ELSEVIER Artificial Intelligence in Medicine 22 (2001) 233–248

Artificial Intelligence in Medicine

www.elsevier.com/locate/artmed

Using Bayesian networks in the construction of a bi-level multi-classifier. A case study using intensive care unit patients data

Basilio Sierra[a,*], Nicolás Serrano[b], Pedro Larrañaga[a], Eliseo J. Plasencia[b], Iñaki Inza[a], Juan José Jiménez[b], Pedro Revuelta[b], María Luisa Mora[b]

[a]*Department of Computer Science and Artificial Intelligence, University of the Basque Country, P.O. Box 649, E-20080 San Sebastián, Spain*

[b]*Intensive Care Unit at Canary Islands University Hospital, 38320 La Laguna, Tenerife, Canary Islands, Spain*

Received 17 April 2000; received in revised form 7 August 2000; accepted 25 September 2000

Abstract

Combining the predictions of a set of classifiers has shown to be an effective way to create composite classifiers that are more accurate than any of the component classifiers. There are many methods for combining the predictions given by component classifiers. We introduce a new method that combine a number of component classifiers using a Bayesian network as a classifier system given the component classifiers predictions. Component classifiers are standard *machine learning* classification algorithms, and the Bayesian network structure is learned using a *genetic algorithm* that searches for the structure that maximises the classification accuracy given the predictions of the component classifiers. Experimental results have been obtained on a datafile of cases containing information about ICU patients at Canary Islands University Hospital. The accuracy obtained using the presented new approach statistically improve those obtained using standard *machine learning* methods. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Supervised classification; Machine learning; Stacked generalization; Bayesian networks; Genetic algorithms; 10-Fold cross-validation

1. Introduction

During the past several years, in a variety of application domains, researchers in *machine learning*, computational learning theory, pattern recognition and statistics have re-ignited

[*] Corresponding author. Tel.: +34-943-015102; fax: +34-943-219306; URL: http://www.sc.ehu.es/isg.
E-mail address: ccpsiarb@si.ehu.es (B. Sierra).

PII: S0933-3657(00)00111-1

the effort to learn how to create and combine an ensemble of classifiers. This research has the potential to apply accurate composite classifiers to real world problems by intelligently combining known learning algorithms.

Classifier combination falls within the *supervised learning* paradigm. This task orientation assumes that we have been given a set of training examples, which are customarily represented by feature vectors. Each training example is labelled with a class target, which is a member of a finite, and usually small set of class labels. The goal of supervised learning is to predict the class labels of examples that have not been seen.

Combining the predictions of a set of component classifiers has shown to yield accuracy higher than the most accurate component on a long variety of supervised classification problems [14,16].

In this paper, we present a new multi-classifier construction methodology based on the well-known stacked generalization paradigm [44], in which a number of classifier layers are designed to be part of a global multi-classifier, where the upper layer classifiers receive the class predicted by its immediately previous layer as input.

The data used in this study has been obtained at a 20-bed general, medical, surgical, and trauma ICU of the Canary Islands University Hospital, a tertiary referral university hospital in the Canary Islands (Spain). There is information about 1210 ICU patients, and each patient record has the values given by medical standard methods such as the Acute Physiology and Chronic Health Evaluation II (APACHE II, [19]), Mortality Probability Model II (MPM II, [26]), and Simplified Acute Physiology Score II (SAPS II, [25]), as well as some routine information recorded for patients (sex, age, etc.). In the datafile used, the class corresponds to Survival (996 cases, 82.31%) and not survival (214 cases, 17.69%) information.

We have designed a two-layer classification system in which we use a set of standard *machine learning* algorithms as layer-0 single classifiers, and we induce, over predictions made, a Bayesian network structure that acts as consensed voting system at layer-1. Once the multi-classifier is constructed, and given a new case to be classified, we run every single classifier with the new case as input, and take the prediction as the corresponding Bayesian network node instantiation. The second step is to propagate the evidence in the obtained Bayesian network and select the class with the highest a posteriori probability as the multi-classifier predicted class. Empirical results show that this multi-classifier outperforms each of the single classifiers used.

The rest of the paper is organised as follows. The multi-classifier schemata and the process of its construction is shown in Section 2. Section 3 shows the level-0 single classifiers used in the experimentation. The level-1 classifier, as well as the methodology used in its construction, is introduced in Section 4. Section 5 presents the experimental results obtained applying the previous methodology to a database of cases containing information about ICU patients. Section 6 presents the conclusions (Fig. 1).

2. Multi-classifier schemata

Primarily, we present a new method for the construction of a multi-classifier based on a straightforward approach that has been termed *stacked generalization* [44]. In its most basic form, a layered architecture consists of a set of *component classifiers* that forms the first layer. Wolpert calls the component classifiers the *level*-0 *classifiers* and the combining classifier, the *level*-1 *classifier*. See Fig. 1 for the schemata of our stacked generalization classifier.

