Statistical and Probabilistic Methods for Data Stream Mining

Von der Carl-Friedrich-Gauß-Fakultät

Technische Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades

Doktorin der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation

von Katharina Tschumitschew

geboren am 19.07.1980

in Solnetschnodolsk, Russland

Eingereicht am: 21.02.2012

Mündliche Prüfung am: 03.07.2012

Referent: Prof. Dr. Frank Klawonn

Korreferent: Prof. Dr. Rudolf Kruse

Korreferent: Prof. Dr. Dirk Christian Mattfeld

(Druckjahr: 2012)

Zusammenfassung

Das Hauptziel dieser Arbeit ist es, zentrale Probleme und wichtige Aspekte im Datastream-Mining zu veranschaulichen und mögliche Lösungen zu diskutierten Problemen vorzustellen. Da die Anzahl der Daten bei Datastreams potenziell unendlich ist und die statistischen Eigenschaften der Daten sich mit der Zeit ändern können, lassen sich klassische Data-Mining- und Statistikmethoden nicht auf Data Streams direkt anwenden. Aus diesem Grund werden im Rahmen dieser Arbeit bereits existierende Ansätze an die Datastream-Problematik angepasst und neue Methoden entwickelt.

Zum Beispiel werden inkrementelle oder rekursive Berechnungen statistischer Parameter und statistischer Tests vorgestellt, die nötig sind, um Berechnungen online und auf Hardware wie Steuergeräten mit teilweise recht begrenzter Rechen- und Speicherkapazität ausführen zu können. Ein wesentliches Problem stellt die Unterscheidung zwischen zufälligen Schwankungen im Sinne von Rauschen und echten Änderungen in Datastreams dar. Es bietet sich an, Hypothesentests mit inkrementeller Berechnung für dieses Problem der Change Detection einzusetzen. In dieser Arbeit werden inkrementelle und auf Fenstertechnik basierende statistische Tests für Change Detection vorgestellt.

Die Mehrzahl der existierenden Algorithmen zum Datastream-Mining verwenden keine expliziten Methoden zur Change Detection, sondern benutzen für die Vorhersage gleitende Fenster fester Breite. Nur wenige dieser Methoden beschäftigen sich mit der Frage wie die Fenstergröße ausgewählt werden soll und welche Effekte Veränderungen in den Daten auf die Vorhersagequalität haben. Hierzu wird eine theoretische Analyse für die optimale Fensterbreite für zwei Datenmodelle durchgeführt und gezeigt, dass eine suboptimale Fenstergröße zur drastischen Senkung der Vorhersagequalität führen kann. Außerdem können die vorgestellten Datenmodelle als Benchmark Tests für fensterbasierte Ansätze verwendet werden. Dies kann einen Eindruck vermitteln, wie stark ein sich an Datastreams automatisch anpassendes “Evolving System” durch Rauschen in den Daten negativ beeinflusst wird.

Abstract

The aim of this work is not only to highlight and summarize issues and challenges which arose during the mining of data streams, but also to find possible solutions to illustrated problems. Due to the streaming nature of the data, it is impossible to hold the whole data set in the main memory, i.e. efficient on-line computations are needed. For instance incremental calculations could be used in order to avoid to start the computation process from scratch each time new data arrive and to save memory. Another important aspect in data stream analysis is that the data generating process does not remain static, i.e. the underlying probabilistic model cannot be assumed to be stationary. The changes in the data structure may occur over time. Dealing with non-stationary data requires change detection and on-line adaptation. Furthermore real data is often contaminated with noise, this causes a specific problem for approaches dealing with the data streams. They must be able to distinguish between changes according to noise and changes of the underlying data generating process or its parameters.

In this work we propose a variety of different methods, which fulfil specific requirements of data stream mining. Furthermore we carry out theoretical analysis of effects of noise and changes in data stream for sliding window based evolving system in order to illustrate the problem of suboptimal window size. In order to do the validation of an evolving system significant, we propose some simple benchmark tests that can give an idea of how much an evolving system might be misled by noise.

Danksagung

Besonderen Dank möchte ich meinem Doktorvater Herrn Professor Dr. Frank Klawonn für die herausragende Betreuung meiner Arbeit, die ständige Unterstützung und wertvollen Ratschläge aussprechen.

Mein Dank gilt auch Herrn Professor Dr. Rudolf Kruse und Herrn Professor Dr. Dirk Christian Mattfeld, die sich bereit erklärt haben als Zweit- und Dritt-Gutachter meine Arbeit zu lesen und zu beurteilen.

Ein weiteres Dankeschön gilt meiner Familie und meinen Freunden, die stets an mich geglaubt haben. Vielen Dank für die moralische Unterstützung.

Contents

1 Introduction **1**

2 Incremental Statistical Measures **7**

2.1 Incremental calculation of moments and the Pearson correlation coefficient. 9

2.2 Hypothesis tests and change detection 16

2.2.1 Incremental hypothesis tests 16

2.2.2 Change detection strategies 27

3 Incremental Quantile Estimation **35**

3.1 Related work 36

3.2 Incremental median estimation 38

3.3 Incremental quantile estimation 43

3.3.1 An ad hoc algorithm 43

3.3.2 Incremental quantile estimation with presampling 46

3.3.3 Complexity of the algorithm 48

3.3.4 Choice of the parameters m, n and l 48

3.3.5 Justification for the choice of $p_l \approx 0.5$ 48

3.3.6 Detecting changes 51

3.3.7 Evolving environment 56

3.4 Experimental results 56

4 Analysis of Effects of Noise and Changes in the Data in Evolving Systems Based on Simple Stochastic Models **60**

4.1 Random walk 61

4.2 Switch model 65

5 Analysis of Regression Models for Sliding Window Based Evolving Systems **69**

5.1 Related work 70

5.2 Constant model with drift and noise 71

5.3 Linear model with drift and noise 77

5.4 Estimation of the optimal window size 86

5.5 Consequences for non-stationary meta-models 88

6 Conclusion **94**

Chapter 1

Introduction

Nowadays it is very important to continuously collect and analyse data sets increasing with time, since the (new) data may contain useful information. Sensor data as well as the seasonal behaviour of markets, weather or animals are in the focus of diverse research studies. The amount of recorded data increases each day. Apart from the huge amount of data to be dealt with, another problem is that the data arrive continuously in time. Such kind of data is called data stream. A data stream can be characterised as an unlimited sequence of values arriving step by step over time. One of the main problems for the analysis of data streams is limited computing and memory capabilities. It is impossible to hold the whole data set in the main memory of a computer or computing device like an ECU (electronic control unit) that might also be responsible for other tasks than just analysing the data. Moreover, the results of the analysis should be presented in acceptable time, sometimes even under very strict time constraints, so that the user or system can react in real time. Therefore, the analysis of data streams requires efficient on-line computations. Algorithms based on incremental or recursive computation schemes satisfy the above requirements. Such methods do not store all historical data and do not need to browse through old data to update an estimator or an analysis, in the ideal case, each data value is touched only once. But even for large sets of collected data such methods are of interest, since complexity should very often be at most linear in the number of data – in the ideal case one-pass methods touching each data value only once – in order to carry out an analysis on large data sets.

Statistical measures provide essential and valuable information about data and are needed for any kind of data analysis. Statistical measures can be used in a

purely exploratory context to describe properties of the data, but also as estimators for model parameters or in the context of hypothesis testing. For example, the mean value is a measure for location, but also an estimator for the expected value of a probability distribution from which the data are sampled. Statistical moments of higher order than the mean provide information about the variance, the skewness and the kurtosis of a probability distribution. The Pearson correlation coefficient is a measure for linear dependency between two variables. In robust statistics, quantiles play an important role, since they are less sensitive to outliers. The median is an alternative measure of location, the interquartile range an alternative measure of dispersion. The application of statistical measures to data streams requires on-line calculation. Since data come in step by step, incremental calculations are needed to avoid to start the computational process each time new data arrive and to save memory so that not the whole data set needs to be kept in the memory. Statistical measures like the mean, the variance, moments in general and the Pearson correlation coefficient render themselves easily to incremental computations, whereas recursive or incremental algorithms for quantiles are not as simple or obvious.

In Chapters 2 and 3, which are mainly based on the publications [45, 42]. we discuss the application of statistical measures to data streams. Equations for incremental computations of the mean, variance, third and fourth moments and the Pearson correlation coefficient are explained in Chapter 2 Section 2.1. Two algorithms for the on-line estimation of quantiles are described in Chapter 3 Section 3.2.

Another important aspect in data stream analysis is that the data generating process does not remain static, i.e. the underlying probabilistic model cannot be assumed to be stationary. The changes in the data structure may occur over time. Dealing with non-stationary data requires change detection and on-line adaptation. Different kinds of non-stationarity have been classified in [4]:

- Changes in the data distribution: the change occurs in the data distribution. For instance mean or variance of the data distribution may change over time.

- Changes in concept: here concept drift refers to changes of a target variable. A target variable is a variable, whose values we try to predict based on the model estimated from the data, for instance for linear regression it is the

change of the parameters of the linear relationship between the data.

- Concept drift: concept drift describes gradual changes of the concept. In statistics, this usually called structural drift.
- Concept shift: concept shift refers to an abrupt change which is also referred to as structural break.

In further analysis we don't differentiate between both types, since the distribution of the target variable will be changed in case of concept drift as well as in case of change in the data distribution.

The real world examples for the non-stationary data are for instance stock market, weather prediction, change of the protein structure through mutation, buying behaviour of customers of an online store and many others. Since non-stationary data models significantly affect the accuracy of prediction, the fact of concept drift should be taken into account by on-line learning. Hence the effective treatment of non-stationarity is an important problem in machine learning. Therefore change detection and on-line adaptation for data stream mining techniques are required for non-stationary data streams. Various strategies to handle non-stationarity are proposed, see for instance [17] for a detailed survey of change detection methods. Statistical hypothesis tests may also be used for change detection. Since we are working with data streams, it is required that the calculations for the hypothesis tests can be carried out in an incremental way. For instance, the χ^2-test and the t-test render themselves easily to incremental computations. On-line adaptations of statistical hypothesis tests and different change detection strategies are described in Chapter 2. The incremental quantile estimator iQPres from Chapter 3 can be used for change detection as well.

Once the change is detected the adaptation of learning methods should be carried out. Commonly used approaches for data stream mining under assumptions of non-stationarity of the data generating process are evolving systems. Evolving systems are designed to cope with streaming non stationary data. There are two important aspects that are usually considered for evolving systems. It is required that the evolving system can react in real time so that efficient computations – usually recursive ones – are needed without analyzing the whole data that have been collected so far again. The second aspect is the change of the underlying

process that generates the data, so that the evolving system must adapt itself online. In the simplest case an evolving system is based on a windowing techniques. For instance a sliding window or weighted window technique could be used. The majority of these approaches don't use explicit change detection strategies, the system adapts itself each time new data arrive.

However the problem with real data is often that noise and other forms of randomness are involved. This causes a specific problem for evolving systems. They must be able to distinguish between changes according to noise and changes of the underlying data generating process or its parameters.

Statisticians categorize models that do not make explicit assumptions about the data generating process as exploratory data analysis techniques. Such methods, like decision or regression trees, neural networks or support vector machines are very successful in classification and regression tasks, but there is a need to evaluate the performance of such models, since they cannot be analyzed in the classical statistical sense where the underlying assumptions about the data generating process must be made explicit. A very common way to evaluate the performance of such models are methods that divide the data into training and test data like, for instance 10-fold cross-validation. The training data are used to construct the model and the performance is evaluated based on the test data.

For evolving systems, the partition into training and test data is more complicated. We cannot simply take out some data for testing, since the data generating process is assumed to change and the evolving system is intended to track these changes. What could be done is to take out single data records of a time series and see how the evolving system performs with respect to these records. However, to be representative, we would need a much larger test data set due to the fact that the data generating process changes and the test data must reflect situations with changes and without changes accordingly. Even then it is not clear how to judge the performance of an evolving system. A large error can be due to overfitting – i.e. erroneously tracking the noise in the data – or because there is so much noise in the data that the performance cannot be better. A rough indication of how much an evolving system tries to track the noise, is the difference between the error on the training and the test data. However, if an evolving system has already a bad performance on the training data, the performance on the test data cannot be much

worse and one might assume that the error comes from the noise in the data.

We propose some simple benchmark tests that can give an idea of how much an evolving system might be misled by noise. For this purpose we set up some simple theoretical models for the data generating process. Therefore we can build a model which takes the information about the data generating process into account. A comparison between such "expert" or "oracle" models and evolving systems is carried out subsequently. Of course it is obvious that the performance of the evolving system will be worse, since the evolving system does not make any specific assumptions about the data generating process. But at least we will have an impression of how much the evolving systems are biased to track the noise or randomness in the data generating process instead of learning the actual dependencies in the data. In Chapter 4 based on the publication [43] we carry out theoretical and experimental analysis for two simple data generating processes: random walk and switching model. The first model can be interpreted as regression and the second either as a regression or a classification problem.

Another interesting aspect of evolving systems is the dependencies between the choice of window size and the accuracy of prediction for non-stationary and noisy data. As we show later, a suboptimal window size can decrease prediction quality drastic. In Chapter 5, based on the manuscript [44] we carry out theoretical analysis of two models: constant and linear model with drift and noise. For this purpose we consider a simple prediction task, the prediction of the next value which can be understood as regression. In order to analyse which effect has drift and noise on optimal window size, the size of the window achieving the minimum of the expected quadratic error should be computed and this optimal window size should be determined as a function of the data generating process parameters. Furthermore we carry out an empirical analysis of the expected quadratic error function as well. In such a way we can compare how much worse the prediction accuracy would be for a suboptimal window size.

Of course such analysis is of theoretical nature and can not be understood as an exact instruction for the choice of the parameters for an evolving system. However it can lead to better understanding about how strong drift, noise and the relationship between both of them can affect the optimal window size. Therefore one might try to choose the amount of the data to be used for prediction correspond-

ing to the knowledge about the data generating process. On the other hand taken into account the information about which window size yields the best results, one could try to estimate the proportion of the drift in comparison to the noise in the data. Since typically the real data does not follow one of the presented models, such estimation of the proportion for drift and noise would be only of approximate character. However those considerations can give us at least a rough idea about how much the data is contaminated with noise compared to the drift in the data.

Both these models, similar to the approach in Chapter 4, could be used as benchmark tests for evolving system.

Chapter 2

Incremental Statistical Measures

Statistics and statistical methods are used in almost every aspect of modern life, like medicine, social surveys, economy and marketing, only to name few of application areas. A vast number of sophisticated statistical software tools can be used to search and test for structures and patterns in data. Important information about the data generating process is provided by the simple summary statistics. Characteristics of the data distribution can be described by summary statistics like the following one.

- Measures of location: The mean and quantiles provide information about location of the distribution. Mean and median are representatives for the centre of the distribution.
- Measures of spread: Common measures for the variation in the data are standard deviation, variance and interquartile range.
- Shape: The third and fourth moments provide information about the skewness and the kurtosis of a probability distribution.
- Dependence: For instance, the Pearson correlation coefficient is a measure for the linear dependency between two variables. Other common measures for statistical dependency between two variables rank correlation coefficients like Spearman's rho or Kendall's tau.

Apart from providing information about location and spread of the data distribution, quantiles also play an important role in robust data analysis, since they are less sensitive to outliers.

Summary statistics can be used in a purely exploratory context to describe properties of the data, but also as estimators for model parameters of an assumed underlying data distribution.

More complex and powerful methods for statistical data analysis are for instance hypothesis tests. Statistical hypothesis testing allows us to discover the current state of affairs and therefore help us to make decisions based on the gained knowledge (see for instance [41]). Hypothesis test can be applied to a great variety of problems. We may need to test just a simple parameter or the whole distribution of the data.

However, classical statistics operates with a finite, fixed data set. On the other hand, these days the need to collect and analyse the data streams increases every day. Consequently the application of statistical methods to data streams requires modifications to the standard calculation schemes in order to be able carry out the computations on-line. Since data come in step by step, incremental calculations are needed to avoid to start the computation process from scratch each time new data arrive and to save memory, so that not the whole data set must be kept in the memory. Statistical measures like the sample mean, variance and moments in general and the Pearson correlation coefficient render themselves easily to the incremental computation schemes, whereas, for instance, for standard quantiles computations the whole data is needed. In such cases, new incremental methods must be developed that avoid sorting the whole data set, since sorting requires in principal to check the whole data set. Several approaches for the on-line estimation of quantiles are presented for instance in [14, 35, 1, 42].

Another main challenge in data stream mining is non-stationarity of the data generating process. As already mentioned in the introduction two types of changes may occur: changes in the data distribution and concept changes. Therefore we need strategies to detect both kind of changes. Various strategies for change detection are proposing during last years (see for instance [17]), in this chapter we focus on statistical hypothesis tests for change detection.

With respect to streaming nature of the data, the calculations for the hypothesis tests must be carried out either in incremental way or window techniques should be used. For instance, the χ^2-test and the t-test[1] render themselves easily to incre-

[1]For precise definitions see Section 2.2.

mental computations. A technique based on window technique and for instance Kolmogorov-Smirnov test can be used for the change detection as well.

This chapter is organised as follows. Incremental computations of the mean, variance, third and fourth moments and the Pearson correlation coefficient are explained in Section 2.1. In Section 2.2 we provide on-line adaptations of statistical hypothesis test and discuss different change detection strategies.

Since the on-line computation of quantiles required detailed explanation, the on-line quantile estimator is described in Chapter 3.

2.1 Incremental calculation of moments and the Pearson correlation coefficient.

Statistical measures like sample central moments provide valuable information about the data distribution. So the sample mean or empirical mean (first sample central moment) is the measure of the centre of location of the data distribution, the measure of variability is sample variance (second sample central moment). The third and fourth central moments are used to compute skewness and kurtosis of the data sample. Skewness provides us the information about the asymmetry of the data distribution and kurtosis give us an idea about the degree of peakedness of the distribution.

Another important statistic is the correlation coefficient. The correlation coefficient is a measure for linear dependency between two variables.

In this section we introduce incremental calculations for these statistical measures.

In the following, we consider a real-valued sample $x_1, \ldots, x_t, \ldots$ ($x_i \in \mathbb{R}$ for all $i \in \{1, \ldots, t, \ldots\}$).

Definition 1 *Let $x_1, \ldots, x_t$ be a random sample from the distribution of the random variable X.*

The sample or empirical mean of the sample of size t, denoted by $\bar{x}_t$, is given by the formula

$$\bar{x}_t = \frac{1}{t} \sum_{i=1}^{t} x_i. \tag{2.1}$$

Equation (2.1) can not be applied directly in the context of data streams, since it would require to consider all sample values at each time step. Fortunately,

Equation (2.1) can be easily transformed into an incremental scheme.

$$\begin{aligned}
\bar{x}_t &= \frac{1}{t}\sum_{i=1}^{t} x_i \\
&= \frac{1}{t}\left(x_t + \sum_{i=1}^{t-1} x_i\right) \\
&= \frac{1}{t}\left(x_t + (t-1)\,\bar{x}_{t-1}\right) \\
&= \bar{x}_{t-1} + \frac{1}{t}\left(x_t - \bar{x}_{t-1}\right). \qquad (2.2)
\end{aligned}$$

The incremental update Equation (2.2) requires only three values to calculate the sample mean at time point t:

- The mean at time point $t-1$.
- The sample value at time point t.
- The number of sample values so far.

The empirical or sample variance can be calculated in an incremental fashion in a similar way.

Definition 2 *Let $x_1, \ldots, x_t$ be a random sample from the distribution of the random variable X. The empirical or sample variance of a sample of size t is given by*

$$s_t^2 = \frac{1}{t-1}\sum_{i=1}^{t}(x_i - \bar{x}_t)^2 \qquad (2.3)$$

Furthermore, $s_t = \sqrt{s_t^2}$ is called the sample standard deviation.

In order to simplify the calculation we use following notation:

$$\tilde{m}_{2,t} = \sum_{i=1}^{t}(x_i - \bar{x}_t)^2 \qquad (2.4)$$

In the following, the formula for incremental calculation is derived from Equation (2.4) using Equation (2.2).

$$
\begin{aligned}
\tilde{m}_{2,t} - \tilde{m}_{2,t-1} &= \sum_{i=1}^{t} x_i^2 - t\bar{x}_t^2 - \sum_{i=1}^{t-1} x_i^2 + (t-1)\bar{x}_{t-1}^2 \\
&= x_t^2 - t\bar{x}_t^2 + (t-1)\bar{x}_{t-1}^2 \\
&= x_t^2 - \bar{x}_{t-1}^2 + t\left(\bar{x}_{t-1}^2 - \bar{x}_t^2\right) \\
&= x_t^2 - \bar{x}_{t-1}^2 + t\left(\bar{x}_{t-1} - \bar{x}_t\right)\left(\bar{x}_{t-1} + \bar{x}_t\right) \\
&= x_t^2 - \bar{x}_{t-1}^2 + t\left(\bar{x}_{t-1} - \bar{x}_{t-1} - \frac{1}{t}\left(x_t - \bar{x}_{t-1}\right)\right)\left(\bar{x}_{t-1} + \bar{x}_t\right) \\
&= x_t^2 - \bar{x}_{t-1}^2 + \left(\bar{x}_{t-1} - x_t\right)\left(\bar{x}_{t-1} + \bar{x}_t\right) \\
&= \left(x_t - \bar{x}_{t-1}\right)\left(x_t + \bar{x}_{t-1} - \bar{x}_{t-1} - \bar{x}_t\right) \\
&= \left(x_t - \bar{x}_{t-1}\right)\left(x_t - \bar{x}_t\right).
\end{aligned}
$$

Consequently, we obtain the following recurrence formula for the second central moment:

$$\tilde{m}_{2,t} = \tilde{m}_{2,t-1} + \left(x_t - \bar{x}_{t-1}\right)\left(x_t - \bar{x}_t\right) \tag{2.5}$$

The unbiased estimator for the variance of the sample according to the Equation (2.5) is given by

$$s_t^2 = \frac{1}{t-1} M_{2,t} = \frac{(t-2)\, s_{t-1}^2 + \left(x_t - \bar{x}_{t-1}\right)\left(x_t - \bar{x}_t\right)}{t-1}. \tag{2.6}$$

Definition 3 *Let $x_1, \ldots, x_t$ be a random sample from the distribution of the random variable X. Then the k-th central moment of a sample of size t is defined by*

$$m_{k,t} = \frac{1}{t} \sum_{i=1}^{t} (x_i - \bar{x}_t)^k. \tag{2.7}$$

In order to simplify the computations and to facilitate the readability of the text we use the following expression for the derivation.

$$\tilde{m}_{k,t} = \sum_{i=1}^{t} (x_i - \bar{x}_t)^k, \tag{2.8}$$

therefore $\tilde{m}_{k,t} = t \cdot m_{k,t}$.

For the third- and fourth-order moments, which are needed to calculate skewness and kurtosis of the data distribution, incremental formulae can be derived in a similar way, in the form of pairwise update equations for $\tilde{m}_{3,t}$ and $\tilde{m}_{4,t}$.

$$
\begin{aligned}
\tilde{m}_{3,t} &= \sum_{i=1}^{t-1} (x_i - \bar{x}_t)^3 + (x_t - \bar{x}_t)^3 \\
&= \sum_{i=1}^{t-1} \left(x_i - \bar{x}_{t-1} - \frac{1}{t}(x_t - \bar{x}_{t-1}) \right)^3 + \left(x_t - \bar{x}_{t-1} + \frac{1}{t}(x_t - \bar{x}_{t-1}) \right)^3 \\
&= \sum_{i=1}^{t-1} ((x_i - \bar{x}_{t-1}) - b)^3 + (tb - b)^3 \\
&= \sum_{i=1}^{t-1} \left((x_i - \bar{x}_{t-1})^3 - 3b(x_i - \bar{x}_{t-1})^2 + 3b^2(x_i - \bar{x}_{t-1}) - b^3 \right) + (t-1)^3 b^3 \\
&= \tilde{m}_{3,t-1} - 3b\tilde{m}_{2,t-1} - ((t-1)b^3 + (t-1)^3 b^3 \\
&= \tilde{m}_{3,t-1} - 3b\tilde{m}_{2,t-1} + t(t-1)(t-2)b^3 \qquad (2.9)
\end{aligned}
$$

where $b = \frac{x_t - \bar{x}_{t-1}}{t}$.

From Equation (2.9) we obtain a one-pass formula for the third-order centred statistical moment of a sample of size t:

$$
\tilde{m}_{3,t} = \tilde{m}_{3,t-1} - 3\frac{(x_t - \bar{x}_{t-1})}{t}\tilde{m}_{2,t-1} + \frac{(t-1)(t-2)}{t^2}(x_t - \bar{x}_{t-1})^3 . \qquad (2.10)
$$

The derivation for the fourth-order moment is very similar to Equation (2.9) and thus is not detailed here.

$$
\begin{aligned}
\tilde{m}_{4,t} &= \tilde{m}_{4,t-1} - 4\frac{(x_t - \bar{x}_{t-1})}{t}\tilde{m}_{3,t-1} + 6\left(\frac{x_t - \bar{x}_{t-1}}{t}\right)^2 \tilde{m}_{2,t-1} \\
&\quad + \frac{(t-1)\left(t^2 - 3t + 3\right)}{t^3}(x_t - \bar{x}_{t-1})^4 . \qquad (2.11)
\end{aligned}
$$

The results presented above offer the essential formulae for efficient, one-pass calculations of statistical moments up to the fourth order. Those are important when the data stream mean, variance, skewness and kurtosis should be calculated. Although these measures cover the needs of the vast majority of applications for data analysis, sometimes higher-order statistics should be used. For the computation of higher-order statistical moments see for instance [10].

Now we derive a formula for the incremental calculation of the sample correlation coefficient

Definition 4 *Let $x_1, \ldots, x_t$ be a random sample from the distribution of the random variable X and $y_1, \ldots, y_t$ be a random sample from the distribution of the random variable Y. Then the sample Pearson correlation coefficient of the sample*

of size t, denoted by $r_{xy,t}$, is given by the formula

$$r_{xy,t} = \frac{\sum_{i=1}^{t}(x_i - \bar{x}_t)(y_i - \bar{y}_t)}{(t-1)\,s_{x,t}s_{y,t}} \tag{2.12}$$

where $\bar{x}_t$ and $\bar{y}_t$ are the sample means of X and Y and $s_{x,t}$ and $s_{y,t}$ are the sample standard deviations of X and Y, respectively.

The incremental formula for the sample standard deviation can be easely derived from the incremental formula for sample variance (2.6). Hence only the numerator of Equation (2.12) needs to be considered further. Furthermore, the numerator of Equation (2.12) represents the sample covariance $s_{xy,t}$.

Definition 5 *Let $x_1,\ldots,x_t$ be a random sample from the distribution of the random variable X and $y_1,\ldots,y_t$ be a random sample from the distribution of the random variable Y. Then the sample covariance $s_{xy,t}$ of the sample of size t is given by t*

$$s_{xy,t} = \frac{\sum_{i=1}^{t}(x_i - \bar{x}_t)(y_i - \bar{y}_t)}{t-1} \tag{2.13}$$

where $\bar{x}_t$ and $\bar{y}_t$ are the sample means of X and Y and $s_{x,t}$ and $s_{y,t}$ are the sample standard deviations of X and Y, respectively.

The formula for the incremental calculation of the covariance is given by

$$\begin{aligned}
(t-1)\,s_{xy,t} &= \sum_{i=1}^{t-1}(x_i - \bar{x}_t)(y_i - \bar{y}_t) + (x_t - \bar{x}_t)(y_t - \bar{y}_t) \\
&= \sum_{i=1}^{t-1}((x_i - \bar{x}_{t-1}) - b_x)((y_i - \bar{y}_{t-1}) - b_y) + (t-1)^2 b_x b_y \\
&= (t-2)\,s_{xy,t-1} + t\,(t-1)\,b_x b_y
\end{aligned} \tag{2.14}$$

where $b_x = \frac{(x_t - \bar{x}_{t-1})}{t}$ and $b_y = \frac{(y_t - \bar{y}_{t-1})}{t}$. Hence the incremental formula for the sample covariance is

$$s_{xy,t} = \frac{(t-2)}{(t-1)}s_{xy,t-1} + \frac{1}{t}(x_t - \bar{x}_{t-1})(y_t - \bar{y}_{t-1}) \tag{2.15}$$

Therefore, to update the Pearson correlation coefficient, we have to compute the sample standard deviation and covariance first and subsequently use Equation (2.12).

Above in this section we presented incremental calculations for the empirical mean, empirical variance, third and fourth sample central moments and sample

correlation coefficient. These statistical measures can also be considered as estimators of the corresponding parameters of the data distribution. Therefore, we are interested in the question how many values x_i do we need to get a "good" estimation of the parameters. Of course, as we deal with a data stream, in general we will have a large amount of data. However, some application are based on time window techniques. For instance, for change detection methods presented in the section (2.2). Here we need to compare at least two samples of data, on that account, the data have to be split into smaller parts. To answer the question about the optimal amount of data for statistical estimators, we have to analyse the variances of the parameter estimators. The variance of an estimator shows how efficient this estimator is.

