A New Sampling Strategy for Building Decision Trees from Large Databases

J.H. Chauchat and R. Rakotomalala

Université Lumière Lyon 2
5 avenue Pierre Mendès France, C.P.11 69676 Bron Cedex, France
e-mail : chauchat,rakotoma@univ-lyon2.fr

Abstract. We propose a fast and efficient sampling strategy to build decision trees from a very large database, even when there are many numerical attributes which must be discretized at each step. Successive samples are used, one on each tree node. Applying the method to a simulated database (virtually infinite size) confirms that when the database is large and contains many numerical attributes, our strategy of fast sampling on each node (with sample size about $n = 300$ or 500) speeds up the mining process while maintaining the accuracy of the classifier.

Keywords: Decision tree, sampling, discretization.

1 Introduction

In this paper we propose a fast and efficient sampling strategy to build decision trees from a very large database, even when there are many numerical attributes which must be discretized at each step.

Decision trees, and more generally speaking decision graphs, are efficient and simple methods for supervised learning. Their "step by step" characteristic allows us to propose a strategy using successive samples, one on each tree node. In that way, one of the decision tree method most limiting aspects is overcome (analyzed data set reduction as the algorithm goes forward, successively dividing the set of training cases).

Working on samples is especially useful in order to analyze very large databases, in particular when these include a number of numerical attributes which must be discretized at each step. Since each discretization requires to sort the data set, this is very time consuming. Section 2 outlines the general decision tree method, the numerical attributes discretization problem and our new sampling strategy at each step.

In section 3, we apply the whole method to a simulated database (virtually infinite size). The results confirm that when the database is large and contains many numerical attributes, our strategy of fast sampling on each node (with sample size about $n = 300$ or 500) reduces drastically learning time while maintaining the accuracy in generalization.

2 Decision trees and induction graphs

2.1 Induction with graphs

Decision trees (Breiman et al, 1984), and more generally speaking decision graphs (Zighed and Rakotomalala, 2000), are efficient, step by step, and simple methods for supervised classification. *Supervised* classification means that a classification pre-exists and is known for each record in the (training) database we are working on: the patient has been cured, or not; the client has accepted a certain offer, or not; the machine breaks down, or not. Those situations have two values; sometimes there are three or more. The final objective is to learn how to assign a new record to its true class, knowing the available attributes (age, sex, examination results, etc.).

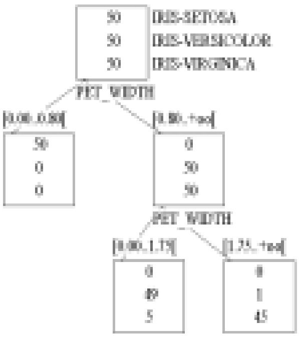

Fig. 1. A decision tree built on Fischer's Iris dataset. Iris classification learning, using petale, and sepale, length and width.

The wide utilization of decision tree method is based on its simplicity and ease of use. One is looking for a dataset partition represented by a lattice graph (Figure 1). This partition must minimize a certain criterion. Generic algorithms (Breiman et al., 1984) (Quinlan, 1986) make local optimization. In spite of their simplicity, decision trees have a very good predictive power compared with more complex method such as Neural Network (Quinlan, 1993). Nowadays, as Knowledge Discovery in Databases (KDD) is growing fast (Fayyad et al., 1996), one can note a growing number of studies on decision trees and induction graphs, as well as broad software diffusion.

2.2 Using continuous attributes in decision trees

Most training-by-examples symbolic induction methods (Cohen, 1995) have been designed for categorical attributes, with finite value sets. For instance "sex" has two values: male or female. However, when we want to use continuous attributes (income, age, blood pressure, etc.), we must divide the value set in intervals so as to convert the continuous variable into a discrete one. This process is named "discretization". The importance of this research area has recently become apparent.

Ever-growing data, due to the extensive use of computers, the ease of data collection with them and the advance in computer technology, drive dataminers into handling databases comprising varied type and non pre-processed attributes.