2.1. Multi-classifier structure

Stacked generalization is a framework for classifier combination in which each layer of classifiers is used to combine the predictions of the classifiers of its preceding layer. A single classifier at the top-most level outputs the ultimate prediction. In our approach, we use a two-level system that has a Bayesian network as this single classifier. The choice is based on the idea that we can assume we are making a *consensus vote system* over the predictions of the level-0 single classifiers: assuming it to be a good idea to take the possible relations existing in the predictions given by each level-0 model into consideration. Therefore, we induce from the datafile obtained with the *machine learning* classifier predictions, a Bayesian network which tries to identify the possible conditional independencies and dependencies existing between the results obtained by these level-0 classifiers. This Bayesian network is used to perform the last classification step.

2.2. Multi-classifier construction

We present, at this point, the methodology used in the construction of the multi-classifier.

For each single classifier, we have learned the model using MLC++ library [21]. In order to obtain a training datafile to be used for the Bayesian network structure learning process, we execute a *Leaving One Out* sequence for each single classifier in which, for each case i in the database we learn the model using the remaining $n - 1$ cases (all the cases except the ith), and test the learned model with the ith case, obtaining the class predicted for this case by the single classifier. Obtained results in this step are used as training set in the Bayesian network construction. In Fig. 2 the construction process is showed.

3. Layer-0 composite classifiers

As established by Skalak [41], the major objective in the selection of the composite classifiers is to obtain diversity in the predictions given by each individual classifier, as well as good accuracy levels. Consequently, we therefore use a variety of standard *machine learning* algorithms.

In the supervised learning task, in the training database used to induce the classification model, we know for each x sample its y label value. Starting from this form of database, we will briefly describe the single paradigms we will use in our experiments. These paradigms come from the world of the artificial intelligence and they are grouped in the family of *machine learning* (ML) paradigms.

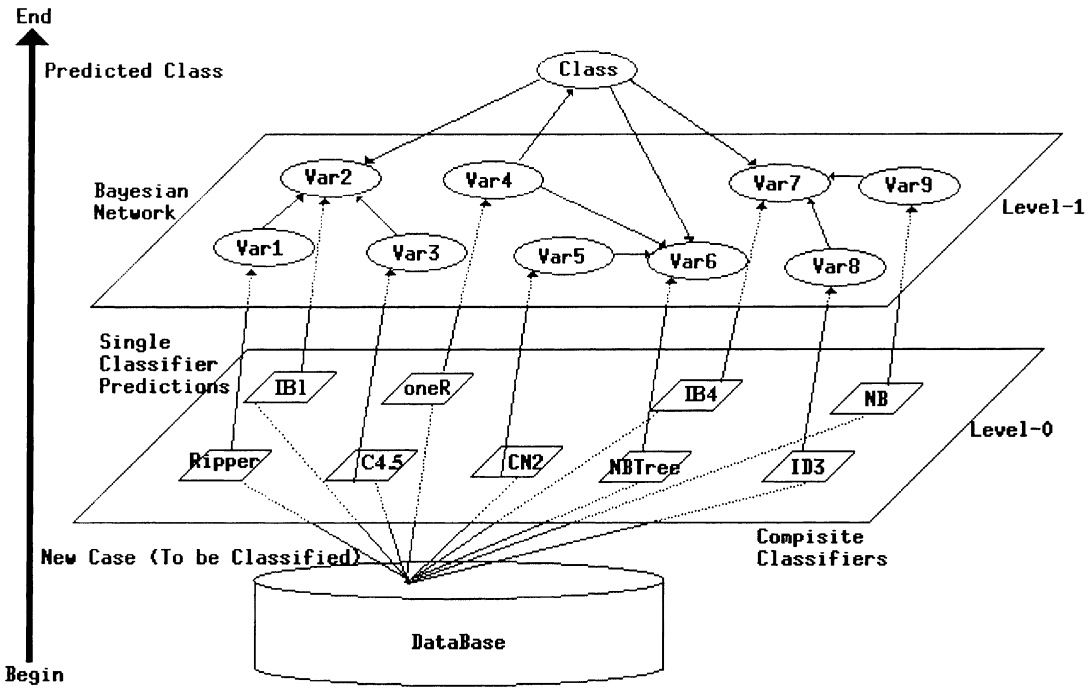

Fig. 1. Multi-classifier schemata.

