The Relevance of Trends for Predictions of Stock Returns

Thomas Hellström[1] and Kenneth Holmström[2]
Center of Mathematical Modeling (**CMM**)
Department of Mathematics and Physics
Mälardalen University
S-721 23 Västeras, Sweden

March 29, 1998

Abstract

Technical prediction of stock returns is known to be an extremely difficult problem. In this report we use the concept of trends as predictor variables. A statistical investigation of the relevance of a trend concept for predictions of future returns is presented for individual stocks on the Swedish stock market. Negative correlation for strong negative trends is shown. The overall correlation between trends and future returns is however found to be very weak. Trend variables are further used as input variables in a k-nearest neighbor analysis to find patterns of trend values that result in non-random future returns. The k-nearest neighbor algorithm is extended with a selection procedure to find regions in the input space where the future returns are non-symmetrically distributed. The new algorithm is successfully tested on artificial stock data with trending patterns introduced. The algorithm is also applied to a number of national stock indexes: the American Dow Jones, the German DAX, the British FTSE, and the Swedish Generalindex. The results are positve for some of the tested indexes and negative for others. Further tests must be conducted in order to give statistically significant results.

The suggested algorithm also has natural application in areas other than stock prediction. It addresses situations in which the overall predictability is low and can only be expected to apply in indeterminate regions of the input space.

KEYWORDS: Clustering, Data Mining, Finance, k-nearest neighbors, Prediction, Statistics, Stock returns.

[1]also Department of Computing Science at Umeå University. thomash@cs.umu.se
[2]hkh@mdh.se

1 Introduction

The purpose of this report is to examine the concept of *trends* and how it can be utilized for predictions of stock returns. A statistical analysis of stock data from the Swedish stock market over the period 1987-1996 shows how the trends are correlated to future returns.

A modified k-nearest neighbor algorithm identifies regions in the input space where a correlation exists and improves prediction performance by issuing a "don't know" answer where no correlation can be found.

2 Descriptive analysis of data

The data examined in this section is collected from the Swedish stock market over the period 1987-1996. Results for two sets of stocks are presented: *SXG* which represents 33 major stocks with active trading and *SXBIG* which represents 210 major and minor stocks (including those in *SXG*).

The k-step return $R_k(t)$ of a stock-price time series $y(t)$ is defined as

$$R_k(t) = 100 \cdot \frac{y(t) - y(t-k)}{y(t-k)}. \tag{1}$$

The returns $R_k(t)$ are the primary target in most research on the predictability of stocks. Some of the reasons for this are:

1. $R_K(t)$ has a relatively constant range even if many years of data have been used as input. The prices $y(t)$ obviously vary much more and make it difficult to create a valid model for a longer period of time.

2. $R_K(t)$ for different stocks may also be compared on an equal basis (however, this is seldom done in published research).

3. It is easy to evaluate a prediction algorithm for $R_K(t)$ by computing the prediction accuracy of the sign of $R_K(t)$. A long time accuracy above 50% (or more precisely above the historical mean) indicates that a true prediction has taken place.

The basic statistical properties of $R_K(t)$ for the two sets of stocks are listed in tables 1 and 2. The values in the tables are mean values for the included stocks. Each column presents data for one particular value of k.

The last six lines in the tables show the distribution of signs for the returns. "$Return = 0$" is the fraction of returns that is equal to zero. "$Return > 0$" is the fraction of returns that is greater than zero and "$Return < 0$" is the fraction of returns that is less than zero. *"Up fraction"* is computed as

$$100 \cdot \frac{"Return > 0"}{"Return > 0" + "Return < 0"}, \tag{2}$$

which is the positive fraction of all non-zero moves. *"Up fraction"* is a relevant measure, when it comes to evaluating the hit rate of prediction algorithms. Looking at one-step returns in the tables, the "$Up\ fraction$" for *SXG* is 50.9% and for *SXBIG* is 50.6%. The "$Mean\ Up$" and "$Mean\ Down$" rows show the mean value on the positive and negative returns respectively.

The fractions of zero returns in the data material are somewhat surprisingly high, 14.0% for *SXG* and 23.4% for *SXBIG*. The higher value in the latter set is related to the lower degree of activity in the smaller stocks included in *SXBIG*. The zero returns must be dealt with in a proper way when evaluating hit rates for prediction algorithms. The "$Up\ fraction$" circumvents the zero returns by simply removing them before calculating the hit rate. In this way, the zero returns will be counted as both increases and decreases, in equal proportions. A similar procedure is proposed in [2] for test metrics when making stock predictions.

3 Trends

A trend-following-trading strategy normally means buying stocks, which have shown a positive trend for the last days, weeks or months. It also suggests selling stocks, which have shown a negative trend. In this section the relevance of such a strategy will be tested statistically.

A trend $T_k(t)$ is defined using the k-step return as

$$T_k(t) = \frac{100}{k} \cdot \frac{y(t) - y(t-k)}{y(t-k)}. \tag{3}$$

By setting k at different numbers we get measures telling how much the stock has increased per day since its value k days ago.

To see if $T_k(t)$ is correlated to future changes, define the profit $P_h(t)$ computed h days ahead as

$$P_h(t) = 100 \cdot \frac{y(t+h) - y(t)}{y(t)}. \tag{4}$$

$P_h(t)$ is obviously equal to $R_h(t+h)$ (i.e. it is achieved by shifting the returns h days backwards).

