Theory of Stochastic Processes
Vol.6 (22), no.3-4, 2000, pp. 47-63

THOMAS HELLSTRÖM

OUTLIER REMOVAL FOR PREDICTION OF COVARIANCE MATRICES

with an Application to Portfolio Optimization[1]

We apply a simple algorithm for systematic removal of outliers in the computation of covariance matrices for portfolio optimization. The algorithm computes a penalty measure for each day depending on the increase or decrease in prediction error the inclusion of the day has given for the previous predictions. The penalty measure is updated at each prediction step by additional predictions where each day in the modelling window is removed in sequence. The algorithm gives a significant reduction in RMSE for the covariance matrices when tested on data with deliberately planted outliers and also for real stock data. The covariance matrices are also fed into a portfolio optimizer that maximizes risk-adjusted return. The algorithm gives higher risk-adjusted return than the naive prediction even if the reduction of RMSE for the covariances is higher than the gain in portfolio optimization. This indicates that outliers in stock data do not affect the computation of optimal portfolios to any significant degree, at least not on the daily scale that is used in the presented tests. However, the algorithm shows a clear ability to detect and eliminate the effect of outliers and could be applied to other time series related modelling problems with outliers in data.

2000 *AMS Mathematics Subject Classifications.* 62M10, 91B28, 62P05, 62G35.

Key words and phrases: correlation, outlier, portfolio optimization, prediction, cross-validation

1. INTRODUCTION

Modern portfolio theory involves solving a quadratic optimization problem where the return vector and covariance matrix are supposed to be known. The normal "naive" method of using historical data to compute sample covariances

[1]The invited paper

and returns results in sub-optimal portfolios since the price generating process is time varying and also since the sample covariance is known to be sensitive to outliers in data. A lot of alternative methods for estimation and prediction of covariance matrices have been suggested in the literature. For a survey and examples of common techniques see for example Ma, Genton (1999) or Ledoit (1999).

In this paper a simple algorithm to improve the naive prediction with systematic removals of outliers in data is presented. Outlier detection is a well established field in statistics. For a thorough introduction refer to Barnett, Lewis (1994). Our method is a *leave-one-out* approach which has been previously (Pena and Yohai (1994)) used for outlier detection in regression problems. However, our algorithm performs repetitive updates of a contamination factor in a way that to the authors knowledge is novel. Also the application to portfolio optimization is new. Section 2 gives an introduction to the applicable parts of modern portfolio theory. Section 3 investigates the naive prediction empirically and determines a bench mark for the evaluation of the new algorithm which is described in Section 4. The empirical test results are presented in Section 5 and Section 6 concludes the report with a summary of results and conclusions.

2. PORTFOLIO THEORY

We are looking at the problem of composing a portfolio out of a set of stocks $s_1, ..., s_n$. The portion of stock s_i is given by the weight w_i such that $\sum_{i=1}^{n} w_i = 1$. The column vector w is defined as $w = (w_1, ..., w_n)^T$. The classical approach to analyze such a portfolio was formulated in Markowitz (1952) and recognizes the relation between the return and the variance of the portfolio. A smaller variance can be achieved by utilizing the correlations between individual stocks in the portfolio. This reduction is normally paid off by a smaller return for the portfolio. The portfolio return R_p is the weighted sum of the individual stock returns $r_1, ..., r_n$ with the expected values $\mu_1, ..., \mu_n$:

$$R_p = \sum_{i=1}^{n} w_i r_i. \tag{115}$$

The expected value for the portfolio return R_p is given by

$$ER_p = \sum_{i=1}^{n} w_i Er_i = \sum_{i=1}^{n} w_i \mu_i.$$

Using matrix notation where R is the column vector $(\mu_1, ..., \mu_n)^T$, we get the following expression for the expected value of the portfolio return R_p :

$$ER_p = w^T R. \tag{116}$$

The variance σ^2 quantifies the risk of the portfolio and is given by the expression

$$\sigma^2 = E\left(R_p - ER_p\right)^2 = E\left(\sum_{i=1}^{n} w_i r_i - \sum_{i=1}^{n} w_i \mu_i\right)^2 \tag{117}$$

which can be further expanded to

$$\sigma^2 = \sum_{i=1}^{n} w_i^2 \sigma_i^2 + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_i w_j \sigma_{ij} \tag{118}$$

where σ_i is the standard deviation for stock i and σ_{ij} is the covariance between stock i and stock j. Using matrix notation where C is the covariance matrix ($C(i,j) = \sigma_{ij}$), we finally arrive at the following expression for the variance σ^2 of the portfolio:

$$\sigma^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \sigma_{ij} = w^T C w. \tag{119}$$

σ^2 will in the following be denoted σ_p^2.

2.1 Portfolio Optimization

Given ER_p and σ_p^2 the following optimization problem can be formulated. The risk tolerance factor α expresses the relative importance of expected portfolio return and variance and is often set to 0.5.

$$\begin{array}{l} \max\limits_{w} \alpha w^T R - w^T C w \\ s.t. \\ w_i \geq 0, i = 1, ..., n \\ \sum_{i=1}^{n} w_i = 1 \end{array} \tag{120}$$

The optimization task is to find the weight vector w so the risk adjusted return, defined as $\alpha w^T R - w^T C w$, is at its maximum for the portfolio. This standard problem of Modern Portfolio Theory (MPT) can be easily solved using off-the-shelf routines for quadratic programming. The optimization requires the vector with expected stock returns R and the covariance matrix C to be estimated. The typical procedure is to compute the sample covariances and returns using historical data. The estimated values for R and C are then plugged into the optimization routine, which computes the weights $(w_1, ..., w_n)$ that maximize $\alpha w^T R - w^T C w$. It is important to realize that both the inputs (R and C) and the outputs (w) are estimated using the same historical time period. The problem solved is therefore an artificial one: "*Find the portfolio weights w that would have given the highest risk adjusted return $\alpha w^T R - w^T C w$*". The computed weights are however normally used to compose a real stock portfolio with the objective to achieve a future behavior similar to the one assumed in the optimization. The success or failure of this assumption depends on the stationarity properties of the covariance matrix C and the returns R.

3. THE NAIVE PREDICTION

The method of using historical data to compute estimates of covariance matrices and return vectors for the portfolio optimization is here called the *naive prediction*. The naive prediction is surprisingly hard to beat and will serve as a bench mark in the evaluation of the new algorithm. Predicting the returns for a set of stocks is a well studied subject where the naive prediction has an even stronger position than in the case with covariance matrices. Indeed there are few indications that it is possible to formulate a general model for prediction of future returns more accurately than using the historical mean. The naive prediction of the covariance matrix C and the return vector R uses the sampled data up to time $T-1$ to compute a prediction of $C(t)$ and $R(t)$ for $t \geq T$. The method obviously relies on a stationarity assumption regarding C and R. In this section we present some empirical studies of how this naive prediction behaves and depends on the amount of historical data that is used to compute the prediction.

