

MiniAnDE: a reduced AnDE ensemble to deal with microarray data

Pablo Torrijos^{1,2}[0000-0002-8395-3848], José A. Gámez^{1,2}[0000-0003-1188-1117],
and José M. Puerta^{1,2}[0000-0002-9164-5191]

¹ Instituto de Investigación en Informática de Albacete (I3A). Universidad de Castilla-La Mancha. Albacete, 02071, Spain.

² Departamento de Sistemas Informáticos. Universidad de Castilla-La Mancha. Albacete, 02071, Spain.

{Pablo.Torrijos, Jose.Gamez, Jose.Puerta}@uclm.es

Abstract. This article focuses on the supervised classification of datasets with a large number of variables and a small number of instances. This is the case, for example, for microarray data sets commonly used in bioinformatics. Complex classifiers that require estimating statistics over many variables are not suitable for this type of data. Probabilistic classifiers with low-order probability tables, e.g. NB and AODE, are good alternatives for dealing with this type of data. AODE usually improves NB in accuracy, but suffers from high spatial complexity since k models, each with $n + 1$ variables, are included in the AODE ensemble. In this paper, we propose MiniAnDE, an algorithm that includes only a small number of heterogeneous base classifiers in the ensemble, i.e., each model only includes a different subset of the k predictive variables. Experimental evaluation shows that using MiniAnDE classifiers on microarray data is feasible and outperforms NB and other ensembles such as bagging and random forest.

Keywords: Bayesian network classifiers · Averaged n -Dependence Estimators · Microarray data · High dimensionality.

1 Introduction

Supervised classification, i.e. predicting the category $c \in \text{dom}(C) = \{c_1, \dots, c_r\}$ for an object \mathbf{x} defined over a set of attributes $\mathbf{X} = \{X_1, \dots, X_k\}$, is one of the most profusely tackled tasks in machine learning. The objective is to learn a classifier $\mathcal{C} : X_1 \times \dots \times X_k \rightarrow C$, from a data set $\mathbf{D} = \{(\mathbf{x}^{(i)}, c^{(i)})\}_{i=1}^m$, such that \mathcal{C} generalises well to new data.

In this paper we focus on a particular niche of supervised classification problems: data defined over a large number of features/attributes and with a scarce number of instances. Such data sets, where $k \gg m$, are common in microarray data problems [1], where the expression level of thousands of genes is analysed simultaneously. Still, due to the cost of obtaining samples, only a few dozen or a few hundred cases are available. This scarcity of cases means that models

that need to estimate complex statistics, e.g. higher-order statistics, or measures subject to a particular context (e.g. a deep branch in a decision tree) cannot be reliably learned. A common solution to combat this curse of dimensionality is to perform a prior feature selection process [5]. However, in this paper we focus on a different solution: using models that, while overall may be complex, only require estimating statistics on a very small number of variables.

The NB classifier [19] is the simplest Bayesian network model used for classification. It is based on the hypothesis (assumption) that all the predictive attributes are independent of each other given the value of the class variable (Figure 1). This independence hypothesis gives rise to the following factorisation:

$$P(c, x_1, \dots, x_k) = P(c) \prod_{i=1}^k P(x_i|c), \quad (1)$$

which enables: (1) NB does not require structural learning; (2) parametric learning is very efficient (a single pass through the BD); and (3) it is only necessary to estimate bi-variate statistics, so a small number of cases is enough.

Among the different improvements made to NB trying to circumvent the independence hypothesis, one of the most outstanding for its exceptional performance is AODE [20]. AODE can be seen as an ensemble formed by n SPODE (Super Parent One Dependence Estimator) classifiers, i.e. a NB extended with one attribute also being the parent of the other features (Figure 2). Thus, in a SPODE each variable depends on another variable apart from the class, which combined with the fact that AODE includes all the n possible SPODEs, allows AODE to consider a large number of possible dependencies between attributes. Despite the strong relaxation of the NB independence assumption that AODE implies, parametric learning is still very efficient and only requires estimating three-variate statistics, so the number of cases needed remains moderate. More dependencies are considered in AnDE [21], where n features play the role of super-parents in each member (SPnDE) of the ensemble. AnDE ($n \geq 2$) can manage more complex dependency relations than AODE (A1DE), however also a greater number of cases is necessary to obtain reliable estimations for $(n+1)$ -ary statistics.

