(1) Which variables matter for predicting S1?

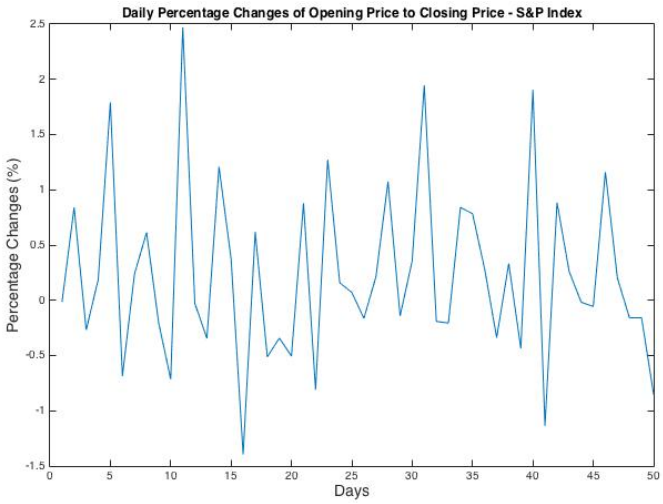
I implement the feature scoring method, which is based on Random Forest to estimate the importance of each feature. The following list is generated by my featureScoring.py program.

Feature Ranking:
1. feature S8 (0.115102)
2. feature S4 (0.115102)
3. feature S5 (0.113878)
4. feature S7 (0.113388)
5. feature S10 (0.112490)
6. feature S9 (0.109551)
7. feature S3 (0.108408)
8. feature S6 (0.107592)
9. feature S2 (0.104490)

Based on this list, we can know S8 is the most important feature for predicting S1 and the next is S4 and so on.

(2) Does S1 go up or down over this period?

I plot the prediction value over this time frame as follows.



Because the prediction value is the daily percentage changes of opening price to closing price, we cannot easily figure the trend over this period. However, we can approximately estimate the trend of S&P index is up. Because from this picture, we can find most of day the percentage changes is positive (larger than 0), that means the overall trend for S&P is going up even the speed of going-up is not so significant.

(3) How much confidence do you have in your model?

In this model, I use 10-fold cross-validation on our training dataset and get the minimum RMSE is 0.3200 with standard deviation 0.1012. That means I have 95% confidence that my model's RMSE can fall into $0.3200 \pm 2*0.1012$ for real test data.

RMSE = sqrt(mean((predictedValue - trueValue).^2))

(4) What techniques did you use?

This problem is a regression problem and the model can be split into two parts:
STEP1. Estimate the importance of features (featureScoring.py)
STEP2. Training gaussian kernel ridge regression model (main.m)

STEP1:
This feature scoring method is based on Random Forest to estimate the importance of each feature. According to features' ranking, I choose the highest 5 features to calculate their mean and standard deviation. And then add them into the original feature set. I attempt to use important features to generate new features so that I can increase feature complexity. This might be helpful for establishing the prediction model in the second step.

STEP2:
For this problem, I choose kernelized ridge regression method with gaussian kernel which can effectively deal with non-linear dataset to achieve higher prediction accuracy. Among this model, there are two hyper-parameters: **gamma** and **lambda**. Gamma is the parameter for gaussian kernel and lambda is regularization parameter. In the main program, I implement n-fold cross-validation to tune the best combination of these two parameters and use RMSE to evaluate model performance.

RMSE = sqrt(mean((predictedValue - trueValue).^2))

After training our gaussian kernel ridge regression model, I apply test data into this model and generate the final prediction.csv dataset.