Data in a window of size N days measured from time $T-N$ to $T-1$ is used to compute the sample covariance matrix and the mean returns in a window of size 20 trading days (1 month) measured from time T to $T+19$. This prediction is then used as input in the portfolio optimizer that computes an optimal portfolio given the estimate of the covariance matrix and the return vector, as described in Section 2.1. The outcome of this prediction is a number of performance measures; direct RMSE (the root of the mean squared error) prediction errors (for covariances and returns) and portfolio measures (return, variance and the risk adjusted return). The predictions are repeated with sliding windows of width 20, thus producing a number of performance measures for each value on N. The mean values of these samples are plotted versus N in Figures 1 and 2. Figure 1 shows the results for 24 major stocks from the Swedish stock market between 1988 and 1997 while Figure 2 shows the results for 29 stocks from the Dow Jones index during the same 10 years.

The presented results are the following: The mean RMSE for all returns $r_1, ..., r_n$ and also for all elements in the covariance matrix C are shown in the first two diagrams. The results for the computed optimal portfolio are shown in the following three diagrams; the portfolio return R_p, the variance σ_p^2 and finally the risk adjusted return $\alpha w^T R - w^T C w$. The risk tolerance factor α is set to 0.5 in all presented results in this report. From the graphs we can conclude that the naive prediction gives higher risk adjusted return the more data we incorporate. However, more than 200 trading days back does not produce any noticeable increase in performance. The naive prediction that will be used as benchmark throughout this report will therefore be based on 200 trading days.

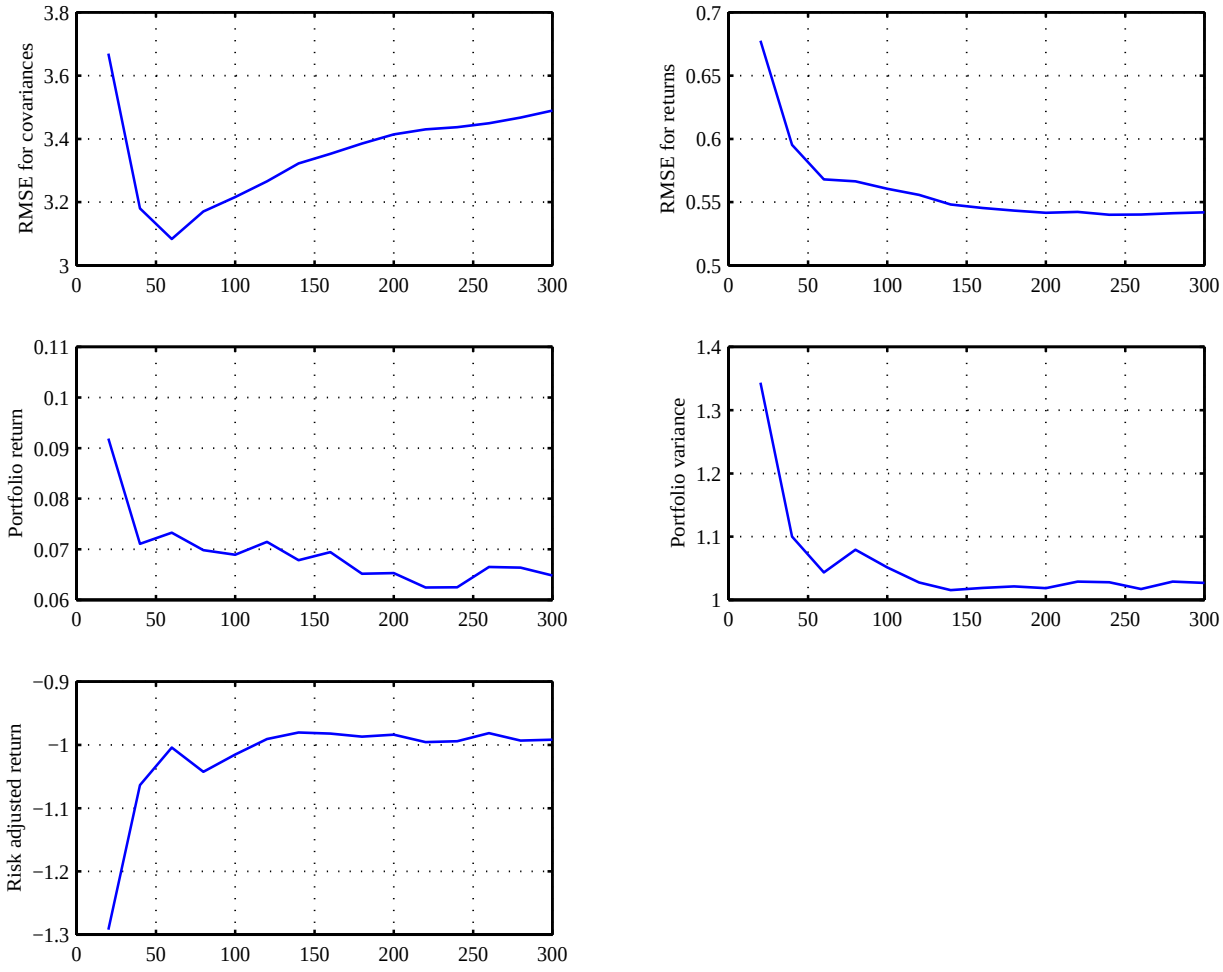

Figure 1: Performance versus N for naive predictions of the covariance matrix for portfolio optimization. The predictions use the N previous days to form estimates of the returns and the covariances. Data from 24 major stocks on the Swedish stock market 1988-1997. The value to be predicted is the sample covariance of daily returns 20 trading days ahead.

