

# Making Data Mining Models Useful to Model Non-Paying Customers of Exchange Carriers

Wei Fan<sup>1</sup>      Janak Mathuria<sup>2</sup>      Chang-tien Lu<sup>2</sup>

<sup>1</sup> IBM T.J.Watson Research Center, Hawthorne, NY 10532

`weifan@us.ibm.com`

<sup>2</sup> Virgia Tech, Dept. of Computer Science, Falls Church, VA 20043

`{janakm, ctlu}@vt.edu`

## Abstract

Due to both limitations of technologies and the nature of the problems, data mining may not be able to solve a problem completely in a way as one wishes. When this happens, we need to first understand the actual need of business, characteristic of available partial solution, and then make compromises between the technology solution and business needs. A majority of the papers published in data mining conferences and journals seem to concentrate only on the success side of the story. In this paper, we discuss our experiences and the complete process from near failure to success when applying inductive learning techniques to predict non-paying customers of competitive local exchange carriers (CLEC's), currently at 20%. Experiments with a number of state-of-the-art methods and algorithms found that most customers were labeled as paying on time. Cost-sensitive learning is not possible since the target company cannot define a cost-model. Finally, after discussing with the billing department, a compromised but still useful solution is to predict the probability that someone will default. The billing team can use the predicted score to prioritize collection efforts as well as to predict cash flow. We have found that two randomized decision tree ensemble methods (Fan's random decision tree and a probabilistic extension of Breiman's random forest) are consistently more accurate in posterior probability estimation than single decision tree based probability calibration methods. The software, both Fan's RDT and probabilistic extension of random forest, as well as a longer version of this paper will be made available by the contacting author.

## 1 Introduction

The enactment of the US Telecommunication Act of 1996 has separated infrastructure provider and service provider in order to break up monopoly and create more competitive market for consumers. With this act,

a service provider to consumer is not necessarily the owner of the physical infrastructure, and an infrastructure provider can lease their lines to multiple service providers. A consumer can choose among many competing serve providers without knowing about the underlying infrastructure. After this act was implemented, two kinds of telecommunication companies were created: ILEC and CLEC. ILEC is short for Incumbent Local Exchange Carrier. An ILEC is a telephone company that owns the infrastructure (phone lines, switches, etc) and may also provide local service to end users. An example of ILEC is Verizon. CLEC is short for Competitive Local Exchange Carrier, a telephone company that does not own any infrastructure but rents the infrastructure from an incumbent local exchange carrier (ILEC) and provides services. An incomplete list of CLEC's include Paetec, CBeyond, Alligance, X0, Level 3, and Cypress Communications. There has been an explosion in the number of CLEC's since 1996. As compared to Incumbent Local Exchange Carriers (ILEC's), CLEC's have always been more susceptible to the risk of a very high level of Days Sales Outstanding (DSO) and customer non-payment (at approximately 20%).

We were approached by one CLEC company to build a model to predict which customer will default. It is important to distinguish between two important concepts: "late in payment" and "default". Assume that the closing date of a CLEC company is always on the 1st of the month and the due date is always on the 21st of the same month, and someone does not have any previous balance. The customer receives his current month's statement with the closing date of March 1st. The current amount due is the total charge incurred during the previous billing period from Feb 2nd to March 1st. If he pays this due amount by the 21st, the account is considered "current". However, if the due amount is not paid in full by March 21st but before the next due date of April 21st, the account is marked

as late in payment. If by the next due date of April 21st, the due amount by March 21st is still not paid in full, the account will be labelled as “default”. Besides possibly reporting to a credit agency, there is almost nothing a CLEC company can do to a late customer (but not default ) since the customer is granted a grace period of 30 days before being labelled as defaulted. Even a customer is defaulted, CLEC companies are required under law to send letters out and wait for the customers to remain defaulted for 3 full months before the company are legally allowed to cut off service and submit the case to a third party collection agency.

