Bayesian Network Classifiers Under the Ensemble Perspective

Jacinto Arias JACINTO.ARIAS@UCLM.ES
José A. Gámez JOSE.GAMEZ@UCLM.ES
José M. Puerta JOSE.PUERTA@UCLM.ES
Department of Computing Systems. University Castilla-La Mancha

Abstract

Augmented naive Bayesian classifiers relax the original independence assumption by allowing additional dependencies in the model. This strategy leads to parametrized learners that can produce a wide spectrum of models of increasing complexity. Expressiveness and efficiency can be controlled to adjust a trade-off specific to the problem at hand. Recent studies have transposed this finding to the domain of bias and variance, demonstrating that inducing complex multivariate probability distributions produces low-bias/high-variance classifiers that are especially suitable for large data domains. Frameworks like AkDE avoid structural learning and reduce variance by averaging a full family of constrained models, at the expense of increasing its spatial and computational complexity. Model selection is then required and performed using Information Theory techniques. We present a new approach to reduce model space from the point of view of ensemble classifiers, where we study the individual contribution to error for each model and how model selection affects this via the aggregation process. We perform a thorough experimentation to analyse bias stability and variance reduction and compare the results within the context of other popular ensemble models such as Random Forest, leading to a discussion on the effectiveness of the previous approaches. The conclusions support new strategies to design more consistent ensemble Bayesian network classifiers which we explore at the end of the paper.

Keywords: Supervised Classification; Bayesian Network Classifiers; Ensemble Classifiers.

1. Introduction

Current trends in technology and their impact on society are enabling the general availability of large datasets and new high-performance computing platforms. Machine Learning (ML) researchers are continuously designing new techniques to adapt to this context, where the scalability and performance of the algorithms must be balanced. Consequently, we often produce overly obfuscated and complex models that may prove difficult for practitioners in the industry to implement.

By studying the most popular algorithms in the reference software packages (Meng et al., 2016), we can observe that most were proposed more than a decade ago, but can still prove to be competitive and considered as the state of the art. In addition to their sound foundations, these models usually provide an intuitive understanding of their inner workings and approachable hyperparameter interfaces for users. In the field of supervised classification, one of the techniques that best fits this description is the Random Forest (RF) (Breiman, 2001) algorithm. By varying the number of trees in the ensemble, both the complexity of the model and its performance can be controlled. Increasing the number of trees has been shown to reduce classification variance asymptotically, while maintaining a stable bias. This guarantees that the performance will either improve or stabilize, allowing this parameter to be set predictably, based only on the available resources and the difficulty of the problem at hand.

Understanding the performance of an algorithm by studying its bias and variance is an easy and natural approach to work with. In this paper we will apply these concepts to the family of Bayesian Network classifiers (BNCs). As out-of-core learners, they can be learned in a single pass through the data, being a good fit for large data domains (Arias et al., 2017). Algorithms such as kDB (Sahami, 1996) provide a single parameter k which controls the complexity of the model, with larger values reducing its bias at the cost of increasing variance and complexity. Another popular approach is AkDE (Webb et al., 2012), which is an ensemble of averaged classifiers, providing variance reduction at the cost of largely increasing model complexity.

The size of the models can pose a scalability problem for many domains and for that reason, several proposals have attempted to reduce model space by performing model selection (Martínez et al., 2016; Chen et al., 2017a,b). The majority of these perform a hybrid filter and wrapper selection based on Information Theory metrics such as Mutual Information (MI), which introduces additional passes through the data. In addition, they produce the sense of obfuscation mentioned above, as the outcome is inconsistent from one problem to another.

Our work proposes a novel approach to evaluate these techniques in terms of bias and variance by analysing their behaviour in domains of different sizes. The results show that the algorithm behaves differently in large domains and in traditional small sample ML benchmarks. By comparing the bias and variance stability of AKDE against RF, we have discovered that there are significant discrepancies that could change the way we understand and work with the AkDE framework, especially as regards its application to large datasets and the implications of performing model selection on an ensemble. We build on these new results to propose a new approach for ensemble-based BNCs that maintains the expected properties of other popular ensemble classifiers such as RF.

The following section starts with a review of BNCs and the recent advances leading to an introduction to bias and variance decomposition metric for ensemble classifiers. We then conduct thorough experimentation using both a large data domain and the classical ML benchmark used to originally evaluate the aforementioned classifiers. The paper ends with an analysis of the results obtained and a proposal for a new ensemble algorithm, which we will briefly explore and evaluate.

2. Bayesian Network Classifiers

We define the task of supervised classification as the problem of assigning a label $y \in \Omega_Y$, from a set of c labels of the variable Y to an example $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_d})$ with values for d attributes in the set $\mathcal{A} = \{X_1, \ldots, X_d\}$. For this purpose we wish to induce a model from a dataset $\mathcal{D}$ consisting of m labelled examples $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$.

