Sampling Strategies for Targeting Rare Groups from a Bank Customer Database

J-H. Chauchat[1], R. Rakotomalala[1], and D. Robert[2]

[1] ERIC Laboratory - University of Lyon 2
5, av. Pierre Mendes-France
F-69676 Bron - FRANCE

[2] Crédit Agricole Centre-Est
1, rue Pierre de Truchis
F-69410 Champagne aux Monts d'Or - FRANCE

Abstract. This paper presents various balanced sampling strategies for building decision trees in order to target rare groups. A new coefficient to compare targeting performances of various learning strategies is introduced. A real life application of targeting specific bank customer group for marketing actions is described. Results shows that local sampling on the nodes while constructing the tree requires small samples to achieve the performance of processing the complete base, with dramatically reduced computing times.

Keywords: sampling, customer targeting, targeting quality coefficient, imbalanced database, decision tree, application

1 Introduction

This paper studies supervised learning using a real life application of targeting for the "Crédit Agricole Centre-Est" bank. More specifically, the use of decision trees [1][2] or induction graphs [19] on large databases to learn discriminating between two unequal size classes is of interest.

Crédit Agricole manages a several hundred of thousands customers data-base for whom some 200 attributes are known, 95% of them continuous. The class attribute is whether a client connected to some remote service. The study should identify those clients most susceptible to connect in the future; these types of clients shall be targeted by remote services marketing campaigns.

Learning form imbalanced classes is known to be difficult, yet it is quite common in practice: detection of rare diseases in epidemiology; detection of bank card frauds; process breakdown forecasting in industry; targeting specific client groups for marketing actions.

Moreover, if the database is large, the computing time is long especially if continuous attributes must be optimally discretized at each step [18][7]. Then, learning must be done on a sample [5], with efficiency gains if the sample is balanced [4].

The paper is organized as follows: the two sampling strategies that were implemented are presented in the next section. A new coefficient to compare

D.A. Zighed, J. Komorowski, and J. Żytkow (Eds.): PKDD 2000, LNAI 1910, pp. 181–190, 2000.

client-targeting performances is introduced in the third section. This coefficient is used to compare the quality of the decision trees derived from the various sampling methods. The fourth section presents numerical results (computing time and targeting quality coefficient) from the bank customers database. Conclusion and future work are in the fifth section.

2 Sampling Strategies and Probability Distributions in Decision Trees

The focus here is on balanced sampling; indeed, the detection of rare classes with the classical induction tree on imbalanced training set works poorly [10]. Sampling for a decision tree can be executed in one of the two following ways:

1. either a sample is drawn from the original database, and the tree is built from the sample;
2. or a random sample is drawn on each node of the tree as it is being constructed.

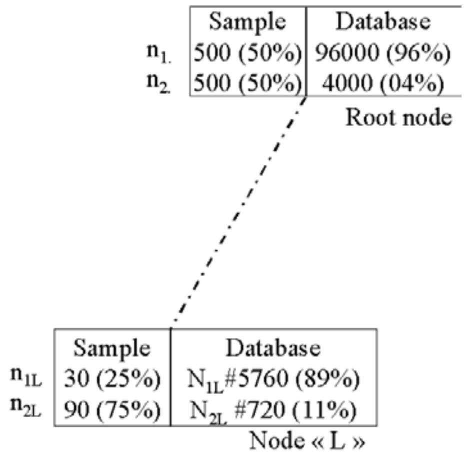
Each method has advantages and disadvantages. The former is quicker, as it accesses the database only once and builds a learning set from the sample. On the other hand, as the tree grows, the leaves become smaller and smaller, making estimation of the probabilities less reliable while a wealth of data is available to comfortably make those estimations. If those probabilities are to be estimated for the initial population (or the complete base taken as the population), then they must be adjusted using Bayes theorem to obtain correct distributions on each node [12].

The latter is not hampered by data fragmentation. As the tree is constructed, on each node, the needed sample is drawn. There is, however, a severe drawback to this method for multiple accesses to the database are required. Even with fast algorithms [16], the method remains computer intensive. On the other hand, at each pass, exact probability distributions can be computed.