The motivation for this work comes from the fact that when dealing with microarray data, the main problem related to AnDE, even with $n = 1$ (AODE), is the size of the ensemble, which can easily run out of memory. For example, let us consider a problem with $k = 10000$ attributes, each taking 5 different values, as well as the class. In this case A1DE have to store 10000 SPODEs, each one with 10000 probability tables of size 5^3 , which assuming 32bits per float value means 50 GB. Of course, things are worse if we increase n , giving rise to the problem of dealing with *big models* [3].

In this work we propose *MiniAnDE*, an algorithm that tries to build small AnDE models in which only a subset of SPnDEs are included in the ensemble, also limiting to a subset of \mathbf{X} the features included in each SPnDE. To do this, we introduce a structural learning stage in which relevant feature-class and feature-feature relations are identified. In the second stage, SPnDEs are

constructed on the basis of the identified relevant relations. Experiments over nineteen microarray datasets confirm the competitiveness of our approach.

This paper is organized as follows. Section 2 revises the algorithm we took as our baseline, Averaged n-Dependence Estimators [21]. Section 3 introduces the MiniAnDE classifier proposed in this paper. Section 4 presents the experimental evaluation carried out. Finally, Section 5 concludes the paper and outlines potential avenues for future research.

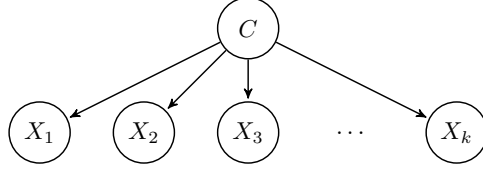


Fig. 1: Graphical structure of NB

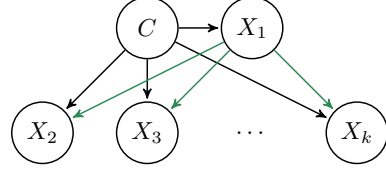


Fig. 2: Graphical structure of SPODE

2 Averaged n -Dependence Estimators (AnDE)

Averaged n-Dependence Estimators (AnDE) [21] extend the AODE (A1DE) algorithm by allowing n super-parent variables in each model (SPnDE). As n grows, the classifier estimates probability distributions of higher dimension, thus reducing its bias but probably increasing its variance, which however will be reduced when all the predictions of the base models are aggregated by the ensemble.

The class label c^* of an instance \mathbf{x} is obtained by:

$$c^* = \arg \max_{c_i \in \text{dom}(C)} P(c_i, \mathbf{x}) = \sum_{S \in \binom{\mathbf{X}}{n}} P(c_i, \mathbf{x}_S) \prod_{X_j \in \mathbf{X} - S} P(x_j | c_i, \mathbf{x}_S), \quad (2)$$

where $\binom{\mathbf{X}}{n}$ represents the subsets of \mathbf{X} having exactly n variables; \mathbf{x}_S is the projection of \mathbf{x} over S ; the expression inside the summation is the factorization of the joint probability carried out by the SPnDE; and the summation stand for the aggregation carried out in the AnDE ensemble.

In particular, for A1DE, the previous expression reduces to:

$$c^* = \arg \max_{c_i \in \text{dom}(C)} P(c_i, \mathbf{x}) = \sum_{l=1}^k P(c_i, \mathbf{x}_l) \prod_{j \neq l} P(x_j | c_i, \mathbf{x}_l). \quad (3)$$

The main problem in AnDE is due to its spatial complexity and the increase in the number of samples needed to make reliable estimates of increasingly larger statistics. Thus, A1DE requires n models, each with $k - 1$ distributions of order 3; A2DE requires $O(n^2)$ models each with $k - 2$ distributions of order 4; A3DE

requires $O(n^3)$ models, each with $k - 3$ distributions of order 5; etc. This means that in practice, AnDE can only be used with $n = 1$ for moderate/large domains and with $n = 2$ for small domains.

In literature, we can find different approaches to make AnDE usable when n and/or k grows. In [15], the A1DE ensemble is replaced by a single model whose super-parent is a latent variable which is estimated by using the EM algorithm. SAnDE [10] and SASAnDE [9] follow a model selection-based approach, which relies on the assumption that the conditional mutual information of the super parent set of attributes given the class is a good approximation of the resulting SPnDE performance. However, the study conducted in [4] over 43 datasets challenges this assumption and the usefulness of using mutual information-based model selection in the AnDE ensemble.

