REAL-TIME STOCHASTIC PORTFOLIO OPTIMIZATION

Ph.D. thesis booklet of

SIPOS ISTVÁN RÓBERT, M.SC.

Supervisor:

Dr. Levendovszky János, D.Sc.
Doctor of the Hungarian Academy of Sciences

Department of Networked Systems and Services
Budapest University of Technology and Economics

Budapest, 2015

1. Motivation

In modern finance, algorithmic trading plays a key role. A widespread set of mathematical models and methods supports the needs for efficient and a profitable trading. However, fast and profitable trading often proves to be difficult in the case of high number of assets, because the profit is considerably decreased by the transaction costs. On the other hand, to develop and run efficient trading algorithms in real-time requires striking a good balance between algorithmic complexity and runtime speed. Therefore one of the central issues of modern portfolio theory is to optimize portfolios subject to cardinality constraint (which specifies the number of non-zero components not to exceed a given threshold) in real-time. This provides sparse portfolios to minimize the transaction costs and to maximize interpretability of the results. In this light, the main objectives of my dissertation are as follows:

- optimizing mean reverting portfolios by new numerical approaches and subject to a novel objective function (maximizing the expected mean return instead of predictability);
- choosing an optimal portfolio by hidden Markov modeling of the portfolio value time series;
- further maximizing the profit by extending the underlying model to AR-HMM;
- extensive speed profiling of the demonstrated algorithms.

The above mentioned modeling approaches aim to capture the underlying characteristics of the observed financial time series. Based on the identified portfolio and the predictions given by the model, the applied trading algorithm triggers the appropriate trading signal. Furthermore, the dissertation takes secondary effects into account (like the bid-ask spread), which adapts the analysis to realistic market scenarios.

In order to evaluate the novel methods, I developed an extensive computational and back-testing framework for detailed performance analysis.

2. Introduction

In this section the theoretical background and its challenges, and the novel contributions are summarized.

2.1. State-of-the art and open questions

Portfolio optimization was first investigated by Markowitz [27] in the context of diversification. Investing into a portfolio instead of a single asset allows one to optimize the return according to various objectives. Modern portfolio theory aims to minimize the associated risk and maximize the expected return by portfolio selection. While there is an evident trade-off between the risk and profit, in mathematical finance researchers used to seek for a portfolio which maximizes its predictability. Unfortunately, finding the most profitable criterion and performing the corresponding portfolio optimization is a rather challenging job, which prompts further investigations in this field.

Mean reversion is a good indicator of predictability, as a result, identifying mean reverting portfolios has become a key research area (d'Aspremont[11], Fogarasi and Levendovszky[15]). Assuming that the asset price vector follows a VAR(1) process, portfolio optimization can be reduced to solving a generalized eigenvalue problem. D'Aspremont [11] analyzed the problem of finding mean-reverting portfolios with cardinality constraints, resulting in sparse portfolios to minimize the transaction costs. However, with this constraint, the optimal portfolio selection becomes NP hard (Natarjan[31]).

While trading with mean reverting portfolios, we need to make a decision on whether the observed process exhibits mean reverting properties, or following a Brownian motion or even driven by mean aversion. Since these properties can change over time, a more flexible time dependent distribution model is needed.

HMMs are widely used for predictions (Durbin *et al.*[12], Hassan and Nath[18], Jurafsky and Martin[23], Mamon and Elliott[25]), however, efficient algorithms have not yet been developed for high speed and quality training of the free parameters, which are appropriate for the purpose of high-frequency algorithmic trading (Hassan *et al.*[19]). In order to optimize the parameters of the HMM, traditionally the Baum-Welch expectation maximization (Baum *et al.*[6]) algorithm is used. Unfortunately, although it has proven to be a computationally efficient algorithm, it can get stuck in local optima. Thus, the quality of the achieved optimum is based on the initialization.

2.2. A brief summary of novel contributions

In the first thesis group, I have developed a fast approximation method for efficient optimal portfolio selection by using FeedForward Neural Networks (FFNN). Furthermore - as an alternative objective function to maximizing the predictability of mean reverting portfolios - I maximized the average return on this type of portfolios. This is carried out assuming the underlying marginal probability density functions of the OU processes. In order to estimate the parameters of OU processes I have used multiple methods and their performance was compared. The novel portfolio identification methods with corresponding trading strategies have been tested numerically on real financial time series, and the results exhibit profits.

The second group of theses aims to overcome the limitations of the Baum-Welch algorithm. First, I provided good quality solutions by the use of simulated annealing. Furthermore, in order to increase the efficiency of the underlying stochastic search, I also introduced a hybrid method for training the HMMs. To further ease the complexity, I introduced clustering and dimension reduction methods (PPCA) as well, to reduce the size and the dimension of the search space. The performance of prediction by the HMMs was investigated on financial time series, which demonstrated that a good average return can be obtained.

As I demonstrated in [1], linear regression based methods proved to be sufficient for OU parameter estimation. However, the shortcomings of this method are that an even higher accuracy would be desired for the purpose of algorithmic trading, and also they are unstable under certain circumstances (e.g. when the level of mean reversion is low).

The novel approach, which is introduced in the thesis, manages to overcome the above mentioned problems and provides a more general model which can handle heterogeneous time series, and gives more accurate and stable predictions for the future portfolio prices.

The new contribution of the third group of theses lies in the following points: (i) instead of the predictability parameter the average return is maximized; (ii) instead of the OU modeling, autoregressive HMM (AR-HMM) is used. In addition to that, a multi state AR-HMM can be considered as a generalization of these PDFs, which enables to perform prediction based algorithmic trading on financial time series. The performance analysis shows that the results exhibit good average returns.

3. Models and methods

In the dissertation I used modeling, analytical and simulation tools (summarized by table 1.).

Method	Use in dissertation	Theses
Analytical	Maximizing predictability	I.1.
Modeling	VAR(1) OU HMM AR-HMM	I. I., III. II. III.
Algorithm	FFNN SA Baum-Welch EM PPCA Clustering	I.1. I.1-2., II.1., III.2. II.1. II.3. II.2.
Simulation	Performance analysis	For each thesis

Table 1. Methods used in the dissertation

Throughout the dissertation, many stochastic models have been introduced which require different parameter identification methods, as detailed by the following points:

- VAR(1): least squares and maximum likelihood estimations;
- Ornstein-Uhlenbeck SDE: sample mean, linear regression based, recursive, mean squared and pattern matching estimation techniques;
- FFNN: back propagation algorithm;
- HMM: Baum-Welch expectation maximization algorithm, simulated annealing and a novel hybrid training method;
- PPCA: an iterative expectation maximization algorithm;
- Clustering with GMM: an expectation maximization algorithm;
- AR-HMM: Baum-Welch expectation maximization algorithm.

