ISSN 2279-9362

Collegio Carlo Alberto

Churn Prediction in Telecommunications Industry. A Study Based on Bagging Classifiers

Antonio Canale
Nicola Lunardon

No. 350
April 2014

Carlo Alberto Notebooks

www.carloalberto.org/research/working-papers

CHURN PREDICTION IN TELECOMMUNICATIONS INDUSTRY. A STUDY BASED ON BAGGING CLASSIFIERS

Antonio Canale[1]
Department of Economics and Statistics, University of Turin, Turin, Italy

Nicola Lunardon
Department of Economics, Business, Mathematics and Statistics "Bruno de Finetti", University of Trieste, Trieste, Italy

Abstract *Churn rate refers to the proportion of contractual customers who leave a supplier during a given time period. This phenomenon is very common in highly competitive markets such as telecommunications industry. In a statistical setting, churn can be considered as an outcome of some characteristics and past behavior of customers. In this paper, churn prediction is performed considering a real dataset of an European telecommunications company. An appealing parallelized version of bagging classifiers is used leading to a substantial reduction of the classification error rates. The results are detailed discussed.*

Keywords: *Classification, Data mining, Computational Statistics.*

1. Introduction

In telecommunications industry, customers are able to choose among multiple service providers and actively exercise their right of switching from one service provider to another. In this strongly competitive market, customers demand better services at less price, while service providers constantly focus on retention as their business goal. In this context it is important to manage the phenomenon of churn, i.e. the propensity of customers to cease doing business with a company in a given time period. Given the fact that the telecommunications industry experiences high churn rate and it costs more to recruit a new customer than to retain an existing one, customer retention is clearly more important than customer acquisition. In US wireless market, for example, the retention cost of a customer was estimated at 60$ while the one to acquire a new one at 400$ (Strouse, 2004). Many telecommunications companies apply retention strategies to keep customers longer. With retention strategies in place, many companies start to include churn reduction as one of their business goals. A good prediction of the next leaving customer, leads

[1]Antonio Canale, antonio.canale@unito.it

marketing managers to conceive targeted retention strategies (Bolton et al, 2000; Ganesh et al, 2000; Shaffer and Zhang, 2002) Churn prediction is, therefore, both a marketing and statistical problem. Statistical models provide the predictions that suggest the marketing retention strategies. An interesting discussion, from the marketing viewpoint, about how to measure and understand the predictive accuracy of statistical and data mining models can be found in Neslin et al (2006).

Conventional statistical methods (e.g. logistics regression, classification tree, neural network, generalized additive models) are quite successful in predicting customer churn. Nevertheless, these methods could be improved using some ensemble techniques such as bagging (Breiman, 1996).

In the next section a review of the statistical methods used in this framework is given, focusing especially on bagging. In Section 3 a real data application on telecommunications industry is performed and discussed from a statistical view point. The paper ends with a final discussion and qualitative evaluation of the method.

2. Review of the methods

In what follows we will refer to a dataset with a binary response vector variable y of dimension $n \times 1$, with $y_i = 1$ $(i = 1, \ldots, n)$ if the ith customer has left, and to a $n \times p$ matrix X of quantitative and qualitative explanatory variables. We denote a classifier based on X as $C(X)$ being either 0 or 1.

2.1. Some basic techniques for classification

All the statistical methods used and described are well known in the literature. For a detailed discussion from a data mining point of view consider the book by Hastie et al (2009).

One of the easier way of constructing a classifier $C(X)$ is fitting a linear regression

$$y = X\beta + \epsilon$$

where β is a $p \times 1$ vector of unknown parameters and ϵ is an error term. Obtaining a suitable estimate $\hat{\beta}$ of the parameters one can compute $\hat{y} = X\hat{\beta}$ and then classifies a unit as 1 if $\hat{y}$ is greater than a given threshold t.

