on k-cv



(a) Sample size 1%

(b) Sample Size 5%

(c) Sample Size 10%

(d) Sample Size 25%

Figure 18: Nemenyi tests of Bias on repeated k-cv

for no-repeated k-cv on small samples. The remaining k values show no significative differences among them (see Figures 17 and 18). 2-cv has the largest bias for both classifiers (NB and NN) because we use only $n/2$ samples for learning. Anyway, the bias is nearly zero for all sample sizes, specially for sample sizes higher than 5%.

		1%	5%	10%	25%
$k=2$	Bias	$\circ 0,00246$	$\circ 0,00226$	$\star 0,00007$	$\star 0,00006$
	Variance	$\circ 0,00195$	$\star 0,00046$	$\star 0,00093$	$\star 0,00757$
$k=5$	Bias	$\circ 0,00025$	$\circ 0,00317$	$\circ 0,00282$	$\circ 0,00426$
	Variance	$\star 0,00128$	$\star 0,00266$	$\star 0,00813$	$\star 0,00814$
$k=10$	Bias	$\circ 0,00061$	$\circ 0,00326$	$\circ 0,00321$	$\circ 0,00487$
	Variance	$\circ 0,00052$	$\star 0,00292$	$\star 0,00049$	$\star 0,00795$
$k=n$	Bias	$\circ 0,00106$	$\circ 0,00329$	$\circ 0,00333$	$\circ 0,00504$
	Variance	$\circ 0,00521$	$\star 0,00120$	$\star 0,00048$	$\star 0,00049$

$\star > 0,01\alpha \rightarrow NB < NN$

$\circ > 0,01\alpha \rightarrow NN < NB$

Table 1: Wilcoxon test at $\alpha < 0,01$ between NB and NN classifiers and the difference

Comparison of NB and NN Finally we have also compared the classifiers. The comparison among classifiers (NB and NN) has been performed using the paired Wilcoxon signed-rank (8) test and we have obtained statistically significative results at $\alpha < 0,01$. Table 1 shows the p-values of the statistical tests and the differences between both classifiers. The variance of NB is lower than in NN (especially in no-repeated k-cv) and NN is less biased than NB due to the differences in the number of parameters (11).

5 Conclusions

This paper proposes a novel decomposition of the variance of the k-fold cross-validation for prediction error estimation. The variance is decomposed into two independent terms (see Eq. 4), the irreducible variance $Var(\varepsilon)$ and the reducible variance $Var(\delta_k)$. The irreducible variance is independent from the value of k and the partitions P used and only depends on the training set. Then, the reducible variance is decomposed into two terms (see Eq. 5) taking into account its sources: the variance due to changes in the training set TS (training sensitivity, see Eq. 6) and due to changes in the partition PS (partition sensitivity, see Equation 7).

Furthermore, the paper empirically studies the statistical properties (bias and variance) of the k-fold cross-validation for error estimation (and its repeated version). The empirical study is divided into three parts: (i) decomposition of the variance, (ii) comparison of bias and variance of the estimator for different k values and training set sizes n and, (iii) comparison of bias and variance of the estimator for different induction algorithms, naive Bayes and nearest neighbor.

In the first study (i) we can conclude that training sensitivity TS is much bigger than partition sensitivity PS. PS decreases with higher values of k. TS does not have a clear behavior in no-repeated k-cv, but in repeated k-cv TS increases with higher k values. The ratio between PS and TS seems to be kept for different k values. We have observed that the repeated version reduces PS to a small fraction of the total variance. In the second study (ii), we have not found significative differences between the variance of no-repeated k-cv when the number of folds changes. On the other hand, for 10 times repeated k-cv estimator, a ranking on the variance appears with significative differences between all k values (from lowest to highest variance: $k = 2, 5, 10, n$). Focusing on the bias, it seems that for k-cv (and its repeated version), $k = 2$ is the most biased estimator. Anyway, the bias is close to zero for different training set sizes, especially for sample sizes higher than 5%. In the third study (iii), we realize that NN is less biased than NB but with more variance due to the differences in the number of parameters.

6 Acknowledgment

This work has been partially supported by the Saiotek and Research Groups 2007-2012 (IT-242-07) programs (Basque Government), TIN2008-06815-C02-01 and Consolider Ingenio 2010 - CSD2007-00018 projects (Spanish Ministry of Science and Innovation) and COMBIOMED network in computational biomedicine (Carlos III Health Institute).

A Exact decomposition of the variance

In this Section we demonstrate the exact decomposition of the variance of a random variable, Z, which depends on two random independent variables, X and Y.

Theorem Given two independent random variables, X and Y, and a third random variable, Z, which depends on X and Y, we have that:

$$\begin{aligned} Var_{X,Y}[Z] &= 1/2(E_X[Var_Y[Z]] + Var_Y[E_X[Z]]) \\ &+ 1/2(E_Y[Var_X[Z]] + Var_X[E_Y[Z]]) \end{aligned} \tag{8}$$

proof: By definition of the variance of Z we have that

$$Var_{X,Y}[Z] = E_{X,Y}[Z^2] - E_{X,Y}[Z]^2 \tag{9}$$

We can rewrite this definition by adding and subtracting the term $E_X[E_Y[Z]^2]$ as follows

$$\begin{aligned} Var_{X,Y}[Z] &= E_{X,Y}[Z^2] - E_X[E_Y[Z]^2] \\ &\quad + E_X[E_Y[Z]^2]) - E_{X,Y}[Z]^2 \\ &= E_X[E_Y[Z^2] - E_Y[Z]^2] \\ &\quad + E_X[E_Y[Z]^2]) - E_X[E_Y[Z]]^2 \\ &= E_X[Var_Y[Z]] + Var_X[E_Y[Z]] \end{aligned} \tag{10}$$

Following the same procedure with the term $E_Y[E_X[Z]^2]$ we obtain the following equality:

$$Var_{X,Y}[Z] = E_Y[Var_X[Z]] + Var_Y[E_X[Z]] \tag{11}$$

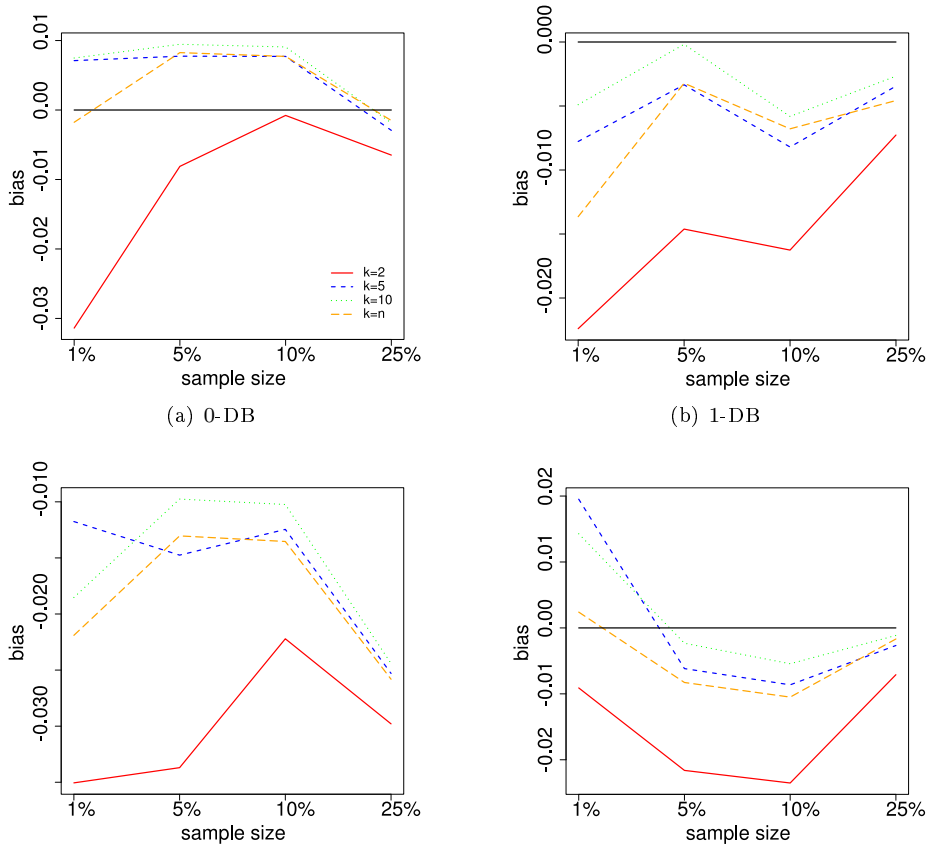
Using Eq. 10 and Eq. 11 and regrouping the terms we prove the theorem

$$\begin{aligned} Var_{X,Y}[Z] &= 1/2(Var_{X,Y}[Z] + Var_{X,Y}[Z]) \\ &= 1/2(E_Y[Var_X[Z]] + Var_Y[E_X[Z]] \\ &\quad + E_X[Var_Y[Z]] + Var_X[E_Y[Z]]) \\ &= 1/2(E_Y[Var_X[Z]] + Var_X[E_Y[Z]]) \\ &\quad + 1/2(E_X[Var_Y[Z]] + Var_Y[E_X[Z]]) \end{aligned}$$

We have decomposed the variance of Z into two additive terms which represent the sources of variance due to variables X and Y respectively (see the two terms of Eq. 9). We call $1/2(E_Y[Var_X[Z]]+Var_X[E_Y[Z]])$ the sensitivity of Z with respect to X , and $1/2(E_X[Var_Y[Z]]+Var_Y[E_X[Z]])$ the sensitivity of Z with respect to Y. This property of the variance allows us to decompose the variance of the estimated prediction error random variable, $\hat{\epsilon}_k$, into the sensitivity to changes in the permutation and the sensitivity to changes in the training set.