Having identified the parameters of the specific model I applied, one can then optimize the portfolio, which leads to:

- solving a generalized eigenvalue problem;
- simulated annealing;
- HMM based prediction methods.

In each case, the objective is to maximize a given performance evaluation function, which in turn results in the optimal portfolio. Then, the identified portfolio is converted to a trading signal for taking the appropriate trading actions. In the phase of performance analysis, various numerical indicators are evaluated for the sake of comparing the profitability of different methods (shown by Fig. 1.).

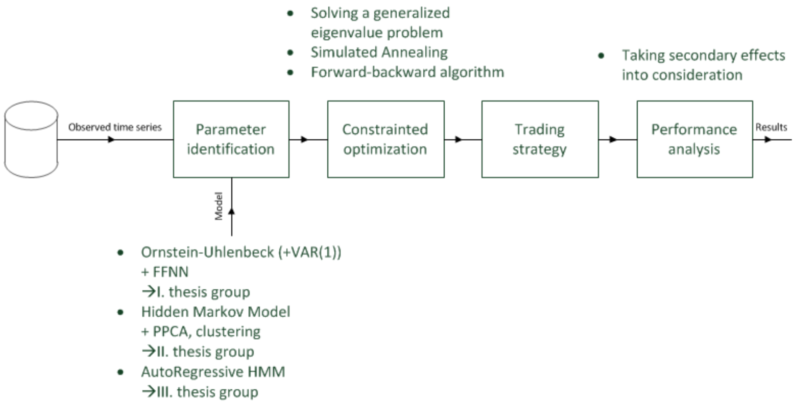

Fig. 1. Computational approach

For the purpose of performance analysis and for the comparison of the different methods detailed in the theses, I have developed numerous models and algorithms in MATLAB®. These algorithms are parts of an extensive simulation framework. The framework can also accommodate real financial time series to obtain back-testing results for verifying profitability.

4. New scientific results

In this section I briefly outline the novel contributions of the dissertation. As was mentioned before, the results are centered around the following topics:

- optimizing mean reverting portfolios;
- choosing an optimal portfolio by hidden Markov modeling;
- further maximizing the profit by extending the underlying model to AR-HMM;
- massively parallel GPGPU implementation of the demonstrated algorithms.

4.1 New results on mean reverting portfolios

I. group of theses - trading with optimal mean reverting portfolios subject to cardinality constraints:

I have developed novel approaches for portfolio optimization, namely a new objective function - maximizing the mean return - and a corresponding trading algorithm have been introduced. The optimization itself is carried out by stochastic search algorithms and FeedForward Neural Networks (FFNNs). By this method a better profit can be achieved as compared to the traditional ones.

(Related publications: [1])

The new results are achieved according to the following notations. The time series describing the prices of assets is denoted by $\mathbf{s}_t^T = \left(s_{1,t},...,s_{n,t}\right)$ where $s_{i,t}$ is the price of asset *i* at time instant *t*. The portfolio vector is denoted by $\mathbf{x}^T = \left(x_1,...,x_n\right)$ where x_i gives the number of possessed quantity from asset *i*. The value of the portfolio at time *t* is denoted by $p(t)$ and defined as

$$p(t) = \mathbf{x}^T \mathbf{s}_t = \sum_{i=1}^{n} x_i s_{i,t} \ . \qquad (1.1)$$

Our objective is to find the optimal portfolio $\mathbf{x}_{opt}$ which maximizes a pre-defined objective function, such as average return, subject to cardinality constraint which specifies that the number of non-zero components in $\mathbf{x}_{opt}$ must not exceed a given number *l*. The optimal portfolio is sought under the assumption that the portfolio value $p(t)$ exhibits mean reverting properties, i.e. it follows an Ornstein-Uhlenbeck (OU) process (Ornstein and Uhlenbeck[32]). This is a frequent assumption in trading (Fama and French[13], Manzan[26], Ornstein and Uhlenbeck[32], Poterba and Summers[33]), as the linear combination of individual assets shows a much better predictability (see Fig. 2).

The OU process is characterized by the following stochastic differential equation (Ornstein and Uhlenbeck[32])

$$dp(t) = \vartheta\left(\mu - p(t)\right)dt + \sigma dW\left(t\right), \qquad (1.2)$$

where W(t) is a Wiener process (Doob[39]) and $\vartheta > 0$ (mean reversion coefficient), μ (long-term mean) and $\sigma > 0$ (volatility) are constants. One can obtain its solution with the Itō-Doeblin formula (Itō[21]), which implies that $\mathbf{E}\left(p(t)\right) = \mu(t) = p(0)e^{-\vartheta t} + \mu\left(1 - e^{-\vartheta t}\right)$.

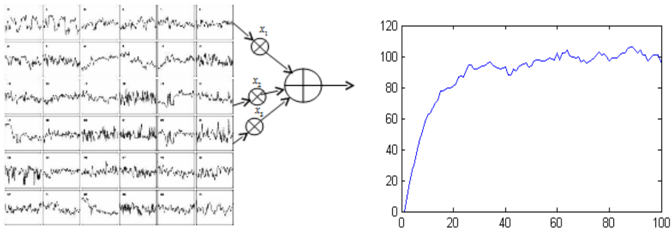

Fig. 2. Individual assets with bad predictability, and their linear combination following an OU process

Parameter θ determines the convergence speed of the process towards the mean, and inversely indicating the level of uncertainty via the standard deviation of its asymptotic distribution.

I.1. I enhanced the performance of the previous methods by combining a FeedForward Neural Network (FFNN) with simulated annealing (SA).

In this method the output of the trained FFNNs was used as the starting point of the simulated annealing.

To estimate the optimal portfolio vector from the identified VAR(1) matrices, we can utilize FFNNs, as they have universal representation capabilities in L^2 (Cybenko[9]). Furthermore, they can learn and generalize from a finite set of examples $\tau^{(K)} = \left\{\left(\mathbf{z}_k, \mathbf{d}_k\right), k = 1, \ldots, K\right\}$ by using the back propagation algorithm (Hagan and Menhaj[16]). As a result, one may look upon the optimal portfolio selection problem as a mapping from the identified, $\mathbf{A}, \mathbf{G}, \mathbf{K}$ matrices of the underlying VAR(1) process to the optimal sparse portfolio vector $\mathbf{x}$. In this case, the input vector of FFNN is $\mathbf{z} = \left(\mathbf{A}, \mathbf{G}, \mathbf{K}\right)$ constructed by matrix flattening and the output is vector $\mathbf{x}$ with $card\left(\mathbf{x}\right) \le l$ to fulfill the sparsity constraint.

One can construct the training set $\tau^{(K)}$ by finding the optimal sparse portfolio vectors for some input matrices by exhaustive search. The construction of the training set is done according to the following computational model (Fig. 3).