3 MiniAnDE

The main objective of the *MiniAnDE* classifier is to reduce the enormous spatial complexity of AnDE which, in practice, impedes their use in databases with thousands of variables (k) in the case of A1DE and hundreds in the case of A2DE. The aim is to reduce both the number of SPnDEs generated (s) and the number of variables included in each SPnDE (r_i) so that $s \ll k$ and $r_i \ll k$. Thus, we create much smaller and faster models that can handle high-dimensional datasets.

As in [10], we need to select the variable(s) that will act as super-parent(s) and thus give rise to the SPnDEs included in the AnDE model. In addition, we also have to select the *child* features to be included in each SPnDE. Unlike previous work, instead of calculating information-based measures, we propose to use a different machine learning model, a decision tree, from which the relationships between features can be borrowed for our MiniAnDE model.

The use of decision trees (DTs) to select the relevant variables for a classification problem is quite old [8]. From a probabilistic point of view, the subset of variables appearing in the tree could be seen to constitute the Markov blanket of the class variable, i.e. the set of variables that makes the rest irrelevant for classification purposes. Later, ensemble-based methods, in particular random forests, have also been used to obtain the importance of predictive variables in the classification process, using so-called out-of-bag estimation [7]. This technique has become very popular and can be found in almost any ML software, e.g. Scikit-Learn.

In this paper we propose to use an ensemble of DTs to identify the SPnDEs to be included in our MiniAnDE model. In addition to the ability of the DTs to select the relevant variables for the class, we will also exploit the location in which these variables are placed in the tree. Thus, it is well known that one of the advantages of DTs is their context-based analysis of the data, where by context we mean a (partial) branch of the tree. Therefore, we traverse the tree to identify all paths of length n and create an SPnDE for each of them by setting the variables in the path as super-parents. Then, all variables in the tree that are

adjacent to the super-parent variables are included as children in that SPnDE. To obtain a more robust MiniAnDE model we consider a set of diverse DTs, that is, an ensemble.

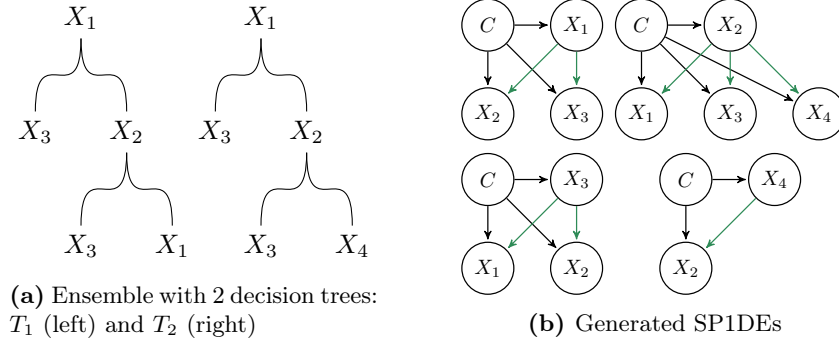
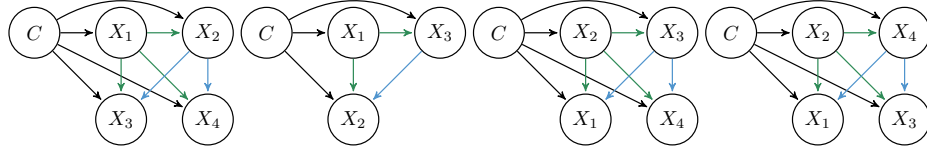
Algorithm 1 MiniAnDE

Require: Dataset \mathbf{D} defined over $\mathbf{X} \cup \{C\}$; n ; t

- 1: $SP \leftarrow \emptyset$
- 2: $\mathcal{T} \leftarrow \emptyset$
- 3: **for** $i \leftarrow 1$ to t **do**
- 4: $T \leftarrow$ learn a DT from a sample of \mathbf{D}
- 5: $\mathcal{T} \leftarrow \mathcal{T} \cup \{T\}$
- 6: $SP^t \leftarrow$ {sets of n consecutive variables in T }
- 7: $SP \leftarrow SP \cup SP^t$
- 8: **end for**
- 9: $\forall sp \in SP, \text{children}(sp) \leftarrow \bigcup_{T \in \mathcal{T} \wedge sp \in T} \left\{ \bigcup_{X \in sp} \text{adjacent}(X, T) \right\}$
- 10: $\mathcal{M} \leftarrow \emptyset$
- 11: **for each** $sp \in SP$ **do**
- 12: Create an SPnDE m with sp as super-parent and $\text{children}(sp)$ as features
- 13: $\mathcal{M} \leftarrow \mathcal{M} \cup \{m\}$
- 14: **end for**
- 15: **return** \mathcal{M}

