Predicting Stock Market Returns

L. Torgo

ltorgo@knoyda.com
KNOYDA, Know Your Data!

Jul, 2019

Problem Description

The Context

- Stock market trading is an application area with a large potential for data mining
- Huge amounts of data (in several formats) are available
- Still, there are researchers claiming the impossibility of making money out of forecasting future prices - the *efficient markets hypothesis*
- The goal of trading is to maintain a portfolio of assets based on buy and sell orders, and achieve profit with this

Problem Description (2)

The Concrete Application

- We will use data mining in a slightly more specific trading context
- We will trade a single security - the S&P 500 market index
- Given **historic prices data** of this security and an **initial capital** we will try to maximize our profit over a future testing period **by means of trading actions** - buy, sell, hold

Problem Description (3)

The Concrete Application (cont.)

- Our trading strategy will **base the decisions on the results of a data mining process**
- This process will try to forecast the future evolution of prices based on historical data
- The overall evaluation criteria will be the profit/loss resulting from the trading actions

The Data

- We will use a data set available in package `DMwR2`

```
library(xts)   # extra package to install
data(GSPC,package="DMwR2")
first(GSPC)
```

```
##            GSPC.Open GSPC.High GSPC.Low GSPC.Close GSPC.Volume
## 1970-01-02     92.06     93.54    91.79         93     8050000
##            GSPC.Adjusted
## 1970-01-02            93
```

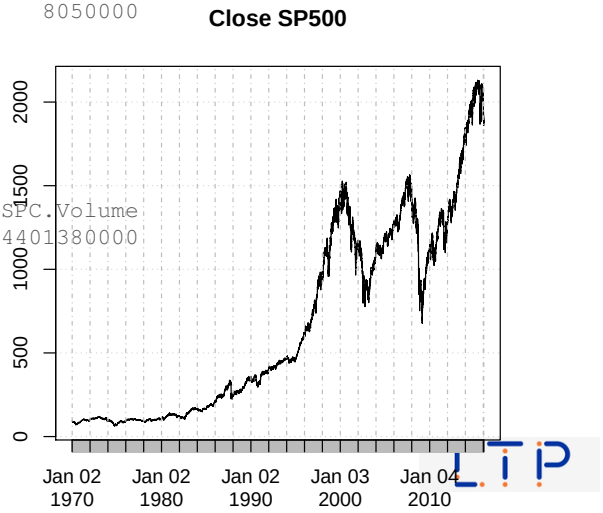
```
last(GSPC)
```

```
##            GSPC.Open GSPC.High GSPC.Low GSPC.Close GSPC.Volume
## 2016-01-25   1906.28   1906.28  1875.97    1877.08  4401380000
##            GSPC.Adjusted
## 2016-01-25       1877.08
```

```
dim(GSPC)
```

```
## [1] 11622     6
```

```
library(quantmod) # extra package you need
plot(Cl(GSPC),main="Close SP500")
```



Obtaining Further Prices Data

- The package `quantmod` has facilities for getting more data from the web

```
getSymbols('MSFT',from='2010-01-01')
getFX("USD/EUR")
getMetals("Gold")
```

```
## [1] "MSFT"
## [1] "USDEUR"
## [1] "XAUUSD"
```

Obtaining Financial Data on a Company

```
getFinancials("AAPL")

## [1] "AAPL.f"

viewFin(AAPL.f)

## Annual Balance Sheet for AAPL

##                                          2016-09-24 2015-09-26
## Cash & Equivalents                               NA         NA
## Short Term Investments                    58554.00   30212.00
## Cash and Short Term Investments           67155.00   41601.00
## Accounts Receivable - Trade, Net          15754.00   16849.00
## Receivables - Other                              NA         NA
## Total Receivables, Net                    29299.00   30343.00
## Total Inventory                            2132.00    2349.00
## Prepaid Expenses                                 NA         NA
## Other Current Assets, Total                8283.00   15085.00
## Total Current Assets                     106869.00   89378.00
## Property/Plant/Equipment, Total - Gross   61245.00   49257.00
## Accumulated Depreciation, Total          -34235.00  -26786.00
## Goodwill, Net                              5414.00    5116.00
## Intangibles, Net                           3206.00    3893.00
## Long Term Investments                    170430.00  164065.00
## Other Long Term Assets, Total              8757.00    5422.00
## Total Assets                             321686.00  290345.00
## Accounts Payable                          37294.00   35490.00
## Accrued Expenses                          20951.00   24169.00
## Other Current liabilities, Total           9156.00    9952.00
## Total Current Liabilities                 79006.00   80610.00
## Current Port. of LT Debt/Capital Leases          NA         NA
```


Defining the Predictive Task

What to Predict?