In Table 3 mean profit $P_1(t)$ is tabulated as a function of trend $T_k(t)$, i.e. 1-step-forward profit versus k-step-backward trends, for stocks *SXG*, over the years 1987-1996. Table 4 shows the "$Up\ fraction$" (2) and Table 5 the number of observations in each table entry. Each column represents one particular value on k covering the values $1, 2, 5, 10, 20, 50, 100$. Note that the time series normally have 5 samples per week, i.e. $k = 5$ represents one week of data and $k = 20$ represents approximately one month.

Tables 6, 7 and 8 show the same, but with $P_5(t)$ and $T_k(t)$, i.e. 5-step-forward profit per day versus k-step-backward trends per day.

To ensure that found patterns reflect fundamental properties of the process generating the data, and not only idiosyncrasies in the data, the relation between trends and future returns are also presented in graphs where one curve represents one year. Figures 1

Table 1: Mean k-step returns for 33 major Swedish stocks (SXG)

	k						
	1	**2**	**5**	**10**	**20**	**50**	**100**
Mean	0.098	0.197	0.479	0.960	1.953	4.871	9.687
Median	0.000	0.029	0.162	0.562	1.533	4.217	7.432
Std. dev	2.17	3.16	5.06	7.23	10.54	18.01	28.08
Skewness	0.53	0.71	0.78	0.71	0.59	0.56	0.63
Kurtosis	12.60	12.19	12.18	10.61	8.51	6.03	5.38
No of points	2090	2088	2084	2078	2065	2036	1987
Returns $= 0$ **(%)**	14.0	9.2	5.4	3.5	2.3	1.2	0.8
Returns > 0 **(%)**	43.8	46.4	49.8	52.5	56.8	61.1	63.9
Returns < 0 **(%)**	42.2	44.4	44.8	43.9	40.9	37.7	35.4
Up fraction (%)	50.9	51.1	52.7	54.4	58.2	61.8	64.3
Mean Up	1.8	2.6	4.1	5.9	8.6	15.3	24.8
Mean Down	-1.6	-2.2	-3.4	-4.8	-6.9	-11.2	-15.9

Table 2: Mean k-step returns for 210 Swedish stocks (SXBIG)

	k						
	1	**2**	**5**	**10**	**20**	**50**	**100**
Mean	0.142	0.272	0.584	1.060	2.011	4.605	8.695
Median	0.000	0.009	0.057	0.254	0.948	2.905	5.216
Std. dev	3.01	4.13	6.12	8.39	11.76	18.76	27.69
Skewness	0.79	1.06	1.02	0.93	0.83	0.77	0.81
Kurtosis	15.75	16.40	11.45	9.21	7.53	5.94	5.55
No of points	1353	1349	1342	1333	1319	1291	1244
Returns $= 0$ **(%)**	23.4	16.9	10.7	7.4	4.9	2.7	1.8
Returns > 0 **(%)**	38.7	42.0	45.7	48.5	52.1	56.3	57.9
Returns < 0 **(%)**	37.9	41.1	43.6	44.1	43.0	41.0	40.3
Up fraction (%)	50.6	50.6	51.2	52.4	54.8	57.8	58.9
Mean Up	2.7	3.5	5.2	7.1	10.1	16.7	26.4
Mean Down	-2.3	-2.9	-4.0	-5.3	-7.3	-11.3	-15.5

Table 3: Mean 1-step returns for 33 stocks in SXG

	k-day trend (%/day)												
k	**-5.00**	**-4.00**	**-3.00**	**-2.00**	**-1.00**	**-0.50**	**0.00**	**0.50**	**1.00**	**2.00**	**3.00**	**4.00**	**5.00**
1	0.53	-0.03	-0.05	-0.04	-0.12	-0.08	0.01	0.11	0.19	0.29	0.38	0.40	0.72
2	1.20	0.22	0.13	-0.08	-0.06	-0.08	0.03	0.13	0.24	0.31	0.28	0.31	0.42
3	1.89	0.08	0.15	0.17	-0.03	-0.07	0.03	0.15	0.20	0.25	0.27	-0.03	0.35
4	3.19	0.48	-0.01	0.13	0.00	-0.04	0.03	0.16	0.20	0.17	0.41	-0.04	0.81
5	3.47	1.03	0.18	-0.02	0.02	0.00	0.05	0.15	0.20	0.15	0.19	0.57	1.11
10	10.64	7.39	0.40	-0.07	0.01	0.00	0.07	0.15	0.17	0.29	0.29	0.44	0.56
20			8.25	1.26	-0.12	-0.05	0.09	0.14	0.20	0.32	0.06	3.28	1.29
30				2.63	0.04	-0.07	0.09	0.13	0.22	0.33	2.19	1.24	1.49
50					0.49	-0.07	0.08	0.14	0.10	1.10	-0.17	1.95	1.21
100					4.68	0.18	0.05	0.14	0.27	0.57	0.78	0.29	0.81