Algorithm 1 provides a scheme of the previous idea. Let us illustrate its working process with an example taking $n = 1$ and $t = 2$. Let us also assume that Figure 3a shows two DTs learnt from two different samples of \mathbf{D} . The algorithm starts with T_1 and identify $SP^1 = \{\{X_1\}, \{X_2\}, \{X_3\}\}$. Now $SP \leftarrow SP^1$ and T_2 is considered. The algorithm computes $SP^2 = \{\{X_1\}, \{X_2\}, \{X_3\}, \{X_4\}\}$, and so $SP = \{\{X_1\}, \{X_2\}, \{X_3\}, \{X_4\}\}$. Next, children sets are computed as: $\text{children}(\{X_1\}) = \{X_2, X_3\}$, $\text{children}(\{X_2\}) = \{X_1, X_3, X_4\}$, $\text{children}(\{X_3\}) = \{X_1, X_2\}$ and $\text{children}(\{X_4\}) = \{X_2\}$. Therefore, the SP1DEs included in the resulting MiniA1DE are those shown in Figure 3b. If the same process is applied with $n = 2$, $SP^1 = \{\{X_1, X_2\}, \{X_1, X_3\}, \{\{X_2, X_3\}\}\}$, $SP^2 = \{\{X_1, X_2\}, \{X_1, X_3\}, \{\{X_2, X_3\}, \{\{X_2, X_4\}\}\}$ and $SP = \{\{X_1, X_2\}, \{X_1, X_3\}, \{X_2, X_3\}, \{X_2, X_4\}\}$. Next, children sets are computed as: $\text{children}(\{X_1, X_2\}) = \{X_3, X_4\}$, $\text{children}(\{X_1, X_3\}) = \{X_2\}$, $\text{children}(\{X_2, X_3\}) = \{X_1, X_4\}$ and $\text{children}(\{X_2, X_4\}) = \{X_1, X_3\}$. Figure 4 shows the resulting MiniA2DE.

Like the original AnDE algorithm, MiniAnDE only works with discrete variables, so if numerical predictive attributes are included in the dataset, they must first be discretized. Once the SPnDEs have been determined, only parametric learning is required, which can be performed in a single pass through the dataset. Therefore, the complexity of learning a MiniAnDE model is dominated by the learning process of the set of decision trees. In this sense, it is worth noting that due to the small number of instances in the microarray data, the obtained tree will be shallow, which coupled with the use of only discrete (discretized) variables, results in a fast learning process. On the other hand, inference is also

**Fig. 3:** MiniA1DE obtained from the ensemble $\{T_1, T_2\}$ **Fig. 4:** MiniA2DE obtained from the ensemble $\{T_1, T_2\}$ in Fig. 3a

faster than in the original AnDE models, since only a few SPnDEs are aggregated instead of k .

The MiniAnDE algorithm can be instantiated with any decision tree and ensemble learning algorithm, e.g. bagging [6] or random forest [7]. This fact together with the own DT/ensemble learning hyperparameters (pruning or no-pruning, max depth, number of trees, etc.) provides a wide range of combinations to generate the MiniAnDE classifier, making possible to fine-tuning it for a given dataset.

To conclude this section, we present a possible extension of the MiniAnDE algorithm. As with AnDE, MiniAnDE is expected to be a better estimator than NB for posterior class labels probabilities. However, in some cases it is possible that some attribute configurations and class values may be missing or underrepresented in the learning dataset, resulting in a nearly uniform posterior probability distribution for the class given the input instance. To alleviate this drawback, we produce the output as a convex combination of MiniAnDE and NB, adding it to the ensemble according to a parameter $\alpha \in [0, 1]$: $p(c|\mathbf{x}) = \alpha p_{NB}(c|\mathbf{x}) + (1 - \alpha) p_{MiniAnDE}(c|\mathbf{x})$. We compare the MiniAnDE algorithm with $\alpha = 0$ and $\alpha \neq 0$ in the experiments performed in Section 4.

