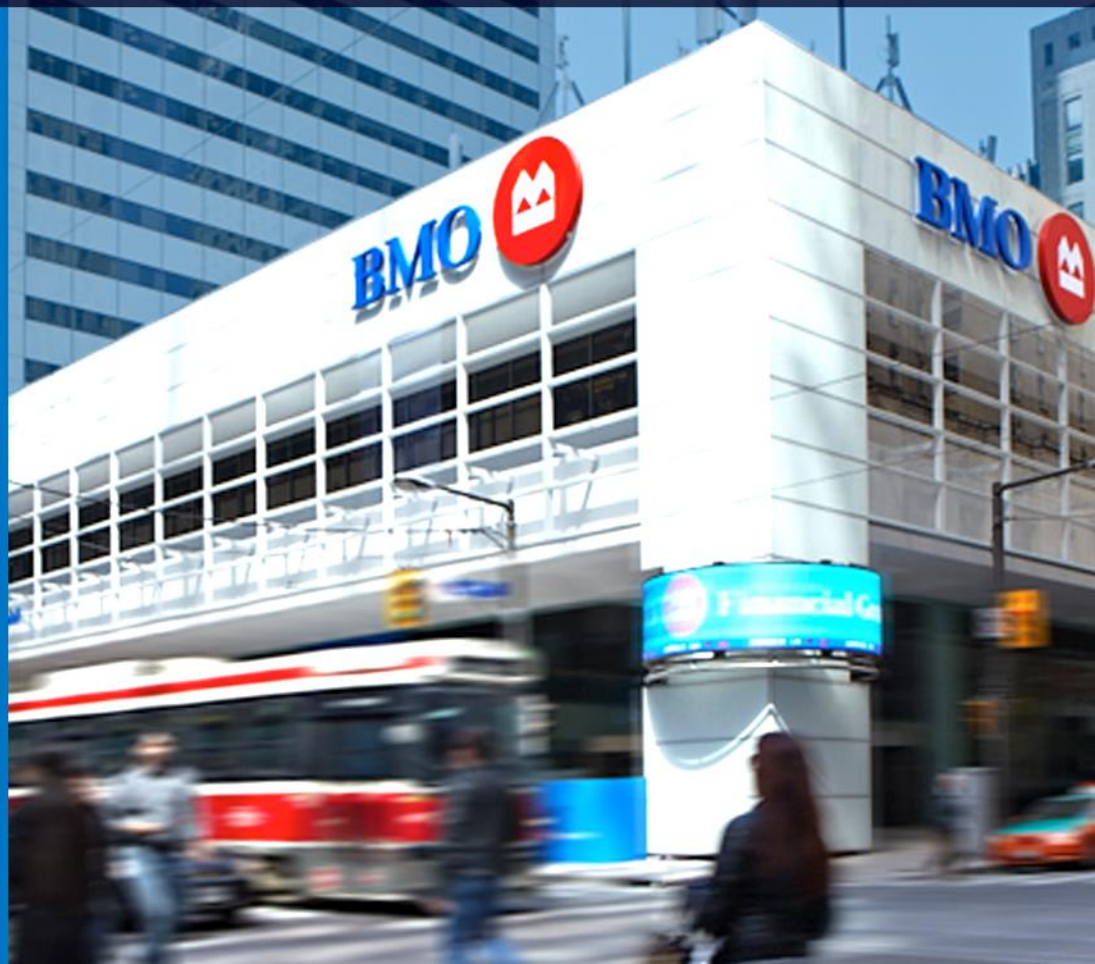
An Introduction to Ensemble Learning in Credit Risk Modelling

October 15, 2014

Han Sheng Sun, BMO
Zi Jin, Wells Fargo



Disclaimer

“The opinions expressed in this presentation and on the following slides are solely those of the presenters, and are not necessarily those of the employers of the presenters (BMO or Wells Fargo). The methods presented are not necessarily in use at BMO or Wells Fargo, and the employers of the presenters do not guarantee the accuracy or reliability of the information provided herein.”

Outline

- Introduction
  - Credit Risk Modelling
  - Ensemble Modelling
- From CART to Random Forest
- Why Random Forest works
- A new algorithm
- Experimental Results
- A note on SAS implementation

Credit Risk

- Credit Risk refers to the risk that a borrower will default on any type of debt by failing to make required payments.