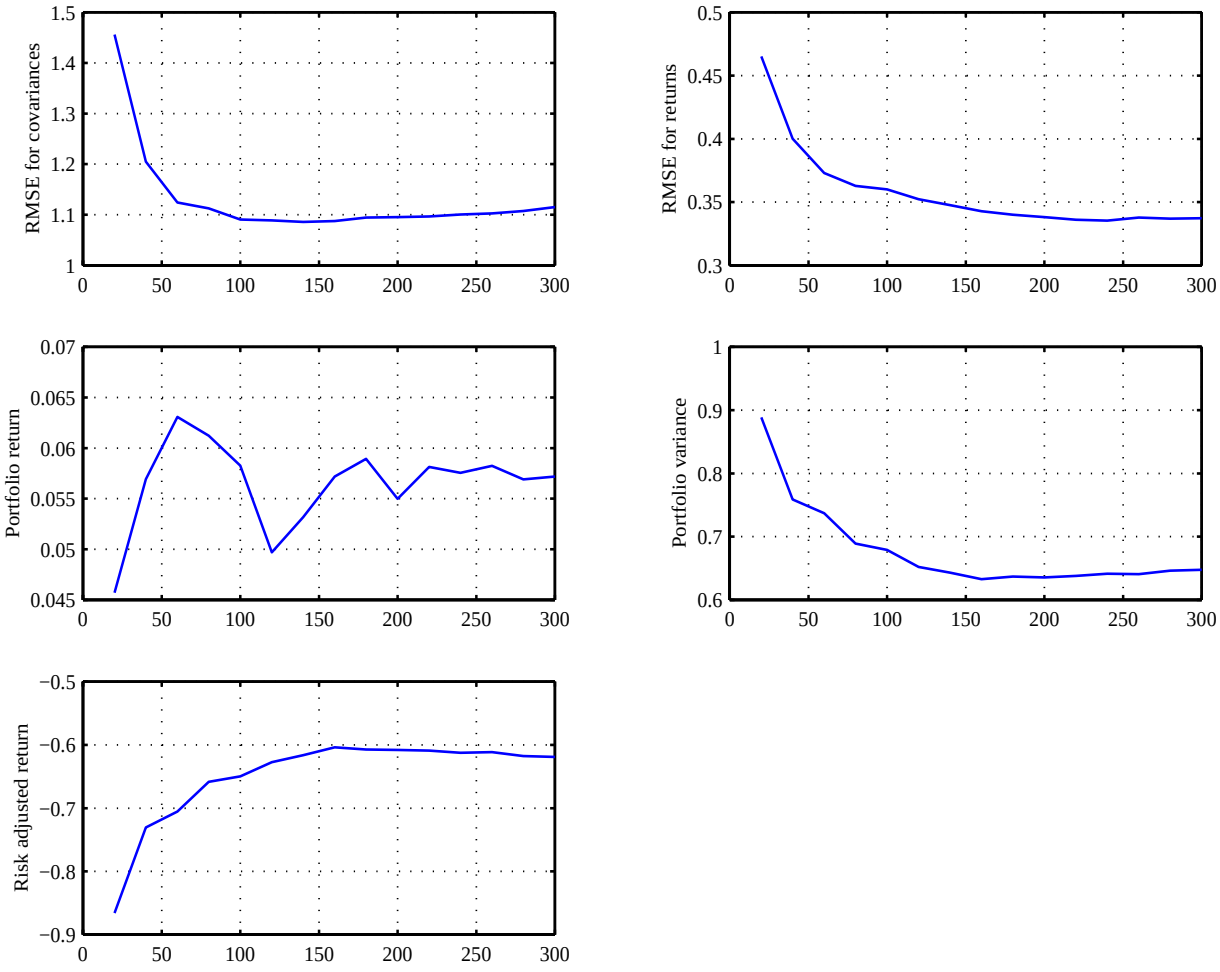

Figure 2: Performance versus N for naive predictions of the covariance matrix for portfolio optimization. The predictions use the N previous days to form estimates of the returns and the covariances. Data from 29 stocks in the Dow Jones index from 1988-1997. The value to be predicted is the sample covariance of daily returns 20 trading days ahead.

4. AN ALGORITHM FOR OUTLIER REMOVAL

Our approach in the present work is that of starting with the naive prediction and trying to improve it by removing outliers. Outliers are detected by updating a *contamination factor* K_i for each day i. This factor is a relative measure of the prediction improvement achieved by removing the particular day from the computation of the sample covariances. The algorithm relies on the assumption that the "badness" of a particular day generalizes into the future. The method is related to *leave-one-out cross-validation* commonly used in classification problems (e.g. Mosteller, Tukey (1968) or Stone (1978)) to estimate the miss-classification risk.

The following *Remove* algorithm is used to produce a sequence of monthly predictions $\hat{C}_0(t)$ of the covariance matrix one month ahead. Historical data for k stocks are assumed to be available for the time period $d_1, ..., d_2$. The prediction horizon is set to v days and w days backwards are used as a sliding window in which the predictions are formed. A prediction is performed every v days starting at time $d_1 + w$. In the examples in this report, v is set to 20 days and w is set to 200 days.

```
1)   K_i ← 0, i = d_1,...,d_2
2)   for t = d_1 + w step v to d_2 − v
3)        r ← {j|K_j > K_limit, j ∈ {t − w, ..., t − 1}}
4)        Ĉ(t) ← cov(t − w, t − 1, r)
5)        C(t) ← cov(t, t + v − 1)
6)        Ĉ_0(t) ← cov(t − w, t − 1)
7)        e_0 ← ‖C(t) − Ĉ_0(t)‖_2^2                                  (121)
8)        for j = t − w to t − 1
9)             Ĉ_j(t) ← cov(t − w, t − 1, j)
10)            e_j ← ‖C(t) − Ĉ_j(t)‖_2^2
11)            K_j ← K_j + ((e_0 − e_j)/e_0) 100
12        next j
13)  next t
```

The function $cov(t_1, t_2)$ computes the sample covariance matrix using data between times t_1 and t_2. If a third argument d to the function cov is given, the days d are removed from the computation of the covariance matrix.
The *Remove* algorithm works as follows:

1. The contamination factor K_i for each day is initialized to zero.
2. The predictions are done every v days.
3. Select those days within the modeling window with a contamination factor $K_j > K_{limit}$.
4. Compute a new prediction $\hat{C}(t)$ where all these points have been removed.
5. The "correct answer" is chosen as the sample covariance v days ahead[1].

[1]The sample covariance is of course not equal to the true covariance but it is a relevant entity since we are aiming at beating a prediction of the sample covariance.

6. The naive predictor $\hat{C}_0(t)$ is computed as the sample covariance w days back.

7. The prediction error e_0 is defined as the squared 2-norm of the difference between the prediction and the correct answer.

8. w additional predictions are made to update the contamination factors.

9. $\hat{C}_j(t)$ is the sample covariance with data between $t-w$ and $t-1$ with day j removed.

10. The errors for each j is defined as the squared 2-norm of the difference between the prediction and the correct answer.

11. Update the contamination factor K_j for the prediction $\hat{C}_j(t)$.

12. Repeat w times, thus removing every day in the modeling window in sequence.

13. Repeat for all days, with a v days step.

For one particular day j, the contamination factor K_j will be updated at w/v predictions. Some of these predictions will indicate that the prediction gains from the removal of day j, while other predictions will indicate the opposite. The net result from the updates performed up to time t for a new prediction will decide if a day should be included or not in the computation of the estimate $\hat{C}(t)$.

4.1 What About the Returns?

The returns vector $R = (\mu_1, ..., \mu_n)$ also has to be estimated in order to compute an optimal portfolio as described in (120). Attempts have been made to include the prediction error for returns in the error terms e_0 and e_j in algorithm 121 above. The empirical results get worse by this modification. The present version of the algorithm therefor predicts the returns in the same way as the covariances. I.e.: step 4 in the algorithm is expanded by the computation

$$\hat{R}(t) \leftarrow ret(t-w, t-1, r) \tag{122}$$

where ret is assumed to compute the return vector using data between times t_1 and t_2. If a third argument r to the function ret is given, the days r are removed from the computation of the return vector.