4 Experimental evaluation

In the next sections we describe the datasets utilized, the algorithms evaluated, the methodology employed, and analyze the results obtained.

4.1 Data sets

Table 1 describes the 19 microarray data sets used to evaluate the proposed algorithms, commonly used in the literature [1,5,12,22].

Table 1: Data sets used in the experimental evaluation. I is the number of instances, N the number of predictable variables and K the number of classes.

DATA SET	FEATURES			DATA SET	FEATURES		
	I	N	K		I	N	K
9 TUMORS	60	5 726	9	LUNG	203	12 600	5
11 TUMORS	174	12 533	11	LYMPHOMA 3	66	4 026	3
BREAST	97	24 481	2	LYMPHOMA 9	96	4 026	9
CNS	60	7 130	2	LYMPHOMA 11	96	4 026	11
COLON	62	2 000	2	MLL	72	12 582	3
DLBCL	77	5 469	2	OVARIAN	253	15 154	2
GLI	85	22 283	2	PROSTATE	102	12 600	2
LEUKEMIA	72	7 129	2	SMK	187	19 993	2
LEUKEMIA 3	72	7 129	3	SRBCT	83	2 308	4
LEUKEMIA 4	72	7 129	4				

4.2 Reproducibility

The entire MiniAnDE algorithm’s family has been programmed from scratch using Java (OpenJDK 8) and the library WEKA 3.9.6 ³. All experiments were conducted on machines running the CentOS 7 operating system with an Intel Xeon E5-2650 8-Core Processor limited to 8 threads and 32 GB of RAM per execution.

To reproduce the experiments, all of the code and the execution scripts are provided at GitHub ⁴. Regarding the data, for convenience, we provide in OpenML ⁵ a common source repository for the 19 datasets, with reference to the original articles.

4.3 Algorithms

In this study, the following algorithms have been evaluated:

- The MiniAnDE algorithm introduced in Section 3, with $n = 1$ and $n = 2$. The following parameters have been fine-tuned by using grid-search for each dataset:

³ <https://www.cs.waikato.ac.nz/ml/weka/>

⁴ <https://github.com/ptorrijos99/mAnDE>

⁵ https://www.openml.org/search?type=data&uploader_id=%3D_33148

- Bagging is considered to generate the ensemble of trees used to learn the structure of those SPnDEs included in the MiniAnDE model. The number of trees is taken from the set $\{50, 100, 150, 200\}$.
 - The weight of NB is chosen from the set $\alpha = \{0.02, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4\}$. The case of $\alpha = 0$ is always reported, as it corresponds to the canonical MiniAnDE as introduced in Algorithm 1.
- The Naive Bayes algorithm [19].
 - The Bagging ensemble algorithm [6]. The number of trees (50, 100, 150 and 200) is selected for each dataset by using grid-search.
 - The Random Forest algorithm [7]. Default value \sqrt{k} is used to select the random subset of variables evaluated at each split. The number of trees (50, 100, 150 and 200) is selected for each dataset by using grid-search.

Please, note that original AnDE algorithm [21] is not included because of its spatial complexity. In fact, under the resources described in previous section, A1DE algorithm only can cope with 1 out of the 19 datasets (`colon`), obtaining an accuracy of 80.64.

4.4 Methodology

We have taken the following design decisions:

- Each algorithm has been evaluated employing a double cross-validation. Leave-one-out cross-validation has been used for external validation, and stratified 5-fold cross-validation has been used for the internal validation in which the best hyperparameter(s) value(s) are selected by using grid-search. This approach ensured that the results were robust and not influenced by the specific partitioning of the data, especially given the small number of instances in microarray data.
- Numerical variables are discretized. Discretization intervals are learn from the training partition and then applied over the validation/test one. We used the following procedure: (1) supervised entropy-based discretization following Fayyad and Irani algorithm [13] was applied; and (2) those variables left in a single interval are then discretized into 2 intervals (bins) by using unsupervised equal frequency. Note that variables discretized in a single bin by Fayyad and Irani algorithm are those *marginally* independent to the class, but can be relevant to the class when used in conjunction with other attributes (e.g. as in an X-OR dataset).
- The study’s results have been analyzed using the methodology specified in [11,17], and the analysis has been conducted using the `exreport` R package [2]. The analysis begins by performing a Friedman test [16] with the null hypothesis that all algorithms have equal performance. If the null hypothesis is rejected, a posthoc test using Holm’s procedure [18] is carried out to

compare all algorithms against the one ranked first by the Friedman test. Both assessments are conducted at a significance level of 5%.

