A Pairs Trading Strategy for GOOG/GOOGL Using Machine Learning

Jiayu Wu

December 9, 2015

Abstract

We apply the spread model, the O-U model and SVM to build a pairs trading strategy for GOOG/GOOGL. There are two parts of the project that we think are novel after reviewing past related work: first, we model not only price spread but also several selected technical indicators' "spread"; second, we use two new metrics for measuring our trading strategy instead of the traditional back-testing method because we want to focus more on future prediction rather than return prediction, and to achieve that, we also propose two algorithms to reconstruct the data set. We outline the process of how our strategy is executed and, at the end, show that our strategy delievers a good win-rate.

1 Introduction

Pairs trading is a popular trading strategy in the last three decades after it was first used by Morgan Stanley in 1980s. Pairs trading means to utilize a pair or a bag of related financial instruments to make profits by exploiting their relations. One important feature of pairs trading is that it is market-neutral, which is particularly appealing in the current volatile and unpredictable macro-economic environments.

In this project, we will use the spread model, the O-U mean-reverting model, and SVM to build a trading strategy and apply the strategy to GOOG/GOOGL. We will first illustrate the spread model and the O-U mean-reverting in detail. Unlike most previous work that only takes price spread into consideration, we will also use the spread model and the O-U mean-reverting model to model the two securities' technical indicators. In other words, we extend the concept of "spread" by also investigating technical indicators' spread. We will construct trading signals by processing different kinds of "spreads" and then use these trading signals as input features for SVM classification. Instead of using the traditional back-testing method to test our trading strategy, we will use SVM binary classification to measure our trading strategy. To achieve that, we will reconstruct the original pricing feeds to labeled examples, and there are two methods we use to reconstruct the labeled examples, one for measuring the strategy's ability to seize profit opportunities, and the other for measuring the strategy's ability to make directional predictions.

One important thing for a pairs trading strategy is to select a proper pair of financial instrucments. For example, if the price of security A always rises when the price of security B rises, it seems that A and B may be used for pairs trading. However, the explicit relation between prices may not be good enough for a good pair. The good pairs should share as many the same intrinsic characteristics as possible. GOOG/GOOGL are both shares of Google Inc. (now Alphabet Inc.) but with different vote rights. GOOGL represents Class A shares while GOOG represents Class C shares. Only Class A shares have voting rights. Therefore, generally, the price of GOOGL is slightly higher than that of GOOG. Other than voting rights, they are essentially the same since their prices are based upon the same fundamentals.

2 Related Work

In *Statistical Arbitrage using Pairs Trading with Support Vector Machine Learning*, Gopal Rao Madhavaram compares O-U model (mean reverting process) and SVM's performance on pairs trading, more specifically, index arbitrage, using constituents stocks to predict the corresponding index. His results show that SVM achieves a slightly better performance than the mean-reverting model.

There are also some CS229 projects from previous years that have interesting methods and results related to pairs trading. In *Machine Learning in Pairs Trading Strategies*, the authors first use linear regression to com-

pute and generate trading signals and obtain a sharperatio of 1.14 from back testing and they also propose a trading strategy using EM algorithm and Kalman filter. Moreover, in *Machine Learnig in Statistical Arbitrage*, the authors investigate *index arbitrage*, and they first use PCA to select first 12 components out of 100 candidates (constituents of FTSE 100 Index), and then use O-U model to produce a mean-reverting process, which helps to create the trading signal. Their back-testing shows that their trading strategy can make reasonable profits.

3 Dataset

We obtain the data set from Quantquote.com. The timeframe for our data set is 1 min, ranging from 10/01/2014 to 10/30/2015.

First we need to do some data preprocessing because pairs trading require the data of the two securities must be consistent. By consistent we mean that the date and time of every feed of both securities should be a exact match. However, in our data set, both securities have some missing bars, most of which are time intervals with no trading volume, explained by Quantquote.com. Therefore we wrote some python script to fill up the missing bars using last close price and volume = 0. These missing bars are about 0.3% of the entire data set.

The original data set has 106141 feeds for GOOG, 106136 feeds for GOOGL; after preprocessing, the data set we use has 105690 feeds for both securities.

