Institut für Numerische Simulation
Rheinische Friedrich-Wilhelms-Universität Bonn

Wegelerstraße 6 ● 53115 Bonn ● Germany
phone +49 228 73-3427 ● fax +49 228 73-7527
www.ins.uni-bonn.de

Jochen Garcke, Thomas Gerstner and Michael Griebel

Intraday Foreign Exchange Rate Forecasting using Sparse Grids

INS Preprint No. 1006

November 2010

Intraday Foreign Exchange Rate Forecasting using Sparse Grids

Jochen Garcke[1], Thomas Gerstner[2] and Michael Griebel[2]

[1]Matheon and
Institut für Mathematik
Technische Universität Berlin
Straße des 17. Juni 136
D–10623 Berlin
garcke@math.tu-berlin.de

[2]Institut für Numerische Simulation
Universität Bonn
Wegelerstr. 6
D–53115 Bonn
{gerstner;griebel}@ins.uni-bonn.de

Abstract

We present a machine learning approach using the sparse grid combination technique for the forecasting of intraday foreign exchange rates. The aim is to learn the impact of trading rules used by technical analysts just from the empirical behaviour of the market. To this end, the problem of analyzing a time series of transaction tick data is transformed by delay embedding into a D-dimensional regression problem using derived measurements from several different exchange rates. Then, a grid-based approach is used to discretize the resulting high-dimensional feature space. To cope with the curse of dimensionality we employ sparse grids in the form of the combination technique. Here, the problem is discretized and solved for a collection of conventional grids. The sparse grid solution is then obtained by linear combination of the solutions on these grids. We give the results of this approach to FX forecasting using real historical exchange data of the Euro, the US dollar, the Japanese Yen, the Swiss Franc and the British Pound from 2001 to 2005.

1 Introduction

After the breakdown of the Bretton Woods system of fixed exchange rates in 1973, the forecasting of exchange rates became more and more important. Nowadays the amounts traded in the foreign exchange market are over three trillion US dollars every day. With the emergence of the Euro in 1999 as a second world currency which rivals the US dollar [27], the forecasting of FX rates got more necessary but also more complicated. Besides incorporating basic economic and financial news, reports by market analysts, and opinions expressed in financial journals, many investors and traders employ in their decision process (besides their guts feelings) technical tools to analyze the transaction data. These data consist of huge amounts of quoted exchange rates, where each new transaction generates a so-called *tick*, often many within a second.

Several academic studies have evaluated the profitability of trading strategies based on daily or weekly data. However, such investigations of trading in the foreign exchange market have not been consistent with the practice of technical analysis [22, 25]. Technical traders transact at a high frequency and aim to finish the trading day with a net open position of zero. In surveys of participants in the foreign exchange market, 90% of respondents use technical analysis in intraday trading [23, 30], whereas 25% to 30% of traders base most of their trades on technical signals [4]. Evidence was presented in [26] that so-called support and resistance levels, i.e. points at which an exchange rate trend is likely to be suspended or reversed, indeed help to predict intraday trend interruptions. On the other hand, the authors of [5] examined filter rules supplied by technical analysts and did not find evidence for profit. Nevertheless, the existence of profit-making rules might be explained from a statistical perspective by the more complex, nonlinear dynamics of foreign exchange rates as observed in [16]. In [6] two computational learning strategies, reinforcement learning and genetic programming, were compared to two simpler methods, a Markov decision problem and a simple heuristic. These methods were able to generate profits in intraday trading when transaction costs were zero, although none produced significant profits for realistic values. In [25], with the use of a genetic program and an optimized linear forecasting model with realistic transaction costs, no evidence of excess returns was found but some remarkable stable patterns in the data were nevertheless discovered. In [35] multiple foreign exchange rates were used simultaneously in connection with neural networks. There, better performance was observed using multiple exchange rates than in a separate analysis of each single exchange rate.

In this paper we tackle the problem of forecasting intraday exchange rates by transforming it into a machine learning regression problem [9, 11, 13]. The idea behind this approach is that the market will behave similarly in similar situations due to the use of technical analysis by market participants. The machine learning algorithm attempts to learn the impact of the trading rules just from the empirical behaviour of the market. To this end, the time series of transaction tick data is cast into a number of data points in a D-dimensional feature space together with a label. The label represents the difference between the exchange rates of the current time and a fixed time step into the future. Therefore one obtains a regression problem. Here, the D features are derived from a delay embedding of the data [24, 29]. For example, approximations of first or second derivatives at each time step of the exchange rate under consideration can be used. Furthermore, we investigate especially the additional use of tick data from further exchange rates to improve the quality of the prediction of one exchange rate.

Delay embedding is a powerful tool to analyze dynamical systems. Taken's theorem [29] gives the conditions under which a chaotic dynamical system can be reconstructed from a sequence of observations. In essence, it states that if the state space of the dynamical system is a k-dimensional manifold, then it can be embedded in $(2k+1)$-dimensional Euclidean space using the $2k+1$ delay values $f(t), f(t-\tau), f(t-2\tau), \ldots, f(t-2k\tau)$. Here, heuristic computational methods, such as the Grassberger-Procaccia algorithm [14] can be used to estimate the embedding dimension k.

In this work we apply our recent approach for data mining problems [10, 11]. It is based on the regularization network formulation [13] and uses a grid, independent from the data positions, with associated local ansatz functions to discretize the feature space. This is similar to the numerical treatment of partial differential equations with finite elements. To avoid the curse of dimensionality, at least to some extent, a so-called sparse grid [3, 34] is used in the form of the combination technique [15]. The approach is based on a hierarchical subspace splitting and a sparse tensor product decomposition of the underlying function space. To this end, the regularized regression problem is discretized and solved on a certain sequence of conventional grids. The sparse grid solution is then obtained by a linear combination of the solutions from the different grids. It turns out that this method scales only linearly with the number of data points to be treated [11]. Thus, this approach is well suited for machine learning applications where the dimension D of the feature space is moderately high, but the amount of data is very large, which is the case in FX forecasting. This is in contrast to support vector machines and related kernel based techniques whose cost scale quadratically or even cubically with the number of data points (but allow to deal with very high-dimensional feature spaces). We show in this article that sparse grid regression can indeed be a useful tool for intraday foreign exchange rate forcasting using real transaction tick data and we present realizable and practical trading strategies. Thereby, we achieve prediction accuracies of almost 60%, profits of up to 25% of the maximum attainable profit and we measure average revenues per transaction larger than typical transaction costs.

The remainder of this article is organized as follows: In Section 2 we describe how we transform the problem of forecasting a foreign exchange rate into a data mining problem via delay embedding and discuss the resulting regularized regression problem. Section 3 gives the basics of the sparse grid combination technique, our employed numerical algorithm. Then, in Section 4, we give the results for our new approach using real historical exchange data of the Euro, the US dollar, the Japanese Yen, the Swiss Franc and the British Pound from the years 2001 to 2005 and compare its performance to a conventional strategy [2]. Here, we also comment on tradeable strategies and transaction costs. Finally, Section 5 gives some concluding remarks.

2 Exchange rate forecasting as a data mining problem

We show how the historical intraday exchange rate data are given and discuss how we can convert the FX forecast problem into a regularized least squares regression problem in a high-dimensional space by delay embedding.

2.1 Input data

Foreign exchange rate tick data consist of the bid and ask quotes of market participants recorded by electronic transaction systems. Note that the tick data are not the real transaction prices for this market, but only the quotes at which market participants want to trade. This results in a small uncertainty in the data. Nevertheless, exchange rate data is presented in this form by most financial news agencies like e.g. Reuters or Bloomberg. Such historical data are collected and sold by these agencies and other data vendors. For example, the database of Olsen Data has recorded in the year 2002 more than 5 million ticks for the EUR/USD exchange rate – the most heavily traded currency pair – which gives about 20,000 ticks per business day. The bid and ask prices are typically converted to midpoint prices: $\frac{1}{2}$(bid price + ask price). Note that the spread, that is the difference between bid and ask prices, has to be included in the analysis of the performance at some point in order to assess the expected trading efficiency of a forecasting tool.