- The risk is primarily that of the lender and includes lost principal and interest, disruption to cash flows, and increased collection costs. For example:
  - A consumer may fail to make a payment due on a mortgage loan, credit card, line of credit, or other loans
  - A company is unable to repay asset-secured fixed or floating charge debt
  - An insolvent insurance company does not pay a policy obligation

- Three key credit risk parameters:
  - probability of default (PD)
  - loss given default (LGD)
  - exposure at default (EAD)

Credit Risk Modelling

- Over the past decade, banks have devoted many resources to developing internal models to better quantify their financial risks and assign regulatory and economic capitals.

- In the context of data mining, the risk parameter modeling is typically casted as a supervised learning problem. Using the historical information as the training data, to predict the future default behaviors of its customers.

- Typically, regressions models or a single decision tree model for estimating parameters (PD/LGD/EAD) will be considered.

- Using a combination of several models is not often considered.

ENSEMBLE

Ensemble

- What is an ensemble?

  - The general idea of ensemble modelling is to combine several individual models (base learner) in a reasonable way to obtain a final model that out-performs every one of them.

- Why ensemble?

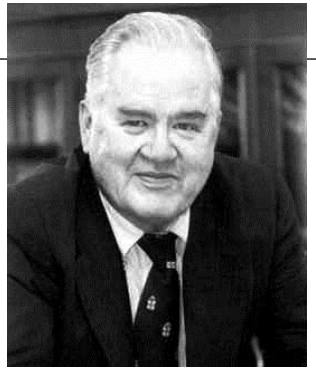
  - The fundamental reason for ensemble to work is that the combination of a collection of weak learners will produce a strong prediction result.

Ensemble

- Condorcet's jury theorem (French mathematician Marquis de Condorcet in 1785):

- If each voter has an independent probability *p* of being correct, and the probability of a majority of voters being correct is *E* then:
  - $p > 0.5$ implies $E > p$
  - *E* approaches to 1, if for all $p > 0.5$ as the number of voter approaches infinity.

Ensemble



- In 1977, John W. Tukey suggested combining two linear regression models, the first for fitting the original data, the second for fitting the residuals. The resulting model is almost always better than a single regression model.

CART → RANDOM FOREST

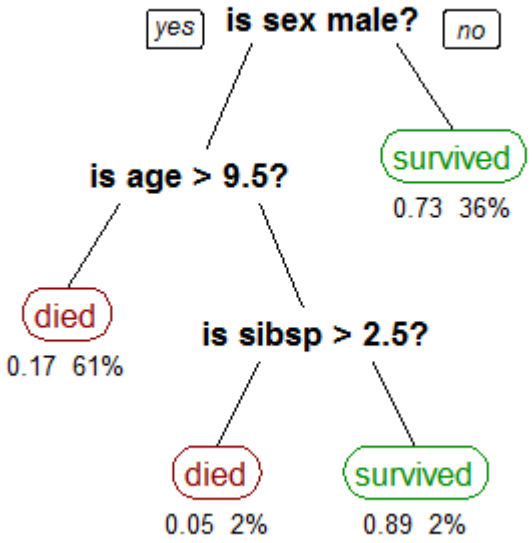
CART to Random Forest

- Professor Leo Breiman (UC, Berkeley)
  - Classification and Regression tree (CART)
    - a.k.a. Decision tree or regression tree
  - Bootstrap Aggregation (Bagging)
    - An ensemble of CART to increase the performance from a single tree
  - Random Forest (RF)
    - A further improvement of the bagging algorithm

- CART (1984) ⟶ Bagging (1996) ⟶ RF (2001)

- Adaptive boosting (AdaBoost) formulated by Yoav Freund and Robert Schapire (1996).

CART (Classification and Regression Tree)



CART

- Advantages:
  - Good predictive power
  - Able to handle both numerical and categorical valued inputs
  - Robust to missing values
  - Highly interpretable results

- Limitations:
  - Greedy algorithm: locally optimal decisions are made at each node and hard to return a globally-optimal tree.
  - Over-fitting: over-complex trees which do not generalize well.