Data example: {'date': 20151030, 'time': 1559, 'open': 711.98, 'high': 712.58, 'low': 710.72, 'close': 710.78, 'volume': 41773}

4 Pairs Trading Model

4.1 Spread Model

The canonical pairs trading spread model is as follows:

$$\frac{dA_t}{A_t} = \alpha dt + \beta \frac{dB_t}{B_t} + dX_t \tag{1}$$

where A_t is the price of security A at time t, B_t is the price of security B at time t, X_t is the residual term, which has the mean-reverting property because mean-reverting spread is the basic assumption of pairs trading and the drift term $\alpha dt \ll \beta \frac{dM_t}{M_t}$, which is neglectable compared to the return of either security. The above model shows that the stock price of the two securities is linear related. The β here helps solve the normalization problem since the two stock prices may not fluctuate in the same range. Furthermore, β may change over time because of change of some intrinsic characteristics of either company or change of overall stock market regime, however, in this project, we assume β is a constant term over the duration of the dataset.

In this project, we will use 'close' - 'open' as dA_t (or dB_t), 'open' as A_t (or B_t) for linear regression.

4.2 O-U Model

Now we investigate the residual term from the above spread model. We will use OrnsteinUhlenbeck process to model the residual term because the O-U process is a stochastic process such that the object modeled by the process always drifts towards its long-term mean. The residual term, namely the spread, has very similar property according the assumption of pairs trading. The residual term X_t from the above spread model satisfies the following stochastic differential equation:

$$dX_t = \theta(\mu - X_t)dt + \sigma dW_t \tag{2}$$

Where θ, μ and σ are the paramters we want to estimate later using linear regression. W_t denotes Wiener process, which suggests that the probability distribution of W_t is a normal distribution with mean = 0 and variance = t.

By integrating (2), we have

$$X_{t+1} = a + bX_t + \epsilon_{t+1} \tag{3}$$

Where

$$a = (1 - e^{-\theta \Delta t})\mu$$
$$b = e^{-\theta \Delta t}$$
$$Var(\epsilon) = \sigma^2 \frac{1 - e^{-2\theta \Delta t}}{2\theta}$$

We can obtain the above parameters from running a linear regression on X_{t+1} against X_t, and then we can have obtain estimates for the parameters in the original O-U process equation:

$$\theta = -log(b) \times \frac{1}{\Delta t}$$
$$\mu = \frac{a}{1-b}$$
$$\sigma = \sqrt{\frac{Var(\epsilon)2\theta}{1-b^2}}$$
$$\sigma_{eq} = \sqrt{Var(X_t)} = \frac{\sigma}{\sqrt{2\theta}} = \sqrt{\frac{Var(\epsilon)}{1-b^2}}$$

5 Pairs Trading Model - A Modified Version

In this project, we extend the above pairs trading model to also take technical indicators into consideration.

Most of the previous work only consider the spread of price. It is possible that some indicators of the two securities may also provide as much useful information as their prices. Moreover, two candidates for a good pair should not only exhibit similar price movement but also similar movements for some technical indicators. Therefore, we want to investigate the difference of indicators like we do for the price spread.

Now the challenge is to figure out what indicators to use. A good indicator for this project must exhibit similar behaviours for both securities. Therefore, empirically, we selected the following 4 indicators:

1.**SMA** : simple moving average.

$$SMA_t \ = \ \sum_{i=t-\$length+1}^{t} P_i^{close}$$

We choose SMA because it is simple to compute and also helps incorporate information from previous $length time periods.

2.**WMA** : weighted moving average.

$$WMA_t \ = \ \sum_{i=t-\$length+1}^{t} W_i P_i^{close}$$

where $W_i = \frac{i-(t-\$length)}{\sum_{j=t-\$length+1}^{t} j-(t-\$length)}$

WMA is also a technical indicator related to average price but it differs from SMA in that it assign weights to prices: the further away from the current time, the less weight that price has.

3.**MFI** : money flow index.

$$MFI_t \ = \ 100 \ - \ (\frac{100}{1 \ + \ MoneyFlowRatio_t})$$

where
$MoneyFlowRatio_t = \frac{\$length-period\ Positive\ MoneyFlow_t}{\$length-period\ Negative\ MoneyFlow_t}$,
$MoneyFlow_t = volume \times \frac{P_t^{high}+P_t^{low}+P_t^{close}}{3}$.
$MoneyFlow_t$ is positive if $\frac{P_t^{high}+P_t^{low}+P_t^{close}}{3} > \frac{P_{t-1}^{high}+P_{t-1}^{low}+P_{t-1}^{close}}{3}$.

MFI takes volume into consideration so that we can also see volumes' impact on the final results. Since GOOG and GOOGL are influenced essentially by the same factors, money should flow in or flow out both stocks at the same time in most cases.

4.**RSI** : relative strength index.

$$RSI_t \ = \ 100 \ - \ (\frac{100}{1 \ + \ RS_t})$$

where $RS = \frac{average\ of\ \$length-period\ up\ closes}{average\ of\ \$length-period\ down\ closes}$.