## 2 Problem Description and Modeling

The back office database of the target company collects call detail history as well as billing and payment history of each customer. The task is simply stated as to “predict if a customer will default in 60 days on the current month’s due solely based on the available billing and calling history”. Ideally, all available history for a particular customer could be used to train the predictive model. However, this may be not be necessary and practical for the billing department. When a customer does not pay for the due amount for three consecutive months, by law, the company can cut off the service and refer the case to a collection agency. Different customers have been with the company for different amount of time, hence there is different amount of information to mine for each customer. In the same time, the target company only has limited database capability and cannot afford large number of sophisticated queries. In the end, the billing department agrees that four months of data is feasible for them and useful under “legal terms”. Assuming that it is currently in March, we use the history data from Dec to March to predict if someone will default in May (or in 60 days from March). We define the following 16 features for each of the four months queried from the database. It is very important to point out that one of these features (the amount outstanding in the >60 days aging bucket) is to label which customer defaults and cannot be used for training.

- Billing Balance Related Features

1. The amount outstanding in the past 0-30 days.
2. The amount outstanding in the 31-60 days.
3. The amount outstanding > 60 days.

- Call Profile Related Features

4. The non-usage revenue for this month.
5. The total number of local calls.
6. The total number of long distance (LD) calls. Long distance calls includes regional toll, inter-state calls and international calls. Long distance calls are generally more expensive than local calls.

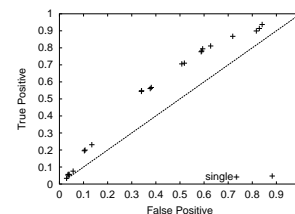
7. The revenue from local calls.
8. The revenue from LD calls.
9. The number of local calls to the 5 most called local numbers
10. The number of LD calls to the 5 most called LD numbers
11. The total amount of local revenue from the calls to the top 5 local numbers.
12. The total amount of LD revenue from the calls to the top 5 LD numbers.
13. The number of after hours local calls. After hour calls are defined as calls originating on weekends or between 6 PM to 8 AM on weekdays.
14. The number of after hours LD calls.
15. The revenue from after hours local calls.
16. The revenue from after hours LD calls.

## 3 Initial Trial, Error, and Failure Experiences

At the start of the project, the billing department aims to catch as many default customers as possible without incurring a large number of false positives. Since the dataset is not skewed at all (approximately 20% non-paying customers), we experimented with decision trees (both commercial C5.0 and free C4.5), naive Bayes, Ripper, Bagging on C4.5, numerical version of AdaBoost on C4.5, and Random Forest (with subset ratio of 0.5). We used the first half of the data (from early months) as training and the remaining half of the data (belonging to later months) for testing. Unfortunately, all these methods nearly predict every customer as “paying on time”. Detailed results can be found in the long version of this paper available upon request from the contacting author. As a twist to the traditional solutions that minimize traditional 0-1 loss, a different approach is to use “probability output” instead of class label output. Probability is a rather continuous output and we may be able to choose a decision threshold with reasonable true and false positive rates. For decision trees, assuming that  $n_c$  is the number of examples with class  $c$  in a node, and  $n$  is the total number of examples in the same node, then the probability that  $\mathbf{x}$  is a member of class  $c$  is simply

$$(3.1) \quad P(c|\mathbf{x}) = \frac{n_c}{n}$$

Figure 1: Complete ROC of Single Decision Tree that outputs probabilities



Assuming that  $t$  is a decision threshold to predict  $\mathbf{x}$  as class  $c$ , i.e.,  $P(c|\mathbf{x}) \geq t$ . Since there are limited number of nodes in a decision tree and each node can output just one probability value for any examples classified by this node, the probability output by decision trees is not completely continuous, and the resultant ROC plot is not continuous either. We draw a complete ROC plot to study if the use of probability will improve the performance. To draw a complete ROC plot, we track all unique probability output of a model and use these values as the decision threshold  $t$ . However, as shown in Figure 1, the single tree’s performance is only slightly better than that of random guessing.