Another simple and reliable solution consists in assuming that $y_i \sim Be(\pi_i)$ and hence fitting a logistic model where

$$\pi_i(\beta) = \frac{e^{x_i\beta}}{1 + e^{x_i\beta}} \quad i = 1, \ldots, n,$$

with x_i the ith row of X. The estimated $\hat{\beta}$ are obtained with the usual Fisher's scoring algorithm and a unit is classified as 1 if its estimated probability $\pi_i(\hat{\beta})$ is greater than a threshold.

Logistic and linear regressions are not proper instruments for classification. There are other methods properly studied for classification problems. One of these is the discriminant analysis. Suppose that X are realizations of random variables with different distribution functions in the two groups labelled by y. Defined $p_k(\cdot)$ as the density function of X in the group k ($k = 0, 1$) and w_k as the weight of the kth group in the total population, using Bayes theorem the probability that a unit belongs to the group k is

$$P(y_i = k|x_i) = \frac{w_k p_k(x_i)}{\sum_{h=\{0,1\}} w_h p_h(x_i)}.$$

The comparison between the two classes can be studied with the log-ratio

$$\log \frac{P(y_i = 1|x_i)}{P(y_i = 0|x_i)} = \log \frac{w_0}{w_1} + \log \frac{p_0(x_i)}{p_1(x_i)}$$

and so a unit is classified in the group for which $d(x_i) = \log w_k + \log p_k(x_i)$ is higher. For more details about classification in this classical setting one can refers to Mardia et al (1979).

Classification trees, models performing binary recursive partitions, represent another approach to statistical classification. At each step of the procedure the method tries to divide the units in two groups as similar as possible with respect to the response variable y. The partition is made choosing the cut point of the variable which minimize some objective function. More details can be founded in Breiman et al (1984) and Hastie et al (2009).

Another alternative is represented by additive models, which assume

$$y = \alpha + \sum_{j=1}^{p} f_j(x_j)$$

where x_j is now the jth column of X and f_j is a non parametric smoother such as loess or splines. A generalized additive model (GAM) can be derived from the equation above as

$$g(E\{Y|x_1, \ldots, x_p\}) = \alpha + \sum_{j=1}^{p} f_j(x_j)$$

using a similar argument of the generalized linear models. Here we can think to a logistic additive model using the logit link function in place of $g(\cdot)$. Also with

this approach the classifier needs the choice of a threshold to classify units. GAM are detailed analyzed in Hastie and Tibshirani (1999).

Another nonparametric classifier is obtained with the so called neural network. A neural network consists in a network of variables related by some function. Suppose that the the p explanatory variables are related to an hidden layer of q non observable variables z, linked to the response variable y. The neural network can be thought as a two step regression with

$$z_j = f_0\left(\sum_{h=1}^{p} \alpha_{(h-1)q+j} x_h\right), \qquad y = f_1\left(\sum_{k=1}^{q} \beta_k z_k\right).$$

In our setting the function f_1 is chosen such as y varies between zero and one. Also here the classifier, needs a threshold to classify the units.

2.2. Bagging

Bagging is a method for generating multiple versions of a classifier to get an aggregated one. The multiple versions of the classifier are formed by making bootstrap replicates from the training sample. The name bagging comes from *b*ootstrap *agg*regat*ing*. The idea is that, as in a parliament, each classifier acts as a voter for each unit. The final classification is given by the majority of the votes, given by the B classifiers. The new classifier will summarize all the information collected from the bootstrap replicates. The algorithm is described in Algorithm 1.

Algorithm 1 Bagging algorithm

for $b = 1, \ldots, B$ **do**
Extract the bth bootstrap sample from the data matrix;
Fit the classifier $C_b(x_i)$ to the b-th bootstrapped training sample;
end for
The bagging classifier is

$$C_{bagg}(x_i) = \begin{cases} 1 & \text{if} \quad \frac{1}{B}\sum_{b=1}^{B} C_b(x_i) > t \\ 0 & \text{otherwise} \end{cases}$$

For huge datasets and high number of bootstrap iterations B, clearly the computational burden of a bagging classifier could be time demanding. Since bootstrap replicates are independent, the method can be easily implemented for parallel computation (see Grama, 2003, and references therein). A parallelized version on S slaves is given in Algorithm 2.