Using a Bayesian network (BN) (Pearl, 2014) we can compute the joint probability for such random variables. The formalism provides a graphical representation in the form of a directed acyclic graph (DAG) that permits efficient decomposition of the joint probability distribution: $p(\mathbf{x}) = \prod_{\mathbf{i=1}}^{\mathbf{d}} \mathbf{p}(\mathbf{x_i} \mid \pi_{\mathbf{x_i}})$, where π_{x_i} denotes the parents of attribute X_i. From a probabilistic approach, given an example $\mathbf{x}$ the problem at hand can be solved by estimating $p(y \mid \mathbf{x})$, and returning the value $y \in \Omega_Y$, which maximizes the posterior probability (MAP principle). To ensure increased accuracy of posterior estimates, all attributes in the class Markov blanket must be connected directly to the class node or its children, producing a particular model that focuses on supervised classification.

A popular approach is to connect all attributes to the class as in the naive Bayes (NB) algorithm, in which the class is the common parent of all other attributes. This produces a model that assumes conditional independence between the attributes given the class variable. In consequence, this model avoids structural learning, thus significantly improving efficiency. However, while some of the independence assumptions cause no harm (Rish, 2001), others come at the expense of discriminative power, resulting in a highly biased classifier. To overcome this problem the NB assumption is relaxed by allowing some additional relationships in the DAG (Bielza and Larrañaga, 2014). These augmented naive Bayesian classifiers are often preferred over general BNs Friedman et al. (1997) for the sake of simplicity and as better approximations to discriminative posteriors. However, parameter estimation plays an important role and can significantly condition the overall quality of the resulting model Madden (2009). Maximum likelihood estimation from frequency counts can yield zero estimates for unlikely events, which harm the predictive power of the models. As a result, smoothing techniques such as Laplace may be applied to overcome this limitation. Recent proposals of sophisticated estimation techniques Petitjean et al. (2018) demonstrate that BNCs can be overly improved with more accurate parameters, especially in high variance domains. In the rest of the paper, we consider that discrete attributes and smoothed Laplacian estimation are employed to learn the models.

Other techniques such as ensemble learning and adaptive structural learning can also be used to deal with such complex domains. Below, we review the most successful approaches.

2.1 k-dependence BN Classifier (kDB)

In k-dependence estimators (Sahami, 1996) the probability of each attribute value is conditioned by the class and, at most, k other attributes. In kDB, a greedy strategy guided by (conditional) mutual information is used in order to identify the graphical structure of the resulting classifier. First, it computes the mutual information $I(X_i; C)$ between each predictive attribute X_i and the class C. Attributes are sorted and processed by following a decreasing order. When the j-th attribute in the order is added, at most $k + 1$ attributes are set as its parents in the graph: the class C and the k preceding variables with greater conditional mutual information with respect to X_j given the class. Finally, the parameters for the resulting structure are computed, which require an additional pass through the dataset when $k \geq 2$.

Model selection in the kDB framework A better fit of the model to the data can be achieved by tuning k. Thus, the higher the value of k the more complex the model, which helps to decrease the bias but usually increases its variance because of overfitting. However, this could be a disadvantage for small data but not in the case of large data (Martínez et al., 2016), where overfitting is less common. **Selective kDB (SkDB)** (Martínez et al., 2016) extends kDB by carrying out a selection between attribute subsets and values of k in a single additional pass through the dataset.

2.2 Averaged k Dependence Estimators

The AkDE classifier can be modelled as an ensemble classifier consisting of a set of independent BNCs $H = \{h_i(\pi_i), i = 1, \ldots, K\}$ where the number of models K is fixed by constraining the model space to a full family of specific k-dependence classifiers. Specifically, each model presents an augmented naive Bayes structure in which a set of k attributes are fixed as the common parents, denoted by π_i, for the remaining $\{\mathcal{A} \setminus \pi_i\}$ ones in addition to the class. The ensemble consists of the full set of all possible models for a given dataset, while the prediction is obtained as the average. As a particular case, the A1DE classifier obtained by setting $k = 1$, assumes that all attributes depend on the class variable and another common parent attribute, called the super parent (SP). This strategy avoids structural learning making AkDE a true out-of-core learner.

Model selection in the AkDE framework As in kDB, the AkDE framework has the ability to adjust the bias-variance trade-off by varying the k parameter. This allows us to represent from high bias/low variance classifiers, such as NB (A0DE), to lower bias but higher variance classifiers as k increases(Webb et al., 2012). This increases model space and thus the dimensionality of the parameters. Specifically, a total of $K = \binom{d}{k}$ models will be learnt, requiring the induction of K k-dimensional joint frequency tables. The complexity of the algorithm is polynomial given the number of attributes and increases in order with the hyperparameter k. This problem can cause a block even for moderate sized datasets, becoming intractable in the case of high dimensional data, not just due to extensive computational requirements but because of the spatial complexity of the model, quickly scaling to Gigabytes even for moderate datasets and values of k (2, 3, ...).