B Extra figures

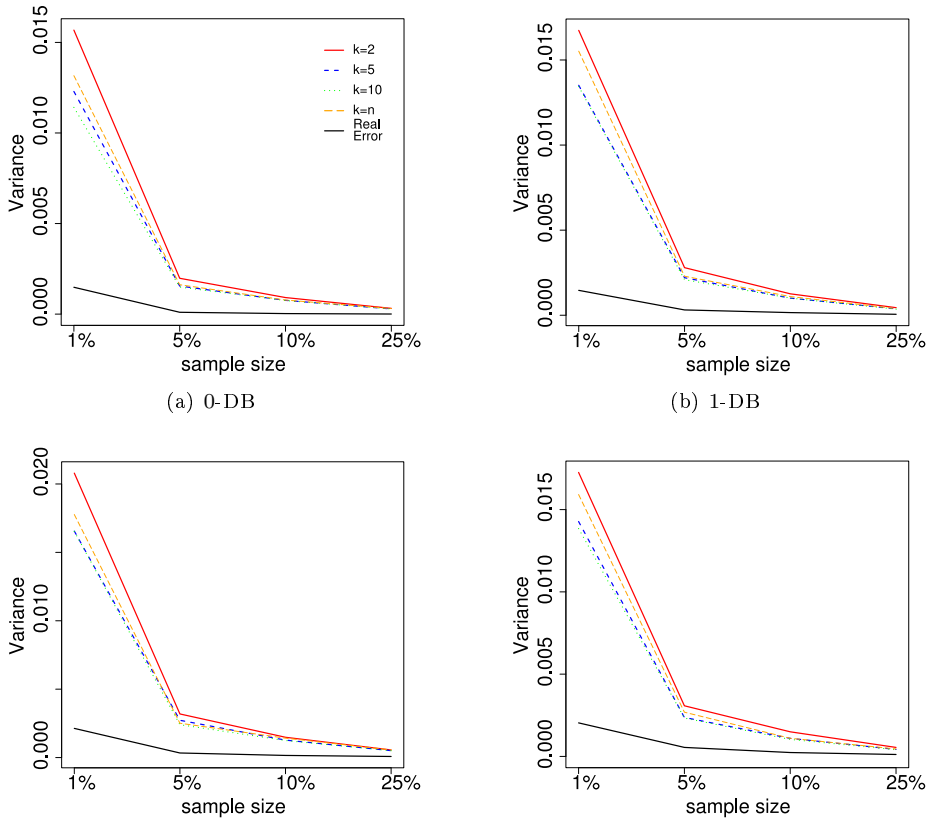
B.1 k-cv on naive Bayes classifier



(a) 0-DB (b) 1-DB

(c) 2-DB (d) 3-DB

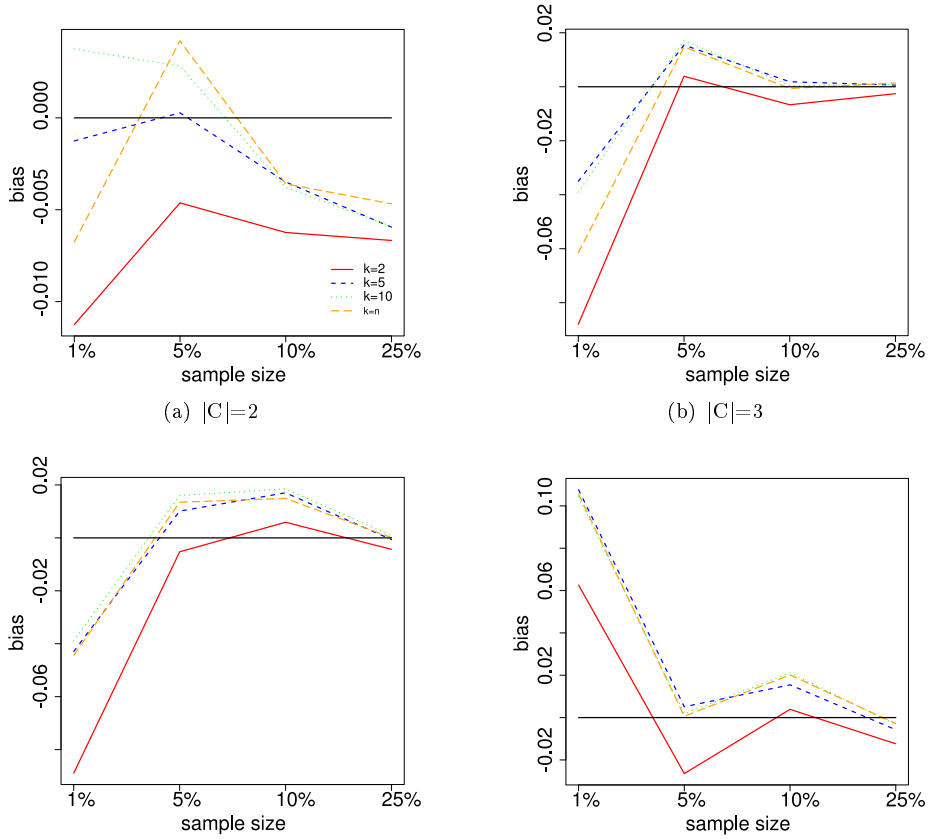
Figure 19: Prediction Error bias



(a) 0-DB (b) 1-DB

(c) 2-DB (d) 3-DB

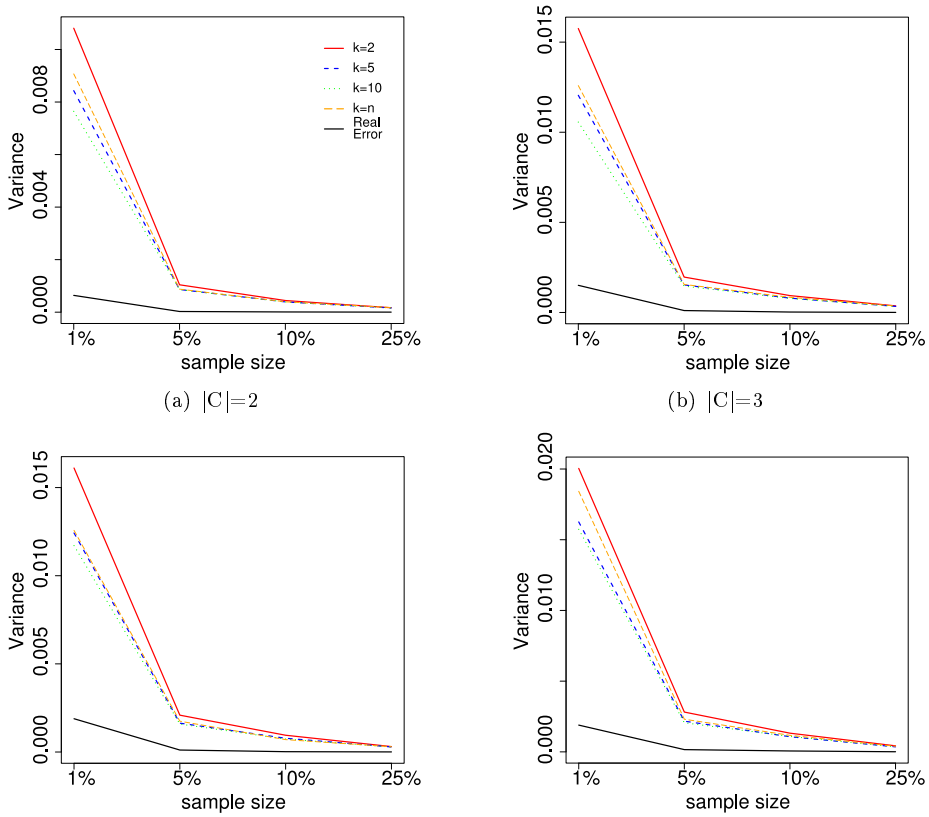
Figure 20: Prediction Error variance



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 21: Prediction Error bias

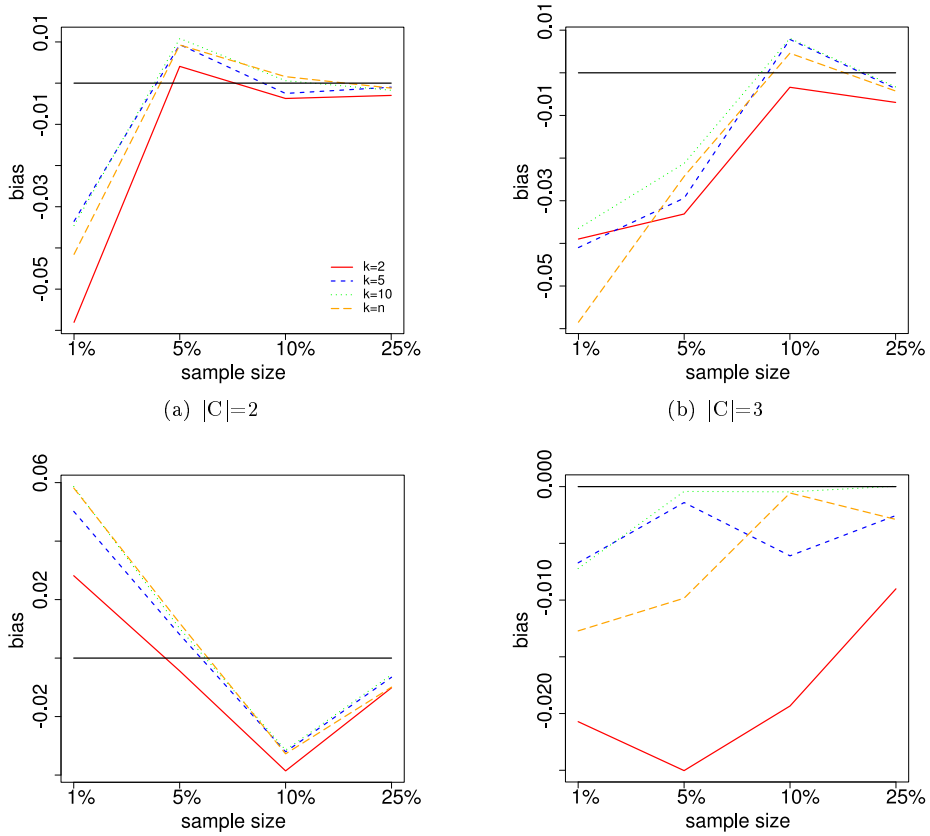


(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 22: Prediction Error variance

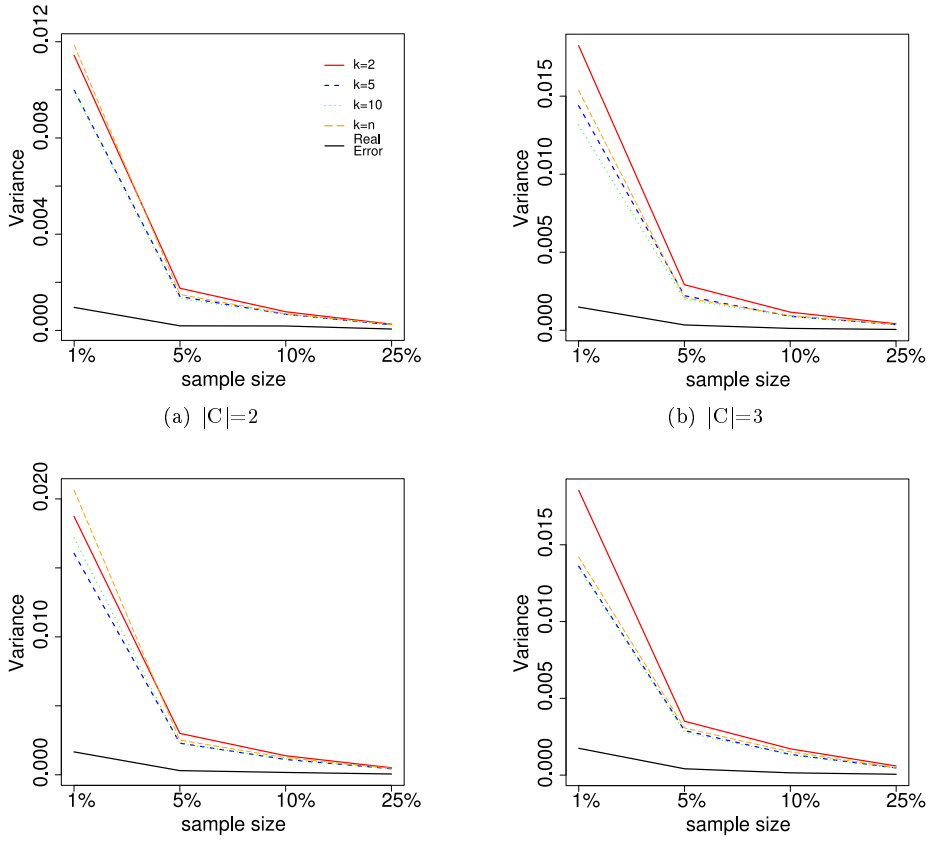
1-**DB Structure**



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 23: Prediction Error bias

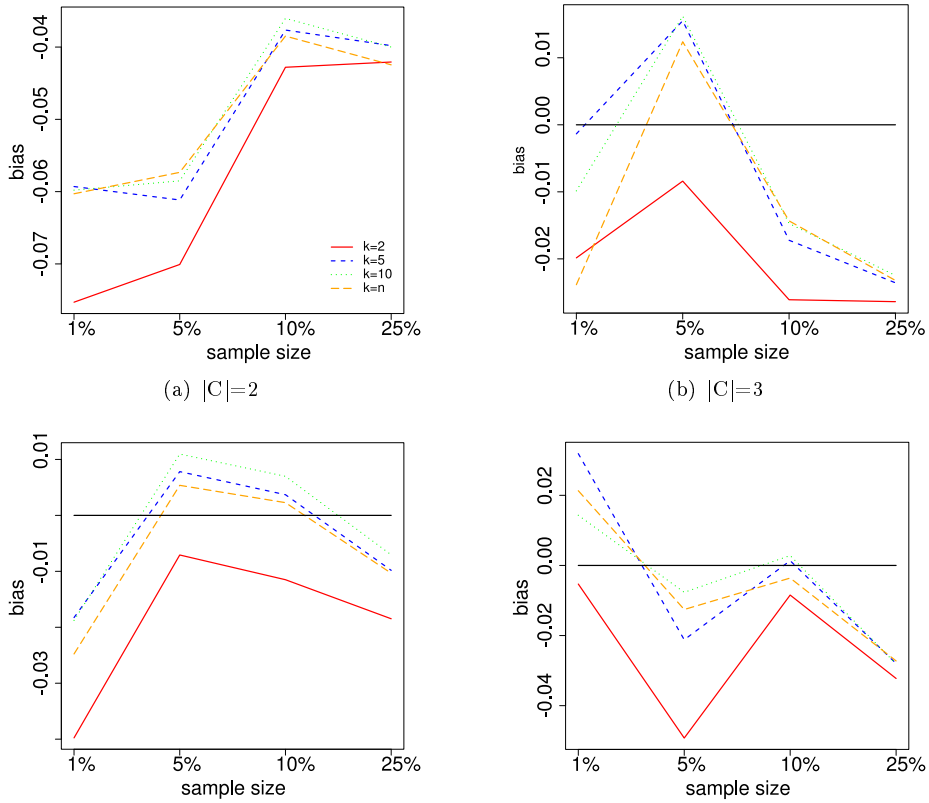


(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 24: Prediction Error variance

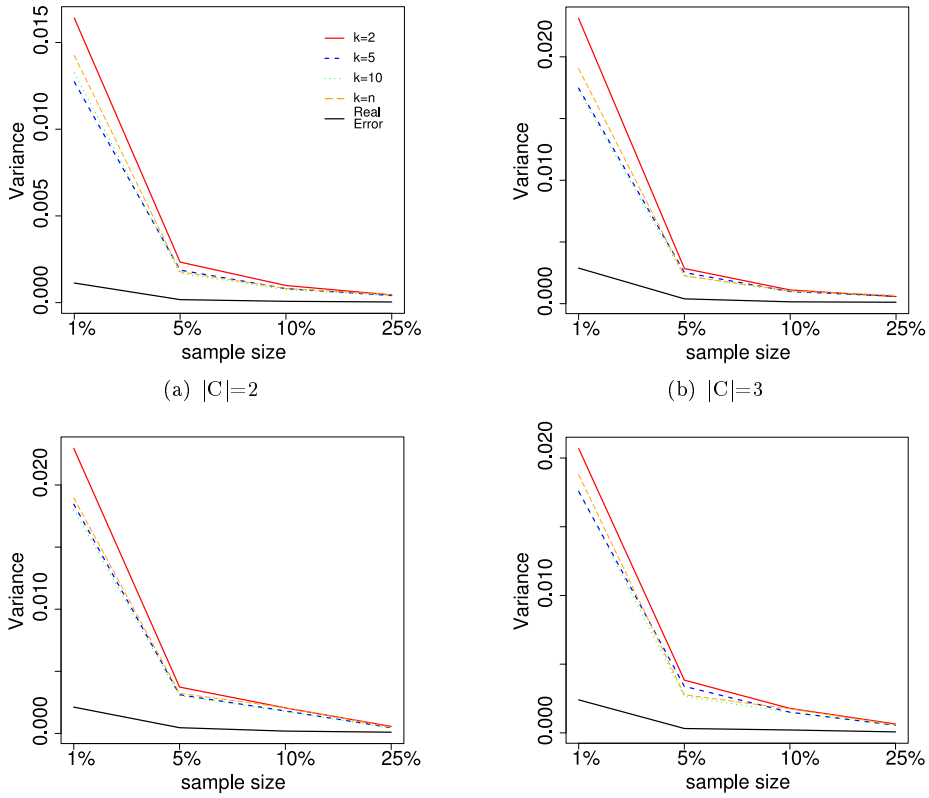
2-DB Structure



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 25: Prediction Error bias

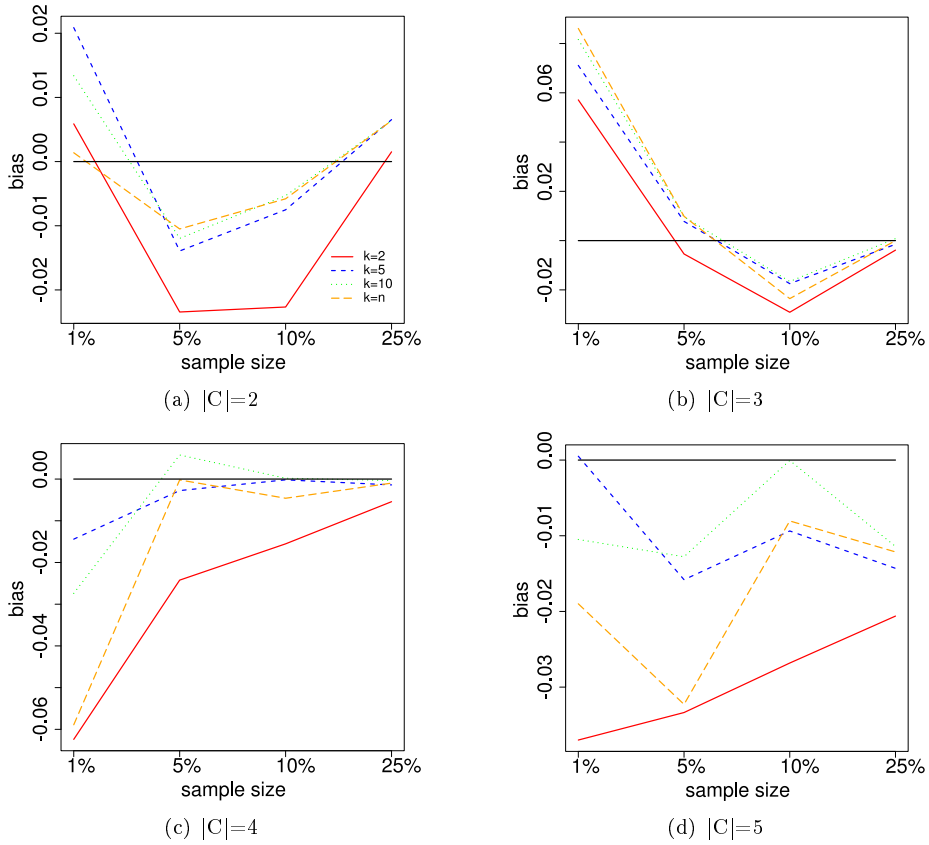


(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 26: Prediction Error variance