5 EMPIRICAL RESULTS

The algorithm has been tested on Swedish and American stock data from 1988 to 1997. The Swedish data consists of 24 major stocks while the American data consists of 29 stocks from the Dow Jones index. The prediction horizon v is 20 days and the modelling window w is 200 days. Figure 3,4 and 5 show the results for the Swedish data. The cut off value K_{limit} for the contamination factor is set to 3. The method's sensitivity to this value has not been examined in this report but should be included in future research. Furthermore, optimal K_{limit} could be computed automatically with a cross-validation technique similar to the one used in the *Remove* algorithm. Figure 3 shows the contamination factor K_i for each day i. A high value for a day means that the day has been detected as outlier by the algorithm. Figure 4 shows the number of days that get removed by the algorithm for every monthly prediction. This number varies between 0 and 28 days with an average number of 12, both for the Swedish and American data. The final prediction performance is shown in Figure 5. The error measure is the RMSE for all distinct elements C_{ij} in the $n \times n$ covariance matrix, averaged over all predictions:

$$RMSE = \operatorname*{ave}_t \sqrt{\frac{2}{n^2+n} \sum_{i=1}^{n} \sum_{j=i}^{n} \left(\hat{C}_{ij}(t) - C_{ij}(t) \right)^2}. \tag{123}$$

The improvement by using the algorithm instead of the naive prediction is 8.9% for the Swedish stock data. The results for the American stock data is shown in Figures 6,7 and 8. The improvement in this case is lower. The *Remove* algorithm reduces the RMSE by in average 3.5%.

5.1 Using the Predictions in Portfolio Optimization

The computed predictions are also input in the portfolio optimization problem defined in Section 2.1. In this way the risk adjusted return can be used as performance measure instead of the mean squared error for the covariance predictions. The portfolio is rebalanced once every 20 trading days (i.e. one calendar month) using predictions for the covariance matrix C and the return vector R. The two prediction methods *Remove 3%* (denoted $\hat{C}$ in the algorithm) and *Naive* (denoted $\hat{C}_0$ in the algorithm) are compared in Tables 124 and 125. The columns show the return R_p, the variance σ_p^2 and the risk adjusted return $R_{ADJ} = 0.5R_p - \sigma_p^2$ for the portfolio. The last column gives the fraction of cases where the method has been at least as good as the other method with respect to R_{ADJ}. As previously mentioned the *Naive* method uses all previous 200 days to compute the covariance matrix and the return vector. For comparison, the performance for an *Equally balanced* portfolio is also presented in the tables.

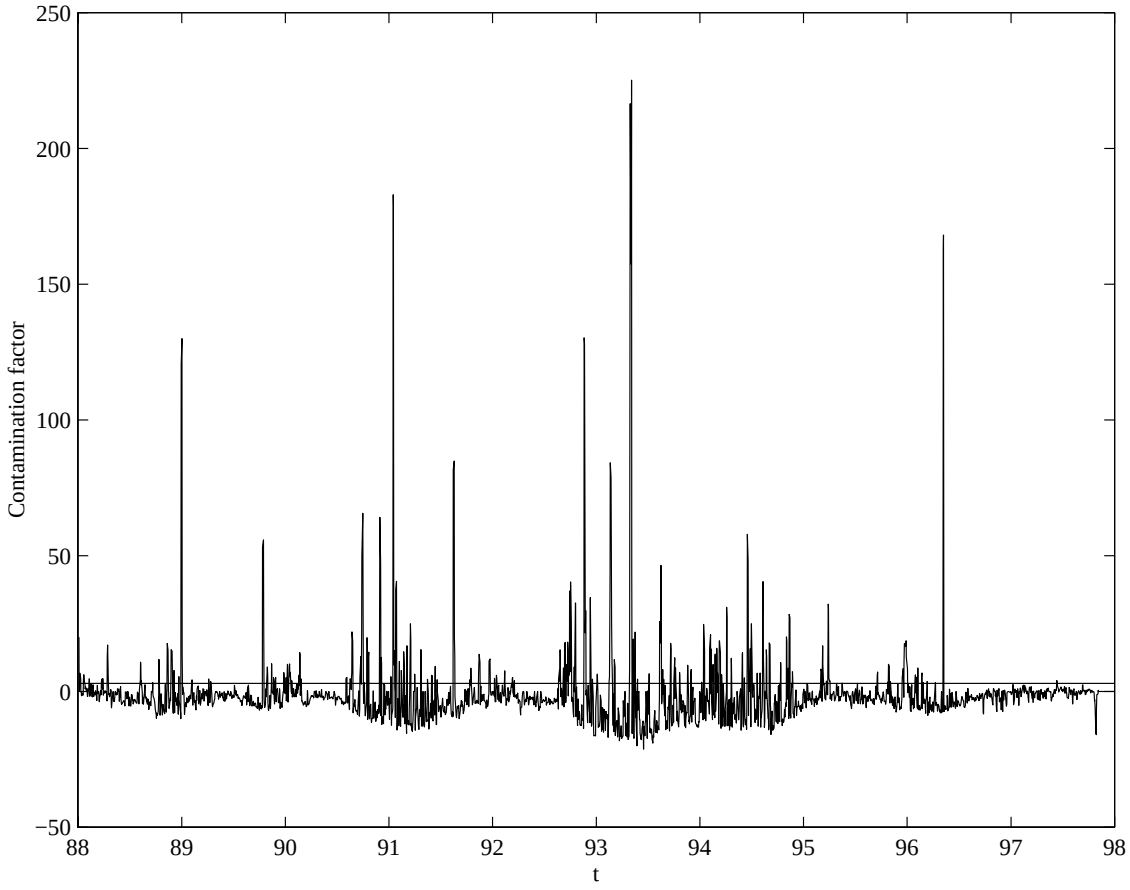

Figure 3: The diagram shows the contamination factor K_t as a function of time t. A high value for a day means that a lot is gained by removing it before computing the sample covariance matrix. The cut off level K_{limit} is in the example set to 3%. Days above this level will be removed. Data from 24 Swedish stocks.

5.1.1 Results for the Swedish Stock Market

The performance for monthly predictions on the Swedish data (24 stocks from 1988-1997) is presented in Table 124.

Method	R_p	σ_p^2	R_{ADJ}	Best
$\hat{C}_j$: Remove 3%	0.067	0.995	-0.959	62%
$\hat{C}_0$: Naive 200 days	0.065	1.019	-0.984	47%
Improvement	3.2%	2.4%	2.5%	
Equally balanced	0.072	1.465	-1.428	

(124)

From the R_{ADJ} column, we can conclude that *Remove* is 2.5% better than *Naive*. It is surprisingly low considering that the reduction in RMSE for the pure covariance predictions is 8.9%. However, the *Remove* method is the best choice in about 62% of the predictions while the *Naive* is the best only 47% of the times.