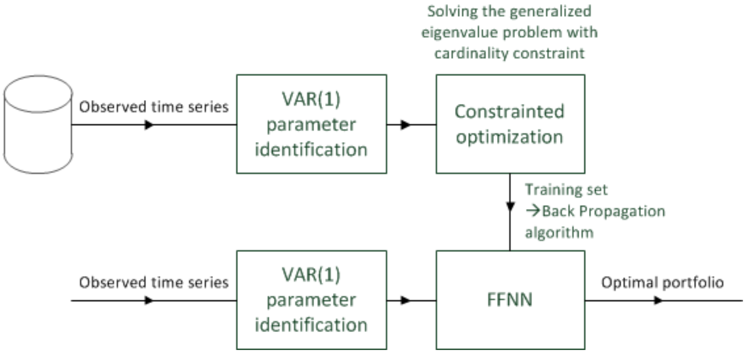

Fig. 3. Training and using FFNNs for portfolio optimization

The neural network had three hidden layers, which can approximate any continuous function on $\mathbb{R}^N$ according to the universal approximation theorem (Cybenko[9]) .

However, instead of maximizing the predictability a more general objective function having a direct impact on the trading profit can be introduced.

I.2. Instead of the predictability I have introduced a new objective function, the average return. By maximizing this objective function a better profit can be achieved.

The maximization problem can be posed as follows:

$$\mathbf{x}_{\text{opt}} = \arg\max_{\mathbf{x}} \Psi(\mathbf{x})$$
$$\Psi(\mathbf{x}) = \max_{0\le t} E\left(p(t)\right) - p(0) = \max_{0\le t}\left(\mu(t) - p(0)\right) = . \qquad (1.3)$$
$$\max_{0\le t}\left(\left(p(0)-\mu\right)e^{-\vartheta t} + \mu - \mathbf{x}^T\mathbf{s}_0\right)$$

Furthermore, another objective function can be obtained if we take into account that a risk free interest is available with interest rate r_f , which allows discounting the expected future portfolio value over time. In this way, we have the following expression by replacing the future value with its net present value:

$$\Psi^*(\mathbf{x}) = \max_{0\le t} \frac{E\left(p(t)\right)}{(1+r_f)^t} - p(0), \qquad (1.4)$$

where the optimal solution for t is given as

$$t=\frac{1}{\vartheta}\ln\left(\frac{\left(\mu-p(0)\right)\left(\vartheta+\ln\left(1+r_f\right)\right)}{\mu\ln\left(1+r_f\right)}\right). \qquad (1.5)$$

By using (1.3) or (1.4), portfolio optimization can again be reduced to a constrained optimization problem:

$$\mathbf{x}_{\text{opt}}=\arg\max_{\mathbf{x}}\Psi\left(\mathbf{x}\right), card\left(\mathbf{x}\right)\leq l\ . \qquad (1.6)$$

However, (1.6) does not have a known analytical solution, therefore I used stochastic search methods to find the optimal solution.

I.3. Having the optimal portfolio at hand, I developed a trading algorithm to carry out the corresponding trading.

In the proposed trading algorithm, trading is described as a walk in a binary state space in which either we already have a portfolio at hand or cash at hand. The transitions between the two states are only affected by the evaluations of the potentially owned and the newly identified optimal portfolio by (1.3) or (1.4). The trading algorithm is then formalized by a state chart (Fig. 4).

Positive evaluation indicates a profitable portfolio, while negative evaluation indicates that the portfolio may produce a loss. Based on this, the agent buys or holds a portfolio only if it has a positive evaluation. A new trading action is taken if a newly identified portfolio (`x_opt`) has higher and also positive evaluation than the present one (`x`). In this case one can sell the owned portfolio and buy the new one with higher expectations instead. This approach treats the present portfolio as a sunk cost, thus only the future expectations are taken into consideration. Hence, we do not have to give up the best available portfolio in favor of a presently unfavorable portfolio.

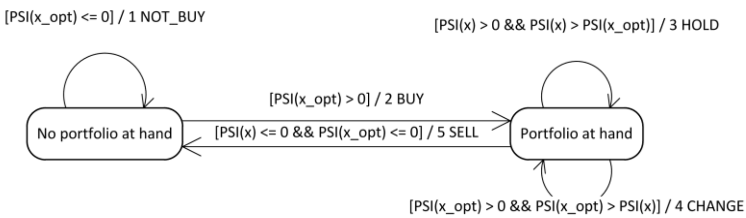

Fig. 4. Trading algorithm

Performance analysis

An extensive back-testing framework was created to provide numerical results for comparison of different methods on different financial time series. The appendix contains further descriptions of the used data sets and introduced measures for performance analysis.

In this period, the U.S. SWAP rates had a decreasing tendency with simple buy and hold strategy (-12.11% on average). The bar chart (Fig. 5) shows that all of the introduced methods outperform this tendency, and in the scenario when a FFNN was deployed the trading was profitable with a 13.49% yearly profit.

In the studied period the S&P 500 asset prices rose only by 12.03%, while maximizing the λ results in 33.40%; and maximizing the average profit (with the first trading algorithm) results in 53.55%. On the other hand, changing the portfolios before they could reach the long term mean proved to be a bad strategy (Fig. 5).

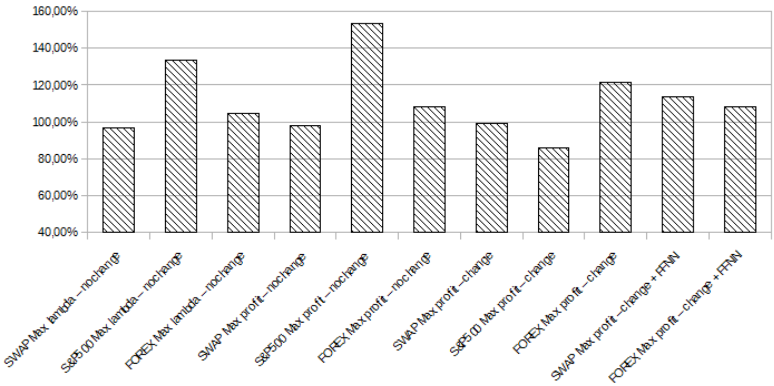

Fig. 5. Trading results with mean reverting portfolios

The FOREX rates in this period showed only a slight increase during the year (1.52%), our methods achieved up to 21.66% profit (Fig. 5).

As one can see, the novel portfolio optimization methods outperform the traditional ones achieved by maximizing the predictability (d'Aspremont[11]) and the trading algorithm shown in (Fogarasi and Levendovszky[14]). Also, deploying an FFNN can further increase the performance.

4.2 New results regarding the application of HMMs for trading

II. group of theses: I have increased the performance of prediction based algorithmic trading by fitting Hidden Markov Models (HMM) to the asset price series.

