# Numerical experiments on resistance optimal strategy - Part II

Training in Industry - Mathmods - INRIA, Project TOSCA

Camilo A. García Trillos
camilo.garcia@mathmods.eu

Advisor: Etienne Tanré

July 15th, 2009

# Outline

1. Introduction
2. Implementing the strategy
   - Resistance detection
   - Parameter estimation
   - Simulated data tests
3. Working with real data
   - Resistance in real series
4. Conclusions and further research
5. C++ library

# Our goal

**Has the derived mathematical strategy presented in the first part of the talk the potential to become a real-life-useful algorithm?**

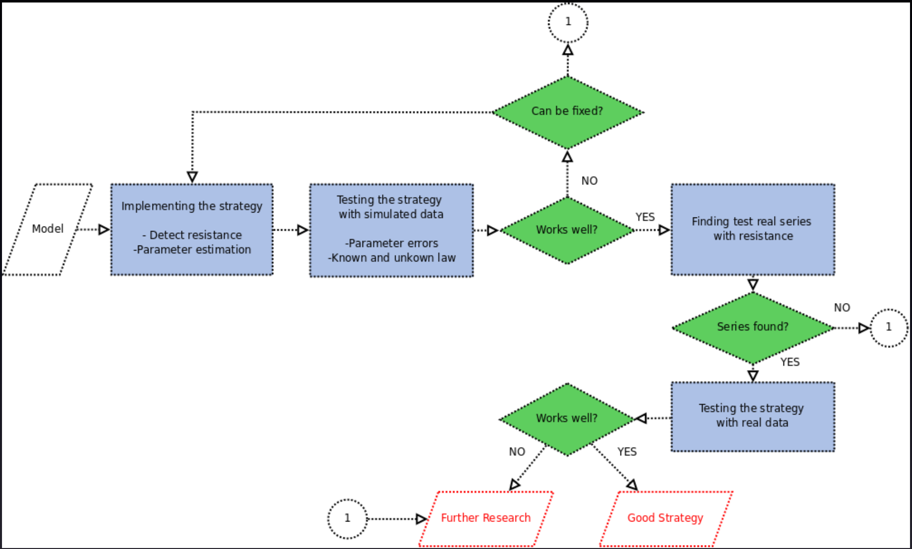
## Question

- Can we define methods for estimate the needed parameters and detecting a resistance level?
- Are these methods useful in real life (precision, speed, data requirements)?
- Is the financial series considered "sufficiently" modeled by the resistance equations?

## Method

- Propose definitions and algorithms for implementing the optimal strategy
- Test the implementation extensively with *simulated data* and controlled unknowns.
- Search and test the algorithm with *real data* that show resistance behavior.

# Methodology



# What is a resistance level?

**Investopedia** "The price at which a stock or market can trade, but not exceed, for a certain period of time. "

**Farlex Financial Dictionary** "(...) When the security approaches the resistance level, it is seen as an indication to sell the security, which will increase the supply, causing the security's price to fall back below the resistance level. If there are too many buyers, however, the security rises above the resistance level."

# Finding resistance levels

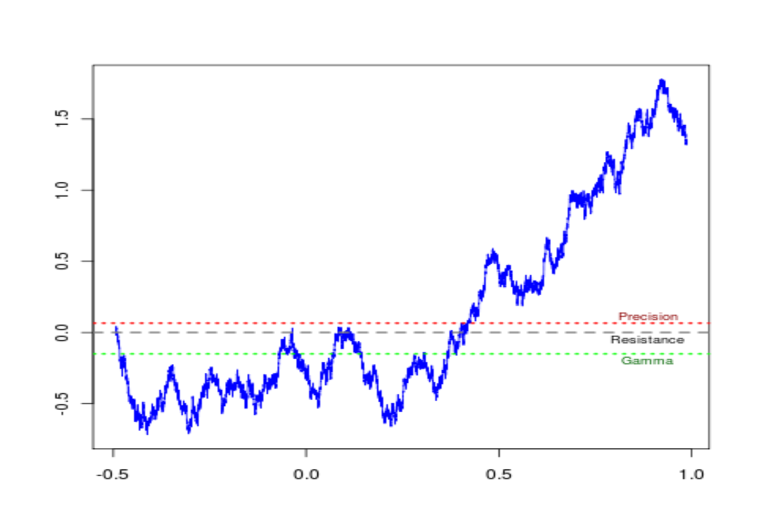
- For technical analyst there are many ways to identify a resistance level (bar chart statistics, trend lines,Fibonacci coefficients,... ).
- We identify a resistance level **as a level that is a maximum in a given time interval and is reached more than once during that interval (up to a given small precision)**.