In the following we assume that for each tick we have the date, time and one exchange rate value, the midpoint. The raw data of each considered currency pair therefore looks like

```
09.06.2002 09:18:54 0.95595
09.06.2002 09:18:55 0.95615
09.06.2002 09:18:58 0.95585
09.06.2002 09:18:59 0.95605
09.06.2002 09:19:11 0.95689.
```

Now the raw tick data are interpolated to equidistant points in time with a fixed distance of τ. Data from the future cannot be used, therefore the value at the latest raw tick is employed as the exchange rate at these points, i.e. piecewise constant upwind interpolation is applied. If the latest raw tick is more then τ in the past, which means it is the one used for the interpolated tick at the position before, the exchange rate is set to *nodata*. Furthermore, for all currency pairs the same positions in time are taken. Some data providers already offer such mapped data in addition to the raw data. This way, given J data points from R currency pairs, the input data for exchange rate forecasting has the form

$$\{t_j, f_r(t_j)\} \text{ for } j = 1, \ldots, J \text{ and } r = 1, \ldots, R.$$

Here, t_j denotes the j-th point in time, f_r denotes the exchange rate of the r-th currency pair and $t_{j+1} = t_j + \tau$. Note here that, for reasons of simplicity, we furthermore assume that the nominal differences between the interest rates for the currencies are constant and therefore do not need to be taken into account.

2.2 Delay embedding into a feature space

Based on these interpolated historical input data consisting of $J \cdot R$ data points we now want to predict the value or trend of the exchange rate of the first currency pair f_1. Given a point in time t_j we want to forecast the trend for f_1 at some time $t_j + \hat{k}\tau$ in the future. To this end, we convert the given series of transaction information up to time t_j into data in a D-dimensional feature space, also called attribute space, which is supposed to describe the market situation at time t_j. The D-dimensional vector in feature space is put together by delay embedding the given tick data (see, for example, [7, 19, 21]). For each exchange rate f_r we consider a fixed number K of delayed values

$$f_r(t_j), f_r(t_j - \tau), f_r(t_j - 2\tau), \ldots, f_r(t_j - (K-1)\tau),$$

where K defines our time horizon $[t_j - (K-1)\tau, t_j]$ backward in time. The resulting $R \cdot K$ delayed values could be directly used to give the D-dimensional feature space with $f_1(t)$ being the first coordinate, $f_1(t-\tau)$ the second, and so on up to $f_R(t - (K-1)\tau)$ being the $(R \cdot K)$-th coordinate.

Note that this is not the only way of delay embedding the data for time t_j. Instead of directly employing the exchange rates, (discrete) first derivatives $f'_{r,k}(t_j) := (f_r(t_j) - f_r(t_j - k\tau))/k\tau$ with $k = 1, \ldots, K-1$ can be used in our backward time horizon yielding $K-1$ coordinates for each exchange rate and $R(K-1)$ coordinates in total. Normalized first derivatives $\tilde{f}'_{r,k}(t_j) := \frac{f_r(t_j) - f_r(t_j - k\tau)}{k\tau f_r(t_j - k\tau)}$ can be considered as well, this takes the assumption into account that trading strategies look for relative changes in the market and not absolute ones. Alternatively, a combination of exchange rates, first derivatives, higher order derivatives or statistically derived values like variances or frequencies can be employed as attributes. Note that the actual use of a given feature at all time positions of our backward time horizon of size K, e.g. all K values of the exchange rates or all $K-1$ values of the first derivative, is usually not necessary. A suitable selection from the possible time positions of a given attribute in the time horizon $[t_j - (K-1)\tau, t_j]$, or even only one, can be enough in many situations.

In any case, the number of features obtained by the delay embedding can easily grow large. Therefore, the number K of delay values, that is the size of our backward time horizon, and the total number of derived attributes D have to be chosen properly from the large number of possible embedding strategies. A good choice of such derived attributes and their parameters is non-trivial and has to be determined by careful experiments and suitable assumptions on the behaviour of the market.

In general, the transformation into feature space, i.e. the space of the embedding, for a given point in time t_j is an operator $T : \mathbb{R}^{R\cdot K} \to \mathbb{R}^D$

$$\underline{x}(t_j) = T\left(f_1(t_j), \ldots, f_1(t_j - (K-1)\tau), f_2(t_j), \ldots, f_2(t_j - (K-1)\tau), \ldots, f_R(t_j), \ldots, f_R(t_j - (K-1)\tau)\right)$$

with the feature vector $\underline{x}(t_j) = (x_1, \ldots, x_D) \in \mathbb{R}^D$, where the single features $x_d, d = 1, \ldots, D$, are any of the derived values mentioned.

As the response variable in the machine learning process we employ the normalized difference between the exchange rate f_1 at the current time t_j and at some time $t_j + \hat{k}\tau$ in the future, i.e.

$$y(t_j) = \frac{f_1(t_j + \hat{k}\tau) - f_1(t_j)}{f_1(t_j)}.$$

This will give a regression problem later on. If one is only interested in the trend, the sign of $y(t_j)$ can be used as the response variable which will result in a classification problem.

This transformation of the transaction data into a D-dimensional feature vector can be applied at $J - (K-1) - \hat{k}$ different time points t_j over the whole data series, since at the beginning and end of the given time series data one has to allow for the time frame of the delay embedding and prediction, respectively. Altogether, the application of such an embedding transformation and the evaluation of the associated forecast values over the whole time series results in a data set of the form

$$S = \{(\underline{x}_m, y_m) \in \mathbb{R}^D \times \mathbb{R}\}_{m=1}^{J-(K-1)-\hat{k}}, \tag{1}$$

with $\underline{x}_m = \underline{x}(t_{m+K-1})$ and $y_m = y(t_{m+K-1})$.

This dataset can now be used by any machine learning algorithm, such as neural networks, multivariate adaptive regression splines or support vector machines, to construct a function $u : \Omega \subset \mathbb{R}^D \to \mathbb{R}$ which describes the relationship between the features $\underline{x}$, i.e. the market situation, and the response y, i.e. the trend. This relationship can then be evaluated at a future time t by using the same operator T to transform its corresponding transaction data into a D-dimensional feature vector $\underline{x}$ which describes this new market situation. Since we assume that the market behaves similarly in similar situations, the evaluation of the reconstructed continuous function u in such a new market situation $\underline{x}$ is supposed to yield a good prediction $u(\underline{x})$.

2.3 Regularized least squares regression

In the following we formulate the scattered data approximation problem in D-dimensional space by means of a regularization network approach [8, 13]. As stated above, we assume that the relation between $\underline{x}$ and y in the data set (1) can be described by an unknown function $u : \Omega \subset \mathbb{R}^D \to \mathbb{R}$ which belongs to some space V of functions defined over $\mathbb{R}^D$. The aim is now to recover the function u from the given data S, of some size M, with e.g. $M := J - (K-1) - \hat{k}$, as good as possible. A simple least squares fit of the data would surely result in an ill-posed problem. To obtain a well-posed, uniquely solvable problem, we use regularization theory and impose additional smoothness constraints on the solution of the approximation problem. In our approach this results in the variational problem

$$\min_{u \in V} R(u)$$

with

$$R(u) = \frac{1}{M}\sum_{m=1}^{M}(u(\underline{x}_m) - y_m)^2 + \lambda\|Gu\|_{L_2}^2. \tag{2}$$

Here, the mean squared error enforces closeness of u to the data, the regularization term defined by the operator G enforces the smoothness of u, and the regularization parameter λ balances these two terms. Other error measurements can also be suitable. Further details can be found in [8, 11, 31]. Note that there is a close relation to reproducing kernel Hilbert spaces and kernel methods where a kernel is associated to the regularization operator G, see also [28, 33].

3 Sparse grid discretization

In order to compute a numerical solution of (2), we restrict the problem to a finite dimensional subspace $V_N \subset V$ of dimension $\dim V_N = N$. Common data mining methods like radial basis approaches or support vector machines work with global ansatz functions associated to data points which leads to $N = M$. These methods allow to deal with very high-dimensional feature spaces, but typically scale at least quadratically or even cubically with the number of data points and, thus, cannot be applied to the huge data sets prevalent in foreign exchange rate prediction. Instead, we use grid based local basis functions, i.e. finite elements, in the feature space, similarly to the numerical treatment of partial differential equations. With such a basis $\{\varphi_n\}_{n=1}^N$ of the function space V_N we can approximately represent the regressor u as

$$u_N(\underline{x}) = \sum_{n=1}^{N}\alpha_n\varphi_n(\underline{x}). \tag{3}$$

Note that the restriction to a suitably chosen finite-dimensional subspace involves some additional regularization (regularization by discretization [18]) which depends on the choice of V_N. In the following, we simply choose $G = \nabla$ as the smoothing operator. Although this does not result in a well-posed problem in an infinite dimensional function space its use is reasonable in the discrete function space $V_N, N < \infty$, see [11, 12].