(Related publications: [4])

HMMs are commonly used in various fields, but in this case they are used to predict future values of financial time series. The computational model of trading by HMM is described by Fig. 6.

If we treat the observations having discrete values over a given alphabet $o_t \in \{k_1, k_2, ..., k_M\}$, then the observation probability matrix, $\mathbf{B}_{NxM}$ describes the probability distribution over the possible values for each state. In the discrete case, HMM is described by $\Theta = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}\}$, where $\boldsymbol{\pi}_N$ denotes the initial distribution vector and $\mathbf{A}$ is the transition probability matrix.

When the observable output (o_t) is continuous, the observation probabilities are described by probability density functions, instead of a probability matrix. For this purpose a multivariate Gaussian mixture model (Dasgupta[10]) was used, in which the density function is composed of a weighted sum (according to the weight matrix ***W***) of *M* independent Gaussian functions: $P\left(o_t \mid q(t) = q_j\right) = \sum_{k=1}^{M} w_{jk} b_{jk}\left(o_t\right)$, where $b_{jk}\left(o_t\right) \sim N\left(\boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}\right)$ and $q(t)$ is the hidden state at *t*. In the continuous case, HMM can then be described as

$$\Theta = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{W}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}. \tag{1.7}$$

The model parameter estimation (learning) is a key aspect when we are using HMMs for prediction during high-frequency trading. During training, the likelihood (or in practice, due to the small order of magnitude of such probabilities, the log-likelihood) of the model is maximized based on the given observations (Rabiner and Juang[34]):

$$\Theta_{opt} = \arg\max_{\Theta} P\left(\mathbf{X} \mid \Theta\right), \tag{1.8}$$

where $\mathbf{X}$ is the training set. The optimization problem described in (1.8) can be solved in multiple ways. However, there is no computationally efficient algorithm known to find the global optimum.

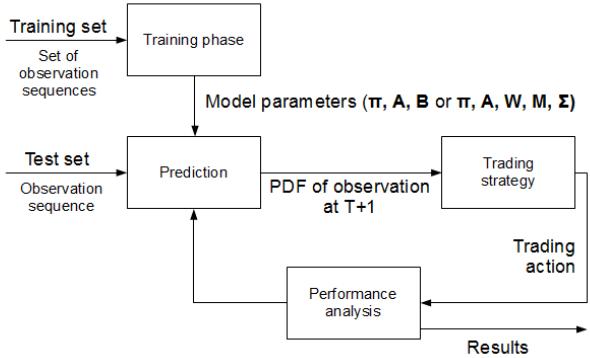

Fig. 6. Computational approach - trading by HMM

The Baum-Welch expectation maximization (BW EM) algorithm (Baum *et al.*[6]) is a mechanism to iteratively update the model (given by (1.7)) starting from an arbitrary initial value and iterating until the likelihood of the model converges to a certain value. Since this is an iterative method, which can use the forward-backward algorithm, implemented in an efficient way by dynamic programming (Rabiner[35]), this proves to be relatively fast. On the other hand, it may get stuck in one of the local maxima making the final result highly dependent on the initialization.

II.1. I have combined the BW algorithm with simulated annealing (SA) to yield a fast and efficient hybrid learning algorithm to find the global optimum of the model parameters faster.

This is motivated by the fact that there are no analytical solution for finding the global optimum of model parameter likelihood, and the BW algorithm tends to get stuck in one of the local maxima.

In the case of applying simulated annealing (Kirkpatrick *et al.*[24]), the energy function is the log-likelihood of the selected model. On the other hand, its convergence is rather slow in large dimensional search spaces. The real-time nature of this algorithm can be ensured by limiting the number of steps of SA making it fit into a predefined time interval.

Our objective is then to couple the BW algorithm with SA. The new algorithm operates as follows: first we start with SA randomly choosing a new accepted state. Then the BW algorithm takes over quickly searching for a local maximum around the state accepted by SA. Once this local maximum is found, then SA generates the next random state subject to a given schedule, from which BW searches for the local maximum again. This cycle is repeated until a given number of steps have been carried out (Fig. 7).

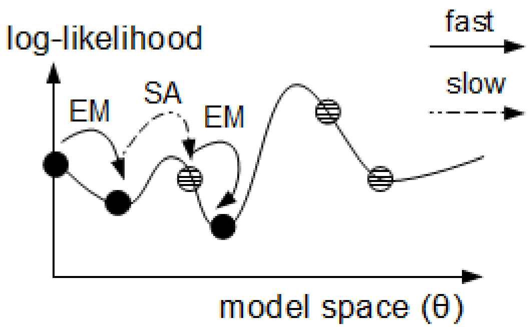

Fig. 7. Optimization with the hybrid algorithm

In this way, good quality solutions can be achieved in relatively short time. This approach can bring more reliable results for trading, which are less dependent on the initialization than the traditional Baum-Welch.

II.2. To further ease the underlying computational complexity, I have employed a new clustering algorithm to pre-process the training set.

Optimizing model likelihood in the full space of Θ including every available observation is computationally exhausting. But by forming clusters out of the observations, and assigning them to the corresponding hidden states, the p.d.f. describing the observation probabilities can be estimated for each cluster, separately. In this way, the complexity can be reduced.

Former attempts to utilize clustering in the HMM training process proved to be beneficial in the field of speech recognition (Rabiner[35]), however, applications for financial time series has yielded much more moderate performance (Idvall and Jonsson[20]).

The main drawback of Rabiner's algorithm is that it performs a separate and linear clustering phase besides a non-linear GMM fitting phase per hidden state. This drawback motivated me to develop the following algorithm (depicted in Fig. 8):

1. Fit a Gaussian mixture distribution using every data point in the training set.

2. Assign each mixture component to a corresponding hidden state, index the price vectors accordingly.

3. Calculate the transition probabilities and the initial distribution vector from relative frequencies (note that the remaining model parameters, namely the p.d.f. of the observations, are already estimated during the first step).

As data points having similar distribution can differ in this aspect, the performance of the algorithm can be further enhanced by also taking into consideration the price vector in the next time instance besides the current one during clustering.

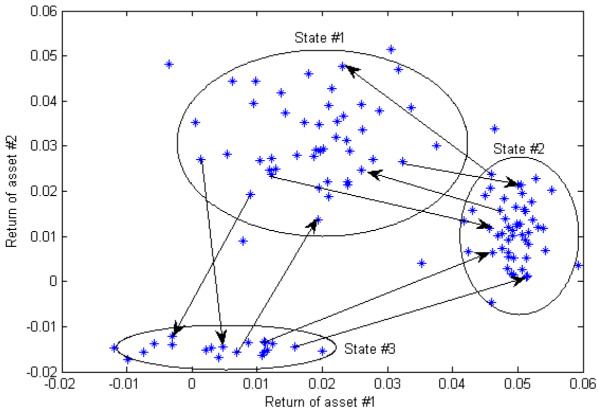

Fig. 8. Clustering by GMM and transitions between the identified hidden states

II.3. I have carried out dimensional reduction by probabilistic PCA (PPCA) which in turn allows one to control the model degree. In this way, one can avoid overfitting and also control the computational time, at the same token.

