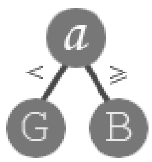

A decision tree approach to predicting business failure

Sam Hallman

Donald Bren School of Information and Computer Sciences, University of California, Irvine

ABSTRACT

Motivation: *<THIS ABSTRACT WILL BE POLISHED…>* The success of our economy is a function of an enormous variety of factors. One important factor is the success of our businesses. Are there special indicators of business failure of which we are not aware? In this paper we run decision tree learning algorithms on a database of over three million businesses to build a model which predicts serious delinquent behavior of a business. We show failed attempts and tolerable attempts and analyze their performance and what went right or wrong.

1 INTRODUCTION

Today's economy is in bad shape and any insight into why would be of considerable value to many people. But to tackle this problem head on, while admirable, is perhaps not the best course of action: the economy is complicated and there are many factors that affect it—wars, natural resources, and so on—and so a better approach might be to work on smaller sub-problems and work our way up.

The problem we consider is that of business behavior. Roughly speaking, we investigate regularities in business behavior to discover indicators of business failure, where "failure" here is defined as a business going bankrupt, being pursued by a collection agency, or being more than 90 days late on a payment.

Knowledge of these failure indicators is useful to businesses and business people as it can help them assess who they should and should not do business with. Indeed, Experian—the credit information organization which provided the data on which we did our analysis—bases their entire business on discovering and utilizing such information for computerized risk management systems and has grown into a successful company whose annual sales reach over $4 billion.[1]

Experian provided us with eight quarters of aggregate data on over three million businesses. Each record was a collection of attributes of a business sampled quarterly from January 2007 to October 2008. Our goal was to build a computerized model which, given four quarters worth of those attributes for a given business, will predict whether or not that business will fail within the next twelve months (with "failure" as defined above).

My team, made up of me and three other students, split the job into three parts: perform data mining with a neural network, a naïve Bayes classifier, and a decision tree. I worked on the decision tree part alone.

The decision tree algorithm thus far has not worked out quite well. In this paper I show my strategy, my results, and attempt to explain why the outcome was not quite what I had hoped and what I think would give better results.

2 METHODS

First I will describe the format of the training data, state precisely what it is I set out to do, and then describe what I did.

2.1 The Data

As mentioned in the introduction, Experian provided us with eight quarters of aggregate data on over three million businesses. For any given business, let a_ij be the value of attribute j for that business during quarter i. The samples were taken beginning in January 2007 and ending in October 2008, so i runs from 1 to 8, with i = 1 corresponding to 1/07 and i = 8 corresponding to 10/08. There are twenty attributes in total, so j runs from 1 to 20. So, every record is of the form

$$a_{1,1}, \dots, a_{1,20}, \dots, a_{8,1}, \dots, a_{8,20}$$

Every record corresponds to one business (unknown to us). Experian's data was a CSV file of over three million such records.

Attributes $a_{i,2}, a_{i,4}, a_{i,7}$ represent filing for bankruptcy, begin pursued by a collection agency, and being more than 90 days late on a payment, respectively, during quarter i. I do not disclose the meanings of any other of the twenty attributes in this paper as that information belongs to Experian.

To make conversation easier we call any business with $a_{i,2} + a_{i,4} + a_{i,7} > 0$ "bad" and "good" otherwise.

2.2 The Goal

[1] http://www.experianplc.com/corporate/about/

Our goal was to build a system which predicted whether or not a business which was initially “good” would go “bad” within the next four quarters, given four quarters worth of attributes on that business. More formally, given a list $a_{1,1}, \dots, a_{4,20}$, the system is to predict whether $a_{i,2} + a_{i,4} + a_{i,7} > 0$ for any $i \in \{5,6,7,8\}$.

2.3 Approach

Throughout the quarter I made exclusive use of decision tree learning, though I did not write my own decision tree program. Instead, I used the C4.5 software developed by Ross Quinlan.[2] I used this software in a variety of ways, but at a high level every approach I used fell into one of two categories:

(1) In the first class of approaches I made *no use* of the fact that the our data was time series data. In other words, I did not make use of the fact that attributes $a_{i,k}$ and $a_{j,k}$ were actually the *same* attributes only sampled at different points in time. In the end, this approach was not very successful.