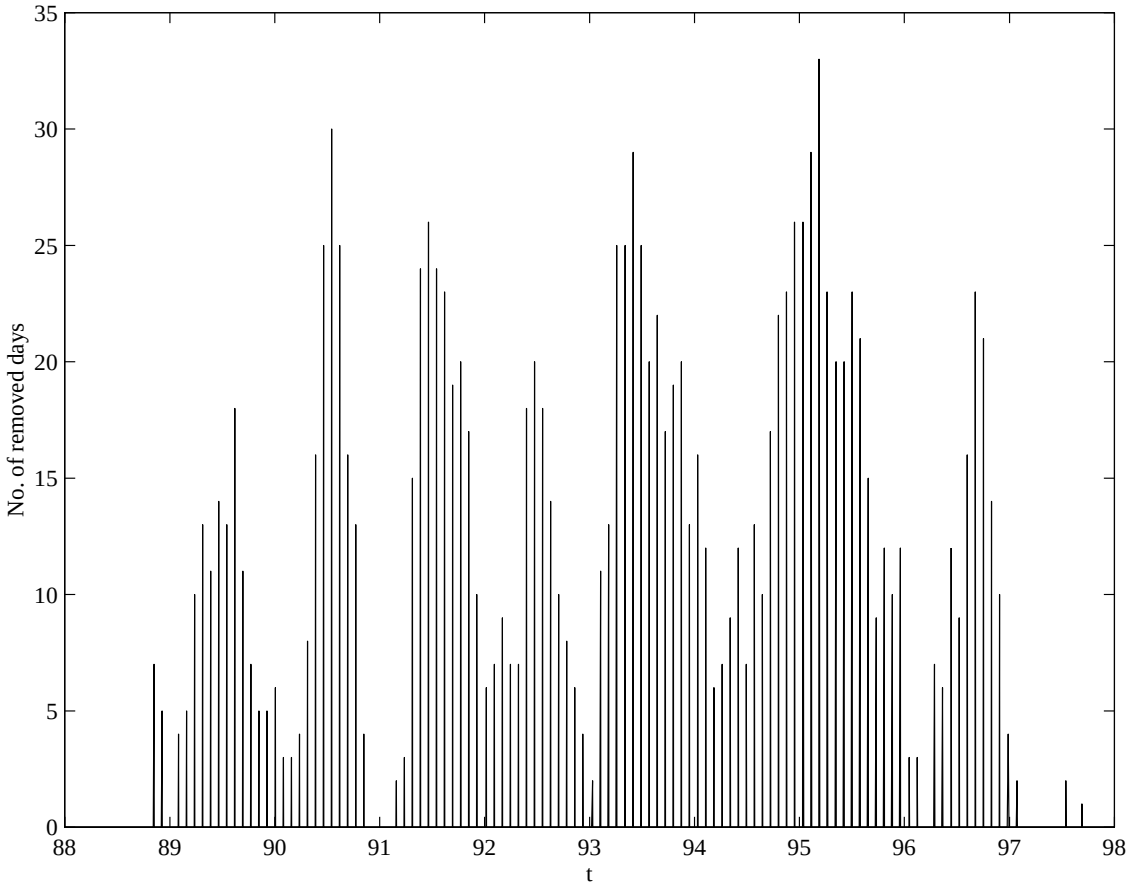

Figure 4: Number of removed days for each monthly prediction $\hat{C}(t)$. A day i is removed if $K_i > 3\%$. The average number of removed days is 11.9. Data from 24 Swedish stocks. The modelling window is 200 days long.

We can also see that both methods, as expected, are better than the *Equally balanced* portfolio.

5.1.2 Results for the American Stock Market

The same performance analysis for monthly predictions on American data (29 stocks from 1988-1997) is presented in Table 125.

Method	R_p	σ_p^2	R_{ADJ}	Best
$\hat{C}_j$: Remove 3%	0.061	0.627	-0.597	54%
$\hat{C}_0$: Naive 200 days	0.055	0.636	-0.608	51%
Improvement	10.5%	1.4%	1.9%	
Equally balanced	0.065	0.727	-0.694	

(125)

The *Remove* method has 1.9% higher R_{ADJ} than *Naive* and also 10.5% higher portfolio return. The *Equally balanced* portfolio exhibits an even higher portfolio return which seems like bad news for the *Remove* algorithm. However, at each

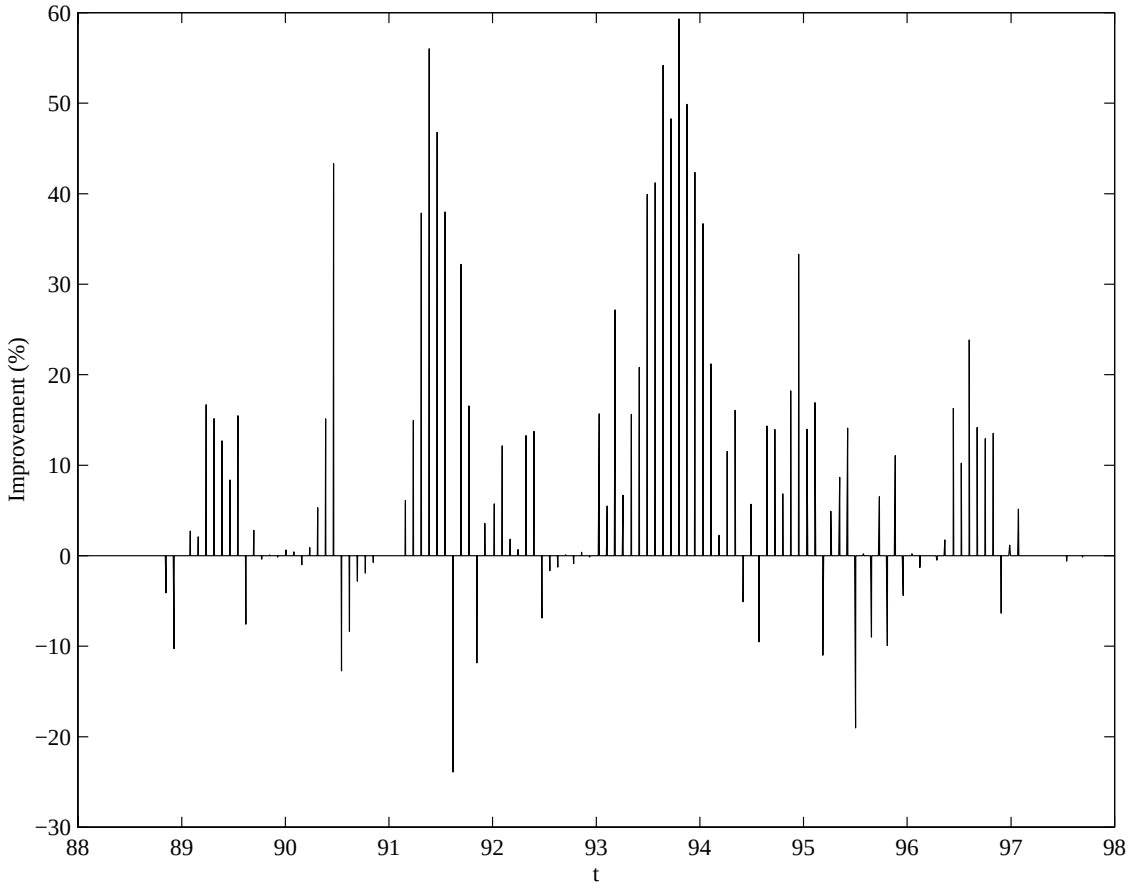

Figure 5: The improvement (reduction of RMSE for the covariance predictions) for each monthly prediction $\hat{C}(t)$ relative the naive prediction $\hat{C}_0(t)$ is plotted versus time. The average improvement is 8.9%. Data from 24 Swedish stocks.

prediction step, portfolio weights are computed to maximize the risk adjusted return $R_{ADJ} = 0.5R_p - \sigma_p^2$. Since the portfolio variance σ_p^2 is one order of magnitude larger than the portfolio return R_p it should not come as a surprise that the generated portfolios have sub-optimal or even random returns. The relevant performance measure is of course the optimization entity, i.e. R_{ADJ}.