**Re-evaluate the Empirical Need** We experimented with feature selection and a few other variations of feature vector, none of them seemed to help. We realized that these results may just be the fact of this particular problem. Some problems are simply stochastic, or the true model will produce different labels for the same data item at different times. When this happens, we will never be able to have 100% accurate model. The best we can do is to predict the label that happens the most often under traditional 0-1 loss, or the label that minimizes a given loss function under cost-sensitive loss. In order to make the model useful, we spoke to the billing department and were told that the cost model would be very difficult to develop and will probably be over simplified. However, a reliable score that predicts the true probability that someone will default in two months will help both the billing department and the Chief Financial Officer. For the billing department, if the probability output of a model is a reliable estimate of the true probability, it will help the collection team to better prioritize on which customers to expend their efforts when the predicted default customers become late in payment. As mentioned above, CLECs only have limited staff and resources for their collection efforts. Often, when presented with a large list of customers who are late in making their payments, the collection team has very little way of knowing which customers to contact first and very little hope of reaching each and every customer. Focusing on the customers most likely to fall substantially behind on paying their dues would help in preventing outright default in many cases or in cases when this is inevitable, would at least help the company minimizing losses by discontinuing services at the earliest. For the CFO, the estimated probability can be used to compute expected cash flows in two months. If an account has an outstanding balance of \$100 and a 30% chance to default, the expected payment from this account in two months will be  $\$100 \times 0.7 = \$70$ . We sum up over all the outstanding balances and will have a good estimate of actually payment in two

months. We experimented with three methods to estimate the posterior probability, including Fan’s random decision tree [Fan et al., 2003], a variation of Breiman’s random forest [Breiman, 2001], the set of calibrated single decision tree probability methods by Zadrozny and Elkan [Zadrozny and Elkan, 2001].

#### 4 Estimating Posterior Probabilities

We have considered the following methods to estimate posterior probability.

**Fan’s Random Decision Tree** Random Decision Trees or RDT was originally proposed by Fan in [Fan et al., 2003]. The idea of RDT exhibits significant difference from the way conventional decision trees are constructed. RDT constructs multiple decision trees *randomly*. When constructing one particular tree, contrary to the use of purity check functions by traditional algorithms, e.g., information gain, gini index and others, RDT chooses a remaining feature randomly. A discrete feature can be chosen only once in a decision path starting from the root of the tree to the current node. A continuous feature can be chosen multiple times in the same decision path, but each time, a different decision threshold is chosen. The tree stops growing if either the current node becomes empty or the depth of the tree exceeds some predefined limit. Since both feature and decision threshold for continuous features are chosen randomly, random decision trees constructed at different times are very likely to be different. If either every features is categorical or continuous features are discretized, the number of different random trees are bounded. When there are continuous features and random decision threshold is picked, the number of different random trees is potentially unlimited. The depth of the tree is limited to be up to the number of features in order to give each feature equal opportunity to be chosen in any decision path.

During classification, each random tree computes a posterior probability from the leaf node as shown in Eq 3.1. The posterior probability outputs from multiple decision trees are averaged as the final probability estimation. In order to make a decision, a well-defined loss function is needed. For example, if the loss function is 0-1 loss or traditional accuracy, the class label with the highest posterior probability will be predicted. As stated in [Fan et al., 2003], typically 30 random trees should give satisfactory results and more trees may not be necessary. However, experimental studies have shown 10 random trees produce results that sufficiently close to that of 30 random trees. Although quite different from well-accepted methods that employ purity check functions to construct decision trees, random trees have been shown to have accuracy comparable

to or higher than bagging and random forest but at a fraction of the training cost, and has been independently implemented and confirmed separately by Ian Davidson and his student, Tony Fei Liu and Kai Ming Ting, Ed Greengrass, Xinwei Li and Aijun An.

**Probabilistic Extension of Breiman’s Random Forest** Random forest [Breiman, 2001] introduces randomness into decision tree by i) training multiple trees from bootstraps and ii) randomly sampling a subset of remaining features (those not chosen yet by a decision path) and then choosing the best splitting criteria from this feature subset. The chosen size of the subset is provided by the user of random forests. Random forests performance simple voting on the final prediction the same way as bagged decision trees. In this paper, we propose a probabilistic variation of random forest. Instead of training from bootstraps and predicting class label as in Breiman’s random forest, each tree in our version is trained from the original dataset, and outputs posterior probability. Similar to random decision trees, the probabilities from all trees in the forest are averaged as the final probability output. Our variation is called Random Forest+.