Table 4: Fraction up/(up+down) moves (%) for stocks in SXG

	k-day trend (%/day)												
k	**-5.00**	**-4.00**	**-3.00**	**-2.00**	**-1.00**	**-0.50**	**0.00**	**0.50**	**1.00**	**2.00**	**3.00**	**4.00**	**5.00**
1	56.0	50.6	49.6	48.5	47.2	47.4	49.2	51.3	53.3	54.9	55.8	54.9	54.8
2	58.1	52.3	53.0	49.5	47.7	47.8	49.5	51.8	54.5	54.5	52.9	51.9	48.5
3	62.3	52.0	49.5	53.5	48.8	47.3	49.8	52.7	53.6	53.6	51.5	42.3	47.5
4	64.5	56.0	49.0	50.6	50.1	47.9	50.0	52.9	54.0	51.1	49.4	43.2	50.0
5	60.6	60.0	50.7	50.1	50.1	48.9	50.2	52.9	52.7	50.1	48.3	49.1	53.3
10	100.0	73.3	52.7	50.8	49.4	48.9	50.8	52.9	50.4	51.8	50.2	47.5	43.5
20			77.8	56.3	47.3	47.7	51.4	52.3	50.6	49.0	49.1	50.0	55.6
30				56.1	48.0	47.5	51.5	51.7	50.2	50.7	55.8	61.1	54.8
50					52.4	46.5	51.2	51.8	48.8	54.4	52.8	57.1	50.0
100					71.4	49.5	50.5	51.6	49.5	54.8	64.7	50.0	53.5

Table 5: Number of points (SXG)

	k-day trend (%/day)												
k	**-5.00**	**-4.00**	**-3.00**	**-2.00**	**-1.00**	**-0.50**	**0.00**	**0.50**	**1.00**	**2.00**	**3.00**	**4.00**	**5.00**
1	1380	1167	2521	6358	8604	7433	12264	7209	8422	6390	3028	1500	1800
2	488	557	1533	4601	9224	10680	12654	10104	9252	5371	2001	754	794
3	242	341	969	3479	8710	11691	15181	11290	9177	4551	1410	477	455
4	115	212	707	2680	8144	12363	17058	11997	9215	3850	949	318	307
5	71	132	546	2136	7398	13075	18264	12742	9129	3211	705	244	214
10	3	30	161	935	4706	13522	23610	15499	7157	1543	286	93	79
20	0	0	9	347	2500	12130	29636	17267	4434	739	123	18	27
30	0	0	0	70	1792	10656	33498	17163	3156	473	48	18	32
50	0	0	0	0	740	8630	38854	15791	1961	216	41	24	42
100	0	0	0	0	7	5616	45695	12134	848	258	40	5	48

Table 6: Mean 5-step returns for 33 stocks in SXG

	k-day trend (%/day)												
k	**-5.00**	**-4.00**	**-3.00**	**-2.00**	**-1.00**	**-0.50**	**0.00**	**0.50**	**1.00**	**2.00**	**3.00**	**4.00**	**5.00**
1	1.18	0.08	0.45	0.21	0.23	0.16	0.33	0.43	0.65	0.67	0.66	0.74	1.20
2	1.94	0.79	0.70	0.31	0.25	0.24	0.28	0.51	0.62	0.69	0.45	0.59	1.35
3	2.77	0.79	0.90	0.65	0.32	0.17	0.30	0.53	0.61	0.65	0.61	0.30	1.51
4	4.39	1.06	1.52	0.54	0.30	0.18	0.36	0.53	0.67	0.50	1.13	-0.41	2.52
5	6.30	1.86	1.12	0.35	0.30	0.24	0.36	0.57	0.62	0.57	0.50	1.69	2.11
10	9.84	5.09	1.48	0.13	0.24	0.17	0.41	0.59	0.75	0.73	1.05	0.51	2.69
20			16.02	3.18	-0.21	-0.07	0.44	0.66	0.79	1.30	-1.05	2.80	8.47
30				6.04	-0.06	-0.19	0.48	0.61	0.85	1.67	6.70	-1.06	6.76
50					1.84	-0.23	0.39	0.60	0.76	3.10	4.13	9.78	4.26
100					7.96	1.01	0.26	0.51	1.36	3.25	1.91	4.36	3.81

Table 7: Fraction up/(up+down) moves (%) for stocks in SXG

	k-day trend (%/day)												
k	**-5.00**	**-4.00**	**-3.00**	**-2.00**	**-1.00**	**-0.50**	**0.00**	**0.50**	**1.00**	**2.00**	**3.00**	**4.00**	**5.00**
1	57.1	51.3	51.4	50.9	50.5	50.0	52.1	52.7	54.6	54.1	53.6	52.4	51.3
2	56.1	55.2	55.8	52.0	50.6	51.2	51.6	53.0	54.1	54.1	50.5	50.3	49.3
3	57.0	55.0	55.8	55.4	51.7	50.2	52.0	53.3	53.7	53.2	49.9	48.6	46.0
4	63.2	56.1	59.2	55.1	52.2	50.6	52.2	52.8	54.2	50.7	51.6	43.6	49.7
5	65.2	57.3	57.3	53.5	52.3	51.2	52.0	53.4	53.5	50.7	48.1	52.2	46.4
10	66.7	62.1	51.6	50.0	51.7	49.9	52.9	53.7	52.7	51.3	50.4	51.2	41.3
20			100.0	60.1	47.2	48.4	53.3	54.3	50.5	50.7	40.2	50.0	56.0
30				69.1	46.7	47.8	53.5	53.1	51.6	52.2	63.0	41.2	62.1
50					55.3	46.1	52.8	53.5	48.9	60.0	63.4	63.6	64.1
100					83.3	51.0	51.7	52.9	54.7	64.8	61.1	80.0	63.6

shows 1-step profits versus 1-step trends for *SXG* and *SXBIG* respectively. Figure 2 shows 5-step profit versus k-step trends.