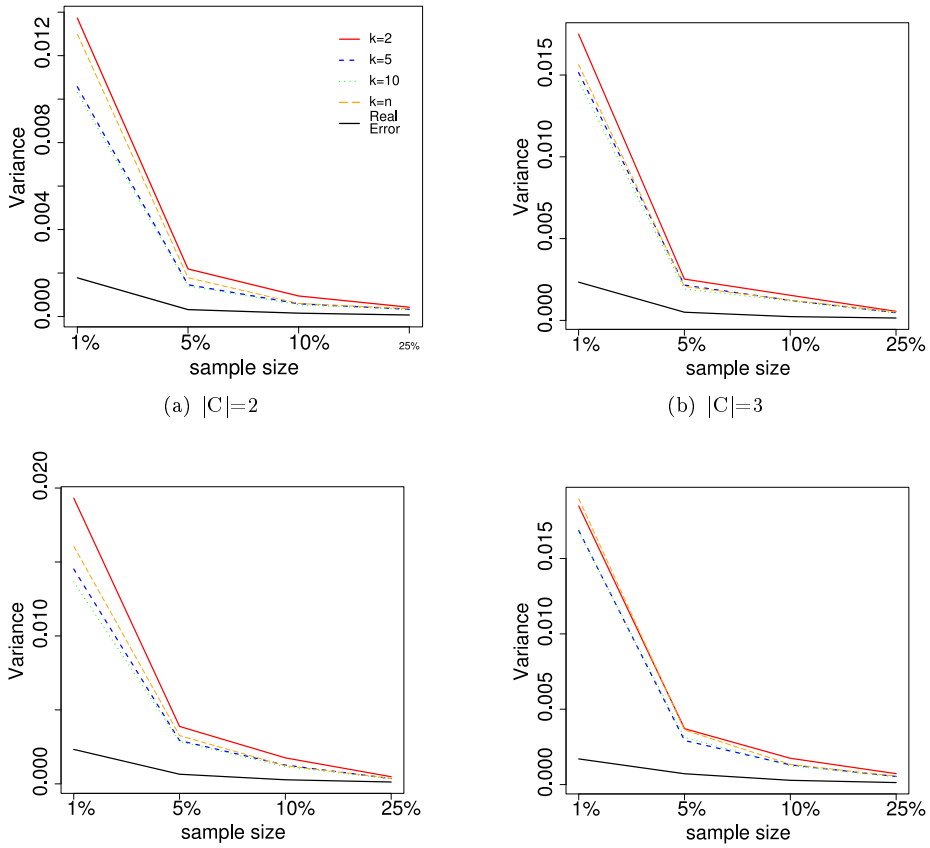
3-DB Structure



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 27: Prediction Error bias

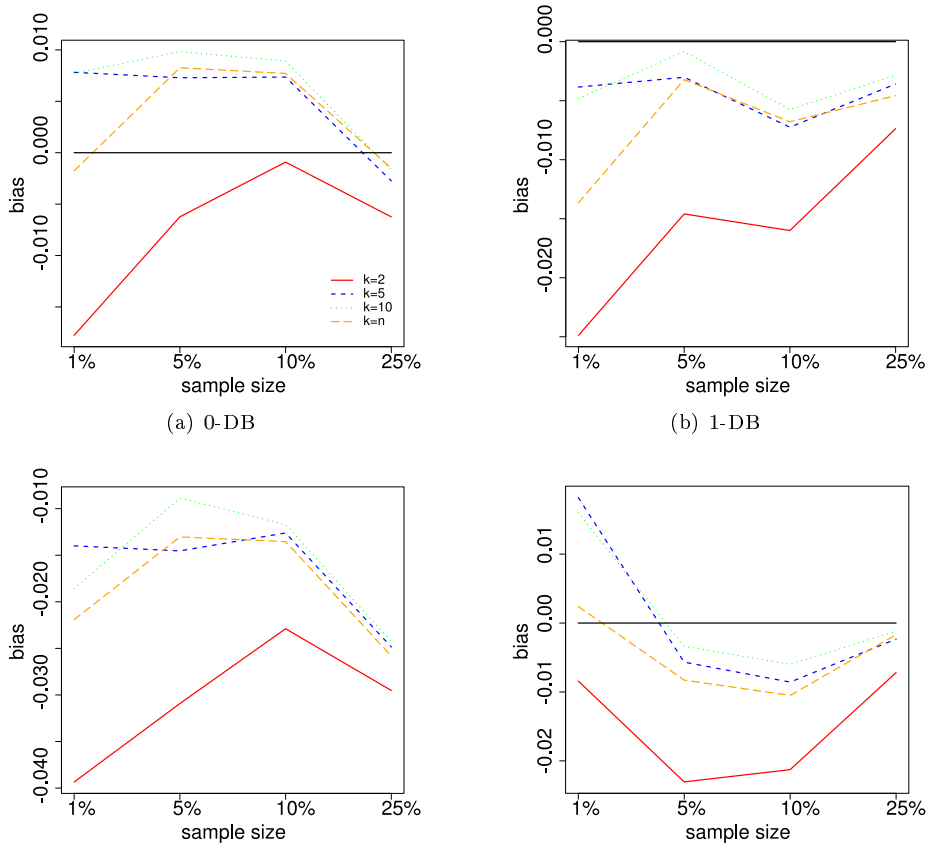


(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 28: Prediction Error variance