Bootstrap Aggregating (Bagging)

Bagging Algorithm:

Given a training set $D = \{(X_i, Y_i)\}$, $i = 1, \ldots, N$:

1. For each b = 1 through B
   - **Bootstrap step:** Draw a bootstrap sample of D; call it D^b.
   - **Tree building**: Train a chosen base leaner (e.g. a decision tree) f_b on D^b
2. Output ensemble by combining results of B trained models:
   - **Regression:** average $$F(x) = \frac{1}{B}\sum_{b=1}^{B} f_b$$
   - **Classification:** take majority vote.

Random Forest (RF)

Random Forest Algorithm:

Given a training set $D = \{(X_i, Y_i)\}$, $i = 1, \ldots, N$:

1. For each b = 1 through B
   - **Bootstrap step:** Draw a bootstrap sample of D; call it D^b.
   - **Random subset step**: When building f_b, randomly select a subset of m < M predictors X^m before making each split, grow f_b on D^b rather than over all possible predictors.

2. Output ensemble by combining results of B trained models:
   - **Regression:** average $F(x) = \frac{1}{B}\sum_{b=1}^{B} f_b$
   - **Classification:** take majority vote.

Why Random Forest Works?

Random Forest (RF)

- Remarkable result:

$$\grave{o}_{RF} \leq \bar{\rho}\left(\frac{1-s}{s^2}\right)$$

- The mean correlation $\bar{\rho}$ between any two member of the forest;
- The mean strength *s* of a typical member of the forest.
- The generalization error of a random forest is bounded.

- A good random forest:
- Small $\bar{\rho}$: reduce the correlation between individual base learners/trees.
- Large *s*: make each base learner/tree as accurate as possible.

Random Forest: Equations

- Definition: The set $RF = \{f(X, \theta_b); \theta_b \sim P_\theta\}$ is called a random forest.
- Prediction error: $\grave{o}_{RF} = P_{(X,Y)}(M_{RF}(X,Y) < 0)$

  where $M_{RF}(X,Y) = \int m(\theta; X, Y) dP_\theta$

  and $m(\theta, x, y) = I(f(x;\theta) = y) - I(f(x;\theta) \neq y)$

- Strength of an individual tree: $s = \int\int m(\theta; X, Y) dP_\theta dP_{(X,Y)}$
- The correlation of any two trees in the forest:

$$\bar{\rho} = \frac{\int\int cov_{(X,Y)}(m(\theta; X, Y)), m(\theta^*; X, Y)) dP_\theta dP_{\theta^*}}{\int sd_{(X,Y)}(m(\theta; X, Y)) dP_\theta \int sd_{(X,Y)}(m(\theta^*; X, Y)) dP_{\theta^*}}$$

A New Algorithm – Ensemble by PLS

Motivation

$$\frac{1}{B}\sum_{b=1}^{B} m(\theta_b; X, Y) \rightarrow \int m(\theta; X, Y) dP_\theta$$

Partial Least Squares (PLS)

- The PLS technique works by successively extracting factors from both X and Y such that covariance between the extracted factors is maximized.

- Linear decomposition of X and Y:

$$X = TP^T + E$$

$$Y = UQ^T + F$$

Ensemble by Partial Least Squares (PLS)

Ensemble by Partial Least Squares Algorithm:

Given a training set $D = \{(X_i, Y_i)\}$, $i = 1, \ldots, N$:

1. For each $b = 1$ through B
   - **Bootstrap step:** Draw a bootstrap sample of D; call it D^b.
   - **Random subset step**: Randomly select a subset of predictors X^m, train a tree f_b on D^b and X^m rather than over all possible predictors.