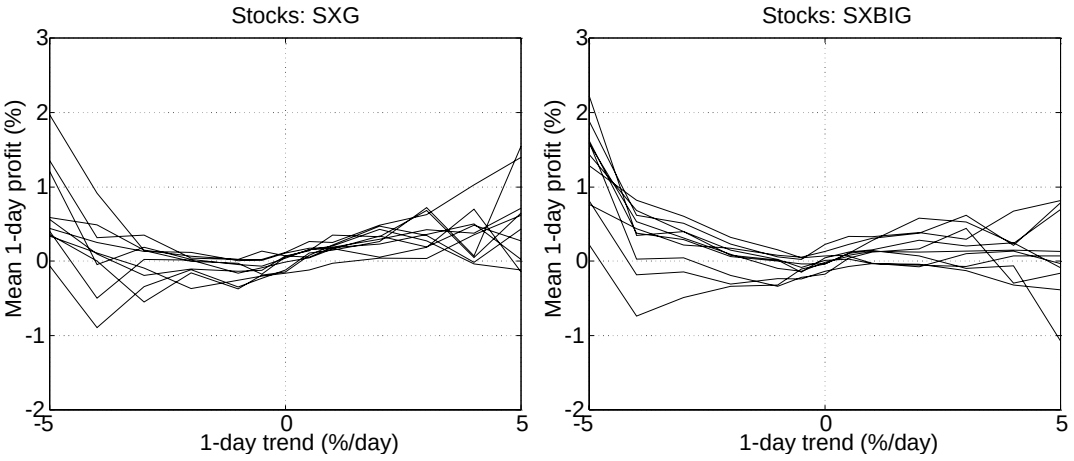

Figure 1: 1-step profits versus 1-step returns for stocks *SXG* and *SXBIG*. Each curve represents one year between 1987 and 1996.

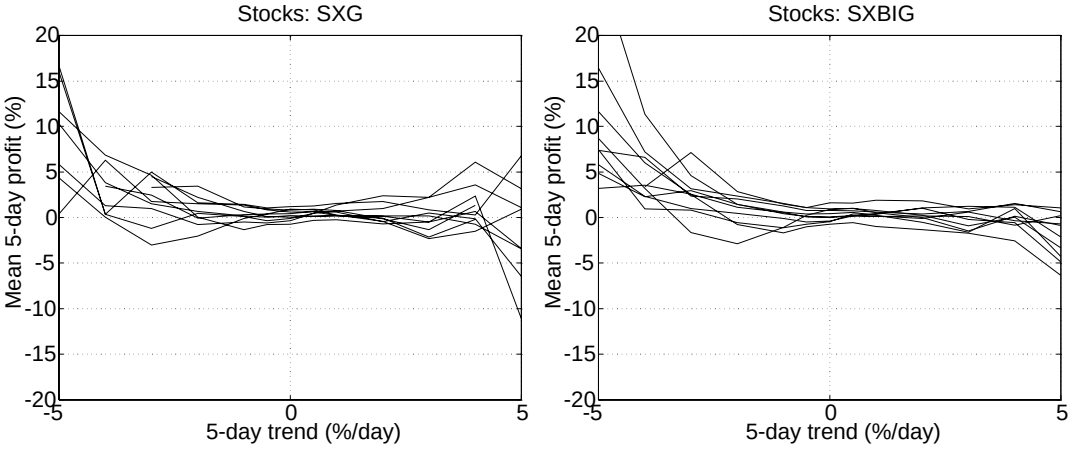

Figure 2: 5-step profits versus 5-step returns for stocks *SXG* and *SXBIG*. Each curve represents one year between 1987 and 1996.

Let us draw some conclusions from these statistical examinations of trends.

- The massive better part of returns falls into a region, where it is very difficult to claim any correlation between past and future price changes. The regions, where any correlation may be significant, are the sparsely populated extreme ones. Looking at Table 3 we observe that a 5% decrease in price since four days ago ($k = 4$), stands a 64.5% probability of showing an increase by tomorrow. However, these cases constitute only a small percentage of the total number of investigated returns.

- The cases that show large increases since yesterday call for a more complex interpretation. Looking at Figure 1, a difference between the two investigated sets of stock becomes apparent. The returns for the 210 stocks in *SXBIG* show a significant *negative* correlation

to both large positive and large negative returns; whereas the returns for the 33 major stocks in *SXG* show a *positive* correlation to large positive returns. This is in accordance with the observed difference in the first lag of the ACF, reported in [1].

4 k-nearest neighbor techniques

The method of k-nearest-neighbors is a general classification technique that makes minimal assumptions on the underlying function to be modeled. To classify a point p one simply finds the set of k closest points from the example set. In the case of time series the input points p_t are typically formed by picking consecutive values from the time series y; $p_t = (y(t-1), y(t-2)..., y(t-d))$. In the general case, the input points p_i can be any type of feature vector that is believed to have predictive power. We have conducted extensive tests with feature vectors consisting of stock trends T_k according to definition 3. For example $p_t = (T_1(t), T_5(t), T_{10}(t), T_{20}(t))$. In stock prediction, the classification $C_k(t)$ for the points is typically then sign of the k-day profit $P_k(t)$:

$$C_k(t) = \left\{ \begin{array}{rl} +1 & \text{if } y(t+k) > y(t) \\ -1 & \text{if } y(t+k) < y(t) \\ 0 & \text{if } y(t+k) = y(t) \end{array} \right\}. \tag{5}$$

The closeness is normally computed as the Euclidean distance in the input space. The mean or median classification of the k nearest points are then taken as estimate of the classification for point p . In this way we can produce classifications immediately, given a set of examples. A lot of variants of the basic algorithm exist. A discussion of weighting schemes can be found in Robinson [6]. For an early work on time series applications and proofs of convergence, see e.g.Yakowitz [7].