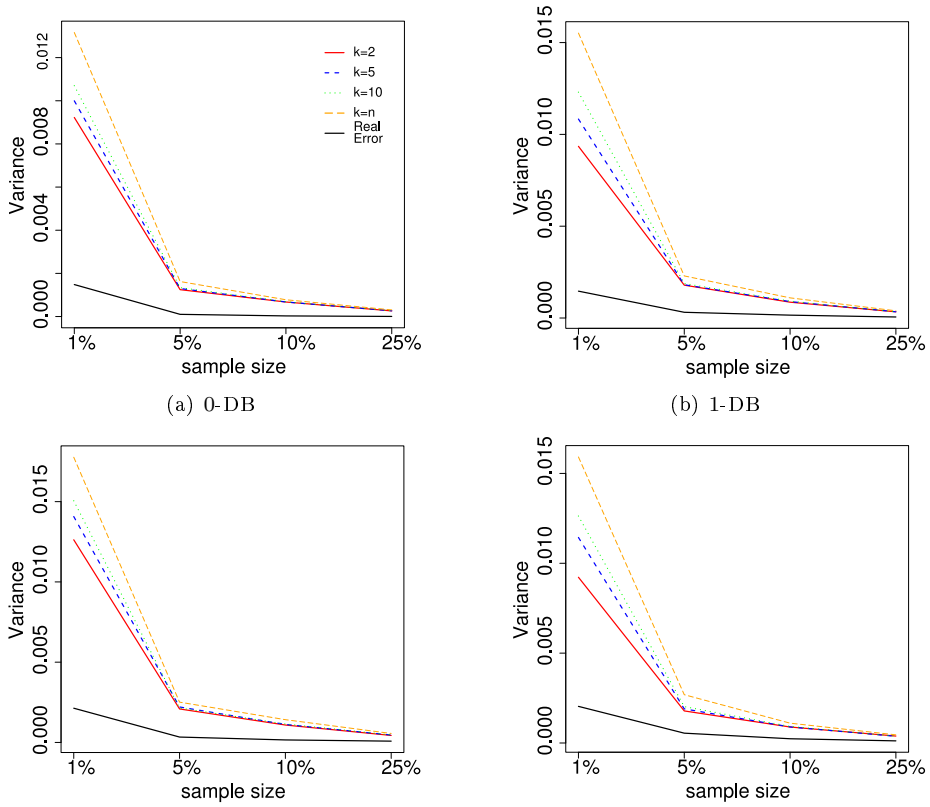
B.2 m-k-cv on naive Bayes classifier



(a) 0-DB (b) 1-DB

(c) 2-DB (d) 3-DB

Figure 29: Prediction Error bias



(a) 0-DB

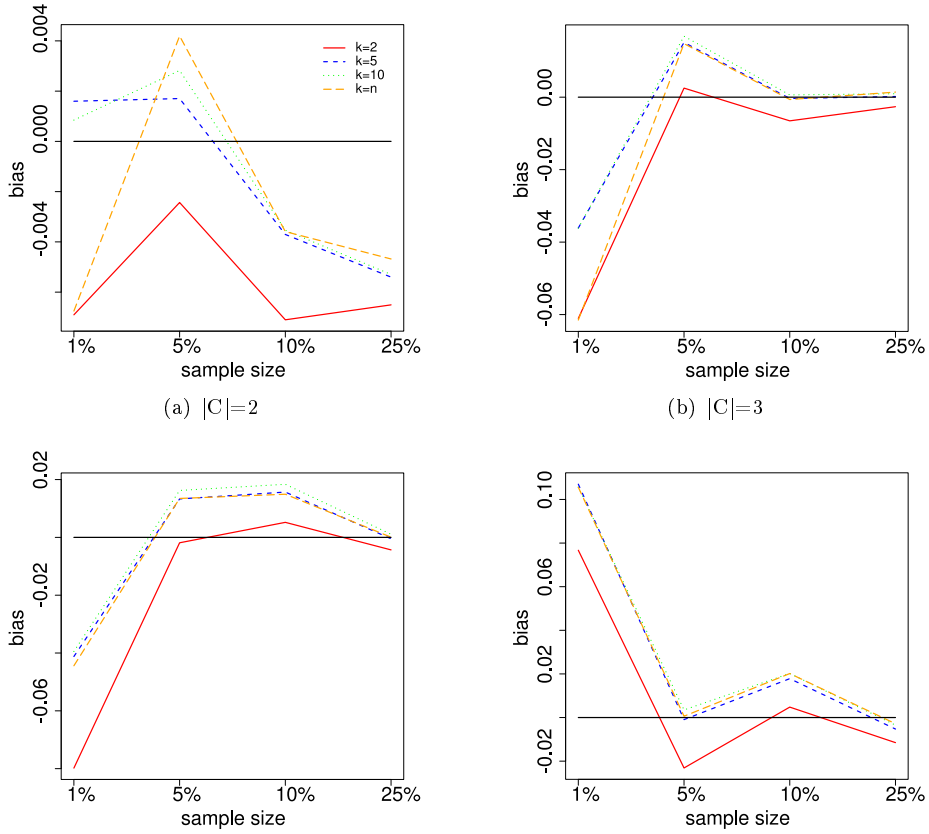
(b) 1-DB

(c) 2-DB

(d) 3-DB

Figure 30: Prediction Error variance

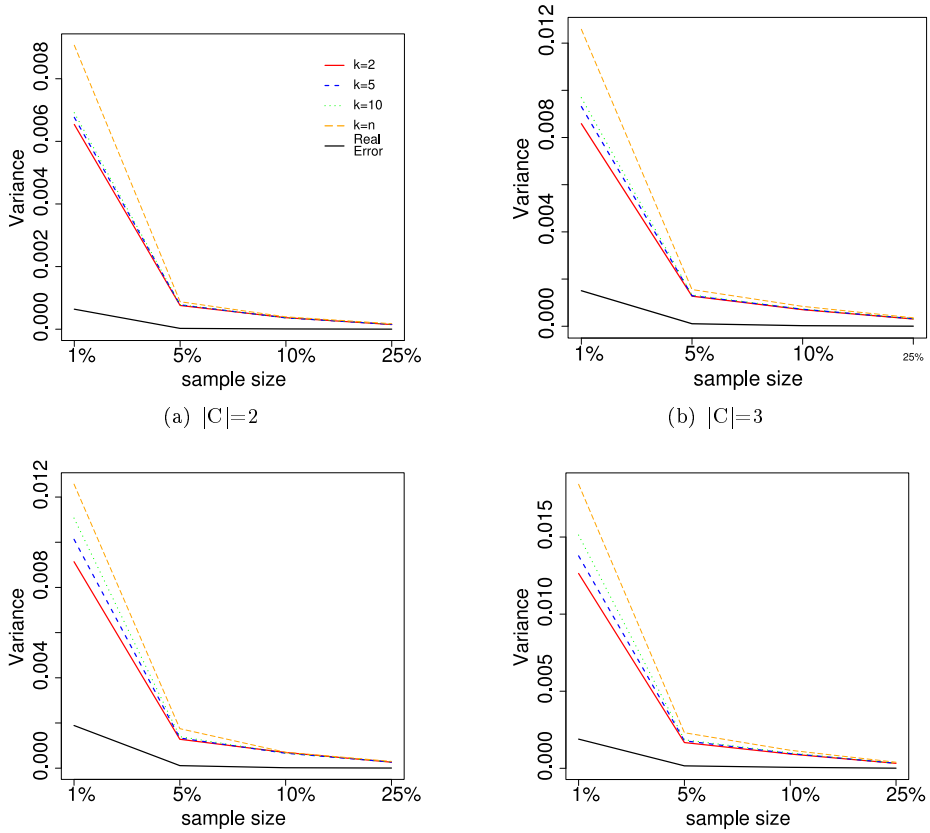
0-DB Structure



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 31: Prediction Error bias

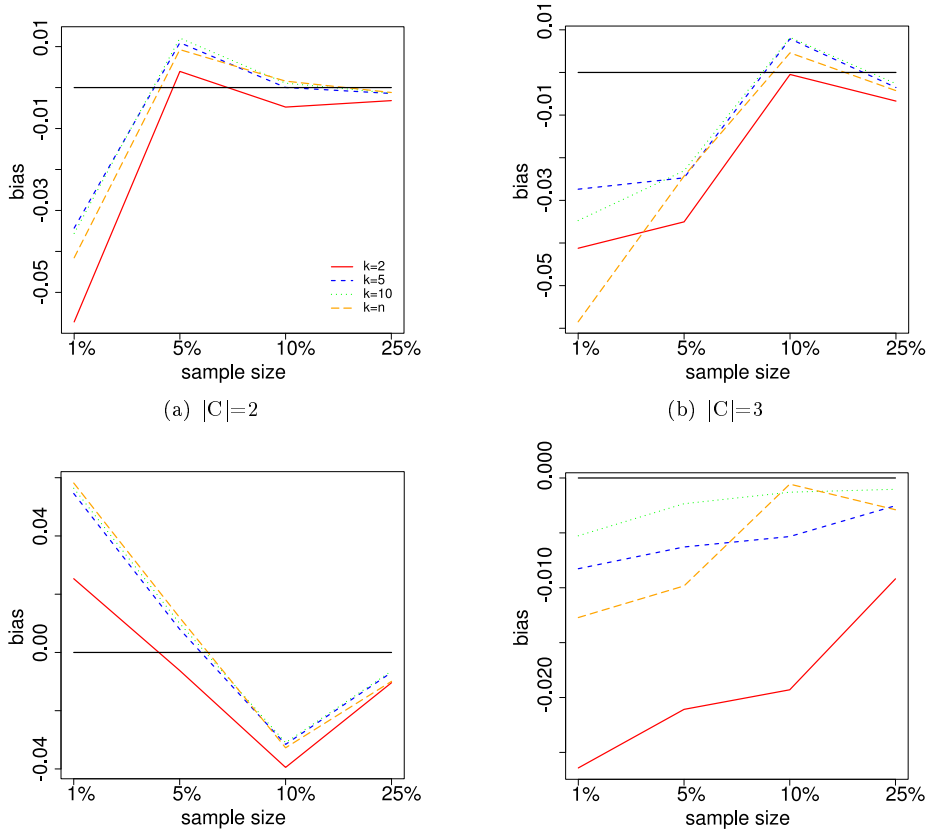


(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 32: Prediction Error variance

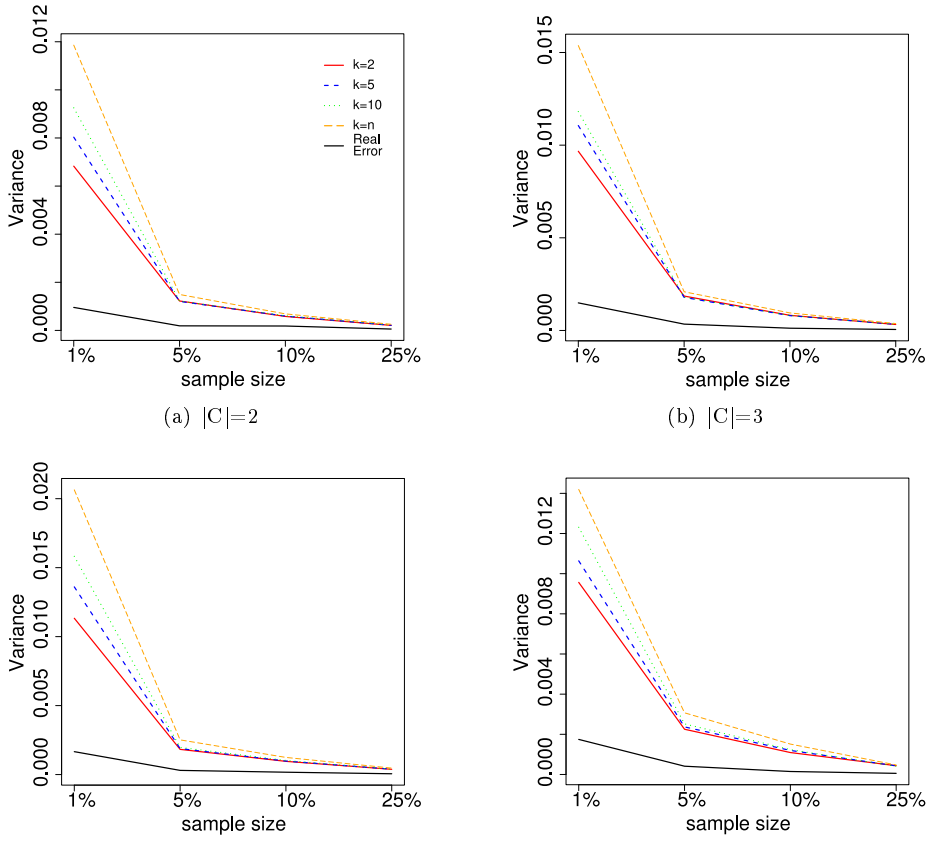
1-**DB Structure**



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 33: Prediction Error bias

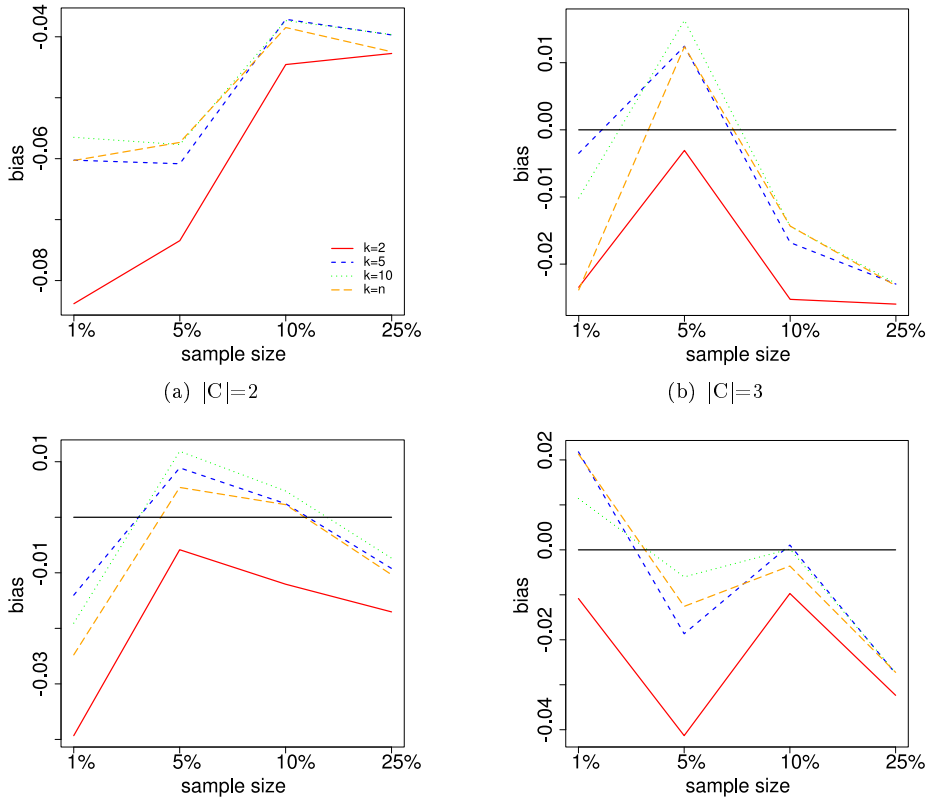


(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 34: Prediction Error variance

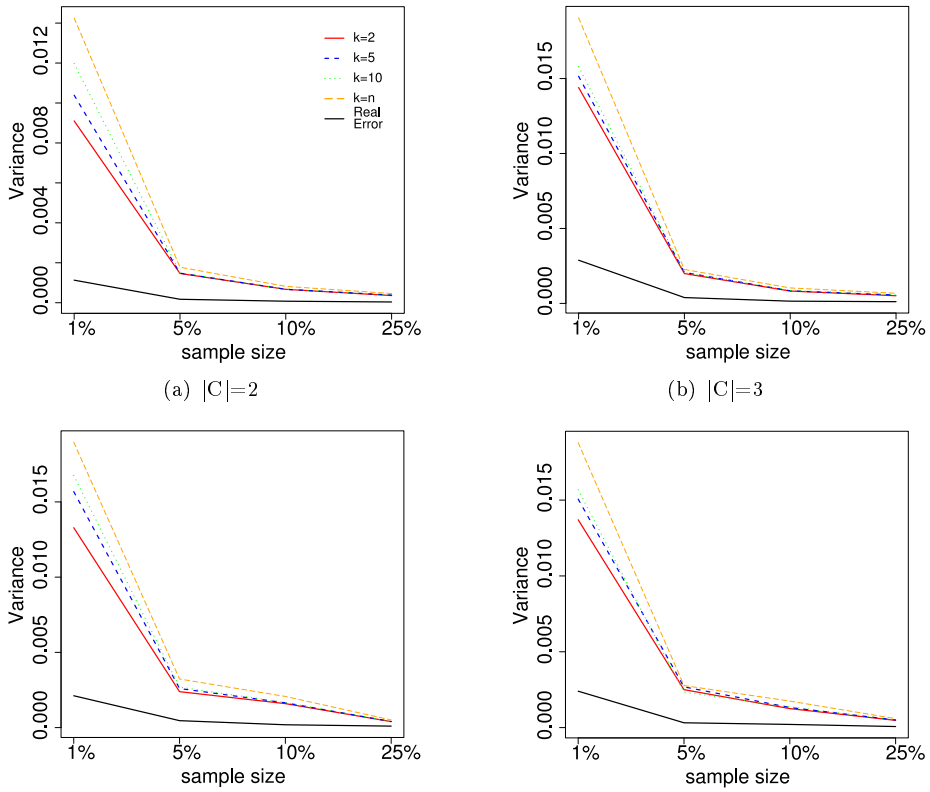
2-DB Structure



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 35: Prediction Error bias

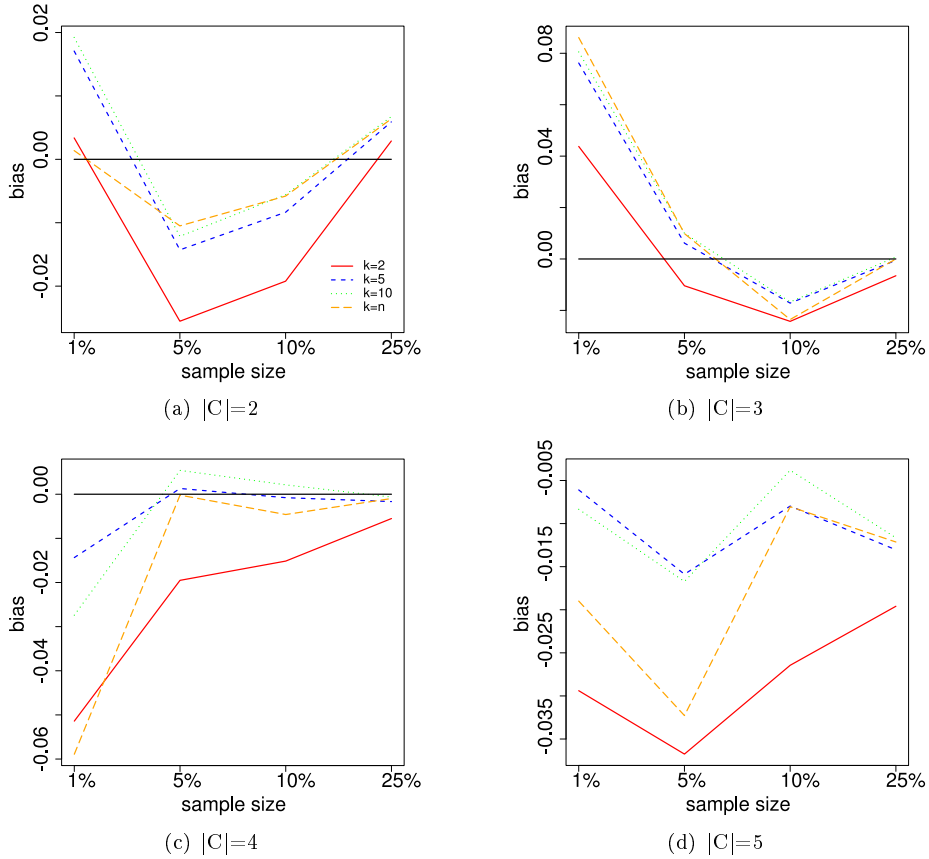


(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 36: Prediction Error variance