Model selection is then mandatory to make use of AkDE in practice. Two recent approaches have proven successful when applied to high dimensional and large data domains, surprisingly not only reducing the spatial complexity but also increasing the performance of the resulting ensemble. **Sample Attribute Selective AkDE (SASAkDE)** (Chen et al., 2017a), extends the same notion applied in SkDB to greedily select attributes as suitable super parents from which to learn individual models. The additional pass through the data required to learn the mutual information statistics is alleviated by using only a sample of the training dataset. **Selective AkDE (SAkDE)** (Chen et al., 2017b) is a pure filter approach that bounds the number of models to be included in the ensemble by directly setting a cut point s in the attribute ranking, with the value being calculated according to problem size and computing power. The models are selected by measuring a pondered metric between the mutual information of the super parents against the class and the conditional mutual information of the parents against the children attributes. The latter approach is the most scalable one at this moment, being able to learn subsets of an A3DE ensemble.

As can be observed, the intuition behind both these algorithms is grounded on the assumption that the conditional mutual information of the super parent set of attributes given the class acts as a good approximation for the performance indicator for the resulting sub model.

3. Bias and Variance Decomposition of Classification Error

A large number of studies (Breiman, 1998; Bauer et al., 1999) have analysed the predictive performance of ensemble classifiers by decomposing their error into *bias* and *variance* terms. Different formal definition can be found on the literature, however, the intuition behind these metrics is the same: A biased learning algorithm shows a persistent error when trained on independent samples while a high variance one has a particular fluctuating error for every sample.

Many algorithms show particular capabilities for optimizing their error towards one of the two components. A taxonomy of stable and unstable learners can be established (Breiman, 1998) where the former show low variance at a high risk of being biased, especially if the data is difficult to fit, and the latter provide low bias models by increasing the variance, resulting in classifiers that achieve good average performance. This framework is optimal for majority voting ensembles of classifiers as they have been shown to reduce model variance (Breiman, 1998). Therefore combining them with unstable learners such as decision trees is an ideal scenario that has remained the state of the art in classification models for a long time. In addition, low-bias/high-variance models have been shown to be good candidates in their application for large data problems (Martínez et al., 2016), as the variance impact is softened when the size of the dataset increases.

However, while the interpretation of bias/variance decomposition could seem intuitive, its application to discrete classification is not. This decomposition was originally proposed for quadratic regression in which the prediction is not categorical but belongs to a continuous domain and has a varying degree of error that can be separated into nonnegative terms. As a result, independent regression functions can be averaged to decrease variance without changing bias, while averaging classifiers models can increase the classification error (Schapire et al., 1998).

There are several formulations of the bias/variance decomposition in the literature (Schapire et al., 1998; Webb, 2000), among which we have selected that described in (Bauer et al., 1999; Breiman, 1998) as implemented in (Webb, 2000)[1]. The metrics are drawn from the stability of a given learner $\mathcal{L}$ when trained and tested repeatedly on a number of dataset samples $\mathcal{T}$. We define the central tendency $C^{\circ}_{\mathcal{L}\mathcal{T}}(\mathbf{x})$ as the class with the maximum probability of being selected for a given example $\mathbf{x}$ by all classifiers learnt from $\mathcal{T}$: $C^{\circ}_{\mathcal{L}\mathcal{T}}(\mathbf{x}) = \arg\max_{y} P_{\mathcal{T}}(\mathcal{L}(\mathbf{x}) = y)$

Bias can be measured as the error introduced by the central tendency of the algorithm, in other words, the error of the most frequent classification, and variance as the error introduced by the deviations from this central tendency. These values are usually referred to as contribution of bias and variance to error respectively. To compute them we must first obtain an estimation of the central tendency from our available sample data $\mathcal{D}$ for which a 10x3 fold cross validation is performed, inducing 30 different models $\mathcal{L}(T_k^i)$ and a corresponding test dataset f_k^i for each of the $i \in \{1, \ldots, 10\}$ repeats and $k \in \{1, 2, 3\}$ training folds. A total of 10 independent predictions for each data point $\mathbf{x}$ are obtained and the central tendency $C^{\circ}_{\mathcal{L}\mathcal{T}}(\mathbf{x})$ for this example is then set as the average.