RSI is good at indicating whether the stock is in oversought or overbought condition. We use RSI because we believe the two securities, GOOG and GOOGL, tend to be in the same condition.

The $\$length$ parameter will be discussed in "Evaluation Metrics" section.

6 SVM

Support Vector Machine (SVM) is a popular and effective machine learning algorithm for both classification and regression. The idea of SVM is to construct a hyperplane to seperate the two classes of data with the gap being as wide as possible.

The goal is do the following optimization:

$$min_{\gamma,w,b} \ \tfrac{1}{2}\|w\|^2 \ + \ C\sum_{i=1}^{m}\xi_i$$

s.t. $y^{(i)}(w^T x^{(i)} \ + \ b) \ \geq 1-\xi_i, \ i=1,...,m$

$\xi_i \geq 0, \ i=1,...,m.$

Alternatively, we can use the following dual form:

$$max_\alpha \, W(\alpha) = \sum_{i=1}^{m}\alpha_i - \tfrac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)}, x^{(j)}\rangle$$

s.t. $0 \leq \alpha_i \leq C, \ i=1,...,m$

$\sum_{i=1}^{m}\alpha_i y^{(i)} = 0.$

7 Trading Strategy Framework

The trading strategy is executed as follows:

1. *run linear regression to get spread model residuals*;
2. *construct X_ts using residuals from the previous step according to $X_t = \sum_{i=t_0}^{t} dX_t$;*
3. *run lag 1 auto-regression on X_ts to get parameters according to O-U model*;
4. *use parameters obtained from the previous steps to compute T-scores for price and selected indicators*;
5. *train SVM with the modified data set (we will talk about data set reconstruction in "Evaluation Metrics" section)*;
6. *apply the trained model to test data set, namely to make trading decisions, which is either "make a bet" or "stay calm and do nothing".*

8 Trading Signal

We give the following definitions for constructing the feature space for SVM classification.

Here we define T-score for each feature:

$$T_{price} = |\frac{X_t^{price} - \mu_{price}}{\sigma_{eq}^{price}}|$$

By the definition, we can see that T-score is standardized version of X_t. We use the absolute value here because in this project we focus on the absolute value of the spread not its sign.

Likewise, we define the following T-scores for other technical indicators: $T_{sma} = |\frac{X_t^{sma} - \mu_{sma}}{\sigma_{eq}^{sma}}|, T_{wma} = |\frac{X_t^{wma} - \mu_{wma}}{\sigma_{eq}^{wma}}|, T_{mfi} = |\frac{X_t^{mfi} - \mu_{mfi}}{\sigma_{eq}^{mfi}}|, T_{rsi} = |\frac{X_t^{rsi} - \mu_{rsi}}{\sigma_{eq}^{rsi}}|$.

T-scores will be used as feature input for SVM binary classification (next section).

9 Evaluation Metrics

We use a different evaluation metrics rather than traditional back-testing in this project. Instead, we use binary classification to evaluate the trading strategy. For classification purpose, we need to reconstruct the data set according to the following algorithm. The trading logic behind this algorithm is that pairs trading is statistical arbitrage, and that means the value investing rule should not be applied here. Therefore, the goal of this trading strategy is to profit in a short period of chosen trading time frame. And only if the spread can narrow to an extent defined by \$threshold in a short period, is the trading strategy willing to take the risk and make a bet. Notice here we use the absolute value of X_t because we only care about betting on that the spread will narrow in the future. The sign of spread only becomes relevant if we want to decide which security to long and which security to short, but for this project, we want to focus on building a trading strategy that bets on the direction of spread's change.

for each time t **do**
  profit = False
  for each of the following \$*length* time intervals, dX_{t+i} **do**
    if $|dX_{t+i}| <= \$threshold \times |dX_t|$ **then**
      profit = True
      break
    end if
  end for
  if profit **then**
    $label_t = +1$
  else
    $label_t = -1$
  end if
end for

\$*length* is set to 5 because we want to profit in the near future before our patience goes out. \$*threshold* should be set according to transaction cost in real-world trading system. For this project, we set \$threshold to be 0.25 because the resulting labels are balanced, which means around half of the data has a label (+1) and another half has a label of (−1), in order for SVM classification to work well.

Another metrics we use is to examine whether the trained model can predict the correct moving direction of T-scores in the next time interval. The data reconstruction algorithm is as follows:

for each time t **do**
  if $|dX_t| > |dX_{t+1}|$ **then**
    $label_t = +1$
  else
    $label_t = -1$
  end if
end for

For metrics 1 reconstruction, we have 53021 examples with label (+1), 52658 examples with label (−1); for metrics 2 reconstruction, we have 52780 examples with label (+1), 52899 examples with label(−1).