Even if the k-nearest-neighbors algorithm is computationally expensive in the application phase, it is very attractive in initial data analysis where questions about predictability and input variable selection are the important issues. It can be argued that failure in applying the k-nearest-neighbors algorithm to a specific problem implies that the problem can not be solved with *any* inductive method. The sole assumption made in the method is that close inputs are mapped to close outputs. It's hard to see how a sufficient amount of data from *any* continuous function should fail such a test. The conclusion would be in such a case that a functional relation between the selected inputs and the output can *not* be shown, given the available data without imposing further restrictions on the functional relationship. However, other methods using *stronger* models, may be more successful then the totally non biased *k-nearest-neighbors* algorithm. One must also realize that k-nearest neighbor in a normal implementation is a *global* method. In the search for nearest neighbors one either scans the entire training set or all previous points in the training set. The latter method avoids peeping into the future when predicting a point classification. In either case, the neighbors are picked from a time period that may very well be too long if the underlying function is non-stationary. Weighting schemes or windowing techniques may be useful in such cases.

4.1 Extensions of the algorithm

A prominent property of stock-price time series is the high level of noise present. Thus there is reason to seriously doubt the possibility of predicting future values, given only past values of the time series. What we can hope for is that the time series *sometimes* is predictable and that in these situations it is possible to create a model that predicts the future better than mere chance. This approach is not acceptable in a general prediction situation, but is perfectly acceptable in the case of stock prediction. The performance of a trading system is normally calculated as the success rate or generated profit in the situations where buy or sell actions are suggested by the algorithm. The overall prediction accuracy is normally of minor interest. This is a natural fact for traders and is implemented in the huge variety of *technical indicators* that issue buy and sell signals when certain conditions are fulfilled. A method where a neural network is combined with a test for statistical dependence in the time series can be found in [5]. We propose below an extension of the *k-nearest-neighbor* algorithm. Given the time series $\{y(t), t = 1, T\}$ the algorithm for prediction of the sign of $P_h(T)$ at time T looks as follows:

1. Generate patterns $\{p(t), t = 1, ..., T - h\}$. In the presented examples the patterns are defined as $p(t) = (T_5(t), T_{20}(t))$. Each pattern $p(t)$ is associated with a target value $P_h(t)$ as defined in 4. Also generate $p(T)$ with an unknown target value $P_h(T)$ to be predicted.

2. Compute the Euclidean distance between $p(T)$ and each of the pattern in $\{p(t), t = 1, T - h\}$. Select the k nearest patterns and denote the set of associated target values Φ. Compute the homogeneity H of Φ as:

$$H = \frac{\max(\|\{x|x \in \Phi, x > 0\}\|, \|\{x|x \in \Phi, x < 0\}\|)}{\|\{x|x \in \Phi, x > 0\}\| + \|\{x|x \in \Phi, x < 0\}\|)}. \quad (6)$$

   The norm $\|.\|$ denotes the number of elements in the argument set. If the majority of elements in Φ is greater than zero, H is the fraction of elements greater than zero. If the majority of elements in Φ is less than zero, H is the fraction of elements less than zero. H is used as a measure of the degree of randomness in the target values Φ. A high value on H is interpreted as a high possibility to predict $P_h(T)$ using a k-nearest-neighbor method.

3. if $H \geq H_{\text{limit}}$ then the selected neighborhood around $p(T)$ is regarded as non-random and the mean value of targets in H is used as predicted target value for $p(T)$. If $H < H_{\text{limit}}$ then the neighborhood around $p(T)$ is regarded as random and a predicted target value for $p(T)$ is not evaluated.

Remark : Note that the search for nearest neighbors in step 2 must not involve any data from the set $t > T$ since this data is highly correlated to the unknown value $P_h(T)$. That is the reason why the nearest neighbors are searched up to $t = T - h$ and no further. Using target values $P_h(t)$ where $t > T - h$ would involve peeping into the future beyond time T which is the point where the prediction is calculated. Furthermore note that the value on H_{limit} affects the number of situations where predictions are produced by the algorithm. The statistical significance of the performance will therefor be reduced if too high a value for H_{limit} is chosen. In our tests, values between 0.6 and 0.9 have been used.

5 Generating Test data

Testing algorithms for stock data predictions is a difficult problem that requires extreme cautions. The risk of interpreting random fluctuations as results of a successful prediction algorithm is always present and sometimes surprisingly high ([2]). Therefore, to properly evaluate the developed algorithm described above we construct an artificial stock time series. The purpose of the algorithm is to identify locations in the input space where a correlation between input patterns and output $P_h(t)$ exists. In all the examples in this report the prediction horizon h has the value 1. Since we use patterns with trend values such as $p(t) = (T_5(t), T_{20}(t))$ as inputs, we introduce locations with correlation in a real stock index time series $y(t)$ by the following algorithm:

1.Compute the time series $T_5(t), T_{20}(t)$ *and* $P_h(t)$ as defined in definition 3 and 4 using real stock data for $t = 1, .., N$.