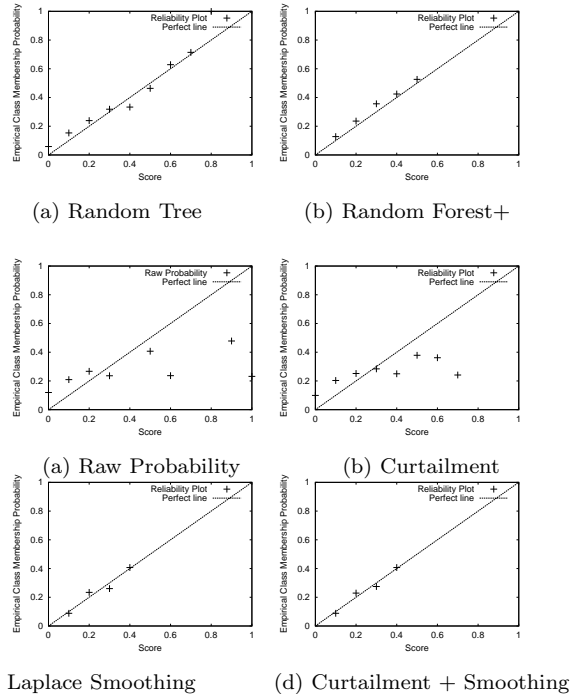
**Zadrozny and Elkan’s Calibrated Probabilities for Decision Trees** Raw probability or original probability of decision trees is defined in Eq 3.1. Smoothed probability considers the base probability in the data and is defined as:  $P(c|\mathbf{x}) = \frac{n_c + m \cdot b}{n + m}$ , where  $b$  is the base rate of positive examples in the training set (i.e., 20% for our data) and  $m$  is chosen to be 100 (suggested by Zadrozny and Elkan [Zadrozny and Elkan, 2001]). Curtailment stops searching down the tree if the current node has less than  $v$  examples and uses the current node to compute probabilities. As discussed and suggested in [Zadrozny and Elkan, 2001], the exact value  $v$  is not critical if  $v$  is chosen small enough. We have used  $v = 100$  in our experiments. Curtailment plus smoothing is a combination of curtailment and smoothing.

## 5 Experiments

Since the problem is time-sensitive in nature, the training set ought to be taken from earlier months while the testing set ought to be taken from remaining later months. Traditional CV is not entirely applicable for this situation. Instead, we use different amount of training and testing data: 50% training- 50% testing, 15% training - 85% testing, as well as 85% training - 15% testing. This will not only give us an idea of the performance on different datasets as well as different amount of data.

Figures 2 to 4 show the “reliability plot” of ensemble-based methods and calibrated decision tree

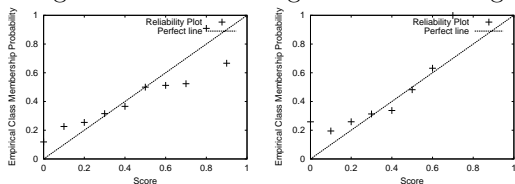
Figure 2: 50% Training and 50% Testing



methods under three combinations of training and testing data sets. “Reliability plot” shows how reliable the “score” of a model is in estimating the empirical probability of an example  $\mathbf{x}$  to be a member of a class  $y$ . To draw a reliability plot, for each unique score value predicted by the model, we count the number of examples ( $N$ ) in the data having this same score, and the number ( $n$ ) among them having class label  $y$ . Then the empirical class membership probability is simply  $\frac{n}{N}$ . Most practical datasets are limited in size; some scores may just cover a few number of examples and the empirical class membership probability can be extremely over or under estimated. To avoid this problem, we normally divide the range of the score into continuous bins and compute the empirical probability for examples falling into each bin. In Figures 2 to 4, the percentage of features sampled by Random Forest+ is 50%, In the extended version of this paper, we have results to sample different percentage of features and 50% appears to be the most accurate.