Here we restrict our considerations to a random sample from a normal distribution with expected value 0. Let $X_1, \ldots, X_t$ be independent and identically-distributed (i.i.d.) random variables following a normal distribution, $X_i \sim N\left(0,\ \sigma^2\right)$ and $x_1, \ldots, x_t$ are observed values of these random variables.

The variance of the estimator of the expected value[2] $\bar{X}_t = \frac{1}{t}\sum_{i=1}^{t} X_i$ is given by

$$Var\left(\bar{X}_t\right) = \frac{\sigma^2}{t}. \tag{2.16}$$

The variance of the unbaised estimator of the variance $S^2 = \frac{1}{t-1}\sum_{i=1}^{t}\left(X_i - \bar{X}_t\right)^2$ is given by

$$Var\left(S_t^2\right) = \frac{2}{(t-1)}\sigma^4. \tag{2.17}$$

The variance of the distribution of the third moment is shown in Equation (2.18) (see [10] for more detailed information)

$$Var\left(M_{3,t}\right) = \frac{6\,(t-1)\,(t-2)}{t^3}\sigma^6. \tag{2.18}$$

Figure 2.1 shows Equations (2.16), (2.17) and (2.18) as functions in t for $\sigma^2 = 1$ (standard normal population). It is obvious that for small amounts of data, the variance of the estimators is quite large, consequently more values are needed to obtain a reliable estimation of distribution parameters. Furthermore the optimal sample size depends on the statistic to be computed. For instance, for the sample mean and a sample of size 50, the variance is already small enough, whereas for

[2] We use capital letters here to distinguish between random variables and real numbers that are denoted by small letters.

the third moment estimator to have the same variance, many more observations are needed.

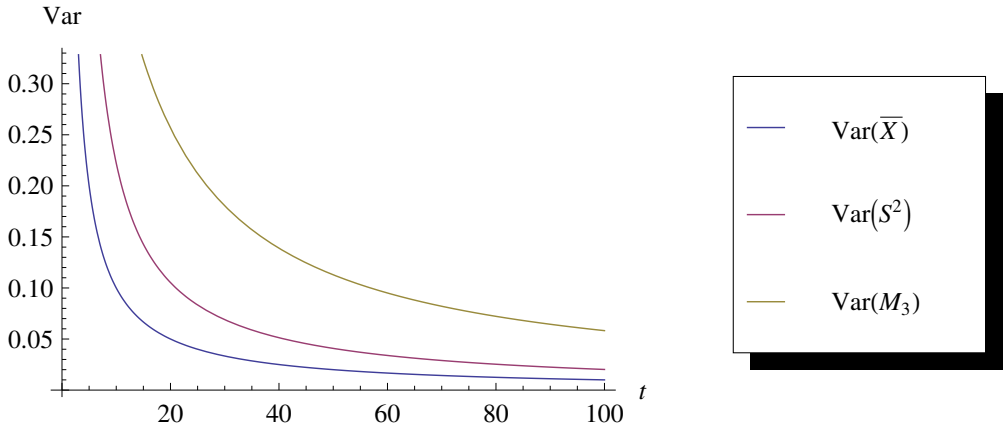

Figure 2.1: Variances from bottom to top of parameter estimators for the expected value, the variance and the third moment of a standard normal distribution

We apply the same considerations to the sample correlation coefficient. Let X and Y be two random variables following normal distributions and let $X_1, \ldots, X_t$ and $Y_1, \ldots, Y_t$ be i.i.d. samples of X and Y, respectively: $X_i \sim N\left(0,\ \sigma_x^2\right)$ and $Y_i \sim N\left(0,\ \sigma_y^2\right)$. Assume the correlation between X and Y is equal to ρ_{XY}. Then the asymptotic variance of the sample correlation coefficient is given by (see [11])

$$Var\left(R_{XY,t}\right) \approx \frac{\left(1-\rho_{XY}^2\right)^2}{t}. \tag{2.19}$$

Attention should be paid to the asymptotic nature of Equation (2.19). This formula can be used only for sufficiently large t (see [11]). Equation (2.19) is illustrated in Figure 2.2 as a function in t for $\rho_{XY} = 0.9$. Since for different values of ρ_{XY}, the plots are very similar, they are not shown here.

In this section we have provided equations for incremental calculation of the sample mean, sample variance, third and fourth moments and the Pearson correlation coefficient. These statistics allow us to summarize a set of observations analytically. Since we assume that the observations reflect the population as a whole, these statistics give us an idea about the underlying data distribution. Other important summary statistics are sample quantiles. Incremental approaches for quantiles estimation are described in the next chapter.

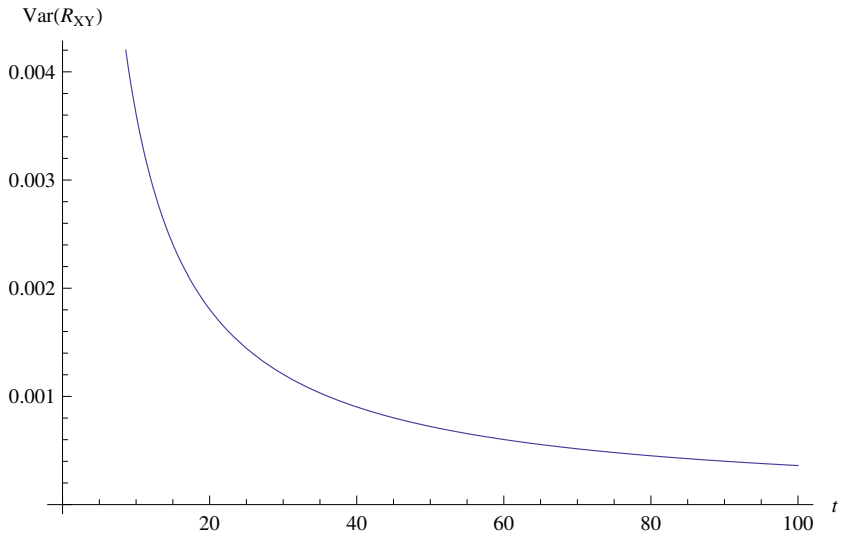

Figure 2.2: Asymptotic variance of the sample correlation coefficient

2.2 Hypothesis tests and change detection

In this section we demonstrate how hypothesis testing can be adapted to an incremental computation scheme for the cases of the χ^2-test and the t-test. Moreover we discus the problem of non-stationary data and explain various change detection strategies with the main focus on the use of statistical tests.

2.2.1 Incremental hypothesis tests

Statistical test are methods to check the validity of hypotheses about distributions or properties of distributions of random variables. Since statistical tests rely on samples, they cannot definitely verify or falsify a hypothesis. They can only provide probabilistic information supporting or rejecting the hypothesis under consideration.

Statistical tests usually consider a null hypothesis H_0 and an alternative hypothesis H_1. The hypotheses may concern parameters of a given class of distributions, for instance unknown expected value and variance of a normal distribution. Such tests are called parameter tests. In such cases, the a priori assumption is that the data definitely originate from a normal distribution. Only the parameters are unknown. In contrast to parameter tests, nonparametric tests concern more general hypothesis, for example whether it is reasonable at all to assume that the data come from a normal distribution.

The error probability that the test will erroneously reject the null hypothesis,

given the null hypothesis is true, is used as an indicator of the reliability of the test. Sometimes a so-called p-value is used. The p-value is smallest error probability that can be admitted, so that the test will still reject the null hypothesis for a given sample. Therefore, a low p-value is a good indicator for rejecting the null hypothesis. Usually, the acceptable error probability α (α-error) should be specified in advance, before the test is carried out. The smaller α is chosen, the more reliable is the test when the outcome is to reject the null hypothesis. However, when α is chosen too small, then the test will not tend to reject the null hypothesis, although the sample might not speak in favour of it.

Some of the hypothesis tests can be applied to data streams, since they can be calculated in an incremental fashion. We discuss in this section the incremental adaptation of two statistical tests, the χ^2-test and the t-test. Note, that the application of hypothesis tests to data streams, using incremental computation or window techniques, requires the repeated execution of the test. This can cause the problem of multiple testing. The multiple testing problem is described later in this section.

χ^2-test

The χ^2-test has various applications. The principal idea of the χ^2-test is the comparison of two distributions. One can check whether two samples come from the same distribution, a single sample follows a given distribution or also whether two samples are independent.

Example 1 *A die is thrown* 120 *times and the observed frequencies are as follows:* 1 *is obtained* 30 *times,* 2-25*,* 3-18*,* 4-10*,* 5-22 *and* 6-15*. We are interested in the question whether the die is fair or not.*

The null hypothesis H_0 for the χ^2-test claims that the data follow a certain (cumulative) probability distribution $F(x)$. The distribution of the null hypothesis is than compared to the distribution of the data. The null hypothesis can for instance be a given distribution, e.g. a uniform or a normal distribution, and the χ^2-test can give an indication, whether the data strongly deviate from this expected distribution. For an independence test for two variables, the joint distribution of the sample is compared to the product of the marginal distributions. If these distributions differ significantly, this is an indication that the variables might not be independent.

The main idea of the χ^2-test is to determine how well the observed frequencies fit the theoretical/expected frequencies specified by the null hypothesis. Therefore, the χ^2-test is appropriate for data from categorical or nominally scaled random variables. In order to apply the test to continuous numeric data, the data domain should be partioned into r categories first.

First we discus the χ^2 goodness of fit test. Here we assume to know from which distribution the data come. Then the H_0 and H_1 hypotheses can be stated as follows:

H_0**:** The sample comes from the distribution F_X

H_1**:** The sample does not come from the distribution F_X

Therefore the problem from example 1 can be solved with the help of the χ^2 goodness of fit test. Consequently, the H_0 and H_1 hypotheses are chosen as follows:

H_0**:** $P(X=1) = p_1 = \frac{1}{6}, \ldots, P(X=6) = p_6 = \frac{1}{6}$

H_1**:** $P(X=i) \neq \frac{1}{6}$ for at least one value $i \in \{1,\ldots,6\}$

Let $X_1,\ldots,X_n$ be i.i.d. continuous random variables and $x_1,\ldots,x_n$ the observations from these random variables. Then the test statistic is computed as follows

$$\chi^2 = \sum_{i=1}^{r} \frac{(O_i - E_i)^2}{E_i} \tag{2.20}$$

where O_i are the observed frequencies and E_i are the expected frequencies.

Since we are dealing with continuous random variables, to compute the observed and expected frequencies we should carry out a discretisation of the data domain.

Let $F_X(x)$ be the assumed cumulative distribution function. The x-axis have to be split into r pairwise disjoint sets or bin S_i. Then the expected frequency in bin S_i is given by

$$E_i = n\left(F_X\left(a_{i+1}\right) - F_X\left(a_i\right)\right) \tag{2.21}$$

where $[a_i, a_{i+1})$ is interval corresponding to bin S_i.

Furthermore, for the observed frequencies we obtain

$$O_i = \sum_{x_{k_i} \in S_i} 1. \tag{2.22}$$

O_i is therefore the amount of observations in the i-th interval.

The statistic (2.20) has an approximate χ^2-distribution with $(r-1)$ degrees of freedom under the following assumptions: First, the observations are independent from each other. Secondly, the categories – the bins S_i – are mutually exclusive and exhaustive. This means that no categories may have an expected frequency of zero, i.e. $\forall i \in 1,\ldots,r : E_i > 0$. Furthermore, no more than 20% of the categories should have an expected frequency less than five. If this is not the case, categories should be merged or redefined. Note that this might also lead to a different number of degrees of freedom.

Therefore, the hypothesis H_0 that the sample comes from the particular distribution F_X is rejected if

$$\sum_{i=1}^{r} \frac{(O_i - E_i)^2}{E_i} > \chi^2_{1-\alpha} \tag{2.23}$$

where $\chi^2_{1-\alpha}$ is the $(1-\alpha)$-quantile of the χ^2-distribution with $(r-1)$ degrees of freedom.

Table 2.1 summarizes the observed and expected frequencies and computations for example 1. All E_i are greater than zero, even greater than 4. Therefore,

Table 2.1: example 1

number i on the die	E_i	O_i	$\frac{(O_i-E_i)^2}{E_i}$
1	20	30	5
2	20	25	1.25
3	20	18	0.2
4	20	10	5
5	20	22	0.2
6	20	15	1.25

there is no need to combine categories. The test statistic is computed as follows:

$$\sum_{i=1}^{r} \frac{(O_i - E_i)^2}{E_i} = 5 + 1.25 + 0.2 + 5 + 0.2 + 1.25 = 12.9 \tag{2.24}$$

The obtained result $\chi^2 = 12.9$ should be evaluated with $(1-\alpha)$-quantile of the χ^2-distibution. For that purpose s. table of the χ^2-distribution ([11]) . The corresponding degrees of freedom are computed as explained above $(r-1) = (6-1) = 5$. For $\alpha = 0.05$ the tabled critical value for 5 degrees of freedom is $\chi^2_{0.95} = 11.07$, which is smaller than computed test statistic. Therefore the null hypothesis is rejected at the 0.05 significance level. For significance level 0.02 the critical value

is $\chi^2_{0.98} = 13.388$ and therefore the null hypothesis cannot be rejected at this level. This result can be summarized as follows: $\chi^2 = 12.9$ with 5 degrees of freedom can be rejected for all significance levels bigger than 0.024. This indicates that the die is unfair.

In order to adapt the χ^2 goodness of fit test to incremental calculation, the observed frequencies should be computed in an incremental fashion.

$$O_i^{(t)} = \begin{cases} O_i^{(t-1)} + 1 & \text{if } x_t \in S_i, \\ O_i^{(t-1)} & \text{otherwise.} \end{cases} \tag{2.25}$$

The expected frequency should also be recalculated corresponding to the increasing amount of observations.

$$E_i^{(t)} = \frac{E_i^{(t-1)}}{(t-1)} t. \tag{2.26}$$

Another very common test is the χ^2 independence test. This test evaluates the general hypothesis that two variables are statistically independent from each other.

Let X and Y be two random variables and $(x_1, y_1), \ldots, (x_n, y_n)$ are the observed values of these variables. For continuous random variables the data domains should be partitioned into r and q categories, respectively. Therefore the observed values of X can be assigned to one of the categories $S_1^X, \ldots, S_r^X$ and the observed values of Y to one of the categories $S_1^Y, \ldots, S_q^Y$. Then O_{ij} is the frequency of occurrence of the observation (x_{k_i}, y_{k_j}), where $x_{k_i} \in S_i^X$ and $y_{k_j} \in S_j^Y$. Furthermore

$$O_{i\bullet} = \sum_{j=1}^{q} O_{ij} \tag{2.27}$$

and

$$O_{\bullet j} = \sum_{i=1}^{r} O_{ij} \tag{2.28}$$

denote the marginal observed frequencies.

Table 2.2 illustrates the observed absolute frequencies. The total number of observations in the table is n. The notation O_{ij} represents the number of observations in the cell with index ij (i-th row and j-th column), $O_{i\bullet}$ the number of observations in the i-th row and $O_{\bullet j}$ the number of observations in the j-th column. This table is called contingency table.

Table 2.2: Contingency table

$X \setminus Y$	S_1^Y	$\dots$	S_j^Y	$\dots$	S_q^Y	marginal of X
S_1^X	O_{11}	$\dots$	O_{1j}	$\dots$	O_{1q}	$O_{1\bullet}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
S_i^X	O_{i1}	$\dots$	O_{ij}	$\dots$	O_{iq}	$O_{i\bullet}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
S_r^X	O_{r1}	$\dots$	O_{rj}	$\dots$	O_{rq}	$O_{r\bullet}$
marginal of Y	$O_{\bullet 1}$	$\dots$	$O_{\bullet j}$	$\dots$	$O_{\bullet q}$	n

It is assumed that the random variables X and Y are statistically independent. Let p_{ij} be the probability of being in the i-th category of the domain of X and the j-th category of the domain of Y. $p_{i\bullet}$ and $p_{\bullet j}$ are the corresponding marginal probabilities. Then, corresponding to the assumption of independence for each pair

$$p_{ij} = p_{i\bullet} \cdot p_{\bullet j} \tag{2.29}$$

holds. Equation (2.29) defines statistical independence. Therefore the null and the alternative hypothesis are as follows:

H_0**:** $p_{ij} = p_{i\bullet} \cdot p_{\bullet j}$

H_1**:** $p_{ij} \neq p_{i\bullet} \cdot p_{\bullet j}$

Thus, if X and Y are independent, then the expected absolute frequencies are given by

$$E_{ij} = \frac{O_{i\bullet} \cdot O_{\bullet j}}{n}. \tag{2.30}$$

The test statistic, again checking the observed frequencies against the expected frequencies under the null hpyothesis, is as follows.

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{q} \frac{\left(O_{ij} - E_{ij}\right)^2}{E_{ij}} \tag{2.31}$$

The test statistic has an approximate χ^2-distribution with $(r-1)(s-1)$ degrees of freedom. Consequently, the hypothesis H_0 that X and Y are independent can be rejected if

$$\sum_{i=1}^{r} \sum_{j=1}^{q} \frac{\left(O_{ij} - E_{ij}\right)^2}{E_{ij}} \geq \chi^2_{1-\alpha} \tag{2.32}$$

where $\chi^2_{1-\alpha}$ is the $(1-\alpha)$-quantile of the χ^2-distribution with $(r-1)(s-1)$ degrees of freedom.

For the incremental computation of $O_{i\bullet}$, $O_{\bullet j}$ and O_{ij} corresponding formulae must be developed. For the time point t and the new observed values (x_t, y_t) the incremental formulae are given by

$$O_{i\bullet}^{(t)} = \begin{cases} O_{i\bullet}^{(t-1)} + 1 & \text{if } x_t \in S_i^X, \\ O_{i\bullet}^{(t-1)} & \text{otherwise.} \end{cases} \tag{2.33}$$

$$O_{\bullet j}^{(t)} = \begin{cases} O_{\bullet j}^{(t-1)} + 1 & \text{if } y_t \in S_j^Y, \\ O_{\bullet j}^{(t-1)} & \text{otherwise.} \end{cases} \tag{2.34}$$

$$O_{ij}^{(t)} = \begin{cases} O_{ij}^{(t-1)} + 1 & \text{if } x_t \in S_i^X \wedge y_t \in S_j^Y, \\ O_{ij}^{(t-1)} & \text{otherwise.} \end{cases} \tag{2.35}$$

The χ^2 goodness of fit test can be extended to a χ^2 homogeneity test ([38]). Whereas the χ^2 goodness of fit test can be used only for a single sample, the χ^2 homogeneity test is used to compare whether two or more samples come from the same population.

Let $X_1, \ldots, X_m$ ($m \geq 2$) be discrete random variables, or continuous random variables discretised into r categories $S_1, \ldots, S_r$. The data for each of the m samples from random variables $X_1, \ldots, X_m$ (overall n values) are entered in a contingency table. This table is similar to the one for the χ^2 independence test.

Table 2.3: Contingency table

values \ variables	X_1	$\ldots$	X_j	$\ldots$	X_m	$\sum$
S_1	O_{11}	$\ldots$	O_{1j}	$\ldots$	O_{1m}	$O_{1\bullet}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
S_i	O_{i1}	$\ldots$	O_{ij}	$\ldots$	O_{im}	$O_{i\bullet}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
S_r	O_{r1}	$\ldots$	O_{rj}	$\ldots$	O_{rm}	$O_{r\bullet}$
$\sum$	$O_{\bullet 1}$	$\ldots$	$O_{\bullet j}$	$\ldots$	$O_{\bullet m}$	n

The samples are represented by the columns and the categories by the rows of Table 2.3. We assume that each of the samples is randomly drawn from the same distribution. The χ^2 homogeneity test checks whether m samples are homogeneous with respect to the observed frequencies. If the hypothesis H_0 is true, the expected frequency in the i-th category will be the same for all of the m random variables. Therefore, the null and the alternative hypothesis can be stated as follows:

H_0: $p_{ij} = p_{i\bullet} \cdot p_{\bullet j}$

H_1: $p_{ij} \neq p_{i\bullet} \cdot p_{\bullet j}$.

From H_0 follows that the rows are independent of the column.

Therefore, the computation of an expected frequency can be summarized by

$$E_{ij} = \frac{O_{i\bullet} \cdot O_{\bullet j}}{n}. \tag{2.36}$$

Although the χ^2 independence test and χ^2 homogeneity test evaluate different hypothesis, they are computed identically. Therefore, the incremental adaptation of the χ^2 independence test can also be applied to the χ^2 homogeneity test.

Commonly in case of two samples the Kolmogorov-Smirnov test is used, since it is an exact test and in contrast to the χ^2-test can be applied directly without previous discretisation of continuous distributions. However, the Kolmogorov-Smirnov test does not have any obvious incremental calculation scheme. The Kolmogorov-Smirnov test is described in Section 2.2.2.

The t-test

The next hypothesis test, for which we want to provide incremental computation is the t-test. Different kinds of the t-test are used. We restrict our considerations to the one sample t-test and the t-test for two independent samples with equal variance.

The one sample t-test evaluates whether a sample with particular mean could be drawn from the population with known expected value μ_0. Let $X_1, \ldots X_n$ be i.i.d. and $X_i \sim N\left(\mu; \sigma^2\right)$ with unknown variance σ^2. The null and the alternative hypothesis for two sided test are:

H_0: $\mu = \mu_0$, the sample comes from the normal distribution with expected value μ_0.

H_1: $\mu \neq \mu_0$, the sample comes from a normal distribution with an expected value differing from μ_0.

The test statistic is given by

$$T = \sqrt{n}\frac{\bar{X} - \mu}{S} \tag{2.37}$$

where $\bar{X}$ is the sample mean and S the sample standard deviation. The statistic (2.37) is t-distributed with $(n-1)$ degrees of freedom. H_0 is rejected if

$$t < -t_{1-\alpha/2} \text{ or } t > t_{1-\alpha/2} \tag{2.38}$$

where $t_{1-\alpha/2}$ is the $(1-\alpha/2)$-quantile of the t-distribution with $(n-1)$ degrees of freedom and t is the computed value of the test statistic (2.37), i.e. $t = \sqrt{n}\frac{\bar{x}-\mu_0}{s}$.

One-sided tests are given by the following null and alternative hypotheses:

H_0**:** $\mu \leq \mu_0$ and $H_1: \ \mu > \mu_0$. H_0 is rejected if $t > t_{1-\alpha}$.

H_0**:** $\mu \geq \mu_0$ and $H_1: \ \mu < \mu_0$. H_0 is rejected if $t < -t_{1-\alpha}$.

This test can be very easily adapted to incremental computation. For this purpose the sample mean and the sample variance have to be updated as in Equations (2.2) and (2.6), respectively, as described in Section 2.1. Note that the degrees of freedom of the t-distribution should be updated in each step as well.

$$t_{n+1} = \sqrt{n+1}\frac{\bar{x}_{n+1} - \mu_0}{s_{n+1}} \tag{2.39}$$

Unlike previous notations we use here $n+1$ for the time point, since the letter t is already used for the computed test statistic. Furthermore, as mentioned above the $(1-\alpha/2)$-quantile of the t-distribution with n degrees of freedom should be used to evaluate the null hypothesis. However for $n \geq 30$, the quantiles of the standard normal distribution could be used as approximation of the quantiles of the t-distribution.

The t-test for two independent samples is used to evaluate whether two independent sample come from two normal distributions with the same expected value. The two sample means $\bar{x}$ and $\bar{y}$ are used to estimate the expected values μ_X and μ_Y of the underlying distributions. If the result of the test is significant, we assume that the samples come from two normal distributions with different expected values. Furthermore, we assume that the variances of the underlying distributions are unknown.

The t-test is based on the following assumptions:

- The samples are drawn randomly.
- The underlying distribution is a normal distribution.

- The variances of the underlying distributions are equal, i.e. $\sigma_X^2 = \sigma_Y^2$.

Let $X_1, \ldots X_{n_1}$ i.i.d. and $X_i \sim N\left(\mu_X; \sigma_X^2\right)$ and $Y_1, \ldots Y_{n_2}$ i.i.d. and $Y_i \sim N\left(\mu_Y; \sigma_Y^2\right)$ with unknow expected values and unknown variances and $\sigma_X^2 = \sigma_Y^2$.

The null and the alternative hypothesis can be defined as follows:

H_0**:** $\mu_X = \mu_Y$, the samples come from the same normal distribution.

H_1**:** $\mu_X \neq \mu_Y$, the samples come from normal distributions with different expected values.

In this case, a two-sided test is carried out, however similar to the one sample t-test also a one-sided test can be defined.

The test statistic is computed as follows.

$$T = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{(n_1-1)S_X^2 + (n_2-1)S_Y^2}{n_1+n_2-2}}} \sqrt{\frac{n_1 n_2}{n_1+n_2}} \tag{2.40}$$

where S_X^2 and S_Y^2 are the unbiased estimators for the variances of X and Y, respectively.

Equation (2.40) is a general equation for the t-test for two independent samples and can be used in both cases of equal and unequal sample sizes.

The statistic (2.40) has a t-distribution with $(n_1 + n_2 - 2)$ degrees of freedom.

Let

$$t = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{(n_1-1)s_X^2 + (n_2-1)s_Y^2}{n_1+n_2-2}}} \sqrt{\frac{n_1 n_2}{n_1+n_2}} \tag{2.41}$$

be the computed value of the statistic (2.40). Then the hypotesis H_0 that the samples come from the same normal distribution is rejected if

$$t < -t_{1-\alpha/2} \text{ or } t > t_{1-\alpha/2} \tag{2.42}$$

where $t_{1-\alpha/2}$ is the $(1-\alpha/2)$-quantile of the t-distribution with $(n_1 + n_2 - 2)$ degrees of freedom.

Similar to the one sample t-test, the t-test for two independent samples can be easily computed in an incremental fashion, since the sample means and the variance can be calculated in an incremental way. Here the degrees of freedom should also be updated with the new observed values.

Multiple Testing

Multiple testing refers to the application of number of tests simultaneously. Instead of a single null hypothesis, a tests for a set of null hypotheses $H_0, H_1, \ldots, H_n$ are considered. These null hypotheses do not have to exclude each other.

An example for multiple testing is a test whether m random variables $X_1, \ldots X_m$ are pairwise independent. This means, the null hypotheses are $H_{1,2}, \ldots, H_{1,m}, \ldots,$ $H_{m-1,m}$ where $H_{i,j}$ states that X_i and X_j are independent.

Multiple testing leads to the undesired effect of cumulating the α-error. The α-error α is the probability to reject the null hypothesis erroneously, given it is true. Choosing $\alpha = 0.05$ means that in 5% of the cases the null hypothesis would be rejected, although it is true. When k tests are applied to the same sample, then the error probability for each test is α. Under the assumption that the null hypotheses are all true and the tests are independent, the probability that at least one test will reject its null hypothesis erroneously is

$$\begin{aligned} P(\ell \geq 1) &= 1 - P(\ell = 0) && (2.43)\\ &= 1 - (1-\alpha)\cdot(1-\alpha)\ldots\cdot(1-\alpha) && (2.44)\\ &= 1 - (1-\alpha)^k. && (2.45) \end{aligned}$$

ℓ is the number of tests rejection the null hypothesis.

A variety of approaches have been proposed to handle the problem of cumulating the α-error. In the following, two common methods will be introduced shortly.

The simplest and most conservative method is Bonferroni correction [37]. When k null hypotheses are tested simultaneously and α is the desired overall α-error for all tests together, then the corrected α-error for each single test should be chosen as $\tilde{\alpha} = \frac{\alpha}{k}$. The justification for this correction is the inequality

$$P\left(\bigcup_i A_i\right) \leq \sum_i P(A_i). \tag{2.46}$$

For Bonferroni correction, A_i is the event that the null hypothesis H_i is rejected, although it is true. In this way, the probability that one or more of the tests rejects its corresponding null hypothesis is at most α. In order to guarantee the significance level α, each single test must be carried out with the corrected level $\tilde{\alpha}$.

Bonferroni correction is a very rough and conservative approximation for the true α-error. One of its disadvantages is that the corrected significance level $\tilde{\alpha}$ becomes very low, so that it becomes almost impossible to reject any of the null hypotheses.

The simple single step Bonferroni correction has been improved by Holm [19]. The Bonferroni-Holm method is a multi-step procedure in which the necessary corrections are carried out stepwise. This method usually yields larger corrected α-values than the simple Bonferroni correction.