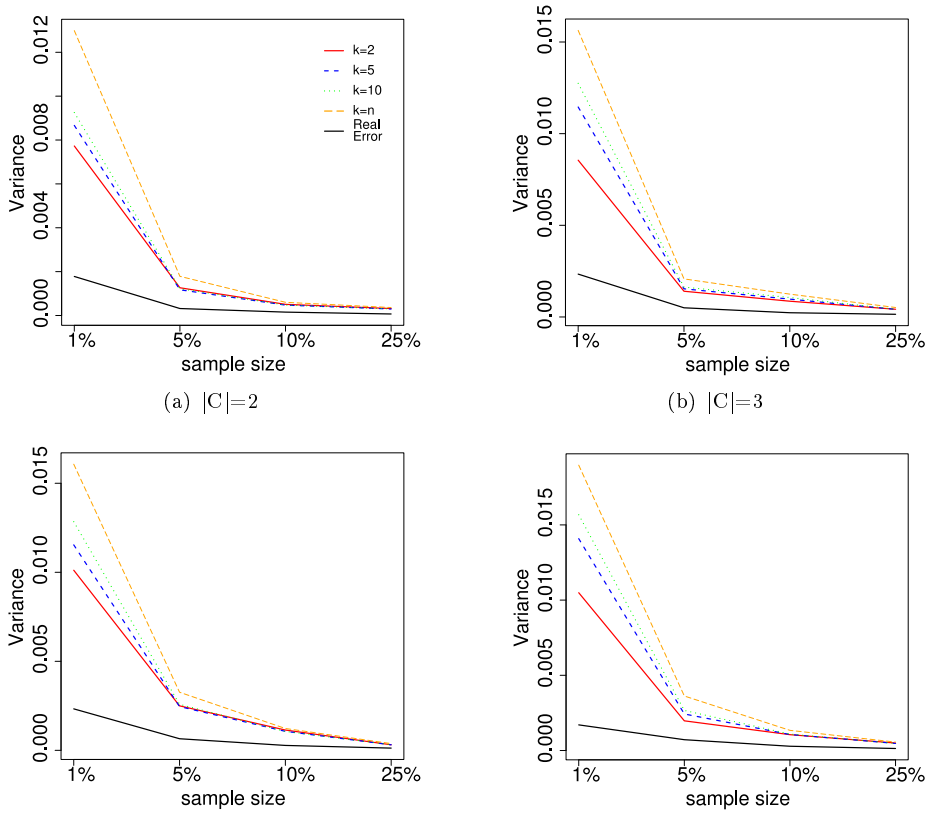
3-DB Structure



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 37: Prediction Error bias

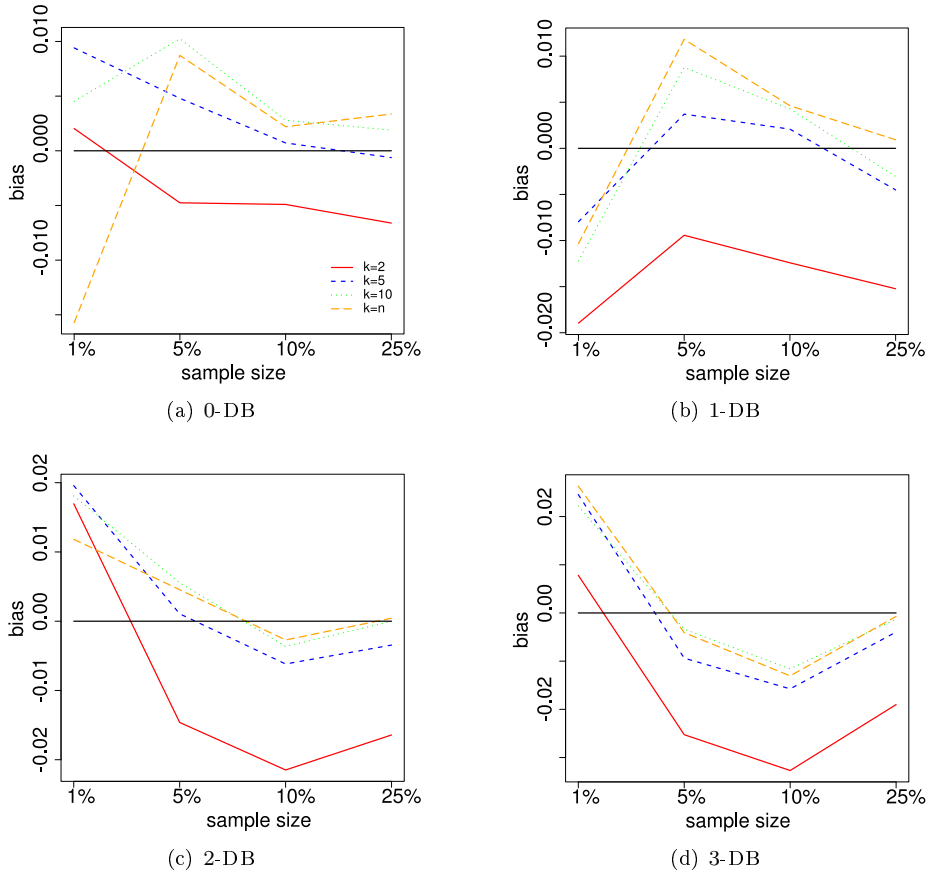


(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 38: Prediction Error variance

B.3 k-cv on nearest neighbour classifier



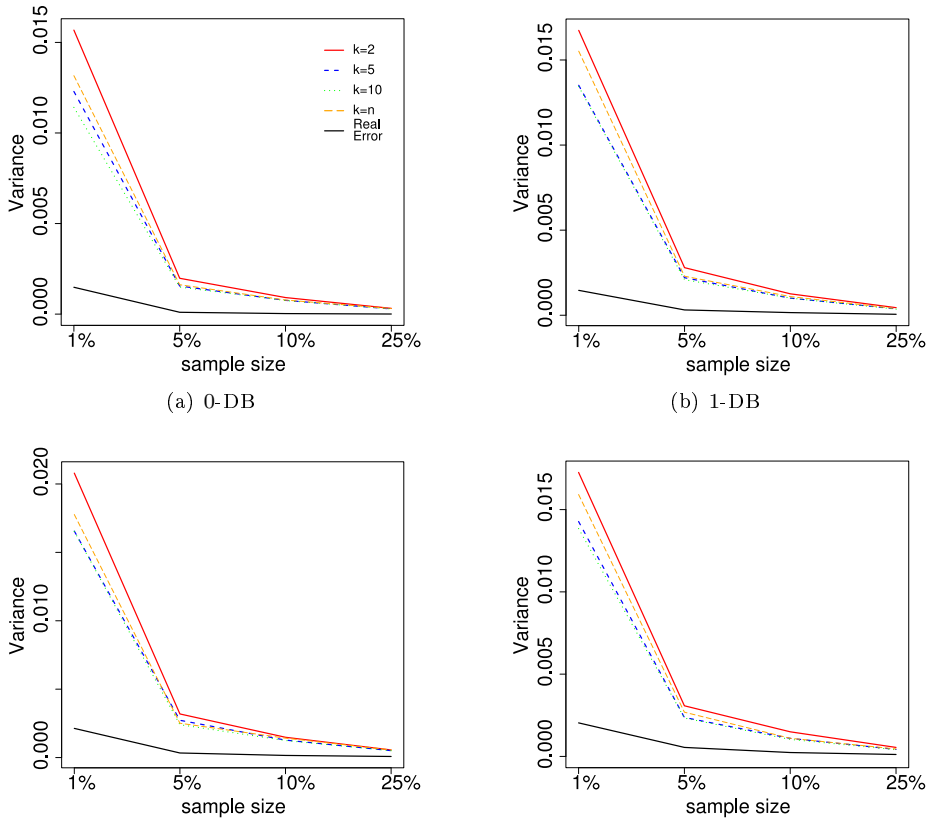
(a) 0-DB

(b) 1-DB

(c) 2-DB

(d) 3-DB

Figure 39: Prediction Error bias

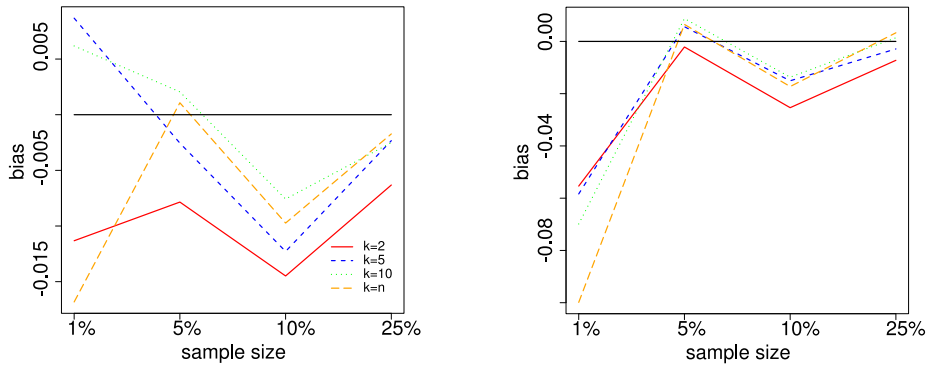


(a) 0-DB (b) 1-DB

(c) 2-DB (d) 3-DB

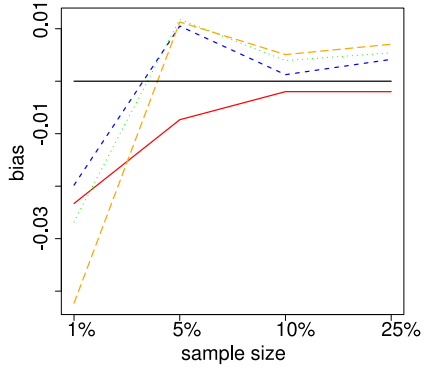
Figure 40: Prediction Error variance

0-DB Structure

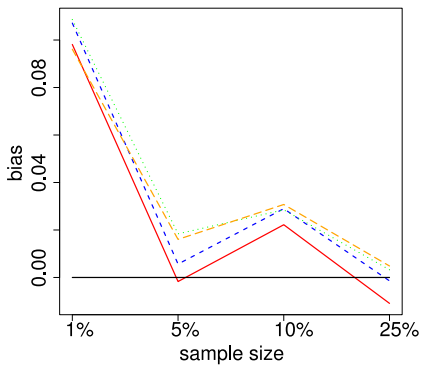


(a) |C|=2

(b) |C|=3

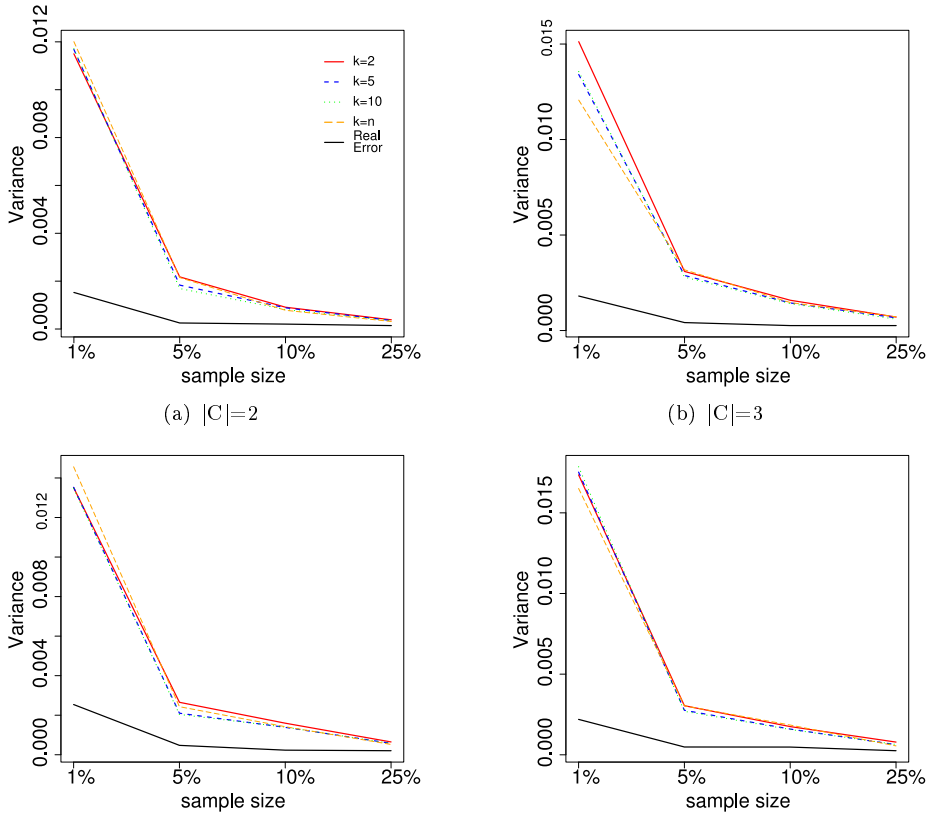


(c) |C|=4



(d) |C|=5

Figure 41: Prediction Error bias

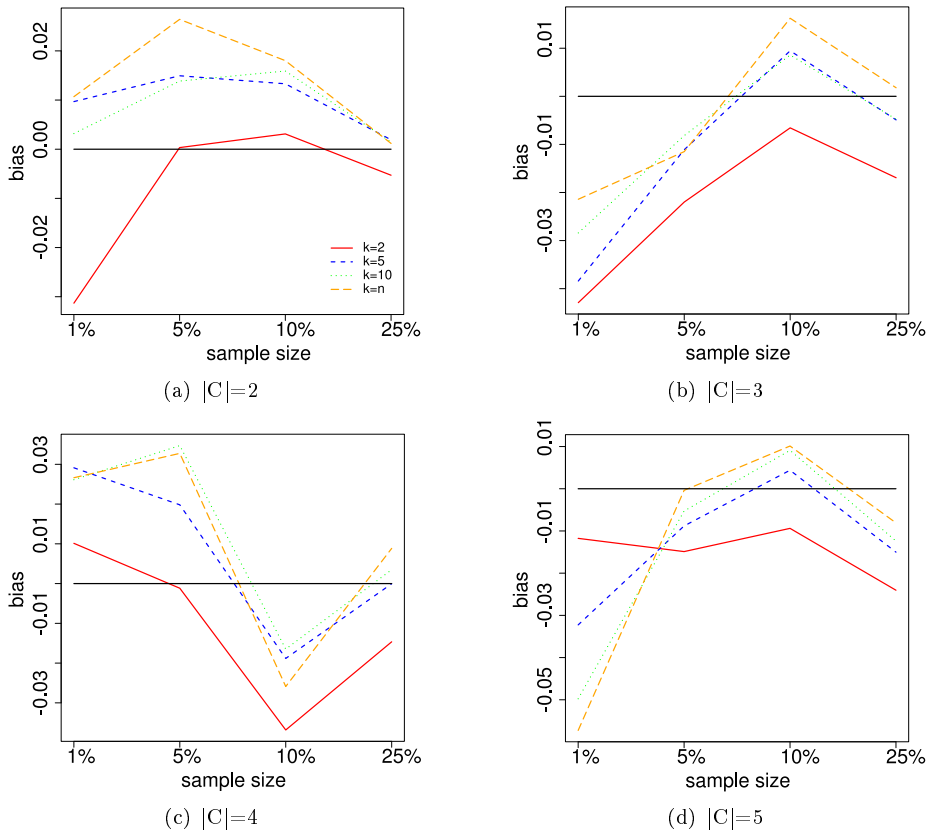


(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 42: Prediction Error variance