Now we plug (3) into (2). After differentiation with respect to the coefficients α_j, the necessary condition for a minimum of $R(u_N)$ gives the linear system of equations [11]

$$(\lambda\mathcal{C} + \mathcal{B}\cdot\mathcal{B}^T)\underline{\alpha} = \mathcal{B}\underline{y}. \tag{4}$$

Here $\mathcal{C}$ is a square $N \times N$ matrix with entries $\mathcal{C}_{n,n'} = M \cdot (\nabla\varphi_n, \nabla\varphi_{n'})_{L_2}$ for $n, n' = 1, \ldots, N$, and $\mathcal{B}$ is a rectangular $N \times M$ matrix with entries $\mathcal{B}_{n,m} = \varphi_n(\underline{x}_m), m = 1, \ldots, M, n = 1, \ldots, N$. The vector $\underline{y}$ contains the response labels $y_m, m = 1, \ldots, M$. The unknown vector $\underline{\alpha}$ contains the degrees of freedom α_n and has length N. A solution of this linear system then gives the vector $\underline{\alpha}$ which spans the approximation $u_N(\underline{x})$ with (3).

3.1 Sparse grid combination technique

Up to now we have not yet been specific what finite-dimensional subspace V_N and what type of basis functions $\{\varphi_n\}_{n=1}^N$ we want to choose. If uniform grids would be used here, we would immediately encounter the curse of dimensionality and could not treat higher dimensional problems. Instead we employ sparse grid subspaces as introduced in [3, 34] to discretize and solve the regularization problem (2), see also [11]. This discretization approach is based on a sparse tensor product decomposition of the underlying function space. In the following we describe the relevant basic ideas, for details see [3, 9, 11, 34].

To be precise, we apply sparse grids in form of the combination technique [15]. There, we discretize and solve the problem on a suitable sequence of small and in general anisotropic grids $\Omega_{\underline{l}}$ of level $\underline{l} = (l_1, \ldots, l_D)$, which have different but uniform mesh sizes $h_d = 2^{-l_d}, d = 1, \ldots, D$, in each coordinate direction. The points of a given grid $\Omega_{\underline{l}}$ are numbered using the multi-index $\underline{i} = (i_1, \ldots, i_D)$ with $i_d \in \{0, \ldots, 2^{l_d}\}$ for $d = 1, \ldots, D$. For ease of presentation, we assume the domain $\Omega = [0, 1]^D$ here and in the following, which can be always achieved by a proper rescaling of the data.

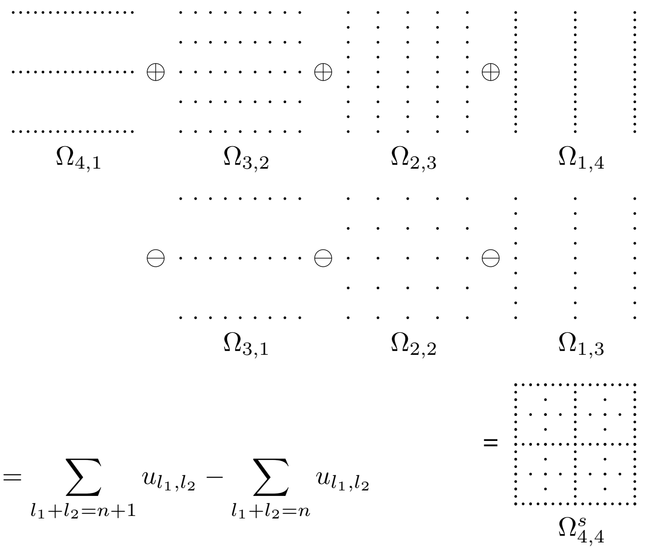


Figure 1: Grids employed by the combination technique of level $L = 4$ in two dimensions.

A finite element approach with piecewise multilinear functions

$$\phi_{\underline{l},\underline{i}}(\underline{x}) \;:=\; \prod_{d=1}^{D} \phi_{l_d,i_d}(x_d), \quad i_d = 0, \ldots, 2^{l_d}, \tag{5}$$

on each grid $\Omega_{\underline{l}}$, where the one-dimensional basis functions $\phi_{l_d,i_d}(x_d)$ are the so-called hat functions

$$\phi_{l_d,i_d}(x_d) = \begin{cases} 1 - |\frac{x_d}{h_{l_d}} - i_d|, & x_d \in [(i_d-1)h_{l_d}, (i_d+1)h_{l_d}] \\ 0, & \text{otherwise,} \end{cases}$$

results in the discrete function space $V_{\underline{l}} \;:=\; \text{span}\{\phi_{\underline{l},\underline{i}}, i_d = 0, ..., 2^{l_d}, d = 1, ..., D\}$ on grid $\Omega_{\underline{l}}$. A function $u_{\underline{l}} \in V_{\underline{l}}$ is then represented as

$$u_{\underline{l}}(\underline{x}) = \sum_{i_1=0}^{2^{l_1}} ... \sum_{i_D=0}^{2^{l_D}} \alpha_{\underline{l},\underline{i}} \phi_{\underline{l},\underline{i}}(\underline{x}).$$

Each multilinear function $\phi_{\underline{l},\underline{i}}(\underline{x})$ equals one at the grid point $\underline{i}$ and is zero at all other points of grid $\Omega_{\underline{l}}$. Its support, i.e. the domain where the function is non-zero, is $\otimes_{d=1}^{D}[(i_d-1)h_{l_d}, (i_d+1)h_{l_d}]$.

To obtain a solution in the sparse grid space V_L^s of level L the combination technique considers all grids $\Omega_{\underline{l}}$ with

$$l_1 + ... + l_D = L + (D-1) - q, \quad q = 0, \ldots, D-1, \quad l_q > 0, \tag{6}$$

see also Figure 1 for an example in two dimensions. One gets an associated system of linear equations (4) for each of the involved grids $\Omega_{\underline{l}}$, which we currently solve by a diagonally preconditioned conjugate gradient algorithm.

The combination technique [15] now linearly combines the resulting discrete solutions $u_{\underline{l}}(\underline{x})$ from the grids $\Omega_{\underline{l}}$ according to the formula

$$u_L^c(\underline{x}) := \sum_{q=0}^{D-1} (-1)^q \binom{D-1}{q} \sum_{|\underline{l}|_1 = L+(D-1)-q} u_{\underline{l}}(\underline{x}). \tag{7}$$

The resulting function u_L^c lives in the sparse grid space V_L^s which has dimension $N = \dim V_L^s = \mathcal{O}(h_L^{-1}(\log(h_L^{-1}))^{D-1})$, see [3]. It therefore depends on the dimension D to a much smaller degree than a function on the corresponding uniform grid $\Omega_{(L,\ldots,L)}$ whose number of degrees of freedom is $\mathcal{O}(h_L^{-D})$. Note that for the approximation of a function u by a sparse grid function $u_L^c \in V_L^s$ the error relation

$$||u - u_L^c||_{L_p} = \mathcal{O}(h_L^2 \log(h_L^{-1})^{D-1})$$

holds, provided that u fulfills certain smoothness requirements which involve bounded second mixed derivatives [3]. The combination technique can be further generalized [9, 17] to allow problem dependent coefficients.

Note that we never explicitly assemble the function u_L^c but instead keep the solutions $u_{\underline{l}}$ which arise in the combination technique (6). If we now want to evaluate the solution at a newly given data point $\underline{\tilde{x}}$ by

$$\tilde{y} := u_L^c(\underline{\tilde{x}}),$$

we just form the combination of the associated point values $u_{\underline{l}}(\underline{\tilde{x}})$ according to (7). The cost of such an evaluation is of the order $\mathcal{O}(L^{D-1})$.

4 Numerical results

We now present results for the prediction of intraday foreign exchange rates with our sparse grid combination technique. Our aim is to forecast the EUR/USD exchange rate. First, we use just the EUR/USD exchange rate time series as input and employ a delay embedding of this single time series. Here we compare the performance with that of a traditional trading strategy using only EUR/USD information. We then also take the other exchange rates into account and show the corresponding results. Furthermore, we present a strategy which involves trading on strong signals only to cope with transaction costs. Based on that, we finally present a trading strategy which in addition reduces the amount of invested capital. Moreover, we compare these approaches and demonstrate their properties in numerical experiments.

4.1 Experimental data

The data were obtained from Olsen Data, a commercial data provider. In the following, we employ the exchange rates from 01.08.2001 to 28.07.2005 between EUR/USD (denoted by €), GBP/USD (£), USD/JPY (¥) and USD/CHF (Fr.). To represent a specific currency pairing we will use the above symbols instead of f_r in the following. For this data set the data provider mapped the recorded raw intraday tick data by piecewise constant interpolation to values $f_r(t_j)$ at equidistant points in time which are $\tau = 3$ minutes apart. No data is generated if in the time interval $[t_j - \tau, t_j]$ no raw tick is present. Due to this, the data set contains a multitude of gaps, which can be large when only sparse trading takes place, for example over weekends and holidays. The properties of this input data concerning these gaps is shown in Table 1. Here, the total number of ticks in the time frame would be 701,280 for each currency pair, but between 168,000 and 186,000 ticks are missing due to the above reasons. The number of gaps varies between about 4,000 and 6,000 while the gap length varies between one and about 900 with an average of about 30. These characteristics are similar for the four currency pairs.