When k hypotheses are tested simultaneously and the overall α-error for all tests is α, for each of the tests the corresponding p-value is computed based on the sample x and the p-values are sorted in ascending order.

$$p_{[1]}(x) \leq p_{[2]}(x) \leq \ldots \leq p_{[k]}(x) \tag{2.47}$$

The null hypotheses H_i are ordered in the same way.

$$H_{[1]}, H_{[2]}, \ldots, H_{[k]} \tag{2.48}$$

In the first step $H_{[1]}$ is tested by comparing $p_{[1]}$ with $\frac{\alpha}{k}$. If $p_{[1]} > \frac{\alpha}{k}$ holds, then $H_{[1]}$ and the other null hypotheses $H_{[2]}, \ldots, H_{[k]}$ are not rejected. The method terminates in this case. However, if $p_{[1]} \leq \frac{\alpha}{k}$ holds, $H_{[1]}$ is rejected and the next null hypothesis $H_{[2]}$ is tested by comparing the p-value $p_{[2]}$ and the corrected α-value $\frac{\alpha}{k-1}$. If $p_{[2]} > \frac{\alpha}{k-1}$ holds, $H_{[2]}$ and the remaining null hypotheses $H_{[3]}, \ldots, H_{[k]}$ are not rejected. If $p_{[2]} \leq \frac{\alpha}{k-1}$ holds, $H_{[2]}$ is rejected and the procedure continues with $H_{[3]}$ in the same way.

The Bonferroni-Holm method tests the hypotheses in the order of their p-values, starting with $H_{[1]}$. The corrected α_i-values $\frac{\alpha}{k}, \frac{\alpha}{k-1}, \ldots \alpha$ are increasing. Therefore, the Bonferroni-Holm method rejects at least those hypotheses that are also rejected by simple Bonferroni correction, but in general more hypotheses can be rejected.

2.2.2 Change detection strategies

Detecting changes in data streams has become a very important area of research in many application fields, such as stock market, web activities or sensors measurements, just to name a few. The main problem for change detection in data streams

is limited memory capacity. It is unrealistic to store the full history of the data stream. Therefore, efficient change detection strategies tailored to the data stream should be used. The main requirements for such approaches are: low computational costs, fast change detection and high accuracy. Moreover it is important to distinguish between true changes and false alarms. Abrupt changes as well as slow drift in the data generating process can occur. Therefore, a "good" algorithm should be able to detect both kinds of changes.

Various strategies are proposed to handle this problem, see for instance [17] for a detailed survey of change detection methods. Most of these approaches are based on time window techniques [4, 23]. Furthermore, several approaches are presented for evolving data streams as they are discussed in [22, 20, 12].

In this section, we introduce two types of change detection strategies: incremental computation and window technique based change detection. Furthermore we put the main focus on statistical tests. We assume to deal with numeric data streams. As already mentioned in the introduction, two types of change are identified: concept change and change of data distribution. We don't differentiate in this work between both of them, since the distribution of the target variable will be changed in both cases. As we will show in Section 3.3.6, the incremental quantile estimator iQPres from Chapter 3 can be used for change detection as well. By using iQPres for change detection in the data distribution, we assume that the median of the distribution changes with the time, however if this is not the case and only another parameter like the variance of the underlying distribution changes, other strategies for change detection should be used. Detailed information about iQPres as change detector is provided in the Chapter 3.

Statistical tests for change detection

The theory of hypothesis testing is the main background for change detection. Several algorithms for change detection are based on hypothesis tests.

Hypothesis tests could be applied to change detection in two different ways:

- Change detection trough incremental computation of the tests: by this approach the test is computed in an incremental fashion, for instance as it is explained in Section 2.2.1. Consequently the change can be detected if the test starts to yield different results as before.

- Window techniques: by this approach the data stream divided into time windows. A sliding window could be used as well as non-overlaping windows. In order to detect potential changes, we need either to compare data from an earlier window with data from newer one or to test only the new data (for instance, whether the data follows a known or assumed distribution). When the window size is not too large, it is not necessary to be able to compute the tests in an incremental fashion. Therefore, we are not restricted to tests that render themselves to incremental computations, but many other tests could be used. Hybrid approaches combining both techniques are also possible. Of course, window techniques with incremental computations within the window will lead to less memory consumptions and faster computations.

We will not give a detailed description for change detection based on incremental computation here, since the principles of these methods are explained in Section 2.2.1. However, the problem of multiple testing as discussed in Section 2.2.1 should be taken into account when a test is applied again and again over time. Even if the underlying distribution does not change over time, any test will erroneously reject the null hypothesis of no change in the long run if we only carry out the test often enough. Different approaches to solve this problem are presented in Section 2.2.1. Another problem of this approach is the "burden of old data". If a large amount of data has been analysed already and the change is not very drastic, it may happen that the change will be detected with large delay or not detected at all when a very large window is used. On that account it may be useful to reinitialise the test from time to time.

To detect changes with by window technique, we need to compare two samples of data and have to decide whether the hypothesis H_0 that they come from the same distribution is true.

First we will present a general meta-algorithm for change detection based on a window technique, without any specific fixed test. This algorithm is presented in Figure 2.3. The constant `step` specifies, after how many new values the change detection should checked again.

This approach follows an simple idea: when the data from two subwindows of W are judged as "distinct enough", the change is detected. Here "distinct enough" is specified by the selected statistical test for distribution change. In general, we

```
1   Initialise window W, i = 0
2   for each new x_t do
3       if i < step then
4           W ← W ∪ {x_t} (i.e., add x_t to the W)
5           W ← W \ w_0 (i.e., remove oldest element in W)
6           i = i + 1
7           if i = step then
8               i = 0
9               split W into W_0 and W_1
10              test W_0 and W_1 for change
11              if change detected then
12                  report change at time t
13              end if
14          end if
15      end if
2   end for
```

Figure 2.3: General scheme of a change detection algorithm based on time windows and statistical tests

assume the splitting of W into two subwindows of equal size. Nevertheless, any "valid" splitting can be used. Valid is meant in terms of the amount of data that is needed for the test to be reliable.

However, by a badly selected cut point the change can be detected with large delay as Figure 2.4 shows. The rightmost part indicates a change in the data

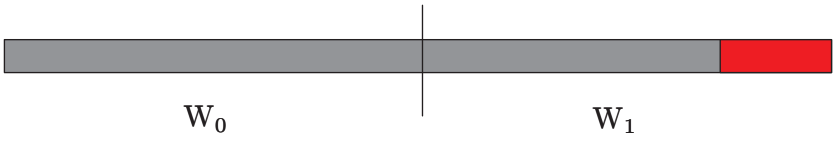

Figure 2.4: Subwindows problem

stream. As the change occurs almost at the end of the subwindow W_1, it is most likely that the change remains at first undetected. Of course, since the window will be moved forward with new data points arriving, at some point the change will be detected, but it may be from essential interest, to detect the change as early as possible.

To solve this problem, we modify the algorithm in Figure 2.3 in the following way: instead of splitting window W only once, the splitting is carried out several times. Figure 2.5 shows the modified part of the algorithm in Figure 2.3 starting at step 9.

How many times the window should be split, should be decided based on the required performance and precision of the algorithm. We can run the test for each sufficiently large subwindow of W, although the performance of the algorithm

```
9   for each valid split W = W_0 ∪ W_1 do
10      test W_0 and W_1 for change
11      if change detected then
12          report change at time t
13      end if
14
```

Figure 2.5: Modification of the algorithm for change detection to avoid the sub-windows problem

will decrease, or we can carry out fixed number of splits. Note that also for the windows technique based approach, attention should be paid to the problem of multiple testing (see Section 2.2.1). Furthermore, we do not specify here the effect of the detected change. The question whether the window should be reinitialised depends on the application. A change in the variance of the data stream might have a strong effect on the task to be fulfilled with the on-line analysis of the data stream or it might have no effect as long the mean value remains more or less stable.

For the hypothesis test in step 10 of the algorithm, any appropriate test for the distribution change can be chosen. Since we do not necessarily have to apply an incremental scheme for the hypothesis test, the Kolmogorov-Smirnov test can also be considered for change detection. The Kolmogorov-Smirnov test is designed to compare two distribution, whether they are equal or not. Therefore two kinds of questions could be answered with the help of the Kolmogorov-Smirnov test:

- Does the sample arise from a particular known distribution?
- Do two samples coming from different time windows have the same distribution?

We are particularly interested in the second question. For this purpose, the two sample Kolmogorov-Smirnov goodness-of-fit test should be used.

Let $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_m$ be two independent random samples from distributions with cumulative distribution functions F_X and F_Y, respectively. We want to test the hypothesis $H_0: F_X = F_Y$ against the hypothesis $H_1: F_X \neq F_Y$. The KolmogorovSmirnov statistic is given by

$$D_{n,m} = \sup_t |S_{X,n}(x) - S_{Y,m}(x)| \tag{2.49}$$

where $S_{X,n}(x)$ and $S_{Y,m}(x)$ are corresponding empirical cumulative distribution

function[3] of the first and second sample. H_0 is rejected at level α if

$$\sqrt{\frac{nm}{m+n}} D_{n,m} > K_\alpha \tag{2.51}$$

where K_α is the α-quantile of the Kolmogorov distribution.

To adapt the Kolmogorov-Smirnov test as a change detection algorithm, first the significance level α should be chosen (we can also use for instance the Bonferroni correction to avoid the multiple testing problem). The value of K_α needs either numerical computation or should be stored in a table[4]. Furthermore, values from the subwindows W_0 and W_1 represent two samples $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$. Then the empirical empirical cumulative distribution functions $S_{X,n}(x)$ and $S_{Y,m}(x)$ and the Kolmogorov-Smirnov statistic should be computed. Note that for the computation of $S_{X,n}(x)$ and $S_{Y,m}(x)$ in case of unique splitting the samples have to be sorted only initially, afterward the new values have to be inserted and the old values must be deleted from the sorted lists. In case of multiple splitting we have to decide either to sort each time from scratch or to save sorted lists for each kind of splitting.

An implementation of the Kolmogorov-Smirnov test is for instance available in the R statistics library (see [7] for more information).

Algorithm 2.5 based on the Kolmogorov-Smirnov test as the hypothesis test in step 10 has been implemented in Java using R-libraries and has been tested with artificial data. For the data generation process the following model was used:

$$Y_t = \sum_{i=1}^{t} X_i. \tag{2.52}$$

We assume the random variables X_i to be normally distributed with expected value $\mu = 0$ and variance σ_1^2, i.e. $X_i \sim N\left(0,\ \sigma_1^2\right)$. Here Y_t is a one dimensional random walk [40]. To make the situation more realistic, we consider the following model:

$$Z_t \sim N\left(y_t, \sigma_2^2\right). \tag{2.53}$$

[3]Let $x_{r_1}, x_{r_2}, \ldots x_{r_n}$ be a sample in ascending order from the random variables $X_1, \ldots, X_n$. Then the empirical distribution function of the sample is given by

$$S_{X,n}(x) = \begin{cases} 0 & \text{if} \quad x \leq x_{r_1}, \\ \frac{k}{n} & \text{if} \quad x_{r_k} < x \leq x_{r_{k+1}}, \\ 1 & \text{if} \quad x > x_{r_k}. \end{cases} \tag{2.50}$$

[4]This applies also to the t-test and the χ^2-test.

The process (2.53) can be understood as a constant model with drift and noise, the noise follows a normal distribution whose expected value equals the actual value of the random walk and whose variance is σ_2^2.

The data were generated with the following parameters: $\sigma_1 = 0.02$, $\sigma_2 = 0.1$. Therefore the data have a slow drift and are furthermore corrupted with noise.

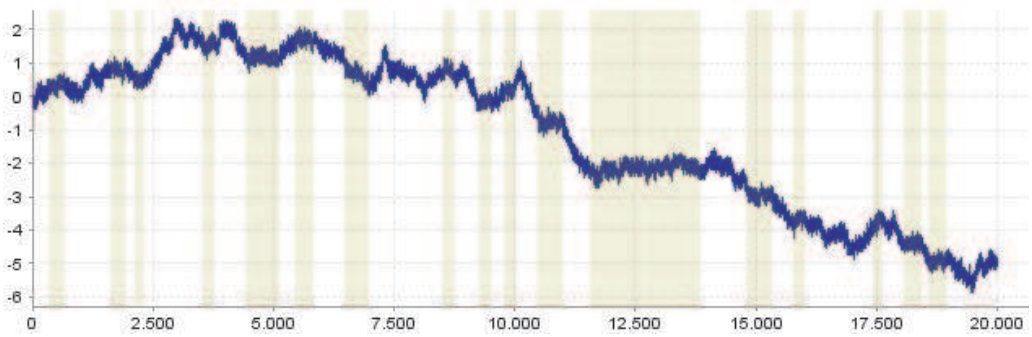

Figure 2.6: An example of change detection for the data generated by the process (2.53).

Algorithm 2.5 has been applied to this data set. The size of the window W was chosen to be 500. The window is always split into two subwindows of equal size, i.e. 250. The data are identified by the algorithm as non-stationary. Only very short sequences are considered to be stationbary by the Kolmogorov-Smirnov test. These sequences are marked by the darker areas in Figure 2.6. In the interval $[11445, 14414]$ stationary parts are mixed with occasionally occurring small non-stationary parts. For easier interpretation we joined these parts to one larger area. Of course, since we are dealing with the window, the real stationary areas are not exactly the same as shown in the figure. The quality of change detection depends on the window. For slow gradual changes in the form of concept drift a larger window is a better choice, whereas for abrupt changes in terms of a concept shift a smaller window is of advantage.

This chapter contained a brief introduction to incremental computation schemes for such statistical measures as: the mean, the variance, the third and fourth central moments and the Pearson correlation coefficient. Such indices provide valuable information about the probability distribution that generates the data stream. Also the problem of possible changes in the data was described in this chapter (see Section 2.2.2) and change detection methods based on hypothesis tests were introduced. Using statistical measures and tests for change detection can help to

discover true changes in the distribution and to distinguish them from random noise.

Incremental computations for the sample quantiles are not as obvious and easy as for the sample moments or Pearson correlation coefficient. On that account the theoretical consideration and corresponding computations are described in a separate chapter. We introduce two algorithms to the incremental computation of the sample quantiles and provide experimental results in the next chapter.

Chapter 3

Incremental Quantile Estimation

Quantiles[1] play an important role in statistics, especially in robust statistics, for instance the median as a robust measure of location and the interquartile range as a robust measure of spread. Incremental or recursive techniques for quantile estimation are not as obvious as for statistical moments. Nevertheless, there are techniques for incremental quantile estimation. However, they are either based on a restricted time window or only suitable for continuous random variables. In this chapter, we propose a more general approach which is not limited to continuous random variables. Our approach can be used for three purposes.

- As an incremental technique for quantile estimation when it is assumed that the underlying data generating process is not changing over time.
- For change detection. The Wilcoxon rank-sum test or MannWhitney U test [48, 29] are non-parametric hypothesis tests to compare two distributions based on the median. They are, however, not suitable as incremental methods. We provide an incremental test for change detection based on our incremental quantile estimation.
- For stabilised on-line adaptation. In combination with the proposed test for change detection, our algorithm can be used to adapt to changes over time in a more stable manner by running two estimations with an offset in parallel and deciding, based on the statistical test for change detection, whether to adapt to the more recent estimation.

[1] For a random variable X with cumulative distribution function F_X, the q-quantile ($q \in (0,1)$) is defined as $\inf\{x \in \mathbb{R} \mid F_X(x) \geq q\}$. If x_q is the q-quantile of a continuous random variable, this implies $P(X \leq x_q) = q$ and $P(X \geq x_q) = 1 - q$.

This chapter is organised as follows. We will briefly review related approaches in Section 3.1. As a preliminary step we need an algorithm for incremental median estimation which is described in Section 3.2. We extend this approach to general quantile estimation and analyse the algorithm in more detail in Section 3.3 where we also present a statistical test for change detection that can be directly incorporated into our algorithm to adapt the quantile estimation to non-stationary data streams. Experimental results are discussed in Section 3.4.

3.1 Related work

Using a time window is a very common way for on-line quantile estimation [14, 35]. However, such approaches will only take a subset of the available data into account and are more useful in a context where the sampled random variable constantly changes over time. A fast update algorithm for on-line calculation of the $\mathscr{Q}_n$ scale estimator is presented in [33]. The $\mathscr{Q}_n$ scale estimator (3.1) allows robust analysis of time series in real time and uses a moving time window for quantile estimation.

$$\mathscr{Q}_n(x_1,\ldots,x_n) = c_n \cdot \left\{ \left| x_i - x_j \right| : \; 1 \le i < j \le n \right\}_{(l)}, \; l = \binom{\left\lfloor \frac{n}{2} \right\rfloor + 1}{2} \qquad (3.1)$$

c_n is a correction factor. The $\mathscr{Q}_n$ estimator is applied at each time t to a time window of length n ($n \le N$). Instead of calculating $\mathscr{Q}_n$ for each window from scratch, an update algorithm is provided. Therefore for each move of the window from time point t to time point $t+1$ all stored information concerning the oldest observation x_{t-n+1} is deleted and new information concerning the incoming observation x_t is inserted. To allow on-line computation, fast insertion and deletion are needed. To achieve this, balanced trees as the main data structure are used.

Concerning computational complexity, a linear algorithm to find the k-th smallest element in an array of N values is described in [1]. However, this algorithm needs a large memory, its space complexity is $N \cdot (N-1)/2$ in the worst case.

Algorithms based on a time window automatically forget older information, which might be desired in some cases. But in some applications neither a suitable length of the time window is known nor is it suitable to simply ignore older measurements. Therefore, we concentrate on an incremental scheme for quantile estimation. Our proposed algorithm is not evolving in sense that it does not adapt

or track changes directly. But it is able to detect changes and to then restart the estimation of the quantiles. Thus, our algorithm is better suited for environments with state changes, but not meant to track constant changes in the form of drift. Of course, our algorithm would notice changes according to drift as well, but it would need to be restarted again and again to compensate the drift.

For continuous random variables, there is already an incremental scheme for quantile estimation based on the following theorem.

Theorem. Let $\{\xi_t\}_{t=0,1,\ldots}$ be a sequence of identically distributed independent (i.i.d.) random variables with cumulative distribution function F_ξ. Assume that the Lebesgue density function $f_\xi(x)$ exists and is continuous in the α-quantile x_α for an arbitrarily chosen α ($0 < \alpha < 1$). Further let the inequality

$$f_\xi(x_\alpha) > 0 \tag{3.2}$$

be fulfilled. Let $\{c_t\}_{t=0,1,\ldots}$ be a (control) sequence of real numbers satisfying the conditions

$$\sum_{t=0}^{\infty} c_t = \infty, \quad \sum_{t=0}^{\infty} c_t^2 < \infty. \tag{3.3}$$

Then the stochastic process X_t defined by

$$X_0 = \xi_0(\omega), \tag{3.4}$$

$$X_{t+1} = X_t + c_t Y_{t+1}(X_t, \xi_{t+1}(\omega)), \tag{3.5}$$

with

$$Y_{t+1} = \begin{cases} \alpha - 1 & \text{if } \xi_{t+1}(\omega) < X_t, \\ \alpha & \text{if } \xi_{t+1}(\omega) \geq X_t, \end{cases} \tag{3.6}$$

almost surely converges to the quantile x_α.

The proof of the theorem is based on stochastic approximation and can be found in [32]. A standard choice of the sequence $\{c_t\}_{t=0,1,\ldots}$ is $c_t = 1/t$. However, convergence might be extremely slow for certain distributions. Therefore, techniques to choose a suitable sequence $\{c_t\}_{t=0,1,\ldots}$, for instance based on an estimation of the probability density function of the sampled random variable, are proposed in [30, 15].

Although this technique of incremental quantile estimation has only minimum memory requirement, it has certain disadvantages.

- It is only suitable for continuous random variables.
- Unless the sequence $\{c_t\}_{t=0,1,\ldots}$ is well chosen, convergence can be extremely slow.
- When the sampled random variable changes over time, especially when the c_t are already close to zero, the incremental estimation of the quantile will remain almost constant and the change will be unnoticed.

In the following we propose a new algorithm to overcome these problems.

3.2 Incremental median estimation

Before we discuss the general problem of incremental quantile estimation, we first focus on the special of case of the median, since we will need the results for the median to develop suitable methods for arbitrary quantiles.

For the rest of the chapter x_i ($i = 1,2,3\ldots$) denotes the data set or data stream to be considered.

The median and the mean are measures of location for a distribution. The mean renders itself easily to incremental computation. Let

$$\hat{\mu}_t \;=\; \frac{1}{t}\sum_{i=1}^{t} x_i$$

denote the mean of the first t data points. Then the recursive formula

$$\hat{\mu}_{t+1} \;=\; \frac{t}{t+1}\hat{\mu}_t + \frac{1}{t+1}x_{t+1}. \tag{3.7}$$

yields the mean for the first $(t+1)$ data points. Equation (3.7) requires only the knowledge – or in terms of computation, the storage – of two values, the previous mean $\hat{\mu}_t$ and the number t of data taken into account so far, in order to compute $\hat{\mu}_{t+1}$. Of course, the latest data point x_{t+1} must be known as well for the computation of $\hat{\mu}_{t+1}$, but there is no need to store this value, only the two values $\hat{\mu}_t$ and t need to be stored and updated. Similar schemes can be easily developed not only for the mean, but for higher (empirical) moments or related concepts like the (empirical) variance. Many other statistical estimators can be reformulated for incremental computation in a similar way, for example the recursive least squares technique (see for instance [18]) or the recursive version of linear discriminant analysis [34].

Although only one or two exact values (the one or two values in the middle of the ordered data, depending on whether the number of data is odd or even) are needed to calculate the median, it is required to order the data in advance to determine the respective values. Since each time a new data point is added to the data, it is possible for all data points to change their position in the ordered data set. Therefore, in principle all data points must be known or stored for the stepwise computation of the median. The idea of our algorithm is to store only a limited number of exact data values, i.e. values around the median, and to count only the number of data points lying outside an interval around the median. Unfortunately, we do not know the true median and it might turn out that the true median lies outside the interval in which we have stored the exact values. We can, however, compute the probability that this will happen and our algorithm will fail. In this sense, we only provide a probabilistic algorithm which guarantees the correct result only with a certain (very high) probability. After introducing the algorithm in detail, we will also compute the failure probability of our algorithm.

For the incremental computation of the median we store a fixed number, a buffer of m sorted data values $a_1, \ldots, a_m$ in the ideal case the $\frac{m}{2}$ closest values left and the $\frac{m}{2}$ closest values right of the median, so that the interval $[a_1, a_m]$ contains the median. We also need two counters L and R to store the number of values outside the interval $[a_1, a_m]$, counting the values left and right of the interval separately. Initially, L and R are set to zero.

The algorithm works as follows. The first m data points $x_1, \ldots, x_m$ are used to fill the buffer. They are entered into the buffer in increasing order, i.e. $a_i = x_{[i]}$ where $x_{[1]} \leq \ldots \leq x_{[m]}$ are the sorted values $x_1, \ldots, x_m$. After the buffer is filled, the algorithm handles the incoming values x_t in the following way.

(a) If $x_t < a_1$, i.e. the new value lies left of the interval supposed to contain the median, then $L^{\text{new}} := L^{\text{old}} + 1$.

(b) If $x_t > a_m$, i.e. the new value lies right of the interval supposed to contain the median, then $R^{\text{new}} := R^{\text{old}} + 1$.

(c) If $a_i \leq x_t \leq a_{i+1}$ $(1 \leq i < m)$, x_t is entered into the buffer at position a_i or a_{i+1}. Of course, the other values have to be shifted accordingly and the old left bound a_1 or the old right bound a_m will be dropped. Since in the

ideal case, the median is the value in the middle of the buffer, the algorithm tries to achieve this by balancing the number of values left and right of the interval $[a_1, a_m]$. Therefore, the following rule is applied:

(c1) If $L < R$, then remove a_1, increase L, i.e. $L^{\text{new}} := L^{\text{old}} + 1$, shift the values $a_2, \ldots, a_i$ one position to the left and enter x_t in a_i.

(c2) Otherwise remove a_m, increase R, i.e. $R^{\text{new}} := R^{\text{old}} + 1$, shift the values $a_{i+1}, \ldots, a_{m-1}$ one position to the right and enter x_t in a_{i+1}.

Table 3.1: A small example data set

t	1	2	3	4	5	6	7	8	9
data	3.8	5.2	6.1	4.2	7.5	6.3	5.4	5.9	3.9

Table 3.2 illustrates how this algorithm works with an extremely small buffer of size $m = 4$ based on the data set given in Table 3.1.

Table 3.2: The development of the buffer and the two counters for the small example data set in Table 3.1

t	L	a_1	a_2	a_3	a_4	R
4	0	3.8	4.2	5.2	6.1	0
5	0	3.8	4.2	5.2	6.1	1
6	0	3.8	4.2	5.2	6.1	2
7	1	4.2	5.2	5.4	6.1	2
8	2	5.2	5.4	5.9	6.1	2
9	3	5.2	5.4	5.9	6.1	2

In each step, the median $\hat{q}_{0.5}$ can be easily calculated from the given values in the buffer and the counters L and R by

$$\hat{q}_{0.5} = \begin{cases} a_{\frac{L+m+R}{2}-L} & \text{if } t \text{ is odd,} \\ \frac{a_{\frac{L+m+R-1}{2}-L} + a_{\frac{L+m+R+1}{2}-L}}{2} & \text{if } t \text{ is even.} \end{cases} \tag{3.8}$$

It should be noted that it can happen that at least one of the indices $\frac{L+m+R}{2} - L$, $\frac{L+m+R-1}{2} - L$ and $\frac{L+m+R+1}{2} - L$ are not within the bounds $1, \ldots, m$ of the buffer indices and the computation of the median fails. The interval length $a_m - a_1$ can only decrease and at least for continuous distributions X with probability density function $f_X(q_{0.5}) > 0$, where $q_{0.5}$ is the true median of X, it will tend to zero with increasing sample size. In an ideal situation for our algorithm, the buffer of m

stored values contains exactly the values in the middle of the sample. Here we assume that at this point in time the sample consists of $m+t$ values. A detailed analysis of our algorithm is only necessary when more than m values are contained in the sample, since the buffer will contain the full sample as long as it contains not more than m values.

Our algorithm tries to maintain the ideal situation of having the true median (of the data) in the middle of the buffer by replacing values from the buffer in such a way that the counters for the values left and right of the buffer are approximately equal. However, replacement of values in the buffer takes only place when a new sample value falls into the buffer. As mentioned before, at least for continuous distributions the interval defined by the buffer tends to have length zero with increasing sample size, so that replacements take place seldomly in later steps of the algorithm. Nevertheless, the algorithm will still tend to roughly maintain the balanced situation. When the interval defined by the buffer has (almost) length zero and contains the true median, then roughly 50% of the sampled values will lie left, respectively right of the interval boundaries. Our algorithm fails when the situation becomes so unbalanced that the difference between the two counters for the values beyond the left and the right boundary of the interval reaches m. Figure 3.1 shows such an unbalanced situation with too many values left of the buffer values.

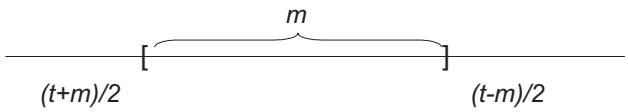

Figure 3.1: A situation when the algorithm fails

Assuming the extreme case that the interval defined by the buffer has already length zero and contains the true median, the probability that a newly sampled value lies on the left- or right-hand side of the interval is 0.5. The probability that among $(m+t)$ sampled values the difference between the number of values left and right of the median is at least m is

$$p_{\text{fail}} = 2 \cdot \sum_{i=0}^{\lceil \frac{t-m}{2} \rceil} \binom{t}{i} \cdot \left(\frac{1}{2}\right)^t. \tag{3.9}$$

Note that this provides only an approximation for the probability that our algorithm fails. On the one hand, we have made the pessimistic assumption that the

interval has reached length zero immediately. On the other hand we have made the optimistic assumption that our interval contains the true median and we have only computed the probability for an unbalanced failure situation after $(t+m)$ values are sampled. Even if we have the ideal situation of the median being exactly in the middle of the buffer after $(t+m)$ steps, an extremely unbalanced situation might have occurred in earlier steps. Tables 3.3 and 3.4 show corresponding failure probabilities for our algorithm with different choices for the buffer size m according to equation (3.9).

Table 3.3: Approximate failure probability for the median computation

$t \backslash m$	100	110	120	130	140	150
1000	0.0017	0.0006	0.0002	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$
2000	0.0268	0.0148	0.0078	0.0039	0.0019	0.0009
3000	0.0707	0.0466	0.0298	0.0185	0.0112	0.0065
4000	0.1175	0.0848	0.0599	0.0414	0.0280	0.0185
5000	0.1615	0.1232	0.0924	0.0681	0.0493	0.0351
6000	0.2012	0.1594	0.1245	0.0958	0.0727	0.0544
7000	0.2367	0.1926	0.1549	0.1231	0.0966	0.0749
8000	0.2684	0.2230	0.1834	0.1492	0.1202	0.0957
9000	0.2967	0.2506	0.2097	0.1739	0.1429	0.1163
10000	0.3222	0.2757	0.2340	0.1971	0.1645	0.1362
11000	0.3452	0.2987	0.2565	0.2187	0.1851	0.1554

Table 3.4: Approximate failure probability for the median computation

$t \backslash m$	160	170	180	190	200
1000	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$
2000	0.0004	0.0002	0.0001	$< 10^{-4}$	$< 10^{-4}$
3000	0.0037	0.0020	0.0011	0.0006	0.0003
4000	0.0119	0.0075	0.0047	0.0028	0.0017
5000	0.0245	0.0168	0.0114	0.0075	0.0049
6000	0.0401	0.0291	0.0208	0.0147	0.0102
7000	0.0574	0.0434	0.0324	0.0239	0.0174
8000	0.0755	0.0588	0.0454	0.0346	0.0261
9000	0.0937	0.0748	0.0592	0.0463	0.0359
10000	0.1118	0.0910	0.0735	0.0588	0.0466
11000	0.1295	0.1071	0.0879	0.0715	0.0578

3.3 Incremental quantile estimation

In this section we generalise and modify the incremental median algorithm proposed in the previous section and analyse the algorithm in more detail.

