Predicting Bank Default from the Quarterly Report

Caiyao Mai and Sunyoung Baek

cymai@stanford.com sbaek@stanford.edu

Abstract

Possibility of a bank default in the quarter after the lastest financial reporting can be estimated from the bank's balance sheet in relation to the general economic situation of the time. Although creating the exact formula is not at all clear, it is possible to construct an algorithm that produces little error in categorical default prediction given bank data and economic indices. This paper seeks to suggest a machine-learning-based-approach to predicting bank default based on the publicly available information.

I. Introduction

During the financial crisis of 2008, the world has seen that solvency of financial institutions must be tightly regulated by the federal government. It is to ensure that no taxpayers' money is to be squandered to cover up the bank's poor financial management. Like in the Lehman's case, banks' financial problems are so well hidden that their default is difficult to predict without insider knowledge. Sudden cascades of default shock shareholders, call for a series of bailouts with taxpayer's money, and cause far-reaching impacts on both large and small businesses. Recent financial crisis has highlighted the importance of estimating each bank's susceptibility to economic recession and forcing the banks that are likely to default to address their balance in advance. As a first step, we aim at creating a machine learning algorithm to predict whether a bank would default in the following quarter based on the latest financial reports and economic indices. This algorithm will be very useful if we can lengthen the period of prediction from a quarter ahead to a year, or even a few years ahead. For the moment, this paper focuses on the quarter-long prediction only.

In section 2, we begin by processing data as preliminary steps. In section 3, we outline our cross-validation scheme and the models that we will test. In section 4, we do feature selection using different methods. In section 5, we present the test errors for each model, choose the optimal hypothesis, and obtain the hypothesis error by running it on the entire dataset. In section 6, We conclude the paper with a note on the further study.

II. Preliminary Steps

We have 1055 data points, each of which contains quarterly financial report data from a period within 2002 - 2011 and economic indices of the corresponding quarter. Each data point is matched with Y which is 1 if the bank defaulted in the quarter following the latest reporting (from which the information in X variable had come from) and 0 if not. Some of the data points are from the the same bank but in a different reporting period. Economic indices that we use in the prediction are published every quarter. Rows in our final design matrix X indicate different data points, and columns indicate different variables. In the entire dataset, we have 78 defaults in total which means our default rate is 0.0739.

II.1 Choose a set of variables to begin with

We are originally given 25 variables but we take out the ones that are obviously redundant. For instance, we have both eq and eqpct, which

mean total equity and total equity divided by total assets, respectively. In this case, we choose to include eqpct only because we prefer to represent each data point in relative terms instead of absolute terms. In the end, we have the following 9 bank variables and 6 econmic indices for each reporting period (columns in X)

- deppct (PS: not dep): Deposits Percentage
- eqpct: Equity Percentage
- rbc1rwaj: Tier 1 Risk-Based Capital Ratio
- rbcrwaj: Total Risk-Based Capital Ratio
- p3assetpct: Percentage of Assets 30 to 89 days pass due
- nclnlspct: Percentage of Noncurrent Loans and Leases
- scpct: Investment Securities Percentage
- voliabpct: Volatile Liabilities Percentage
- roa: Return on Assets

- LIBOR1M: London Interbank 1 Month Borrowing Rate
- LIBOR3M: London Interbank 3 Month Borrowing Rate
- LIBOR6M: London Interbank 6 Month Borrowing Rate
- CSW: Case Shiller Home Price Index
- CPI: Consumer Price Index published by U.S. Bureau of Labor Statistics
- Unemploymt: Monthly National Unemployment Rate published by U.S. Bureau of Labor Statistics

III. Method and Strategy

III.1 Algorithm Outline

Using the logistic regression model, we run cross-validation for each feature set and obtain the optimal variable set. We assume that the selected features are inherently significant so that the choice of the optimal set does not depend on whether we use SVM or logistic regression (though the resulting error might vary). We have the following algorithm:

```
for each feature set {
    K-fold Cross Validate {
        divide X to train set and test set
        obtain optimal hypothesis in the model
        based on the train set
        compute error on test set
    }
}
```

After getting the best feature set, we are testing which model to use.

```
for each model {
    run cross validation
    return cross-validated error
}
```

Test optimal model with the optimal variables in the entire data points to obtain optimal hypothesis and the hypothesis error.