5.2 Simulating Outliers

The developed algorithm is now tested with artificial data. The same data sets as before are used but this time with noise injected in a systematic fashion. Data for every 50th trading day is exposed to a shock such that the true stock prices get multiplied by $(1 + rnd * NoiseLevel/100)$ where rnd is a normally distributed random sample $\in N(0, 1)$ and $NoiseLevel$ is a pre defined constant. Even if the economic motivation for this way of introducing noise might be questionable, it serves the purpose of demonstrating the algorithm's ability to detect outliers in

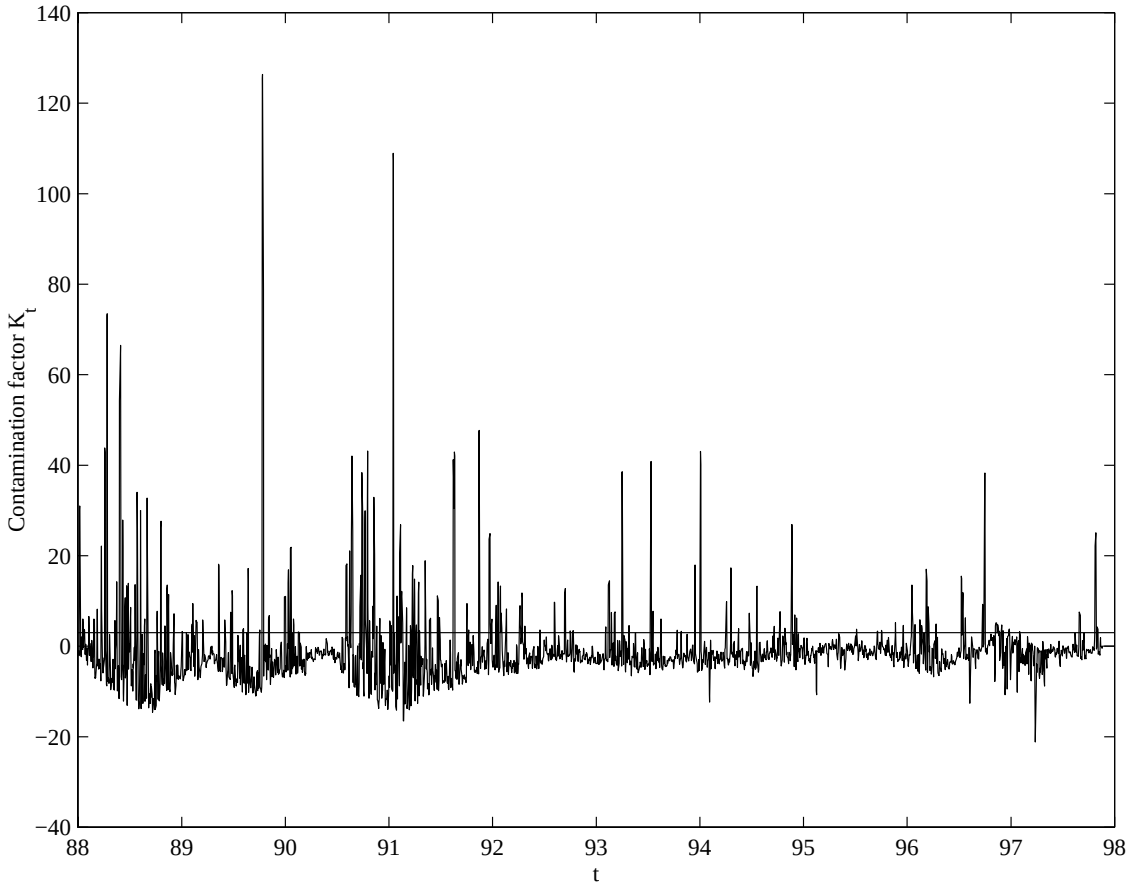

Figure 6: Contamination factor K_t as a function of time t. A high value for a day means that a lot is gained by removing it before computing the sample covariance matrix. The cut off level K_{limit} is in the example set to 3%. Days above this level will be removed. Data from 29 American stocks.

the data. The results are shown in Figures 9 and 10 and illustrate that the 50-day interval gets clearly detected and that the distorted days will be removed before the covariance matrices are estimated (note that one calendar year comprises approximately 250 trading days). *NoiseLevel* is in the shown examples set to 5.