Algorithm 2 Parallel version on S slaves of Algorithm 1

for each slave $s = 1, \ldots, S$ **do**
  for $b = 1, \ldots, B/S$ **do**
    Extract the bth bootstrap sample on sth slave from the data matrix;
    Fit the classifier $C_{sb}(x_i)$ to the bth bootstrapped training sample;
  end for
end for
The bagging classifier is

$$C_{bagg}(x_i) = \begin{cases} 1 & \text{if} \quad \frac{1}{B}\sum_{s=1}^{S}\sum_{b=1}^{B/S} C_{sb}(x_i) > t \\ 0 & \text{otherwise} \end{cases}$$

A critical issue is whether bagging will improve accuracy for the given classifier. An improvement will occur for unstable procedures where a small change in the training set can result in large changes in $C(x)$. Particularly in classification trees and neural networks, bagging makes a substantial improvement, and depending on the application the reduction in validation set misclassification rates varies from 6% up to 77%. For both a theoretical and applicative point of view see Breiman (1996).

3. Real data analysis

This section illustrates how the bagging applies to a real dataset of an European telecommunications company available at `azzalini.stat.unipd.it/Book-DM/data.html`. The dataset consists of 30,619 customers for which are collected 99 variables. Among them some are static socio-demographic variables related to the customer or to his/her contractual plan and other are dynamic variables available each month, e.g. duration and number of phone calls, text messages, and so forth, for the 10 months before the target event (churn or not). A detailed list of the variables is shown in Table 1.

The datased is construct to mimic the operative behavior of a company. At month t data for the 10 previous months will be available and the prediction for churn at month $t + 1$ would be performed. Note that, as commonly done in these data mining context, the monthly variables are trated as independent, and without considering their natural dynamic dependence.

The application context requires that the number of churned customer not predicted as such needs to be low, i.e. the false negative error rate must be min-

imized. Descriptive analysis reveal that the proportion of churned customers is $13,8\%$ and this suggests that we should take into proper account the unbalance between the two classes of the response variable. Indeed, it may lead to heavy consequences in pursuing the classification task as, typically, classification rules are overwhelmed by the prevalent class (non churned customers), almost ignoring the rare class (churned customers). There is a rich literature dealing with this issue, so we defer the interested reader to the recent reviews by He and Garcia (2009) and Sun et al (2009).

However, in our case the class imbalance is not severe and we have enough rare observations to construct a simple learning procedure just balancing the training set without having to resort to more sophisticated methods for managing the class imbalance. Indeed such methods are usually adopted in more extreme contexts. Our strategy is, hence to apply the bagging Algorithm 1 to a balanced training set.

The analysis have been carried out by using the statistical software `R` (R Core Team, 2013). We were able to overcome the time demanding nature of the strategy by implementing the parallelized bagging algorithm presented in Section 2.2 based on the package `snowfall` (Knaus, 2013).

Table 1: Variables description

Static varibles	Montly variables (available for 10 months)
Customer identification number	number of exit calls in peak time
Price contract (factor, 5 levels)	duration of exit calls in peak time
Method of payment (factor, 3 levels)	value of exit calls in peak time
Customer sex (factor, 3 levels)	number of exit calls in non-peak time
Age	duration of exit calls in non-peak time
Geographic zone of activation (factor, 4 levels)	value of exit calls in non-peak time
Sales channel (factor, 8 levels)	number of received calls
Presence of service 1 (binary)	duration of received calls
Presence of service 2 (binary)	number of SMS sent
	number of calls to Customer Care Service

We fitted to the data the following models: linear regression, logistic regression, linear discriminant analysis, logistic additive model, a classification tree and a neural network with 5 latent knots with threshold $t = 1/2$, where needed. In particular, we split at random the available data into a (balanced) training set and validation set of size $3,140$ and $1,000$, respectively. The results of the analysis described below have been repeated with different training sets obtaining equivalent results (not reported).