The first methods for discretization were relatively simple, and few papers have been published to evaluate their effects on machine learning results. From the beginning of the '90s much theoretical research has been done on this issue. The general problem has been clearly formulated (Lechevallier, 1990) and several discretization methods are now in use (Zighed et al., 1998). Initial algorithms processed discretization during the pre-processing stage: each continuous attribute was converted to a discrete one; after which, a regular symbolic learning method was used.

Within the particular framework of the decision graphs, it is possible to simplify the discretization of a continuous attribute by carrying out a binary local cutting. The process is as follows: on each node of the tree, each continuous variable is first of all sorted, then all the possible cutting points are tested so as to find the binary cut which optimizes a criterion such as information gain or mutual information measure (Shannon and Weaver, 1949).

This strategy thus makes it possible to compare the predictive capacity of all the attributes, whether continuous or not. In spite of this simplicity, we are facing here one of the principal bottlenecks in the development of graphs. Cross tabulation, on which the criteria of the partitions are calculated, is a relatively inexpensive phase: it is of $O(n)$. On the other hand the processing of the continuous variables requires initially a sorting of the values in ascending order which, in the best case, is of $O(n \log n)$. This is why we propose to use sampling by reducing the number n of records to which these calculations apply.

2.3 Using successive samples, one on each tree node

To each node of the graph corresponds a subpopulation; it can be described by the conjunction of the attribute values located on the outgoing path from the root to the considered node. Thus, to split a node, the following general framework reduces dramatically the computing time :

1. draw a sample from the subpopulation corresponding to the node (see Figure 2);
2. use this sample to discretize continuous attributes and to determine the best splitting attribute.

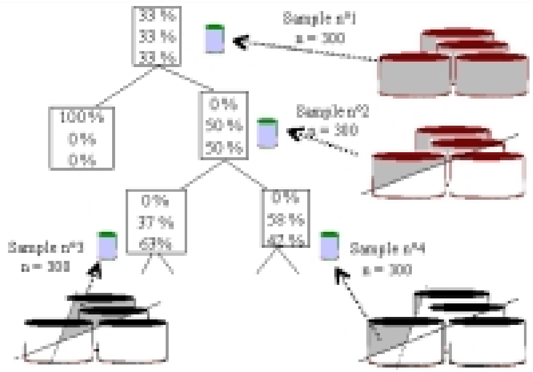

Fig. 2. Sampling and re-sampling on each node to build a decision tree.

Sampling should save time during data processing, but the sampling operation itself should not be time consuming. One can use very fast sampling methods (Vitter, 1987).

In order to determine the sample size, elements of statistical theory (using statistical test power function and non central chi-squared distribution) are presented in (Chauchat and Rakotomalala, 1999). Samples of a few hundred are usually enough to determine interesting predictive attributes.

3 Implementation on simulated databases

We will now apply the whole method (with sampling and binary discretization on each node of the tree) to a well known artificial problem: the "Breiman's waves" (Breiman et al., 1984). In § 2.6 of his book, Breiman poses this problem, now traditional: each of three classes is characterized by a specific weighting combination of 21 pseudo-random standardized normal variables.

We generated 100 times two files, one of $500,000$ records for the training, the other of $50,000$ records for the validation. Binary discretization was preprocessed on each node for each attribute. Sample size drawn from the file on each node varies from $n = 100$ to $n = 500$. ID3 method has been used because it is fast; it uses pre-pruning with the χ^2- test.

The learning time is quasi null for $n = 100$ because the tree stops very quickly, even immediately: the pruning χ^2-test has a low power (if n is too small, the *observed*-χ^2 is small too, even if an attribute is useful). From $n = 200$, the run time increases a little quicker than linearly with n, in accordance with theory ($nlog(n)$).

Figure 3 shows how the error in generalization decreases as the sample size n increases. Even for this problem considered as a difficult one, the marginal profit becomes weak starting from sample sizes of $n = 300$ records; one approaches then 19%, the minimum of error in generalization obtained with trees using the entire database, (with its $N = 500,000$ records).