2.1 Building a Global Sample Before the Learning Process

A random sample of size $n_{k.}$ is to be drawn from the original database for each of the values y_k of the class attribute Y $(k = 1, \ldots, K)$. The size of the sample file is n $(n = \sum_{k=1}^{K} n_{k.})$. If the $n_{k.}$ are equal, the sample is said to be balanced. This sampling scheme is a K-sample retrospective sampling [3], the $n_{k.}$ are not random and cannot be used to estimate the $\pi_{k.} = P(Y = y_k)$, the prior probabilities of obtaining one of the values for the class attribute. Here, the $\pi_{k.}$ are considered as computed from the complete database.

Let ℓ be a leaf on the decision tree. This leaf can be described by a statement such as $(X_1 = x_1, \ldots, X_p = x_p)$, and correct estimates of the conditional probabilities $P(Y = y_k/l) = P(Y = y_k/X_1 = x_1, \ldots, X_p = x_p)$ can be obtained; the later can be derived from the n_{kl}, the observed empirical frequencies on ℓ, the leaf of interest (Figure 1).



	Sample	Database
$n_{1.}$	500 (50%)	96000 (96%)
$n_{2.}$	500 (50%)	4000 (04%)

	Sample	Database
n_{1L}	30 (25%)	N_{1L}#5760 (89%)
n_{2L}	90 (75%)	N_{2L} #720 (11%)

Fig. 1. Estimated sample size and conditional distribution on a node using global sampling

Posterior probabilities can be obtained using Bayes theorem:

$$\begin{aligned} \pi_{k/l} &= P(Y = y_k/l) && (1)\\ &= \frac{P(Y = y_k) \times P(l/Y = y_k)}{P(l)} && (2)\\ &= \frac{P(Y = y_k) \times P(l/Y = y_k)}{\sum_{j=1}^{K} P(Y = y_j) \times P(l/Y = y_j)} && (3) \end{aligned}$$

The estimates from the learning sample are readily obtained as:

$$\widehat{\pi}_{k/l} = \frac{\pi_{k.} \times \frac{n_{kl}}{n_{k.}}}{\sum_{j=1}^{K} \pi_{j.} \times \frac{n_{jl}}{n_{j.}}} \tag{4}$$

If the population size is noted N, the number of individuals accounted for by the leaf ℓ is given by:

$$\widehat{N}_{.l} = N \times \left[\sum_{k=1}^{K} \pi_{k.} \times \frac{n_{kl}}{n_{k.}} \right] \tag{5}$$

The main advantage of this method is that a single pass is required to obtain N and the $\pi_{k.}$. The induction tree and probability estimates are obtained from the sample, which can be a separate file created once and for all before the learning process (if the sample size is reasonable, it can fit in memory). The reliability of the estimates of the conditional probabilities depends on the sample size [14].

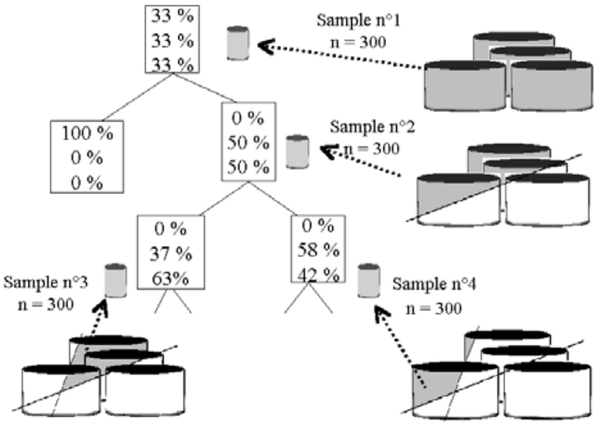


Fig. 2. Steps of building decision tree using local sampling

2.2 Local Sampling while Constructing the Tree

This approach follows work developed in [5]. On each leaf, while constructing the tree, a sample is drawn from the section of the base outlined by the rules defined by the path leading to the leaf (Figure 2). Each time, the sample is full size as long as the database contains enough individuals for the leaf; otherwise the available individuals are selected. Thus, little information is spoiled: at first, information is superabundant and a sample is enough to set the correct rules in a reasonable time; by the end, when information becomes scarce, a larger fraction of what is available is drawn, even all of it.

Computing time is less than that in learning from the complete base, especially when the database contains many continuous attributes that need to be sorted and discretized.

The property of decreasing global entropy may be lost when selecting a new sample on each node, but this is of little consequence. When a tree is built on a fixed set of examples, the global entropy can only decrease at each step [13], but this is an artefact of the learning set. In general, this property does not translate well to another set on which the tree would be applied, for example, a test sample. And, truly, the dataminer is especially interested in the generalization properties.

