Stock market prediction by Bayesian methods

Bayesian Logistic Model

1. Methodology introduction

Logistic regression is a very common method for modeling categorical response variable. With the power of Bayes, we implemented the Bayesian logistic regression model on the stock market, which uses the past observations to give an inference of model parameters.

2. Model setting

We specified two models for two general cases, which are single asset and multiple assets.

- For single assets, we define a Bernoulli random variable $Y_t$ that indicate if the asset price is going up or down (1 for up, 0 for down) at day $t$. We select features $X_t$ as the open, high, low, close, and volume at day $t$ to $t-5$.
- For multiple assets, we define a Bernoulli random variable $Y_{it}$ that indicate if the asset price is going up or down (1 for up, 0 for down) at day $t$. We select features $X_{it}$ as price to book ratio, price to earnings ratio, earnings per share, returns on asset, returns on equity, market capital size, market float shares as features.

- Models are as follows.
  - Single model
    - $a_0 \sim N(0, \theta)$
    - $\beta_0 \sim N(0, \theta)$
    - $\beta_i \sim N(0, \theta)$
    - $\gamma_i \sim \text{Bernoulli}(\mu)$  \hspace{1cm} $\mu = \frac{1}{1 + e^{-\beta_0 - \beta_i X_{it}}}$
  - Multiple asset
    - $a_0 \sim N(0, \theta)$
    - $\beta_0 \sim N(0, \theta)$
    - $\beta_i \sim N(0, \theta)$
    - $\gamma_i \sim \text{Bernoulli}(\mu_i)$  \hspace{1cm} $\mu_i = \frac{1}{1 + e^{-\beta_0 - \beta_i X_{it}}}$

3. Results

By look at the trace plot of some parameters, we believe that we achieved good convergence by metropolis sampling.

Bayesian Neutral Network

1. Methodology introduction

(1) Assume samples are independent, and sample size is $m$ with $p$ input variables, then the input matrix $X$ will be a $p \times m$ matrix, the response $Y$ is 0/1 binary.

(2) The weight for the first hidden layer is $W_1$ with dimension $4 \times p$, the bias for the first hidden layer is $B_1$, calculate $Z_1 = W_1 \times X + B_1$, which is a $4 \times m$ matrix, apply activation function to each element of $Z_1$, then we get $A_1$ which is the input for the second hidden layer.

(3) The weight for the output layer is $W_2$ with dimension $1 \times 4$, the bias for the output layer is $B_2$, calculate $Z_2 = W_2 \times A_1 + B_2$, which is a $1 \times m$ vector, apply sigmoid function to each element of $Z_2$, then we get $A_2$ which are the probabilities of each response $Y$ equals to 1.

(4) Since $A_2$ are the probabilities of each response $Y$ equals to 1, then $Y_i \sim \text{Bernoulli}(A_2[i])$.

(5) After getting the joint likelihood, we can put it into the classic MCMC framework with prior distribution of all parameters.

2. Model setup

- We use dynamic model to make prediction, because we assume the patterns of S&P 500 index movement are changing over time.
- After tuning hyperparameters, we choose to use the previous 6 days', open, close, high, low, volume, change rate of S&P 500 index as input variables with normalization (scale to mean 0 and variance 1), one hidden layer with 4 neurons.
- We use the previous 30 days as the training set to make prediction for the following day, so if we want to predict 100 days, then I will need to fit 100 models.

Summary

Implication: directed the activity we take in stock market

Thoughts and future directions:

(1) Without any nice prior information, We do not recommend using MCMC on Neural Network. Because there are many parameters in Neural Network, it will take a long time to sample all parameters till converge.
(2) Try using other faster sampling methods.
(3) Try using hierarchical model.
(4) Try finding more relevant variables.
(5) Try using Recurrent Neural Network to account for correlations among the sequence of observations.

We set the prior distribution of parameters of model to be normal distribution with mean 0 and variance 100, set the MCMC candidate distribution also to be normal with variance 0.01. We also add a regularization term with coefficient 0.5.

Trading Strategy

- Calculate the mean of predicted probabilities, if smaller than 0.5, then we short the S&P 500 index at the open of the trading day, if bigger than 0.5, then we long the S&P 500 index at the open of the trading day, and close the position at the close of trading day.
- Test whether 0.5 is in the 80% credible set of predicted probabilities, if 0.5 is not within credible set, then we choose to trade (if 0.5 is on the left of credible set, then we long S&P500, if 0.5 is on the right of credible set, then we short S&P500); if 0.5 is within credible set, then we choose not to trade.

* Assuming no transaction fee, no leverage, trading can be made at open and close time.

3. Results

- Trace plot

Because it take very long time to sample so many parameters in Neural Network, so I just run one chain with 6000 iterations and burn in the first 3000 samples. By looking at the trace plots, the convergence seems ok after 3000 iterations.

- Back test result of Bayesian Neural Network on the recent 150 trading days:

  The accuracy is: 59%; The close price of S&P 500 has increased 6.4% over the recent 150 trading days; By strategy 1, we get 13% return; By strategy 2, we get 3% return;

* data from https://finance.yahoo.com/