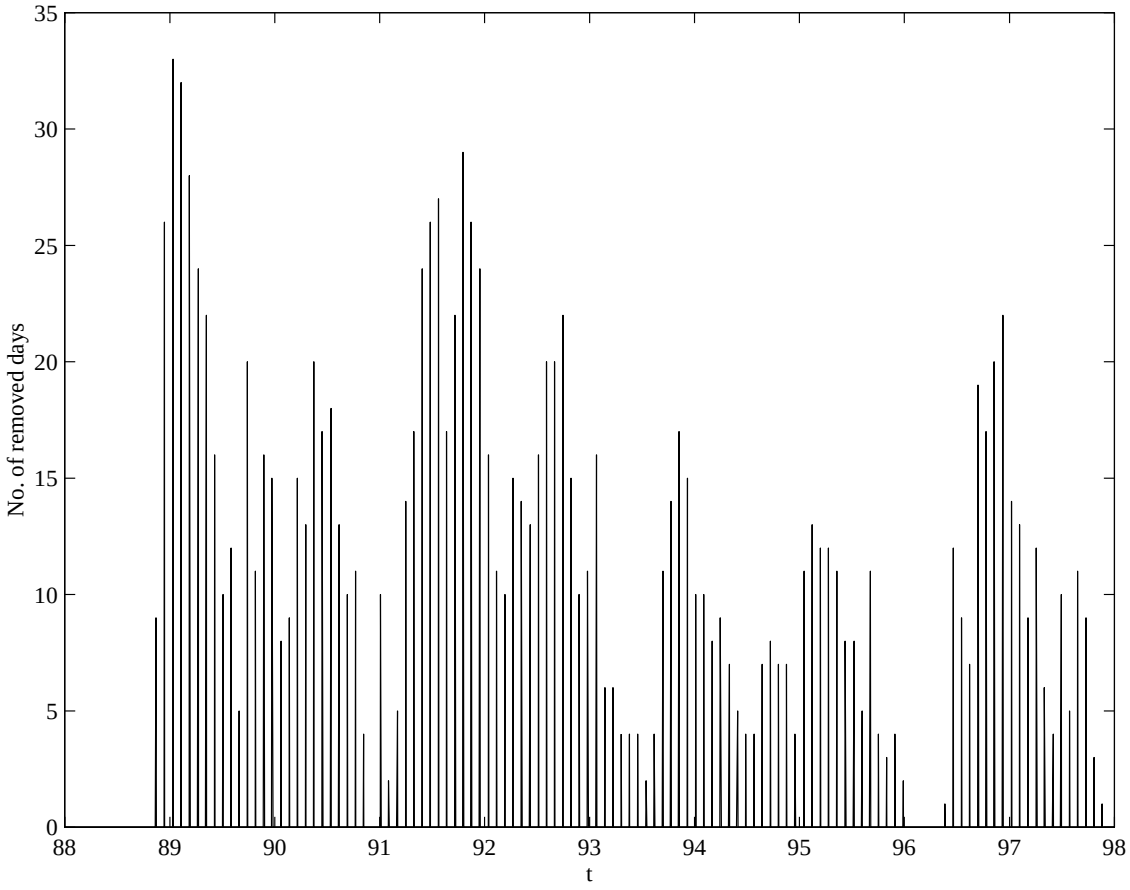

Figure 7: Number of removed days for each monthly prediction $\hat{C}(t)$. A day i is removed if $K_i > 3\%$. The average number of removed days is 11.7. Data from 29 American stocks. The modelling window is 200 days long.

6. SUMMARY AND CONCLUSIONS

The described *Remove* algorithm gives a significant reduction in prediction error for the covariance matrix. The RMSE for the covariances is reduced by 8.9% for the Swedish stock data and by 3.5% for the American stock data. In the portfolio optimization, the *Remove* algorithm gives 2.5% higher R_{ADJ} for the Swedish stock data and 1.9% higher for the American stock data. The fact that the increase in R_{ADJ} is smaller than the decrease in RMSE indicates that outliers in data do not affect the computation of optimal portfolios to any significant degree. It is possible that outlier detection has to be brought down to stock level instead of looking at a day level and the algorithm could be expanded in this direction. The computational demands would in such case rule out an exhaustive search and applying genetic algorithms would be a plausible and interesting track for future research.

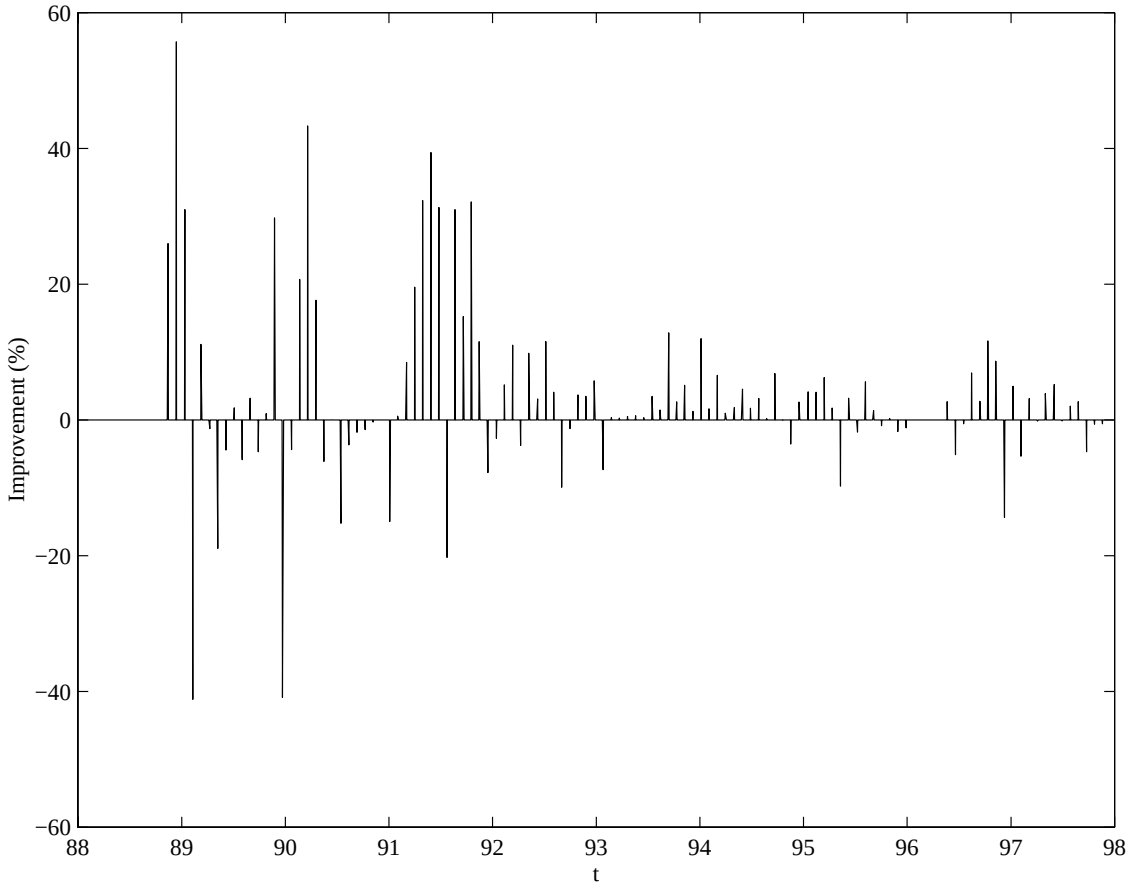

Figure 8: The improvement (reduction of RMSE for the covariance predictions) for each monthly prediction $\hat{C}(t)$ relative the naive prediction $\hat{C}_0(t)$ is plotted versus time. The average improvement is 3.5%. Data from 29 American stocks.

The *Remove* algorithm shows a clear ability to detect and eliminate the effect of simulated outliers and could also be applied to other time series related modelling problems with outliers in data.

7. ACKNOWLEDGEMENT

A very special thanks to Professor Zvi Gilula who provided me with invaluable help and encouragement throughout the process of writing this paper. Also thanks to Heim Levy for enlightening discussions and to Xavier de Luna who pointed out the references to earlier work in outlier detection. Part of this work has been performed during two inspiring visits to the department of Statistics at the Hebrew University in Jerusalem, Israel.