Figure 1 shows the global error evaluated under each model in function of the number of bootstrap iterations. The global errors reported are evaluated with respect to the validation sample. These plots confirm what is stated in Breiman (1996). In our application the worst classifiers are heavily improved. For instance, the bagged version of the linear model reaches a global error of 18%, starting from an error of 31.5%. Clearly, the classifiers that already performed well are improved but not considerably, e.g. using the GAM model the global error starts at 19.4% and ends with "only" 14.4%.

The plots in Figure 2 show the smoothed version of the global error, the false positive and false negative rates. These plots are very interesting since they summarize all the information about the behavior of the bagging classifier. In particular it can be seen that only few classifiers are of particular interest in churn prediction since the goal is the minimization of the false negative rate. These classifiers are the linear model and the neural net, as we can see from plot (a) and (e) of Figure 2; the remaining models produce bagging classifiers that have a false negative rate always bigger than the global error rate.

4. Conclusion

The results of all the methods are reported in Table 2. The global error is decreased using bagging procedures. The improvement on the overall error has not the same magnitude across the two components of the errors: false positives and false negatives. In fact, the false negatives are not really decreased while the false positives strongly turn down. Only the bagged neural network has an appreciable decrease of both the components of the classification error and in particular of the false negative rate. This is probably due to the fact that the validation sample is unbalance and so the main improvement interests especially the majority of the observations, i.e. those with $y = 0$.

From a marketing viewpoint, the improvement in the classification of the customer, according to their churn-propensity, allows several strategies. For example, it allows to better allocate marketing budget to the most likely leaving customer (possibly weighted for their monetary value) or to route incoming calls to the customer care more efficiently according to the estimated propensity of leaving. The predictive gain of the methods discussed here, has a direct effect on the efficiency of any possible marketing operations.

From an operative viewpoint, the parallel implementation of such a methods is appealing for practitioners that are stressed to produce reliable results in reasonable time.