PPCA is a derivation of principal component analysis (PCA), a well-known technique for dimension reduction (Jolliffe[22]), having a proper density-estimation framework (Tipping and Bishop[36]). PPCA model is described by the following conditional probability distribution: $\mathbf{r} \mid \mathbf{x} \sim N\left(\mathbf{W}\mathbf{x} + \boldsymbol{\mu}, \sigma^2 \mathbf{I}\right)$, where the marginal distribution for the latent variables is Gaussian by convention with $\mathbf{x} \sim N(0, \mathbf{I})$ causing the marginal distribution for the observed data to be $\mathbf{r} \sim N\left(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}\right)$.

Due to lack of a closed-form solution, an iterative expectation-maximization algorithm was used to obtain the maximum-likelihood estimator of the model parameters (Tipping and Bishop[36]).

Transforming the data from a higher dimensional space into a space of fewer dimensions as a pre-processing step for the HMM yields the following advantages:

- optimization in a smaller dimensional space requires much less computational time, which enables one to do higher frequency trading, or to involve a larger number of assets, which was not feasible formerly;
- one can control the model degree which is vital to avoid overfitting.

Performance analysis

The performance was tested by the methods described above. The bar chart (Fig. 9) show that in the case of the U.S. SWAP rates all of the introduced methods beat the market tendency (-39.27%), and in the scenario when SA was used to train the continuous mode HMM the trading was profitable with a 108.52% yearly profit. Regarding the FOREX rates, our methods achieved up to 26.62% profit.

As one can see, the novel HMM optimization methods outperform the traditional EM algorithm in most scenarios, while the hybrid approach speeds up the convergence with one order of magnitude.

As it is shown in the figure, clustering and dimension reduction methods were also proven to be profitable on the tested time series, outperforming the uniform portfolio. The numerical results show that applying PPCA is beneficial with normal clustering (`CLUST`), and less profitable with the enhanced clustering version (`CLUST (2)`).

In comparison with the results achieved without pre-processing, the range of the realized profits are comparable, reaching up to total 138.28% yearly profit on SWAP. Nevertheless, the applied pre-processing steps resulted in a considerable speed-up (another order of magnitude) compared to the previous methods.

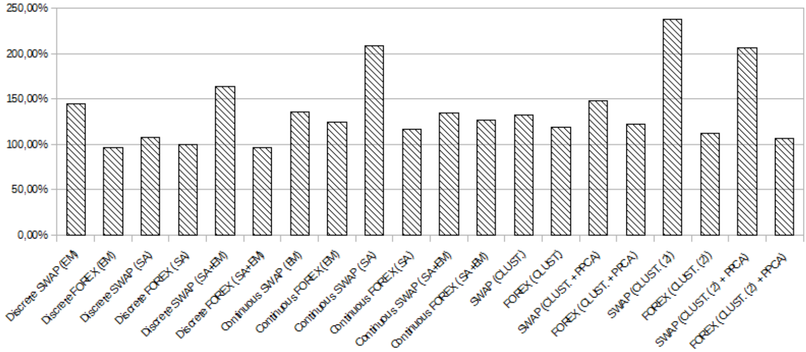

Fig. 9. Trading results with HMMs

4.3 New results for the application of AR-HMMs and GPGPU for trading

III. group of theses: I generalized the mean reversion based trading from the OU SDE to a more flexible modeling with autoregressive HMMs (AR-HMM) which includes the OU SDE. In this way, AR-HMM can better capture the stochastic nature of p(t). I have introduced novel parameter estimation methods and a corresponding objective function.

(Related publications: [2, 3])

As a type of mixture models, having enough degree of freedom (number of hidden states) HMMs are capable of modeling a large class of distributions. Moreover, by AR-HMM we can capture both the long and short range dependencies, as it combines a Markov chain on the hidden variables, and statistical dependencies on the observed variables (Berchtold[7]). Unlike the standard HMM assumption, in the case of AR-HMMs the emissions are conditionally not independent given the hidden state (Murphy[30]). The observable output was treated as a continuous value, described by a univariate Gaussian probability density function. Then the probability of emitting a specific output is determined by the conditional probability

$$P\left(p(t)\middle|p(t-1),q_t=j,\Theta\right)=N\left(p(t)\middle|\varphi_j p(t-1)+\mu_j,\sigma_j\right). \quad (3.1)$$

In other words, the observation depends on the hidden state, and on the previous observation through an additive autoregressive component.

The novel approach introduced manages to overcome the problems arising from the fact that the traditional OU approach is unable to handle processes changing their characteristic over time. Besides providing a more general model, it gives more accurate and stable predictions for the future portfolio prices.

Real-time implementation on GPGPUs

Besides the profit achieved by the introduced methods, the computational time is also an important aspect in real environment. Along with the algorithmic approaches developed for fast trading, the software implementation in massively parallel architectures can also be used to decrease the running time and ensure a real-time algorithmic trading. This paves the way towards high-frequency trading, as well as trading on a larger number of assets. Furthermore, it helps perform a deeper analysis providing better quality results.

The nature of the used methods and algorithms, including linear algebra operations, simulated annealing and the forward-backward algorithm for HMMs allows parallel implementation and requires lesser data transfer for the input and output variables. These algorithmic properties make a GPGPU based implementation feasible and beneficial.

General-Purpose computing on Graphics Processing Units (GPGPU) is a computing concept which utilizes the massively parallel architecture of a Graphics Processing Unit (GPU) to solve arbitrary numerical problems (Nvidia[8]). In terms of floating-point operations per second, which is the primary measure of hardware performance for scientific or engineering computations, GPUs are superior compared to traditional CPUs.

GPUs are also outperforming the CPUs in power consumption per floating-point operation, hence if energy efficiency is a concern then this approach proves to be favorable.

As the second part of my group of theses, I intend to provide GPGPU based parallel implementations for the described methods to further enhance the practical usability of results.

III.1. I have shown that we can consider the AR-HMM as a time dependent generalization of the OU and Brownian motion (BM) SDEs.