4.5 Results

The summary of the accuracy results is shown in Table 2, including the result of each algorithm⁶ for each database as well as the total average of each algorithm. The algorithm(s) with the highest accuracy are highlighted in bold. In accordance with the procedure described in Section 4.4, we analyzed the results of our experiments. We found evidence to reject the null hypothesis of equal performance across all algorithms with a computed p-value of 1.490×10^{-2} . The detailed results of the posthoc test are presented in Table 3, which shows the ranking generated by the Friedman test and the p-value adjusted using Holm’s procedure (non-rejected null hypotheses are boldfaced), along with the number of wins, ties, and losses for each algorithm versus the algorithm that ranked first. Based on the statistical analysis, we draw the following conclusions:

Table 2: Accuracy of each algorithm.

DATA SET	ALGORITHM						
	MA1DE	MA2DE	MA1DE $\alpha > 0$	MA2DE $\alpha > 0$	NB	BAGGING	RF
11 TUMORS	83.91	85.06	89.66	88.51	84.48	87.36	85.06
9 TUMORS	33.33	35.00	50.00	48.33	53.33	36.67	36.67
BREAST	67.01	67.01	64.95	68.04	69.07	67.01	62.89
CNS	60.00	65.00	63.33	71.67	60.00	73.33	65.00
COLON	85.48	87.10	87.10	87.10	87.10	85.48	87.10
DLBCL	89.61	84.42	84.42	81.82	80.52	87.01	88.31
GLI	87.06	85.88	85.88	84.71	82.35	85.88	85.88
LEUKEMIA	95.83	95.83	97.22	94.44	87.50	91.67	94.44
LEUKEMIA 3	94.44	94.44	95.83	94.44	83.33	94.44	87.50
LEUKEMIA 4	91.67	90.28	90.28	90.28	79.17	88.89	77.78
LUNG	90.64	91.13	92.61	94.09	72.91	96.55	89.16
LYMPHOMA 11	77.08	81.25	90.62	91.67	91.67	81.25	84.38
LYMPHOMA 3	95.45	93.94	98.48	98.48	100.00	93.94	93.94
LYMPHOMA 9	78.12	76.04	89.58	91.67	95.83	81.25	81.25
MLL	94.44	95.83	97.22	95.83	90.28	93.06	94.44
OVARIAN	97.63	98.42	97.63	98.02	92.49	98.02	95.26
PROSTATE	93.14	91.18	91.18	88.24	65.69	91.18	86.27
SMK	70.05	70.05	71.66	71.66	65.24	70.59	65.24
SRBCT	98.80	97.59	98.80	97.59	92.77	95.18	100.00
MEAN	83.35	83.44	86.13	86.14	80.72	84.15	82.14

⁶ mANDE denotes the canonical MiniAnDE algorithm ($\alpha = 0$) and mANDE $\alpha > 0$ denotes its combination with NB using $\alpha > 0$, the parameter α is set using a grid search and CV, as noted above.

Table 3: Post-hoc test results for the accuracy of each algorithm.

ALGORITHM	P-VALUE	RANK	WIN	TIE	LOSS
MINIA1DE ($\alpha > 0$)	-	2.84	-	-	-
MINIA2DE ($\alpha > 0$)	7.073×10^{-1}	3.11	9	4	6
MINIA2DE ($\alpha = 0$)	2.419×10^{-1}	4.00	11	5	3
BAGGING	2.419×10^{-1}	4.08	12	2	5
MINIA1DE ($\alpha = 0$)	2.419×10^{-1}	4.16	12	2	5
RANDOM FOREST	3.432×10^{-2}	4.74	14	2	3
NAIVE BAYES	8.492×10^{-3}	5.08	13	1	5