Following this estimation strategy, the central tendency is obtained by the following expression:

$$C^{\circ}_{\mathcal{L}\mathcal{T}}(\mathbf{x}) = \arg\max_{y} P\left(\sum_{i=1}^{10}\sum_{k=1}^{3} 1\left[\mathbf{x} \in f_k^i \wedge \mathcal{L}(T_k^i)(\mathbf{x}) = y\right]\right) \tag{1}$$

The bias and variance contributions to error are then computed for each instance and aggregated over the dataset:

$$\begin{aligned} bias &= P_{(\mathbf{x},y),\mathcal{T}}(\mathcal{L}(\mathcal{T})(\mathbf{x}) \neq y \wedge \mathcal{L}(\mathcal{T})(\mathbf{x}) = C^{\circ}_{\mathcal{L}\mathcal{T}}(\mathbf{x})) \\ variance &= P_{(\mathbf{x},y),\mathcal{T}}(\mathcal{L}(\mathcal{T})(\mathbf{x}) \neq y \wedge \mathcal{L}(\mathcal{T})(\mathbf{x}) \neq C^{\circ}_{\mathcal{L}\mathcal{T}}(\mathbf{x})) \end{aligned}$$

We can easily see that the classification error can be expressed as the sum of the two components, hence the terminology of decomposition or contribution to error for bias and variance.

4. AkDE Under the Ensemble Perspective

Perhaps on of the most popular ensemble classification models is the RF classifier (Breiman, 2001). This combines Bagging, which is learning each model from a bootstrapped sample, and Random Subspaces applied to Decision Trees (DTs) which involves selecting a suboptimal set of nodes for each split. The randomness increases introduced predictably the diversity of the resulting models in the ensemble. This is especially so if we learn fully developed DTs, as suboptimal choices do not have such a large impact on successive levels of the tree. Fully developed DTs are known to be high

[1] This methodology has previously been used to study the initial proposal of the AkDE classifier (Webb et al., 2005, 2012)

variance learners, so by taking the majority vote of an arbitrary number of trees we are correcting the variance error component. In addition, the original paper states that RF has an asymptotic behaviour in terms of bias convergence given a large enough number of trees.

If we compare these models against BNCs, we can find a number of problematic differences between DTs-based ensembles and averaged BNCs such as AkDE. First, our classifiers consider only a finite number of models given k, imposing a boundary on the impact of variance reduction and making it impossible to measure asymptotic convergence or stability properties. A fixed number of models will also reduce diversity and will thus mitigate the effect of the averaging process.

Secondly, the strategy for learning diverse submodels in AkDE is not driven by a randomized suboptimal optimization. Unlike in RF, the individual classifiers follow a fixed constrained structure imposed by the super parent attribute. In general, bias is reduced in a k-dependent BNC as k grows, by inducing higher dimensional multivariate distributions towards the minimum bias. This is achieved theoretically at the irreducible error inherent to the randomness of the dataset. Supposing we had an optimal BNCs H' with the lowest possible bias, algorithms such as KDB or TAN would try to approximate the distribution represented by H' given the available constraints. In contrast, fixed structure models such as naive Bayes or AkDE often add node edges arbitrarily, so one could easily suppose that the underlying distributions would differ from the optimal classifier. The combination of both the NB assumption and the imposed inter-dependencies among the attributes would rarely hold in the data and would likely have a negative effect on inductive bias of the classifier.

Experiments in the literature show that AkDE is a low variance classifier. To the best of our knowledge, however, there is no evidence of this being either a direct result of the averaging process produced by the ensemble or a property of the individual models, as their particular contribution to error has not yet been studied. Several questions about the real effectiveness of AkDE emerge from the previous discussion. In order to answer these, we will study the bias and variance decomposition of error for a range of BNC classifiers compared to RF and DTs.

4.1 Evaluating Bias and Variance through Incremental Samples of Data

One of the best properties of low bias learners is the ability to induce more accurate distributions from larger data samples, making them excellent classifiers in large data domains. A good strategy to compare several models in term of bias is to evaluate them through incremental samples of a large dataset and measure the stability of bias reduction. For this purpose, we conducted an experiment[2] using the well-known *pokerhand* synthetic dataset (Dheeru and Karra, 2017) (8 classes, 10 categorical features). We evaluated the bias and variance decomposition of the error for 20 samples from 50k instances to 1M instances.

The results shown in Figure 1 confirm the expected behaviour of the described models. Regarding trees, RF is the model with the least variance, while, in contrast, a single fully developed DT holds the maximum value for the variance. Moreover, the non-randomized tree is a less biased classifier than the ensemble one, which is composed of noisy trees that do not fit the data perfectly. If we add together bias and variance, the ensemble is a more stable classifier with a lower error rate, especially for smaller samples of data when the variance is harder to reduce.

[2] The experiments in this paper were run on a seven-node Apache Spark cluster with Intel Xeon E5-2609v3 1.90GHz hexacore processors and 64GB of RAM each. The implementation of the algorithms is based on the software package introduced in Arias et al. (2017). More information and source code can be found at `http://github.com/jacintoArias/pgm2018`.