- To make a proper decision concerning our current position we need to be able to anticipate the future trend of the prices
- The following details our approach:
  - If prices vary more than p% we consider that worthwhile for trading
  - We want to forecast if this margin is attainable in the next k days - note that prices may go up and down during these k days
  - This is different from predicting the price for a certain future time tag
  - What we want is a good prediction of the general tendency of the prices in the next k days

What to Predict? (2)

- We will propose a variable, calculated with the quotes data, that reflects the tendency of the prices in a set of days
- We will try to forecast the future value of this variable
- Positive values of this tendency indicator will lead us to buy, whilst negative values will lead us to sell

What to Predict? (3)

- Let the daily average price be approximated by:

$$\bar{P}_i = \frac{C_i + H_i + L_i}{3}$$

- Let V_i be the set of k percentage variations of today's close to the following k days average prices (often called arithmetic returns):

$$V_i = \left\{ \frac{P_{i+j} - C_i}{C_i} \right\}_{j=1}^{k}$$

- The proposed indicator is the sum of the variations whose absolute value is above our target margin $p\%$:

$$T_i = \sum_{v} \{ v \in V_i : v > p\% \vee v < -p\% \}$$

What to Predict? (4)

- The following function implements the proposed indicator:

```
T.ind <- function(quotes,tgt.margin=0.025,n.days=10) {
   v <- apply(HLC(quotes),1,mean)
   v[1] <- Cl(quotes)[1]

   r <- matrix(NA,ncol=n.days,nrow=NROW(quotes))
   for(x in 1:n.days) r[,x] <- Next(Delt(v,k=x),x)

   x <- apply(r,1,function(x)
                  sum(x[x > tgt.margin | x < -tgt.margin]))
   if (is.xts(quotes)) xts(x,time(quotes)) else x
}
```


Inspecting the Values of the *T* Indicator

```
library(quantmod)
data(GSPC,package="DMwR2")
candleChart(last(GSPC,'3 months'),theme='white',TA=NULL)
avgPrice <- function(p) apply(HLC(p),1,mean)
addAvgPrice <- newTA(FUN=avgPrice,col=1,legend='AvgPrice')
addT.ind <- newTA(FUN=T.ind,col='red',legend='tgtRet')
addAvgPrice(on=1)
addT.ind()
```

What to Predict? - summary

- We will forecast the value of the T indicator using data mining models
- If the predicted value is above a certain threshold we will buy our asset
- If the predicted value is below a certain threshold we will sell our asset
- Otherwise we will just hold our current position

Which Predictors to Use?

- Which information should we give to our models (in the form of predictors) to obtain good predictions of the T indicator?
- The main assumption behind trying to forecast the future behavior of financial markets is that it is possible to do so by observing the past behavior of the market
- More precisely, that if in **the past behavior** p **was followed by** f, and **this pattern occurs frequently**, then if we are observing again p **we are confident that** f **will follow**

Which Predictors to Use? (2)

- We are approximating the future behavior using our T indicator
- We need to decide how to describe the recent past behavior of the prices
- We will try to collect a series of indicators that capture the recent dynamics of the prices

Which Predictors to Use? (3)

- Obvious candidates are the recent prices of the asset
- We will focus on the Closing prices, more precisely on the arithmetic h-days returns:

$$R_t^h = \frac{C_t - C_{t-h}}{C_{t-h}}$$

Which Predictors to Use? (4)