From the OU process, defined in (1.2), one can obtain the following probabilistic model (Gillespie[40]):

$$P\left(p(t)\middle|p(t-1),\vartheta,\mu,\sigma\right)=$$
$$N\left(p(t)\middle|e^{-\vartheta\Delta t}p(t-1)+\left(1-e^{-\vartheta\Delta t}\right)\mu,\sigma\sqrt{\frac{1-e^{-2\vartheta\Delta t}}{2\vartheta}}\right). \quad (4.1)$$

This equation describes the OU process in the same form as (3.1), where the conditional observation probability of an AR-HMM is defined. By carrying out the following mappings

$$\varphi_1=e^{-\vartheta\Delta t},\ \mu_1=\left(1-e^{-\vartheta\Delta t}\right)\mu \text{ and } \sigma_1=\sigma\sqrt{\frac{1-e^{-2\vartheta\Delta t}}{2\vartheta}}, \quad (4.2)$$

we can consider a single state AR-HMM being equivalent as an OU process. However, this can be either a mean reverting or a mean averting process. Setting $\vartheta=0$ in (1.2) yields a BM, more generally, an AR-HMM state can represent BM with a possible linear drift as well.

Based on this approach, a multi state AR-HMM can be considered as a generalization of the OU and BM SDEs. Furthermore, having multiple states brings even more flexibility into modeling the price distributions, and also captures if the process is driven by different models over different intervals in time (often called as regimes (Hamilton and Susmel[17])).

III.2. I adapted the objective function introduced in I.2. to maximize the expected return for the case of modeling by AR-HMMs. In addition, I extended it to cope with the secondary effects (bid-ask spread).

Similarly to (1.3), the objective function is expressed as follows:

$$\Psi(\mathbf{x})=\max_{0\le t}E\left(p(t)\right)-p(0), \quad (4.3)$$

but in this case $E\left(p(t)\right)=\left(E\left(p(t-1)\right)\boldsymbol{\varphi}+\boldsymbol{\mu}\right)^T\boldsymbol{\gamma}_t$ and $\boldsymbol{\gamma}_t=\mathbf{A}^T\boldsymbol{\gamma}_{t-1}$ is described recursively, where $E\left(p(0)\right)=p(0)=\mathbf{x}^T\mathbf{s}_0$ and $\boldsymbol{\gamma}_t^{(j)}=P\left(q_t=j\middle|\mathbf{X}\right)$.

First, we need to identify the AR-HMM model parameters for an observed process, and then, as a novel approach, optimize the prediction based average return.

Including the bid-ask spread into this model can be done in a straightforward manner, the current and the future prices, for fitting the AR-HMM, of the portfolio should be taken into consideration as the corresponding bid and ask prices.

III.3. I generalized the objective function introduced in III.2. Instead of the expected return, the complete probability density function of the future portfolio evaluations is modelled by the AR-HMM fit to the time series.

This goal function maximizes the lower bound on return achieved with a pre-defined probability η, i.e.:

$$\omega(t): P\left(p(t) \geq \omega(t)\right) = \eta \,. \tag{4.4}$$

Then the objective function is expressed as follows:

$$\Psi(\mathbf{x}) = \max_{0<t} \omega(t) - p(0), \quad E\left(p(t)\right) - p(0) > 0 \,. \tag{4.5}$$

Taking into account the cardinality constraint as well, portfolio optimization can be reduced to a constrained optimization problem:

$$\mathbf{x}_{opt} = \arg\max_{\mathbf{x}} \Psi(\mathbf{x}), card(\mathbf{x}) \leq l \,. \tag{4.6}$$

This objective function may prove to be much more beneficial for the portfolio holder than just optimizing the predictability or maximizing the average return (introduced in I.2.), however, its solution does not lend itself to analytical tractability (as opposed to the maximal predictability which leads to a generalized eigenvalue problem). Thus, to come to grip with maximizing $\omega(t)$ for a given portfolio, we identify AR-HMM model parameters for an observed process, and then, as a novel approach, we calculate the prediction based CCDF by Monte Carlo methods enabled by the higher computational power of the GPGPU.

Parameter η allows the investors to adjust their willingness to take risk. One can see that $\eta < 0.5$ results in a braver strategy, while $\eta > 0.5$ yields a rather cautious one. The bid-ask spread can be included to the model alike as in III.2.

III.4. To utilize the GPGPU environment, I have ported the methods introduced in III.1. and III.2. to this architecture. Motivated by the high number of possible parallel threads the SA was extended to cope with multiple portfolios simultaneously. As a result, in the same time span a

larger population of portfolio candidates can be evaluated based on a more precise objective function compared to the traditional implementation. This improvement leads to higher profits and also casts trading feasible on higher frequency data.

The parallelism of the proposed method is twofold with respect to granularity:

- Coarse-grained parallelism: as opposed to the sequential SA in which in each run we start from a given initial condition, in the case of our parallel implemented SA several algorithms run from several initial points at the same time, i.e. AR-HMM parameter fittings in each subspace are done simultaneously.
- Fine-grained parallelism: in each step of SA, (i) the algorithm for AR-HMM parameter estimation is parallel along the hidden states, and (ii) the evaluation of the objective function through the Monte Carlo simulations for estimating the complementary cumulative distribution function is also carried out in a parallel manner.

Performance analysis

Mean reversion parameter estimation by AR-HMM

As it was detailed in III.1., the single state AR-HMM can be treated as a generalization of the OU process, hence it is suitable as a parameter estimation method. Estimating the long term mean (μ) of the process of portfolio valuations is instrumental for mean reverting trading. For the sake of comparison with other estimation procedures (Sipos and Levendovszky[1], Fogarasi and Levendovszky[14]), different methods were tested on artificially generated data.

As one can see (Fig. 10), the newly proposed way of OU parameter identification gives the most precise estimations, outperforming the traditional methods by an order of one magnitude.

Trading results

For back-testing on FOREX time series, we used the *MetaTrader® 5* platform providing a real environment with secondary effects including, like the bid-ask spread. During the simulations, the leverage was set to 10.

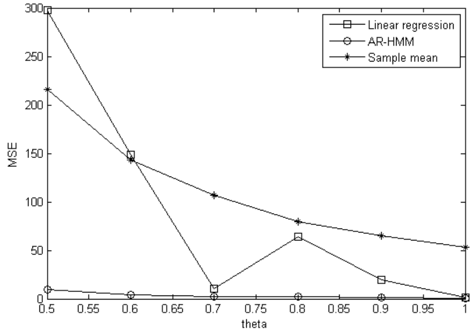

Fig. 10. Comparison of different mean estimation techniques

In this period, the FOREX rates had a slightly decreasing tendency, an equally weighted portfolio would result in -2.24%. In the scenario when a longer time window was used, the trading was even more profitable with a 144.10% yearly profit including every secondary effects. However, considering longer sets of data, the training of the AR-HMM requires substantially longer computational time.