**Proposed resistance detection algorithm**

1. Suppose the resistance line corresponds to the last available data
2. From this point look back for the nearest local maximum
3. If both points are closer than a given precision ($Z_{\min}$), and there is at least one point in between below a band level ($\gamma$), we count it as a *touch*
4. Iterate until the desired number of *touches* is reached (detected resistance) or the end of the time window (not detected).

# Resistance detection

Example



# Resistance detection

## Results

To test the algorithm we performed 1000 simulations with the following parameters:

| $\mu$ | $\sigma$ | $\alpha$ | $\epsilon$ | $Z_{\text{min}}$ | $\gamma$ | Touches |
|---|---|---|---|---|---|---|
| 0.925 | 0.15 | 0.5 | 0.001 | 0.015 | 7.23 | 2 |

Results:

- 100% detection rate
- In 71.05% of the cases the identified resistance level was the modeled one (100).
- Average detected 98.84.

# Parameter estimation

The paper's optimal strategy relies on the knowledge of the parameters that define a particular instance of the model ( $\sigma, \mu_0, \alpha$ and $\epsilon$).

We have to take into account that

- Estimation accuracy will depend in general on the number of points available or on the time length of the estimation window
- But the parameters for real life series are not constant in time.

To try to include both conditions, we will assume constant coefficients for a moving time window of fixed size.

The size of the time window should be neither to small or too large (the actual size will depend on each time series).

# Parameter estimation

Time window $[t_i, t_f]$ with $N$ samples $t_1, t_2, \ldots, t_N$, $t_1 = t_i, t_N = t_f$.
Recalling that $\Delta X_t$ ($X_t = \frac{1}{\sigma} \log(Z_t/S_0)$) we define:

- $\hat{\epsilon} := \frac{1}{\sigma} \log(1 + Z_{\min}/S_0)$ where $Z_{\min}$ represents the price precision (maximum change in value that is perceived by the market as being the same price).

# Parameter estimation

Time window $[t_i, t_f]$ with $N$ samples $t_1, t_2, \ldots, t_N$, $t_1 = t_i, t_N = t_f$. Recalling that $\Delta X_t$ ($X_t = \frac{1}{\sigma} \log(Z_t/S_0)$) we define:

- $\hat{\epsilon} := \frac{1}{\sigma} \log(1 + Z_{\min}/S_0)$ where $Z_{\min}$ represents the price precision (maximum change in value that is perceived by the market as being the same price).
- $\hat{\sigma}$ We use the standard deviation of $\log(Z_t/S_0)$. This corresponds to the usual unbiased least square estimator (LSE) for this case.

# Parameter estimation

Time window $[t_i, t_f]$ with $N$ samples $t_1, t_2, \ldots, t_N$, $t_1 = t_i, t_N = t_f$.
Recalling that $\Delta X_t$ ($X_t = \frac{1}{\sigma}\log(Z_t/S_0)$) we define:

- $\hat{\epsilon} := \frac{1}{\sigma}\log(1 + Z_{\min}/S_0)$ where $Z_{\min}$ represents the price precision (maximum change in value that is perceived by the market as being the same price).
- $\hat{\sigma}$ We use the standard deviation of $\log(Z_t/S_0)$. This corresponds to the usual unbiased least square estimator (LSE) for this case.
- $\hat{\mu}$ and $\hat{\alpha}$. Estimated simultaneously. Define $\hat{\mu}_\alpha$ as the LSE estimator for $\mu$ in $X_t$ excluding the down-crossing phase, for a fixed $\alpha$. Then choose the couple $\hat{\alpha}$, $\hat{\mu}_{\hat{\alpha}}$ that minimize the square error.

# Parameter estimation

## $\hat{\mu}$ and $\hat{\alpha}$ estimation

**Algorithm**

1. Identify the minimum of the series in the time window
2. For each $\alpha$ the corresponding $\mu_\alpha$ is calculated simply as