Note that the trading volumes are not constant during the day. The main trading starts each day in the East-Asian markets with Tokyo and Sydney as centers, then the European market with London and Frankfurt dominates, while the main trading activity takes place during the overlap of the European business hours and the later starting American market with New York as the hub [16, 20].

For the following experiments with the sparse grid regression approach the associated input data set S is obtained from the given tick data. Note that the embedding operator T at a time t_j depends on a certain number of delayed data positions between $t_j - (K-1)\tau$ and t_j. Typically not all time positions in the backward time horizon are employed for a given T. Nevertheless, the feature vector at time t_j can only be computed if the data at the positions necessary for T are present, although small data gaps in between these required points are allowed. Note here that we employ the common practice of restricting the values of outliers to a suitable chosen maximum value. Afterwards we linearly map the derived delay embedded features into $[0,1]^D$.

In all our experiments we attempt to forecast the change in the EUR/USD exchange rate. The aim of our regression approach is to predict the relative rate difference $y(t_j) = (€(t_j + \hat{k}\tau) - €(t_j))/€(t_j)$ at $\hat{k}$ steps into the future (*future step*) in comparison to the current time. Such a forecast is often also called (trading) signal.

exchange rate	total ticks	missing ticks	number of gaps	max. gap length	avg. gap length
EUR/USD	701,280	168,952	6,403	879	26
USD/CHF	701,280	171,015	6,192	920	27
USD/JPY	701,280	184,264	4,144	911	44
GBP/USD	701,280	185,442	5,278	912	35

Table 1: Total and missing number of ticks, number of gaps, and maximum and average gap length of the input data.

For the experiments we separate the available data into training data (90%) and test data (10%), this split is done on the time axis. On the training data we perform 3-fold cross-validation (again splitting in time) to find good values for the level parameter L from (7) and the regularization parameter λ from (2) of our regression approach. To this end, the training data set is split into three equal parts. Two parts are used in turn as the learning set and the quality of the regressor (see the following section) is evaluated on the remaining part for varying L and λ. The pair of values of L and λ which performs best in the average of all three splittings is then taken as the optimum and is used for the forecast and final evaluation on the 10% remaining newest test data.

4.2 Quality assessment

To judge the quality of the predictions by our sparse grid combination technique for a given number M of data we use the so-called realized potential

$$rp := cp/mcp$$

as the main measurement. Here cp is the cumulative profit

$$cp := \sum_{m=1}^{M} \frac{\operatorname{sign}(u_L^c(\underline{x}_m)) \cdot (f_1(t_m + \hat{k}\tau) - f_1(t_m))}{f_1(t_m)},$$

i.e. the sum of the actual gain or loss in the exchange rate realized by trading at the M time steps according to the forecast of the method, while mcp is the maximum possible cumulative profit

$$mcp := \sum_{m=1}^{M} \frac{|f_1(t_m + \hat{k}\tau) - f_1(t_m)|}{f_1(t_m)},$$

i.e. the gain when the exchange rate would have been predicted correctly for each trade. For example $M = J - (K-1) - \hat{k}$ if we would consider the whole training data from Section 2.2. Note that these measurements also take the amplitude of the potential gain or loss into account. According to practitioners, a forecasting tool which achieves a realized potential rp of 20% starts to become useful.

Furthermore, we give the prediction accuracy pa, often also called hit rate or correctness rate,

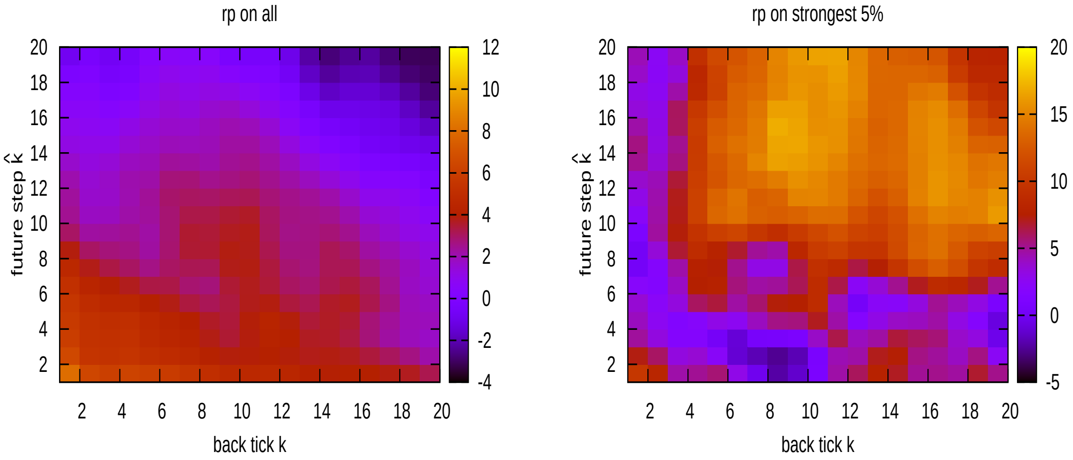
$$pa := \frac{\#\{u_L^c(\underline{x}_m) \cdot (f_1(t_m + \hat{k}\tau) - f_1(t_m)) > 0\}_{m=1}^{M}}{\#\{u_L^c(\underline{x}_m) \cdot (f_1(t_m + \hat{k}\tau) - f_1(t_m)) \neq 0\}_{m=1}^{M}}$$

which denotes the percentage of correctly predicted forecasts. Prediction accuracies of more than 55 % are often reported as worthwhile results for investors [1, 32]. So far, all these measurements do not yet directly take transaction costs into account. We will address this aspect later in Section 4.5 in more detail.

4.3 Forecasting using a single currency pair

In a first set of experiments we aim to forecast the EUR/USD exchange rate from the EUR/USD exchange data. We begin with using one feature, the normalized discrete first derivative

$$\tilde{\text{€}}'_k = \frac{\text{€}(t_j) - \text{€}(t_j - k\tau)}{k\tau\text{€}(t_j - k\tau)}.$$


rp on all

rp on strongest 5%

Figure 2: Realized potential rp for the currency pair EUR/USD for all predictions (left) and for the 5% ticks with the strongest predictions (right), $L = 4$ and $\lambda = 0.0001$.

1 feature	$\lambda = 0.0001$			$\lambda = 0.001$			$\lambda = 0.01$			$\lambda = 0.1$		
$\tilde{\text{€}}'_9$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$
Level 2	1.89	1.72	50.7	2.10	1.91	50.6	2.40	2.18	50.6	2.40	2.18	50.6
Level 3	3.02	2.76	51.5	2.44	2.22	51.0	2.26	2.05	50.8	2.40	2.18	50.6
Level 4	**3.37**	**3.07**	**51.9**	3.08	2.81	51.6	2.33	2.12	51.0	2.40	2.18	50.6

Table 2: 3-fold cross-validation results for the forecast of EUR/USD using $\hat{k} = 15$ and the feature $\tilde{\text{€}}'_9$ for varying refinement level L and regularization parameter λ.

Here, *back tick* k is a parameter to be determined as is $\hat{k}$, the time horizon for the forecast into the future. The results of experiments for the prediction of the EUR/USD exchange rate from the first derivative for several values of k and $\hat{k}$ are shown in Figure 2. We observe the best results[1] for $k = 9$ and $\hat{k} = 15$ which we will use from now on. Since we consider a single currency pair we obtain just a one-dimensional problem here. The combination technique then falls back to conventional discretization.

In Table 2 we give the results of the 3-fold cross-validation on the training data for several λ and L. We observe the highest rp for $\lambda = 0.0001$ and $L = 4$. Using these parameters we now learn on all training data. The evaluation on the remaining 10% test data then results in $cp = 0.741$, $rp = 2.29\%$, and $pa = 51.5\%$ on 51,056 trades. Of course, such small values for rp and pa are far from being practically relevant. Therefore we investigate in the following different strategies to improve performance. We start by adding an additional feature.

To this end, we consider a two-dimensional regression problem where we take – besides $\tilde{\text{€}}'_9$ – the normalized first derivative $\tilde{\text{€}}'_4$ as the second attribute. We choose the value $k = 4$ for the back tick since the combination with the first derivative $\tilde{\text{€}}'_9$ can be interpreted as an approximation to a normalized second derivative

$$\tilde{\text{€}}''_k = \frac{\text{€}(t_j) - 2\text{€}(t_j - k\tau) + \text{€}(t_j - 2k\tau)}{(k\tau)^2 \text{€}(t_j - k\tau)}$$

[1] In particular we take here the performance on the stronger signals into account. To do that we consider the 5% ticks for which we obtain the strongest predictions.