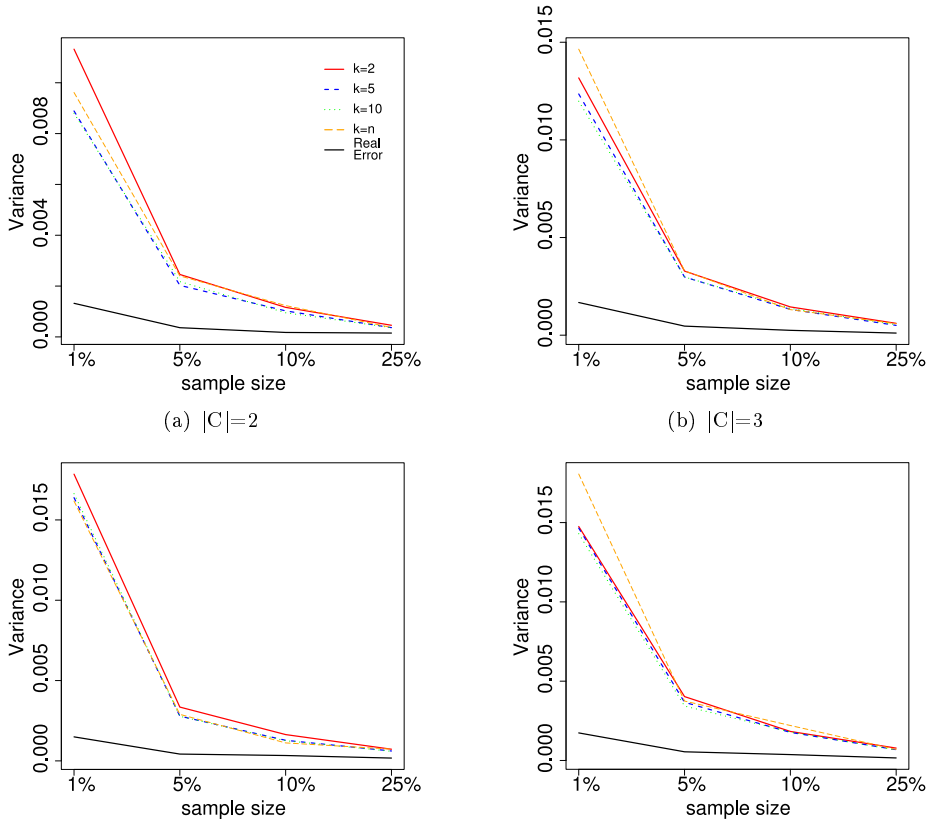
1-DB Structure



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 43: Prediction Error bias

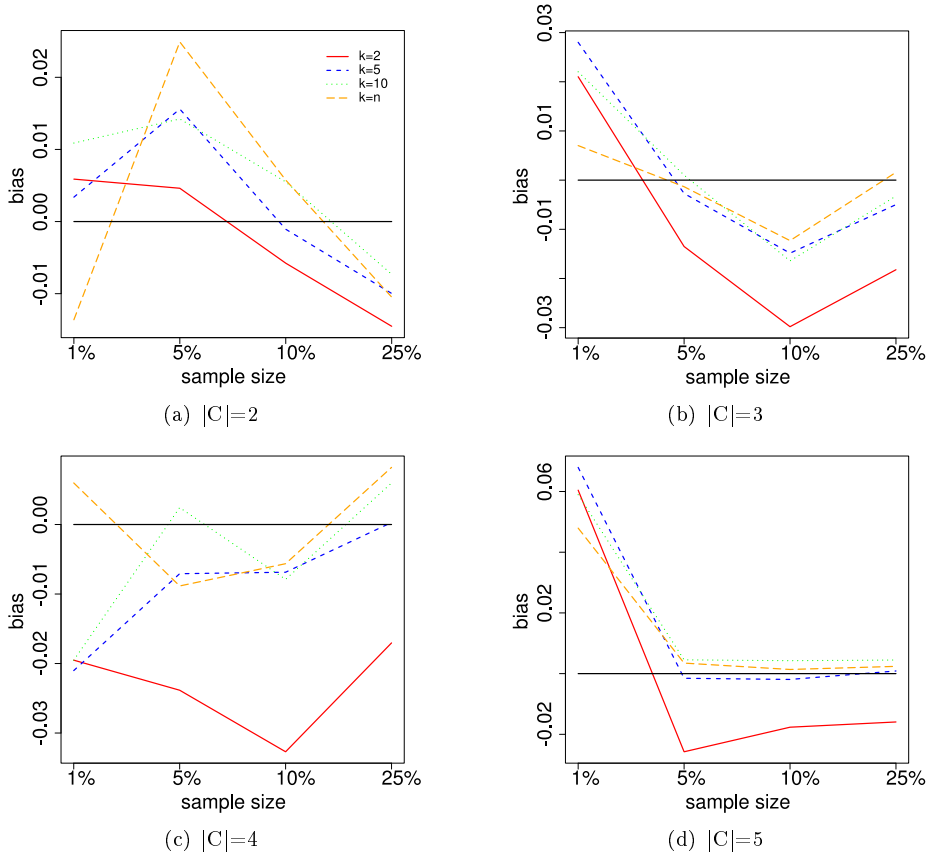


(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 44: Prediction Error variance

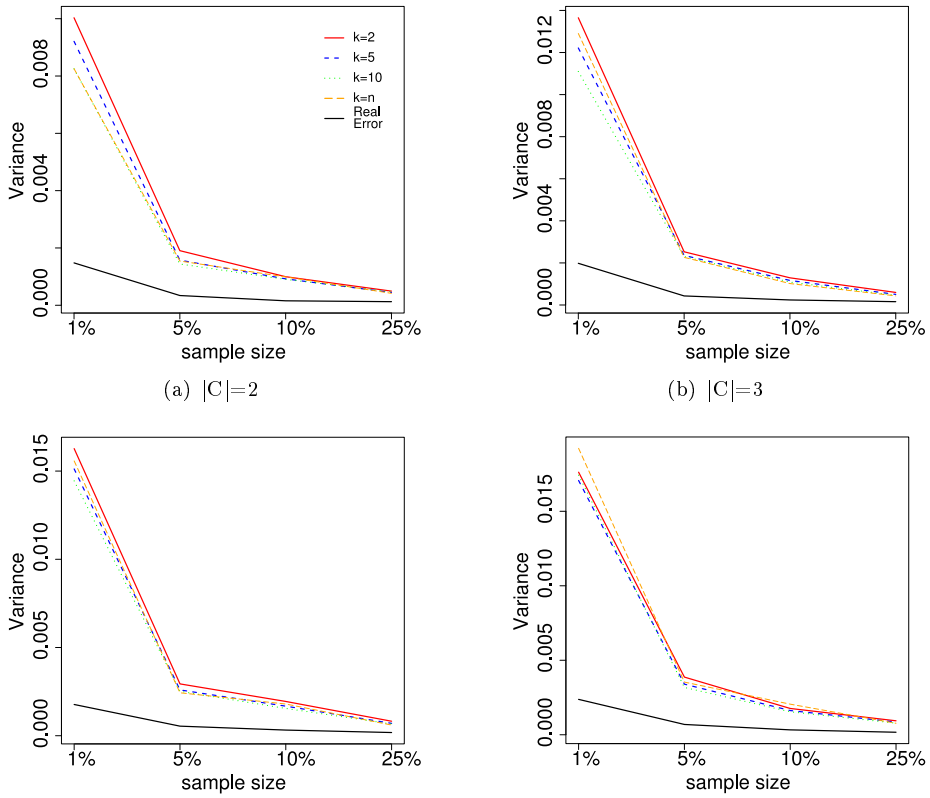
2-DB Structure



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 45: Prediction Error bias

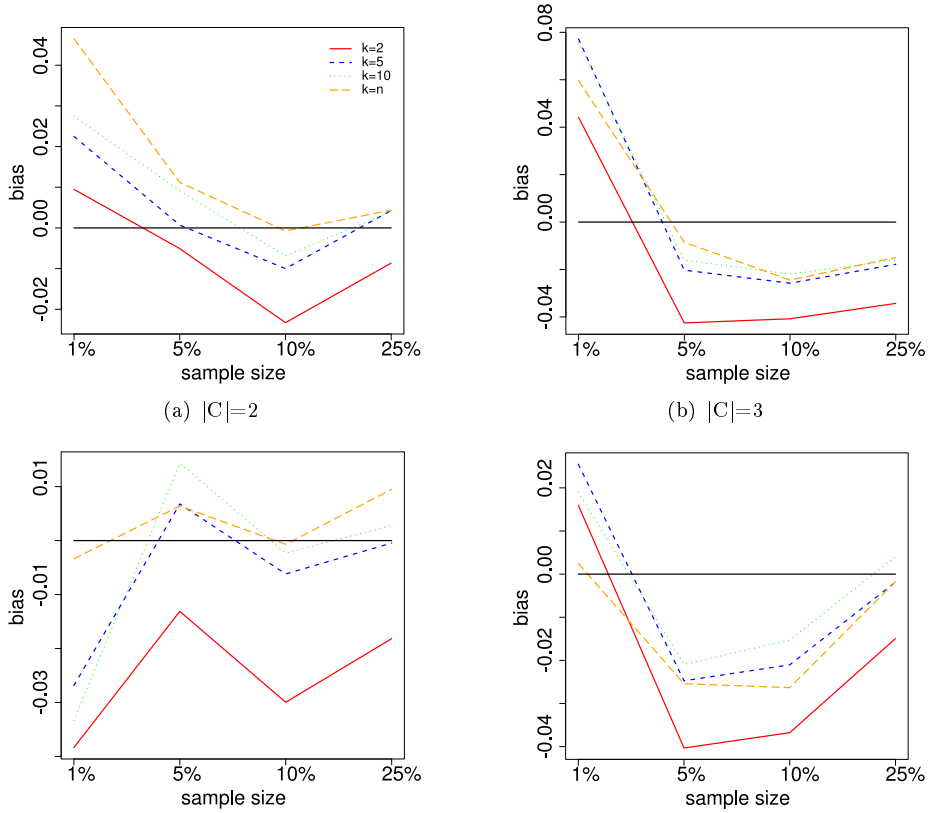


(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 46: Prediction Error variance

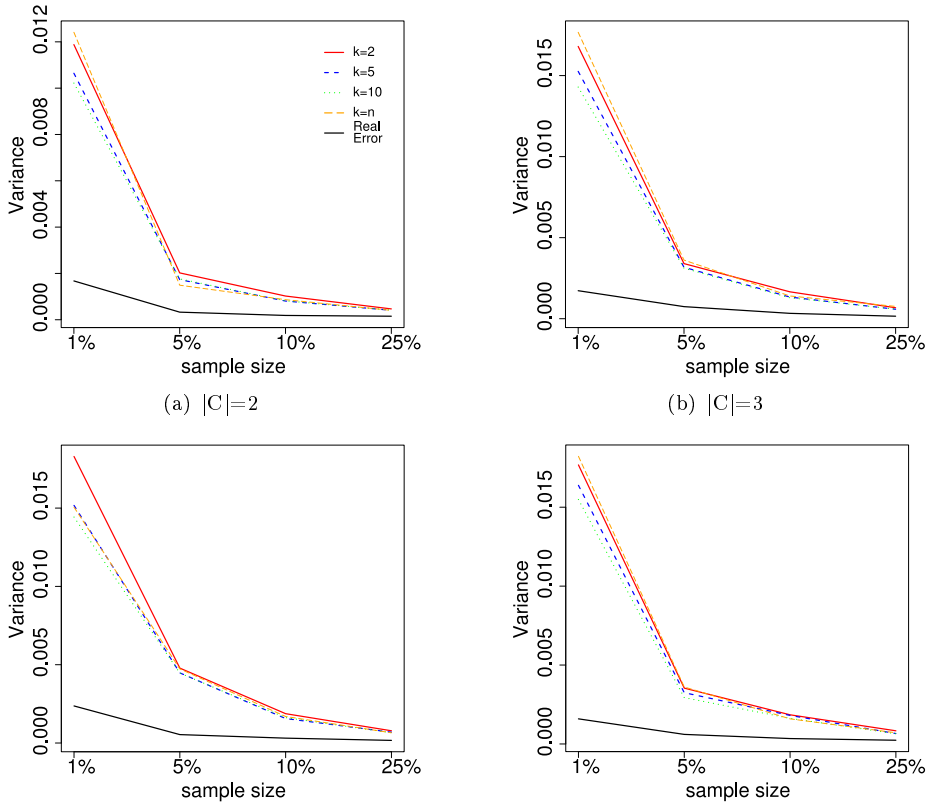
3-DB Structure



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 47: Prediction Error bias

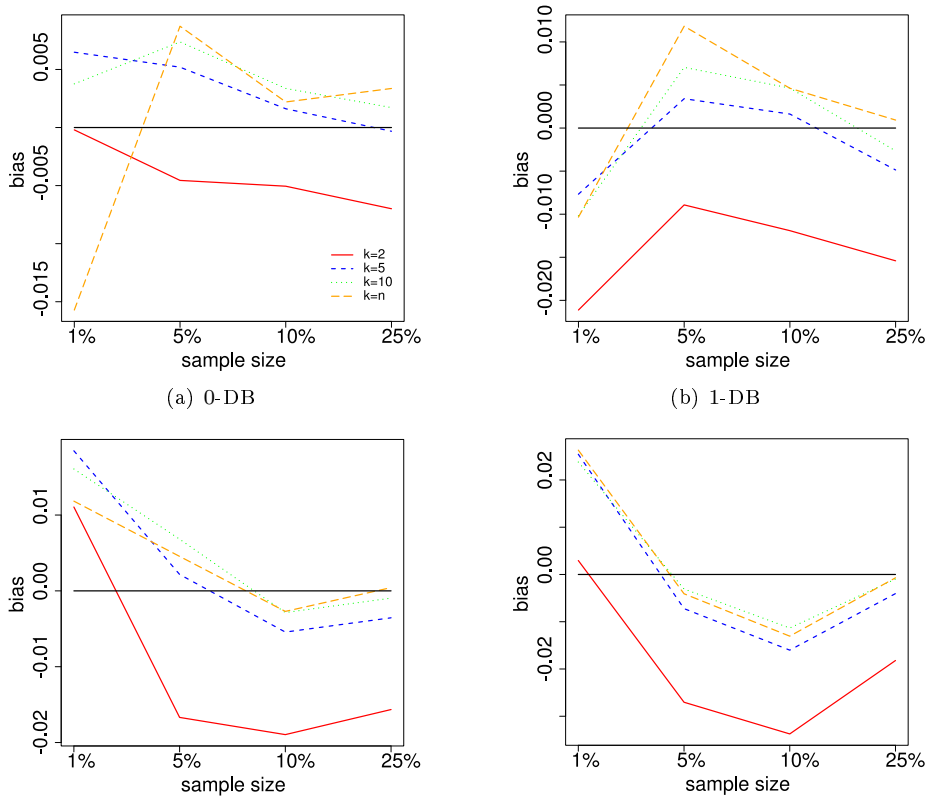


(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 48: Prediction Error variance