(2) In the second class of approaches I did make use of the fact that the data was time series data. <HAVEN’T IMPLEMENTED/CODED THIS PART YET>

Note that, as discussed in Section 2.2, no decision tree was trained on any examples which were already bad in 2007. Next I discuss each part in turn.

2.3.1 Category 1 approaches Here I tried two different methods:

- In the first method, I built three decision trees. The first decision tree predicted whether or not the business would file for bankruptcy; specifically, it predicted whether or not $a_{5,2} + a_{6,2} + a_{7,2} + a_{8,2} > 0$. The second tree predicted whether or not the business would be pursued by collection agencies; specifically, it predicted whether or not $a_{5,4} + a_{6,4} + a_{7,4} + a_{8,4} > 0$. Finally, the third decision tree predicted whether or not the business would be more than 90 days late on a payment; specifically, it predicted whether or not $a_{5,7} + a_{6,7} + a_{7,7} + a_{8,7} > 0$. Then for prediction I would run all three decision trees and ‘OR’ their results: if *any one of* them output “bad,” then the business is predicted to be “bad,” otherwise it is “good.”
- In the second method, I build just one decision tree that predicts “good” or “bad.”

2.3.2 Category 2 approaches Here I tried three different methods:

- In the first method, I averaged corresponding attributes together to create new, non-time-series attributes and trained on those. That is, for every $1 \le i \le 20$ I replaced the group of attributes $a_{1,i}, a_{2,i}, a_{3,i}, a_{4,i}$ with their average, $\bar{a}_i$ and therefore ended up with a new training set for which every training example was a list of the form $\bar{a}_1, \bar{a}_2, \dots, \bar{a}_{20}$.
- In the second method, I fit a line to each group of corresponding attributes and trained a decision tree on their slopes. That is, for every $1 \le i \le 20$, I fit a line to the data $\{(1, a_{1,i}), (2, a_{2,i}), (3, a_{3,i}), (4, a_{4,i})\}$ obtaining a slope and intercept α_i, β_i and replaced those four attributes with α_i. Thus I end up with a new training set for which every training example was a list of the form $\alpha_1, \alpha_2, \dots, \alpha_{20}$.
- Next I used the fitted lines from the previous method but trained on their intercepts rather than their slopes. So the new training set contained training examples of the form $\beta_1, \beta_2, \dots, \beta_{20}$.

3 RESULTS

First I will describe my results for the non-time-series approaches (see section 2.3.1 above) and then I will move onto the time series approaches (2.3.2), giving slight analysis of the results along the way but leaving the more broad analysis for the ANALYSIS section below (section 4).

3.1 Non-time-series approaches

3.1.1 Three-tree approach The bankruptcy predictor tree did not perform well because of the extreme dearth of examples of businesses which were bankrupt in 2008 but not in 2007. There were roughly 50 examples of businesses which went bankrupt in 2008, compared with roughly 50,000 examples of good businesses. Such a small number of examples are simply not enough for the tree to learn the structure of negative examples. Add to that the fact that there were 20×4=80 features, and the situation becomes hopeless. The result was that, in the end, the decision tree was just one long path with a “bad” leaf at the tip. From a glance it was apparent that this tree failed to capture any meaningful structure, but with one last glimmer of hope I checked every negative example in order to find if any of them exhibited the supposed negative pattern and in the end almost zero of them did.

To help prevent overfitting, C4.5 outputs a second, pruned decision tree. However, again due to the scarcity of negative training examples, the pruned tree was pruned all the way down to its root, thus classifying every business as “good” no matter what the business’s attributes. Of course, this yielded near 100% accuracy but only because the test set was nearly 100% “good.”

Though the other two decision trees’ training sets were not nearly as imbalanced, the three-tree approach simply could not work when one of its three trees is so dysfunctional and so the idea was abandoned.

[2] http://www.rulequest.com/Personal/

Put in here a pretty picture of the decision tree.