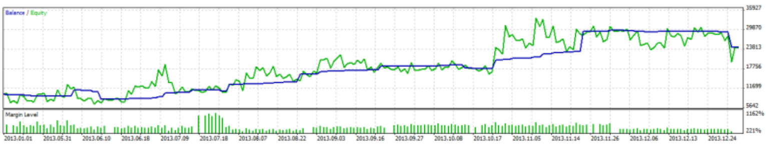

Fig. 11. Detailed AR-HMM trading results on FOREX

As it is shown in figure 11, using this strategy not only results in a favorable profit, but without major drawdowns, the balance is almost monotonously increasing.

GPGPU implementation

In this section an extensive speed profiling is given, demonstrating that the GPGPU based implementation outperforms the traditional one in terms of the required computational time. Hence, this allows one to involve more assets into the analysis or perform an even higher frequency trading. The numerical experiments lasted the same time interval on both architectures. Moreover, through the decreased power consumption, it reduces operating costs and makes the algorithmic trading more eco-friendly.

The next figures indicate real tests running on FOREX data obtained at 15 minute and daily tick time (Fig. 12).

One can see, in each case we managed to secure positive profit. On the 15 minute tick time the GPGPU has achieved 2.7% profit, while 0.1% loss achieved by CPU in a 3 days interval (annualization is often misleading on high-frequency data). On the daily averages the system performed more modestly, but even in this case the GPGPU secured a 22.3% yearly profit as opposed to the 3.1% loss achieved by the CPU.

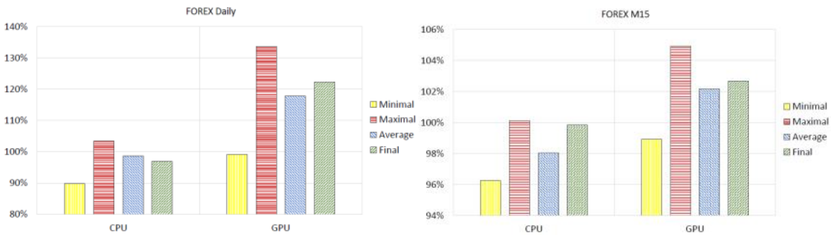

Fig. 12. Trading performance on FOREX data

The experiments were conducted on a PC with a second generation Intel i7 CPU and on one NVIDIA GeForce GTX 570 GPU. Fig. 13 shows the speedup over serial implementation, that is the ratio between the computation time on CPU and the computation time on GPGPU.

To demonstrate the difference between CPU and GPGPU, in the M15 simulations the portfolio optimization took 478 ms on average, which means about 10 minutes on CPU casting it unfeasible to use in this time scale.

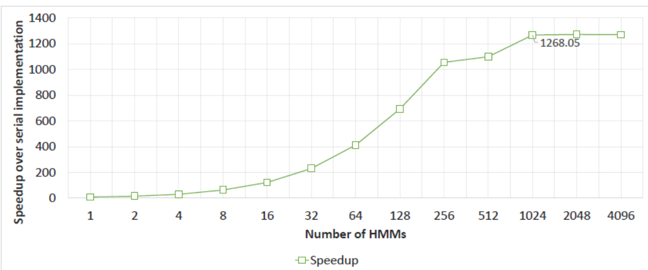

Fig. 13. Speedup over serial implementation

5. Conclusions and applications

In the theses, I have proposed novel algorithms for model parameter estimation and predicting future values of financial time series for portfolio optimization with cardinality constraints. The optimal portfolio selection has been carried out by maximizing the mean return as an objective function.

Table 2 summarizes the profit achieved by the novel methods trading on different assets compared to the traditional approaches. The proposed trading algorithms has proven to be profitable on real financial time series taking into account the bid ask spread as well. The performance analysis demonstrated that the trading based on the novel parameter identification algorithms and objective functions could increase the trading efficiency and the profit compared to the traditional methods. It is also shown that the efforts from both algorithmic (e.g. clustering and PPCA, hybrid training algorithm for HMMs) and implementation (using GPGPUs) side can further speed up the algorithms giving rise to high frequency trading applications. Consequently, I have managed to improve the performance of algorithmic trading.

		FOREX	**SWAP**	**S&P500**
I. thesis group (MR)	*Traditional*	104,86%	96,57%	133,40%
	Novel	121,66%	113,49%	153,55%
II. thesis group (HMM)	*Traditional*	124,47%	135,65%	-
	II.1.	126,62%	208,52%	-
	II.2. II.3.	122,49%	238,28%	-
III. thesis group (AR-HMM)	*Traditional*	193,02%	-	85,73%
	Novel	244,10%	-	110,15%

Table 2. Comparison of profitability of novel methods compared to the traditional ones

While my contributions mainly focus on the purposes of algorithmic trading, the ones concerning model parameter identification can be applied to model and predict any time series. For instance, HMMs are widely used in speech recognition and synthesis and bioinformatics (e.g. DNA motif discovery, protein folding).

Since algorithmic trading tends to be the backbone of modern financial industry, the novel methods can contribute to reliability of financial processes, as a result they can also be beneficial for the economy and the society, on the whole.

Acknowledgements

I would like to express my gratitude to my supervisor for his continuous support and encouragement. Special thanks also to dr. Norbert Fogarasi, Attila Ceffer, Farhad Kia and dr. Gábor Jeney for their helpful suggestions related to this work. Finally, I would like to acknowledge the support of my family and friends keeping me motivated along these years.

Publications of the author

Journal articles

1. SIPOS, I. Róbert; LEVENDOVSZKY, János. Optimizing sparse mean reverting portfolios. *Algorithmic Finance*, 2013, 2.2: 127-139. DOI: 10.3233/AF-13021

2. SIPOS, I. Róbert; LEVENDOVSZKY, János. Optimizing Sparse Mean Reverting Portfolios with AR-HMMs in the Presence of Secondary Effects. *Periodica Polytechnica Electrical Engineering and Computer Science*, 2015, 59(1), 1-8. DOI: 10.3311/PPee.7352

3. SIPOS, I. Róbert; CEFFER, Attila; LEVENDOVSZKY, János. Parallel optimization of sparse portfolios with AR-HMMs. *Submitted to: Computational Economics.*

Conferences

4. SIPOS, I. Róbert; LEVENDOVSZKY, János. High frequency trading with Hidden Markov Models by using clustering and PPCA algorithms. *In: Proceedings, 15th Applied Stochastic Models and Data Analysis (ASMDA2013)*, pp. 67-80. Barcelona, 2013.

References

5. BAUM, L. E. and PETRIE, T. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. The Annals of Mathematical Statistics 37 (6): 1554–1563, 1966.