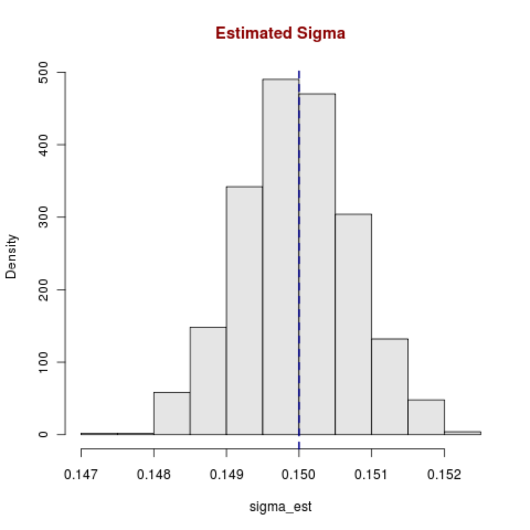
$$\mu_\alpha = \sum \alpha / T_\alpha$$

   where $T_\alpha$ is the time that the path spent in reaching 0 from $-\alpha$, and the sum includes all the up-crossings.
3. Define the residual as the sum of the squares of the errors between $\Delta X_t$ and the $\Delta \hat{X}_t$ estimated with the researched parameters.
4. Find $\alpha$ that minimizes the residual with an iterative method (Regula Falsi, dichotomous search, etc.)

# Estimation results

$\sigma$ estimation

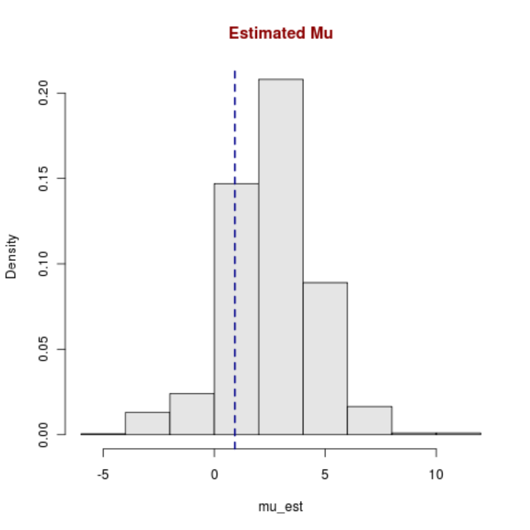
| | |
|---|---|
| mean est. | 0.1498 |
| real | 0.15 |
| relative bias | -0.15 % |



# Estimation results

$\mu$ estimation

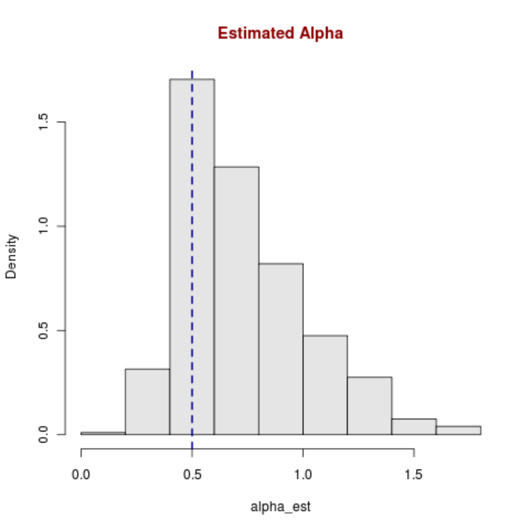
| | |
|---|---|
| mean est. | 0.779 |
| real | 0.93 |
| relative bias | -15.7 % |



# Estimation results

$\alpha$ estimation

| | |
|---|---|
| mean est. | 0.868 |
| real | 0.5 |
| relative bias | 73.61 % |



# Simulated data test

We want to obtain information regarding how well the implementation of the strategy behaves with respect to the logarithm of the wealth obtained.

Again, we will consider the case of a self financed portfolio consisting of two assets, one risky and one riskless.

For evaluating different error causes, we test the results when:

- The down-cross probability law is known but inexact parametrization occurs.
- We estimate the parameters and the probability law is known
- The parameters are known but an inexact down-cross probability law is used
- We estimate the parameters and an inexact down-cross probability is used.

# Simulated data test

Parameter of the simulation

Parameters:

| Parameter | Value |
|---|---|
| $\mu_0$ | 0.15 |
| $\sigma$ | 0.15 |
| $r$ | 0.02 |
| $\alpha$ | 0.5 |
| $\epsilon$ | 0.066 |
| Information days | 180 |
| Test days | 180 |

Down-crossings law:

| N | Probabilty |
|---|---|
| 0 | 0.1 |
| 1 | 0.2 |
| 2 | 0.3 |
| 3 | 0.2 |
| 4 | 0.1 |
| 5 | 0.1 |

In all the simulations, we will assume we count with the historical information for *information days*. The estimated parameters and resistance will be used for *test days*. The wealth is evaluated for the latter.