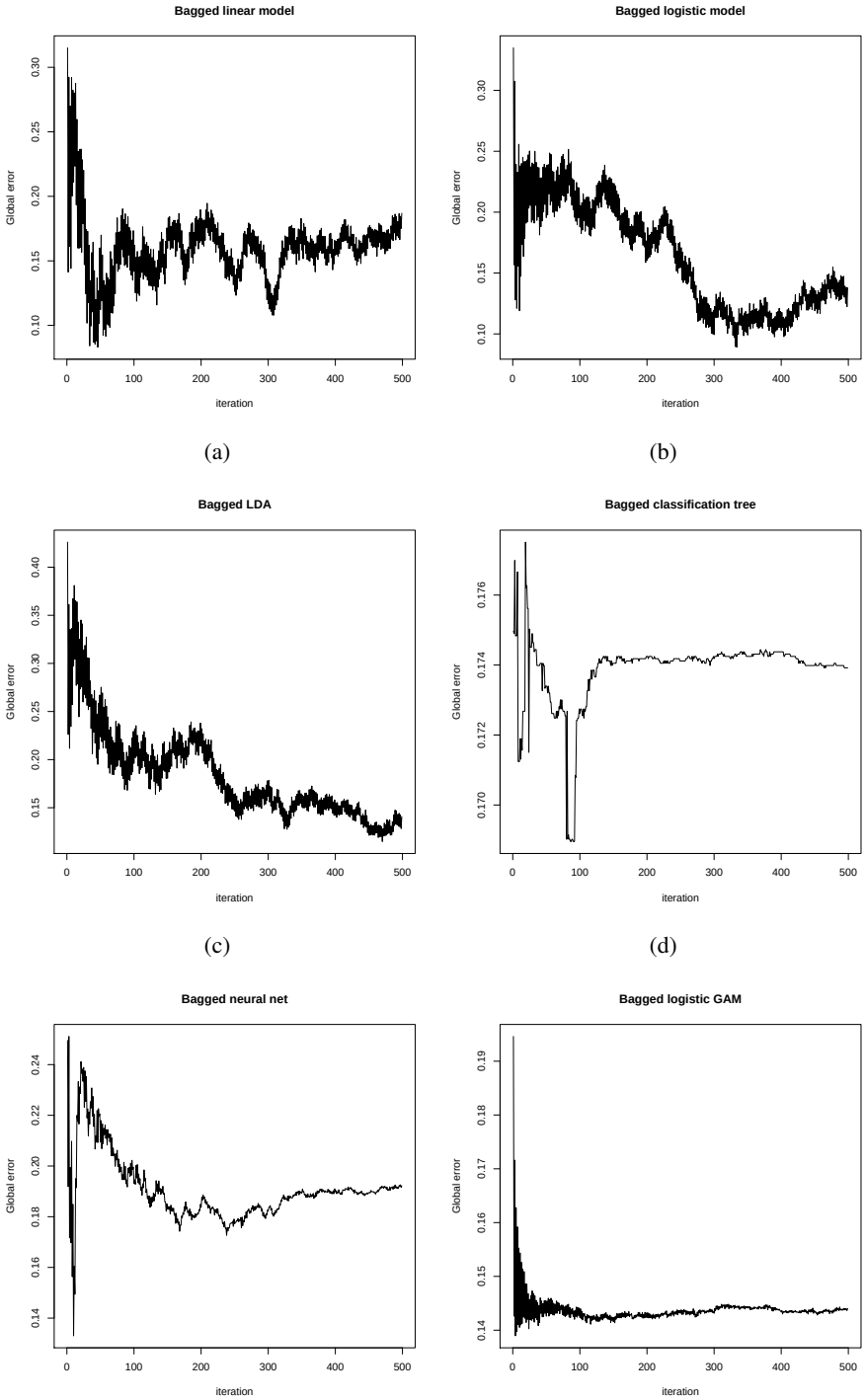
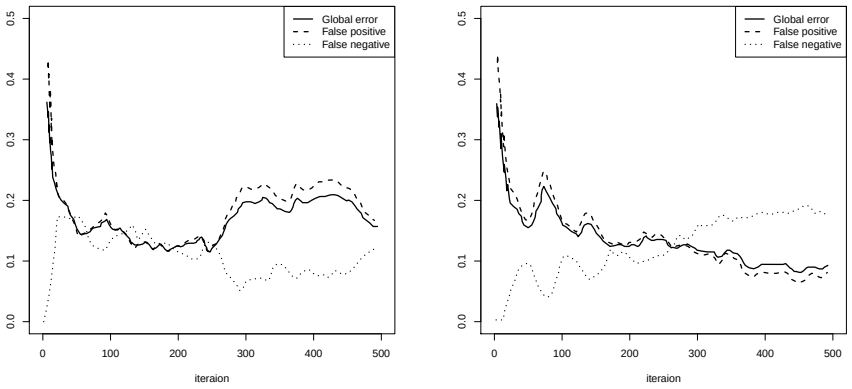
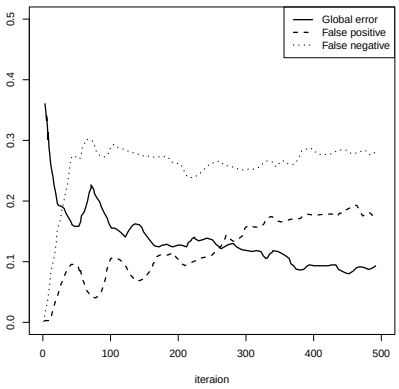


Figure 1: Global error on the validation sample

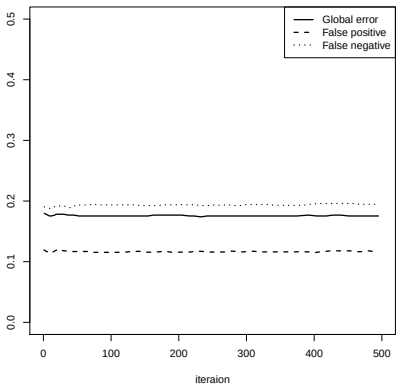


(a)