2 features	$\lambda = 0.0001$			$\lambda = 0.001$			$\lambda = 0.01$			$\lambda = 0.1$		
$\tilde{€}'_9, \tilde{€}'_4$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$
Level 2	3.81	3.48	51.9	3.53	3.22	51.6	2.42	2.2	50.6	2.41	2.19	50.6
Level 3	**4.69**	**4.29**	**52.6**	4.57	4.17	52.4	3.56	3.25	51.8	2.25	2.05	50.8
Level 4	4.58	4.18	52.5	4.61	4.21	52.5	4.29	3.91	52.4	2.39	2.17	51.0

Table 3: 3-fold cross-validation results for the forecast of EUR/USD using $\hat{k} = 15$ and the features $\tilde{€}'_9$ and $\tilde{€}'_4$ for varying refinement level L and regularization parameter λ.

	features	data points	L	λ	trades	$pa\%$	cp	mcp	$rp\%$
all ticks									
1 feature	$\tilde{€}'_9$	510,553	4	0.0001	51,056	51.5	0.741	32.37	2.29
2 features	$\tilde{€}'_9, \tilde{€}'_4$	508,616	3	0.0001	50,862	52.1	1.084	32.28	3.36
signal $> 10_{-4}$									
1 feature	$\tilde{€}'_9$	510,553	4	0.0001	460	56.7	0.070	0.650	10.8
2 features	$\tilde{€}'_9, \tilde{€}'_4$	508,616	3	0.0001	916	58.6	0.291	1.206	24.2

Table 4: Forecast of the EUR/USD for $\hat{k} = 15$ on the 10% remaining test data using first derivatives of the EUR/USD exchange rate.

with $k = 4$. The use of two first derivatives $\tilde{€}'_9$ and $\tilde{€}'_4$ captures more information in the data than just the second derivative would.

The results from the 3-fold cross-validation on the training data are shown in Table 3. Again we pick the best parameters and thus use $\lambda = 0.0001$ and $L = 3$ for the prediction on the 10% remaining test data. The additional attribute $\tilde{€}'_4$ results in a significant improvement of the performance: We achieve $cp = 1.084$, $rp = 3.36\%$, and $pa = 52.1\%$ on 50862 trades, see also Table 4 for the comparison with the former experiment using only one feature. In particular we observe that rp grows by about 50%. Furthermore, we observe that the profit is to a significant degree in the stronger signals. If we only take predictions into account which indicate an absolute change[2] larger than 10_{-4}, we trade on 916 signals and achieve $cp = 0.291$, $rp = 24.2\%$ and $pa = 58.6\%$, see Table 4. Thus, trading on 1.8% of the signals generates 26.8% of the overall profit. Note again that a rp-value of more than 20% and a pa-value of more than 55% is often considered practically relevant. Therefore trading on the stronger signals may result in a profitable strategy. Nevertheless, the use of just two features is surely not yet sufficient.

Before we add more features we need to put the performance of our approach into context. To this end, we compare with results achieved by the moving average-oscillator, a widely used technical trading rule [2]. Here buy and sell signals are generated by two moving averages, a long-period average x_l and a short-period average x_s. They are computed according to

$$x_{\{s,l\}}(t_j) = \frac{1}{w_{\{s,l\}}} \sum_{i=0}^{w_{\{s,l\}}-1} €(t_{j-i}),$$

where the length of the associated time intervals is denoted by w_l and w_s, respectively. To handle small gaps in the data, we allow up to 5% of the tick data to be missing in a time interval when computing an average, which we scale accordingly in such a case. Furthermore we neglect data positions in our experiments where no information at time t_j, or $t_j + \tau\hat{k}$ is present.

In its simplest form this strategy is expressed as buying (or selling) when the short-period moving average rises

[2] Observe that a change of 10_{-4} in our target attribute is roughly the size of a pip (the smallest unit of the quoted price) for EUR/USD.

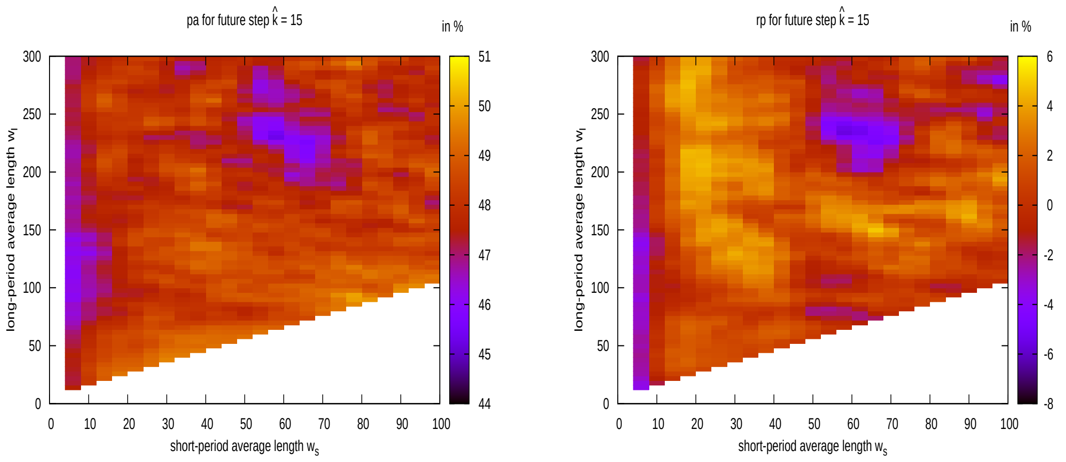


Figure 3: The prediction accuracy and the realized potential on the training data for the fixed-length moving average trading strategy for $\hat{k} = 15$ and varying lengths of the moving averages.

above the long-period moving average by an amount larger than a prescribed *band*-parameter b, i.e.

$$x_s(t_j) > b \cdot x_l(t_j)$$

(or falls below it, i.e. $x_s(t_j) < (2 - b) \cdot x_l(t_j)$). This approach is called variable length moving average. The band-parameter b regulates the trading frequency.

This conventional technical trading strategy is typically used for predictions on much longer time frames and did not achieve any profitable results in our experiments. Therefore we considered a different moving average strategy which performed better. Here, a buy signal is generated as above at time t_j when $x_s(t_j) > b \cdot x_l(t_j)$, but a trade only takes place if $x_s(t_{j-1}) \leq b \cdot x_l(t_{j-1})$ holds as well. Such a position is kept for a number $\hat{k}$ of time steps and is then closed. In the same way, sell signals are only acted upon if both conditions (with reversed inequality sign) are fulfilled. Here, several positions might be held at a given time. This rule is called fixed-length moving average (FMA) and stresses that returns should be stable for a certain time period following a crossover of the long- and short-period averages [2].

In Figure 3 we give the results for EUR/USD of the fixed-length moving average technical rule on the training data. Here, we vary the intervals for both the long-period and the short-period average while using a fixed time horizon in the future of $\hat{k} = 15$. We use the prediction at 15 time steps into the future for two reasons: First we want to be able to compare the results with that of our other experiments which employ the same time horizon, and, second, this value turned out to be a very good choice for the FMA trading rule. As the parameters which achieve the highest rp on the training data we found $w_s = 20$, $w_l = 216$ and $b = 1.000001$.

With these values we obtain $cp = 0.026$, $rp = 9.47\%$, and $pa = 47.4\%$ on the remaining 10% test data using a total of 414 trades. Although the rp with FMA is higher in comparison to the results of our approach when trading on all signals in the test data (compare with the first two rows of Table 4), much less trading takes place here. This small amount of trading is the reason for the quite tiny cp for FMA which is almost 40 times smaller. In addition the prediction accuracy for the signals where trading takes place is actually below 50%. But if we compare the results of FMA with our approach which acts only on the stronger signals $> 10_{-4}$ we outperform the FMA strategy on all accounts (compare with the last two rows of Table 4).