# Simulated data tests

## Known parameters and resistance

- Unbounded strategy: The investment may be any real number (any short or long position). Avg. time x iteration: 69.88 ms

$\log(Wealth)$ in (%)

| Classical | Tech. An. | Optimal Resist. |
|---|---|---|
| $15.34 \pm 4.13$ | $2.50 \pm 0.45$ | $36.74 \pm 4.54$ |

- Bounded strategy: We assume the possible invested amount in the risky asset belongs to $[0, 1]$. Avg. time x iteration: 259.57 ms

$\log(Wealth)$ in (%)

| Classical | Tech. An. | Optimal Resist. |
|---|---|---|
| $3.88 \pm 0.67$ | $2.49 \pm 0.48$ | $5.61 \pm 0.68$ |

# Simulated data tests

Estimated parameters - Known law - Bounded strategy

- Known resistance: Assuming the resistance value used for simulating the path. Avg. time x iteration: 665.36 ms

log($Wealth$) in (%)

| Classical | Tech. An. | Optimal Resist. |
|---|---|---|
| $3.75 \pm 0.57$ | $2.40 \pm 0.46$ | $4.42 \pm 0.55$ |

- Detected resistance: Using the detection algorithm. Avg. time x iteration: 538.22 ms

log($Wealth$) in (%)

| Classical | Tech. An. | Optimal Resist. |
|---|---|---|
| $3.48 \pm 0.58$ | $2.63 \pm 0.47$ | $3.84 \pm 0.57$ |

# Simulated data tests

Estimated parameters - Unkown law - Bounded strategy

- Law with similar mean: Assuming the law is $[0, 0, 0.7, 0.3, 0, 0]$. Avg. time x iteration: 567.17 ms

$\log(\mathit{Wealth})$ in (%)

| Classical | Tech. An. | Optimal Resist. |
|---|---|---|
| $3.11 \pm 0.58$ | $2.37 \pm 0.45$ | $4.22 \pm 0.56$ |

- Different law: Assuming the law is $[0.5, 0.3, 0.2, 0, 0, 0]$. Avg. time x iteration: 532.96 ms

$\log(\mathit{Wealth})$ in (%)

| Classical | Tech. An. | Optimal Resist. |
|---|---|---|
| $3.65 \pm 0.56$ | $2.46 \pm 0.45$ | $3.79 \pm 0.55$ |

# Resistance in real series

- As a first step for testing the strategy with real series, we need to find a financial asset whose price may be well fitted by the given model.
- We test using the detection algorithm with different parameters.
- It is necessary to find resistance that show an important number of *touches*. Otherwise the strategy will not give much information.
- To test this situation we use the already presented resistance detection algorithm in a set of stocks and commodities prices and an index.

# Resistance in real series

## Some results

We tested different assets using adjusted daily close prices between January 1990 and June 2009.

In each case we estimated the precision as the 10% percentile of the absolute change in value, and $\gamma$ as the average of those changes.

| Asset | Type | 2 Touches (%) | 3 Touches (%) |
|---|---|---|---|
| Dow Jones | Index | 23.73 | 0.935 |
| WTI (oil) | Commodity | 38.19 | 0.234 |
| GE | Stock | 14.09 | 1.402 |
| IBM | Stock | 16.66 | 0.212 |
| CocaCola Co. | Stock | 24.5 | 2.932 |
| Pfizer | Stock | 19.20 | 0.595 |

# Conclusions

- We were able to implement the derived optimal strategy for resistance presence.
- In the numerical tests run over simulated data, the algorithm outperformed the results of the optimal strategy for a classical Black an Scholes model, and those of applying a moving average with resistance (technical analysis technique).
- In particular, we obtain fiarly good results when the parameters are unknown and there is no prior knowledge of the resistance level. Improvements in the estimation technoiques may increase the results even more.
- We could not find a financial series which shows enough presence of resistance to test the algorithm.
- Therefore, although the implementation of the algorithm is possible we believe its use in a real world environment would be limited.

# Further research

- To extend the applicability of the strategy, it would be interesting to generalize the derived mathematical model to **resistance lines** in which the resistance is allowed to show a slope. This model may be applied to a wider range of financial series.
- Some improvements in the parameter estimation when applied to real life series may be incorporated to increment the performance (Ex: considering high and low levels instead of close levels may improve estimation of some parameters).

# C++ library

As part of our work we created a C++ library that helps in the process of performing numerical experiments for stochastic processes which contains mainly:

- A class for modeling stochastic paths
- Function for performing basic operations: shifts and scales in time, multiplication, addition, exponential, logarithm
- Functions for saving and loading
- Functions for finding maxima and minima
- Implementations of some technical analysis tools: finding resistance, moving average, parameter estimation