6. BAUM, L. E., PETRIE, T., SOULES, G. and WEISS, N. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann. Math. Statist., vol. 41, no. 1, pp. 164–171, 1970.
7. BERCHTOLD, Andre. The double chain Markov model. Communications in Statistics-Theory and Methods, 1999, 28.11: 2569-2589.
8. CUDA C Programming Guide, Nvidia.com, http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html
9. CYBENKO, G. (1989) Approximations by superpositions of sigmoidal functions. Mathematics of Control, Signals, and Systems, 2 (4), pp. 303-314.
10. DASGUPTA, S. Learning Mixtures of Gaussians. Proceedings of Symposium on Foundations of Computer Science (FOCS), 1999.
11. D'ASPREMONT, A. (2011) Identifying small mean-reverting portfolios. Quantitative Finance, 11:3, pp. 351-364.
12. DURBIN, R., EDDY, S. R., KROGH, A. and MITCHISON, G. Biological sequence analysis, Cambridge University Press, 1998.
13. FAMA, E. and FRENCH, K. (1988) Permanent and Temporary Components of Stock Prices. The Journal of Political Economy, 96(2), pp. 246-273.
14. FOGARASI, Norbert; LEVENDOVSZKY, János. Improved parameter estimation and simple trading algorithm for sparse, mean reverting portfolios. In: Annales Univ. Sci. Budapest., Sect. Comp. 2012. p. 121-144.
15. FOGARASI, Norbert; LEVENDOVSZKY, János. A simplified approach to parameter estimation and selection of sparse, mean reverting portfolios. Periodica Polytechnica Electrical Engineering and Computer Science, 2013, 56.1: 21-28.
16. HAGAN, M.T. and MENHAJ, M.B. (1994) Training feed-forward networks with the Marquardt algorithm. IEEE Transactions on Neural Networks, Vol. 5, No. 6, pp. 989–993.
17. HAMILTON, James D.; SUSMEL, Raul. Autoregressive conditional heteroskedasticity and changes in regime. Journal of Econometrics, 1994, 64.1: 307-333.
18. HASSAN, R. and NATH, B. Stock Market Forecasting Using Hidden Markov Model: A New Approach. Proceedings of the 2005 5th International Conference on Intelligent Systems Design and Applications.
19. HASSAN, R., NATH, B. and KIRLEY, M. A fusion model of HMM, ANN and GA for stock market forecasting. Expert Systems with Applications 33 (2007) 171–180.
20. IDVALL, P. and JONSSON, C. Algorithmic Trading - Hidden Markov Models on Foreign Exchange Data. Master’s Thesis, Department of Mathematics, Linköpings Universitet, 2008.
21. ITO, K. (1944) Stochastic Integral. Proc. Imperial Acad. Tokyo, 20, pp. 519-524.
22. JOLLIFFE, I. T. Principal Component Analysis. Springer-Verlag, New York, 1986.
23. JURAFSKY, D. and MARTIN, J. H. Speech and Language Processing: An Introduction to Natural Language Processing. Computational Linguistics, and Speech Recognition, 2nd Edition, Pearson Prentice Hall, London, 2006.
24. KIRKPATRICK, S., GELATT, C.D. and VECCHI, M.P. (1983) Optimization by Simulated Annealing. Science, 220, pp. 671-680.
25. MAMON, R. S. and ELLIOTT, R. J. Hidden Markov Models in Finance, Springer Science Business Media, 2007.
26. MANZAN, S. (2007) Nonlinear Mean Reversion in Stock Prices. Quantitative and Qualitative Analysis in Social Sciences, 1(3), pp. 1-20.
27. MARKOWITZ, H. (1952) Portfolio Selection. The Journal of Finance, Vol. 7 (1), pp 77–91.

28. METATRADER (2014) FOREX time series [WWW]. Available from: http://www.metaquotes.net/en/metatrader5 [Accessed 02/2014].
29. MORGAN STANLEY MARKETONE (2010) SWAP series [DATABASE].
30. MURPHY, Kevin P. Machine learning: a probabilistic perspective. MIT Press, 2012.
31. NATARJAN, B.K. (1995) Sparse approximate solutions to linear systems. SIAM J. Comput., 24(2), pp. 227-234.
32. ORNSTEIN, L.S. and UHLENBECK, G.E. (1930) On the Theory of the Brownian Motion. Physical Review, 36(5), p. 823.
33. POTERBA, J.M. and SUMMERS, L.H. (1988) Mean reversion in stock prices: Evidence and implications. Journal of Financial Economics, 22(1), pp. 27-59.
34. RABINER, L. R. and JUANG, B. H. An introduction to hidden Markov models. IEEE ASSP Magazine 3:4-16, 1986.
35. RABINER, L. R.. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, 77 (2), p. 257–286, February 1989.
36. TIPPING, M. and BISHOP, C. Probabilistic principal component analysis. Journal of the Royal Statistical Society: Series B, 61, part 3, pp. 611-622, 1999.
37. VAN GELDER, P. (2009) Statistical modeling of financial markets. In: TU Delft, UFS Workshop, Bloemfontein, March 4-5 2009.
38. YAHOO (2011) S&P 500 asset series [WWW]. Available from: http://finance.yahoo.com [Accessed 07/2011].
39. DOOB, J. L. (1953). *Stochastic processes* (Vol. 101). Wiley: New York.
40. GILLESPIE, D. T. (1996). Exact numerical simulation of the Ornstein-Uhlenbeck process and its integral. Physical review E, 54(2), 2084.
41. FOGARASI, Norbert; LEVENDOVSZKY, János. Sparse, mean reverting portfolio selection using simulated annealing. Algorithmic Finance, 2013, 2.3: 197-211.

Appendix

Throughout the performance analysis parts of the thesis groups, the shown numerical results obtained on the following data sets: (i) Daily closing prices of 500 stocks from the S&P 500 (between July 2010 and July 2011) (Yahoo[38]) – Thesis group I. and III.; (ii) U.S. SWAP rates in daily resolution (Morgan Stanley[29]) – Thesis group I.: from the year of 1998, Thesis group II.: between August 2008 and August 2010; (iii) FOREX rates in daily resolution (EUR/USD, GBP/USD, AUD/USD, NZD/USD, USD/CHF, USD/CAD) (Metatrader[28]) – Thesis group I.: mid-prices between October 2005 and September 2006, Thesis group II.: mid-prices between December 2009 and 2011, Thesis group III.: bid and ask prices from the year of 2013.

For a detailed comparative analysis the following performance measures were calculated for each simulation: (i) minimal value $G_{\min} = \frac{1}{c_0} \min_{0 \le t \le T} c_t$; (ii) final value $G_{final} = \frac{c_T}{c_0}$; (iii) maximal value $G_{\max} = \frac{1}{c_0} \max_{0 \le t \le T} c_t$; (iv) average value $G_{avg} = \frac{1}{c_0} \frac{1}{T} \sum_{t=1}^{T} c_t$, where c_t denotes the sum of owned cash and the market value of the owned portfolio at time instance *t*, while c_0 denotes the initial cash (in each case the agent started with \$10,000). Regarding the sparsity constraint, 3 assets were selected in each transaction.