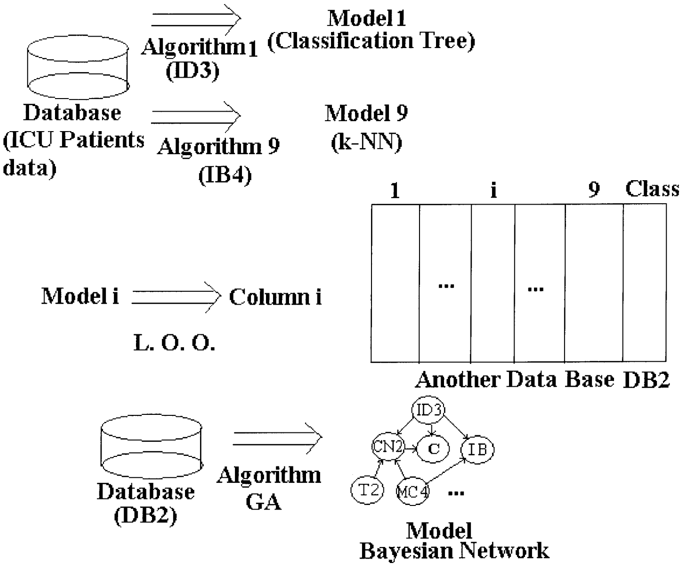

Fig. 2. Multi-classifier constraction.

3.1. Decision trees

A *decision tree* consists of nodes and branches to partition a set of samples into a set of covering decision rules. In each node, a single test or decision is made to obtain a partition. The starting node is usually referred as the root node. An illustration of this appears in Fig. 3. In the terminal nodes or leaves a decision is made on the class assignment.

In each node, the main task is to select an attribute that makes the best partition between the classes of the samples in the training set. There are many different measures to select the best attribute in a node of the decision trees: two works gathering these measures are [27,29]. In more complex works like [30] these tests are made applying the linear discriminant approach in each node. In the induction of a decision tree, an usual problem is the overfitting of the tree to the training dataset, producing an excessive expansion of the

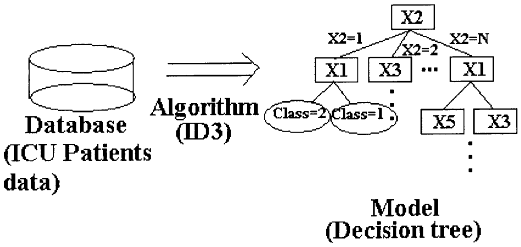

Fig. 3. Single classifier construction.

tree and consequently losing predictive accuracy to classify new unseen cases. This problem is overcome in two ways:

- Weighing the discriminant capability of the attribute selected, and thus discarding a possible successive splitting of the dataset. This technique is known as ‘‘prepruning’’.
- After allowing a huge expansion of the tree, we could revise a splitting mode in a node removing branches and leaves, and only maintaining the node. This technique is known as ‘‘postpruning’’.

The works that have inspired a lot of successive papers in the task of the decision trees are [4,34]. In our experiments, we will use two well-known decision tree induction algorithms, ID3 [34] and C4.5 [36]. Both come from the same assumptions, but while ID3 only makes ‘‘prepruning’’, C4.5 incorporates both pruning techniques.

3.2. Instance-based learning

Instance-based learning (IBL) has its root in the study of nearest neighbour algorithm in the field of *machine learning*. The simplest form of nearest neighbour (NN) or *k*-nearest neighbour (*k*-NN) algorithms [9] simply store the training instances and classify a new instance by predicting that it has the same class as its nearest stored instance or the majority class of its *k*-nearest stored instances according to some distance measure, as described in [42]. The core of this non-parametric paradigm is the form of the similarity function that computes the distances from the new instance to the training instances, to find the nearest or *k*-nearest training instances to the new case.

Some topics of interest in this paradigm are: selection of good cases from training set, reduction of storage and computing-time requirements, tolerating noise, learning attribute relevances and discretization of continuous attributes. Some works proposing solutions to these topics are [1,43]. In our experiments we will use two instance-based inducers:

- IB4, a inducer of the family of the IBL algorithms developed by Aha et al. [1]. IB4 stores only misclassified instances, it keeps a classification performance record for each saved instance and removes some of the saved instances that are believed to be noisy instances using a significance test.
- IB, a inducer developed in the MLC++ project [21] and based on the works of Aha [2] and Wettschereck [43]. IB is very similar to IB4, which has all the functions of IB3 with an additional attribute weight learning capability. The attribute weights are increased for attributes with similar values for correct classifications or for attributes with different values for incorrect classifications, and they are decreased otherwise. The weight of each attribute reflects the attribute’s relative importance for classification.

3.3. Rule induction

One of the most expressive and human readable representations for learned hypothesis are the sets of IF–THEN rules, where in the IF part, there are conjunctions and disjunctions of conditions composed of the predictive attributes of the learning task, and in the THEN part, the class predicted for the samples that carry out the IF part appears.