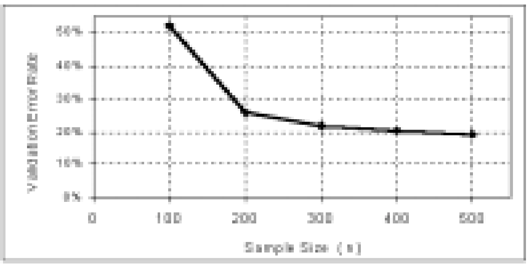

Fig. 3. Average ERROR RATE according to the sample size drawn on each node (Breiman's Waves Dataset; Optimal discretization of 21 continuous attributes at each step).

These results have been confirmed by several empirical studies on real databases. From a sample of approximately 300 records on each node, we obtain trees error rate close to that obtained using the whole database.

4 Conclusions

Decision trees, and more generally speaking decision graphs, are efficient, step by step, and simple methods for supervised classification. However, mining on very large databases, in particular when these include a number of numerical attributes which must be discretized at each step, is very time consuming. In these cases, working on samples is especially useful. The decision tree "step by step" characteristic allows us to propose a strategy using successive samples, one on each tree node. Empirical evidences show that our strategy of fast sampling on each node (with samples size about $n = 300$ or 500) reduces considerably learning time while preserving the accuracy. Sampling in data mining is not a new approach. (Toivonen, 1996) for instance is looking for the minimum sample size to get all the useful association rules. Our goal is different : we are not looking for the same decision tree as that built on the whole databases, what is impossible using a sample, but one which have roughly the same error rate.

This work raises some open questions. Optimal sampling methods (stratified random sampling, selection with unequal probabilities, etc.) may be used. However, those methods were developed for surveys on economic or sociological fields, when the cost of the information collected is high compared to calculation time. They must be adapted for data mining: in our situation the information is already known, it is in the database. We have to check if the gain in accuracy obtained by these methods, the sample size being fixed, may not be supplied by sample size enlargement, for the same learning time. An interesting way may be the balanced sampling strategy on each node (Chauchat et al., 1998).

An other question is the implementation of sampling in the core of the queries in databases.

References

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984) *Classification and Regression Trees.* California : Wadsworth International.

Chauchat, J., Boussaid, O., and Amoura, L. (1998) Optimization sampling in a large database for induction trees. In *Proceedings of the JCIS'98-Association for Intelligent Machinery*, 28–31.

Chauchat, J., and Rakotomalala, R. (1999) Détermination statistique de la taille d'échantillon dans la construction des graphes d'induction. In *Actes des* 7^{emes} *journées de la Société Francophone de Classification*, 93–99.

Cohen,W. (1995) Fast effective rule induction. In *Proc. 12th International Conference on Machine Learning*, 115–123. Morgan Kaufmann.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996) Knowledge discovey and data mining : Towards an unifying framework. In *Proceedings of the* 2^{nd} *International Conference on Knowledge Discovery and Data Mining.*

Lechevallier, Y. (1990) Recherche d'une partition optimale sous contrainte d'ordre totale. Technical Report 1247, INRIA.

Quinlan, J. (1986) Induction of decision trees. *Machine Learning* 1:81–106.

Quinlan, J. (1993) Comparing connectionist and symbolic learning methods. In Hanson, S.; Drastal, G.; and Rivest, R., eds., *Computational Learning Theory and Natural Learning Systems : Constraints and Prospects.* MIT Press.

Shannon, C. E., and Weaver, W. (1949) *The mathematical theory of communication.* University of Illinois Press.

Toivonen, H. (1996) Sampling large databases for association rules. In *Proceedings of* 2^{nd} *VLDB Conference*, 134–145.

Vitter, J. (1987) An efficient algorithm for sequential random sampling. *ACM Transactions on Mathematical Software* 13(1):58–67.

Zighed, D. and Rakotomalala R. (2000) *Graphes d'Induction : Apprentissage et Data Mining.* Hermes.

Zighed, D., Rabaseda, S., and Rakotomalala, R. (1998) Fusinter : a method for discretization of continuous attributes for supervised learning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6(33):307–326.