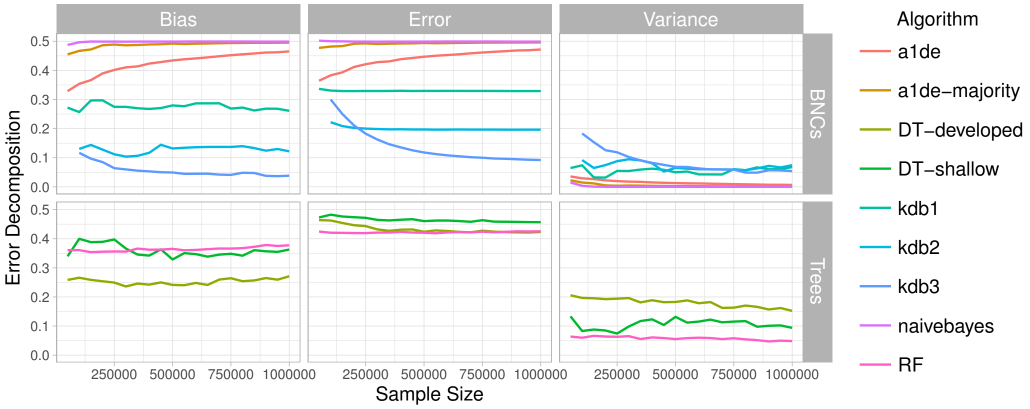


Figure 1: Evolution of bias and variance contribution to error by repeating the experiment with samples of incremental sizes. The plots show variance, error and bias from left to right, for the family of BNC classifiers on top and tree based models on bottom. RF has been configured with 50 trees of 10 nodes depth.

Looking at the BNC models, we can again confirm that highly biased models such as naive Bayes or kDB with $k = 1$ will not improve an increased sample size. However, bias is greatly improved for higher values of k but it is only relevant for large sample sizes, where there is enough data to calibrate a larger number of parameters. Regarding variance, it is clear that the aforementioned simple models are very stable while the complex models again require more data to stabilize.

We might expect A1DE to achieve similar performance to kDB with $k = 1$ or $k = 2$ given its number of parameters and its ability to reduce variance. Surprisingly, A1DE worsen its performance as bias is increased with the sample size, meaning that the central tendency suffers a drift as the sample grows. To delve deeper into this anomaly, we ran a variant of the A1DE classifier by using majority voting among the individual models instead of averaging the probabilities for classification. This is included in Figure 1 as *a1de-majority*, where we can observe that its bias is now stable over the different data samples, although unfortunately at its highest level,. As a conclusion, averaging properties soften the errors of biased classifiers, while majority voting biases the ensemble towards extreme models. In this experiment, increased sample sizes causes a calibration in the probability tables which produces more extreme probability distributions increasing the distance to the correct class in the case of biased models.

4.2 Bias and Variance Contribution of Individual Models

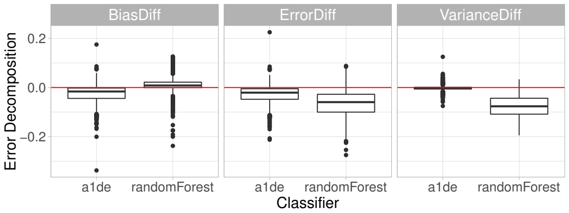
In our second experiment we will evaluate the concept of error decomposition evaluation over a classical benchmark (see Table 1) obtained from the UCI repository Dheeru and Karra (2017) which has been used to evaluate many proposals based on the AkDE classifier. Traditionally, AkDE models have performed well in this domain, without the anomalies detected in the previous experiment. The data shown in Figure 2.(a) corroborates this, showing that in fact, A1DE is the best performing classifier followed by random forest with no significant differences and that all three versions of kDB have obtained worse results. This particular scenario is the opposite of the previous case and can be justified by the small sample size of the majority of the benchmark datasets. As we have seen, kDB requires a moderate amount of data to stabilize and calibrate its parameters.

Database	Cases	Atts.	Classes
Pokerhand	1000000	10	8
Adult	48842	15	2
Chess	28056	6	18
Letter	20000	17	26
Nursery	12960	9	5
PenDigits	10992	17	10
CensusIncome	10419	14	2
Mushrooms	8124	23	2
Musk	6598	168	2
OpticalDigits	5620	49	10
PageBlocks	5473	11	5
Spambase	4601	58	2
Hypothyroid	3772	30	4
Kr.vs.kp	3196	37	2
Splice	3190	62	3
Segment	2310	20	7
Mfeat	2000	6	2
Car	1728	8	4
Contraceptive-mc	1473	10	3
German	1000	21	2
Vowel	990	14	11
Tic-Tac-Toe	958	10	2
Anneal	898	39	6
Vehicle	846	19	4
PimaIndiansDiabetes	768	9	2
BreastCancer-w	699	10	2
BalanceScale	625	5	3
CreditApproval	690	15	2
Cylinder-bands	512	39	2
Haberman	306	3	2
HouseVotes84	435	17	2
HorseColic	368	22	2
Ionosphere	351	35	2
PrimaryTumor	339	18	22
Soybean	307	35	15
Haberman	306	3	2
HeartDisease-c	303	14	2
Audiology	226	70	24
New-Thyroid	215	6	3
Glass-id	214	10	3
Sonar	208	61	2
Autos	205	26	7
Wine	178	14	3
Hepatitis	155	20	2
TeachingAssistant	151	6	3
Iris	150	5	3
Promoters	106	58	2
Zoo	101	17	7
Post-operative	90	9	3
LaborNegotiations	57	17	2
LungCancer	32	57	3

Table 1: Properties of the datasets used in the experiments. Continuous features have been discretized for BNCs by using 4 equal frequency bins.