We can interpret a decision tree like the set of rules generated by a rule induction classifier: the tests that appear in the way from the root of a decision tree to a leaf, can be translated to a rule's IF part, the predicted class of the leaf also in the THEN part appears. Some approaches in this way can be found in [35].

Some problems that may be overcome by the rule induction paradigm are: generation of simple rules when noise is present to avoid the overfitting and efficient rule generation when using large databases. Some works in this paradigm are [6,28,31].

In our experiments, we will use Clark and Nibblet's [6] cn2 rule induction program, as well as oneR [15] and Ripper [7]: cn2 has been designed with the aim of inducing short, simple, comprehensible rules in domains where problems of poor description language and/or noise may be present. The rules are searched in a general-to-specific way, generating rules that satisfy large number of examples of any single class, and few or none of other classes. To use the rule set to classify unseen examples, cn2 applies a "strict match" interpretation by which each rule is tried in order until one is found whose conditions are satisfied by the attributes of the example to classify. The oneR is a very simple rule inductor that searches and only applies the best rule in the datafile. Ripper is a fast rule inductor based on ideas obtained in the work made by Cohen [7].

3.4. Naive Bayes classifiers

Theoretically, Bayes' rule minimises error by selecting the class y_j with the largest posterior probability for a given example X of the form $X = \langle X_1, X_2, \ldots, X_n \rangle$, as indicated below:

$$P(Y = y_j|X) = \frac{P(Y = y_j)P(X|Y = y_j)}{P(X)}$$

Since X is a composition of n discrete values, one can expand this expression to:

$$P(Y = y_j|X_1 = x_1, \ldots, X_n = x_n) = \frac{P(Y = y_j)P(X_1 = x_1, \ldots, X_n = x_n|Y = y_j)}{P(X_1 = x_1, \ldots, X_n = x_n)}$$

where $P(X_1 = x_1, \ldots, X_n = x_n|Y = y_j)$ is the conditional probability of the instance X given the class y_j. $P(Y = y_j)$ is the a priori probability that one will observe class y_j. $P(X)$ is the prior probability of observing the instance X. All these parameters are estimated from the training set. However, a direct application of these rules is difficult due to the lack of sufficient data in the training set to reliably obtain all the conditional probabilities needed by the model. One simple form of the previous diagnose model has been studied that assumes independence of the observations of feature variables $X_1, X_2, \ldots, X_n$ given the class variable Y, which allows us to use the next equality

$$P(X_1 = x_1, \ldots, X_n = x_n|Y = y_j) = \prod_{i=1}^{n} P(X_i = x_i|Y = y_j)$$

where $P(X_i = x_i|Y = y_j)$ is the probability of an instance of class y_j having the observed attribute value x_i. In the core of this paradigm there is an assumption of independence between the occurrence of features values, that is not true in many tasks; however, it is empirically demonstrated that this paradigm gives good results in medical tasks.

In our experiments, we use this Naive Bayes (NB) classifier. Furthermore, we use a Naive Bayes Tree (NBTree) classifier [20], which builds a decision tree applying the Naive Bayes classifier at the leaves of the tree.

4. Layer-1 classifier: Bayesian network

In this section we present the level-1 classifier used in the proposed approach. We use a Bayesian network (BN) as a classifier system, using the results given by the level-0 classifiers as predictor variables that we instantiate in order to give the multi-classifier prediction as the most probable class predicted by the BN once propagation is made. We briefly present Bayesian networks and *genetic algorithms*, and then go on to show the BN structure obtained in this experiment. We use *genetic algorithms* in order to carry out the search in the BN structures space [22], as we are looking for the best BN from the classification point of view. Obtained BN is used as a consensed voting system for the level-0 single classifiers. There is assumed that the BN reflects the interrelation among the different classifiers used [17].

4.1. Bayesian networks

Bayesian networks (BNs) [5,8,18,23,33] constitute a probabilistic framework for reasoning under uncertainty. From an informal perspective, BNs are directed acyclic graphs (DAGs), where the nodes are random variables and the arcs specify the independence assumptions that must be held between the random variables.

BNs are based upon the concept of conditional independence among variables. This concept makes a factorisation of the probability distribution of the n-dimensional random variable $(X_1, \ldots, X_n)$ possible in the following way:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathrm{pa}(x_i))$$

where x_i represents the value of the random variable X_i, and $\mathrm{pa}(x_i)$ represents the value of the random variables parents (direct precessors in the graphical representation) of X_i.

Thus, in order to specify the probability distribution of a BN, one must give prior probabilities for all root nodes (nodes with no predecessors) and conditional probabilities for all other nodes, given all possible combinations of their direct predecessors. These numbers in conjunction with the DAG, specify the BN completely.