2.*Repeat for* $t = 1$ *to* N

3. if $T_5(t) > 2$ *and* $T_5(t) < 4$ *and* $T_{20}(t) > 0$ *and* $T_{20}(t) < 5$ then
   (force a 1% increase in stock price after a positive 5-day trend
   and positive 20-day trend :)
   with probability 0.75 set $P_h(t) \leftarrow 1$

4. if $T_5(t) > -4$ *and* $T_5(t) < -2$ *and* $T_{20}(t) > -5$ *and* $T_{20}(t) < 0$ then
   (force a 1% decrease in stock price after a negative 5-day trend
   and negative 20-day trend :)
   with probability 0.75 set $P_h(t) \leftarrow -1$

5.*next* t

In this way the original time series obtains a correlation between the inputs and the output injected in two well defined regions of the input space. It should be mentioned that the introduced correlation is arbitrarily chosen just for illustration of the working of the algorithm. The regions in a real application are automatically detected by the algorithm. The deterministic points are few compared to the total number of generated points. For a 10 year long time series (2500 data points) only 5% are typically generated in each of the two correlated regions in the input space. Such low signal to noise ratios is normally difficult to handle with methods such as global regression analysis.

6 Results

In the two diagrams in figure 3 the relation between the inputs $T_5(t), T_{20}(t)$ and the sign of the output $P_1(t)$ is illustrated. The top diagram shows the original data from the German stock index DAX. The lower diagram has correlation injected as described in the previous section. The input region $\{2 < T_5 < 4\ , 0 < T_{20} < 5\}$ was set to positive output with a probability of 75% and the input region $\{-4 < T_5 < -2\ , -5 < T_{20} < 0\}$ was set

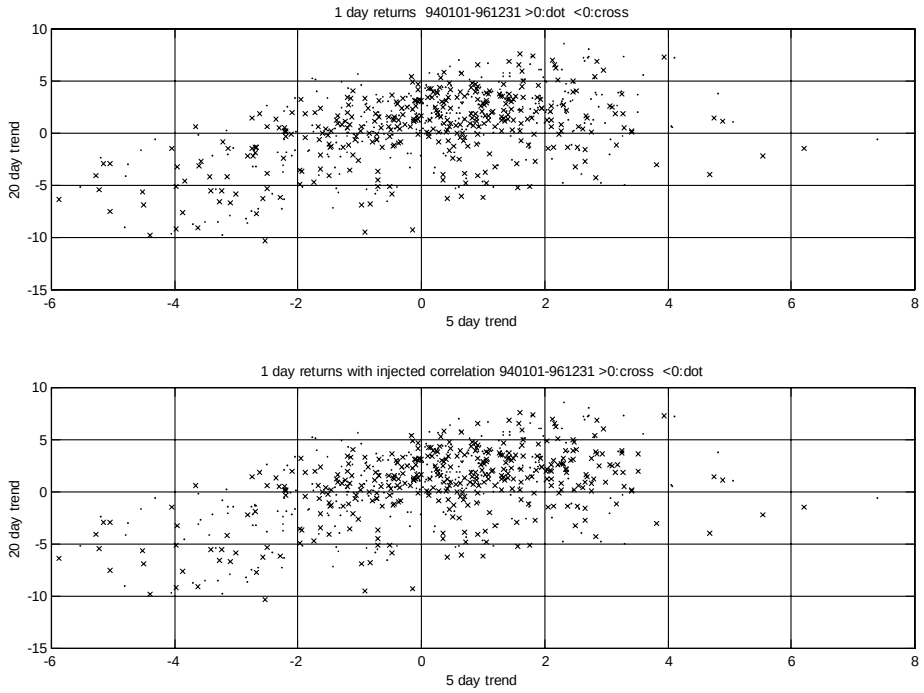

Figure 3: 1-day returns as a function of 5- and 20-day trends for DAX stock index.

to negative output with a probability of 75%. The task of the prediction algorithm will be to identify these regions and assign them to the correct class by the nearest-neighbor principle. As one can see in the bottom diagram, the correlation is not obvious even in this artificial case with extremely high correlation introduced in the data. In the case of the unmodified DAX data the task will be even harder indeed, since no obvious regions with correlation can be seen in the top diagram. Figure 5 shows the situation when all the artificial data points have been run through the extended *k-nearest-neighbor* algorithm. The value of H_{limit} was set to 0.8 and the value of k, number of selected nearest neighbors, was set to 10. The figure denotes points where the homogeneity was greater than 0.8 by crosses and the other points by dots. As can be seen in the diagram, the regions where correlation was injected are clearly identified by the homogeneity measure H. This also shows up in the very high performance presented in table 9. Figure 4 shows a similar diagram for the non-modified stock index DAX. Any clear regions with high homogeneity can be hardly identified.

