Predicting Takeover Success Using Machine Learning Techniques

Mei Zhang, Ph.D., Rowan University, USA
Gregory Johnson, Temple University, USA
Jia Wang, Ph.D., Rowan University, USA

ABSTRACT

A takeover success prediction model aims at predicting the probability that a takeover attempt will succeed by using publicly available information at the time of the announcement. We perform a thorough study using machine learning techniques to predict takeover success. Specifically, we model takeover success prediction as a binary classification problem, which has been widely studied in the machine learning community. Motivated by the recent advance in machine learning, we empirically evaluate and analyze many state-of-the-art classifiers, including logistic regression, artificial neural network, support vector machines with different kernels, decision trees, random forest, and Adaboost. The experiments validate the effectiveness of applying machine learning in takeover success prediction, and we found that the support vector machine with linear kernel and the Adaboost with stump weak classifiers perform the best for the task. The result is consistent with the general observations of these two approaches.

Keywords: Predicting Takeover Success; Machine Learning Techniques

INTRODUCTION

A takeover can be defined as the acquisition or control of one company by another or occasionally by an individual or group of investors. Takeovers are usually established by purchasing shares at a premium over existing prices. They can be financed in several ways, including a cash payment or using shares of the acquiring company. It can be complete or partial and may or may not involve merging the operations of the acquired and acquiring firms.

Empirically, approximately ten percent of the announced takeover attempts fail. Either the acquiring company withdraws or the target company rebuffs the offer. A takeover success prediction model attempts to use information that is publicly available at the time of the announcement in order to predict the probability that the takeover attempt will succeed. Such a model can help investors predict the outcome of a takeover attempt and is especially of interest to merger arbitraguers.

The purpose of this paper is to compare the predictive performance of several machine learning algorithms for takeover success prediction, including the traditional logistic regression model, artificial neural networks, support vector machines with different kernels, decision trees, random forest and Adaboost. Logistic regression is the most commonly used technique in the literature. Branch et. al. (2008) used the artificial neural networks model to predict the takeover success and their result outperformed the traditional logistic regression model. Compared with classical models such as the logistic regression and neural networks, modern machine learning methods such as support vector machines and Adaboost often achieve better performance in terms of accuracy as well as generalization ability.

Logistic regression, invented in the 19th century for the description of the growth of populations and the course of chemical reactions, predicts the probability of an occurrence of an event by fitting data to a logistic curve. The logistic function used in this prediction method is useful in that it can take any value from negative infinity to positive infinity as input.

There are two types of logistic regression. Simple logistic regression is used when the data consists of only one attribute, or independent variable, and one target variable, or dependent variable. This method is comparable to linear regression, except that with simple logistic regression, the variable is nominal and not some measurement. Multiple logistic regression is used when there is more than one independent variable to be analyzed. Logistic regression is different from linear regression in the sense that, unlike linear regression, the target variable itself isn't predicted. Instead, the algorithm predicts the probability of obtaining a certain value for the target variable.

The structure of the neural networks algorithm is derived from biological neural networks in neuroscience. It is consisted of a group of artificial neurons that are used to model potentially complex relationships between inputs and outputs or to even find patterns within a dataset. These networks, sometimes called artificial neural networks, learn by example, so it is configured for a specific application through a learning process.

Neural networks have three groups of units. The input group, or layer, represents raw data that is put into the network. This input layer is connected to what is called a hidden layer, which is then connected to either another hidden layer or finally the output layer. The weights between the input and hidden units determine when each hidden unit is active.

Feed-forward networks allow signals to travel only from input to output, not the other way around. Therefore, there is no case where the output of a layer can affect that same layer. These types of networks are generally used in pattern recognition. Feedback networks are generally more complicated than feed-forward, but are more powerful. Also known as interactive or recurrent, these networks allow signals to travel in both directions and uses loops in the network.

The neural network used in this analysis was the multilayer perceptron. It is consisted of multiple layers of nodes in a directed graph with each layer fully connected to the next layer. It uses back propagation to classify instances. The network can be built by hand, created by an algorithm, or both. It can also be monitored and modified during training time.

Support vector machines (SVM) treat the classification problem as finding the separation hyper plane with the maximum margin in the high dimensional kernel space. The kernel space is mapped from the original relatively low dimensional feature space implicitly through a kernel function. It has been shown that the maximum margin strategy effectively reduces error bound of the Bayesian classification error.