Algorithm	rank	pvalue	win	tie	loss
A1DE	1.99	-	-	-	-
RF	2.04	**8.7795e-01**	26	0	27
kDB1	2.68	4.9900e-02	37	0	16
kDB2	3.74	3.9885e-08	46	1	6
kDB3	4.56	2.6303e-16	50	0	3

(a) Win, Tie and Loss columns represent the **error** of the best model (A1DE) compared against the rest (e.g. A1DE wins 26 times to RF and loses 27). Ranks are the statistics computed for a Friedman test with the hypothesis of all classifiers being equivalent, p-values were obtained by a post-hoc test with the Holm correction for a 5% significance level where boldfaced values represent the non-rejected hypotheses.



(b) Distribution of differences for the error decomposition metrics between the averaged ensemble and its individual models. The red horizontal line at y=0 helps to discern when the differences are mostly positive or negative.

Figure 2: Results for experiment 2.

Knowing that both ensemble methods are the top performers, we can now look at their error decomposition to check whether their advantage really comes from the properties of averaging independent classifiers. Figure 2.(b) shows the distribution of bias and variance differences between the individual models and the averaged ensembles. We can observe that both ensemble algorithms have a very different relationship with the individual models that conform them. While random forest shows a largely expected variance reduction, in A1DE this reduction is almost non-existent. In fact, the advantage of A1DE seems to result from the bias component, which we can see consistently reduced from the individual models. We can look at these results graphically to fully understand this behaviour, Figure 3 details the bias and variance values for each individual model and the corresponding averaged ensemble. We can observe a more consistent behaviour of random forest, where the individual models are evenly spread in a cloud and the ensemble moves away from this cloud in the variance dimension. In the case of A1DE, the point cloud is less even and the relative position of the ensemble is less consistent both in terms of bias and variance.

This experiment proves that A1DE does not follow the same design premises as other ensemble classifiers, such as random forest or bagging, with which it is usually compared. This might explain why subsets of an AkDE ensemble outperform the full model, which is a harmful and difficult to control property for an averaged classifier as stated in (Schapire et al., 1998).

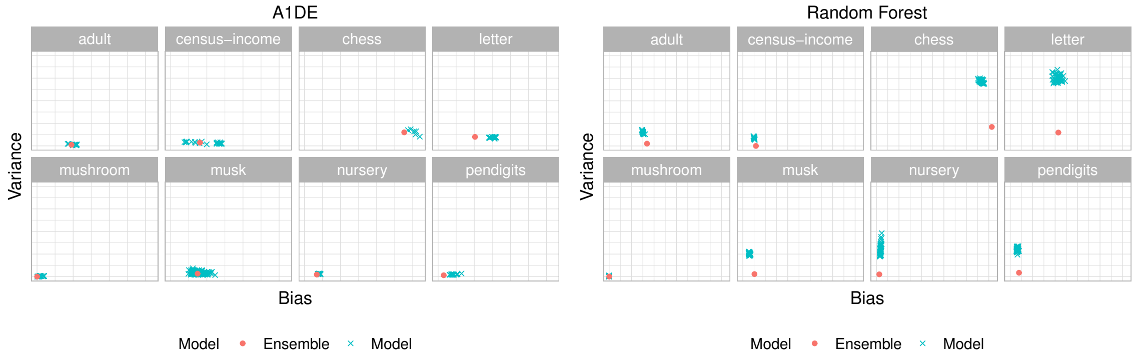


Figure 3: Bias (x) and variance (y) bidimensional distribution for the individual models (blue crosses) and the averaged ensemble (red point).

4.3 On the Effectiveness of Mutual Information for Model Selection

We introduced several extensions to the AkDE framework for model selection. The purpose of such algorithms is twofold: to reduce model space and to improve performance. All of them are based in some way on the hypothesis of using the (conditional) mutual information of attributes given the class as an indicator of classification performance of the (individual) model. However, we have seen that AkDE reduces the bias rather than the variance of the models, and for that reason, the randomness introduced in the models can be harmful in some situations. We should correct only the variance by averaging, as the deviations of individual models do not affect the central tendency of the classification. However, we should avoid intentionally increasing the bias as it affects the averaged classification directly, generating anomalies such as those observed in Figure 1.