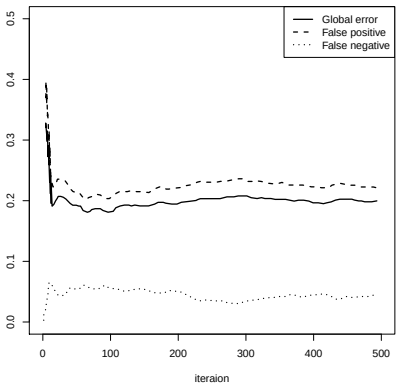
(b)



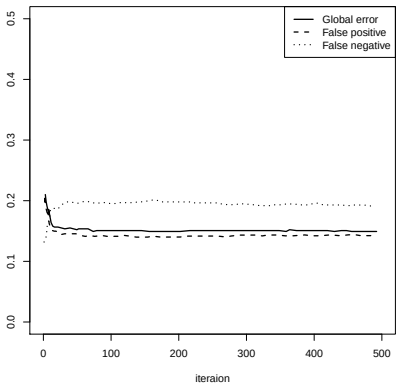
(c)



(d)



(e)



(f)

Figure 2: Smoothed global error and its components

Table 2: Global error, false positive error and false negative error rates for the real data application

	Basic methods			Bagging		
	Global error	False negative	False positive	Global error	False negative	False positive
linear model	33.97	9.02	77.59	16.21	10.84	17.13
logistic regression	34.46	9.08	77.87	8.72	16.93	7.14
discriminant analysis	33.97	9.02	77.59	15.91	26.61	14.19
classification tree	24.94	9.55	72.64	17.44	19.32	11.47
neural net	40.32	10.90	81.88	19.35	4.33	21.76
logistic GAM	31.75	8.49	75.86	14.86	19.14	14.10

To conclude, if companies introduce bagging versions of their commonly used classificators, they might better identify the riskiest customer and thus improve their retention rates.

Acknowledgements

The authors thank Bruno Scarpa for his suggestions in early versions of the paper and the Associate Editor and the referees for comments. This work has been carried out during the PhD studies of the authors at the Department of Statistics, University of Padua, Italy.

References

Bolton R, Kannan P, Bramlett M (2000) Implications of loyalty program membership and service experiences for customer retention and value. Journal of the Academy of Marketing Science 28:95–108

Breiman L (1996) Bagging predictors. Machine Learning 24:123–140

Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA

Ganesh J, Arnold M, Reynold K (2000) Understanding the customer base of service providers: An examination of the differences between switchers and stayers. Journal of Marketing 64:65–87

Grama A (2003) Introduction to parallel computing. Pearson Education

Hastie T, Tibshirani R (1999) Generalized Additive Models. Chapman & Hall Ltd

Hastie T, Tibshirani R, Friedman J (2009) Elements of Statistical Learning. Springer

He H, Garcia EA (2009) Learning from imbalanced data. IEEE Transaction on Knowledge and Data Engineering 21(9)

Knaus J (2013) R package `snowfall`: Easier cluster computing (based on snow). URL `http://CRAN.R-project.org/package=snowfall`

Mardia K, Kent J, Bibby J (1979) Multivariate analysis. Academic Press

Neslin S, Gupta S, Kamakura W, Lu J, Mason C (2006) Defection detection: Measuring and understanding the predictive accuracy of customer churn models. Journal of Marketing Research 43:204–211

R Core Team (2013) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, URL `http://www.R-project.org/`

Shaffer G, Zhang J (2002) Competitive one-to-one promotions. Management Science 48:1143–1160

Strouse K (2004) Customer-centered telecommunications services marketing. Artech House Inc.

Sun Y, Wong AKC, Kamel MS (2009) Classification of imbalanced data: a review. International Journal of Pattern Recognition and Artificial Intelligence 23(4):687–719