3.3.1 An ad hoc algorithm

The algorithm for incremental median estimation described in the previous section can be generalised to arbitrary quantiles in a straight forward manner. For the incremental q-quantile estimation ($0 < q < 1$) only case (c) requires a modification. Instead of trying to get the same values for the counters L and R, we now try to balance the counters in such a way that $qR \approx (1-q)L$ holds. This means, step (c1) is applied if $L < (1-q)t$ holds, otherwise step (c2) is carried out. t is the number of data sampled after the buffer of length m has been filled.

Therefore, in the ideal case, when we achieve this balance, a proportion of q of the data points lies left and a proportion of $(1-q)$ lies right of the interval defined by the buffer of length m.

In the case of an arbitrary q-quantile, the approximation of the probability for the algorithm to fail becomes

$$p_{\text{fail}} = \sum_{i=0}^{\lceil (t+m)\cdot q \rceil - m} \binom{t}{i} \cdot q^i \cdot (1-q)^{t-i} + \sum_{i=0}^{t-\lceil (t+m)\cdot q \rceil} \binom{t}{i} \cdot (1-q)^i \cdot q^{t-i}. \quad (3.10)$$

The formula (3.9) for the median failure probability is a special case of (3.10) with $q = 0.5$.

Tables 3.5 and 3.6 show these failure probabilities according to equation (3.10) for the 10%-quantile for different values of m.

Unfortunately, these probabilities are much larger than those in Tables 3.3 and 3.4 for the median and they also decrease much slower with increasing buffer size m.

Furthermore we are interested in the properties of the incremental quantile estimator presented above. Since we are simply selecting the k-th order statistic of the sample, at least for continuous random variables and larger presampling sizes, we can provide an asymptotic distribution of the order statistic and therefore for the estimator.

Assume, the sample comes from a continuous random variable X and we are interested in an estimation of the q-quantile x_q. Assume furthermore that the prob-

Table 3.5: Approximate failure probability for the simple q-quantile estimation ($q = 0.1$)

$t \backslash m$	100	110	120	130	140	150
1000	0.1583	0.1342	0.1127	0.0938	0.0774	0.0632
2000	0.2394	0.2169	0.1957	0.1757	0.1572	0.1399
3000	0.2816	0.2614	0.2420	0.2234	0.2057	0.1888
4000	0.3083	0.2900	0.2722	0.2550	0.2384	0.2224
5000	0.3271	0.3103	0.2939	0.2778	0.2623	0.2471
6000	0.3414	0.3257	0.3103	0.2953	0.2806	0.2663
7000	0.3527	0.3379	0.3234	0.3092	0.2953	0.2817
8000	0.3621	0.3479	0.3341	0.3207	0.3074	0.2945
9000	0.3701	0.3564	0.3432	0.3303	0.3176	0.3052
10000	0.3772	0.3637	0.3509	0.3385	0.3264	0.3144
11000	0.3836	0.3702	0.3577	0.3457	0.3340	0.3225

Table 3.6: Approximate failure probability for the simple q-quantile estimation ($q = 0.1$)

$t \backslash m$	160	170	180	190	200
1000	0.0511	0.0410	0.0325	0.0256	0.0199
2000	0.1240	0.1094	0.0961	0.0840	0.0731
3000	0.1728	0.1577	0.1434	0.1301	0.1177
4000	0.2070	0.1923	0.1782	0.1648	0.1520
5000	0.2325	0.2183	0.2047	0.1916	0.1790
6000	0.2524	0.2388	0.2257	0.2130	0.2007
7000	0.2684	0.2555	0.2428	0.2305	0.2186
8000	0.2817	0.2693	0.2571	0.2453	0.2337
9000	0.2930	0.2810	0.2693	0.2578	0.2466
10000	0.3027	0.2912	0.2798	0.2687	0.2578
11000	0.3111	0.3000	0.2890	0.2783	0.2677

ability density function f_X is continuous and positive at x_q. Let ξ_k^t ($k = \lfloor tq \rfloor + 1$) denote the k-th order statistic from an i.i.d. sample. Then ξ_k^t has an asymptotic normal distribution [11]

$$N\left(x_q;\ \frac{1}{f\left(x_q\right)}\sqrt{\frac{q\left(1-q\right)}{t}}\right) \tag{3.11}$$

From Equation (3.11) we can obtain valuable information about the quantile estimator.

In order to have a more efficient and reliable estimator, we want the variance of (3.11) to be as small as possible. Under the assumption that we know the data distribution, we can compute the variance of ξ_k^t.

Let X be a random variable following a standard normal distribution and as-

sume we have a sample $x_1, \ldots, x_t$ of X, i.e. these values are realizations of the i.i.d. random variables $X_i \sim N(0,\ 1)$. We are interested in the median of X. According to Equation (3.11), the sample median $\xi^t_{\lfloor 0.5t \rfloor + 1}$ is follows asymptotically a normal distribution:

$$\xi^t_{\lfloor 0.5t \rfloor + 1} \sim N\left(0;\ \sqrt{\frac{\pi}{2t}}\right). \tag{3.12}$$

Figure 3.2 shows the variance of the order statistic $\xi^t_{\lfloor 0.5t \rfloor + 1}$ as a function in t when the chosen quantile is $q = 0.5$, i.e. the median, and the original distribution from which the sample comes is a standard normal distribution $N(0;1)$. The second curve in the Figure corresponds to the variance of the sample mean.

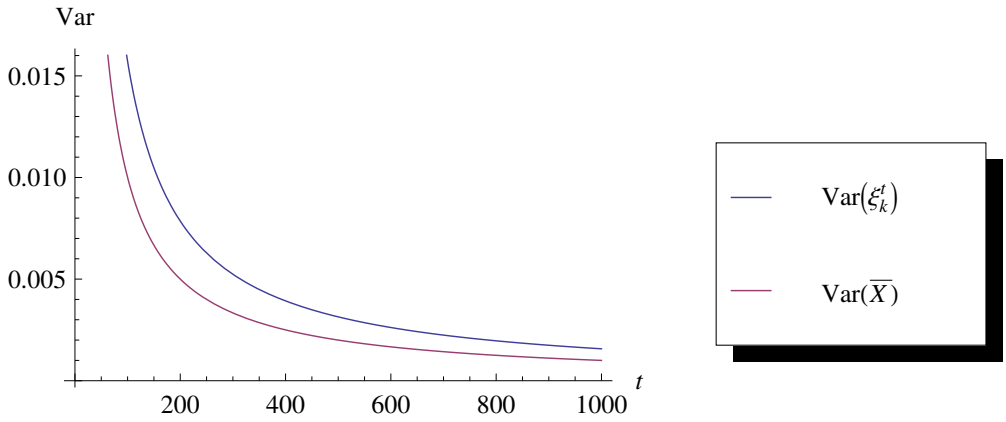

Figure 3.2: Variance from bottom to top of $\bar{X}$ and ξ^t_k under the assumption of a standard normal distribution of X

The variance of the sample mean $\bar{X}$ is only slightly better than that of the order statistic $\xi^t_{\lfloor 0.5t \rfloor + 1}$, nevertheless we should keep in mind the asymptotic character of the distribution (3.11).

Furthermore, from Equation (3.11) we obtain the other nice property of the incremental quantile estimator: It is an asymptotically unbiased estimator of sample quantiles. It is even a consistent estimator.

Unfortunately, as it was shown in Tables 3.5 and 3.6, the probability for the algorithm to fail is much smaller for the estimation of the median than for arbitrary quantiles. Therefore, despite the nice properties of this estimator this simple generalisation of the incremental median estimation algorithm to arbitrary quantiles is not very useful in practice. In order to amend this problem, we provide a modified algorithm based on presampling.

3.3.2 Incremental quantile estimation with presampling

Comparing Tables 3.3 and 3.4 with Tables 3.5 and 3.6 suggests that our proposed algorithm is much better suited for incremental median estimation than for other quantiles. Later in this section, we will provide a more rigorous explanation for this observation. In order to get back to the incremental median estimation for arbitrary quantiles, we introduce our new algorithm iQPres (incremental quantile estimation with presampling). Before we describe the modified algorithm for incremental quantile estimation, we illustrate the idea by a concrete example.

Assume we want to estimate the 10%-quantile. Instead of using the simple generalisation of our incremental median algorithm to the 10%-quantile, we apply presampling. We choose a number of n values, say $n = 21$, for presampling. This means we take groups of 21 values and for each group we estimate the 10%-quantile. In this case we would take the third smallest value for each presample of 21 values. This means that we now consider, instead of the original random variable X, the order statistic $X_{(3)}$ (for a sample size of 21). We could now simply estimate the median of $X_{(3)}$ and use this as an estimator for the 10%-quantile of X. However, the median of $X_{(3)}$ might be close to the 10%-quantile of X, but it will in general not be the same. The combination of the presampling idea with median estimation is not applicable to arbitrary quantiles, since the presampling size must be tailored to the quantile. Therefore, we still need another modification of this naïve presampling idea.

Assume we want to estimate the q-quantile. We presample n values and we simply take the l-th smallest value $x_{(l)}$ from the presample for some fixed $l \in \{1,\ldots,n\}$. At the moment, l does not even have to be related to the q-quantile. The probability that $x_{(l)}$ is smaller than the q-quantile of interest is

$$p_l = \sum_{i=0}^{l} \left(\begin{array}{c} n \\ i \end{array} \right) \cdot q^i \cdot (1-q)^{n-i}. \tag{3.13}$$

So when we apply presampling in this way, we obtain the new (presampled) distribution (order statistic) ξ_l^n. From equation (3.13) we can immediately see that the $(1-p_l)$-quantile of ξ_l^n is the same as the q-quantile of X. Therefore, instead of estimating the q-quantile of X, we estimate the $(1-p_l)$-quantile of ξ_l^n. Of course, this is only helpful, when l is chosen in such a way that the failure probabilities for the $(1-p_l)$-quantile are significantly lower than the failure probabilities for the

q-quantile. In order to achieve this, l should be chosen in such a way that $(1-p_l)$ is as close to 0.5 as possible as we will see later on. How the parameters n and l for the presampling procedure should be chosen, will be discussed after we have described the incremental quantile estimation algorithm with presampling.

We want to estimate the q-quantile ($0 < q < 1$). Fix the parameters m, l, n. (For an optimal choice see Subsection 3.3.4).

1. Presampling: n succeeding values are stored in increasing order in a buffer b_n of length n. Then we select the l-th element in the buffer. The buffer is emptied afterwards for the next presample of n values.

2. Estimation of the $(1-p_l)$-quantile based on the l-th element in the buffer for presampling: This is carried out according to the algorithm described in Subsection 3.3.1.

The quantile is then estimated in the usual way, i.e.

$$\begin{aligned} k &= \lceil (m+L+R)*(1-p_l)-l+0.5 \rceil, \\ r &= (m+L+R)*(1-p_l)-l+0.5-k, \\ \hat{q} &= (1-r)\cdot a_{k-R} + r \cdot a_{k-R+1} \qquad \text{(quantile estimator)} \end{aligned}$$

Of course, this does only work when the algorithm has not failed, i.e. the corresponding index k is within the buffer of m values.

It should be mentioned that the size of the buffer needed for presampling can be reduced to

$$\min\{l, n-l\}, \tag{3.14}$$

since we only need the lth-smallest element or, equivalently, the $(n-l-1)$th-largest element.

During a presampling step the buffer of this reduced size is first filled and the values are sorted in increasing order. When a further value has to be entered, the corresponding position of the value in the buffer is determined and the smallest element is deleted if $l > n-l$ holds. Otherwise, the largest element is deleted. After the presample is completed, the corresponding l-th value is the first value in the buffer, if $l > n-l$ holds, and the last value otherwise.

3.3.3 Complexity of the algorithm

The worst case complexity of our algorithm can be computed as follows. For the presampling, we need to sort n values which needs $O(n\log(n))$ steps. For a data stream of length t, this needs to be carried out t/n times. Entering the values derived from the presampling into the buffer of size m for the actual quantile estimation needs in the worst case $\log_2(m+2)$ comparisons and m values to be shifted. This also needs to be carried out t/n times. Therefore, the worst case complexity of our algorithm for a data stream of length t is

$$O\left(\left(t\cdot\log(n)+\frac{t}{n}\cdot\log_2(m+2)\right)\cdot\text{comparisons}+\frac{t}{n}\cdot m\cdot\text{shift\_operations}\right).$$

3.3.4 Choice of the parameters m, n and l

We assume that a fixed memory size M for the two buffers of length n (for presampling) and m (for the estimation of the quantile p_l based on presampling) is available. The goal is to find a pair $(m,n)\in\mathbb{N}^2$ such that p_{fail} is as small as possible under the constraint $M=m+n$. For a fixed pair (m,n), the integer $0<l\leq n$ is always chosen in such a way that p_l according to equation (3.13) is as close to 0.5 as possible. The parameters m,n,l are determined once in advance, before running the algorithm, by a brute force search strategy, i.e. by calculating the probability p_{fail} for all values $1<m<M$ and choosing the value for m that yields the smallest failure probabilty p_{fail}.

3.3.5 Justification for the choice of $p_l\approx 0.5$

The comparison of Tables 3.3 and 3.4 with Tables 3.5 and 3.6 already suggested that the failure probability for the incremental quantile estimation algorithm in Subsection 3.3.1 is lower for the median than for extreme quantiles. Figure 3.3 shows the failure probability (3.10) of the ad hoc algorithm from Subsection 3.3.1 for quantile estimation for $t=10000$ with a buffer length of $m=200$. With this parameter setting it is obvious that the estimation of the median yields the lowest failure probability.

The computation of the failure probability (3.10) is based on a binomial distribution with mean value $q\cdot t$ and variance $q\cdot(1-q)\cdot t$. Since t will always be a larger number here, this binomial distribution can be well approximated by a

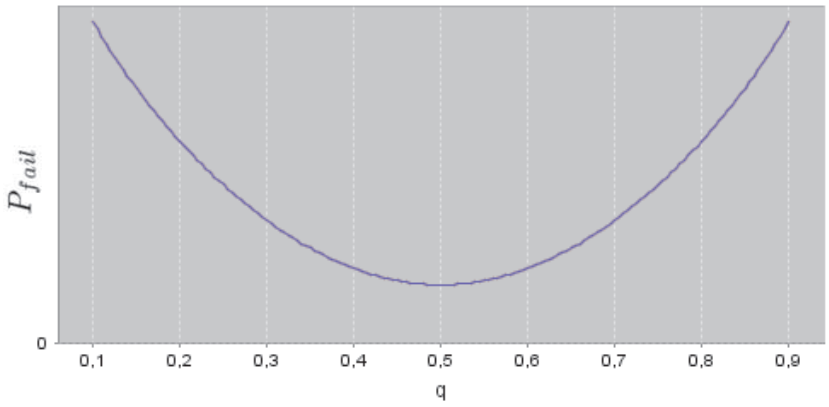

Figure 3.3: p_{fail} for $m = 200$, $t = 10000$

normal distribution. With this approximation, the failure probability becomes

$$p_{\text{fail}} \approx \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-\frac{m}{tq}} e^{-\frac{1}{2}x^2} dx + \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-\frac{m}{t(1-q)}} e^{-\frac{1}{2}x^2} dx \tag{3.15}$$

where we have already normalised the normal distribution to a standard normal distribution. In order to determine the minimum of (3.15), we take the derivative with respect to the parameter q, which yields

$$(p_{\text{fail}})'_q \approx \frac{1}{\sqrt{2\pi}} \left(\frac{m}{tq^2} e^{-\frac{1}{2}\left(\frac{m}{tq}\right)^2} - \frac{m}{t(1-q)^2} e^{-\frac{1}{2}\left(\frac{m}{t(1-q)}\right)^2} \right). \tag{3.16}$$

The right-hand side of (3.16) is zero for $q = 0.5$. The second derivative of (3.15) is

$$\begin{aligned} (p_{\text{fail}})''_q \approx \frac{m}{t\sqrt{2\pi}} \Bigg(& \left(\frac{m^2}{t^2 q^5} - \frac{2}{q^3} \right) e^{-\frac{1}{2}\left(\frac{m}{tq}\right)^2} \\ & + \left(\frac{m^2}{t^2 (1-q)^5} - \frac{2}{(1-q)^3} \right) e^{-\frac{1}{2}\left(\frac{m}{t(1-q)}\right)^2} \Bigg) \end{aligned} \tag{3.17}$$

For $q = 0.5$, the second derivative $(p_{\text{fail}})''_q$ is negative if $m < t\sqrt{\frac{1}{2}}$. Therefore, p_{fail} has a minimum at $q = 0.5$. It should be noted that choosing $m \geq t\sqrt{\frac{1}{2}}$ cannot really be considered as an incremental on-line algorithm, since the buffer size m is almost as large as the sample size t.

At least for continuous random variables with a unimodal symmetric distribution with a well pronounced maximum and larger presampling sizes, we can provide another argument in favour of the median estimation. As already mentioned in Section 3.3.1 for continuous random variable X and the estimation of the q-quantile x_q, k-th order statistic has an asymptotic normal distribution:

$$N\left(x_q;\ \frac{1}{f\left(x_q\right)}\sqrt{\frac{q\left(1-q\right)}{n}}\right)$$

(s. equation 3.11), here the probability density function f_X is continuous and positive at x_q and ξ_k^n ($k = \lfloor nq \rfloor + 1$) denote the k-th order statistic from an i.i.d. sample.

Therefore, in order to estimate the quantile x_q, we can also estimate the centre of this (asymptotic) normal distribution by the mean or – to be more robust – by the median. Of course, since n is limited by the available memory for the buffer, this estimator will only be asymptotically unbiased.

In order to have a more efficient and reliable estimator for the median of ξ_k^n, we want the variance of (3.11) to be as small as possible. In order to compute the value q for which the variance $v(q)$ of (3.11) has a minimum, we define

$$h\left(q\right) \ := \ \sqrt{\frac{q\left(1-q\right)}{n}}, \tag{3.18}$$

$$v\left(q\right) \ := \ \frac{h\left(q\right)}{f\left(x_q\right)}. \tag{3.19}$$

With

$$h'\left(q\right) \ = \ \frac{1}{2\sqrt{n}}\left(\frac{1-2q}{\sqrt{\left(1-q\right)q}}\right) \tag{3.20}$$

we obtain

$$\begin{aligned}\left(v\left(q\right)\right)' \ &= \ \left(\frac{h\left(q\right)}{f\left(x_q\right)}\right)' \ = \ \left(\frac{h\left(q\right)}{F'\left(x_q\right)}\right)' \ = \ \left(\frac{h\left(q\right)}{F'\left(F^{-1}\left(q\right)\right)}\right)' \\ &= \ \frac{h'\left(q\right)\cdot\left(F'\left(F^{-1}\left(q\right)\right)\right)^2 - F''\left(F^{-1}\left(q\right)\right)\cdot h\left(q\right)}{\left(F'\left(F^{-1}\left(q\right)\right)\right)^3}.\end{aligned} \tag{3.21}$$

Since we have assumed that the distribution of X is symmetric and unimodal, we have $F''\left(F^{-1}\left(0.5\right)\right) = 0$, i.e. the maximum of the probability density function is at the median which is also equal to the mean and the mode in this case. Furthermore, from (3.20) we can see that

$$\left(h\left(0.5\right)\right)' \ = \ 0 \tag{3.22}$$

holds. This implies that

$$\left(v\left(0.5\right)\right)' = 0 \tag{3.23}$$

holds for symmetric and unimodal distributions.

Therefore, the function $v(q)$ has an extreme value at 0.5. In order to show that this extreme value is a minimum, we consider the second derivative, taking (3.22) into account.

$$h''(q) = \frac{1}{4\sqrt{n}}\left(\frac{(1-2q)^2}{((1-q)q)^{\frac{3}{2}}}\right) - \frac{1}{\sqrt{n(1-q)q}} \tag{3.24}$$

$$\begin{aligned}(v(0.5))'' &= \frac{\left(h'(0.5)\cdot\left(F'\left(F^{-1}(0.5)\right)\right)^2 - F''\left(F^{-1}(0.5)\right)\cdot h(0.5)\right)'}{\left(F'\left(F^{-1}(0.5)\right)\right)^3} \\ &- \frac{0\cdot\left(\left(F'\left(F^{-1}(0.5)\right)\right)^3\right)'}{F'\left(F^{-1}(0.5)\right)^6} \\ &= \frac{h''(0.5)\cdot F'\left(F^{-1}(0.5)\right)^3 - h(0.5)\cdot F'''\left(F^{-1}(0.5)\right)}{\left(F\left(F^{-1}(0.5)\right)\right)^4}\end{aligned} \tag{3.25}$$

(3.19) has a local minimum at 0.5 when $v'(0.5) = 0$ and $v''(0.5) > 0$ hold. Therefore, we need (3.25) to be positive. From $h(0.5) = \frac{1}{2\sqrt{n}}$, $h''(0.5) = -\frac{2}{\sqrt{n}}$ and (3.25) we obtain that we require

$$-\frac{2}{\sqrt{n}}\cdot F'\left(F^{-1}(0.5)\right)^3 - \frac{1}{2\sqrt{n}}\cdot F'''\left(F^{-1}(0.5)\right) > 0 \tag{3.26}$$

or equivalently

$$-4 > \frac{F'''\left(F^{-1}(0.5)\right)}{F'\left(F^{-1}(0.5)\right)^3} = \frac{f''\left(F^{-1}(0.5)\right)}{f\left(F^{-1}(0.5)\right)^3}. \tag{3.27}$$

$f''\left(F^{-1}(0.5)\right)$ is always negative, since the probability density function of X has its maximum at $F^{-1}(0.5)$. Because $f'\left(F^{-1}(0.5)\right) = 0$, the numerator of the right-hand side of (3.27) is the (negative) curvature of f at its maximum, condition (3.27) is satisfied when the probability density function f has a clear and not plateau-like maximum.

In the case of a normal distribution, condition (3.27) becomes $-4 > -2\pi$ and is therefore satisfied. Figure 3.4 shows the deviation of the normal distribution (3.11) depending on the chosen quantile q when the original distribution from which the sample comes is a standard normal distribution $N(0,1)$.

3.3.6 Detecting changes

As already mentioned in the Chapter 2, algorithm iQPres could be used for change detection. Assuming that the sampled data come from the same distribution and

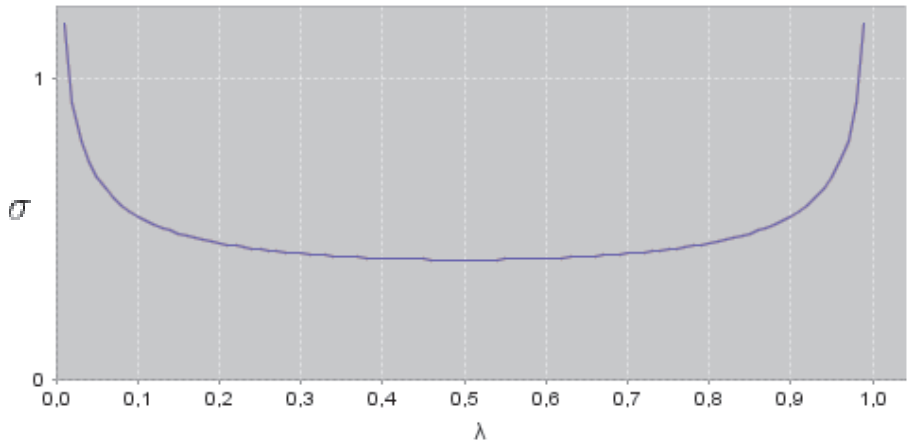

Figure 3.4: Deviation of the estimator depending on the chosen quantile $\lambda = q$.

are sampled independently, our algorithm might fail with probability (approximately) p_{fail}. As long as p_{fail} is small enough, this is not a serious disadvantage. In the case that we cannot be sure that the sampling distribution does not change over time, this is even an advantage. When p_{fail} is chosen small enough and the algorithm still fails, this can be seen as an indication for a change in the sampling distribution. The algorithm can even be used as a statistical hypothesis test for detecting changes (concerning the chosen quantile). In this case, one would specify the desired significance level for the test and the length of the considered stream of data and then choose the buffer sizes accordingly so that p_{fail} is (almost) equal to the significance level.

Table 3.7 shows results of simulations for different distribution where we have carried out the simulation until our algorithm "failed", i.e. until the counters left and right of our buffer where so unbalanced that the quantile to be estimated dropped out of the buffer. N(0,1) stands for a standard normal distribution, U(0,1) for a uniform distribution on the unit interval and Exp(0.25) for an exponential distribution with rate $\lambda = 0.25$. In all cases, the last estimation of the quantile before the overflow of the buffer was very precise already.

Table 3.7: Average number of steps until failure

Distribution	Quantile	n	m	Average no. of steps until failure
N(0,1)	0.8	53	247	64809
U(0,1)	0.25	134	166	30217
Exp(0.25)	0.5	51	249	67460

For longer data streams the failure probability of our algorithm would be too

high. It also does not very useful to refine the estimation of the corresponding quantile, when the length of the interval defined by the left and right value in the buffer is almost zero. We therefore recommend to use our algorithm in the following way as an algorithm for change detection. Depending on the required precision and the available memory size, we stop the estimation of the quantile after a fixed number of data values. We restart this estimation procedure for the quantile in regular intervals and compare the results of the different estimates for the purpose of change detection.

After two such estimates have been provided, we can derive a p-value for the null hypothesis that the estimated quantiles are the same, i.e. that the underlying distribution from which we sample has not changed. The two estimates provide counters L_i and R_i ($i \in \{1,2\}$) for the number of values left and right of the corresponding buffer. As an extremely simple, but illustrative example we take the results shown in the last line of Table 3.2 – the estimation after only 9 steps. For this first estimation, we have $L_1 = 3$ and $R_1 = 2$. Assume we have a second estimation after 9 steps with $L_2 = 1$ and $R_2 = 4$ and the entries $6.0, 6.2, 6.3, 6.5$ in the buffer. In this case, we might suspect a shift to the right of the quantile. But how can we test this? In terms of order statistics, we know the values $x_{(4)} = 5.2$, $x_{(5)} = 5.4$, $x_{(6)} = 5.9$, $x_{(7)} = 6.1$ for the first sample and the values $y_{(2)} = 6.0$, $y_{(3)} = 6.2$, $y_{(4)} = 6.3$, $y_{(5)} = 6.5$. In order to define a test for the null hypothesis, whether the (true) quantile has not changed for the two samples, we can compare these order statistics. In order to obtain a p-value for this test, we can simply ask how probable it is that for two samples the order statistic $x_{(7)}$ of the first sample yields a smaller value than the order statistic $y_{(3)}$ of the second sample.

More generally, we have the following problem to determine a suitable p-value. The two samples are generated by random variables $X_1, \ldots, X_K$ and $Y_1, \ldots, Y_N$, respectively. The null hypothesis assumes that all X_i and Y_i are independent and identically distributed. Let us assume here that we sample from a continuous random variable, so that all X_i and Y_i have the same cumulative distribution function F and probability density function f. (Note that in our case, we could assume equal samples sizes: $K = N$.)

We are interested in the probability

$$P(Y_{(\ell)} > X_{(r)})$$

that the order statistic $Y_{(\ell)}$ is larger than the order statistic $X_{(r)}$.

The likelihood for $X_{(r)} = x$ is

$$L(X_{(r)} = x) \;=\; K\binom{K-1}{r-1} f(x)F(x)^{r-1}(1-F(x))^{K-r}.$$

The reason for this is the following. One of the $X_1, \ldots, X_m$ must be equal to x. The likelihood for any of the X_i being equal to x is $f(x)$ and we have K possibilities to choose the corresponding i. This explains the factors K and $f(x)$. $(r-1)$ of the X_i must be lower than x. The Probability for each of them is $F(x)$ and we have $K-1$ positions left for these $(r-1)$ X_i. This contributes the factors $F(x)^{r-1}$ and $\binom{K-1}{r-1}$. The remaining X_i must be larger than x. The probability for each of them is $(1-F(x))$.

Now consider the probability that $Y_{(\ell)}$ is larger than x.

$$P(Y_{(\ell)} > x) \;=\; \sum_{i=1}^{\ell-1} \binom{N}{i} F(x)^i (1-F(x))^{N-i}.$$

This holds for the following reason. 1 or 2 or ... or $\ell-1$ of the Y_j can be smaller than x. This is where the sum comes from. The probability that exactly i of the Y_j are smaller than x is $\binom{N}{i}F(x)^i(1-F(x))^{N-i}$. There are i out of n positions to place these smaller Y_j. The probability for each of the i Y_j to be smaller than x is $F(x)$, the probability for each of the other $(N-i)$ Y_j to be larger than x is $(1-F(x))$.

