Predicting Stock Prices

Riga workshop 28-29/11 1997
Thomas Hellström

- Common viewpoints
- Types of in and out data
- Prediction as Inductive Learning
- Performance evaluation

 1

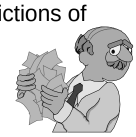
Thomas Hellström

- "Industrial" background:
- Ionospheric research at EISCAT
- Product development in my own company Seapacer AB
  - Optimisation and Control computers for ferries
  - Real time data analysis
- Teaching Artificial Intelligence at Umeå University
- Involved in the financial research project at Mälardalens högskola.

- The work I will present is done in collaboration with Kenneth Holmström

 2

What's so special about predictions of Stock time series?



⊖

- A hard problem! Is it even possible?
- Looks very much like random walk!
- The process is "regime shifting". The markets move in and out of periods of ''turbulence'', ''hause'' and ''baise''. Hard for traditional algorithms!
- The evaluation of predictability is extremely hard!

  When have we learned and when have we memorised?

⊕

- A successful prediction algorithm does not have to give predictions for all points in the time series.

  Can we predict predictability?

 3

1

Common viewpoints

- The efficient market hypothesis 

  The prices reflect ALL available information and new information is assimilated immediately.

  Implies a random walk. " Impossible to predict! "

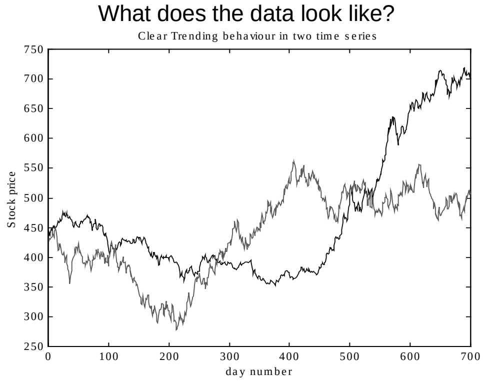
- Traders viewpoints 

  " Just a question of hard work and good intuition! "

  The market clearly goes through periods of positive and negative trends. It's just to identify the peaks and the troughs

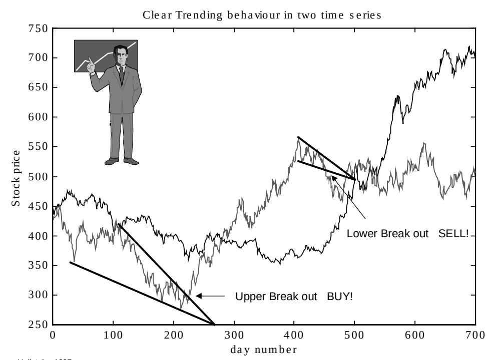
 4

What does the data look like?

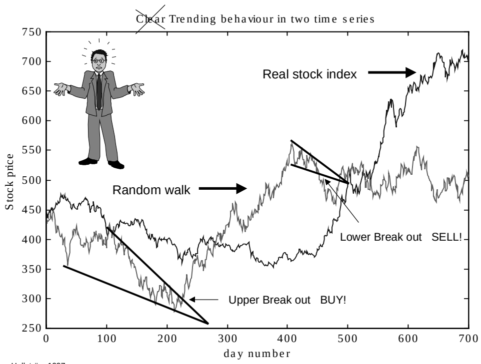


 5

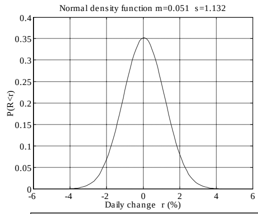
Technical analysis: Triangles



Ooops!



Does the Dow Jones index follow a random walk?



- Normal distribution is a consequence of pure random walk.
- Statistics for daily changes Dow Jones 1984-1996: Mean=0.05% Std. dev.=1.1
- Question: How often can we expect a crash like november 1987 (-28% in one day) ?

r	P(R<r)	Years between events	No. of real obs.
0	5.00E-01	0	1063
-1	2.00E-01	0	201
-2	4.00E-02	0	56
-3	4.00E-02	1	19
-4	2.00E-04	23	9
-5	4.00E-06	982	3
-6	5.00E-08	88244	3
-7	3.00E-10	20,000,000	1
-8	6.00E-13	7000,000,000	1
-9	7.00E-16	6000,000,000,000	1

8

Data in Technical analysis

- Close price
- Highest payed during day
- Lowest payed during day
- Volume (no. of traded stocks)

"tick" data sometimes available

9

Stock price (High Low Close) and Volume

2

10

Data in Fundamental analysis

1) The general economy
- inflation
- interest rates
- trade balance etc.