Compared to a global sample drawn prior to learning, the need to go back to the base to sample for each node allows the determination of the size of the population concerned and of the exact probabilities.

In practice, build a decision tree with local sampling is as follows:

1. first, a complete list of examples on the base is drawn;
2. the first sample is selected while the base is being read; an array of records associated with each attribute is kept in memory;

3. this sample is used to identify the best segmentation attribute, if it exists; otherwise, the stopping rule has played its role and the node becomes a terminal leaf;
4. if a segmentation is possible, then the list in (1.) is broken up into sub-lists corresponding to the various leaves just obtained;
5. step (4.) requires passing through the database to update each example's leaf; this pass is an opportunity to select the samples that will be used in later computations.

Steps (3.) to (5.) are iterated until all nodes are converted to terminal leaves.

3 TQC, a New Coefficient to Compare Tracking Procedures

In this section, TQC (Targeting Quality Coefficient), a coefficient to compare two tracking procedures is introduced. The coefficient is similar to a Gini coefficient in statistics [8], lift charts used in marketing, ROC curves from signal theory [6] or medicine [15]. The coefficient can help comparing trees derived from different sampling processes and that constructed on the complete database.

In general, classifiers are compared using the "test error rate", that is the proportion of "misclassified" among a sample independent of the learning sample [11]. For the situation at hand (tracking rare groups), the usual error rate is ill adapted. Rather than looking for the most likely class of an individual given his characteristics, the probability of having a rare characteristic is estimated: disease, fraud, breakdown, tele-purchase...

Individuals with a predicted probability of belonging to the rare group of at least $x\%$ are tracked; by varying $x\%$ with respect to cost and expected benefits ensuing actions, a larger or smaller set of individuals "at risk" is selected.

Hence, the quality coefficient must depend on the predicted probabilities given by the classifier: it ranges from 0 for a random classification (i.e. all predicted probabilities are equal to p, the global probability of having the rare characteristic), to 1 if the classifier recognizes perfectly the members of both classes (in this case, the predicted probability is 1 for members having the rare characteristic, and is set to 0 for the other ones).

Table 1 shows how the TQC coefficient is constructed. Individuals are sorted by decreasing predicted probabilities; then, two cumulative functions of the relative frequencies are computed:

1. the cumulative proportion of individuals in the population,
2. the cumulative proportion of individuals with the rare characteristic.

Computations from the decision tree built on a validation file of size $N = 1000$ individuals, with $A = 100$ bearing the rare characteristic, are displayed in Table 1. For a given individual, the predicted probability is the proportion of "rare" individuals among the individuals on the same leaf of the decision tree. For example, selecting the 4 individuals with the largest predicted probabilities (that

Table 1. Building TQC, the Quality Targeting Coefficient, on an artificial example

Rank $= i$	% Total Population	Pred. Prob $= P_i$	Class	% Cumulative Class "1"$= F_i$	Surface element $= (1/N) * (F_{i-1} + F_i)/2$
1	1/N = 1/1000	100 %	1	1/ A = 1/100	(1/N)*(1/A)/2
2	2/N = 2/1000	100 %	0	1 / A = 1/100	(1/N)*(1/100+1/100)/2
3	3/N = 3/1000	70 %	0	1 / A = 1/100	(1/N)*(1/100+1/100)/2
4	4/N=4/1000	70 %	1	2 / A = 2/100	(1/N)*(2/100+1/100)/2
...	...	...	...	...	...
N	N/N = 100%	0 %	0	A/A=100 %	(1/N)*(F_{N-1}+1)/2
SUM =	—	—	A	—	$Area$

is $x\% = 4/1000$ of the population), $F_i = 2/100$ of the "rare" individuals are expected to be covered.

The two cumulative distributions are linked in Figure 3 : the proportion of selected population on the horizontal axis and the estimated proportion of targeted individuals on the vertical axis. The true curve must lie between two extremes:

1. Perfect targeting, displayed as two straight segments joining three points: $(0; 0)$ where no one is selected and no one is covered, $(\frac{A}{N}; 100\%)$ exactly $\frac{A}{N}$ of the population is selected and it is the whole targeted group, and $(100\%; 100\%)$ where every one is selected hence the target is attained;
2. Random targeting, displayed as the diagonal: selecting $x\%$ of the population covers $x\%$ of the targeted group.