Therefore, we have

$$P(Y_{(\ell)} > X_{(r)} \mid X_{(r)} = x) \;=\; P(Y_{(\ell)} = x) \;=\; \sum_{i=1}^{\ell-1} \binom{N}{i} F(x)^i (1-F(x))^{N-i}$$

which implies

$$\begin{aligned} P(Y_{(\ell)} > X_{(r)}) \;&=\; \int_{-\infty}^{\infty} P(Y_{(\ell)} > X_{(r)} \mid X_{(r)} = x) L(X_{(r)} = x)\, dx \\ &=\; K\binom{K-1}{r-1} \int_{-\infty}^{\infty} f(x)F(x)^{r-1}(1-F(x))^{K-r} \\ &\qquad \cdot \sum_{i=1}^{\ell-1} \binom{N}{i} F(x)^i (1-F(x))^{N-i}\, dx. \end{aligned}$$

The substitution $t = F(x)$ yields

$$P(Y_{(\ell)} > X_{(r)}) \;=\; K\binom{K-1}{r-1} \int_0^1 t^{r-1}(1-t)^{K-r} \sum_{i=1}^{\ell-1} \binom{N}{i} t^i (1-t)^{N-i}\, dt.$$

This is an integral over a polynomial independent of F. The solution of this integral is

$$\int_0^1 t^{r+i-1}(1-t)^{K+N-r-i}dt \;=\; \frac{(a-1)!(K+N-a)!}{(K+N)!}$$

where $r+i=a$. Therefore, we have

$$P(Y_{(\ell)} > X_{(r)}) \;=\; K\binom{K-1}{r-1}\sum_{i=1}^{\ell-1}\binom{N}{i}\frac{(r+i-1)!(K+N-r-i)!}{(K+N)!} \qquad (3.28)$$

Table 3.8: Selected values for the probability in equation (3.28).

N	K	r	l	$P(Y_{(\ell)} > X_{(r)})$
10	10	6	4	0.179507025
40	40	24	16	0.036452904
100	100	60	40	0.002277662
300	300	180	120	0.000000444

Table 3.8 shows that the values for this probability, i.e. for the p-value of interest, become small for larger values of K and N. Note that K and N are the numbers of samples that are taken for one estimation of the quantile. The buffer size can be significantly smaller. For instance, in the last line of Table 3.8, we might have a buffer size of only 100 and could still calculate the values $x_{(180)}$ and $y_{(120)}$ required for the probability in the table, since the buffer tries to store the 100 values in the middle out of he 300 values in the sample, when the values are ordered.

The choice of a suitable p-value might be a very difficult, since we have to face the problem of multiple test. For a very long data stream, we would initiate the estimation of the quantile again and again and carry out the test each time. For example, the test based on the line with $K = N = 100$ in Table 3.8 would wrongly reject the null hypothesis in average once in $1/0.002277662 \approx 440$ times. So if we repeat the test after $K = N = 100$ sampled data each time, we would the test would indicate a change of the quantile in average once in a data stream of length $440 \cdot 100 = 44000$, although the underlying distribution has not changed. Of course, one can correct the p-value according to the number of times the test should be applied. However, such a correction of the p-value for multiple testing [19, 37] would lead to extremely low p-values, so that even true changes would have an

extremely low probability to be detected. But this is not a specific problem of our algorithm. This a general problem for any statistical test for change detection.

Note that algorithms based on the theorem mentioned in Section 3.1 are not suitable for change detection.

3.3.7 Evolving environment

Our algorithm can be used as an incremental technique for quantile estimation under the assumption of a stationary data generating process. Together with the statistical test for discovering significant differences in the quantiles over time, it is also useful as an on-line algorithm for change detection. Based on this idea of change detection, we can also use it in an evolving environment with changing parameters of the underlying distribution. In this case, the algorithm would be applied in the following way.

For quantile estimation, it is not necessary to refine the estimation constantly on the basis of tens of thousands of sample values. Therefore, the incremental quantile estimation should stop after a fixed number of steps. After one estimation is finished, or even even earlier, a new incremental quantile estimation is started. Then the resulting estimations can be compared by the hypothesis test that has been described in the previous subsection. Only when the corresponding hypothesis test indicates that the change is significant from the statistical point of view, the estimated value for the quantile will be updated. In this way, we can even stabilise the estimation by avoiding permanent changes that are only caused by noise, but not due to parameter drift.

3.4 Experimental results

In this section we present an experimental evaluation of our proposed algorithm iQPres based on a artificial data sets as well as on real world data set from a waste water treatment plant.

First, we consider estimations of the lower and upper quartile as well as the median for different distributions:

- Exponential distribution with parameter $\lambda = 4$ (Exp(4))
- Standard normal distribution (N(0,1))

- Uniform distribution on the unit interval (U(0,1))

- An asymmetric bimodal distribution given by a Gaussian mixture model (GM) of two normal distributions. The cumulative distribution function of this distribution is given by

$$F(x) = 0.3 \cdot F_{\mathrm{N}(-3,1)} + 0.7 \cdot F_{\mathrm{N}(1,1)}$$

where $F_{\mathrm{N}(\mu,\sigma^2)}$ denotes the cumulative distribution function of the normal distribution with expected value μ and variance σ^2. Its probability density function is shown in Figure 3.5.

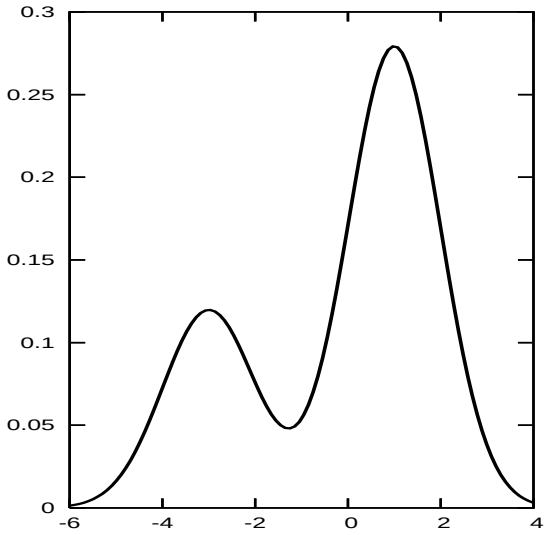

Figure 3.5: An example for an asymmetric, bimodal probability density function

The quantile estimations were carried out for samples of size of 10000 that were generated from these distributions. We have repeated each estimation 1000 times. Tables 3.9-3.11 show the average over all estimations for our algorithm (iQPres with a memory size of $M = 150$) and for the technique based on Theorem 3.5 where we used the control sequence $c_t = \frac{1}{t}$. The mean squared error over the 1000 repeated runs is also shown in the tables.

For the uniform distribution, incremental quantile estimation based on equation (3.5) and iQPres leads to very similar and good results. For the normal distribution, both algorithms yield quite good results, but iQPres seems to be slightly more efficient with a smaller mean square error. For the bimodal distribution

Table 3.9: Estimation of the lower quartile $q = 0.25$

Distr.	True quantile	iQPres	Equation 3.5	MSE (iQPres)	MSE (Equation 3.5)
Exp(4)	1.150728	1.152182	1.718059	2.130621E-5	2.675568
N(0,1)	-0.674490	-0.672235	-0.678989	5.611009E-6	0.008013
U(0,1)	0.250000	0.250885	0.250845	1.541123E-6	4.191695E-5
GM	-2.043442	-2.042703	0.185340	1.087618E-5	5.331730

Table 3.10: Estimation of the median $q = 0.5$

Distr.	True quantile	iQPres	Equation 3.5	MSE (iQPres)	MSE (Equation 3.5)
Exp(4)	2.772589	2.7462635	5.775925	7.485865E-4	10.906919
N(0,1)	0.000000	6.8324E-4	-0.047590	1.786715E-5	0.009726
U(0,1)	0.500000	0.495781	0.499955	1.779917E-5	2.529276E-6
GM	0.434425	0.434396	0.117499	2.365156E-6	0.451943

Table 3.11: Estimation of the upper quartile $q = 0.75$

Distr.	True quantile	iQPres	Equation 3.5	MSE (iQPres)	MSE (Equation 3.5)
Exp(4)	5.545177	5.554385	5.062660	1.054132E-4	0.919735
N(0,1)	0.674490	0.674840	0.656452	3.600748E-7	0.003732
U(0,1)	0.750000	0.750883	0.749919	8.443136E-7	2.068730E-5
GM	1.366114	1.366838	0.027163	1.193377E-6	2.207112

based on the Gaussian mixture model and a skewed distribution such as the exponential distribution, the estimations for the algorithm based on equation (3.5) are more or less useless, at least when no specific effort is invested to find an optimal control sequence $\{c_t\}_{t=0,1,\ldots}$. iQPres does not have any problems with these distributions. As already mentioned before, it is also not required for iQPres that the sampling distribution is continuous whereas it is a necessary assumption for the technique based on equation (3.5).

As mentioned before, another advantage of iQPres is that in case the sampling distribution changes, having a drift of the quantile to be estimated as a consequence, such changes will be noticed, since the simple version of iQPres without shifted parallel estimations will fail in the sense that it is not able to balance the counters L and R any more.

In order to illustrate how iQPres can be applied to change detection, we consider daily measurements for gas production in a waste water treatment plant over a period of more than eight years. The measurements are shown in Figure 3.6.

iQPress has been applied to this data set to estimate the median with a memory size $M = 30$. The optimal choice for the sizes of the buffers for presampling and

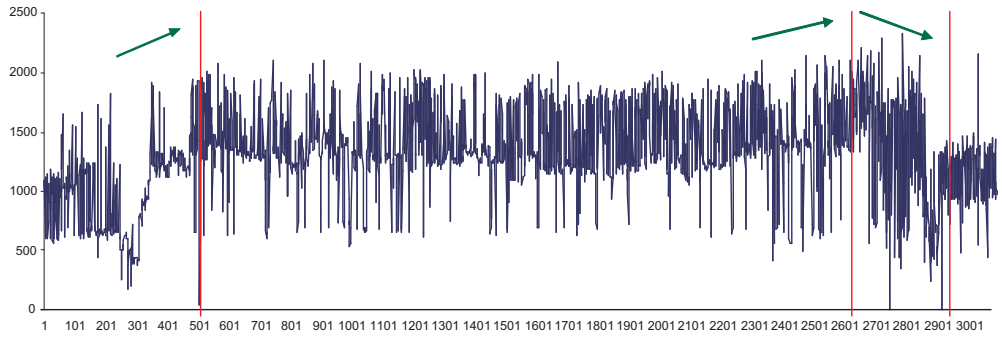

Figure 3.6: An example of median estimation for time series data from a waste water treatment plant

median estimation is then $n = 3$ and $m = 27$, respectively. At the three time points 508, 2604 and 2964, the buffer cannot be balanced anymore, indicating that the median has changed. These three time points are indicated by vertical lines in Figure 3.6. The arrows indicate whether the median is increased or decreased. An increase corresponds to an unbalanced buffer with the right counter R becoming too large, whereas a decrease leads to an unbalanced buffer with the left counter L becoming too large. The median increases at the first point at 508 from 998 before and 1361 after this point. At time point 2604 the median increases to 1406 and drops again to 1193 at time point 2964.

I this chapter we have proposed an algorithm for incremental or recursive sample quantile estimation for arbitrary distributions. Furthermore we compared our approach with already existing method. The experimental results have shown that also for continuous distributions our algorithm outperforms other approaches. The iQPres algorithm can be also applied for the change detection. For that purpose we have developed a statistical test that can be easily integrated into the algorithm. Our algorithm can also be used in evolving systems by repeating the quantile estimation at regular intervals, but accepting changes only when the they statistically significant based on the hypothesis test in order to avoid unnecessary changes and to stabilise the estimation procedure.

In the next chapters we analyse which effect has change and noise in the data on the quality of prediction and speak about the need of the benchmarks in evolving systems.

Chapter 4

Analysis of Effects of Noise and Changes in the Data in Evolving Systems Based on Simple Stochastic Models

Evolving systems are designed to cope with streaming data under the assumption of non-stationarity of the data generating process. Here "evolving" means developing and adaptation of the system corresponding to the current situation in the data. Consequently an ideal evolving system should be able:

- cope with huge amounts of data,
- process streaming data online and in real time,
- adapt itself fast to the changes in the data,
- be robust against the noise.

However as mentioned in the introduction, noise and changes in the data generating pose the problem for evolving systems. They must be able to distinguish between changes according to noise and changes of the underlying data generating process or its parameters. In the worst case, an evolving system might just try to track the noise in the data and is unable to learn the actual relationship inherent in the data. But how can we make sure that this will not be the case for a given evolving system?

In this chapter, we propose to set up simple theoretical models for the data generating process for which we can give at least a rough answer to the question, how well a model that takes assumptions on the data generating process into

account could perform. We can then compare how much worse the evolving system performs. Of course, we cannot expect the evolving system to have a similar performance, since it is not allowed to make specific assumptions about the data generating process. But this comparison will give us an idea, how much an evolving system can deviate from an optimal model and whether a simpler model that would not track the noise might not have a better performance.

We consider simple prediction tasks like classification and regression in this chapter. Evolving system have been proposed for such problems for instance in [2, 3, 5, 24, 28].

In section 4.1 we carry out a theoretical analysis between an extremely simplified evolving system based on a windowing technique and a model tailored to the known assumptions of the data generating process. This can be considered as a regression problem with the identity function as the regression function.

Section 4.2 introduces a simple switching model which can be interpreted as a regression or a classification problem. Here we carry out an experimental comparison between a maximum likelihood estimator exploiting the assumptions on the underlying data generating process and an evolving system without specific assumptions on the data generating process.

4.1 Random walk

A very simple example for systems with a random change over time is one dimensional random walks [40].

Definition 6 *A random walk $(Y_t)_{t\in\mathbf{N}}$ is obtained by adding up values from independent and identically-distributed (i.i.d.) random variables X_i with expected value zero, i.e. $E(X_i) = 0$ and variance $\sigma^2 = Var(X_i) = Var(X)$.*

$$Y_t = \sum_{i=1}^{t} X_i \tag{4.1}$$

The expected value for the random walk is then equal to zero:

$$E(Y_t) = E\left(\sum_{i=1}^{t} X_i\right) = \sum_{i=1}^{t} E(X_i) = 0 \tag{4.2}$$

Furthermore the expected value is independent of t, whereas the variance of the random walk increases linearly with t.

$$Var(Y_t) = Var\left(\sum_{i=1}^{t} X_i\right) = t \cdot Var(X) = t \cdot \sigma^2. \tag{4.3}$$

According to equations (4.1), (4.2) and (4.3), the random variables Y_t follow a normal distribution, i.e. $Y_t \sim N\left(0,\, t \cdot \sigma^2\right)$.

Furthermore the covariance of a random walk is given by

$$Cov(Y_t, Y_s) = s \cdot \sigma^2,\ (s < t) \tag{4.4}$$

and also tends to infinity with increasing difference between the time points t and s.

The best prediction that one can do for a random walk is the naïve approach, simply using the last value is a prediction for the next value. $\hat{y}_{t+1} = y_t$. Therefore, the difference between the true value and the predicted value is $Y_{t+1} - Y_t = X_{t+1}$. Thus, the expected quadratic error is

$$E\left((Y_{t+1} - Y_t)^2\right) = E\left((X_{t+1})^2\right) = Var(X). \tag{4.5}$$

Now assume, we do not know that we deal with a random walk and try an extremely simple "evolving system" using a windowing technique with a window size of T, using the mean of the last T values as a prediction for the next value. The expected quadratic error for the prediction can then be computed as follows, where $t_0 = t - T + 1$ is the start of the window:

$$
\begin{aligned}
E\left(\left(Y_{t+1}-\frac{1}{T}\sum_{i=t_0}^{t}Y_i\right)^2\right) &= E\left(Y_{t+1}^2-\frac{2}{T}\sum_{i=t_0}^{t}Y_i\cdot Y_{t+1}+\frac{1}{T^2}\left(\sum_{i=t_0}^{t}Y_i\right)^2\right)\\
&= E\left(Y_{t+1}^2\right)-\frac{2}{T}\sum_{i=t_0}^{t}E\left(Y_i\cdot Y_{t+1}\right)+\frac{1}{T^2}E\left(\left(\sum_{i=t_0}^{t}Y_i\right)^2\right)\\
&= Var\left(Y_{t+1}\right)-\frac{2}{T}\sum_{i=t_0}^{t}Cov\left(Y_i,Y_{t+1}\right)\\
&\quad+\frac{1}{T^2}E\left(\sum_{i=t_0}^{t}Y_i^2+2\sum_{i=t_0+1}^{t}\sum_{j=t_0}^{i-1}Y_i\cdot Y_j\right)\\
&= Var\left(Y_{t+1}\right)-\frac{2}{T}\sum_{i=t_0}^{t}Cov\left(Y_i,Y_{t+1}\right)\\
&\quad+\frac{1}{T^2}\left(\sum_{i=t_0}^{t}Var(Y_i)+2\sum_{i=t_0+1}^{t}\sum_{j=t_0}^{i-1}Cov(Y_i,Y_j)\right)\\
&= (t+1)\cdot Var\left(X\right)-\frac{2}{T}\sum_{i=t_0}^{t}i\cdot Var\left(X\right)\\
&\quad+\frac{1}{T^2}\left(\sum_{i=t_0}^{t}i\cdot Var(X)+2\sum_{i=t_0+1}^{t}\sum_{j=t_0}^{i-1}j\cdot Var(X)\right)\\
&= Var\left(X\right)\left(t+1-\frac{2}{T}\sum_{i=t_0}^{t}i+\frac{1}{T^2}\sum_{i=t_0}^{t}i+2\sum_{i=t_0+1}^{t}\sum_{j=t_0}^{i-1}j\right)\\
&= \frac{\left(2t-2t_0+3\right)\left(t-t_0+2\right)}{6\left(t-t_0+1\right)}Var(X) \qquad (4.6)
\end{aligned}
$$

It is easy to show that (4.6) has its minimum at $t_0 = t$, i.e. for a window of size one, where we simply use the last value as a prediction for the next value. The worst case is to use all values, i.e. $t_0 = 0$. In this case, the expected quadratic error tends to infinity with increasing t. Figure 4.1 shows (4.6) as a function in t for $t_0 = 0$ and $Var(X) = 1$. In this case, t can be interpreted as the window size. The expected quadratic error increases with $O(T)$ in terms of the window size T.

One might criticize that the mean over a window is a much too simple prediction. But actually, any more complicated function will tend to make the expected quadratic error even worse.

The next example is a random walk to which we add noise in the form of a random variable Z with $E(Z) = 0$.

$$
Y_t = \sum_{i=1}^{t}X_i + Z_t \qquad (4.7)
$$

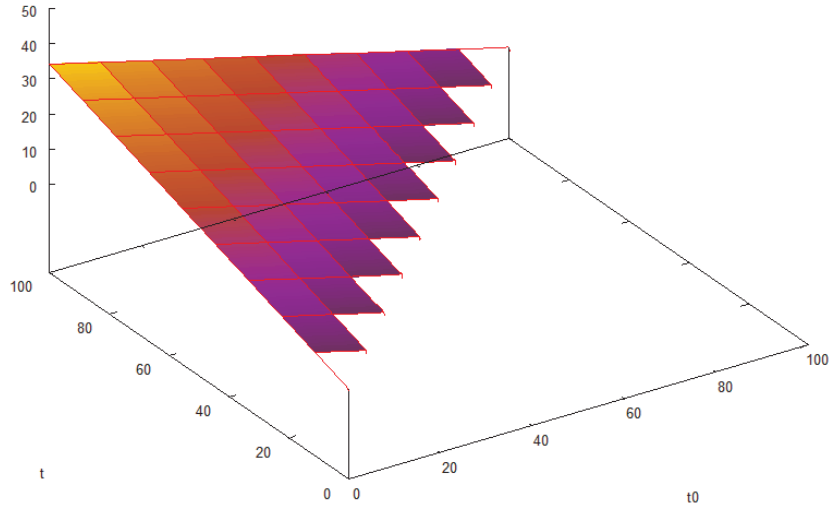

Figure 4.1: The expected quadratic error for a random walk prediction depending on the window size.

where the X_i are i.i.d. as X and the Z_t are i.i.d. as Z. We assume also that the random variables X_i and Z_j are all independent. Since $E(X) = 0$ and $E(Z) = 0$ holds, we have again $E(Y_t) = 0$.

As in the previous example of the random walk without noise, the best prediction for the next value we can choose is the previous value: $\hat{y}_{t+1} = y_t$. The difference between the true and the predicted value is then $Y_{t+1} - Y_t = X_{t+1} + Z_{t+1} - Z_t$.

The variance and covariance of a random walk with noise are

$$Var(Y_t) = t \cdot Var(X) + Var(Z) \tag{4.8}$$

and

$$Cov(Y_t, Y_s) = s \cdot Var(X), \ (s < t), \tag{4.9}$$

respectively.

The expected quadratic error can be computed as follows:

$$\begin{aligned} E\left((Y_{t+1} - Y_t)^2\right) &= E\left((X_{t+1} + Z_{t+1} - Z_t)^2\right) \\ &= Var(X) + 2Var(Z) \end{aligned} \tag{4.10}$$

Based on (4.8) and (4.9) we can calculate the expected quadratic error for the prediction based on a windowing technique.

$$\begin{aligned} E\left(\left(Y_{t+1}-\frac{1}{T}\sum_{i=t_0}^{t}Y_i\right)^2\right) &= E\left(Y_{t+1}^2-\frac{2}{T}\sum_{i=t_0}^{t}Y_i\cdot Y_{t+1}+\frac{1}{T^2}\left(\sum_{i=t_0}^{t}Y_i\right)^2\right) \\ &= E\left(Y_{t+1}^2\right)-\frac{2}{T}\sum_{i=t_0}^{t}E\left(Y_i\cdot Y_{t+1}\right)+\frac{1}{T^2}E\left(\left(\sum_{i=t_0}^{t}Y_i\right)^2\right) \\ &= \frac{(2t-2t_0+3)(t-t_0+2)}{6(t-t_0+1)}Var(X) \\ &\quad +2(t-t_0+1)Var(Z) \end{aligned} \tag{4.11}$$

As before, this function has its minimum at $t_0 = t$ (window size $T = 1$) and is largest for the highest window size. The expected quadratic error is also increasing in $O(T)$.

4.2 Switch model

The two random walk examples described in the previous section can be understood as data generating models that change continuously, representing very simple examples for shift. In this section, we discuss an example for a switching model with sudden jumps in the change of the data generating process. To keep the model as simple as possible, we consider a data generating process that switches between two normal distributions with the same variance σ^2, but one with expected value $\mu_1 = 0$ and the other with expected value $\mu_2 = 1$. The probability to switch from one normal distribution to the other is p in each step and the probability to stay with the same normal distribution is $(1-p)$. The random switching between the two normal distributions is carried out independently in each time step. This model for the data generating process is illustrated as an automaton with two states in figure 4.2.

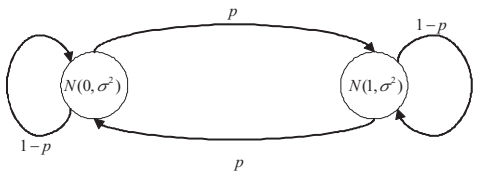

Figure 4.2: A switching model.

Depending on the switching probability p, we can distinguish three cases.

- $p \approx \frac{1}{2}$ The data are generated randomly from any of the two normal distributions in each step of the process. It is impossible to make a prediction from which normal distribution the next value will be drawn based on previous data.

- $p \gg \frac{1}{2}$ There is tendency to permanently switch from one normal distribution to the other in each step.

- $p \ll \frac{1}{2}$ The data generating process tends to stay with the same normal distribution in each step and switches only once in a while.

The first case is does not really represent a switching model. It can be understood as drawing independent random samples from a mixture of the two normal distributions. The second case is less interesting from the practical point of view. We normally assume that normally the model will not stay stable and changes happen only once in a while. Therefore, we only consider the last case. There it is interesting to consider the relation between p and σ^2. Small values for p, i.e. changing from one normal distribution to the other rarely happens, and small values for σ^2, i.e. we distinguish between values from the one or the other normal distribution with a higher probability, make the predictions much easier.

Assuming that we know that our data generating process is as described above, we can try to predict the next value based on a maximum likelihood estimation. We can compute the two likelihoods that the previous value was generated by the normal distribution with expected value $\mu_1 = 0$ and that it comes from the normal distribution with expected value $\mu_2 = 1$. We can then use as a simplified prediction, the mean value of the normal distribution with the higher likelihood.

Given a sequence of values $x_1, x_2, \ldots, x_m$ and a sequence of states $\mu_{i_1}, \mu_{i_2}, \ldots, \mu_{i_m}$ where $\mu_{i_j} \in \{0, 1\}$ in the automaton in figure 4.2, the likelihood for this sequence of states or paths is

$$\prod_{i=j}^{m} f(x_j|\mu_{i_j}) \cdot q_{j-1} \tag{4.12}$$

where $f(x|\mu)$ is the probability density function of the normal distribution with

expected value μ and variance σ^2 and

$$q_{j-1} = \begin{cases} (1-p) & \text{if } \mu_{i_j} = \mu_{i_{j-1}}, \\ p & \text{otherwise.} \end{cases} \tag{4.13}$$

q_0 is the probability for the initial state μ_{i_1}.

The likelihood for a sequence $x_1, x_2, \ldots, x_m$ is then computed as the sum of the likelihoods over all paths.

$$P(\mu = c | x_1, \ldots, x_m) = \sum_{paths} P(path) \tag{4.14}$$

The corresponding path trees for the likelihood computations are shown in figures 4.3 and 4.4 for the state corresponding to the normal distribution $N\left(0, \sigma^2\right)$ and the normal distribution $N\left(1, \sigma^2\right)$, respectively.

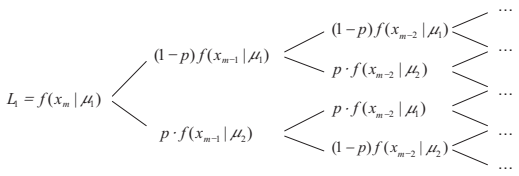

Figure 4.3: Likelihood function for the state 1

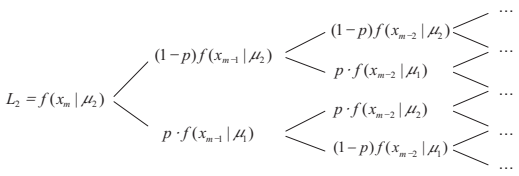

Figure 4.4: Likelihood function for the state 2

It is, of course impossible, to compute the exact likelihood as the sum over all possible paths for larger m, since there are 2^m possible paths. Therefore, we restrict the likelihood computation to the last k states, so that the likelihoods are actually computed "backwards" along the paths. In this way, we also do not require the probabilities for the initial states. The error for the differences in the two likelihoods will be quite small with this procedure, since the conditional likelihood for the two states k steps backwards given the final state will be more or less independent of the final state for larger k according to the theory of Markov chains.

Table 4.1 shows a comparison of the described maximum likelihood estimation where we have chosen $k = 7$ with the Takagi Sugeno evolving fuzzy model

parameters	Maximum likelihood estimation	TS Evolving Fuzzy Models
$\sigma = 0.5$, $p = 0.25$	0.6390	0.7116
$\sigma = 1.0$, $p = 0.1$	1.2792	2.1124
$\sigma = 1.5$, $p = 0.2$	2.6678	3.7769
$\sigma = 2.0$, $p = 0.1$	4.3743	6.1057
$\sigma = 3.0$, $p = 0.001$	9.1785	12.3010
$\sigma = 6.0$, $p = 0.001$	35.7901	50.3640

Table 4.1: MSE

decribed in [28]. In the case of the maximum likelihood estimation we simply predict the expected value of the normal distribution with the higher likelihood. This prediction could even be improved by choosing a weighted mean between the two expected values.

If we knew from which normal distribution the last value had been sampled, we could base our prediction on the knowledge that we sample from a mixture of two normal distributions. In this case, the expected quadratic error would be $\sigma^2 + p - p^2$. From table 4.1 we can see that for small values p, the mean square error of the maximum likelihood-based estimations is very close to theoretical lowest mean square error whereas the evolving system has a much larger error.

In this chapter we cared out the theoretical analysis for two simple well-define stochastic models. The purpose was hereby the comparison between an estimator tailored to the problem and an evolving system without any specific assumptions about the data generating process. In such a way we can identify how near an evolving systems can come to the optimal solution. Of course, our comparisons are "unfair", in the sense that we compare techniques from evolving systems, that do not make any specific assumptions about the data generating process, with estimators tailored to the specific problem. The assumption that we know the data generating process in principle and just need to estimate the parameters is unrealistic. Our intention is to give an impression of how much the evolving systems are biased to track the noise or randomness in the data generating process instead of learning the actual dependencies in the data.

In next chapter we introduce more complicated models. The main focus hither lay on the theoretical analysis of the optimal window size depending on drift and noise in the data.