This reasoning leads us to a new hypothesis: The best subset of models for AkDE is the one that minimizes the individual bias of the models. We can test this empirically by comparing different model selection approaches, evaluating all possible ensembles from one single model to the full ensemble by adding one model at a time for a range of criteria. Three simple wrapper selection method for AkDE guided by error, bias and variance decomposition will be used along with mutual information and random selection to establish a baseline. Below we show the results for the sum of the error on every stage of the ensemble for all different criteria compared over the previous ML benchmark, where metrics and hypothesis tests are obtained as in Figure 2.(a):

Criterion	Rank	p-value	Win	Tie	Loss
error	1.52	-	-	-	-
bias	1.83	**3.6273e-01**	24	3	17
mi	3.75	7.8353e-11	43	0	1
variance	3.81	3.7139e-11	41	1	2
random	4.09	1.0269e-13	43	0	1

Results show that the best methods are the wrapper ones maximizing the error or its bias component, while variance and surprisingly mutual information are equivalent to randomly selected models. This supports our hypothesis empirically and raises a question about the effectiveness of mutual information for model selection in the BNC ensemble framework.

4.4 Designing Alternative Ensemble Bayesian Network Classifiers

While AkDE obtains good results in practice, it can be inconsistent and unreliable in particular situations, especially since it has few of the desirable properties of ensemble classifiers we have discussed. By selecting subsets of models we can improve the overall performance of the ensemble but experiments have shown that only wrapper approaches are significantly better than random choice, and these are not efficient to compute in a real-world scenario.

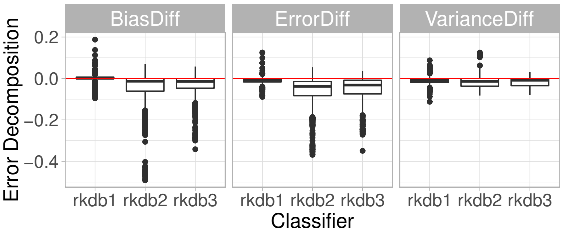
A recent trend is to consider a different approach to define an ensemble of BNCs. Since AkDE requires model selection it no longer benefits from avoiding structural learning, and more complex models such as kDB can be considered for the aggregation. A study of different types of ensembles is presented in Duan and Wang (2017), highlighting the kDF (k-dependence forest) algorithm. The authors define a new ensemble classifier by learning an altered version of a kDB model for each predictive attribute X using a more sophisticated ordering algorithm based on the conditional mutual information. Such an approach outperforms A1DE empirically but still lacks some of the desired properties of an ensemble: It only considers a limited number of models to average and is guided by a finite non-randomized learning algorithm.

Our hypothesis is that a good ensemble model should capture these two properties. Thus we propose a basic ensemble based on kDB that is able to capture them. We define the Random k-dependent classifiers (RkDB) as an ensemble of $h \in [1, \inf)$ independent models learnt by a slightly altered version of the kDB algorithm to introduce diversity. Taking inspiration from the RF strategy of considering only a subset of attributes for each node split, we will consider only a sampled proportion of $\alpha \in [0, 1]$ from the candidate parents available for each attribute when building each model. This strategy adds diversity by controlling randomness but still induces low bias models from the data.

We have conducted preliminary experiments for this new approach, using the same benchmark as before. Figure 4.(a) shows that in fact RkDB with $k = 1$ performs comparably to A1DE and random forest. Unfortunately, however larger values of k suffer from the same problem as the individual kDB models, that is, high bias due to poor probability calibration from small data samples. Interestingly, if we check on the bias and variance decomposition in Figure 4.(b) we can observe that this new ensemble behaves similarly to RF by reducing the variance of the individual models. In addition, it seems to retain some of the good properties of AkDE as it also reduces the bias of the individual models, Figure 5 shows a more uniform point cloud in which the submodels have considerable more variance than the averaged ensemble and the bias is less disperse.

Algorithm	rank	pvalue	win	tie	loss
A1DE	2.48	-	-	-	-
RF	2.49	**9.7929e-01**	26	0	27
RkDB1	2.76	**8.7224e-01**	30	2	21
kDB1	3.58	7.1644e-03	37	0	16
RkDB2	4.75	1.8647e-09	46	0	7
kDB2	4.93	7.4299e-11	46	1	6

(a) Error comparison adding different instances of RKDB. Metrics and hypothesis tests are obstained as in Figure 2.(a).



(b) Distribution of differences for the error decomposition metrics between the averaged ensemble and its individual models.

Figure 4: Results for experiment with RkDB.