III.2 Models

In the beginning, we considered the three models: Simple Logistic Regression, Support Vector Machine with five different Kernels, and Naive Bayes. However, when we tested each model on the entire dataset to get a rough idea, the error we obtained from the Naive Bayes model was over 60%. So we decided not to include Naive Bayes in the further research process. We test the Logistic Regression and the Support Vector Machine in in the following sections. For the Support Vector Machine, we consider linear, quadratic, polynomial, Gaussian radial basis function, and multilayer perceptron kernels. In this case, we return the optimal Kernel along with cross-validated error to be run on the entire data points.

IV. Feature Selection

For feature selection, we have two approaches: the first approach is (1) to use principal component analysis (PCA) to reduce feature dimensions and then (2) run feature selection algorithm. In this case, we add one more method of feature selection by choosing the first n prin-

ciple components; the second approach is to run feature selection directly on the original dataset. For both approches, we use three methods to evaluate the features: (a) MI Scoring (b) forward search and (c) backward search. In every case, model errors in the feature selection are obtained by using 10-fold cross validation with the linear regression model.

IV.1 Principal Component Analysis

This method applies only to the PCA-Feature Selection approach: After converting 15 feature vectors to 15 principle components, we run logistic regression cross-validation on first to nth principle components (thus, n = 1, 2, ... , 15). We choose the n for which model error is the lowest. We obtained 10 principle components as a result.

IV.2 MI Scoring

We use mutual information to see the correlation between each feature and y. Since our features are continuous variables, we assume that our features are jointly Gaussian distributed.

Firstly, we rearrange our data to be three sets: the original data, the data with label $y = 0$ and the data with label $y = 1$. Secondly, we use Matlab library function to fit each feature to the Gaussian distribution for each sets and got the corresponding mean and covariance for each Gaussian distribution. This means we got $p(x)$, $p(x|y = 0)$ and $p(x|y = 1)$. And we change the formula for the Mutual Information to be:

$$MI(x_i, y) = \int_{-\infty}^{\infty} \sum_{y \in \{0,1\}} p(x_i|y)p(y) log \frac{p(x_i|y)p(y)}{p(x_i)p(y)}$$

We do MI-scoring for principle components set and original dataset separately. We choose the top k features which minimize the error rate.

Table 1: *MI Scoring*

Data	Number of Features	Error
PCA	9	0.0209
Original	8	0.0227

The result shows that the algorithm chose the first 8 principal components except for the 7th. For original dataset, the model chooses the features with the following index: 1, 2, 3, 4, 8, 9, 10, 15.

IV.3 Forward Search and Backward Search

From the forward and backward search, we obtain the following result:

Table 2: *Forward Seach*

Data	Number of Features	Error
PCA	8	0.0209
Original	6	0.0218

For PCA, the result shows that the first 10 principal components except for the 7th and 8th are chosen to be included. For original dataset, the 2nd, 3rd, 4th, 5th, 6th and 10th features are picked up.

Table 3: *Backward Seach*

Data	Number of Features	Error
PCA	8	0.0208
Original	11	0.0220

For PCA, the first 8 components except for the 7th are chosen. For original data, the model chooses features with the following index: 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 15.

IV.4 Decision

From the above analysis, we can see for each feature selection approach, we got a better result from the PCA. The error rates for three methods are very close because the three feature sets are very similar. So we run the cross-validation for each feature sets and compare them with the precision and recall rate. We got the following table:

Table 4: *Comparing PCA Features*

Feature Set	Precision	Recall
X_{MI}	0.8660	0.8625
$X_{Forward}$	0.8858	0.8542
$X_{Backward}$	0.8918	0.8476

For predicting bank default, we want to obtain the best precision because we want to minimize the false alarms as much as possible. Since much of a bank's business is based on its client's trust on its security and stability, false alarm might be a self-fulfulling prophesy which jeopardizes financial world even more. For this reason, we pick the PCA features obtained by backward search as our feature set.

V. Model Selection

We run Logistic Regression and Support Vector Machine models and obtain the following error rate, precision and recall rate:

Table 5: *Simple Logistic Regeression Error*

Generization Error	CV Error
0.0190	0.0208

Table 6: *SLR Precision and Recall*

Test Set	Precision	Recall
Generization	0.8701	0.8590
Cross Validation	0.8918	0.8476

Table 7: *Support Vector Machine Generization*

Kernel	Error	Precision	Recall
Linear	0.0597	0.5532	1
Quadratic	0.0360	0.6724	1
Polynomial	0.0104	0.8764	1
Gaussian	0.0332	0.6903	1
Perception	0.2275	0.2128	0.7692