Chapter 5

Analysis of Regression Models for Sliding Window Based Evolving Systems

Various strategies are proposed to handle the problem of change and noise in data. The change of the underlying data distribution and changes in concept can be detected as proposed for instance in Chapters 2 and 3. Learning algorithms for data with concept drift are proposed in [27, 22, 21, 6]. All these approaches focus on the question how to deal with changes in the data, however it is also very important to know how does drift and noise in the data affect the quality of prediction. Furthermore it might be from utmost significance to have a "good" strategy for choosing which and how many data instances should be used for prediction, since it is impossible and moreover might be, as we will show later, disadvantageous to use all previous data.

In this chapter we carry out a theoretic analysis for two data generating processes: a constant and a linear model with drift and noise. For these two models we consider a simple prediction task, the prediction of the next value which can be understood as regression. As we assume to cope with streaming data, evolving systems should be used. For such kind of problem evolving systems have been proposed for instance in [2, 3, 5, 24, 28]. In this work we use an extremely simplified evolving system based on a windowing technique. Under the assumption that the data generating process is known, we are interested to find the optimal window size as a function of the process parameters. In such a way we can analyse the behaviour of the optimal window size depending on the parameters of the data model. Those simple theoretical models for the data generating process could be

also used as benchmarks for evolving systems [43].

This chapter is organised as follows. We will briefly review existing techniques for selection of window size in Section 5.1. In Section 5.2 we carry out a theoretical analysis for a constant model with drift and noise. This can be considered as a regression problem with a constant function as the regression function. Section 5.3 introduces a simple linear model with drift and noise, which can be interpreted as linear regression. First we compute the optimal window size separately with respect to the prediction of the slope and intercept and finally for the dependent variable. Experimental results are discussed in Section 5.4. For the linear model with drift and noise different parameter settings have been used and empirical error functions are analysed. In all these consideration it is assumed that the parameters of the data generating process change over time, but that the meta-model is stationary, i.e. that the drift is random, but not very high at a certain interval and low in another interval. Consequences for non-stationary meta-models are shown in Section 5.5.

5.1 Related work

The majority of existing machine learning techniques for the mining of data streams uses a sliding time window of fixed size, another small part tries to adapt the size of a sliding window based for instance on the quality of prediction. In [13] Gather et al. apply robust regression techniques to the data streams. They analyse and compare four different techniques, such as repeated median, least median of squares, least trimmed squares and deepest regression. All these approaches are applied to sliding time windows of fixed length. Though the influence of the window size on the prediction is discussed in this work, nevertheless only heuristic methods for the choice of the widow size are used. Similar work is carried out in [8], also here the robust methods for on-line regression are applied to the time windows with fixed size without taking the possible non-stationarity and noisiness of the data into account. In [31] not only the data from the current time window but also the previous data are used for prediction. As we will show in Section 5.4 it could have even more dramatic effects on the quality of prediction.

In the above approaches the size of the sliding window is selected without respect to the possible changes of the data concept. Whereas the techniques pre-

sented below try to select the window size based on some additional information like for instance prediction accuracy or relevance of the data instances. In [25] the window size is automatically adjust in such a way that the estimated generalization error is minimized. Windows of different sizes are therefore used, the window with the highest prediction accuracy is finally chosen. This approach is computationally expensive, therefore efficient computations are needed. An approach for supervised learning under the assumption of concept drift is presented in [5]. Here consistency, temporal and spatial relevance of the data is taken into account. Furthermore a statistical test based on the prediction error is used to detect abrupt concept changes. When they occur, a certain amount of the instances is deleted randomly from the time window according to a distribution which is spatially uniform but temporally skewed. The number of instances to be removed is estimated based on increase of the prediction error. The main idea in [49] is to select the instances for the training data set based on the combination between space and time distance. The size of the training data set is defined with the help of k-fold cross validation. Between N classifiers using the training sets with different sizes the classifier with the best classification accuracy is chosen. This approach requires high computational costs, since for each new data point the validation process should be repeated k-times for every training set (N different sets). Another problem is that the authors use the whole data-set the for search of relevant training instances. This is however impossible due to limited memory and computational time capacity.

In the next sections we will demonstrate that the choice of the window size is one of the crucial points in data stream mining, and therefore this question deserves more attention.

5.2 Constant model with drift and noise

A very simple example for systems with a random change over time is a one-dimensional random walk [40]. The theoretical analysis between an extremely simplified evolving system based on a windowing technique and a model tailored to the known assumptions that the data generating process is a random walk is discussed in Chapter 4.

A slightly more complex model in comparison to the random walk is the fol-

lowing:

$$Z_t \sim N(Y_t, 1) \tag{5.1}$$

where Y_t is a random walk $Y_t = \sum_{i=1}^{t} X_i$ and $X_i \sim N\left(0,\ \sigma^2\right)$.

The process (5.1) can be understood as a constant model with drift and noise, at least when the variance σ^2 of the underlying random walk Y_t is small compared to the noise, generated by the normal distribution (5.1). σ^2 determines how fast or strong the drift is. Without loss of generality, we have chosen the variance $\sigma^2_{\text{noise}} = 1$ for the noise generating normal distribution, since for the analysis of the process only the proportion of the noise in comparison to the drift is of importance.

Figure 5.1 shows data generated by the process (5.1) with $\sigma = 0.2$. This data set exhibits drift as well as noise. In this case, the noise has a stronger effect on the data than the drift. The best strategy for the prediction of the next value for a

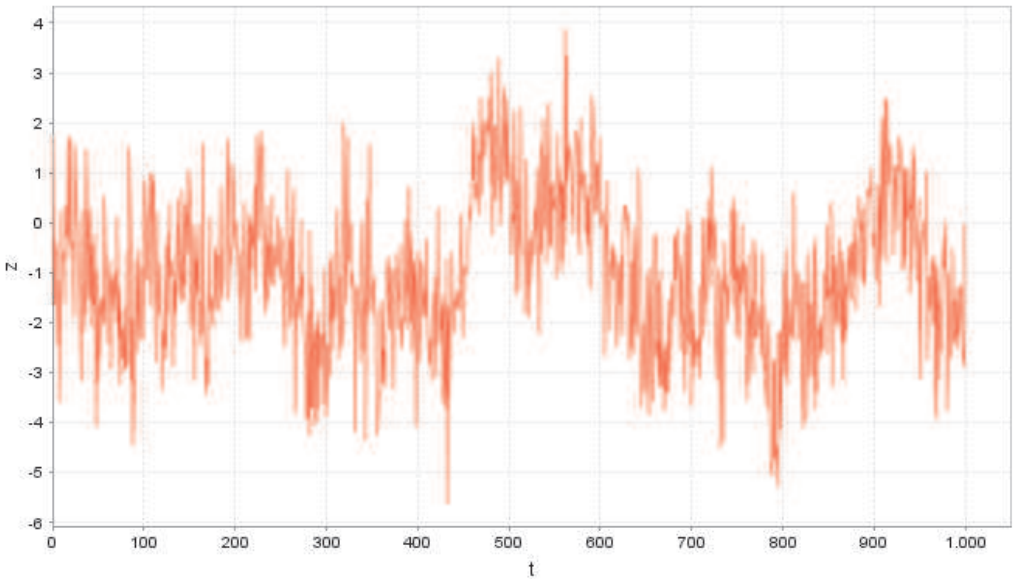

Figure 5.1: constant model with drift and noise

simple random walk (see Section 4 for more information) in terms of minimising the squared error of the predicition is the naïve approach, simply to use the last value as a prediction for the next one. However, in the process (5.1) the data have not only drift (random walk), but are also contaminated with noise. Hence we need to find out a better strategy for the prediction for the data generated by the process (5.1). In the following, we analyse how such a strategy depends on the proportion of the drift compared to the noise.

For the prediction we analyse analogous to the approach in Chapter 4 a very simple evolving system based on a time window technique with a fixed window size of T. As a prediction for the next value, the mean of the last T values is used. Therefore, the question that arises is now: how does the noise affect the optimal size T of the window? For this purpose, we try to minimise the expected quadratic error. The expected quadratic error for the prediction of the next value can be computed as follows.

$$\begin{aligned} E\left(\left(\frac{1}{T}\sum_{i=t_0}^{t} Z_i - Y_{t+1}\right)^2\right) &= E\left(\frac{1}{T^2}\left(\sum_{i=t_0}^{t} Z_i\right)^2 - \frac{2}{T}\sum_{i=t_0}^{t} Z_i \cdot Y_{t+1} + Y_{t+1}^2\right) \\ &= \frac{1}{T^2}E\left(\left(\sum_{i=t_0}^{t} Z_i\right)^2\right) - \frac{2}{T}\sum_{i=t_0}^{t} E\left(Z_i \cdot Y_{t+1}\right) \\ &\quad + E\left(Y_{t+1}^2\right). \end{aligned} \tag{5.2}$$

where t_0 is the first value in the window, i.e. $t_0 = t - T + 1$.

The expression $E\left(Z_i \cdot Y_{t+1}\right)$ can be further simplified as follows:

$$\begin{aligned} E\left(Z_i \cdot Y_{t+1}\right) &= E\left(Z_i \cdot \left(Y_i + \sum_{j=i+1}^{t+1} X_j\right)\right) \\ &= E\left(Z_i \cdot Y_i\right) + \left(\sum_{j=i+1}^{t+1} E\left(\underbrace{Z_i \cdot X_j}_{independent}\right)\right) \\ &= E\left(Z_i \cdot Y_i\right) + \left(\sum_{j=i+1}^{t+1} E\left(Z_i\right) \cdot \underbrace{E\left(X_j\right)}_{=0}\right) \\ &= E\left(Z_i \cdot Y_i\right). \end{aligned} \tag{5.3}$$

Taking Equations (5.3) and $E\left(Y_{t+1}^2\right) = Var\left(Y_{t+1}\right) = (t+1)\cdot\sigma^2$ into account, Equation (5.2) is transformed to

$$\begin{aligned} E\left(\left(\frac{1}{T}\sum_{i=t_0}^{t} Z_i - Y_{t+1}\right)^2\right) &= \frac{1}{T^2}\sum_{i=t_0}^{t} E\left(Z_i^2\right) + \frac{2}{T^2}\sum_{i=t_0+1}^{t}\sum_{j=t_0}^{i-1} E\left(Z_i \cdot Z_j\right) \\ &\quad - \frac{2}{T}\sum_{i=t_0}^{t} E\left(Z_i \cdot Y_i\right) + (t+1)\cdot\sigma^2. \end{aligned} \tag{5.4}$$

In order to find the optimal window size, the minimum Equation (5.4) as a function in t_0 should be computed. For this purpose and to facilitate the readability

of the text, the terms from formula (5.4) will be treated separately.
First of all, we calculate the expected value of Z_i.

$$E(Z_i) = \int_{-\infty}^{+\infty} \mu \cdot f_{Y_i}(\mu)\, d\mu = \frac{1}{\sqrt{2\pi a}} \int_{-\infty}^{+\infty} \mu \cdot e^{\frac{-\mu^2}{2\cdot a}}\, d\mu = 0$$

where a is the variance of Y_i with $a = i \cdot \sigma^2$.
The variance of Z_i is

$$\begin{aligned} Var(Z_i) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (z-\mu)^2 \cdot f_{Z_i|Y_i}(z) \cdot f_{Y_i}(\mu)\, dz d\mu \\ &= \frac{1}{2\pi\sqrt{a}} \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} z^2 \cdot e^{-\frac{(z-\mu)^2}{2} - \frac{\mu^2}{2a}}\, dz d\mu \\ &= \frac{1}{2\pi\sqrt{a}} \cdot \left(2\pi a^{3/2} + 2\pi\sqrt{a}\right) \\ &= a+1. \end{aligned} \tag{5.5}$$

For the expected value of the product $Z_i \cdot Y_i$, we obtain from $E(Y_i) = 0$ and $E(Z_i) = 0$ (hence $E(Z_i \cdot Y_i) = Cov(Z_i, Y_i)$) the following expression.

$$\begin{aligned} E(Z_i \cdot Y_i) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (z-\mu_z)(y-\mu_y) \cdot f_{Z_i \cdot Y_i}(z,y)\, dz dy \\ &= \frac{1}{2\pi\sqrt{a}} \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} z \cdot y \cdot e^{-\frac{(z-y)^2}{2} - \frac{y^2}{2a}}\, dz dy \\ &= \frac{1}{2\pi\sqrt{a}} \cdot 2\pi a^{3/2} \\ &= a. \end{aligned} \tag{5.6}$$

The last term to be computed is $E\left(Z_i \cdot Z_j\right)$. As mentioned above, a is the variance of Y_i and $a = i\sigma^2$. Moreover, we assume that $j = i+k$ and $k \neq 0$, corresponding to $Y_j = Y_i + \sum_{l=i+1}^{j} X_l$. According to the infinite divisibility of the normal distribution, the sum of the random variables X_l follows also a normal distribution: $\sum_{l=i+1}^{j} X_l \sim N\left(0, k\sigma^2\right)$. Furthermore, let us denote $E(Y_i) = \mu$ and $E\left(Y_j\right) = \mu + \eta$. Therefore $E\left(Z_i \cdot Z_j\right)$ can be computed as follows:

$$\begin{aligned} E\left(Z_i \cdot Z_j\right) &= \frac{1}{4\pi^2\sqrt{a \cdot b}} \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} z_i \cdot z_j \\ &\quad \times e^{-\frac{\left(z_j-\mu-\eta\right)^2}{2} - \frac{(z_i-\mu)^2}{2} - \frac{\mu^2}{2a} - \frac{\eta^2}{2b}}\, d\mu\, d\eta\, dz_i dz_j \\ &= \frac{1}{4\pi^2\sqrt{a \cdot b}} \cdot 4\pi^2 a^{3/2}\sqrt{b} = a. \end{aligned} \tag{5.7}$$

Based on Equations (5.5), (5.6) and (5.7), we obtain the following expression for Equation (5.4)

$$
\begin{aligned}
E\left(\left(\frac{1}{T}\sum_{i=t_0}^{t} Z_i - Y_{t+1}\right)^2\right) &= \frac{1}{T^2}\sum_{i=t_0}^{t}\left(i\sigma^2+1\right)+\frac{2}{T^2}\sum_{i=t_0+1}^{t}\sum_{j=t_0}^{i-1} j\sigma^2 - \frac{2}{T}\sum_{i=t_0}^{t} i\sigma^2 + (t+1)\cdot\sigma^2 \\
&= \frac{3\left(t+t_0\right)+2\left(t-t_0\right)\left(t+2t_0-1\right)+6\left(1-t_0\right)\left(t-t_0+1\right)}{6T}\sigma^2+\frac{1}{T} \\
&= \frac{\left(2t^2-4t\cdot t_0+7t+2t_0^2-7t_0+6\right)\sigma^2+6}{6\left(t-t_0+1\right)}. \qquad (5.8)
\end{aligned}
$$

In order to determine the minimum of the function (5.8), we take the derivate with respect to the parameter t_0, which yields

$$
\begin{aligned}
\left(E\left(\left(\frac{1}{T}\sum_{i=t_0}^{t} Z_i - Y_{t+1}\right)^2\right)\right)' &= \frac{(4t_0-4t-7)}{6\left(t-t_0+1\right)}\sigma^2 + \\
&\quad + \frac{\left(2t^2-4t\cdot t_0+7t+2t_0^2-7t_0+6\right)\sigma^2+6}{6\left(t-t_0+1\right)^2}. \qquad (5.9)
\end{aligned}
$$

The right-hand side of the function (5.9) is zero, t_0 is chosen as

$$
t_0 = t+1\pm\sqrt{\frac{1}{2}+\frac{3}{\sigma^2}}. \qquad (5.10)
$$

Hence the optimal windows size is

$$
T = t-t_0+1 = \sqrt{\frac{1}{2}+\frac{3}{\sigma^2}}. \qquad (5.11)
$$

From Equation (5.11) we can see – as would be expected – that with increasing drift (σ^2), the window size decreases. When $\sigma^2 \geq 6$, i.e. the drift becomes too large in comparison to the noise, the window size shrinks to 1 as in the ordinary random walk.

We have assumed that Z_t has a constant variance of one. As mentioned before, the optimal window size depends only on the quotient of the noise and the drift. By assuming that the data generating process (5.1) has variance σ_2^2, i.e.

$$
Z_i \sim N\left(y_t, \sigma_2^2\right), \qquad (5.12)
$$

and denoting the variance of X_i by σ_1^2, we can recalculate the expected quadratic error (5.2). In fact only Equation (5.5) needs to be changed to

$$
Var\left(Z_i\right) = i\cdot\sigma_1^2+\sigma_2^2. \qquad (5.13)
$$

Hence, Equation (5.8) becomes

$$E\left(\left(\frac{1}{T}\sum_{i=t_0}^{t} Z_i - Y_{t+1}\right)^2\right) = \frac{\left(2t^2 - 4t\cdot t_0 + 7t + 2t_0^2 - 7t_0 + 6\right)\sigma_1^2 + 6\sigma_2^2}{6\left(t - t_0 + 1\right)}$$
$$= \frac{1}{\sigma_2^2}\frac{\left(2t^2 - 4t\cdot t_0 + 7t + 2t_0^2 - 7t_0 + 6\right)\frac{\sigma_1^2}{\sigma_2^2} + 6}{6\left(t - t_0 + 1\right)}. \tag{5.14}$$

Accordingly, the optimal windows size is then

$$T = t - t_0 + 1 = \sqrt{\frac{1}{2} + \frac{3}{\frac{\sigma_1^2}{\sigma_2^2}}}. \tag{5.15}$$

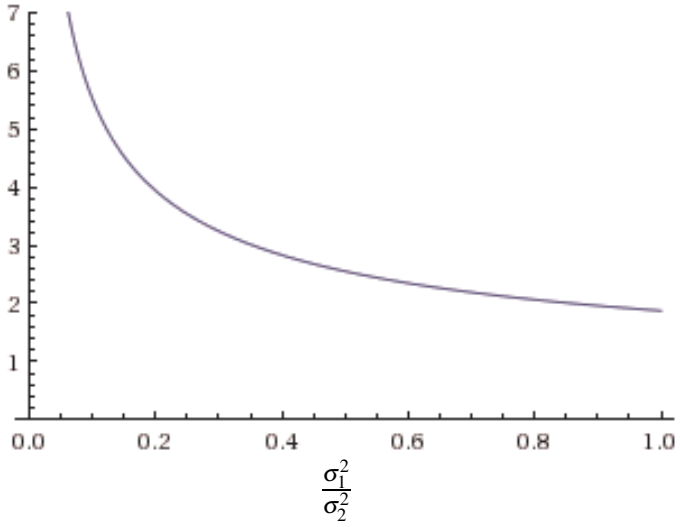

Figure 5.2: Optimal windows size depending on the ratio of the drift and the noise.

Figure 5.2 shows Equation (5.15) as a function of the ratio of the drift and the noise $\frac{\sigma_1^2}{\sigma_2^2}$. We can see from the figure that with decreasing $\frac{\sigma_1^2}{\sigma_2^2}$, the windows size increases. That means if the noise in the data is stronger in comparison to the drift, for the prediction of the next value more previous values should be used (larger window size). And for stronger drift compared to the noise, the windows size tends to 1, which relates to the prediction for an ordinary random walk, since the noise is negligible in comparison to the drift.

5.3 Linear model with drift and noise

The constant model with drift and noise described in the previous section can be considered as a regression problem with a constant function as the regression function. In this section, we discuss an example for linear regression with drift and noise. The model for linear regression is given by

$$y_i = B_{1,i} \cdot x_i + B_{0,i} + \varepsilon_i \tag{5.16}$$

where the random variables B_0 and B_1 are random walks (see Chapter 4, Equation (4.1)) and follow normal distributions: $B_{0,i} \sim N\left(0, i\sigma_0^2\right)$, $B_{1,i} \sim N\left(0, i\sigma_1^2\right)$. The random variable ε_i represents noise and is normally distributed with expected value zero and variance σ^2, i.e. $\varepsilon_i \sim N\left(0, \sigma^2\right)$. It is assumed that the ε_i-s are independent. Therefore the model (5.16) has drifts in the slope B_1 and intercept B_0 of the regression line. Furthermore, the random variable ε_i adds noise to the linear relationship between the dependent variable y and the predictor x.

In order to simplify the analysis, the sampling values x_l for the predictor are assumed to be fixed in the following way: we sample repeatedly n points $x_1, \ldots, x_n$. The values x_l are equi-distant values from the sampling interval $[-a;\ a]$, i.e. $x_l = -a + (l-1)\,\frac{2a}{(n-1)}$ for $l = 1, \ldots, n$.

The sum of the values x_l is equal to zero (5.17) and according to that the expected value of X and the mean are also equal to zero: $\bar{x} = 0$.

$$\sum_{l=1}^{n} x_l = 0 \tag{5.17}$$

Figure 5.3 shows data generated by the process (5.16) with the following settings: $\sigma_1 = 0.7$, $\sigma_0 = 0.5$ and $\sigma = 0.1$. The values for the predictor are selected from the interval $[-1;\ 1]$ with $n = 100$, the start value for the slope is chosen as 1 and for the intercept as 0.

We assume that the drift occurs only after a whole cycle of sampled predictor values $x_1, \ldots, x_n$, i.e. the random walks B_0 and B_1 are only updated after one sequence of the values $x_1, \ldots, x_n$. The line starting at the highest point corresponds to the first sampling cycle: $y_{1,l} = B_{1,1}x_l + B_{0,1} + \varepsilon_{1,l}$, $l = 1, \ldots, n$. The line starting at the lowest point corresponds to the second cycle where the slope and the intercept have drifted the first time. The line in the middle corresponds to the third sampling cycle, where another drift of the slope and the intercept has taken place.

Figure 5.3: Data generated with the model (5.16).

It can be seen clearly that the first two lines have almost the same intercept, but different slope, the second and third line have almost the same slope, but differ in their intercept. This means that after the first cycle, $B_{1,1}$ did a larger jump, whereas $B_{0,1}$ changed only very little. In contrast, in the next step the larger drift occurred for the intercept.

In order to predict the dependent variable y, we need to estimate the parameters of the model B_0 and B_1 first. A large variety of methods have been developed for the estimation of the parameters of a linear model. In this work we restrict our consideration to the simplest and most common method for linear regression: the least squares estimator. The least squares principle for simple linear regression can be explained as follows: choose the estimates for B_0 and B_1 in such a manner that the sum of squared residuals becomes minimal. A formal development of the least squares estimates can be found for instance in [47, 16]. Furthermore, the least squares estimator has nice statistical properties. The estimator is unbiased, consistent and efficient under certain assumptions (see [16]).

Let $\hat{B}_0$ and $\hat{B}_1$ be the unbiased estimators for B_0 and B_1 (when no drift occurs). The least squares estimates are in this case given by

$$\begin{aligned} \hat{B}_1 &= \frac{\sum_{i=1}^{t}\sum_{l=1}^{n}(x_l-\bar{x})\,y_{i,l}}{\sum_{i=1}^{t}\sum_{l=1}^{n}(x_l-\bar{x})^2}, \\ \hat{B}_0 &= \bar{y}-\hat{B}_1\bar{x}. \end{aligned}$$

In order to distinguish the estimators of B_1 and B_0 from the true parameters B_1

and B_0, which are also random variables in our case, we denote the estimators by $\hat{B}_0$ and $\hat{B}_1$.

Similar to the previous section, we use a sliding window technique for the estimation of the next value of the slope and intercept. As already in Section 5.2, the window starts at the value with index t_0 and the window size is denoted by T. According to the fact that the model has noise and drifts in the slope and intercept the question presently is: which effects have drifts and noise on the optimal window size?

First we analyse the optimal window size for B_1, without taking B_0 into account. The estimator for B_1 is given by

$$\begin{aligned} \hat{B}_1 &= \frac{\sum_{i=t_0}^{t}\sum_{l=1}^{n}(x_l-\bar{x})\,y_{i,l}}{\sum_{i=t_0}^{t}\sum_{l=1}^{n}(x_l-\bar{x})^2} \\ &= \frac{\sum_{i=t_0}^{t}\sum_{l=1}^{n}x_l y_{i,l}}{\sum_{i=t_0}^{t}\sum_{l=1}^{n}x_l^2}. \end{aligned} \tag{5.18}$$

To define the best possible window size we have to minimise the expected quadratic error. The expected quadratic error for the prediction of the slope can be computed as follows.

$$\begin{aligned} E\left(\left(\frac{\sum_{i=t_0}^{t}\sum_{l=1}^{n}x_l y_{i,l}}{\sum_{i=t_0}^{t}\sum_{l=1}^{n}x_l^2}-B_{1,t+1}\right)^2\right) &= E\left(\left(\frac{\sum_{i=t_0}^{t}\sum_{l=1}^{n}x_l y_{i,l}}{\sum_{i=t_0}^{t}\sum_{l=1}^{n}x_l^2}\right)^2\right. \\ &\quad \left.-2\cdot B_{1,t+1}\frac{\sum_{i=t_0}^{t}\sum_{l=1}^{n}x_l y_{i,l}}{\sum_{i=t_0}^{t}\sum_{l=1}^{n}x_l^2}+B_{1,t+1}^2\right) \\ &= \frac{1}{(C)^2}E\left(\left(\sum_{i=t_0}^{t}\sum_{l=1}^{n}x_l y_{i,l}\right)^2\right)+E\left(B_{1,t+1}^2\right) \\ &\quad -\frac{2}{C}E\left(B_{1,t+1}\sum_{i=t_0}^{t}\sum_{l=1}^{n}x_l y_{i,l}\right) \end{aligned} \tag{5.19}$$

where $C=\sum_{i=t_0}^{t}\sum_{l=1}^{n}x_l^2=(t-t_0+1)\sum_{l=1}^{n}x_l^2$. It follows from Equation (5.17) that

$$\sum_{i=t_0}^{t}\sum_{l=1}^{n}x_l y_{i,l}=\sum_{i=t_0}^{t}B_{1,i}\sum_{l=1}^{n}x_l^2+\sum_{i=t_0}^{t}\sum_{l=1}^{n}x_l\varepsilon_{l,i} \tag{5.20}$$

holds. Now we compute each term separately.

$$\begin{aligned} E\left(\left(\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l y_{i,l}\right)^2\right) &= E\left(\left(\sum_{i=t_0}^{t} B_{1,i}\sum_{l=1}^{n} x_l^2\right)^2\right) \\ &\quad +2\sum_{i=t_0}^{t}\underbrace{E\left(B_{1,i}\right)}_{=0}\sum_{l=1}^{n} x_l^2 \sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l \underbrace{E\left(\varepsilon_{l,i}\right)}_{=0} + E\left(\left(\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l\varepsilon_{l,i}\right)^2\right) \\ &= \sum_{i=t_0}^{t} E\left(B_{1,i}^2\right)\left(\sum_{l=1}^{n} x_l^2\right)^2 + 2\sum_{i=t_0+1}^{t}\sum_{j=t_0}^{i-1} E\left(B_{1,i}B_{1,j}\right)\left(\sum_{l=1}^{n} x_l^2\right)^2 \\ &\quad +E\left(\left(\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l\varepsilon_{l,i}\right)^2\right). \end{aligned} \tag{5.21}$$

The first two terms in Equation (5.21) are easy to compute. For the computation of the third term further steps are needed.

$$\begin{aligned} E\left(\left(\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l\varepsilon_{l,i}\right)^2\right) &= E\left(\sum_{i=t_0}^{t}\left(\sum_{l=1}^{n} x_l\varepsilon_{l,i}\right)^2\right) \\ &\quad +2\sum_{i=t_0+1}^{t}\sum_{j=t_0}^{i-1}\left(\sum_{l=1}^{n} x_l\underbrace{E\left(\varepsilon_{l,i}\right)}_{=0}\sum_{l=1}^{n} x_l\underbrace{E\left(\varepsilon_{l,j}\right)}_{=0}\right) \\ &= \sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l^2 E\left(\varepsilon_{l,i}^2\right) + 2\sum_{i=t_0+1}^{t}\sum_{l=1}^{n}\sum_{m=1}^{n} x_l x_m \underbrace{E\left(\varepsilon_{l,i}\right)}_{=0}\underbrace{E\left(\varepsilon_{m,i}\right)}_{=0}. \end{aligned} \tag{5.22}$$

Taking Equation (5.22) into account, (5.21) becomes

$$\begin{aligned} E\left(\left(\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l y_{i,l}\right)^2\right) &= \sum_{i=t_0}^{t} E\left(B_{1,i}^2\right)\left(\sum_{l=1}^{n} x_l^2\right)^2 + 2\sum_{i=t_0+1}^{t}\sum_{j=t_0}^{i-1} E\left(B_{1,i}B_{1,j}\right)\left(\sum_{l=1}^{n} x_l^2\right)^2 \\ &\quad +E\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l^2 E\left(\varepsilon_{l,i}^2\right). \end{aligned} \tag{5.23}$$

Hence the first term of the expected quadratic error (5.19) is computed. Now we can compute the third term in Equation (5.19).

$$E\left(B_{1,t+1}\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l y_{i,l}\right) = \sum_{i=t_0}^{t} E\left(B_{1,t+1}B_{1,i}\right)\sum_{l=1}^{n} x_l^2. \tag{5.24}$$

Therefore, we obtain the following equation for the expected quadratic error.

$$
\begin{aligned}
E\left(\left(\frac{\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l y_{i,l}}{\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l^2} - B_{1,t+1}\right)^2\right) &= \frac{1}{C^2}\left(\sum_{i=t_0}^{t} i\sigma_1^2\left(\sum_{l=1}^{n} x_l^2\right)^2 + 2\sum_{i=t_0+1}^{t}\sum_{j=t_0}^{i-1} j\hat{\sigma}^2\left(\sum_{l=1}^{n} x_l^2\right)^2\right. \\
&\quad \left. + \sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l^2\sigma^2\right) - \frac{2}{C}\sum_{i=t_0}^{t} i\sigma_1^2\sum_{l=1}^{n} x_l^2 + (t+1)\,\sigma_1^2 \\
&= \sigma_1^2\left(\frac{3\,(t+t_0)+2\,(t-t_0)\,(t+2t_0-1)}{6\,(t-t_0+1)} + (1-t_0)\right) \\
&\quad + \frac{\sigma^2}{(t-t_0+1)\sum_{l=1}^{n} x_l^2} \qquad (5.25)
\end{aligned}
$$

The minimum of the function (5.25) is at

$$t_0 = t + 1 - \sqrt{\frac{1}{2} + \frac{3}{\tilde{n}}\cdot\frac{\sigma^2}{\sigma_1^2}} \qquad (5.26)$$

where $\tilde{n} = \sum_{l=1}^{n} x_l^2 = \frac{a^2 n(n+1)}{3(n-1)}$. According to Equation (5.26), the optimal window size is given as a function of σ_1^2 and σ^2 when a and n are considered as constant values.

$$T = \sqrt{\frac{1}{2} + \frac{3}{\tilde{n}}\cdot\frac{\sigma^2}{\sigma_1^2}}. \qquad (5.27)$$

It is obvious that the optimal window size in this case is independent of the drift of the intercept (σ_0^2). The reason for this is the following. The random walks B_1 and B_0 are independent that means that the drift of the slope occurs independently of the drift of the intercept. Consequently, $B_{0,t+1}$ is irrelevant for the estimation of $B_{1,t+1}$.