- The MiniA1DE algorithm with $\alpha > 0$ is ranked in the first place, although there is no significant difference (confidence level 0.05) with respect to the other three MiniAnDE algorithms and bagging. A significant difference is observed with respect to NB and random forest.
- Both MiniAnDE algorithms with $\alpha > 0$ rank ahead, although without significant difference among them, of their counterpart canonical versions without incorporating NB. This corroborated the fact that in some cases, due to the small sample size in microarray datasets, it is good to incorporate the prediction of a simple low-bias classifier.
- Regarding the use of $n = 1$ or $n = 2$, there do not seem to be major differences in either MiniAnDE or MiniAnDE-NB, with either option working better depending on the data set, resulting in an almost identical average accuracy.
- NB is ranked in the last position, which is not unexpected due to the fact that it is by far the simpler model tried. However, it is interesting to observe the bad results obtained by RF, which is ranked behind bagging. It seems that the use of pseudorandom DTs does not match with the large number of variables and small data size of microarray data.

As for computational efficiency, the CPU time is shown in Table 4. As expected, NB is the fastest algorithm (linear in the number of variables and instances). On the other hand, the MiniAnDE algorithms require an affordable amount of CPU time, almost identical to bagging, the classifier it uses to train the trees. Furthermore, the effect of using MiniAnDE with $\alpha > 0$ is practically insignificant. In general, we can say that the MiniAnDE approach is the best choice among the tested hypotheses when dealing with microarray data.

5 Conclusions

A new algorithm for learning AnDE-like classifiers has been proposed. The method is tailored to the special case of microarray data, where few data instances are available but the number of variables is so large (thousands) that standard AnDE classifiers do not fit in memory. The proposed algorithm incorporates a structural learning stage, which based on the use of shallow decision

Table 4: Execution time per L.O.O. iteration (seconds) of each algorithm.

DATA SET	ALGORITHM						
	MA1DE	MA2DE	MA1DE $\alpha > 0$	MA2DE $\alpha > 0$	NB	BAGGING	RF
11 TUMORS	4.05	4.94	5.31	5.23	0.83	4.50	1.50
9 TUMORS	0.96	0.92	0.95	1.05	0.20	0.95	0.37
BREAST	3.23	3.27	3.49	3.58	0.83	3.44	1.41
CNS	0.59	0.66	0.66	0.68	0.18	0.61	0.42
COLON	0.72	0.74	0.74	0.77	0.18	0.66	0.18
DLBCL	0.40	0.43	0.54	0.42	0.14	0.37	0.28
GLI	2.08	2.07	1.94	2.12	0.66	1.58	1.03
LEUKEMIA	0.46	0.48	0.44	0.45	0.19	0.47	0.34
LEUKEMIA 3	0.60	0.53	0.52	0.52	0.17	0.53	0.37
LEUKEMIA 4	0.73	0.60	0.69	0.66	0.19	0.57	0.36
LUNG	3.69	3.81	3.80	4.03	0.95	4.00	1.36
LYMPHOMA 11	0.96	1.06	1.02	0.90	0.18	0.87	0.42
LYMPHOMA 3	0.37	0.35	0.32	0.33	0.12	0.28	0.24
LYMPHOMA 9	0.81	0.86	0.88	0.86	0.21	0.80	0.29
MLL	0.92	0.81	0.89	0.85	0.29	0.70	0.55
OVARIAN	3.00	3.03	3.07	2.98	1.23	2.94	1.50
PROSTATE	1.38	1.27	1.38	1.46	0.41	1.37	0.71
SMK	7.42	8.98	7.69	7.98	1.22	7.46	2.07
SRBCT	0.28	0.29	0.29	0.29	0.09	0.30	0.18
MEAN	1.72	1.85	1.82	1.85	0.44	1.70	0.72

trees, allows the selection of a few SPnDEs in the resulting MiniAnDE ensemble. Furthermore, a small subset of variables is included in each SPnDE, leading to a very light model regarding spatial needs and providing fast inference. The experiments' results over 19 microarray datasets show the competitiveness of our proposal regarding decision tree-based ensembles, both in accuracy and efficiency.

As future works, we plan to study our proposal without the need of discretizing numerical variables, by considering AnDE models based on the use of conditional Gaussian networks [14].

Acknowledgements This work has been funded by the Government of Castilla-La Mancha and “ERDF A way of making Europe” under project SBPLY/21/180225/000062. It is also partially funded by MCIN/AEI/10.13039/501100011033 and “ESF Investing your future” through the projects PID2019-106758GB-C33 and FPU21/01074.