6.1 Prediction Performance

Evaluation of prediction performance is an important, difficult and often overlooked stage in the development of prediction algorithms for financial data. Since we are looking for very weak correlations, we always run a big risk of interpreting random fluctuations in data as regularities with predictive power. The problem is further discussed for example in [4] and [2]. In table 9 with performance measures for the predictions, we also present values for two bench marks: *Previous-Increase Model* and the ϵ*-increase Model*. The

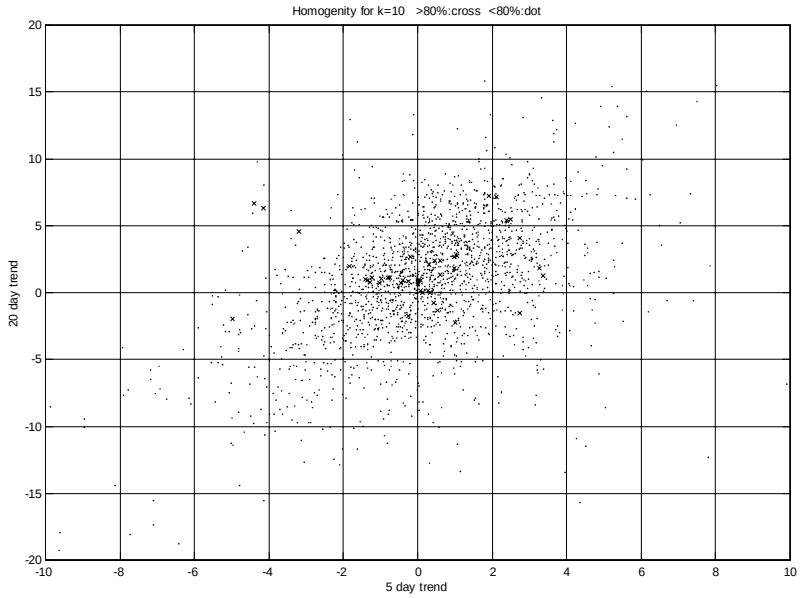

Figure 4: Homogeneity for DAX index. k=10

former is the naive prediction assuming that today's return will be the same as yesterday's. The ϵ-increase Model assumes that today's stock price will increase by a very small amount ϵ since yesterday. The reason why we don't use the usual zero-change model instead of the ϵ is that we want to compare hit rate performance to this bench mark model. Since the hit rate is the ability to predict the sign of the returns, we stipulate that the bench mark predicts an ϵ increase instead of zero. The *ϵ-increase Model* is also used for calculation of the Theil coefficient which is the quotient between the RMSE for the *Modified k-nearest-neighbor algorithm* and the *ϵ-increase Model.* The investigated time period ranges from the beginning of 1987 until the end of 1996. The limit of selection of points, H_{limit} , is set to 0.80 for all examples in this report. The parameter k , the number of selected nearest neighbors, is set to 10. The DAX column presents results of the real **DAX** index, whereas **Correlated DAX** presents results of the time series with injected correlation. As we can see from the first three lines, the performance for **Correlated DAX** is excellent. The hit rate 75.26% clearly outperforms both of the bench marks! The Theil coefficient is also below 1, indicating a true predictive power beyond that of the *ϵ-increase Model*. The algorithm produces 197 predictions, i.e. 10% of the total number. This is approximately the number of points affected by the injection of correlation described in section 5. The hit rate 75.26% also conforms nicely to the 75% randomness in the algorithm. The results for the real **DAX** index data appear to indicate predictability even if the performance is clearly much lower than in the previous column with artificial data. However, one must bear in mind that the statistics are based on 45 selected data points only. The risk of data snooping is huge. The last three columns present prediction results for three other international stock indexes; the Swedish Generalindex, The American Dow Jones and the English FTSE 100. The results are somewhat contradicting and a statistically significant conclusion can hardly be made based on these tests only.

Table 8: Number of points (SXG)

	k-day trend (%/day)												
k	**-5.00**	**-4.00**	**-3.00**	**-2.00**	**-1.00**	**-0.50**	**0.00**	**0.50**	**1.00**	**2.00**	**3.00**	**4.00**	**5.00**
1	1366	1160	2531	6337	8584	7410	12257	7179	8381	6361	3003	1490	1786
2	482	553	1516	4595	9204	10641	12617	10057	9215	5341	1994	748	791
3	235	335	949	3472	8679	11646	15173	11263	9142	4517	1395	468	457
4	112	206	700	2672	8113	12337	17016	11976	9159	3816	933	316	309
5	71	129	540	2135	7363	13042	18246	12717	9087	3180	704	244	207
10	3	30	160	924	4686	13447	23564	15497	7132	1535	283	90	80
20	0	0	8	349	2485	12132	29546	17183	4427	732	123	17	27
30	0	0	0	70	1787	10646	33412	17076	3144	467	48	17	32
50	0	0	0	0	735	8605	38753	15722	1950	214	43	24	42
100	0	0	0	0	7	5589	45585	12088	848	259	40	5	48

Table 9: Prediction performance for the period 1987-1996

Modified k-nearest neighbor	Corr. DAX	DAX	Generalindex	Dow Jones	FTSE
Hit rate relative Previous-Increase	1.40	1.14	0.92	1.02	1.02
Hit rate relative ϵ-increase	1.42	1.10	0.98	0.93	0.96
Theil coefficient	0.93	0.78	1.00	0.86	1.13
RMSE	1.03	0.86	1.15	0.96	0.91
Hit rate (%)	75.26	57.14	52.85	50.46	50.68
Number of points	197	45	125	123	74
Mean($\|predictions\|$)	0.79	0.52	0.58	0.54	0.49
Mean homogenity H	66.03	62.26	65.82	63.46	64.11
Mean($\|returns\|$)	0.79	0.76	0.77	0.64	0.61
Previous-Increase Model					
RMSE	1.54	1.58	1.49	1.64	1.10
Hit rate (%)	53.80	49.95	57.72	49.63	49.86
Number of points	1930	1930	2317	2218	1848
ϵ-increase Model					
RMSE	1.10	1.10	1.15	1.12	0.81
Hit rate (%)	53.05	52.16	54.08	54.25	52.68
Number of points	1930	1930	2317	2218	1848
Standard k-nearest neighbor					
Hit rate relative Previous-Increase	1.04	1.01	0.94	1.02	1.02
Hit rate relative ϵ-increase	1.05	0.97	1.00	0.93	0.96
Theil coefficient	1.02	1.05	1.04	1.08	1.06
RMSE	1.13	1.15	1.19	1.21	0.85
Hit rate (%)	55.77	50.34	54.33	50.71	50.63
Number of points	1929	1929	2316	2217	1847