Once the network is constructed, it constitutes an efficient device to perform probabilistic inference. This probabilistic reasoning inside the net can be carried out by exact methods [3,24], as well as by approximate methods [13,32]. Nevertheless, the problem of building such a network remains. The structure and conditional probabilities necessary for characterising the network can be either provided externally by experts or obtained, as in this paper, from an algorithm which automatically induces them.

4.2. Bayesian networks as classifiers

During the last 5 years a good number of algorithms with the aim of inducing the structure of the Bayesian network that best represents the conditional independence relationships in a database of cases have been developed. We are using the well-classified percentage of cases as fitness of the BNs structures. In our opinion, the main reason for continuing the research in the structure learning problem is that modelling the expert knowledge has become an expensive, unreliable and time-consuming job. See [12] for a good review.

4.2.1. Naive Bayes approach

This simple, but effective approach, assumes independence among all the predictor variables given the class. In this model the Bayesian network structure is fixed, having all predictor variables as sons of the variable to be predicted, as can be seen in Fig. 4.

Although this independence assumption seems to be very strong, this approach works very well in the medical world, perhaps because chosen symptoms usually have some degree of independence.

4.2.2. Markov Blanquet approach

The Naive Bayes method takes the fact that there is a special variable to be classified into account. However, this approach does not manage the intrinsic semantics of BNs in an adequate manner.

Taking into account that in a BN any variable is influenced only by its Markov Blanquet (MB), that is, its parent variables, its children variables and the parent variables of its children variables, it would therefore seem to be intuitive to do the search in the set of structures that are MB of the variable to be classified. A general Markov Blanquet structure with respect to the class variable C can be seen in Fig. 5.

This concept of Markov Blanquet has been used for the construction of Bayesian network classifier by Friedman et al. [10] and by Sierra and Larrañaga [38].

4.3. Genetic algorithm

The computing complexity inherent in a great number of real problems of combinatorial optimisation has carried, as a consequence, the development of heuristic methods that try to tackle these problems successfully. A heuristic is a procedure which will give a good

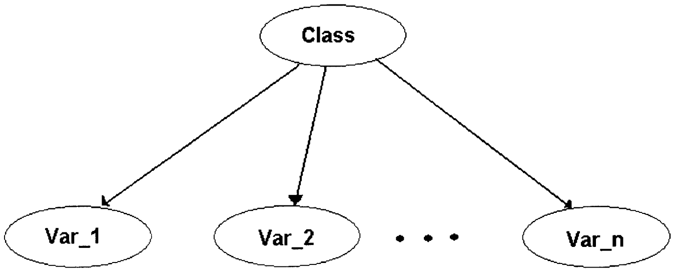

Fig. 4. Naive Bayes structure.

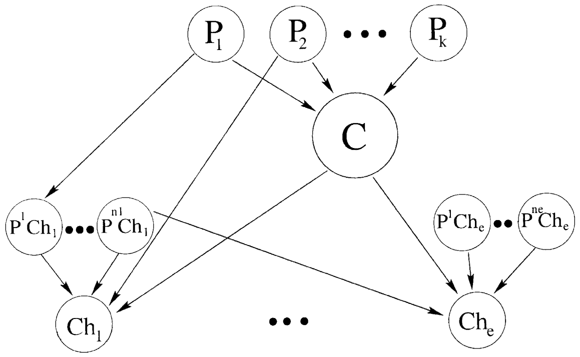

Fig. 5. General Markov Blanquet structure.

solution — not necessarily the optimal — to problems which can be catalogued as difficult, if you try to solve them looking for the exact solution. Although there are heuristics developed for specific problems, in the past years there has been an explosion in the applications of what we could call meta-heuristics, because their formulation is independent of the problem to solve. Among the most studied meta-heuristics, we quote *simulated annealing*, *tabu search* and *genetic algorithms*.

Genetic algorithms [11] are adaptive methods that can be used for solving problems of search and optimisation. They are based on the genetic process of living organisms. Through generations, the populations evolve in nature according to the principles of natural selection and survival of the fittest postulated by Darwin. Imitating this process, *genetic algorithms* are capable of creating solutions for real world problems.

Genetic algorithms use a direct analogy with natural behaviour. They work with a population of individuals, each individual representing a feasible solution to a given problem. To each individual we assign a value or score according to the goodness of its solution. The better the adaptation of the individual to the problem, the more likely it is that the individual will be selected for reproduction, crossing its genetic material with another individual selected in the same way. This cross will produce new individuals, offspring of the previous, which will share some of the features of their parents. In this way, a new population of feasible solutions is produced, replacing the previous one and verifying the interesting property of having greater proportion of good features than the previous population. Thus, through generations good features are propagated among the population. Favouring the cross of the fittest individuals, the most promising areas of the search space are being explored. If the *genetic algorithms* have been well designed, the population will converge to an optimal solution of the problem.