2 features	$\lambda = 0.0001$			$\lambda = 0.001$			$\lambda = 0.01$			$\lambda = 0.1$		
$\tilde{€}'_9, \tilde{\text{Fr.}}'_9$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$
Level 2	4.96	4.56	52.2	4.63	4.25	51.9	2.76	2.53	50.7	2.47	2.27	50.7
Level 3	**4.98**	**4.58**	**52.3**	4.76	4.38	52.2	3.40	3.12	51.2	2.47	2.27	50.7
Level 4	4.57	4.21	52.0	4.89	4.49	52.1	4.02	3.69	51.6	2.51	2.30	50.7

3 features	$\lambda = 0.0001$			$\lambda = 0.001$			$\lambda = 0.01$			$\lambda = 0.1$		
$\tilde{€}'_9, \tilde{\text{Fr.}}'_9, \tilde{£}'_9$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$
Level 2	4.84	4.49	52.2	4.63	4.29	52.0	3.23	2.98	50.9	2.56	2.36	50.7
Level 3	4.71	4.38	52.3	**4.90**	**4.56**	**52.3**	4.21	3.91	51.8	2.68	2.47	50.7
Level 4	4.28	3.97	52.2	4.84	4.49	52.1	4.60	4.27	51.9	2.74	2.52	50.9

4 features	$\lambda = 0.0001$			$\lambda = 0.001$			$\lambda = 0.01$			$\lambda = 0.1$		
$\tilde{€}'_9, \tilde{\text{Fr.}}'_9, \tilde{£}'_9, \tilde{¥}'_9$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$	cp	$rp\%$	$pa\%$
Level 2	4.26	3.96	52.1	4.40	4.08	51.9	3.24	3.01	51.1	2.61	2.41	50.6
Level 3	4.42	4.11	52.2	**4.61**	**4.29**	**52.2**	4.21	3.92	51.9	2.50	2.32	50.9
Level 4	3.69	3.43	51.9	4.30	4.00	52.1	4.50	4.18	52.0	3.18	2.95	51.3

Table 5: 3-fold cross-validation results for the forecast of EUR/USD using $\hat{k} = 15$ and features derived from different exchange rates for varying refinement level L and regularization parameter λ. Results for just $\tilde{€}'_9$ are given in Table 2.

4.4 Forecasting using multiple currency pairs

Now we are interested in the improvement of the prediction of the EUR/USD exchange rate if we also take the other currency pairs £, ¥, and Fr. into account. This results in a higher-dimensional regression problem. We employ first derivatives using the same backticks as before for the different currency pairs.[3] Note that the number of input data points decreases slightly when we add further exchange rate pairs since some features cannot be computed any longer due to overlapping gaps in the input data.

For now we only consider the first derivatives for $k = 9$ to observe the impact due to the use of additional currency pairs. According to the best rp we select which of the three candidates $\tilde{\text{Fr.}}'_9, \tilde{£}'_9, \tilde{¥}'_9$ is successively added. For example $\tilde{\text{Fr.}}'_9$ in addition to $\tilde{€}'_9$ gave the best result using two currency pairs to predict EUR/USD. We then add $\tilde{£}'_9$ before using $\tilde{¥}'_9$. As before, we select the best parameters L and λ for each number of features according to the rp achieved with 3-fold cross-validation on the training data, see Table 5. Note that the values of L and λ with the best performance do not vary much in these experiments. This indicates the stability of our parameter selection process.

Using the combination with the best performance in the 3-fold cross-validation we then learn on all training data and evaluate on the before unseen test data. The results on the training data are given in Table 6, both for the case of all data and again for the case of absolute values of the signals larger than 10_{-4}. Note that the performance on the training data in Table 5 suggests to employ the first two or three attributes. In any case, the use of information from multiple currencies results in a significant improvement of the performance in comparison to just using one attribute derived from the exchange rate to be predicted. The results on the test data given in Table 6 confirm that the fourth attribute $\tilde{¥}'_9$ does not achieve much of an improvement, whereas the additional features $\tilde{\text{Fr.}}'_9, \tilde{£}'_9$ significantly improve both cp and rp. Trading on signals larger than 10_{-4} now obtains a pa of up to 56.7% and, more importantly, $rp = 20.1\%$ using 3 attributes. This clearly shows the potential of our approach. Altogether, we see the gain in performance which can be achieved by a delay embedding of tick data of several currencies into a higher dimensional regression problem while using a first derivative for each exchange rate.

In a second round of experiments we use two first derivatives with back ticks $k = 9$ and $k = 4$ for each exchange rate. We add step-by-step the different currencies in the order of the above experiment from Table 6. To be precise, we

[3] Different back ticks might result in a better performance, but we restricted our experiments to equal back ticks for reasons of simplicity.

	currencies	data points	L	λ	trades	$pa\%$	cp	mcp	$rp\%$
all ticks									
1 feature	$\tilde{€}'_9$	510,553	4	0.0001	51,056	51.5	0.741	32.37	2.29
2 features	$\tilde{€}'_9, \tilde{\text{Fr.}}'_9$	503,939	3	0.0001	50,394	51.8	1.380	32.08	4.30
3 features	$\tilde{€}'_9, \tilde{\text{Fr.}}'_9, \tilde{£}'_9$	495,654	3	0.001	49,566	51.6	1.469	31.76	4.62
4 features	$\tilde{€}'_9, \tilde{\text{Fr.}}'_9, \tilde{£}'_9, \tilde{¥}'_9$	494,238	3	0.001	49,424	51.7	1.478	31.70	4.66
signal $> 10_{-4}$									
1 feature	$\tilde{€}'_9$	510,553	4	0.0001	460	56.7	0.070	0.650	10.8
2 features	$\tilde{€}'_9, \tilde{\text{Fr.}}'_9$	503,939	3	0.0001	2,318	54.0	0.469	2.504	18.8
3 features	$\tilde{€}'_9, \tilde{\text{Fr.}}'_9, \tilde{£}'_9$	495,654	3	0.001	2,379	54.3	0.516	2.566	20.1
4 features	$\tilde{€}'_9, \tilde{\text{Fr.}}'_9, \tilde{£}'_9, \tilde{¥}'_9$	494,238	3	0.001	3,559	53.8	0.614	3.484	17.6

Table 6: Forecast of EUR/USD for $\hat{k} = 15$ on the 10% remaining test data using one first derivative from multiple currency pairs. Results are for trading on all signals and on signals $> 10_{-4}$.

use $\tilde{\text{Fr.}}'_9$ before $\tilde{\text{Fr.}}'_4$, but both before $\tilde{£}'_9, \tilde{£}'_4$, etc.[4] We thus obtain a higher dimensional regression problem. Again we look for good values for λ and L via 3-fold cross-validation.

In Table 7 we give the results which were achieved on the test data. Note that the numbers obtained on the training data suggest to use the four features $\tilde{€}'_9, \tilde{€}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4$ only; nevertheless we show the test results with the two additional features $\tilde{£}'_9, \tilde{£}'_4$ as well. Again, the use of information from multiple currencies gives an improvement of the performance in comparison to the use of just the attributes which were derived from the EUR/USD exchange rate. In particular cp grows from one to several currencies. With four features based on two first derivatives for each currency pair we now achieve a somewhat better performance for all trading signals than before using several first derivatives, compare Table 6 and Table 7. We obtain $rp = 4.80$ for four attributes in comparison to $rp = 4.62$ with three attributes. The results on the stronger signals are also improved, we now achieve $rp = 25.0\%$ in comparison to $rp = 20.1\%$.

4.5 Towards a practical trading strategy

For each market situation $\underline{x}$ present in the test data, the sparse grid regressor $u^s_L(\underline{x})$ yields a value which indicates the predicted increase or decrease of the exchange rate f_1. So far, trading on *all* signals showed some profit. But if one would include transaction costs this approach would no longer be viable, although low transaction costs are nowadays common in the foreign exchange market. Most brokers charge no commissions or fees whatsoever and the width of the bid/ask spread is thus the relevant quantity for the transaction costs. We assume here for simplicity that the spread is the same whether the trade involves a small or large amount of currency. It is therefore sufficient to consider the profit per trade independent of the amount of currency. Consequently, the average profit per trade needs to be at least above the average spread to result in a profitable strategy. This spread is typically five pips or less for EUR/USD and can nowadays even go down to one pip during high trading with some brokers. Note that in our case one pip is roughly equivalent to a change of 8.5_{-5} of our normalized target attribute for the time interval of the test data with an EUR/USD exchange rate of about 1.2. If the average cp per trade is larger than this value one has a potentially profitable trading strategy. In Table 8 we give this value for the different experiments of the previous section. We see that trading on all signals results in values which are below this threshold. The same can be observed for FMA.[5] However, trading on the strongest signals results in a profitable strategy in the experiments with more attribute combinations. For example, the use of $\tilde{€}'_9, \tilde{€}'_4$ results in 3.0_{-4} cp per trade, $\tilde{€}'_9, \tilde{€}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4$ gives 2.9_{-4} cp per trade and $\tilde{€}'_9, \tilde{\text{Fr.}}'_9, \tilde{£}'_9$ achieves 2.2_{-4} cp per trade. But note that with this strategy one might need to have more than one position open at a given time, which means that more capital is involved. This number of open positions can vary between zero and $\hat{k}$. It is caused by the

[4] Note that a different order might result in a different performance.