Now the problem is transformed to a binary classification problem, where (+1) means "make a bet" and (−1) means "stay calm and do nothing". Furthermore, now the input features for each time t is a vector of T-scores defined in last section, e.g.

$\{'T'_{price} : 0.0745, 'T'_{sma} : 0.3250, 'T'_{wma} : 0.6684, 'T'_{rsi} : 0.3421, 'T'_{mfi} : 1.837\}$

Notice that this is an example before feature rescaling since we will later rescale each feature for SVM to work well.

10 Results

We first split each reconstructed data set into training data set (80%) and test data set (20%), then use scikit-learn package to train a binary classification SVM model with a linear kernel. Finally, we apply the trained model to test data set, and obtain the following results for each

metrics:

Metrics 1:

	Positive	Negative	Total
Positive	$TP = 6745$	$FN = 3878$	10623
Negative	$FP = 2291$	$TN = 8222$	10513
Total	9036	12100	N

accuracy	precision	recall	F-measure	AUC
0.7081	0.7465	0.6349	1.0670	0.7085

Metrics 2:

	Positive	Negative	Total
Positive	$TP = 6993$	$FN = 3552$	10545
Negative	$FP = 2238$	$TN = 8353$	10591
Total	9231	11905	N

accuracy	precision	recall	F-measure	AUC
0.7261	0.7576	0.6632	1.0574	0.7259

Results from Metrics 1 show that our trading strategy is able to moderately beat the market with a win-rate of 0.7465 (win-rate is defined as (profit trades / total trades made), which is (true positive / positive predictions)). Moreover, the results for Metrics 1 also exhibit risk-averse property of the trained SVM model. While the test data set is well balanced (half positive labels and half negative labels), the model made 9036 positive predictions and 12100 negative predictions. In addition, the model missed 3878 profitable trades and had 2291 loss trades. From the above two sets of statistics, we can see that the model would rather miss a profitable trader to avoid a possible loss trade. From the confusion matrix, we can also know that the model successfully seized 63.49% (TP/(TP+FN)) of available profit opportunities and avoided 78.21% (TN/(TN+FP)) of possible loss trades. Therefore, our strategy may be a better candidate for conservative investors than for risk-seeking investors.

Results from Metrics 2 show that the ability of our strategy to predict future direction of spread change is slightly better than the ability of our strategy to make profitable trades, because all statistics except F-measure are better than those of Metrics 1. It is also expected because even for human traders, making directional predictions is generally easier than making profitable trades. The results also suggest that our trading strategy can also be used as a technical indicator for pairs trading purpose because it can give a prediction on whether the spread will narrow in the next time period.

By investigating the model's predictions in more detail, we have an interesting finding, which we call *Opportunity Window*. For example, if $|X_t|$ is 0 or very close to 0, then for a trade made in any of the previous $\$length$ time periods, the trade is a succesful bet given the algorithm in "Evaluation Metrics" section. Then the $\$length$ time periods preceding the time t when $|X_t|$, which is equal to 0 or very close to 0, occurs form a *Opportunity Window*. Therefore, if we can have a powerful prediction on when $|X_t|$ is 0 or very close to 0, we can freely enter trades during the preceding $\$length$ time periods. This also explains that in the Metrics 1 reconstruction data set, consecutive examples' label tend to be the same because these examples share the same small $|X_t|$ in their following time periods.

11 Conclusion

In this project, we utilize both price and technical indicators for our pairs trading strategy while most of previous work only uses price to build pairs trading strategy. We also devise two new metrics to measure our strategy because the traditional back-testing for return prediction is not our focus and can be not very useful since it involves much more practical problems, such as slippage and transaction cost. Instead, we want to focus more on directional prediction. The results show that our strategy is moderately predictive, and has slightly better performance on predicting spread change direction than predicting profitable trades. For future work, we can include more features in our model, and may do PCA to select most useful technical indicators for pair trading purpose. Moreover, the β in the spread model may change over time, but in this project, we treat it as a constant over the time duration of our data set. The performance may be improved if we design a update rule to update β for every n-period of time.

12 References

[1] Gopal Rao Madhavaram. *Statistical Arbitrage Using Pairs Trading With Support Vector Machine Learning*
[2] Yuxing Chen, Weiluo Ren, Xiaoxiong Lu. *Machine Learning in Pairs Trading Strategies.*
[3] Xing Fu, Avinash Patra. *Machine Learning in Statistical Arbitrage.*