The coefficient TQC is defined as the ratio of two areas: the "$Area$" between the real curve and the diagonal, and the area between perfect targeting and the diagonal. From Table 1,

$$TQC = \frac{2 \times Area - 1}{1 - \frac{A}{N}}$$

Hence, $TQC = 0$ for random targeting (no one selected), and $TQC = 1$ when targeting is perfect.

TQC may be negative if a very bad targeting procedure is used: few targeted instances would be selected first, and most of them at the end.

4 Results from Crédit Agricole Client Database

4.1 Characteristics of the Client Database - Sampling Strategies

The Crédit Agricole client base contains several hundreds of thousands of individuals, with some 200 attributes (95% of them continuous). Given the computer available to us, a master sample of 200,000 was drawn to represent the complete database because we want to fit all databases in memory to speed up computing. The attribute of interest is quite skewed, with a prior distribution of 4% "positive" and 96% "negative".

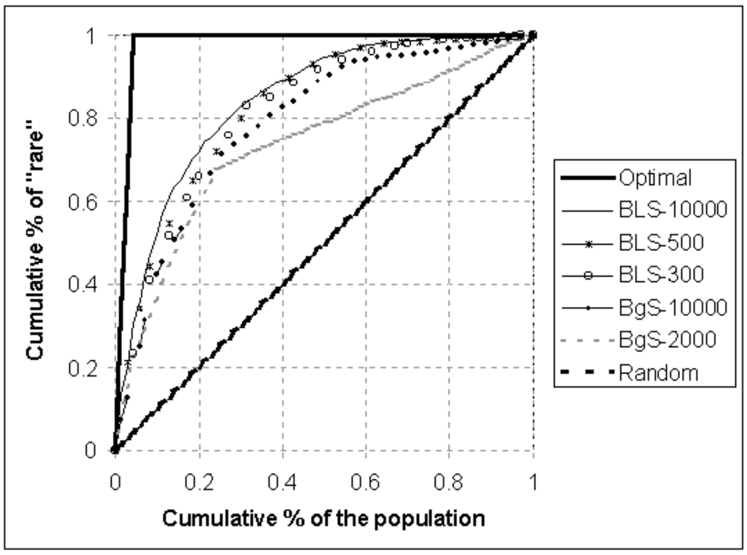


Fig. 3. Comparing various sampling strategies: cumulative proportions scatter-plot for targeting clients most susceptible to connect to some remote service

Three strategies were laid out:

BgS(n) - Balanced global sampling : extract a balanced learning size n sample, and the rules will be applied to the 200, 000 individuals of the master sample; sample sizes of $n = 2000$, $n = 10000$ and $n = 20000$ were tested;

BLS(n) - Balanced local sampling : extract a balanced sample at each node, then apply the classification rules; sizes n=500, 1000, 1500, 2500, 5000 and 10000 were tested;

ALL - All database : work on the full database and apply the rules.

We use the ChAID algorithm [9], the experimentation protocol was as follows: construct a tree according to the suggested strategy (BgS, BLS, ALL), apply the ensuing classification rules to the master sample to obtain predicted probabilities for every individual [17]. Each procedure was replicated ten times.

4.2 Computing Time

Changes in computing time are as expected (Table 2):

- creating a decision tree from the complete database is rather long compared to processing samples;
- computing time for BgS(n) and BLS(n) grows with the sample size;
- learning from a prior sample is quicker than from sample drawn at each node, partly because the number of examples processed at each node diminishes with the growth of the tree;

Table 2. Computing times (in seconds) according to various sampling strategies and sample size

Size	100	200	300	500	1,000	2,000	10,000	20,000	ALL
BgS	-	-	-	-	-	6	42	92	1,381
BLS	2	8	20	45	93	212	311	-	-

Table 3. Quality of targeting coefficient TQC according to various sampling strategies and sample size

Size	100	200	300	500	1,000	2,000	10,000	20,000	ALL
BgS	-	-	-	-	-	0.498	0.600	0.628	0.737
BLS	0.524	0.619	0.664	0.694	0.711	0.722	0.722	-	-

- for comparable computing times [BgS(10,000)-BLS(500), or BgS(20,000)-BLS(1,000)], the quality of prediction for local sampling surpasses that for global sampling (Table 3). This last point is further developed in the next section.