Fig. 6 summarises the pseudo-code for the so-called *abstract genetic algorithm* where parent selection does not need to be done by assigning a value proportional to its objective function to each individual, as is usual in the so-called *simple genetic algorithm*. This selection can be carried out by any function that selects parents in a natural way. It is worth noticing that descendants are not necessarily the next generation of individuals, but that this generation is made by a selection done from the union of parents and descendents. As a result, the operations of extension and reduction in the cycle are needed.

Abstract Genetic Algorithm

```
begin AGA
    Make initial population at random
    WHILE NOT stop DO
      BEGIN
      Select parents from the population
      Produce children from the selected parents
      Mutate the individuals
      Extend the population by adding the children to it
      Reduce the extended population
      END
    Output the best individual found
end AGA
```

Fig. 6. The pseudo-code of the *abstract genetic algorithm.*

4.4. Obtained model

In this paper, we use a methodology for automatically inducing Bayesian networks with a Markov Blanquet structure with respect to the class variable based on *genetic algorithms* [38].

In this approach, each individual in the *genetic algorithm* will be a Bayesian network structure, and all the predictor variables form the so-called Markov Blanquet of the variable to be classified.

In Fig. 7 the structure of the induced MB structure is presented. As shown, given a datafile of cases, we learn the model for each of the level-0 classifiers, and then we learn the structure of the Bayesian network that maximises the performance as classifier system.

5. Experimental results

In order to give a real perspective of applied methods, we use 10-fold cross-validation [40] in all the experiments. The data has been collected at the ICU service of the Canary

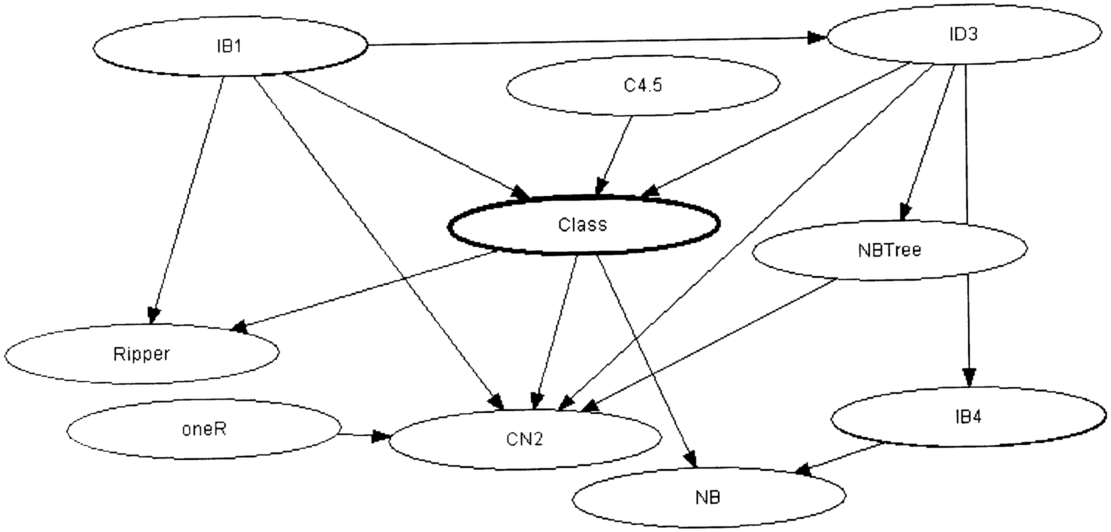

Fig. 7. Bayesian network structure.

Table 1
ICU datafile variables

Variable	Type	Explanation
APACHE II	Continuous	Medical standard method (probability)
SAPS II	Continuous	Medical standard method (probability)
MPM II	Continuous	Medical standard method (probability)
Age	Continuous	Years old
Sex	Discrete	Patient sex
Comes from	Discrete	The place the patient comes from
Admission	Discrete	Date
Readmision	Discrete	Date
Cause	Discrete	Hospital internal code
Days before	Continuous	Days at hospital before charged at ICU
Diagnose code	Discrete	Hospital internal code
Diagnose sub-code	Discrete	Hospital internal code

Islands University Hospital, and have been used by Serrano et al. [37] in another kind of medical experiments.

5.1. Datafile

The datafile used in this experimentation contain data about ICU patients at Canary Islands University Hospital. As seen in Table 1, the probabilities given by APACHE II, SAPS II and MPM II as well as some patient data have been taken into account.

The accuracy of each *standard medical method* in the survival classification is shown in Table 2. This is not comparable with used methods, because it cannot be done with cross-validation. So, we can suppose that there may be an overfitting in the percentages of well classified cases.

We have carried out the experiments with the above datafile using all level-0 classifiers. Table 3 shows the experimental results obtained.