B.4 m-k-cv on nearest neighbour classifier



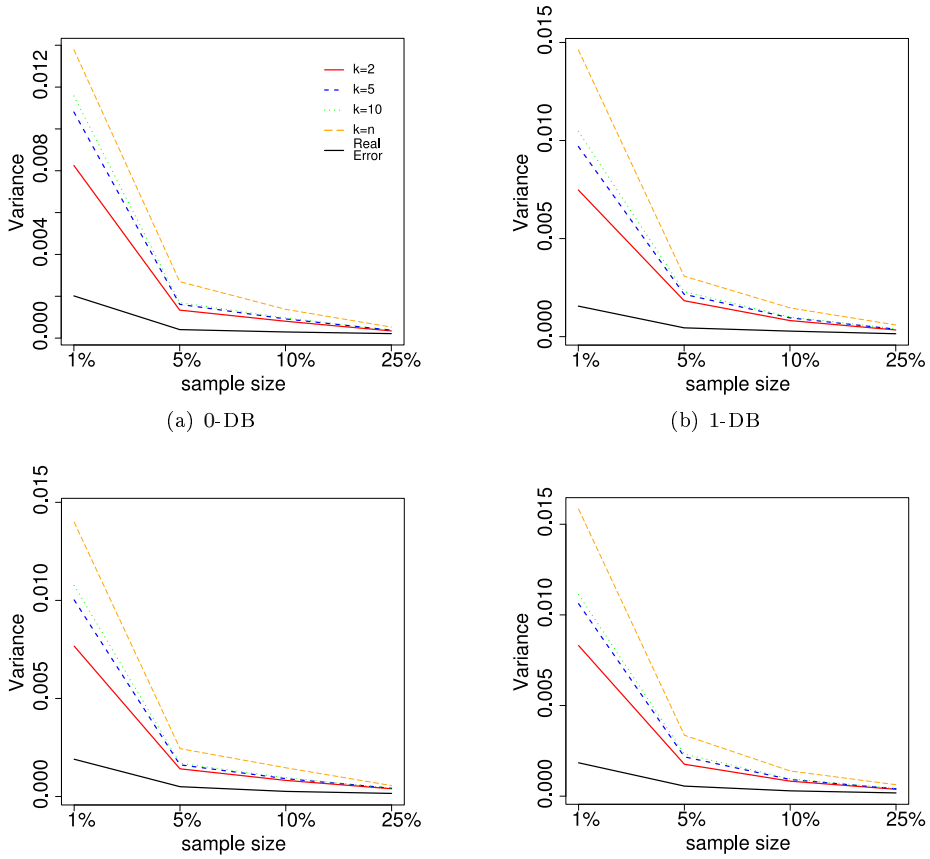
(a) 0-DB

(b) 1-DB

(c) 2-DB

(d) 3-DB

Figure 49: Prediction Error bias



(a) 0-DB

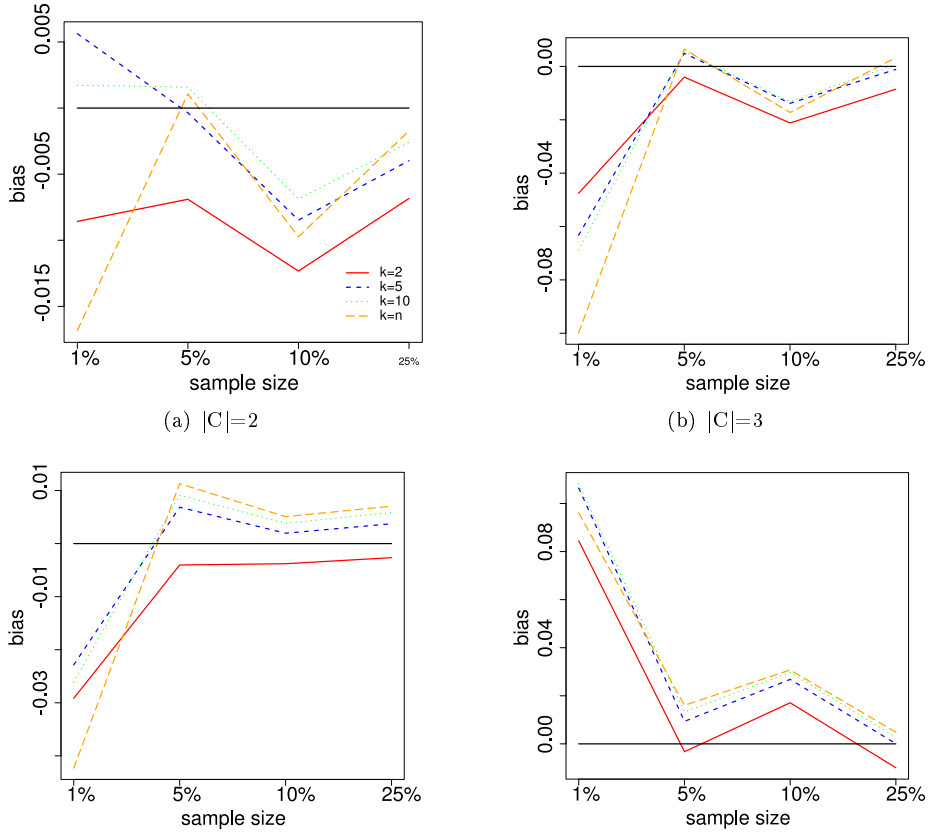
(b) 1-DB

(c) 2-DB

(d) 3-DB

Figure 50: Prediction Error variance

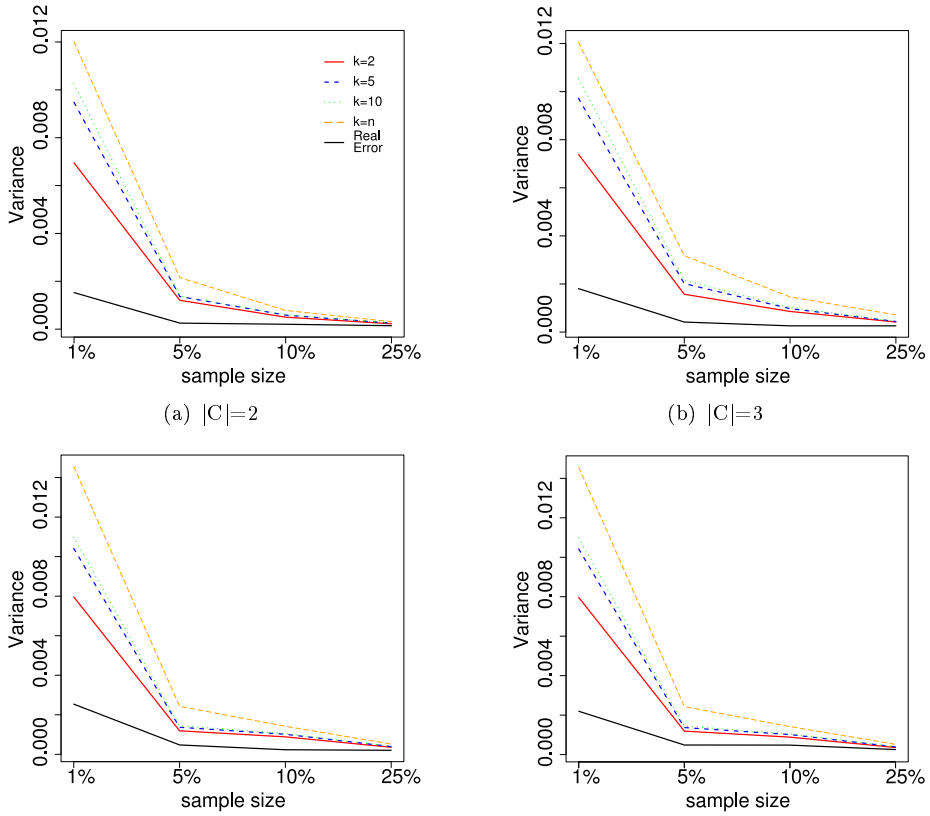
0-DB Structure



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

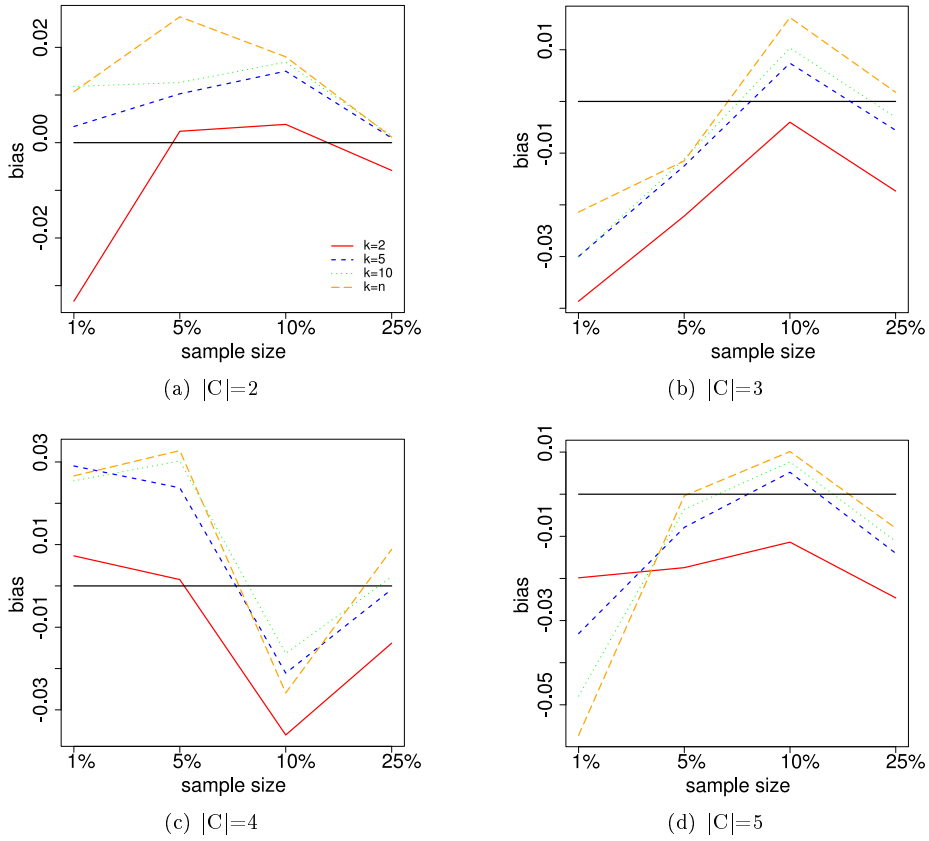
Figure 51: Prediction Error bias



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

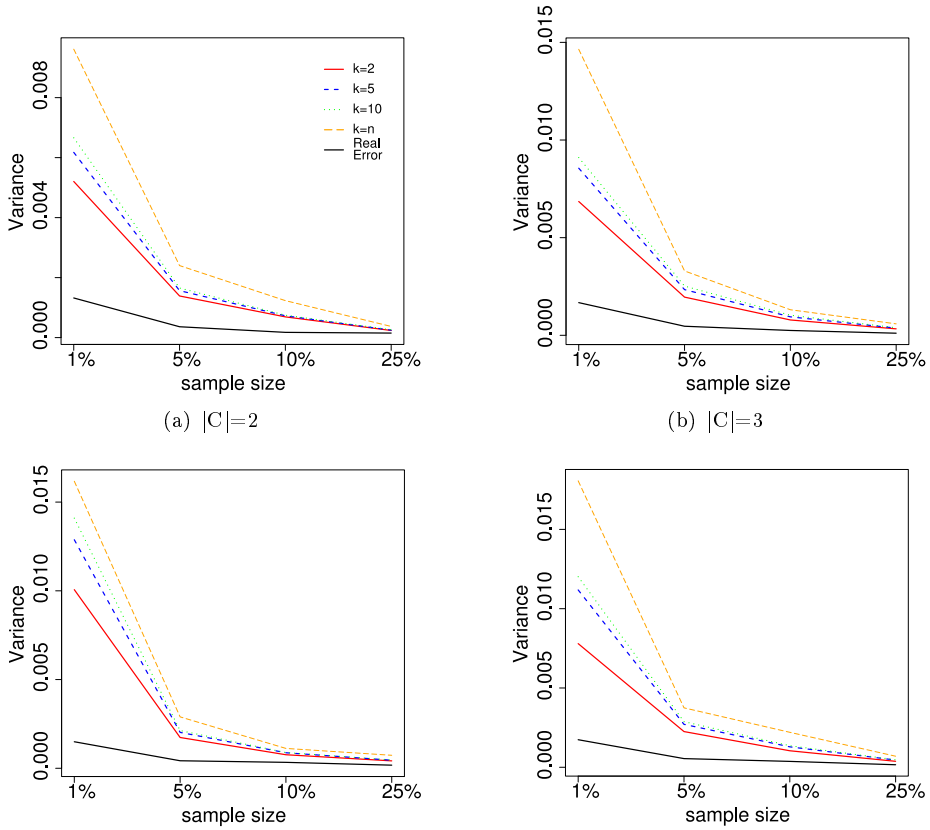
Figure 52: Prediction Error variance



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 53: Prediction Error bias

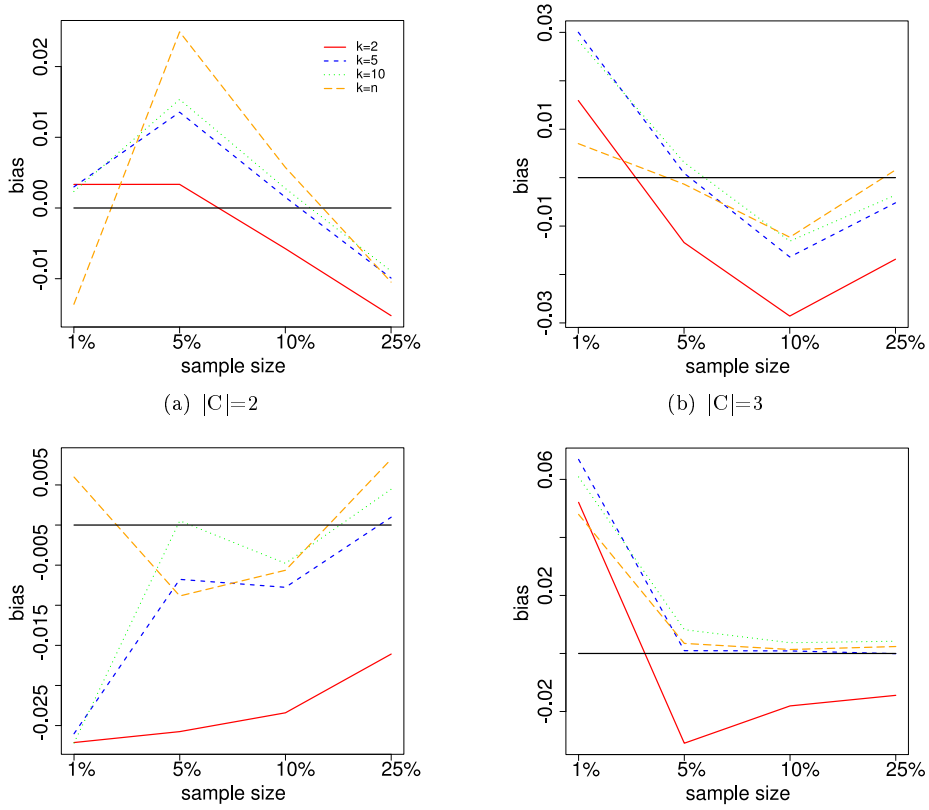


(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 54: Prediction Error variance

2-DB Structure



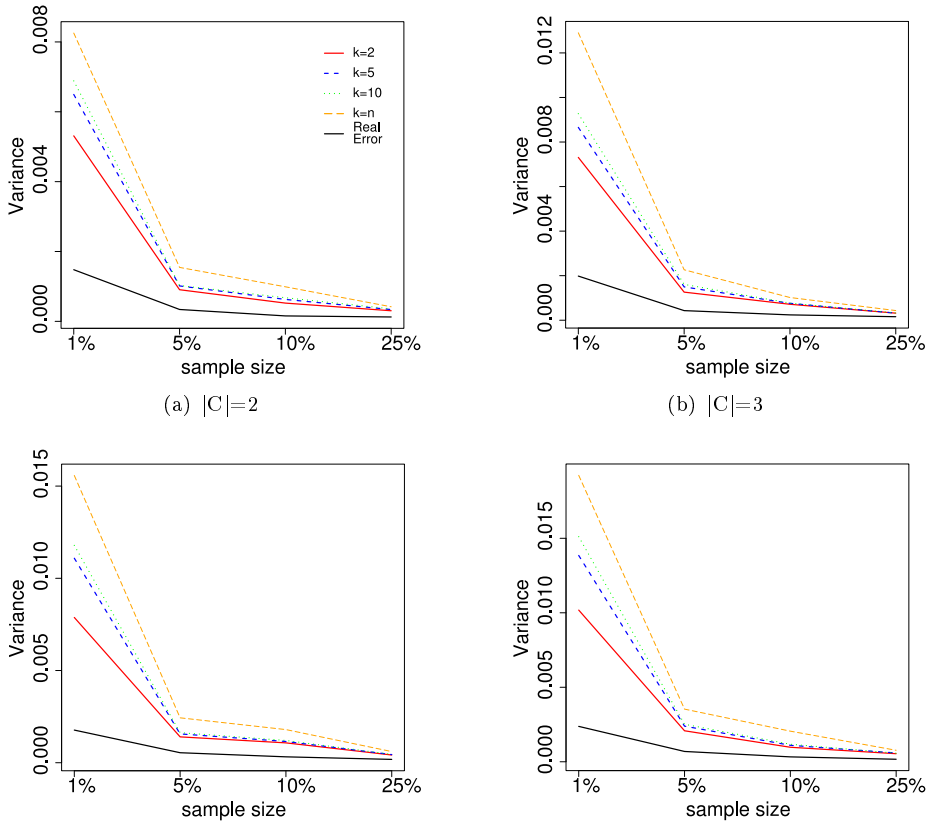
(a) |C|=2

(b) |C|=3

(c) |C|=4

(d) |C|=5

Figure 55: Prediction Error bias

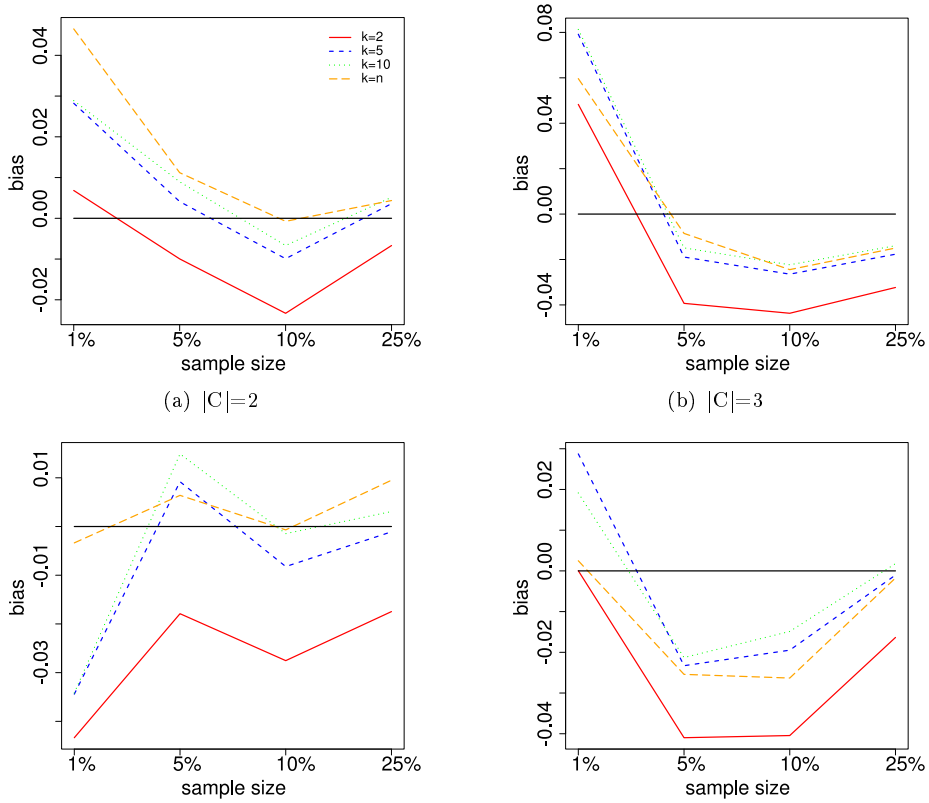


(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 56: Prediction Error variance

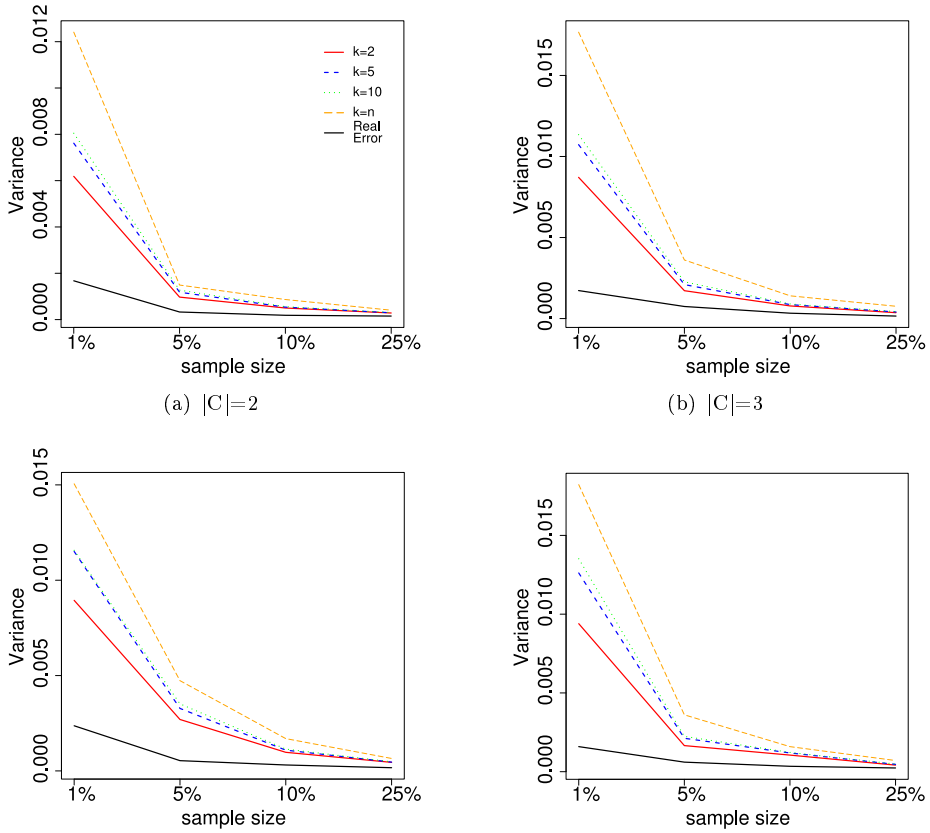
3-DB Structure



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 57: Prediction Error bias



(a) |C|=2 (b) |C|=3

(c) |C|=4 (d) |C|=5

Figure 58: Prediction Error variance

References

[1] Y. Bengio and Y. Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, 5:1089–1105, 2004.

[2] Y. Bengio and Y. Grandvalet. *Bias in estimating the variance of k-fold cross-validation*, volume 1 of *Statistical modeling and analysis for complex data problems*, pages 75–95. 2005.

[3] U. Braga-Neto, R. Hashimoto, E.R. Dougherty, D.V. Nguyen and R.J. Carroll. Is cross-validation better than resubstitution for ranking genes? *Bioinformatics*, 20(2):253–258, 2004.

[4] U. M. Braga-Neto. Small-sample error estimation: mythology versus mathematics. In *Proceedings of SPIE*, volume 5916, pages 304–314, 2005.

[5] U. M. Braga-Neto and E. R. Dougherty. Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20(3):374–380, 2004.

[6] J. H. Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1:55–77, 1997.

[7] L. Breiman and P. Spector. Submodel selection and evaluation in regression. The x random case. *International Statistical Review*, 60 (3):291–319, 1972.

[8] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[9] L. Devroye. *Non-Parametric Density Estimation*. Wiley Series in Probability and Mathematical Statistics. 1985.

[10] L. Devroye and T. Wagner. Distribution-free performance bounds with the resubstitution error estimate. *IEEE Transactions on Information Theory*, 25(2):208–210, 1979.