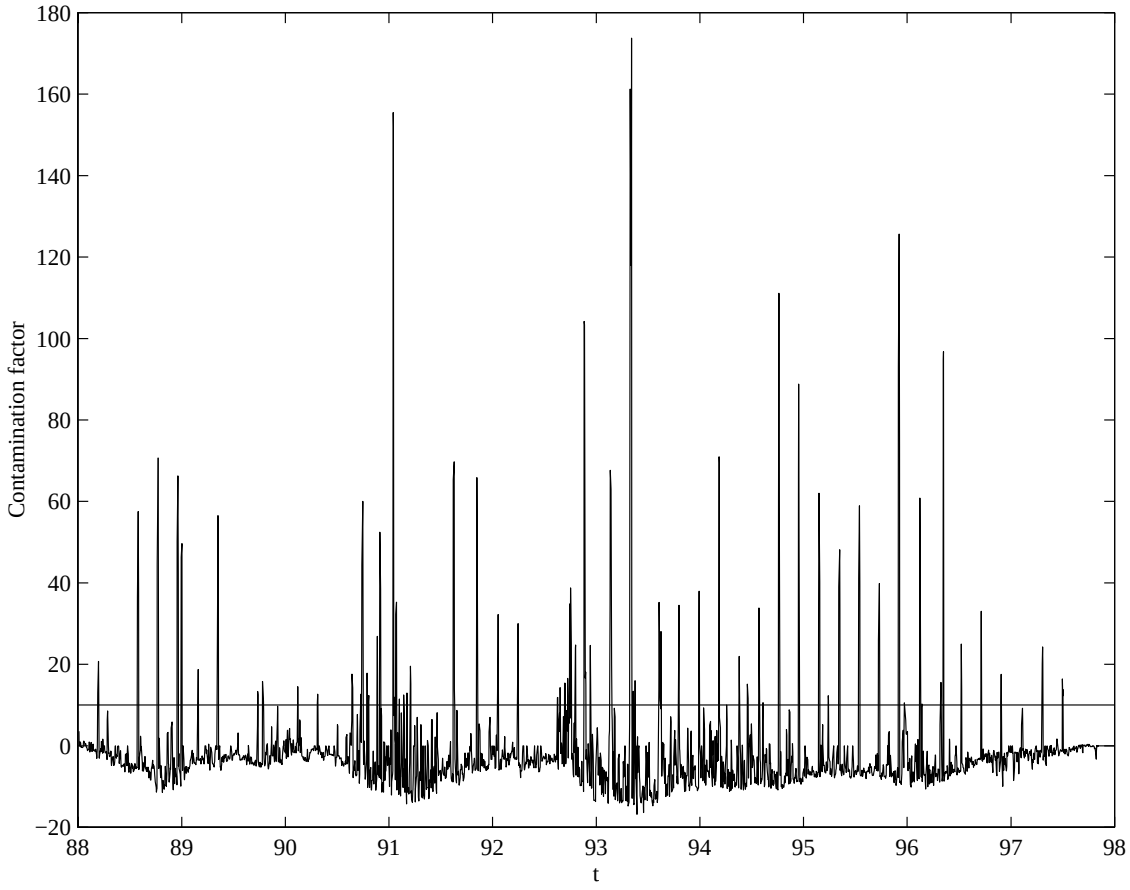

Figure 9: Noise injected in the data. Every 50th trading days has a random noise added to the stock prices. The noisy days get clearly detected and removed by the algorithm. Original data is taken from 24 Swedish stocks.

BIBLIOGRAPHY

1. Barnett, V. and Lewis, T., *Outliers In Statistical Data*, John Wiley and Sons, (1994).
2. Ledoit, O. *Improved estimation of the covariance matrix of stock returns with an application to portfolio selection*, Technical report, Anderson Graduate School of Management at UCLA, (1999).
3. Ma, Y. and Genton, M. G., *Highly robust estimation of dispersion matrices* Technical report, Department of Mathematics, 2-390, Massachusetts Institute of Technology, (1999).
4. Markowitz, H. M., *Portfolio selection*, Journal of Finance, **7**, (1952),77–91.
5. Mosteller, F. and Tukey, J. W., *Data analysis, including statistics.* In Lindzey, G. and Aronson, E., (ed), Handbook of Social Psychology, Addison Wesley, **2**,(1968),1–26.

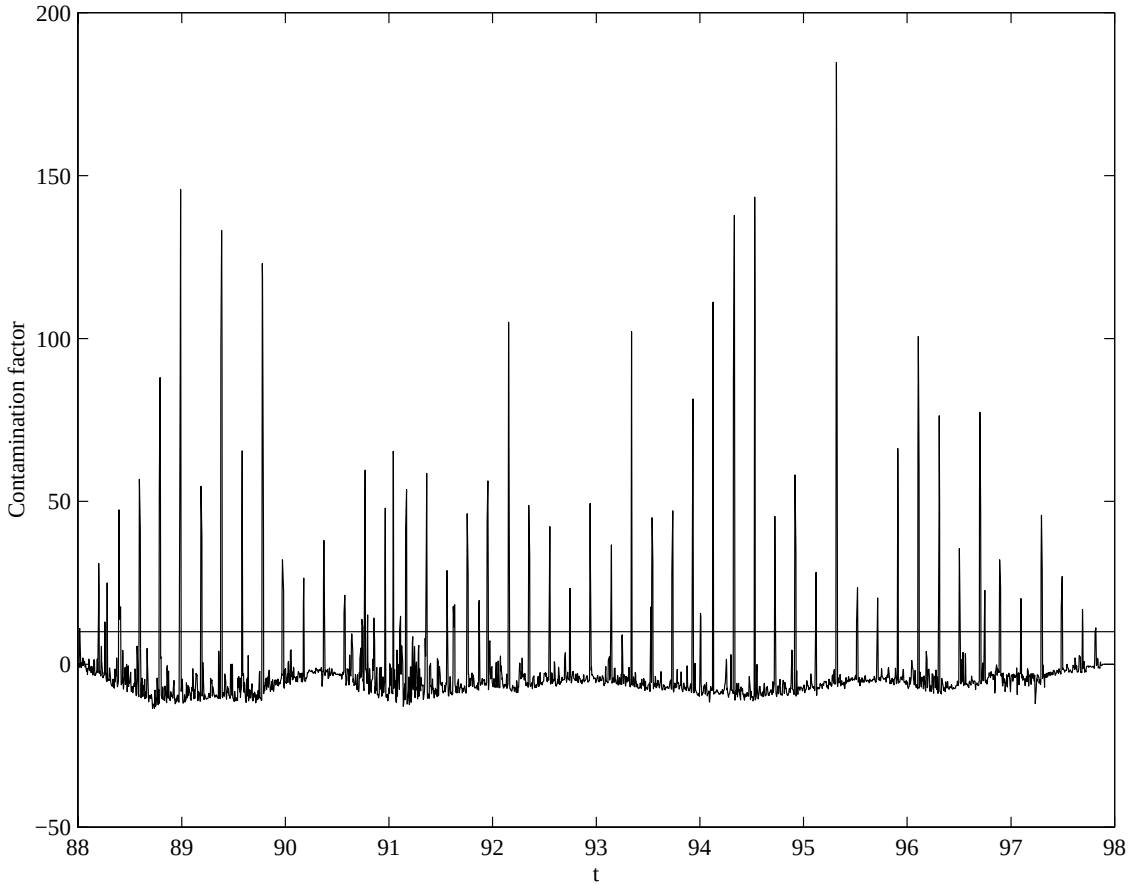

Figure 10: Noise injected in the data. Every 50th trading days has a random noise added to the stock prices. The noisy days get clearly detected and removed by the algorithm. Original data is taken from 29 American stocks.

6. Pena and Yohai. *A fast procedure for outlier diagnostics in large regression problems,* J. of the American Statistical Association, (1994), 434–445.
7. Stone, M., *Cross-validation: A review,* Mathematische Operationsforschung und Statistik, **9(1)**, (1978), 127–139.

Department of Computing Science, Umeå University, 901 87, Umeå, Sweden.
E-mail: thomash@cs.umu.se
http://www.cs.umu.se/~thomash