Fig. 1. The top portion of the decision tree resulting from algorithm 3.1.2.

3.1.2 Single-tree approach The lone decision tree approach fared better than the three-tree approach described above since there was a better balance of bad businesses to good ones. However, though there was balance relative to the previous approach, the positive examples still far outnumbered the negative ones <FILL IN PRECISE RATIO>. As a result, in the end the results were still roughly equivalent to what would result from simply guessing "good" for all examples.

3.2 Time-series approaches

<HAVEN'T IMPLEMENTED ANY OF THESE YET>

4 ANALYSIS

The disappointing results of the first few approaches it seems were to some degree due to the model working on the time series data as if they were not time series data. It was my gut feeling during the development period that this approach was wrong and that for such data you want to observe trends over time (i.e., *time wise* behavior), not just any one particular feature in isolation at a time. (Though I believe this could really depend on the problem domain.) For example, it seems unlikely that predicting future delinquent business behavior would be a matter of observing whether the jth attribute over the third quarter was above some threshold, etc. Though I have not proved anything, I believe that this is likely to be the case.

The other major reason for poor behavior I believe was the very large dimensionality of the model. Without shrinking the dimensionality as was done in 2.3.2, there were 80 attributes. The size of the search space increases exponentially with the dimensionality of the model, a phenomenon termed the curse of dimensionality. [2] even states that researchers have estimated that the number of training samples must increase *exponentially* with the dimensionality in order to "have an effective estimate of multivariate densities."

Furthermore the number of possible decision trees for any given problem of any more than six dimensions is absolutely massive, resulting in a high likelihood that there are trees which match your training data well but which do not force any kind of meaningful generalization. In fact, if there are no two examples which have the same attributes but different classes, then it is easy to fit the training data perfectly.

Possible inappropriate structure?

<REPLACE THIS TABLE WITH ONE IN THIS SAME STYLE THAT SHOWS THE ACCURACY OF THE DECISION TREE AS A FUNCTION OF THE TRAINING SET SIZE SO AS TO SHOW VIA A TABLE THE LEARNING CURVE… OR PERHAPS DO A GRAPH INSTEAD>

Table 2. Results of the decision tree learner as a function of training set size

|S|	Predicted cost	Timing	Predicted speed	Speed
100	S219.20(100%)	68m43s	1.00	1.00
1000	2^9.10+2^{19}.10(~50%)	35m13s	2.00	1.95
10,000	2^{19}.20(100%)	68m43s	1.00	1.00
100,000	2^9.10+2^{19}.10(~50%)	35m13s	2.00	1.95
1,000,000	2^{19}.20(100%)	68m43s	1.00	9.5

This is table foot note sample text This is table foot note sample text This is table foot note sample text

ACKNOWLEDGEMENTS

I would like to thank my teammates for their support and especially for helping me understand what went on during the weekly conferences since I would never understand anything related to the business/finance side of the project. Also, I sincerely thank Sung Park for giving us the opportunity to work on a *real* data set on a *real* industry problem, an invaluable opportunity to any university student. And thanks to Professor Lathrop for giving me so much advice during the weekly meetings. I am also grateful to Ross Quinlan for providing his decision tree software free of charge as this saved me a lot of time. Finally, thank you to Experian for the terrific opportunity.

REFERENCES

[1] Bishop M. (2007) *Pattern Recognition and Machine Learning.* Springer.

[2] Rokach L. & Maimon O. (2008). *Data Mining with Decision Trees: Theory and Applications.* World Scientific Publishing Company.

[3] J. Ross Quinlan (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann.

[4] Bishop M. (1995). *Neural Networks for Pattern Recognition: Theoretical Foundations*. Cambridge University Press.

[5] Witten I. H. & Frank E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (2nd ed.). Morgan Kaufmann Publishers.

[6] Gonzalez R. C. & Woods R. E. (1992). Decision-Theoretic Methods, *Digital Image Processing*. Addison-Wesley Publishing Company, Inc.

[7] Stuart R. & Norvig P. (2003). *Artificial Intelligence: A Modern Approach* (2nd ed.). Prentice Hall.