4.3 Quality of Targeting

Using the complete base as our yardstick, for which TQC=0.737 (ALL), the alternative sampling strategies are ranked (Table 3):

- for global sampling (BgS), all possible file sizes were exhausted, yet performances can never approach those achieved by working on the complete file. Indeed, the number of targeted individuals in the master sample does not exceed 8,000 (4% of 200,000); so the largest size of a balanced sample is 16,000 (8,000 positives and 8,000 others). In the sample of 20,000, the balanced sample had to be packed with others (8,000 positives and 12,000 others). Relatively bad targeting quality results of data fragmentation as the tree grows : stoping rules are activated on small sets of individuals; then, test powers are low and no more significant segmentation is find.
- local sampling approaches maximum performance as soon as the local sample size reaches 2,000 on each node. It is remarkable that n=300 seems enough to beat the best performance of global sampling [BgS(20,000)]. This result conforms with earlier empirical and theoretical work [5] on sample sizes for a classic learning problem.

5 Conclusion

The work described here aimed at building an efficient client targeting tool for Crédit Agricole Centre-Est. A number of sampling strategies were developed, well adapted to tracking rare target groups with decision trees. A new quality coefficient was introduced to assess the quality of a tracking strategy. This coefficient is better suited to our study as recognizing individuals is not the goal; isolating those of interest is.

The study shows that local sampling on the nodes while constructing the tree requires small samples to achieve the performance of processing the complete base, with dramatically reduced computing times.

Acknowledgements

This work was done in collaboration with the marketing service of Crédit Agricole Centre-Est. The authors thank Mrs. Morel and M. Chapuisot for their support. We also thank M. Jean Dumais, Section Chief, Methodology Branch, Statistics Canada and guest lecturer at Université Lumière for his careful proof reading of the manuscript and his suggestions.

References

1. L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees.* California : Wadsworth International, 1984.
2. L. A. Breslow and D. W. Aha. Simplifying decision trees: a survey. *Knowledge Engineering Review*, 12(1):1–40, 1997.
3. G. Celeux and A. Mkhadri. Méthodes dérivées du modèle multinomial. In G. Celeux and J.P. Nakache, editors, *Analyse Discriminante Sur Variables Qualitatives*, chapter 2. Polytechnica, 1994.
4. J.H. Chauchat, O. Boussaid, and L. Amoura. Optimization sampling in a large database for induction trees. In *Proceedings of the JCIS'98-Association for Intelligent Machinery*, pages 28–31, 1998.
5. J.H. Chauchat and R. Rakotomalala. A new sampling strategy for building decision trees from large databases. In *Proceedings of the 7th Conference of the International Federation of Classification Societies, IFCS'2000*, pages 45–50, 2000.
6. J.P. Egan. *Signal Detection Theory and ROC Analysis.* Series in Cognition and Perception. Academic Press, New York, 1975.
7. Eibe Frank and Ian H. Witten. Making better use of global discretization. In *Proc. 16th International Conf. on Machine Learning*, pages 115–123. Morgan Kaufmann, San Francisco, CA, 1999.
8. C.W. Gini. Variabilita e mutabilita, contributo allo studio delle distribuzioni e relazioni statische. Technical report, Studi Economico-Giuridici della R. Universita di Caligiari, 1938.
9. G.V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29(2):119–127, 1980.
10. Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: one-sided selection. In *Proc. 14th International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann, 1997.
11. T.M. Mitchell. *Machine learning.* McGraw Hill, 1997.
12. Y.H. Pao. *Adaptive pattern recognition and neural networks.* Addison Wesley, 1989.
13. J.R. Quinlan. Discovering rules by induction from large collections of examples. In D. Michie, editor, *Expert Systems in the Microelectronic Age*, pages 168–201, Edinburgh, 1979. Edinburgh University Press.

14. R. Rakotomalala. *Graphes d'Induction*. PhD thesis, University Claude Bernard - Lyon 1, December 1997.
15. J. Swets. Measuring the accuracy of diagnostic systems. *Science*, 240:1285–1293, 1988.
16. J.S. Vitter. Faster methods for random sampling. In *Communications of ACM*, volume 27, pages 703–718, 1984.
17. I.H. Witten and E. Frank. *Data Mining: practical machine learning tools and techniques with JAVA implementations*. Morgan Kaufmann, 2000.
18. D.A. Zighed, S. Rabaseda, R. Rakotomalala, and F. Feschet. Discretization methods in supervised learning. In A. Kent and J.G. Williams, editors, *Encyclopedia of Computer Science and Technology*, volume 40, pages 35–50. Marcel Dekker, Inc., 1999.
19. D.A. Zighed and R. Rakotomalala. *Graphes d'Induction - Apprentissage et Data Mining*. Hermes, 2000.