Stock Trend Prediction with Technical Indicators using SVM

Xinjie Di
dixinjie@gmail.com
SCPD student from Apple Inc

Abstract

This project focuses on predicting stock price trend for a company in the near future. Unlike some other approaches which are concerned with company fundamental analysis (e.g. Financial reports, market performance, sentiment analysis etc.), the feature space is derived from the time series of the stock itself and is concerned with potential movement of past price. Tree algorithm is applied to feature selection and it suggests a subset of stock technical indicators are critical for predicting the stock trend. It explores different ways of validation and shows that overfitting tend to occur due to fundamentally noisy nature of a single stock price. Experiment results suggest that we are able to achieve more than 70% accuracy on predicting a 3-10 day average price trend with RBF kernelized SVM algorithm.

Keywords: stock prediction, feature selection, SVM, stock technical indicator, scikit.

1 Introduction

Short-term prediction of stock price trend has potential application for personal investment without high-frequency-trading infrastructure. Unlike predicing market index (as explored by previous years' projects), single stock price tends to be affected by large noise and long term trend inherently converges to the company's market performance.

So this project focuses on short-term (1-10 days) prediction of stock price trend, and takes the approach of analyzing the time series indicators as features to classify trend (Raise or Down). The validation model is chosen so that testing set always follows the training set in time span to simulate real prediction. Cross validated Grid Search on parameters of rbf-kernelized SVM is performed to fit the training data to balance the bias and variances. Although the efficient-market hyothesis suggests that stock price movements are governed by the random walk hypothesis and thus are inherently unpredictable, the experiment shows that with 1000 transaction days as training data, we are able to predict AAPL's next day acutual close price trend with 56% accuracy, better than the random walk; and more than 70% accuracy on next 3-day, 5-day, 7-day, 10-day price trend. In the end, we conclude that stock technical indicators are very effective and efficient features without any sentiment data in predicting short-term stock trend.

2 Feature Space

2.1 Data Collection

The data is pulled from http://finance.yahoo.com . I picked the 3 stocks (AAPL, MSFT, AMZN) that have time span available from 2010-01-04 to 2014-12-10 to get enough data and 2 market index (NASDAQ and SP&500). The goal is to predict a single stock's trend (e.g. AAPL) using features derived from its time series plus those from NASDAQ and SP&500 for augmentation.

2.2 Features of Stock Indicators

A stock technical indicator is a series of data points that are derived by applying a function to the price data at time t and study period n. Below is a table of indicators that I compute from time series and transform to features:

Indicators	Name	Description	Formula
WILLR	Williams %R	Determines where today's closing price fell within the range on past 10-day's transaction.	(highest-closed)/(highest-lowest)*100
ROCR	Rate of Change	Compute rate of change relative to previous trading intervals	(Price(t)/Price(t-n))*100
MOM	Momentum	Measures the change in price	Price(t)-Price(t-n)
RSI	Relative Strength Index	Suggests the overbought and oversold market signal.	Avg(PriceUp)/(Avg(PriceUP)+Avg(PriceDown)*100 Where: PriceUp(t)=1*(Price(t)-Price(t-1)){Price(t)-Price(t-1)>0}; PriceDown(t)=1*(Price(t-1)-Price(t)){Price(t)-Price(t-1)<0};
CCI	Commodity Channel Index	Identifies cyclical turns in stock price	Tp(t)-TpAvg(t,n)/(0.15*MD(t)) where: Tp(t)=(High(t)+Low(t)+Close(t))/3; TpAvg(t,n)=Avg(Tp(t)) over [t, t-1, …, t-n+1]; MD(t)=Avg(Abs(Tp(t)-TpAvg(t,n)));
ADX	Average Directional Index	Discover if trend is developing	Sum((+DI-(-DI))/(+DI+(-DI))/n
TRIX	Triple Exponential Moving Average	Smooth the insignificant movements	TR(t)/TR(t-1) where TR(t)=EMA(EMA(EMA(Price(t)))) over n days period
MACD	Moving Average Convergence Divergence	Use different EMA to signal buy&sell	OSC(t)-EMAosc(t) where OSC(t)=EMA1(t)-EMA2(t); EMAosc(t)=EMAosc(t-1)+(k*OSC(t)-EMAosc(t-1))
OBV	On Balance Volume	Relates trading volume to	OBV(t)=OBV(t-1)+/-Volume(t)

		price change	
TSF	Time Series Forcasting	Calculates the linear regression of 20-day price	Linear Regression Estimate with 20-day price.
ATR	Average True Range	Shows volatility of market	ATR(t)=((n-1)*ATR(t-1)+Tr(t))/n where Tr(t)=Max(Abs(High-Low), Abs(Hight-Close(t-1)), Abs(Low-Close(t-1));
MFI	Money Flow Index	Relates typeical price with Volume	100-(100/(1+Money Ratio)) where Money Ratio=(+Moneyflow/-Moneyflow); Moneyflow=Tp*Volume

The above technical indicators cover different type of features:

1) Price change – ROCR, MOM
2) Stock trend discovery – ADX, MFI
3) Buy&Sell signals – WILLR, RSI, CCI, MACD
4) Volatility signal – ATR
5) Volume weights – OBV
6) Noise elimination and data smoothing – TRIX

Also TSF is another feature that itself does linear regression to suggest trend.

2.2 Feature Construction and Data Labeling

Those indicators are computed against different periods from 3-day to 20-day and each of them is treated as individual feature in feature space. E.g. ROCR3, ROCR6 means relative price to 3 days ago and to 6 days ago relatively.