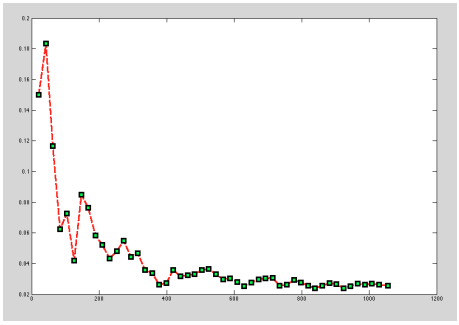
Table 8: *Support Vector Machine Cross Validation*

Kernel	Error	Precision	Recall
Linear	0.0577	0.5646	1
Quadratic	0.0483	0.6349	0.9416
Polynomial	0.0407	0.7221	0.7903
Gaussian	0.0625	0.5543	0.9041
Perception	0.2358	0.2043	0.7632

Based on the result, we decided that the logistic regression creates a model balanced and stable error, precision, and recall rate. According to the simple logistic regression result, we have both low generalization and cross-validation error, and precision and recall rates are stable around high 80 percent. Polynomial kernel in the SVM, on the other hand, have a very good generalization result but less promising cross-validation result. We suspect that optimal number of features might be different for SVM model (since its selection was based on logstic regression model) and we might be doing overfitting in polynomial kernel's case.

VI. Result and Analysis

We got the following learning curve for our variable set and the linear regression model.

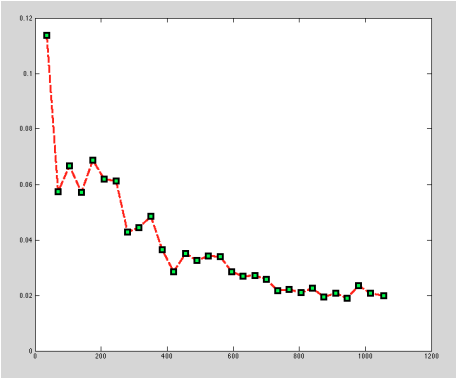


General

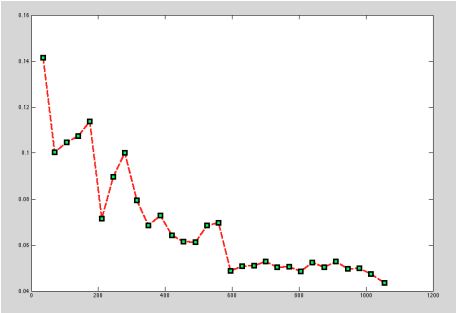
And the error rate, precision and recall rate for both generization and cross validation of the whole data set are in Table 5 and Table 6.

VI.1 Learning Rate

The followings are the learning plots for Logistic Regression and SVM. We plotted cross-validation error result for a randomly chosen sample dataset with size n. n is indenpendent variable in the below plots. Since our default rate is low, our cross-validation is subject to random effects that creates variance in error rate. However, in general we can see that error rate decreases. Note that th error rate doesn't improve much as n goes close to 1000 and that SVM learning curve is not much better than Logistic Regression's. For the given dataset, we might need a different kernel function to have effective SVM result. Furthermore, since the feature selection process used logistic regression model, it might be the case that the optimal set is different for SVM especially if we use higher-dimensional kernel. If time allows, we should repeart the feature selection process for SVMs with different kernel and see how the result goes. Also, we might want to change the data format to have more effective learning curve.



Linear Regression



Supporter Vector Machine

VI.2 False Prediction

After selecting the optimal features and optimal model, we analyzed the data points for which our model made a false prediction. We observed the following two types of mistakes:

(1) false positive for the quarter prior to the bank's default

(2) false negative for the first quarter of bank default

The problem is due to the fact that we have multiple data points from one bank. A series of reports from one bank are closely related. Bank balance shows ominous signs when its default comes close even though the response variable Y does not catch that. Their balance in the defaulting quarter and a quarter prior to the default are very similar, and our algorithm often makes mistake classifying them.

VI.3 Default rate

Our dataset has a very low default rate. Gathering default data is inherently difficult because one bank defaults in one quarter only. Since we take take multiple data points from non-defaulting periods but end up with just one default quarter, we cannot gather comparable number of dataset for each. Also, majority of banks do not default at all. Because we have a very low default rate, our cross-validation result varies widely sometimes, having no default at all in the k-fold and uniformly predicting non-default to get a zero error rate. By changing the data structure, we might be able to form a dataset with higher default ratio.

References

[Ng, Andrew] Lecture notes of CS229 (2012).