[5] Furthermore only relatively few trades take place with FMA which makes this a strategy with a higher variance in the performance.

	currencies	data points	L	λ	trades	$pa\%$	cp	mcp	$rp\%$
all ticks									
1 feature	$\tilde{\text{€}}'_9$	510,553	4	0.0001	51,056	51.5	0.741	32.37	2.29
2 features	$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4$	508,616	3	0.0001	50,862	52.1	1.084	32.28	3.36
3 features	$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9$	503,017	2	0.0001	50,300	52.1	1.315	32.03	4.10
4 features	$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4$	502,243	2	0.001	50,220	52.4	1.536	31.98	4.80
5 features	$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4, \tilde{\pounds}'_9$	494,975	2	0.001	49,497	52.1	1.556	31.73	4.90
6 features	$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4, \tilde{\pounds}'_9, \tilde{\pounds}'_4$	492,965	2	0.001	49,296	52.1	1.538	31.60	4.87
signal $> 10_{-4}$									
1 feature	$\tilde{\text{€}}'_9$	510,553	4	0.0001	460	56.7	0.070	0.650	10.8
2 features	$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4$	508,616	3	0.0001	916	58.6	0.291	1.206	24.2
3 features	$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9$	503,017	2	0.0001	1,811	58.9	0.467	2.048	22.8
4 features	$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4$	502,243	2	0.001	1,557	59.6	0.447	1.785	25.0
5 features	$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4, \tilde{\pounds}'_9$	494,975	2	0.001	2,178	58.7	0.523	2.392	21.9
6 features	$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4, \tilde{\pounds}'_9, \tilde{\pounds}'_4$	492,965	2	0.001	2,711	56.8	0.508	2.796	18.2

Table 7: Forecast of EUR/USD 15 ticks into the future using multiple currency pairs and derivatives on the 10% remaining test data. Results are for trading on all signals and on signals $> 10_{-4}$.

currencies	strategy	cp	trades	cp per trade
$\tilde{\text{€}}'_9$	all ticks	0.741	51,056	1.4_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{Fr.}}'_9$	all ticks	1.380	50,394	2.7_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{Fr.}}'_9, \tilde{\pounds}'_9$	all ticks	1.469	49,566	3.0_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{Fr.}}'_9, \tilde{\pounds}'_9, \tilde{\yen}'_9$	all ticks	1.478	49,424	3.0_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4$	all ticks	1.084	50,862	2.1_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9$	all ticks	1.315	50,300	2.6_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4$	all ticks	1.536	50,220	3.1_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4, \tilde{\pounds}'_9$	all ticks	1.556	49,497	3.1_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4, \tilde{\pounds}'_9, \tilde{\pounds}'_4$	all ticks	1.538	49,296	3.1_{-5}
$\tilde{\text{€}}'_9$	signal $> 10_{-4}$	0.070	460	1.5_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{Fr.}}'_9$	signal $> 10_{-4}$	0.469	2,318	2.0_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{Fr.}}'_9, \tilde{\pounds}'_9$	signal $> 10_{-4}$	0.516	2,379	2.2_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{Fr.}}'_9, \tilde{\pounds}'_9, \tilde{\yen}'_9$	signal $> 10_{-4}$	0.614	3,559	1.7_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4$	signal $> 10_{-4}$	0.291	916	3.0_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9$	signal $> 10_{-4}$	0.467	1,811	2.6_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4$	signal $> 10_{-4}$	0.447	1,557	2.9_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4, \tilde{\pounds}'_9$	signal $> 10_{-4}$	0.523	2,178	2.4_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4, \tilde{\pounds}'_9, \tilde{\pounds}'_4$	signal $> 10_{-4}$	0.508	2,711	1.9_{-4}
€	FMA	0.026	414	6.3_{-5}

Table 8: cp per trade the forecast of EUR/USD for $\hat{k} = 15$ using different attribute selections on the 10% remaining test data.

currencies	strategy	cp	trades	cp per trade
$\tilde{\text{€}}'_9$	opening $= 0$, closing $= 0$	0.416	9,465	4.4_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{Fr.}}'_9$	opening $= 0$, closing $= 0$	0.859	12,897	6.7_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{Fr.}}'_9, \tilde{\pounds}'_9$	opening $= 0$, closing $= 0$	0.824	11,735	7.0_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{Fr.}}'_9, \tilde{\pounds}'_9, \tilde{\text{¥}}'_9$	opening $= 0$, closing $= 0$	0.791	12,029	6.6_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4$	opening $= 0$, closing $= 0$	0.563	12,045	4.7_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9$	opening $= 0$, closing $= 0$	0.931	13,331	7.0_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4$	opening $= 0$, closing $= 0$	1.013	14,178	7.1_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4, \tilde{\pounds}'_9$	opening $= 0$, closing $= 0$	1.042	13,936	7.5_{-5}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4, \tilde{\pounds}'_9, \tilde{\pounds}'_4$	opening $= 0$, closing $= 0$	1.022	14,493	7.1_{-5}
$\tilde{\text{€}}'_9$	opening $= 10_{-4}$, closing $= 0.5 \cdot 10{-}4$	0.018	108	1.7_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{Fr.}}'_9$	opening $= 10_{-4}$, closing $= 0.5 \cdot 10{-}4$	0.117	718	1.6_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{Fr.}}'_9, \tilde{\pounds}'_9$	opening $= 10_{-4}$, closing $= 0.5 \cdot 10{-}4$	0.133	748	1.7_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{Fr.}}'_9, \tilde{\pounds}'_9, \tilde{\text{¥}}'_9$	opening $= 10_{-4}$, closing $= 0.5 \cdot 10{-}4$	0.169	1,213	1.4_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4$	opening $= 10_{-4}$, closing $= 0.5 \cdot 10{-}4$	0.076	248	3.1_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9$	opening $= 10_{-4}$, closing $= 0.5 \cdot 10{-}4$	0.166	667	2.5_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4$	opening $= 10_{-4}$, closing $= 0.5 \cdot 10{-}4$	0.173	593	2.9_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4, \tilde{\pounds}'_9$	opening $= 10_{-4}$, closing $= 0.5 \cdot 10{-}4$	0.176	839	2.1_{-4}
$\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4, \tilde{\pounds}'_9, \tilde{\pounds}'_4$	opening $= 10_{-4}$, closing $= 0.5 \cdot 10{-}4$	0.186	1,177	1.6_{-4}

Table 9: *cp* per trade for the forecast of EUR/USD for $\hat{k} = 15$ using the trading strategy with opening and closing thresholds on the 10% remaining test data.

possibility of opening a position at all times between t_j and $t_j + \hat{k}\tau$, when the first position opened at time t_j is closed again. We observed in our experiments $\hat{k}$ as the maximum number of open positions even when only trading on the stronger signals. This also indicates that a strong signal is present for a longer time period.

To avoid the need for a larger amount of capital we also implement a tradeable strategy where at most one position is open at a given time. Here one opens a position if the buy/sell signal at a time t_j for a prediction at $\hat{k}$ time steps into the future is larger - in absolute values - than a pre-defined *opening threshold*, and no other position is open. The position is closed when a prediction in the opposite direction occurs at some time t_e in the time interval $[t_j, t_j + \tau\hat{k}]$ and the absolute value of that prediction is greater than a prescribed *closing threshold*. At the prediction time $t_j + \tau\hat{k}$ the position is closed, unless a trading signal in the same direction as that of the original prediction is present which is larger than the *opening threshold*. The latter condition avoids an additional, but unnecessary trade. Furthermore, the closing at the forecast time $t_j + \tau\hat{k}$ avoids an open position in situations with no trading activity and where no signals can be generated.

When both of the above thresholds are zero the proposed new strategy is acting on all ticks, but at most one position is open at any given time. Besides the reduction in invested capital this strategy also improves the performance with respect to the *cp* per trade, see top half of Table 9. In comparison to trading on all ticks this strategy improves the results by more than a factor of two while only considering, but not acting on all ticks. Altogether, this strategy is getting close to the profitable threshold of one pip, i.e. 8.5_{-5} in our scaling, but it is still not yet creating a true profit.

However, as observed before, a large part of the profit is generated by acting only on the strong signals. We now set for our new strategy the opening threshold to 10_{-4} and the closing threshold to $0.5 \cdot 10_{-4}$. This adaption of our strategy achieves results which are comparable to the trading on the strong signals only. Since at most one position is open, less capital is involved than in the case of trading on all strong signals. In the bottom half of Table 9 we give the corresponding results. We see that the *cp* per trade is now always above the threshold 8.5_{-5}. The highest *cp* per trade is

3.1_{-4}, where 248 trades take place while using $\tilde{\text{€}}'_9, \tilde{\text{€}}'_4$. For the attributes $\tilde{\text{€}}'_9, \tilde{\text{€}}'_4, \tilde{\text{Fr.}}'_9, \tilde{\text{Fr.}}'_4$ we achieve 2.9_{-4} cp per trade while acting 593 times. This might be preferable due to the larger number of trades which should lead to more stable results. Thus, we finally obtained a profitable strategy, which promises a net gain of more than one pip per trade if the spread is less than two pips.