As we can see, by using ML standard approaches the best results (cross-validated) are obtained with oneR [15]. The reason for this could be found in the existence of a variable (APACHE II) which captures vital information about a patient's state of health, having a lot of classification power, as can be seen in the work of Sierra et al. [39]. Therefore, more

Table 2
Accuracy level percentage of the probabilities standard medical methods with different separator numbers (thresholds)

	Threshold							
	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70
APACHE II	79.42	83.14	84.05	85.37	86.45	86.86	87.11	86.44
MPM II	83.64	83.47	83.80	85.12	85.70	85.79	86.61	86.52
SAPS II	78.43	80.00	81.16	82.07	83.06	84.63	85.15	85.62

Table 3
Details of accuracy level percentage estimations obtained using standard *machine learning* algorithms and the multi-classifier introduced

Inducer	10-Fold cross-validation accuracy
ID3	73.64 ± 1.55
C4.5	79.59 ± 1.85
NB	75.64 ± 1.53
NBTree	62.64 ± 2.64
IB1	64.30 ± 2.84
oneR	84.55 ± 1.35
cn2	77.52 ± 1.67
Ripper	80.81 ± 1.14
IB4	63.63 ± 1.22
Multi-classifier	87.27 ± 1.07

complex approaches do not induce a model as good as oneR; decision trees with 2-depth (ID3, C4.5), Bayesian structures containing all predictor variables (NB, NBTree), nearest neighbour approaches considering all the features (IB1, IB4), and more complex decision rule inducers (cn2, Ripper [7]), induce models that are more complex than the one by oneR; however, they do not necessarily outperform it, as evidenced by Holte [15].

The multi-classifier can be seen as a classification model which combines the models and predictions induced by ML standard approaches outperforming the single model induced by oneR and APACHE II feature.

6. Conclusion and further work

In our experiments, we have run these single methods using leave-one-out and we have generated other datafile containing obtained classification for each case and each classifier. Using this second datafile we have learned a Markov Blanquet Bayesian network structure by using a *genetic algorithm*. We then ran this classifier using 10-fold cross-validation.

A new multi-classifier construction method is presented in this work for predicting the survival of patients at ICU that outperform existing standard *machine learning* methods by combining them.

As further work, this method will be applied taking the specificity and sensitivity of the data we are using into account, and the method will be applied to bigger databases.

Acknowledgements

This work was supported by the CICYT under Grant TIC97-1135-C04-03, by the Grant UPV 140.226-EA186/96 from the University of the Basque Country, and by the Grant PI 96/12 from the Gobierno Vasco, Departamento de Educación, Universidades e Investigación.

References

[1] Aha D, Kibler D, Albert MK. Instance-based learning algorithms. Machine Learning 1991;6:37–66.

[2] Aha D. Tolerating, irrelevant and novel attributes in instance-based learning algorithms. Int J Man-Machine Studies 1992;36(1):267–87.

[3] Andersen SK, Olesen KG, Jensen FV, Jensen F. HUGIN — a shell for building Bayesian belief universes for expert systems. In: Proceedings of the 11th International Joint Conference on Artificial Intelligence, 1989. p. 1128–33.

[4] Breiman L, Friedman J, Olshen R, Stone C. Classification and regression trees. Monterey (CA): Wadsworth, 1984.

[5] Castillo E. Gutiérrez JM, Hadi AS. Expert systems and probabilistic network models. Berlin: Springer, 1997.

[6] Clark P, Nibblet T. The *cn2* induction algorithm. Machine Learning 1989;3(4):261–83.

[7] Cohen WW. Fast effective rule induction, machine learning. In: Proceedings of the 12th Internacional Conference, 1995.

[8] Cowell RG, Dawid APh, Lauritzen SL, Spiegelharter DJ. Probabilistic networks and expert systems. Berlin: Springer, 1999.

[9] Dasarathy BV. Nearest neighbor (NN) norms: NN Pattern recognition classification techniques. Silver Spring (MD): IEEE Computer Society Press, 1991.

[10] Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. Machine Learning 1997;19(4):131–63.

[11] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Reading (MA): Addison-Wesley, 1989.

[12] Heckerman D, Geiger D, Chickering DM. Learning Bayesian networks: the combination of knowledge and statistical data. Machine Learning 1995;20:197–243.

[13] Henrion M. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In: Proceedings of the 4th Conference on Uncertainty in Artificial Intelligence, 1988. p. 149–63.

[14] Ho TK, Srihati SN. Decision combination in multiple classifier systems. IEEE Trans Pattern Anal Machine Intell 1994;16:66–75.

[15] Holte RC. Very simple classification rules perform well on most commonly used databases. Machine Learning 1994;11:63–90.