- Additional information can be given by calculating relevant statistics on the **recent evolution of the prices**
- Technical indicators are **numeric summaries** that reflect some **properties of the price time series**
- We will select an illustrative set of technical indicators calculated with the recent prices and use them as predictors for our models
  - Package `TTR` contains a huge sample of technical indicators

Which Predictors to Use? (5)

- Auxiliary functions we will use to obtain the predictors

```
library(TTR)
myATR       <- function(x) ATR(HLC(x))[,'atr']
mySMI       <- function(x) SMI(HLC(x))[, "SMI"]
myADX       <- function(x) ADX(HLC(x))[,'ADX']
myAroon     <- function(x) aroon(cbind(Hi(x),Lo(x)))$oscillator
myEMV       <- function(x) EMV(cbind(Hi(x),Lo(x)),Vo(x))[,2]
myMACD      <- function(x) MACD(Cl(x))[,2]
myMFI       <- function(x) MFI(HLC(x),  Vo(x))
mySAR       <- function(x) SAR(cbind(Hi(x),Cl(x))) [,1]
myVolat     <- function(x) volatility(OHLC(x),calc="garman")[,1]
```

The Prediction Task and Data We Will Use

- The following code creates the objects with the data we will use

```
data.model <- specifyModel(T.ind(GSPC) ~ myATR(GSPC) + mySMI(GSPC) +
                 myADX(GSPC) + myAroon(GSPC) + myEMV(GSPC) +
                 myVolat(GSPC) + myMACD(GSPC) + myMFI(GSPC) + mySAR(GSPC) +
                 runMean(Cl(GSPC)) + runSD(Cl(GSPC)))

Tdata.train <- as.data.frame(modelData(data.model,
                                       data.window=c('1970-01-02','2005-12-30')))
Tdata.eval <- na.omit(as.data.frame(modelData(data.model,
                                       data.window=c('2006-01-01','2016-01-25'))))
Tform <- as.formula('T.ind.GSPC ~ .')  # the formula to be used in models
```

The Prediction Task and Data We Will Use (2)

```
head(Tdata.train)
```

```
##            T.ind.GSPC myATR.GSPC mySMI.GSPC myADX.GSPC myAroon.GSPC
## 1970-02-18 0.15215888   1.757594 -27.544696   48.67410          -30
## 1970-02-19 0.05307516   1.757766 -22.066236   45.56942          -30
## 1970-02-20 0.05120619   1.765782 -16.483777   42.76333          -25
## 1970-02-24 0.00000000   1.756084 -11.374860   39.93884          -20
## 1970-02-25 0.00000000   1.822792  -4.722482   37.88671           85
## 1970-02-26 0.00000000   1.835450   0.825322   35.98116           85
##              myEMV.GSPC myVolat.GSPC myMACD.GSPC myMFI.GSPC mySAR.GSPC
## 1970-02-18 0.0002233277    0.2144511   -2.124947   68.64115   85.02000
## 1970-02-19 0.0002792395    0.2151897   -1.976305   75.99052   85.02000
## 1970-02-20 0.0001116343    0.2181275   -1.818074   75.65063   85.16720
## 1970-02-24 0.0002632325    0.2184983   -1.658710   74.89682   85.38157
## 1970-02-25 0.0003729626    0.2296881   -1.478420   75.28083   85.66384
## 1970-02-26 0.0003360702    0.2291771   -1.299327   74.16197   86.07746
##            runMean.Cl.GSPC runSD.Cl.GSPC
## 1970-02-18          86.583     0.4582108
## 1970-02-19          86.769     0.5230780
## 1970-02-20          86.939     0.6298933
## 1970-02-24          87.037     0.7129286
## 1970-02-25          87.362     0.9422276
## 1970-02-26          87.558     1.0431437
```

Hands On Data Creation

- Data download and pre-processing
  1. Experiment with data downloading
     - Search for ticker IDs at Yahoo Finance
  2. Create different data sets for modeling (try different predictors and targets)
  3. Create a dynamic document that allows you to obtain a regular report on the evolution of the prices of some stock ticker during the last x days. Note: it should be easy to change (e.g. through variables in the beginning of the document) both the stock ticker and the value of x and thus being able to obtain a different report