2) The condition of the industry
- Other stock's prices, normally presented as indexes.
- The prices of related commodities such as oil, metal prices and currencies
- The value on competitors stocks

11

Data in Fundamental analysis

3) The condition of the company

- p/e: Stock price divided by last 12 months earning per share
- Book value per share: Net assets (assets minus liabilities) divided by total number of shares
- Net profit margin: Net income divided by total sales
- Debt ratio: Liabilities divided by total assets
- Prognoses of future profits
- Prognoses of future sales

12

Derived entities

- k-day Returns:

$$R_k(t) = \frac{y(t) - y(t-k)}{y(t-k)} \approx \log\left(\frac{y(t)}{y(t-k)}\right)$$

- Moving average of order k:

$$mav_k(y) \equiv (z(1), z(2), \ldots, z(N))$$

$$z(t) \equiv \frac{1}{k}\sum_{i=1}^{k} y(t-i)$$

Derived entities

- Volatility (standard dev. of the log returns)

$$V = \sqrt{\frac{1}{N-1}\sum_{t=1}^{N} \ln\left(\frac{y(t)}{y(t-1)}\right) - m}$$

where

$$m = \sqrt{\frac{1}{N}\sum_{t=1}^{N} \ln\left(\frac{y(t)}{y(t-1)}\right)}$$

Inductive Learning

Given: A set of N examples $\{(x_i, z_i), i = 1, N\}$ and an unknown function f such that $f(x_i) = z_i \quad \forall i$

The task of **pure inductive inference** or **induction** is:
Learn a function g that minimises the norm of the error vector E: $|E| = |(e_1, \ldots e_N)|$
where $e_i = e(g(x_i), z_i)$
I.e: g should "approximate" f

Note:
The error function e and the norm |E| are still not defined.

3

Inductive Learning

The case of Prediction:
Examples: { (**X**(t), z(t+*h*)), t=1,N }
where *h* is the prediction horizon
Learn a function g that minimises $|E| = |(e_1, \ldots e_N)|$ where
$e_t = e(g(\mathbf{X}(t)), z(t+h))$

Specifying a Prediction problem

- "Inputs", i.e. The **X** vector
- "Output", i.e the z vector
- Error function $e(g(x), z)$
- Vector norm for computing $|E| = (e_1, \ldots e_N)$
- Bias for g (prior knowledge of g)

1) Standard Time series approach

Inputs: **X**(t) = (y(t),…,y(t-k+1))
Output: z(t+h) = y(t+h) where h is the prediction horizon
I.e Predict future prices with past prices

$$e_t = g(\mathbf{X}(t)) - z(t+h)$$

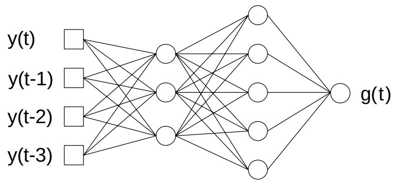
$$|E| = \sqrt{\frac{1}{N-h-k+1}\sum_{t=k}^{N-h} e_t^2} \quad \text{(RMSE)}$$

Typical choices of function g:

- $g(t) = \sum_{i=0}^{k} a_i\, y(t-i)$ AR-model
- g is a general non linear function Neural network

Feed-forward neural network

- Input layer with 4 inputs
- Two Hidden layers with 3 and 5 nodes
- Output layer with 1 output node



The weights w are selected to minimise

$$|E| = \sqrt{\frac{1}{N-4}\sum_{t=4}^{N-1} (g_w(t) - y(t+1))^2}$$

1) Standard Time series approach

Drawbacks:

- A stationary model is not realistic
- Fixed horizon not realistic. A profit 2 days ahead is as good as 1 day ahead.
- The MSE measure treats all predictions g, small as large as equal.