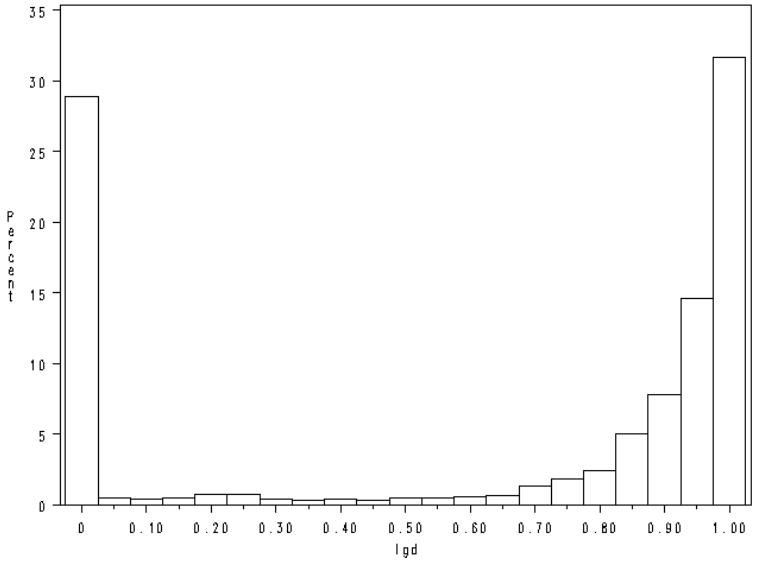
2. Output ensemble by combining results of B trained models:
   - **PLS:** use the output of individual tree $\hat{f}_b$ to formula a pseudo dataset $\{(\hat{f}_b, y_i)\}$, $i = 1, \ldots, N$, and fit a partial least square model to make the final prediction.

Experimental Results

LGD data

- Retail Portfolio LGD data:
  - 3500 accounts with 499 predictive variables
  - Randomly split into 60% training samples (2100 accounts) and 40% validation samples (1400 accounts).

- Empirical distribution of the target variable:



Regression Tree

- Tree setting:

MAXBRANCH	MAXDEPTH	LEAFSIZE
3	6	10

- Discriminatory Power:
  - RCAP = 0.354 on Validation
  - RCAP = 0.623 on Training

Random Forest

- Tree setting (Base learner):

MAXBRANCH	MAXDEPTH	LEAFSIZE
3	3	10

- Discriminatory Power:

Num. of Trees	RCAP on Validation	RCAP on Training
1	0.304	0.339
3	0.358	0.52
5	0.361	0.534
10	0.395	0.534
20	0.401	0.554
30	0.405	0.549
50	0.408	0.554
100	0.408	0.554

Ensemble by PLS

- Tree setting (Base learner):

MAXBRANCH	MAXDEPTH	LEAFSIZE
3	3	10

- Discriminatory Power:

	Random Forest		Ensemble-PLS	
Num. of Trees	RCAP on Validation Samples	RCAP on Training Samples	RCAP on Validation Samples	RCAP on Training Samples
100	0.400	0.549	0.419	0.589

Summary Result

- Tree setting (Base learner):

Method	RCAP on Validation
Regression Tree	0.354
Random Forest	0.408
Ensemble by PLS	0.419

Remarks

❑ Advantages:

- Strong model performance : as in summary result
- Resist to over-fitting/robustness: fitting a larger model on the training data rarely leads to over-fitting on the validation data.
- Fool proof modelling method: no transformation on the input variables, no strict requirement on variables selections, does not require the pruning of the tree.

❑ Disadvantages:

- Interpretability: (solution) variable importance measure

$$Imp_j^2(\hat{f}^{tree}) = \sum_{k=1}^{m} \hat{d}_k \times I\{\text{split at node } k \text{ is on variable } j\}$$

- Computation difficulty: (solution) parallel computing

SAS Implementation

- PROC SURVEYSELECT
- PROC ARBORETUM
- PROC PLS
- SAS/IML

Reference

[Marquis de Condorcet Essay on the Application of Analysis to the Probability of Majority Decisions. 1785]

[J.W. Tukey(1977) Exploratory data analysis. Addison-Wesley, Reading]

[L. Breiman, J. Friedman, C.J. Stone, and R. A. Olshen, Classification and regression trees. Chapman & Hall/CRC, 1984]

[L. Breiman Bagging Predictors. 1996 Machine Learning 24 (2):123-140.]

[L. Breiman. Random Forests. 2001 Machine Learning 45(1):5-32.]

[S. Wold, A. Ruhe, H. Wold. The collinearity problem in linear regression: the partial least squares (PLS) approach to generalized inverse. SIAM Journal on Scientific and Statistical Computing 5(3): 735-743.]

Thank you