This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this contribution is published in Communications in Computer and Information Science, vol 1826, and is available online at https://doi.org/10.1007/978-3-031-34204-2_12.

References

1. Abd-Elnaby, M., Alfonse, M., Roushdy, M.: Classification of breast cancer using microarray gene expression data: A survey. *Journal of Biomedical Informatics* **117**, 103764 (May 2021)
2. Arias, J., Cozar, J.: *exreport: Fast, Reliable and Elegant Reproducible Research* (2016), <https://CRAN.R-project.org/package=exreport>, R package version 0.4.1
3. Arias, J., Gámez, J.A., Puerta, J.M.: Learning distributed discrete bayesian network classifiers under mapreduce with apache spark. *Knowl. Based Syst.* **117**, 16–26 (2017)
4. Arias, J., Gámez, J.A., Puerta, J.M.: Bayesian network classifiers under the ensemble perspective. In: Studený, M., Kratochvíl, V. (eds.) *International Conference on Probabilistic Graphical Models, PGM 2018*, 11–14 September 2018, Prague, Czech Republic. *Proceedings of Machine Learning Research*, vol. 72, pp. 1–12 (2018)
5. Bolón-Canedo, V., Sánchez-Marño, N., Alonso-Betanzos, A., Benítez, J., Herrera, F.: A review of microarray datasets and applied feature selection methods. *Information Sciences* **282**, 111–135 (Oct 2014)
6. Breiman, L.: Bagging predictors. *Machine Learning* **24**(2), 123–140 (Aug 1996)
7. Breiman, L.: Random forests. *Machine Learning* **45**(1), 5–32 (2001)
8. Cardie, C.: Using decision trees to improve case-based learning. In: *Proceedings of the Tenth International Conference on Machine Learning (ICML-93)*. p. 25–32 (1993)
9. Chen, S., Martínez, A.M., Webb, G.I., Wang, L.: Sample-based attribute selective AnDE for large data. *IEEE Trans. Knowl. Data Eng.* **29**(1), 172–185 (2017)
10. Chen, S., Martínez, A.M., Webb, G.I., Wang, L.: Selective AnDE for large data learning: a low-bias memory constrained approach. *Knowl. Inf. Syst.* **50**(2), 475–503 (2017)
11. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* **7**, 1–30 (01 2006)
12. Díaz-Uriarte, R., de Andrés, S.A.: Gene selection and classification of microarray data using random forest. *BMC Bioinformatics* **7**(1) (Jan 2006)
13. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: *International Joint Conference on Artificial Intelligence* (1993)
14. Flores, M.J., Gámez, J.A., Martínez, A.M., Puerta, J.M.: GAODE and HAODE: two proposals based on AODE to deal with continuous variables. In: *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009*, Montreal, Quebec, Canada, June 14–18, 2009. vol. 382, pp. 313–320. ACM (2009)
15. Flores, M.J., Gámez, J.A., Martínez, A.M., Puerta, J.M.: HODE: hidden one-dependence estimator. In: *ECSQARU-2009. Lecture Notes in Computer Science*, vol. 5590, pp. 481–492. Springer (2009)
16. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics* **11**(1), 86–92 (Mar 1940)
17. García, S., Herrera, F.: An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research* **9**, 2677–2694 (2008)
18. Holm, S.: A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* **6**, 65–70 (1979)
19. Webb, G.I.: Naïve Bayes. In: *Encyclopedia of Machine Learning*, pp. 713–714. Springer Science+Business Media (2010)

20. Webb, G.I., Boughton, J.R., Wang, Z.: Not so naive bayes: Aggregating one-dependence estimators. *Machine Learning* **58**(1), 5–24 (Jan 2005)
21. Webb, G.I., Boughton, J.R., Zheng, F., Ting, K.M., Salem, H.: Learning by extrapolation from marginal to full-multivariate probability distributions: decreasingly naive bayesian classification. *Machine Learning* **86**(2), 233–272 (Oct 2011)
22. Zhu, Z., Ong, Y.S., Dash, M.: Markov blanket-embedded genetic algorithm for gene selection. *Pattern Recognition* **40**(11), 3236–3248 (Nov 2007)