2) Pattern classification approach

Inputs: $\mathbf{X}(t) = (R_1(t), R_5(t), R_{10}(t), R_{20}(t))$

Output: $z(t+5) = R_5(t+5)$

$$e_t = \begin{cases} 1 & g(\mathbf{X}(t)) > \alpha \text{ AND } z(t) < 0 \\ 1 & g(\mathbf{X}(t)) < -\alpha \text{ AND } z(t) > 0 \\ 0 & \text{otherwise} \end{cases}$$

α is a trading threshold ≥ 0

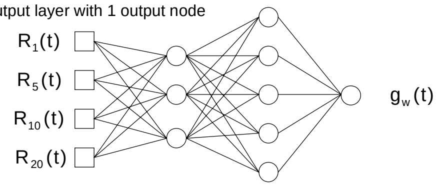
$$|E| = \sqrt{\frac{1}{N}\sum_{t=1}^{N} e_t^2}$$

g can be used in a trading rule T:

$$T(t) = \begin{cases} \text{buy} & \text{if } g(\mathbf{X}(t)) > \alpha \\ \text{sell} & \text{if } g(\mathbf{X}(t)) < -\alpha \\ \text{do nothing} & \text{otherwise} \end{cases}$$

Feed forward neural network

- Input layer with 4 inputs
- Two Hidden layers with 3 and 5 nodes
- Output layer with 1 output node



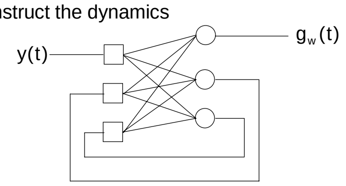
The weights w are selected to minimise:

$$|E| = \sqrt{\frac{1}{N-24}\sum_{t=20}^{N-5} e_t^2}$$

4

Recurrent neural network

- Feedback to input layer
- The hidden layer stores previous values and can reconstruct the dynamics



The weights w are selected to minimise:

$$|E| = \sqrt{\frac{1}{N-4}\sum_{t=4}^{N-1} (g_w(t) - y(t+1))^2}$$

Technical Indicators

- The tools for Technical trading
- Include principles such as:
  - The trending nature of prices
  - Volume mirroring changes in price
  - Support/Resistance
- Examples:
  - Moving averages
  - Formations such as triangles
  - RSI - the relation between the average upward price change and the average downward price change within a time window normally 14 days backwards

Technical Indicators

- Can often be described as a trading rule:

$$T(t) = \begin{cases} \text{buy} & \text{if } g(\mathbf{X}(t)) > \alpha \\ \text{sell} & \text{if } g(\mathbf{X}(t)) < -\alpha \\ \text{do nothing} & \text{otherwise} \end{cases}$$

where $\mathbf{X}(t) = (y(t),\ldots,y(t-k+1))$

- Example:

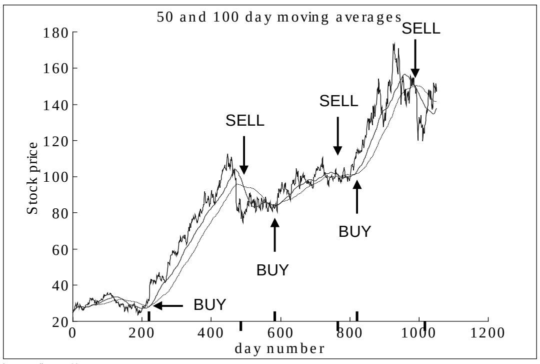
$mav_k(y) \equiv \{z(1), z(2), \ldots, z(N)\}$

$$z(t) \equiv \frac{1}{k}\sum_{i=0}^{k-1} y(t-i)$$

$g \equiv \Delta(\text{sign}(mav_{50}(y) - mav_{100}(y)))$

$\Delta v(t) \equiv v(t) - v(t-1)$

$$g = \Delta(\ \text{sign}(\text{mav}_{50}(y) - \text{mav}_{100}(y))\ )$$



Relative stock performance

- Portfolio management
  - Minimise the variance in a portfolio by quadratic programming
  - Also possible with single stock methods by:
    - Computing relative stocks:

$$y_1^R(t) = y_1(t) / \sum_{i=1}^{K} y_i(t) / K$$

- Ranking stock returns:

  The stock with highest R gets rank 1:

$$\text{Rank}_k(t) = 1 + \left| \{ R_i(t) \mid R_i(t) > R_k(t), i = 1..N \} \right|$$