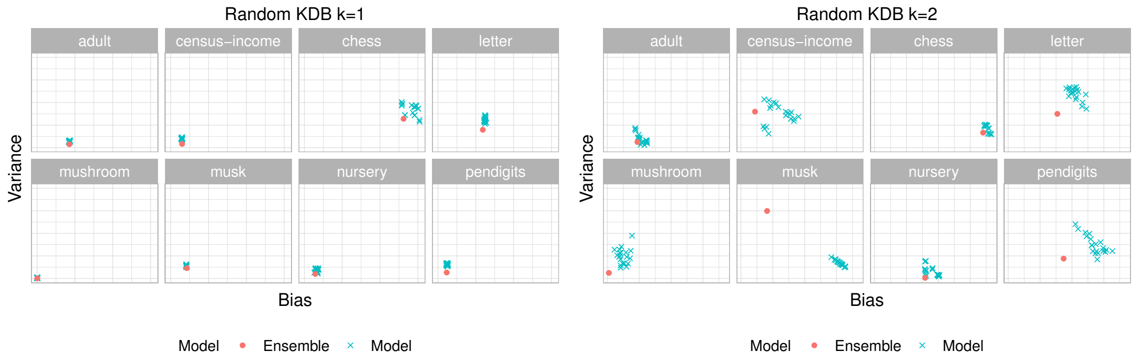


Figure 5: Bias (x) and variance (y) bidimensional distribution for the individual models (blue crosses) and the averaged ensemble (red point).

5. Conclusions

We have conducted a series of experiments that explain a number of previously untested properties of ensemble BNCs. This paper opens new research paths regarding the future of the AkDE framework and BNCs ensemble models in general. We show that averaging models with low variance and fixed dependencies, e.g. A1DE, leads to confusing and non-relevant results in terms of bias and variance. Furthermore, we observe that kDB performs excellently when the data sample is large enough for it to correct and calibrate its parameters.

In future work we will develop and explore the performance of the RkDB algorithm in larger domains where we have seen the kDB algorithm is much superior, aiming for bias and variance correction similar to those achieved in RF. In addition, researching new approaches for better probability estimation and bias reduction in small samples can be combined with these new results to create fully adaptive and high-performance BNCs ensembles.

Acknowledgments

This work has been partially funded by FEDER funds and the Spanish Research Agency (MINECO) through project TIN2016-77902-C3-1-P. Jacinto Arias is also funded by the MECD grant FPU13/00202.

References

J. Arias, J. A. Gamez, and J. M. Puerta. Learning distributed discrete Bayesian Network Classifiers under MapReduce with Apache Spark. *Knowledge-Based Systems*, 117:16 – 26, 2017.

E. Bauer, R. Kohavi, P. Chan, S. Stolfo, and D. Wolpert. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, 36:105–139, 1999.

C. Bielza and P. Larrañaga. Discrete Bayesian Network Classifiers: A Survey. *ACM Comput. Surv.*, 47(1):5:1–5:43, jul 2014.

L. Breiman. Arcing classifiers. *Annals of Statistics*, 26(3):801–849, 1998.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

S. Chen, A. M. Martinez, G. I. Webb, and L. Wang. Sample-Based Attribute Selective AnDE for Large Data. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):172–185, 2017a.

S. Chen, A. M. Martínez, G. I. Webb, and L. Wang. Selective AnDE for large data learning: a low-bias memory constrained approach. *Knowledge and Information Systems*, 50(2):475–503, 2017b.

D. Dheeru and E. Karra. UCI Machine Learning Repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Z. Duan and L. Wang. K-Dependence Bayesian Classifier Ensemble. *Entropy*, 19(12), 2017.

N. Friedman, D. Geiger, and M. Goldszmit. Bayesian Network Classifiers. *Machine Learning*, 29: 131–163, 1997.

M. G. Madden. On the classification performance of tan and general bayesian networks. *Knowledge-Based Systems*, 22(7):489 – 495, 2009. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2008.10.006. Artificial Intelligence 2008.

A. M. Martínez, G. I. Webb, S. Chen, and N. A. Zaidi. Scalable learning of Bayesian network classifiers. *Journal of Machine Learning Research*, 17:1–30, 2016.

X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar. MLlib: Machine Learning in Apache Spark. *Journal of Machine Learning Research*, 17(34):1–7, 2016.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. 2014.

F. Petitjean, W. Buntine, G. Webb, and N. Zaidi. Accurate parameter estimation for Bayesian network classifiers using hierarchical Dirichlet processes. *Machine Learning*, In press:1–29, 2018.

I. Rish. An Empirical Study of the naive Bayes Classifier. In *IJCAI workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, jan 2001.

M. Sahami. Learning Limited Dependence Bayesian Classifiers. In *Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining*, KDD'96, pages 335–338. AAAI Press, 1996.

R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26(5):1651–1686, 1998.

G. I. Webb. MultiBoosting: a technique for combining boosting and wagging. *Machine Learning*, 40(2):159–196, 2000.

G. I. Webb, J. R. Boughton, and Z. Wang. Not so naive Bayes: Aggregating one-dependence estimators. *Machine Learning*, 58(1):5–24, 2005.

G. I. Webb, J. R. Boughton, F. Zheng, K. M. Ting, and H. Salem. Learning by extrapolation from marginal to full-multivariate probability distributions: Decreasingly naive Bayesian classification. *Machine Learning*, 86(2):233–272, oct 2012.