In this analysis, four different kernels were used for support vector machines. The linear kernel is the simplest kernel and generally performs well for data that is linearly separable. With Polynomial kernels, a kernel function of a number order can be used to transform vectors that are linearly dependent on that number of dimensions, into linearly independent vectors. The Radial Basis Function (RBF) is a function where only the distance from the origin determines the value of the function. A sigmoid function is similar to the logistic function, created to generate some non-linearity between the input and output of the function.

The goal of a decision tree model is to predict the value of the target variable based on several input variables. The nodes of a decision tree describe different attributes of the data. The branches that come from each node tell the possible values for that corresponding attribute. The terminal nodes at the bottom of a tree say the predicted value of the target variable.

Decision trees are of two main types: classification and regression. Regression tree analysis is when the predicted outcome is a real number. Examples of this include median income and height. Classification tree analysis is when the predicted outcome is a possible class outcome of the target variable. In our case, we are using classification tree analysis, with the possible outcomes being whether or not the takeover attempt was successful.

One type of Decision Tree is the J48 Algorithm. In this case, in order to classify a new item, it needs to create a decision tree based on the attribute values of the training data. Whenever it encounters a set of items it identifies the attribute that discriminates the various instances most clearly. This feature that is able to tell us most about the data is said to have the highest information gain. Among the possible values of this feature, if there is any value for which the data instances falling within its category have the same value for the target variable, then the algorithm terminates that branch and assigns to it the target value that is obtained. If this is not the case, the algorithm looks for another attribute that gives the highest information gain. The algorithm continues in this manner until either there is a clear decision of what combination of attributes gives a particular target value or all attributes have been used. If the algorithm runs out of attributes, or cannot deduct a clear result from what is available, the target value is based on the majority of the items that would be under that specific branch.

Another Decision Tree used in this analysis is the fast learner REPTree. It builds a decision regression tree using information gain and reduces it using error-pruning. It only sorts values for numeric attributes once and missing values are dealt with by splitting corresponding instances into pieces.

The Decision Stump algorithm builds binary decision "stumps" for classification problems. It is essentially a decision tree with one node. This algorithm makes a prediction based on the value of just one feature in the data. For nominal features, a stump is usually built either with a leaf for each possible feature value or with two leaves, one corresponding to a chosen category and the other to all remaining categories. This could even work with missing values, with those being considered as a separate category. For continuous features, a threshold is normally established and one leaf will be for values less than the threshold and one leaf will be for values greater.

The Decision Stump is commonly used with a boosting algorithm, such as Adaboost. The Adaboost, originally proposed by Freund and Schapire (1997), used an additive model to combine sets of weak classifiers to achieve strong discriminative power. It has shown that the method is robust to overfitting and also very flexible in feature selection.

The Random Forest algorithm, developed by Leo Breiman and Adele Cutler, is considered an ensemble classifier, in the sense that it typically consists of several decision trees and uses them to come to a consensus for a prediction.

DATA

We used the dataset in Branch et. al. (2008). The dataset was a sample of both successful and failed takeover attempts for the 1991-2004 period using Securities Data Company's database. The final sample includes 1196 takeover offers with 146 failed takeovers and 1050 successful takeovers. There were ten variables available to predict takeover success: target size, target leverage, target book-to-market ratio, target resistance, arbitrage spread, deal structure, termination fees for the target, termination fees for the acquirer, poison pills and bid premium. The variables used in testing of the prediction algorithms were the target size, post price, transaction size bid premium, and debt to asset ratio, in addition to the binary variables corresponding to attitude, stock swap options, and collar.

EXPERIMENTS AND RESULTS

The software package Weka was used to implement the multiple prediction algorithms. Weka, standing for Waikato Environment for Knowledge Analysis, is a collection of machine learning algorithms for data mining purposes. Weka was able to import the data and determine the different parameters for each algorithm.

For a fair evaluation and to avoid randomness, we used ten-fold cross validation in the experiments. Specifically, the dataset was divided into 10 equal subsets. Then, in each run, one subset was chosen as the testing set and the remaining is used for the training set. We then recorded the average performance over the 10 runs. We evaluated the accuracy for the positive samples and negative samples separately, as well as the prediction rate over the entire dataset. In our study, we evaluated the SVM with several different kernels including the radial basis function, linear and polynomial, and sigmoid kernel. For Adaboost, used the standard decision stumps as weak classifiers, i.e., binary thresholding. We used 100 weak classifiers.