Benchmarks

- Naive prediction of stock prices:

  $y'(t)=y(t-1)$
- Naive prediction of returns:

  $R'(t) = R(t-1)$

The naive predictors are local minimum in many models e.g AR-models (but also Neural Networks):

$$y'(t) = \sum_{i=1}^{K} a_i\, y(t-i)$$

- Buy and hold:

  Buy at day 1 and sell at day N

5

Performance measures

- Theil coefficient:

  Compares the RMSE (root mean square error) for our predictions with the naive price predictions

  Predicting {y(t), t=1,N} with {y´(t), t=1,N}

$$T = \frac{\sqrt{\sum_{i=1}^{N} (\ y(t) - y'(t)\ )^2}}{\sqrt{\sum_{i=1}^{N} (\ y(t) - y(t-1)\ )^2}}$$

T<1 for real predictive power

Performance measures

- Directional prediction “Hit rate”

  Predicting {R(t), t=1,N} with {R’(t), t=1,N}

$$H = \frac{\left| \{ t \mid R(t)R'(t) > 0,\ t = 1,N \} \right|}{\left| \{ t \mid R(t)R'(t) \neq 0,\ t = 1,N \} \right|}$$

  For the naive return predictor:

$$H_N = \frac{\left| \{ t \mid R(t)R(t-1) > 0,\ t = 1,N \} \right|}{\left| \{ t \mid R(t)R(t-1) \neq 0,\ t = 1,N \} \right|}$$

- Normalised hit rate:

$$H_0 = \frac{H}{H_N}$$

H_0 <1 for real predictive power

Performance measures

Mean profit per trade:

- Trading rule approach:
  - “Run” the trading and compute the mean profit
- Time series approach:

$$\text{Mean profit} = \sum_{t=1}^{N} \text{sign}(\ y'(t) - y(t-1)\ )\ (y(t) - y(t-1)) / N$$

I.e:
A trade is assumed at every time step, in the direction of the predicted change.

Evaluating performance

What is a reasonable goal?

- Efficient market hypothesis implies random walk which is impossible to predict!
- The ACF has very low values
- Nearest neighbour analysis shows very low correlation
- There are so few $100 notes laying around!
- Published research (with proper evaluation) often shows about 54% hit rate.
- Even 54% real hit rate is enough to make a fortune!
- Compare with a casino: They don't know what number comes up next, they just improve the odds by adding the 0 and 00

Evaluating performance

- We are predicting a stock with equal numbers of moves up and down during one year of 250 trading days.
- Apply a totally random prediction algorithm on each day
- What is the probability that the hit rate>54% ?

The distribution for number of hits is given by:

$$P(H = x) = \binom{250}{x} 0.5^{x}\ 0.5^{250-x}$$

$$P(H > x) = 1 - P(H \le x) = 1 - \mathrm{bincdf}(x,250,0.5)$$

x=0.54*250=135 gives P(H>135)=0.092

I.e.There is a 9% risk that a random algorithm gives 54% hit rate.

Evaluating performance

- We want to compare 100 indicators that each produce Sell and Buy signals on average once a week. The test period is 10 years! We demand 55% hit rate!
- Apply 100 totally random prediction algorithm on each week.
- The probability that any one of them gets exactly x hits is:

$$P(H > x) = 1 - P(H \le x) = 1 - \mathrm{bincdf}(x,500,0.5)$$

P(H>0.55*500)=0.0112

The probability that ANY of the 100 indicators produce 55% hit rate is 1-minus the probability that all are less then 55%:

$$1 - (1 - 0.0112)^{100} = 0.68$$

How do know when we have learned?

6

Evaluating performance

Algorithm evaluation is a part of the learning process!

- It must be done "in sample" and not on the test set. Best: A final test on data that didn't exist at the time of the development of the algorithm
- It is sensitive to "over training".
- Be aware of the data-snooping problem!

Results so far

Used methods

- Artificial Neural Networks
- Fuzzy rule bases
- State space reconstruction and local models
- k nearest neighbour techniques
- Adaptive AR
- Hundreds of technical indicators

Results:

- No statistically significant predictions
- Significant seasonal patterns in data

Future work

- Finding regions with predictability
- How do we know that we have learned?
- Fundamental analysis much easier?
  - Problem: lack of huge amounts of data
- Other methods