[16] Inza I, Larrañaga P, Sierra B, Niño M. Combination of Classifiers. A Case Study in Oncology. Internal Report EHU-KZAA-IK-1-98, 1998.

[17] Inza I, Larrañaga P, Sierra B, Etxeberria R, Lozano JA, Peña JM. Representing the joint behaviour of machine learning inducers by Bayesian networks. Pattern Recognition Letters 1999;20(11-13):1201–9.

[18] Jensen FV. Introduction to Bayesian Networks. University College of London, 1996.

[19] Knaus WA, Draper EA, Wagner DP, Zimmerman JE. APACHE II: a severity of disease classification system. Crit Care Med 1985;13:818–29.

[20] Kohavi R. Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996.

[21] Kohavi R, Sommerfield D, Dougherty J. Data mining using MLC++, a machine learning library in C++, Int J Artif Intell Tools 1997;6(4) 537–66 (http://www.sgi.com/Technology/mlc/).

[22] Larrañaga P, Poza M, Yurramendi Y, Murga R, Kuijpers C. Structure learning of Bayesian networks by genetic algorithms: a performance analysis of control parameters. IEEE Trans Pattern Anal Machine Intell 1996;18:912–26.

[23] Lauritzen SL. Graphical models. Oxford: Oxford University Press, 1996.

[24] Lauritzen SL, Spiegelhalter DJ. Local computations with probabilities on graphical structures and their application on expert systems. J R Statist Soc B 1988;50:157–224.

[25] Le Gall JR, Lemeshow S, Saulnier F. A new simplified acute physiology score (SAPS II) based on a European/North American multicenter study. JAMA 1993;270:2957–63.

[26] Lemeshow S, Teres D, Klar J, Avrunin JS, Gehlbach SH, Rapoport J. Mortality probability models (MPM II) based on an international cohort of intensive care unit patients. JAMA 1993;270:2478–86.

[27] Martin JK. An exact probability metric for decision tree splitting and stopping. Machine Learning 1997;28(2/3).

[28] Michalski RS, Mozetic I, Hong J, Lavrac N. The AQ15 inductive learning system: an overview and experiments. In: Proceedings of IMAL 1986. Orsay (France): Universitéde Paris-Sud, 1986.

[29] Mingers J. A comparison of methods of pruning induced Rule Trees. Technical Report. Coventry (UK): University of Warwick, School of Industrial and Business Studies, 1988.

[30] Murthy SK, Kasif S, Salzberg S. A system for the induction of oblique decision trees. J Artif Intell Res 1994;2:1–33.

[31] Niblett T, Bratko I. Learning decision rules in noisy domains. In: Bramer MA, editor. Research and development in expert systems III, Cambridge: Cambridge University Press, 1987. p. 25–34.

[32] Pearl J. Evidential reasoning using stochastic simulation of causal models. Artif Intell 1987;32(2):245–57.

[33] Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. San Mateo: Morgan Kaufmann, 1988.

[34] Quinlan JR. Induction of decision trees. Machine Learning 1986;1:81–106.

[35] Quinlan JR. In: McDermott J, editor. Generating production rules from decision trees, IJCAI-87. San Francisco (CA): Morgan Kaufmann, 1987. p. 304–7.

[36] Quinlan JR. C4.5: Programs for Machine Learning. Los Altos (CA): Morgan Kaufmann Publishers, 1993.

[37] Serrano N, Revuelta P, Jiménez JJ, Plasencia E, Martínez J, Brouard MT, Mora ML. Levels of severity at ICU discharge related those at ICU admission: Criterion for ICU discharge? Intensive Care Med 24 (1998) s10.

[38] Sierra B, Larrañaga P. Predicting survival in malignant skin melanoma using Bayesian networks automatically induced by genetic algorithms. An empirical comparision between different approaches. Artif Intell Med 1998;14:215–30.

[39] Sierra B, Serrano N, Larrañaga P, Plasencia EJ, Inza I, Jiménez JJ, Revuelta P, Mora ML. Machine learning inspired approaches to combine standard medical measures at an intensive care unit. Lecture Notes in Artificial Intelligence 1999;1620:366–71.

[40] Stone M. Cross-validation choice and assessment of statistical procedures. J R Statist Soc 1974;36:111–47.

[41] Skalak D. Prototipe selection for composite Nearest Neighbor classifiers. Ph.D. Thesis, Amherst: University of Massachusetts, 1997.

[42] Ting KM. Common issues in instance-based and Naive-Bayesian classifiers. Ph.D. Thesis, Basser Department of Computer Science. The University of Sidney, NSW, Australia, 1995.

[43] Wettschereck D. A study of distance-based machine learning algorithms. Ph.D. Thesis, Oregon State University, 1994.

[44] Wolpert D. Stacked generalization. Neural Networks 1992;5:241–59.