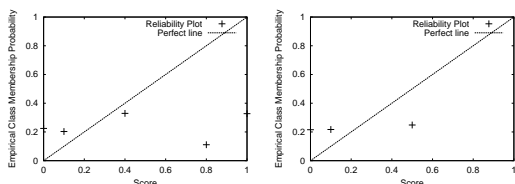
Comparing different methods for different combinations of training and testing data size, random decision tree and random forest+ are the most reliable and stable methods, as shown in the top two plots of Figures 2 to 4. Their probability estimation is insensitive to the amount of training data, i.e., 15%, 50% or 85%. The four calibrated decision tree methods (the bottom four

Figure 3: 15% Training and 85% Testing



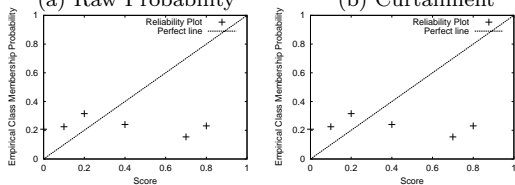
(a) Random Tree

(b) Random Forest+



(a) Raw Probability

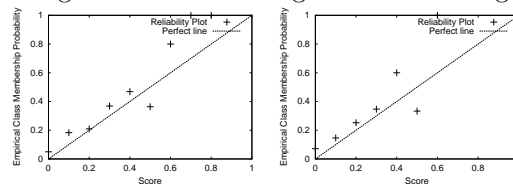
(b) Curtailment



(c) Laplace Smoothing

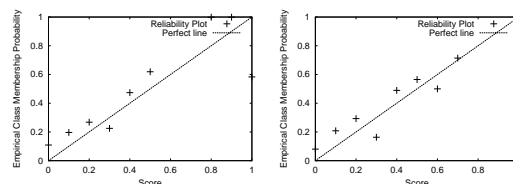
(d) Curtailment + Smoothing

Figure 4: 85% Training and 15% Testing



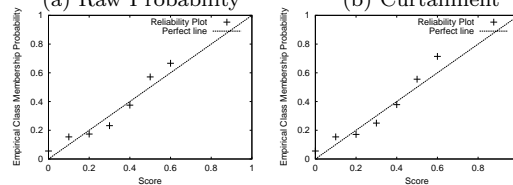
(a) Random Tree

(b) Random Forest+



(a) Raw Probability

(b) Curtailment



(c) Laplace Smoothing

(d) Curtailment + Smoothing

plots in Figures 2 to 4) appear to be very sensitive to the amount of training data. When the training data size is small, 15% in our experiment or approximately 670 data items, all other method except for random decision tree and random forest appears random and do not a clear pattern patterns. However, when the training data increases to 85%, every tested method appear to have a pattern that is well correlated with the perfect line.

## 6 Conclusions

In this paper, we formulated the problem to mine defaulted customers for competitive local exchange carriers or CLEC. We discussed the empirical importance, i.e., managing collection efforts as well as projecting cash flow, of this effort for the survival of these companies. We detailed the complete feature construction process to model the calling, billing and payment history from back office raw data. We also addressed the important problem of how to make a data mining model useful for a business based on the nature of the problem and the actual performance of a mined model. We evaluated many methods and found that the most successful method is to use random decision and random forest plus to estimate the true probability that a customer will default. The probability estimation by both methods have been found to be reliable under two different

customer groups and very different amount of training data. This solution helps the CLEC in two important ways. The predicted score prioritizes the collection effort by the billing company. The product of the due amount by the probability of default gives a good estimate of the cash flow in the future. In the algorithm part of this paper, we find that both random decision tree and random forest plus are reliable and stable in estimating probabilities even when the amount of data is extremely small. However, single decision trees are extremely sensitive to the amount of training data. When the data is small, their probability output are close to random, which prohibits the application of Zadrozny and Elkan's calibration methods.

## References

- [Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- [Fan et al., 2003] Fan, W., Wang, H., Yu, P. S., and Ma, S. (Nov 2003). Is random model better? on its accuracy and efficiency. In *Proceedings of Third IEEE International Conference on Data Mining (ICDM-2003)*, Melbourne, FL.
- [Zadrozny and Elkan, 2001] Zadrozny, B. and Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of Eighteenth International Conference on Machine Learning (ICML'2001)*.