In the next step we compute the expected quadratic error for the intercept B_0. The estimator of B_0 is $\hat{B}_0 = \bar{y} - \hat{B}_1\bar{x}$, where in our case $\bar{x} = 0$ holds. Therefore the expected quadratic error is $E\left((\bar{y} - B_{0,t+1})^2\right)$. First we need to compute the expected value of Y.

$$
\begin{aligned}
\bar{y} &= \frac{1}{n\cdot T}\sum_{i=t_0}^{t}\sum_{l=1}^{n}\left(B_{1,i}x_l + B_{0,i} + \varepsilon_{l,i}\right) \\
&= \frac{1}{n\cdot T}\left(\sum_{i=t_0}^{t} B_{1,i}\underbrace{\sum_{l=1}^{n} x_l}_{=0} + \sum_{i=t_0}^{t} B_{0,i}\sum_{l=1}^{n} 1 + \sum_{i=t_0}^{t}\sum_{l=1}^{n}\varepsilon_{l,i}\right) \\
&= \frac{1}{n\cdot T}\left(\sum_{i=t_0}^{t} n\cdot B_{0,i} + \sum_{i=t_0}^{t}\sum_{l=1}^{n}\varepsilon_{l,i}\right). \qquad (5.28)
\end{aligned}
$$

According to Equation (5.28), we have

$$\begin{aligned}
E\left(\left(\bar{y}-B_{0,t+1}\right)^2\right) &= \frac{1}{T^2}E\left(\left(\sum_{i=t_0}^{t}B_{0,i}\right)^2\right)+\frac{1}{n\cdot T^2}E\left(\left(\sum_{i=t_0}^{t}\sum_{l=1}^{n}\varepsilon_{l,i}\right)^2\right)\\
&\quad +E\left(\left(B_{0,t+1}\right)^2\right)+\frac{1}{n\cdot T}\underbrace{E\left(\sum_{i=t_0}^{t}B_{0,i}\sum_{l=1}^{n}\varepsilon_{l,i}\right)}_{=0}\\
&\quad -\frac{2}{T}E\left(\sum_{i=t_0}^{t}B_{0,i}\cdot B_{0,t+1}\right)-\frac{2}{n\cdot T}\underbrace{E\left(B_{0,t+1}\sum_{i=t_0}^{t}\sum_{l=1}^{n}\varepsilon_{l,i}\right)}_{=0}\\
&= \frac{1}{T^2}\sum_{i=t_0}^{t}i\sigma_0^2+\frac{2}{T^2}\sum_{i=t_0+1}^{t}\sum_{j=t_0}^{i-1}j\sigma_0^2-\frac{2}{T}\sum_{i=t_0}^{t}i\sigma_0^2+(t+1)\,\sigma_0^2\\
&\quad +\frac{1}{n^2\cdot T^2}E\left(\left(\sum_{i=t_0}^{t}\sum_{l=1}^{n}\varepsilon_{l,i}\right)^2\right). \qquad (5.29)
\end{aligned}$$

The only remaining term to be computed is $\frac{1}{n^2\cdot T^2}E\left(\left(\sum_{i=t_0}^{t}\sum_{l=1}^{n}\varepsilon_{l,i}\right)^2\right)$.

$$\begin{aligned}
\frac{1}{n^2\cdot T^2}E\left(\left(\sum_{i=t_0}^{t}\sum_{l=1}^{n}\varepsilon_{l,i}\right)^2\right) &= \frac{1}{n^2\cdot T^2}E\left(\left(\sum_{i=t_0}^{nT+t_0-1}\varepsilon_{l,i}\right)^2\right)\\
&= \frac{1}{n^2\cdot T^2}\sum_{i=t_0}^{nT+t_0-1}E\left(\varepsilon_{l,i}^2\right)+2\sum_{i=t_0+1}^{nT+t_0-1}\sum_{j=t_0}^{i-1}\underbrace{E\left(\varepsilon_i\right)}_{=0}\underbrace{E\left(\varepsilon_j\right)}_{=0}\\
&= \frac{\sigma^2}{n^2\cdot T^2}\sum_{i=t_0}^{nT+t_0-1}1\\
&= \frac{\sigma^2}{n^2\cdot T^2}\left(nT+t_0-1-t_0+1\right)=\frac{\sigma^2}{n\cdot T}. \qquad (5.30)
\end{aligned}$$

The expected quadratic error for $B_{0,t+1}$ is therefore

$$E\left(\left(\bar{y}-B_{0,t+1}\right)^2\right)=\frac{6+7t+2t^2-7t_0-4t\cdot t_0+2t_0^2}{6\left(t-t_0+1\right)}\sigma_0^2+\frac{\sigma^2}{n\cdot T}. \qquad (5.31)$$

The minimum of the function (5.31) is similar to the one for the slope (5.26).

$$t_0=t+1-\sqrt{\frac{1}{2}+\frac{3}{n}\frac{\sigma^2}{\sigma_0^2}}. \qquad (5.32)$$

Figure (5.4) shows the optimal window size for $\hat{B}_0$ and $\hat{B}_1$ corresponding to the functions (5.32) and (5.26) with $n=90$ and $X\in[-1;1]$.

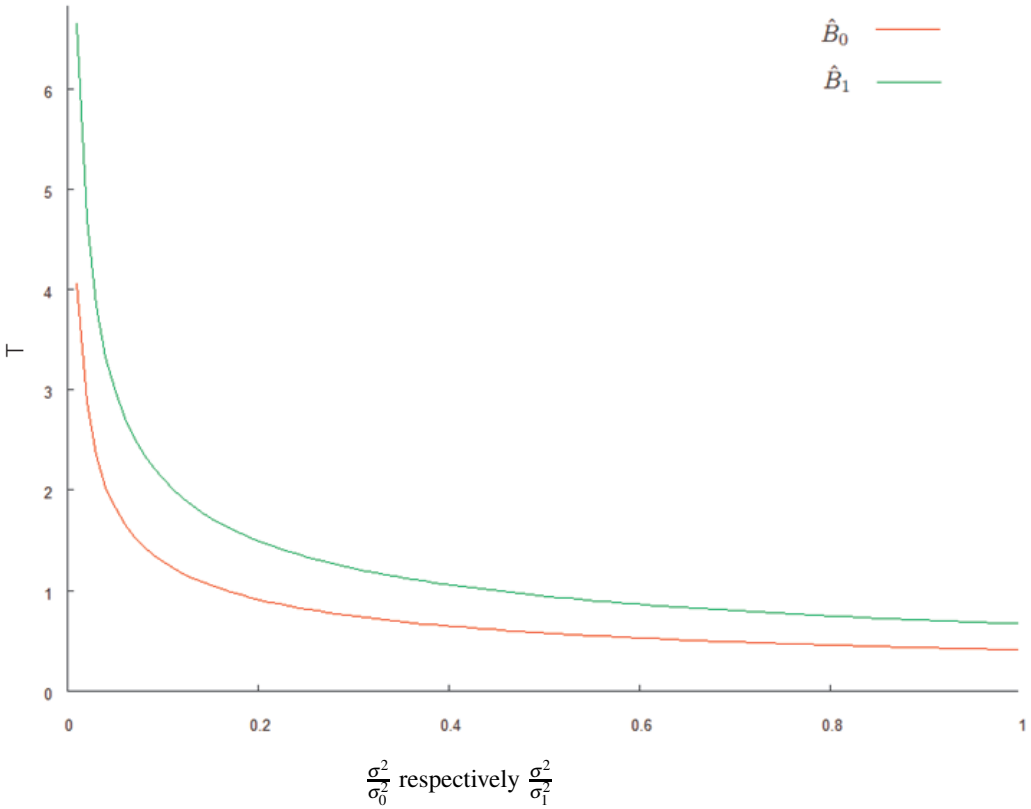

Figure 5.4: Optimal window size for $\hat{B}_0$ and $\hat{B}_1$.

So far, we have computed the optimal window size separately for the estimator for the intercept $\hat{B}_0$ and for the estimator for the slope $\hat{B}_1$. However, usually the values for $B_{0,t+1}$ and $B_{1,t+1}$ are unknown. Only the value for y_{t+1} can be observed. Therefore it makes sense to determine the optimal window size with respect to the prediction of the next value for y. For this purpose, the minimum of the expected quadratic error $E\left((y-y_{t+1})^2\right)$ needs to be computed.

$$\begin{aligned} E\left((y-y_{t+1})^2\right) &= E\left(\left(\hat{B}_1 x+\hat{B}_0-B_{1,t+1}x-B_{0,t+1}-\varepsilon_{t+1}\right)^2\right) \\ &= E\left(\left(\left(\hat{B}_1-B_{1,t+1}\right)x+\left(\hat{B}_0-B_{0,t+1}\right)-\varepsilon_{t+1}\right)^2\right) \\ &= E\left(\left(\hat{B}_1-B_{1,t+1}\right)^2\right)x^2+E\left(\left(\hat{B}_0-B_{0,t+1}\right)^2\right) \\ &\quad +E\left((\varepsilon)^2\right)+2E\left(\left(\hat{B}_1-B_{1,t+1}\right)\left(\hat{B}_0-B_{0,t+1}\right)\right) \\ &\quad +2E\left(\left(\hat{B}_1-B_{1,t+1}\right)\varepsilon_{t+1}\right)+2E\left(\left(\hat{B}_0-B_{0,t+1}\right)\varepsilon_{t+1}\right). \end{aligned} \tag{5.33}$$

The first two terms are the same as in the analysis before, so that we can use Equations (5.25) and (5.31). The third term is the variance of ε and is therefore

equal to σ^2. So we just need to evaluate the three remaining terms. In order to simplify the computation, we rewrite $\hat{B}_1$ and $\hat{B}_0$ in the following forms.

$$\hat{B}_1 = \frac{1}{\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l^2}\left(\sum_{i=t_0}^{t} B_{1,i}\sum_{l=1}^{n} x_l^2 + \sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l \varepsilon_{l,i}\right). \tag{5.34}$$

$$\hat{B}_0 = \frac{1}{n\cdot T}\left(\sum_{i=t_0}^{t} nB_{0,i} + \sum_{i=t_0}^{t}\sum_{l=1}^{n} \varepsilon_{l,i}\right). \tag{5.35}$$

From Equation (5.34) we obtain that $\hat{B}_1$ and $B_{0,t+1}$ are independent, so that $E\left(\hat{B}_1 B_{0,t+1}\right) = E\left(\hat{B}_1\right)\cdot E\left(B_{0,t+1}\right) = 0$ holds. The same applies to $E\left(\hat{B}_0 B_{1,t+1}\right)$ and $E\left(B_{1,t+1}B_{0,t+1}\right)$ (see Equation (5.35)). Taking this fact into account, the remaining terms can be computed as follows.

$$\begin{aligned} E\left(\left(\hat{B}_1 - B_{1,t+1}\right)\left(\hat{B}_0 - B_{0,t+1}\right)\right) &= E\left(\hat{B}_1\hat{B}_0\right) - \underbrace{E\left(\hat{B}_1 B_{0,t+1}\right)}_{=0} - \underbrace{E\left(\hat{B}_0 B_{1,t+1}\right)}_{=0} \\ &\quad + \underbrace{E\left(B_{1,t+1}B_{0,t+1}\right)}_{=0}. \end{aligned} \tag{5.36}$$

Furthermore, we have

$$\begin{aligned} E\left(\hat{B}_1\hat{B}_0\right) &= \frac{1}{nT\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l^2} \; E\left(\sum_{i=t_0}^{t} B_{1,i}\sum_{l=1}^{n} x_l^2 + \sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l\varepsilon_{l,i}\right) \\ &\qquad \cdot E\left(\sum_{i=t_0}^{t} nB_{0,i} + \sum_{i=t_0}^{t}\sum_{l=1}^{n}\varepsilon_{l,i}\right) \\ &= \frac{1}{nT\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l^2} \; E\left(\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l\varepsilon_{l,i}\cdot\sum_{i=t_0}^{t}\sum_{l=1}^{n}\varepsilon_{l,i}\right) \\ &= \frac{1}{nT\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l^2} \; \left(\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l E\left(\varepsilon_{l,i}^2\right) + \sum_{i=t_0}^{t}\sum_{l=1}^{n}\sum_{j=t_0}^{t}\sum_{m\in M} x_l \underbrace{E\left(\varepsilon_{l,i}\right)}_{=0}\underbrace{E\left(\varepsilon_{j,m}\right)}_{=0}\right) \\ &= \frac{\sigma^2}{nT\sum_{i=t_0}^{t}\sum_{l=1}^{n} x_l^2} \; \sum_{i=t_0}^{t}\underbrace{\sum_{l=1}^{n} x_l}_{=0} = 0 \end{aligned} \tag{5.37}$$

where $M = \{h \in \{1,\ldots,n\} \,|\, j = i \Rightarrow h \neq l\}$.

According to Equation (5.37) we have

$$E\left(\left(\hat{B}_1 - B_{1,t+1}\right)\left(\hat{B}_0 - B_{0,t+1}\right)\right) = 0. \tag{5.38}$$

Analogously to Equation (5.36), we obtain

$$E\left(\left(\hat{B}_1 - B_{1,t+1}\right)\varepsilon_{t+1}\right) = 0. \tag{5.39}$$

$$E\left(\left(\hat{B}_0 - B_{0,t+1}\right)\varepsilon_{t+1}\right) = 0 \tag{5.40}$$

Since the tree last terms in Equation (5.33) are equal to zero, we obtain the following expression for the expected quadratic error.

$$\begin{aligned} E\left((y - y_{t+1})^2\right) &= \left(\frac{6 + 7t + 2t^2 - 7t_0 - 4t \cdot t_0 + 2t_0^2}{6T}\right)\left(x^2\sigma_1^2 + \sigma_0^2\right) \\ &\quad + \left(\frac{n \cdot x^2 + \tilde{n}}{n \cdot \tilde{n} \cdot T} + 1\right)\sigma^2. \end{aligned} \tag{5.41}$$

The function (5.41) has its minimum at $t_0 = t + 1 - \sqrt{\frac{1}{2} + \frac{3}{b}\frac{\sigma^2}{\left(x^2\sigma_1^2 + \sigma_0^2\right)}}$, therefore the optimal window size for the prediction is in this case

$$T = \sqrt{\frac{1}{2} + \frac{3}{b}\frac{\sigma^2}{\left(x^2\sigma_1^2 + \sigma_0^2\right)}} \tag{5.42}$$

where $b = \frac{n \cdot \tilde{n}}{n \cdot x^2 + \tilde{n}}$.

As we can see by Equation (5.42), the optimal window size is a function of σ^2, σ_1^2 and σ_0^2. Hence the choice of the optimal window size depends on the drifts in slope and intercept, but also on the noise. For highly noisy data with relatively small drifts, the best option is to look "more backward", i.e. $\lim_{\sigma^2 \to \infty} T = \infty$, whereas for very little noise only the last value should be used for the prediction, $\lim_{\frac{\sigma^2}{\sigma_1^2 x^2 + \sigma_0^2} \to 0} T = \frac{1}{\sqrt{2}}$, i.e. $T = 1$ have to be selected in this case. Note that a window of size m in our case contains $m \cdot n$ points for the regression, since one sampling step corresponds to obtaining the points $(-a + (l-1)\frac{2a}{(n-1)}, y_l)$ ($l \in \{1, \ldots, n\}$). This means even a window of size 1 contains n sampling points. Apart from that, the effect of σ_1^2 depends on the sampled X-values and can be decreased or increased with decreasing or increasing X. Since it is impossible to compute a window size for each new value x, the expected value of X^2 could be used for x^2 in Equation (5.42).

By using different values t_0 in Equation (5.41), one for B_0 and one for B_1 (respectively $t_0^{(0)}$ and $t_0^{(1)}$), the result would be equal to Equations (5.32) and (5.26), respectively.

Figure (5.5) shows the function (5.42), where $n = 30$, $X \in [-1; 1]$ and instead x^2 we used the expected value of X^2, $E\left(X^2\right) = 1/3$.

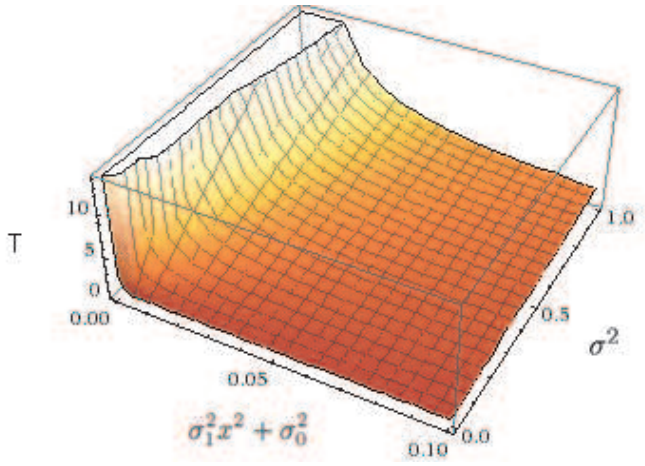

Figure 5.5: Optimal window size for linear regression with drifts in slope and intercept plus noise.

5.4 Estimation of the optimal window size

In this section we present an experimental evaluation of our theoretical analysis of the optimal window size. For this purpose we consider an artificial data set. The data were generated according to the models described in Sections 5.2 and 5.3.

For the linear model with drift and noise, different settings for σ_1, σ_0 and σ were used. In such a way we simulate the cases with different optimal window sizes. For each case the mean squared errors (MSE) for different values of T were computed. The computations were carried out for samples of size 10000 that were generated according to Equation (5.12). The mean squared error over 10 repeated runs is shown in figures 5.6, 5.7 and 5.8. Figure 5.6 shows the case with optimal theoretical window size of 2.5. The model has the following parameters: $\sigma_1 = 0.3$, $\sigma_0 = 0.3$ and $\sigma = 1$. This means that the drift and the noise have almost the same effect on the data, with the noise being slightly stronger than the drift. It is obvious that the empirical MSE-function in Figure 5.6 has a minimum point at $T = 3$ which is equal to the theoretical minimum. It is easy to see that using a window size different to the optimal size, the prediction error increases drastically. In that case, a window of size 25 will double the error.

Almost the same situation is shown in Figure 5.7. The data was generated with the following parameters: $\sigma_1 = 0.06$, $\sigma_0 = 0.06$ and $\sigma = 1$. Here the optimal theoretical window size is 12, which is also the minimum point for the empirical error function. By using a window size different from 12, the mean squared error

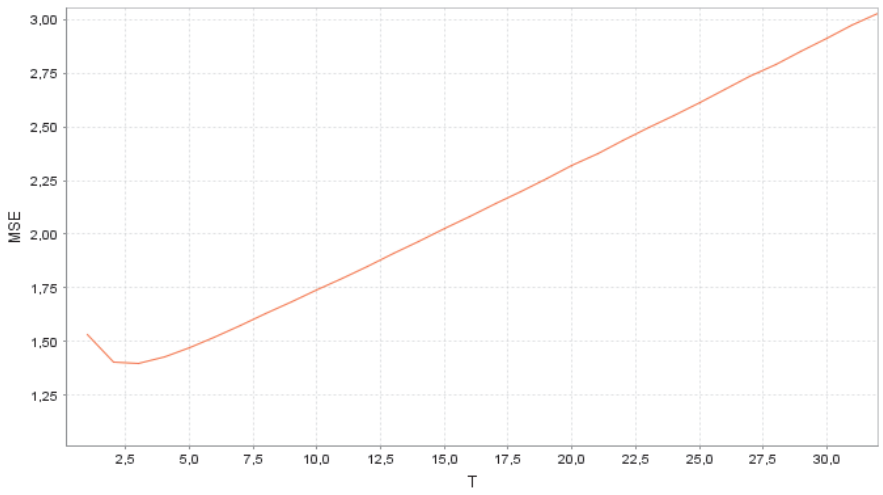

Figure 5.6: MSE of a linear model depending on the window size with optimal window size $T = 3$.

increases clearly. The situation is even more dramatic when only the last cycle of the sampled data should be used ($T = 1$).

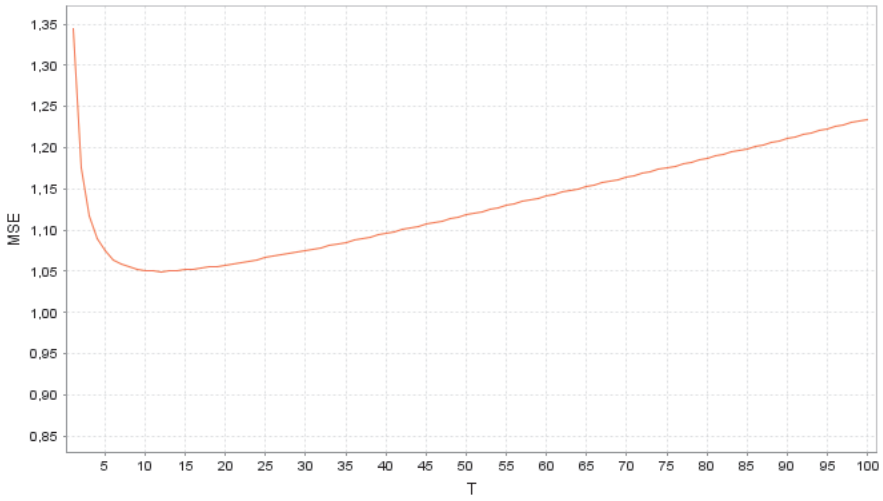

Figure 5.7: MSE of a linear model depending on the window size with optimal window size $T = 12$.

The situation in Figure 5.8 is different. Here the model has the following settings $\sigma_1 = 0.005$, $\sigma_0 = 0.005$ and $\sigma = 1$. Therefore the drift is negligible in comparison to the noise. This is also reflected in the optimal size of the data window $T = 144$. As we can see in Figure 5.8, the empirical error function has a plateau like minimum at the point $T = 144$, so that a slight change of the window size does not have much effect on the MSE.

Also for the constant model with drift and noise described in Section (5.2),

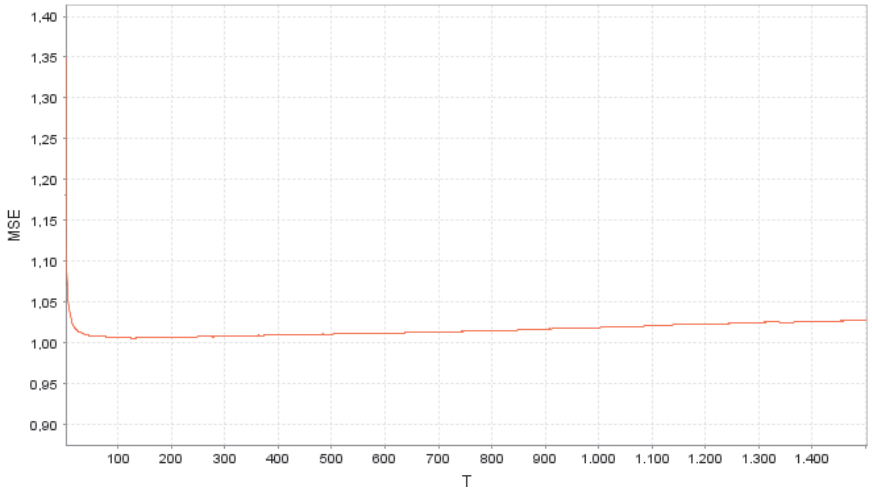

Figure 5.8: MSE of a linear model depending on the window size with optimal window size $T = 144$

the data with different σ_1 and σ_2 was generated and the mean squared errors were computed. The achieved results for this model are similar to the results described above for the more complex linear model and therefore are not presented here.

5.5 Consequences for non-stationary meta-models

In the previous sections we have assumed that the we have a non-stationary data generating process. But the meta-models were assumed to be stationary, i.e. the random drifts and the noise were assumed not to change over time. We would not have phases with higher noise or lower drift. Now we are interested in the following question: which effect would a non-stationary meta-model have on the quality of prediction? In this section we discuss some examples for non-stationary meta-models. As the underlying model we use the constant model described in Section 5.2.

$$\begin{aligned} Z_t &\sim N\left(y_t,\ \sigma_2^2\right), \\ Y_t &= \textstyle\sum_{i=1}^{t} X_i,\ X_i \sim N\left(0,\ \sigma_1^2\right) \end{aligned} \tag{5.43}$$

There are two parameters in the meta-model: the step size σ_1^2 for the random walk and the intensity of the noise σ_2^2. We assume that only one of these two parameters changes randomly. For reasons of simplicity, we will assume that the random values for the standard deviation are generated by a normal distribution. This can in principle lead to a negative standard deviation, but since we only make

use of the variance, i.e. the squared standard deviation in our models, this will not cause any problems.

For the first of our two non-stationary meta-models, we assume that the intensity of the noise (σ_2^2) remains constant, but the step size in the random walk changes randomly. Therefore, in each step the (signed) standard deviation σ_1 of the normal distribution for the random walk is changed randomly, following a normal distribution with expected value μ_{σ_1} and variance $\sigma_{\sigma_1}^2$.

$$\sigma_1 \sim N\left(\mu_{\sigma_1},\, \sigma_{\sigma_1}^2\right) \tag{5.44}$$

Z_t follows the same distribution as in Equation (5.43).

The second type of a non-stationary meta-model assumes the step size of the random walk to be constant, but the intensity of the noise to be changing over time. So for this model we assume

$$\sigma_2 \sim N\left(\mu_{\sigma_2},\, \sigma_{\sigma_2}^2\right). \tag{5.45}$$

Both kinds of non-stationary meta-models cannot be distinguished from a stationary meta-model with a new comparatively larger variance of the random walk or noise, respectively. Therefore, in fact, from such a non-stationary meta-model we can easily come to a stationary meta-model and all we need, is to estimate the optimal window size for the already known stationary meta-model with unknown variances.

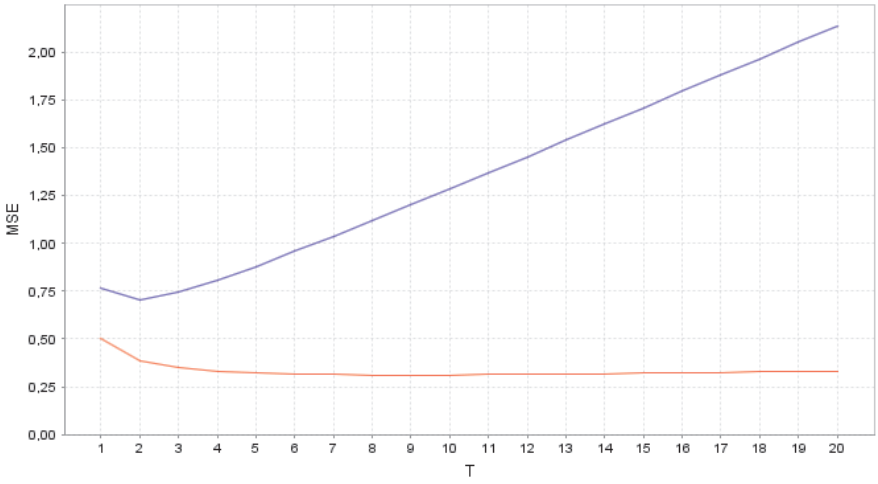

Figure 5.9: MSE curves for non-stationary (5.44) and stationary meta-models.