The feature set used for this project is defined as follows:

full_features = [**'Adj Close'**, **'OBV'**, **'Volume'**, **'RSI6'**, **'RSI12'**, **'SMA3'**, **'EMA6'**, **'EMA12'**, **'ATR14'**, **'MFI14'**,**'ADX14'**, **'ADX20'**, **'MOM1'**, **'MOM3'**,**'CCI12'**, **'CCI20'**, **'ROCR3'**, **'ROCR12'**,**'outMACD'**, **'outMACDSignal'**, **'outMACDHist'**, **'WILLR'**, **'TSF10'**, **'TSF20'**, **'TRIX'**, **'BBANDSUPPER'**, **'BBANDSMIDDLE'**, **'BBANDSLOWER'**]

The feature matrix X is defined as: X(t) is a row vector of [full_features(stockToPredict), full_features(SP&500), full_features(NASDAQ)]. Hence the total features for an example is of size 3*len(full_features) = 84.

Depends on what we want to predict, the data is labeledwith "Up or Down". Y(t) is the label for data at time t, Y(t) = f(Price(t+n),Price(t+n-1),…Price(t)) where function f has the value in {-1, 1}. In this setting, it is guaranteed that future price trend is unseen to any features.

For predicting the next day trend,
Y(t) = 1 if Price(t+1)>Price(t); Y(t) = -1 if Price(t+1)<Price(t).

For predicting the next 3-day average price trend,

Y(t) = 1 if SMA(t+3)>SMA(t); Y(t) = -1 if SMA(t+3) < Price(t).

3 Model Fitting and Results

3.1 Feature Selection with ensemble Extremely Randomized Tree Algorithm

Limited by the length of document, I'm not describing the Extremely Randomized Tree algorithm here and it is in the reference. With total 84 features, there can exist a lot of noisy features that will overfit the training data and mislead the prediction. So the goal is to run an algorithm on training data to decide the ranking of features and pick only top 30% of features to feed into SVM classifier. The feature selection significantly helps to handle overfitting and below is a table of some top features chosen by Extremely Randomized Tree for each stock:

AAPL	WILLR,MOM3,ROCR3,RSI6,CCI12,CCI20,MOM1,N-ADX14,TRIX,N-CCI12..	28
AMZN	MOM1, N-MOM1, WILLR,SP-MOM1,ROCR3,CCI12,RSI6,MOM3,N-ROCR3	40
MSFT	MOM1,RSI6,WILLR,N-MOM1,CCI12,SP-MOM1,MOM3,ROCR3,MACDHist	30

From the above table we found that AAPL's feature ranking tend to favor more longer-term indicators such as WILLR, CCI12 etc that reflects a general trend, whereas AMZN's ranking shows that those 1-3 day features are the highest which means it tends to overfit and has high variance. MSFT is in the middle. We will see in the next section that this agrees surprisingly well with the validation of predicting results. The last column number is the number of relevant features selected by ER tree. AMZN uses significantly more features than the other two which implies the trained model is likely to overfit.

3.2 RBF-Kernelized SVM and Grid Search on parameters

The fitting model is soft-margin SVM with RBF kernel exp(|x-p|/l). We have two parameter to fit C and l. For each training set, I do 5-fold cross-validation and grid search on parameter pair <C,l> and pick the best parameter to do validation on test set.

3.3 Validation Model and Results

To validate the prediction accuracy, we train on 950 examples and predict on next 50 days. And we repeat the training and testing with a step window 10 examples so that in total we get 10 testing precision, taking the mean of them we get the following.
Preprocessing is applied to eleminate the mean and normalize the vaiance to 1. This is to avoid the magnitude difference of features will poise some significant weight which is undesirable.

Company/Accuracy	Next 3-day	Next 5-day	Next 7-day	Next 10-day
Apple	73.4%	71.41%	70.25%	71.13%
Amazon	63%	65%	61.5%	71.25%
Microsoft	64.5%	73%	77.125%	77.25%

4 Analysis and Conclusion

The result shows that for Apple the prediction is robust and above 70%. For Amazon, it suggests that longer-period prediction (10-day) is significantly better than the shortterm. For Microsoft, the split is between 5-day, shorter than that it tends to have very high noise while 7-day and 10-day prediction is extremely good. These can be mapped to the top features extracted: Apple stock has

more generalized indicators weighted heavier than others so that it has low variance. Amazon's top features are all 1-3 days volatility, so that the model tend to overfit. Actually the experiment shows the training error for Amazon is 80% which is significantly higher than the other two stocks.

Last result is directly predicting next day price trend, it shows on average it acheives 56% accuracy. This is very sound result since the next 1 day price is highly noise in stock market and should be close to random. With this model, we can do better than random walk consitently.

The result is very helpful in real-world investment for non-HFT inverster. By learning from past data we are able to get above 70% accurate prediction on the next couple day's trend. For future work, it worth adding sentiment data as features to augment the technical features. The challenge is how to eliminate as much as noise in sentiment data and quantify them.

Reference:

Learning to rank with extremely randomized trees

Efficient Machine Learning Techniques for Stock Market Prediction

Feature Investigation for Stock market Prediction

An SVM-based Approach for Stock Market Trend Prediction

http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:introduction_to_technical_indicators_and_oscillators

FEATURE SELECTION FOR PRICE CHANGE PREDICTION