The results are summarized in Table 1. From the table, we see that the Support Vector Machines with linear kernel achieves the best performance, followed by Adaboost, which validate our motivation. While the positive examples performed just as well with the SVM with linear kernel as other algorithms, the major difference came when predicting the negative examples. The SVM with polynomial kernel performed exceptionally well in the positive examples but particularly poor in the negative examples. With only a few exceptions, this algorithm classified nearly all examples as a success.

Table 1
Comparison of different machine learning models for takeover success prediction

Algorithm	Positive	Negative	Total
Logistic Regression	.9707	.5036	.9102
Neural Networks (Multilayer Perceptron)	.9674	.5182	.9092
SVM – RBF Kernel	.9631	.5255	.9064
SVM – Linear Kernel	.9707	.5474	.9159
SVM – Polynomial Kernel	.9967	.1241	.8837
SVM – Sigmoid Kernel	.9631	.2554	.8715
REPTree	.9739	.4964	.9121
Decision Tree (J48)	.9783	.4672	.9121
Random Forest	.9739	.4745	.9093
AdaBoost (Decision Stump)	.9739	.5036	.9130

CONCLUSIONS AND FUTURE WORK

In this paper, we perform a thorough study using machine learning techniques to predict takeover success. Specifically, we model takeover success prediction as a binary classification problem, which has been widely studied in the machine learning community. Motivated by the recent advance in machine learning, we empirically evaluate and analyze many state-of-the-art classifiers, including logistic regression, artificial neural network, support vector machines with different kernels, decision trees, random forest and Adaboost. We found that support vector machines with linear kernel and the Adaboost with stump weak classifiers perform the best for the task.

Future studies include analyzing the effect that other factors of the takeover attempts have over the success rate. Depending on availability of data, these factors can include date of attempt as well as countries of origin of the companies involved. The probability of takeover success may be lower or higher based on the strength of the economy at the time of the attempt, and that strength varies based on countries as well as time periods.

AUTHOR INFORMATION

Dr. Mei Zhang received B.S. and M.S. from Tsinghua University in 1998 and 2001, respectively, and Ph.D. from the University of Maryland in Accounting in 2008. From 2008 to 2009, she was an assistant professor at Montclair University in New Jersey. Since fall 2009, she has been an assistant professor at Rowan University. Dr. Zhang's research interests include financial reporting, valuation and auditing issues. E-mail: zhangm@rowan.edu (Corresponding author)

Gregory Johnson is a Master of Science graduate from Temple University in the Computer and Information Science Department. He is the webmaster and mentor in the MARC program in Temple's Biology Department. E-mail: gjohns5@temple.edu

Jia Wang is an Associate Professor of Finance at Rowan University. She holds a B.S. in Accounting from Tsinghua University, China, an M.S. in Statistics and a Ph.D. in Finance from the University of Massachusetts, Amherst. Her research interests include investments, event studies and empirical methods in finance. She has published in a number of journals including *Journal of Banking and Finance, Journal of Alternative Investments, International Review of Financial Analysis, Financial Services Review, Journal of Investing*, among others. E-mail: wangji@rowan.edu

REFERENCES

1. Branch, B. , Wang, J., Yang, T. (2008) A note on takeover success prediction. In *International Review of Financial Analysis*, *17*, 1186-1193.

2. Cramer, J.S. "The Origins of Logistic Regression,"Tinbergen Institute Discussion Papers 02-119/4, Tinbergen Institute. (2002).
3. Freund, Y. and Schapire, R. E., "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Comput. Syst. Sci*, 55(1): 119-139 (1997).
4. Glossary of Industrial Organisation Economics and Competition Law, compiled by R. S. Khemani and D. M. Shapiro, commissioned by the Directorate for Financial, Fiscal and Enterprise Affairs, OECD, 1993.
5. H.-T. Lin and C.-J. Lin. "A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods." Technical report, Department of Computer Science and Information Engineering, National Taiwan University, (2003).
6. Hoffmeister, J., & Dyl, E. (1980). Predicting outcomes of cash tender offers. *Financial Management, 9*, 50−58.
7. Palepu, K. (1986) Predicting Takeover Targets. In *Journal of Accounting and Economics,* 3-35.
8. Samuelson, W., & Rosenthal, L. (1986). Price movements as indicators of tender offer success. *The Journal of Finance, 41*, 481−510.
9. Walkling, R. (1985). Predicting tender offer success: A logistic analysis. *Journal of Financial and Quantitative Analysis, 20*, 461−478.

NOTES