The MSE curves for the non-stationary (upper line) and the stationary (lower line) meta-models are shown in Figure 5.9. Here the following settings are used:

$\sigma_2 = 0.5$, $\mu_{\sigma_1} = 0.1$ and $\sigma_{\sigma_1} = 0.5$, therefore the change occurs in the meta-model for the random walk (5.44). The expected optimal window size for the stationary meta-model is $T = 9$, whereas the optimal window size of the non-stationary one is $T = 2$. The situation is similar for the changes of the variance of the noise (see Equation (5.45)). Here the optimal window size for the non-stationary meta-model is larger than the window size for the stationary meta-model. For sufficiently small variances $\sigma^2_{\sigma_1}$ and $\sigma^2_{\sigma_2}$, the optimal window size does not differ from the optimal size for the stationary meta-model. For instance, with $\sigma_{\sigma_1} = 0.03$ the optimal window size is exactly the same as for the stationary model. It is $T = 9$ in both cases.

As already mentioned before, such kind of non-stationarity is in fact a stationary meta-model. Therefore we consider the following non-stationary meta-model.

$$\begin{gathered} Z_t \sim N\left(y_t,\ \sigma^2_{2,t}\right), \\ Y_t = \textstyle\sum_{i=1}^{t} X_i,\ X_i \sim N\left(0,\ \sigma_1^2\right) \\ \sigma_{2,t} = \textstyle\sum_{i=1}^{t} X_i^{(\sigma_2)},\ X_i^{(\sigma_2)} \sim N\left(0,\ \sigma^2_{\sigma_2}\right) \text{ and } X_1^{(\sigma_2)} = \mu_{\sigma_2}. \end{gathered} \tag{5.46}$$

Hence the standard deviation of the random variables Z_i is the value of another random walk. Moreover, since the variance of this random walk tends to infinity with increasing t, we define a threshold, which should not be exceeded by this random walk.

This kind of non-stationarity could also be considered for the variance σ_1 of the original random walk. However first we restrict our considerations to the situation described in Equation (5.46).

Figure 5.10 shows the data generated by the process (5.46) with $\sigma_1 = 0.05$, $\mu_{\sigma_2} = 0.5$ and $\sigma_{\sigma_2} = 0.01$. Therefore in each step we have a different optimal window size T and according to this a different MSE curve (see Figure 5.11). Here we have computed the MSE curves for the next 5000 data points at different time points. For instance, the uppermost line shows the MSE curve for the first 5000 values, the lowest one for the data points with the indices between $25000 - 30000$. The second line from below represents the MSE curve for a stationary meta-model. As can be seen in Figure 5.11, each MSE curve has a different minimum. The minimum points are indicated in the figure by vertical dashed lines. Hence, instead of computing one MSE curve for all data as before, we analyze the time behaviour of the data. For this purpose we use the windowing

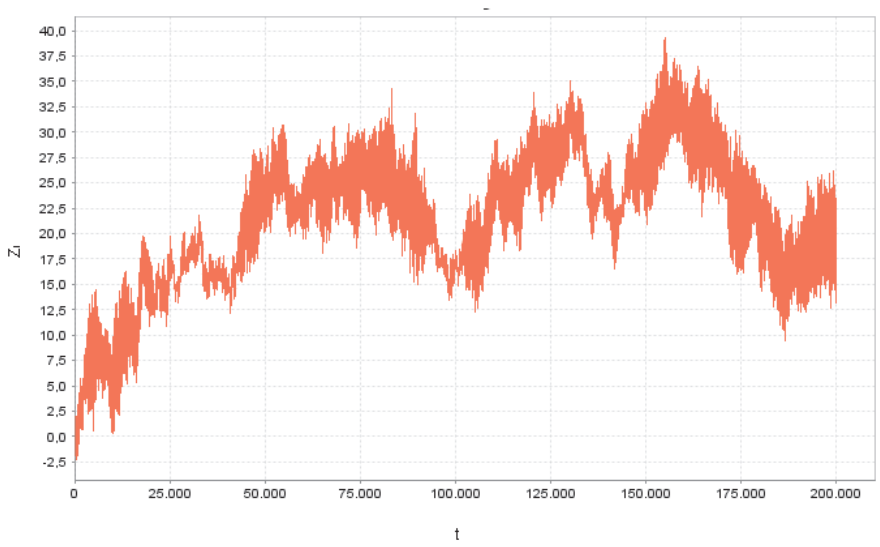

Figure 5.10: Data generated from the non-stationary meta-model in Equation (5.46)

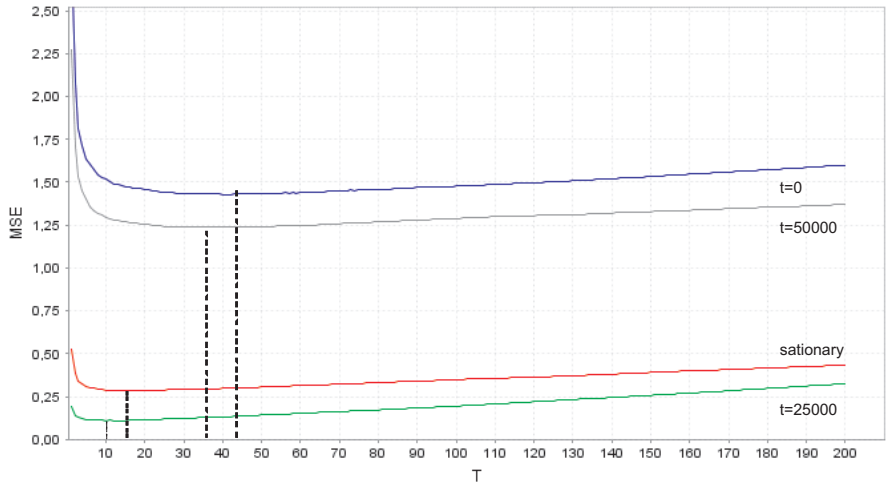

Figure 5.11: MSE curves for the non-stationary meta-model (5.46)

technique. The optimal window size for the stationary meta-model ($\sigma_2 = 0.5$) is $T = 17$. For time frame of size 5000, the MSE using $T = 17$ is computed. Afterward the window is moved and so forth. The measurements are shown in Figure 5.12. The highly fluctuating curve corresponds to the non-stationary meta-model and the almost constant one to the stationary. The MSE for the non-stationary meta-model is mostly larger than for the stationary, partly because of the larger variance of the noise, partly because the MSE of the non-stationary meta-model is computed with suboptimal T.

As Figure 5.11 shows, the quality of prediction could be improved by using

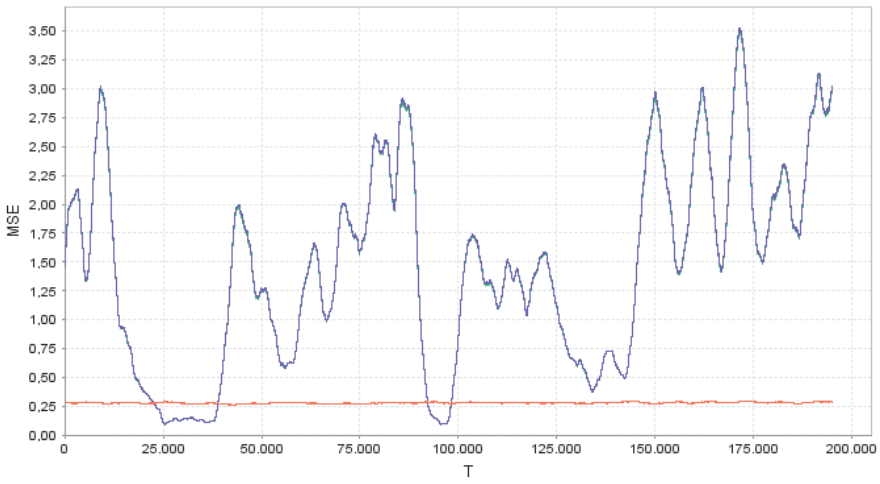

Figure 5.12: MSE for the non-stationary meta-model (5.46)

the optimal window size. For instance for the MSE curve for the time point $t = 0$ the optimal window size is 45. However this improvement is not very noticeable, since the MSE curves have very flat minima. Hence by using $T = 17$ we will gain slightly worse results.

Now we consider the situation where the use of the "wrong" window size could cause an even more drastic increase of the error.

$$\begin{gathered} Z_t \sim N\left(y_t,\ \sigma_2^2\right), \\ Y_t = \sum_{i=1}^{t} X_i,\ X_i \sim N\left(0,\ \sigma_{1,t}^2\right) \\ \sigma_{1,t} = \left|\sum_{i=1}^{t} X_i^{(\sigma_1)}\right|,\ X_i^{(\sigma_1)} \sim N\left(0,\ \sigma_{\sigma_1}^2\right) \text{ and } X_1^{(\sigma_1)} = \mu_{\sigma_1} \end{gathered} \tag{5.47}$$

For the model (5.47) we have the following settings: $\sigma_1 = 0.003$, $\mu_{\sigma_2} = 0.5$ and $\sigma_{\sigma_1} = 0.0001$. Therefore, the optimal window size for the corresponding stationary meta-model is $T = 289$. As Figure (5.13) shows using a constant window size of 289 will lead by some of the MSE curves to one rather poor prediction, since each MSE curve has a different minimum. The minimum point for stationary meta-model is indicated by vertical dashed line.

From Figures 5.11 and 5.13 it is obviously that for such kind of non-stationary meta-model the optimal window size changes in each step. Therefore T, estimated for the stationary meta-model might differ very much from the actual optimal window size. However, under the assumption that the parameters of the model are known and the values for σ_{σ_2}, σ_{σ_1} and the threshold value are small enough, it might still be a good alternative.

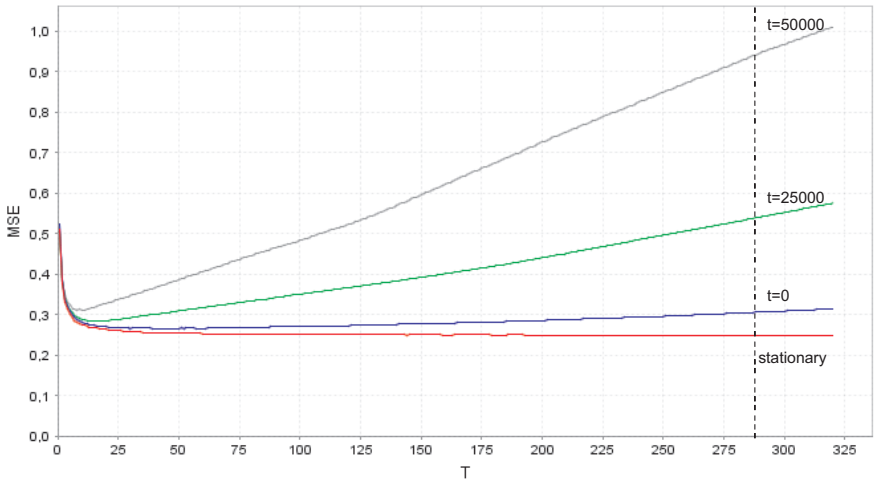

Figure 5.13: MSE curves

In this chapter we have proposed a theoretical analysis for optimal window size for the prediction of the next value. For that purpose two simple theoretical models for data generating process have been set up: a constant model with drift and noise and a linear model with drift and noise. For both models the minimum of the expected quadratic error has been computed and subsequently the optimal window size has been determined as a function of the data generating process parameters. With the help of such calculations we could analyse which effects the noise and drift have on the choice of the optimal window size. By stronger drift a smaller window (in the extreme case a window of size 1) should be chosen, whereas for noisy data more previous values should be used for prediction. Furthermore, similar to the approach from Chapter 4, both models presented in this chapter could be used as simple benchmark tests for an evolving system.

Chapter 6

Conclusion

The main focus of this work is the mining of data streams. Since the amount of the data collected in different areas of modern life increases with each day, this subject became more important during the last years. The approaches for data stream mining should fulfil specific requirements, which are not essential for instance for classic intelligent data analysis techniques. Time and space are the most important characteristics for the analysis of data streams. Since data is coming continuously and the amount of data is supposedly unlimited it is impossible to hold all data records in the memory. On the other hand efficient on-line computations are needed, which would make it possible to analyse data and to react according to the result of this analysis in real time.

In this work we presented important aspects and proposed new approaches for the mining of data streams. Thus in Chapter 2 we have introduced incremental computation schemes for statistical measures or indices like the mean, the variance or the Pearson correlation coefficient. Furthermore we have proposed a new algorithm for incremental or recursive quantile estimation for arbitrary distributions. The experimental results have shown that for continuous distributions our algorithm outperforms other approaches.

The efficient on-line computation of such statistical indices provides information about the characteristics of the probability distribution that generates the data stream. Although incremental computations are designed to handle large amounts of data, it is not extremely useful to calculate the above mentioned statistical measures for extremely large data sets, since they quickly converge to the parameter of the probability distribution they are designed to estimate as can be seen in Figure 2.1 and 2.2 in Chapter 2. Of course, convergence will only occur when the

underlying data stream is stationary.

Another crucial aspect for non-stationary data streams and therefore for evolving systems is change detection. It has been demonstrated in [43] that naïve adaption without taking any effort to distinguish between noise and true changes of the underlying sample distribution can lead to very undesired results. The advantage of using statistical tests compared to heuristic adaptation strategies is that we can distinguish between fluctuations due to the randomness inherent in the underlying distribution while it remains stationary and real changes of the distribution from which we sample. Applications of such change detection methods can be found in areas like quality control and manufacturing [26, 36], intrusion detection [46] or medical diagnosis [9]. Consequently it is very important to adapt statistical measures and hypothesis tests for the change detection in data streams. In Chapter 2 we derived the incremental computations of the χ^2-test and the t-test and presented a window based technique for change detection. Moreover in order to apply the algorithm iQPres from Chapter 3 in the context of non-stationary data streams, we have developed a statistical test for change detection tha t can be easily integrated into the iQPres algorithm.

The majority of the existing approaches for the mining of data streams is using sliding time window of fixed size. Mostly they either do not bother about the question which size of the window should be selected in order to get best results or apply heuristic methods for the choice of the amount of the data to be used for prediction. Hence in Chapters 4 and 5 we have proposed a theoretical analysis of effects of noise and changes in data for sliding window based evolving systems in order to illustrate the problem of suboptimal window size. For that purpose, simple theoretical models for data generating process have been set up: constant model with drift, constant model with drift and noise and linear model with drift and noise. For these models the smallest achievable expected quadratic error was computed and the optimal window size has been determined as a function of the data generating process parameters. In such a way we can analyse which effect has noise and drift on optimal window size. This can help us to better understand which problems can arise during the prediction of the next value of non-stationary noisy data. Furthermore such analysis demonstrates how important the correct choice of window size is if the window techniques are used for prediction. More-

over Chapter 5 provides the consequences of non-stationarity of the meta-model. Thereby different kinds of non-stationarity are considered.

Apart from the considerations above, such simple stochastic models could be used as benchmark tests that can give an idea of how much an evolving system might be misled by drift and noise. The simple switching model introduced in Chapter 4 could be used as benchmark test for evolving systems as well. This model can be interpreted as a regression or a classification problem. An experimental comparison between a maximum likelihood estimator exploiting the assumptions on the underlying data generating process and an evolving system without specific assumptions on the data generating process was carried out. Furthermore we strongly argue in favour to use benchmarks for evolving systems that are based on well-defined stochastic data generating processes. If only real world data are considered for benchmarks, it is not clear at all, how close an evolving system comes to the *unknown* best solution. With our simple stochastic models, we can explicitly provide or at least estimate the best solution that can be achieved from a theoretical point of view, so that we have a clear measure, how close the evolving system can come to the best solution.

Our examples were all restricted to the prediction of the next value. In terms of learning a function, the function was the identity function in the case of the random walks and a binary function – providing only two possible outputs – in the case of the switching model. To make the situation more complicated, we could replace the identity function or the binary output by another function. We could also consider a number of our models in parallel to have multiple inputs and aggregate them by a simple function for the output. We can also use more complicated models for the data generating process. But we should keep in mind that we need to have an idea, how well an optimized model could predict in order to have a comparison with the evolving system. Our future work in this area will focus on the theoretical analysis of non-stationary meta-models.

The main subject area of this work is univariate methods. First steps to extend change detection to multidimensional data can be found in [39].

Curriculum Vitae

Personal Data

Name:	Katharina Tschumitschew
Address:	Harzstr. 21, 38300 Wolfenbüttel, Germany
Telephone:	+49 5331 939 31320
E-mail:	katharina.tschumitschew@fh-wolfenbuettel.de
Date of Birth:	19 July 1980
Place of Birth:	Solnetschnodolsk, Russia
Citizenship:	German, Russian

Education

1987 - 1997	School education
1997 - 2002	Studies of Applied Mathematics at the Cuban State University of Krasnodar (Russia)
2002	M.Sc. degree in Applied Mathematics, Cuban State University of Krasnodar (Russia)
2006 - 2007	Studies of Computer Science at the University of Applied Sciences Braunschweig/Wolfenbüttel
2007	M.Sc. degree in Computer Science, University of Applied Sciences Braunschweig/Wolfenbüttel
since 2007	PH.D. student in the Department of Computer Science of the University of Applied Sciences Braunschweig/Wolfenbüttel

Work Experience

since 2004	Research Fellow in the Department of Computer Science of the University of Applied Sciences Braunschweig/Wolfenbüttel

Awards

2007	Best Master thesis by the Fachbereichstag Informatik (FBTI), Germany

Publications

- Katharina Tschumitschew, Frank Klawonn: Incremental statistical measures. In: M. Sayed-Mouchaweh, E. Lughofer (eds.) Learning in non-stationary environments: Methods and Applications, chap. 2. Springer, New York (2012).

- Katharina Tschumitschew, Frank Klawonn: Effects of drift and noise on the optimal sliding window size for data stream regression models (2011). Submitted for publication.

- Olga Georgieva, Katharina Tschumitschew, Frank Klawonn: Cluster Validity Measures Based on the Minimum Description Length Principle. KES (1) 2011: 82-89.

- Katharina Tschumitschew, Frank Klawonn: Incremental quantile estimation. Evolving Systems (2010): 1, 253-264.

- Katharina Tschumitschew, Detlef Nauck, Frank Klawonn: A Classification Algorithm for Process Sequences Based on Markov Chains and Bayesian Networks. KES 2010, 141-147.

- Katharina Tschumitschew, Frank Klawonn, Nils Obermller, Wolfhard Lawrenz: Visualisation of Test Coverage for Conformance Tests of Low Level Communication Protocols. KES 2010, 244-252.

- Katharina Tschumitschew, Frank Klawonn: The Need for Benchmarks and Meta-Models in Evolving Systems. In: P. Angelov, D. Filev, N. Kasabov: Proceedings of the International Symposium on Evolving Intelligent Systems (SSAISB). Leicester, 30-33

- Katharina Tschumitschew, Frank Klawonn: AVEDA: Statistical Tests for Finding Interesting Visualisations. KES 2009, 235-242.

- F. Klawonn, Detlef D. Nauck, K. Tschumitschew: Measuring and Visualising Similarity of Customer Satisfaction Profiles for Different Customer Segments. In: E. Corchado, X. Wu, E. Oja, A. Herrero, B. Baruque (eds.): Hybrid Artificial Intelligence Systems (HAIS 2009). Springer, Berlin (2009), 60-67.

- K. Tschumitschew, F. Klawonn, F. Hppner, V. Kolodyazhniy: Landscape Multidimensional Scaling. In: M.R. Berthold, J. Shawe-Taylor, N. Lavrac: Advances in Intelligent Data Analysis VII. Springer, Berlin (2007) 263-273.

- Vitaliy Kolodyazhniy, Frank Klawonn, Katharina Tschumitschew: A Neuro-Fuzzy Model for Dimensionality Reduction and its Application. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 15(5): 571-593 (2007).

- O. Georgieva, F. Klawonn, K. Tschumitschew: Noise Clustering via Dynamic Data Assignment Assessment. Proc. EUSFLAT 2005.

- F. Klawonn, K. Tschumitschew: A Classifier System-Based Learning Algorithm for Interpretable, Adaptive Nearest Neighbour Classifiers. In: Proc. 1st Workshop on Genetic Fuzzy Systems. Granada (2005), 64-67.

- F. Klawonn, K. Tschumitschew: Solving the Travelling Salesman Problem via a Fuzzified Objective Function. In: Proc. Recent Advances in Soft Computing, Nottingham 2004.

Bibliography

[1] Aho, A.V., Ullman, J.D., J.E., H.: Data Structures and Algorithms. Addison Wesley, Boston (1987)

[2] Angelov, P., Filev, D.: An approach to on-line identification of evolving takagi-sugeno models. IEEE Trans. on Systems, Man and Cybernetics, part B **34**(1), 484–498 (2004)

[3] Angelov, P., Lughofer, E.: A comparative study of two approaches for data-driven design of evolving fuzzy systems: ets and flexfis. Intern. Journal of General Systems **37**(1), 45–67 (2008)

[4] Basseville, M., Nikiforov, I.: Detection of Abrupt Changes: Theory and Application (Prentice Hall information and system sciences series). Prentice Hall, Upper Saddle River, New Jersey (1993)

[5] Beringer, J., Hüllermeier, E.: Efficient instance-based learning on data streams. Intell. Data Anal. **11**(6), 627–650 (2007)

[6] Chu, F., Wang, Y., Zaniolo, C.: An adaptive learning approach for noisy data streams. In: In ICDM, pp. 351–354 (2004)

[7] Crawley, M.: Statistics: An Introduction using R. Wiley, New Yourk (2005)

[8] Davies, P.L., Fried, R., Gather, U.: Robust signal extraction for on-line monitoring data. J. Statist. Plann. Inference **122**, 65–78 (2004)

[9] Dutta, S., Chattopadhyay, M.: A change detection algorithm for medical cell images. In: Proc. Intern. Conf. on Scientific Paradigm Shift in Information Technology and Management, pp. 524–527. IEEE, Kolkata (2011)

[10] Fischer, R.: Moments and product moments of sampling distributions. In: Proceedings of the London Mathematical Society, Series 2, 30, pp. 199–238 (1929)

[11] Fisz, M.: Probability Theory and Mathematical Statistics. Wiley, New York (1963)

[12] Ganti, V., Gehrke, J., Ramakrishnan, R.: Mining data streams under block evolution. SIGKDD Explorations **3**, 1–10 (2002)

[13] Gather, U., Schettlinger, K., Fried, R.: Online signal extraction by robust linear regression. Computational Statistics **21**(1), 33–51 (2006)

[14] Gelper, S., Schettlinger, K., Croux, C., Gather, U.: Robust online scale estimation in time series: A model-free approach. Journal of Statistical Planning & Inference **139**(2), 335 – 349 (2008)

[15] Grieszbach, G., Schack, B.: Adaptive quantile estimation and its application in analysis of biological signals. Biometrical journal **35**(2), 166–179 (1993)

[16] Groß, J.: Linear Regression: Vol 175 (Lecture Notes in Statistics). Springer, Berlin (2008)

[17] Gustafsson, F.: Adaptive Filtering and Change Detection. Wiley, New York (2000)

[18] Hayes, M.: Statistical Digital Signal Processing and Modeling. Wiley, New York (1996)

[19] Holm, S.: A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics **6**, 65–70 (1979)

[20] Hulten, G., Spencer, L., Domingos, P.: Mining time changing data streams. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (2001)

[21] Ikonomovska, E., Gama, J., Džeroski, S.: Learning model trees from evolving data streams. Data Mining and Knowledge Discovery **23**(1), 128–168 (2011)

[22] Ikonomovska, E., Gama, J., Sebastião, R., Gjorgjevik, D.: Regression trees from data streams with drift detection. In: 11th int conf on discovery science, LNAI, vol 5808, pp. 121–135. Springer, Berlin (2009)

[23] Kifer, D., Ben-David, S., Gehrke, J.: Detecting change in data streams. In: In Proc. 30th VLDB Conf., pp. 199–238. Toronto, Canada (2004)

[24] Klawonn, F., Angelov, P.: Evolving extended naive bayes classifiers. In: S. Tsumoto, C. Clifton, N. Zhong, X. Wu, J. Liu, B. Wah, Y. Cheung (eds.) Sixth IEEE International Conference on Data Mining: Workshops. IEEE, Los Alamitos, pp. 643–647 (2006)

[25] Klinkenberg, R.: Learning drifting concepts: Example selection vs. example weighting. Intell. Data Anal. **8**(3), 281–300 (2004)

[26] Lai, T.: Sequential changepoint detection in quality control and dynamic systems. Journal of the Royal Statistical Society, Series B **57**, 613–658 (1995)

[27] Lindstrom, P., Delany, S.J., Namee, B.M.: Handling concept drift in a text data stream constrained by high labelling cost. In: FLAIRS Conference (2010)

[28] Macias-Hernandez, J., Angelov, P., Zhou, X.: Soft sensor for predciting crude oil distillation side streams using takagi sugeno evolving fuzzy models. In: A. Famili, J. Kook, J. Peña, A. Siebes (eds.) IEEE International Conference on Systems, Man, and Cybernetics, Montreal, Canada (2007), pp. 3305–3310. Springer, Berlin (2007)

[29] Mann, H., Whitney, D.: On a test of whether one of two random variables is stochastically larger than the other. Annals of Mathematical Statistics **18**(1), 50–60 (1947)

[30] Möller, E., Grieszbach, G., Schack, B., Witte, H.: Statistical properties and control algorithms of recursive quantile estimators. Biometrical Journal **42**(6), 729–746 (2000)

[31] Nadungodage, C.H., Xia, Y., Li, F., Lee, J.J., Ge, J.: Streamfitter: A real time linear regression analysis system for continuous data streams. In: Database Systems for Advanced Applications, pp. 458–461 (2011)

[32] Nevelson, M., Chasminsky, R.: Stochastic approximation and recurrent estimation. Verlag Nauka, Moskau (1972)

[33] Nunkesser, R., Fried, R., Schettlinger, K., Gather, U.: Online analysis of time series by the $\mathscr{Q}_n$ estimator. Computational Statistics and Data Analysis **53**(6), 2354–2362 (2009)

[34] Pang, S., Ozawa, S., Kasabo, N.: Incremental linear discriminant analysis for classification of data streams. IEEE Transactions on Systems, Man, and Cybernetics – Part B **35**, 905–914 (2005)

[35] Qiu, G.: An improved recursive median filtering scheme for image processing. IEEE Transactions on Image Processing **5**(4), 646–648 (1996)

[36] Ruusunen, M., Paavola, M., Pirttimaa, M., Leiviska, K.: Comparison of three change detection algorithms for an electronics manufacturing process. In: Proc. 2005 IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 679–683 (2005)

[37] Shaffer, J.P.: Multiple hypothesis testing. Ann. Rev. Psych **46**, 561–584 (1995)

[38] Sheskin, D.: Handbook of Parametric and Nonparametric Statistical Procedures. CRC-Press, Boca Raton, Florida (1997)

[39] Song, X., Wu, M., Jermaine, C., Ranka, S.: Statistical change detection for multi-dimensional data. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 667–676. ACM, New York (2007)

[40] Spitzer, F.: Principles of Random Walk (2nd edition). Springer, Berlin (2001)

[41] Tschumitschew, K., Klawonn, F.: Aveda: Statistical tests for finding interesting visualisations. In: KES (1), pp. 235–242 (2009)

[42] Tschumitschew, K., Klawonn, F.: Incremental quantile estimation. Evolving Systems **1**, 253–264 (2010)

[43] Tschumitschew, K., Klawonn, F.: The need for benchmarks with data from stochastic processes and meta-models in evolving systems. In: N.K. P. Angelov D. Filev (ed.) International Symposium on Evolving Intelligent Systems. SSAISB, Leicester, pp. 30–33 (2010)

[44] Tschumitschew, K., Klawonn, F.: Effects of drift and noise on the optimal sliding window size for data stream regression models (2011). Submitted for publication.

[45] Tschumitschew, K., Klawonn, F.: Incremental statistical measures. In: M. Sayed-Mouchaweh, E. Lughofer (eds.) Learning in non-stationary environments: Methods and Applications, chap. 2. Springer, New York (2012)

[46] Wang, K., Stolfo, S.: Anomalous payload-based network intrusion detection. In: E. Jonsson, A. Valdes, M. Almgren (eds.) Recent Advances in Intrusion Detection, pp. 203–222. Springer, Berlin (2004)

[47] Weisberg, S.: Applied Linear Regression (Wiley Series in Probability and Statistics). Wiley, Hoboken, New Jersey (2005)

[48] Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin **1**, 8083 (1945)

[49] Zliobaite, I.: Combining similarity in time and space for training set formation under concept drift. Intell. Data Anal. **15**(4), 589–611 (2011)