5 Conclusions

We presented a machine learning approach based on delay embedding and regression with the sparse grid combination technique to forecast the intraday foreign exchange rates of the EUR/USD currency pair. It improved the results to take not only attributes derived from the EUR/USD rate but from further exchange rates like the USD/JPY and/or GBP/USD rate into account. In some situations a realized potential of more than 20% was achieved. We also developed a practical trading strategy using an opening and closing threshold which obtained an average profit per trade larger than three pips. If the spread is on average below three pips this results in profitable trading. Thus, our approach seems to be able to learn the effect of technical trading tools which are commonly used in the intraday foreign exchange market. It also indicates that FX rates have an underlying process which is not purely Markovian, but seems to have additional structure and memory which we believe is caused by technical trading in the market.

Our methodology can be further refined especially in the choice of attributes and parameters. For example, we considered the same time frame for the first derivatives of all the involved currency pairs, i.e. $k = 9$ and $k = 4$. Using different time frames for the different exchange rates might result in a further improvement of the performance. Other intraday information like the variance of the exchange rates or the current spread can also be incorporated. Furthermore, we did not yet take the different interest rates into account, but their inclusion into the forecasting process can nevertheless be helpful. The time of day could also be an useful attribute since the activity in the market changes during the day [20].

In our experiments we used data from 2001 to 2005 to forecast for five months in the year 2005. Therefore, our observations are only based on this snapshot in time of the foreign exchange market. For a snapshot from an other time interval one would most likely use different features and parameters and one would achieve somewhat changed results. Furthermore, it has to be seen if today's market behaviour, which may be different especially after the recent financial crisis, can still be forecast with such an approach, or if the way the technical trading takes place has changed fundamentally. In any case, for a viable trading system, a learning approach is necessary which relearns automatically and regularly over time, since trading rules are typically valid only for a certain period.

Note finally that our approach is not limited to the FX application. In finance it may be employed for the prediction of the technical behaviour of stocks or interest rates as well. It also can be applied to more general time series problems with a large amount of data which arise in many applications in biology, medicine, physics, econometrics and computer science.

Acknowledgements

We thank Bastian Bohn and Alexander Hullmann for their assistance with the numerical experiments.

References

[1] D. J. E. Baestaens, W. M. van den Bergh, and H. Vaudrey. Market inefficiencies, technical trading and neural networks. In C. Dunis, editor, *Forecasting Financial Markets*, pages 245–260. Wiley, 1996.

[2] W. Brock, J. Lakonishok, and B. LeBaron. Simple technical trading rules and the stochastic properties of stock returns. *The Journal of Finance*, 47(5):1731–1764, 1992.

[3] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.

[4] Y.-W. Cheung and C. Y.-P. Wong. Foreign exchange traders in Hong Kong, Tokyo, and Singapore: A survey study. In T. Bos and T. A. Fetherston, editors, *Advances in Pacific Basin Financial Markets*, volume 5, pages 111–134. JAI, 1999.

[5] R. Curcio, C. Goodhart, D. Guillaume, and R. Payne. Do technical trading rules generate profits? Conclusions from the intra-day foreign exchange market. *Int. J. Fin. Econ.*, 2(4):267–280, 1997.

[6] M. A. H. Dempster, T. W. Payne, Y. S. Romahi, and G. W. P. Thompson. Computational learning techniques for intraday FX trading using popular technical indicators. *IEEE Trans. Neural Networks*, 12(4):744–754, 2001.

[7] M. Engel. Time series analysis. Part III Essay, University of Cambridge, 1991.

[8] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13:1–50, 2000.

[9] J. Garcke. Regression with the optimised combination technique. In W. Cohen and A. Moore, editors, *Proceedings of the 23rd ICML '06*, pages 321–328, 2006.

[10] J. Garcke and M. Griebel. Classification with sparse grids using simplicial basis functions. *Intelligent Data Analysis*, 6(6):483–502, 2002.

[11] J. Garcke, M. Griebel, and M. Thess. Data mining with sparse grids. *Computing*, 67(3):225–253, 2001.

[12] J. Garcke and M. Hegland. Fitting multidimensional data using gradient penalties and the sparse grid combination technique. *Computing*, 84(1-2):1–25, April 2009.

[13] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219–265, 1995.

[14] P. Grassberger and I. Procaccia. Characterization of strange attractors. *Phys. Rev. Lett*, 50:346–349, 1983.

[15] M. Griebel, M. Schneider, and C. Zenger. A combination technique for the solution of sparse grid problems. In P. de Groen and R. Beauwens, editors, *Iterative Methods in Linear Algebra*, pages 263–281. IMACS, Elsevier, North Holland, 1992.

[16] D. M. Guillaume, M. M. Dacorogna, R. R. Davé, U. A. Müller, R. B. Olsen, and O. V. Pictet. From the bird's eye to the microscope: A survey of new stylized facts of the intra-daily foreign exchange markets. *Finance Stochast.*, 1(2):95–129, 1997.

[17] M. Hegland, J. Garcke, and V. Challis. The combination technique and some generalisations. *Linear Algebra and its Applications*, 420(2–3):249–275, 2007.

[18] M. Hegland, O. M. Nielsen, and Z. Shen. Multidimensional smoothing using hyperbolic interpolatory wavelets. *Electronic Transactions on Numerical Analysis*, 17:168–180, 2004.

[19] I. Horenko. Finite element approach to clustering of multidimensional time series. *SIAM Journal on Scientific Computing*, 2008, to appear.

[20] K. Iwatsubo and Y. Kitamura. Intraday evidence of the informational efficiency of the yen/dollar exchange rate. *Applied Financial Economics*, 19(14):1103–1115, 2009.

[21] H. Kantz and T. Schreiber. *Nonlinear time series analysis*. Cambridge University Press, 1997.

[22] K. Lien. *Day Trading the Currency Market*. Wiley, 2005.

[23] Y.-H. Lui and D. Mole. The use of fundamental and technical analyses by foreign exchange dealers: Hong Kong evidence. *J. Int. Money Finance*, 17(3):535–545, 1998.

[24] A. L. M. Verleysen, E. de Bodt. Forecasting financial time series through intrinsic dimension estimation and non-linear data projection. In J. Mira and J. V. Sánchez-Andrés, editors, *Engineering Applications of Bio-Inspired Artificial Neural Networks, Volume II*, volume 1607 of *Lecture Notes in Computer Science*, pages 596–605. Springer, 1999.

[25] C. J. Neely and P. A. Weller. Intraday technical trading in the foreign exchange market. *J. Int. Money Finance*, 22(2):223–237, 2003.

[26] C. Osler. Support for resistance: Technical analysis and intraday exchange rates. *Economic Policy Review*, 6(2):53–68, 2000.

[27] T. R. Reid. *The United States of Europe, The New Superpower and the End of the American Supremacy*. Penguin Books, 2004.

[28] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[29] F. Takens. Detecting strange attractors in turbulence. In D. Rand and L.-S. Young, editors, *Dynamical Systems and Turbulence*, volume 898 of *Lecture Notes in Mathematics*, pages 366–381. Springer, 1981.

[30] M. P. Taylor and H. Allen. The use of technical analysis in the foreign exchange market. *J. Int. Money Finance*, 11(3):304–314, 1992.

[31] A. N. Tikhonov and V. A. Arsenin. *Solutions of ill-posed problems*. W. H. Winston, Washington D.C., 1977.

[32] G. Tsibouris and M. Zeidenberg. Testing the efficient markets hypothesis with gradient descent algorithms. In A.-P. Refenes, editor, *Neural Networks in the Capital Markets*, chapter 8, pages 127–136. Wiley, 1995.

[33] G. Wahba. *Spline models for observational data*, volume 59 of *Series in Applied Mathematics*. SIAM, Philadelphia, 1990.

[34] C. Zenger. Sparse grids. In W. Hackbusch, editor, *Parallel Algorithms for Partial Differential Equations, Proceedings of the Sixth GAMM-Seminar, Kiel, 1990*, volume 31 of *Notes on Num. Fluid Mech.* Vieweg-Verlag, 1991.

[35] H. G. Zimmermann, R. Neuneier, and R. Grothmann. Multi-agent modeling of multiple FX-markets by neural networks. *IEEE Trans. Neural Networks*, 12(4):735–743, 2001.