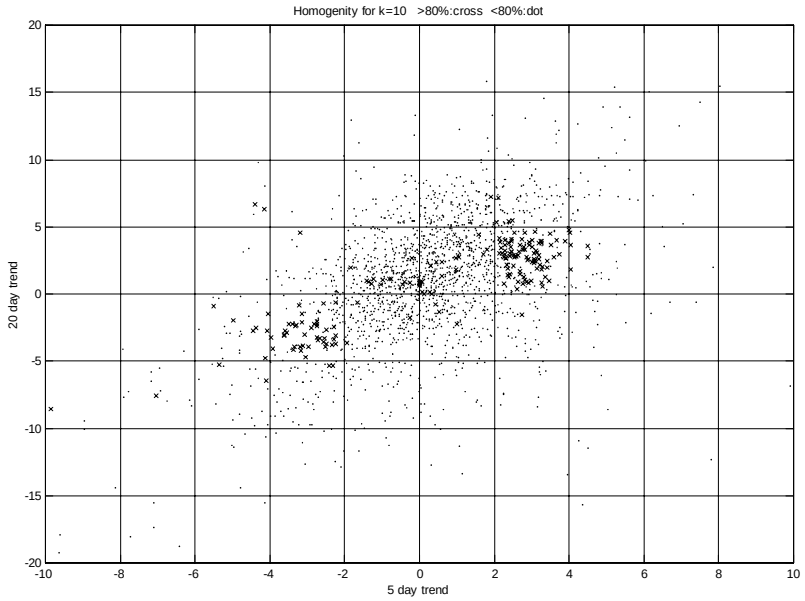

Figure 5: Homogeneity for correlated DAX. k=10

An important parameter to alter in the *k-nearest-neighbor* algorithm is the value on k. The table 9 shows results of $k = 10$. A thorough test with varying values on k has been conducted for the DAX and for the other reported stock indexes. The relative hit rate varies but cannot be shown to significantly exceed 1 for any value on k.

7 Conclusions and further development

The developed algorithm works well on the synthetic data with correlation injected and also finds predictable patterns in the real stock indexes. However, the results are somewhat weak and further tests have to be conducted in order to obtain statistically significant results. For future work we will investigate higher dimensional patterns of trend variables, such as $p(t) = (T_1(t), T_2(t), T_5(t), T_{20}(t))$, and also combine them with normalized volume values to test the hypothesis (see e.g. [3]) that patterns in traded volume bear relevance to predictions of future returns. The algorithm can also be further developed with more sophisticated decision criteria for predictability. The size of the neighborhood could be determined in a more intelligent way than the fixed value of k neighbors used by the basic k-nearest-neighbor algorithm. Since the input space with trend patterns such as $p(t) = (T_5(t), T_{20}(t))$ is not populated in a homogenious fashion, the k nearest neighbors may be picked in some areas of the input space from a very small volume whereas in other areas, the k nearest neighbors have to be picked from a large volume, where the input space is sparsely populated. The suggested algorithm has natural application in areas other than stock predictions. It addresses all situations where predictability can only be expected to apply in small and indeterminate regions of the input space.

References

[1]T. Hellström and K. Holmström. Predictable patterns in stock returns. Technical Report IMa-TOM-1997-9, Department of Mathematics and Physics, Mälardalen University, Sweden, 1997.

[2]T. Hellström and K. Holmström. Predicting the stock market. Technical Report IMa-TOM-1997-7, Department of Mathematics and Physics, Mälardalen University, Sweden, 1997.

[3]B. LeBaron. Persistence of the dow jones index on rising volume. Technical report, Department of Economics, University of Wisconsin - Madison, Madison, WI, July 1992.

[4]A. W. Lo. Data snooping and other selection biases in financial econometrics. In *Tutorials NNCM-96 International Conference, Neural Networks in the Capital Market Pasadena.*, 1996.

[5]D. Ormoneit and R. Neuneier. Reliable neural network predictions in the presence of outliers and Non-Constant variences. In A.-P. Refenes, Y. Abu-Mostafa, J. Moody, and A. Welgend, editors, *Neural Networks in Financial Engineering, Proc. of the 3rd Int. Conf. on Neural Networks in the Capital Markets*, Progress in Neural Processing, pages 578–587, Singapore, 1996. World Scientific.

[6]P. M. Robinsson. Asymptotically efficient estimation in the precence of heteroskedacticity of unknown form. *Econometrica*, 55:875–891, 1987.

[7]S. Yakowitz. Nearest-Neighbour methods for time series analysis. *Journal of Time Series Analysis*, 8(2):235–247, 1987.