[11] R.O. Duda, P.E. Hart and D.G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[12] B. Efron and R.J. Tibshirani. *An introduction to the bootstrap*, volume 57 of *Monographs on statistics and applied probability*. Chapman and Hall, 1993.

[13] J. S. Ide and F. G. Cozman. Random generation of Bayesian networks. In *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence*, pages 365–375, 2002.

[14] J. S. Ide and F. G. Cozman. Generating random Bayesian networks with constraints on induces widht. *ECAI 04 IOS Press*, 2004.

[15] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, pages 1137–1145, 1995.

[16] R. Kohavi. *Wrappers for Performance Enhancement and Oblivious Decision Graphs*. PhD thesis, Stanford University, Stanford, USA, Computer Science Department, 1995.

[17] R. Kohavi and D. H. Wolpert. Bias plus variance decomposition for zero-one loss functions. In *International Conference on Machine Learning*, pages 275–283, 1996.

[18] G. M. James. Variance and bias for general loss functions. *Machine Learning*, 51:115–135, 2003.

[19] P. Langley, W. Iba and K. Thompson. An analysis of bayesian classifiers. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 223–228, 1992.

[20] P. Lucas. Restricted Bayesian network structure learning. In *Advances in Bayesian Networks, Studies in Fuzziness and Soft Computing*, pages 217–232, 2004.

[21] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society Series B*, 36:111–147, 1974.

[22] G.J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley and Sons Inc., New York, 1992.

[23] M. Minsky. Steps toward artificial intelligence. *Transactions on Institute of Radio Engineers*, 49:8–30, 1961.

[24] J. Pearl. *Probabilistic Reasoning in Intelligence Systems*. Springer series in Statistics. Morgan-Kaufman, 1988.

[25] M. Sahami. Learning limited dependence Bayesian classifiers. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 335–338, 1996.

[26] P. Domingos. A unified bias-variance decomposition and its applications In *Proceedings of the 17th International Conference on Machine Learning*, pages 231–238, 2000.

[27] S.M. Weiss and C.A. Kulikowski. *Computer systems that learns*. Morgan-Kaufmann, 1991.

[28] I. H. Witten and E. Frank. *Data mining: practical machine learning tools and techniques with Java implementations*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, 2000.

[29] P. Zhang. On the distributional properties of model selection criteria. *Journal of the American Statistical Association*, 87 (419):732–737, 1992.