Time Series Clustering in the Field of Agronomy

Cluster Analyse agronomischer Zeitreihen
Master-Thesis von Irina Alles
September 2013

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Department of Computer Science
FG Language Technology

Time Series Clustering in the Field of Agronomy
Cluster Analyse agronomischer Zeitreihen

Vorgelegte Master-Thesis von Irina Alles

1. Gutachten: Prof. Dr. Chris Biemann
2. Gutachten: Florent Masseglia

Tag der Einreichung:

Erklärung zur Master-Thesis

Hiermit versichere ich, die vorliegende Master-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den September 25, 2013

(I. Alles)

Abstract

In recent years, the agricultural sector has been subject to many technological advances. Farmers do not necessarily have to base their decisions only on proper observatory skills any longer. They can be supported by systems, analysing the satellite images of their fields and data obtained from sensors mounted on agricultural vehicles. This is called precision agriculture or satellite farming, and in most cases aims at maximising the obtained amount of yield. Considering the growing world population, in order to assure a stable food supply, not only an optimal yield performance but also plants adapted to the given conditions are of high importance.

The branch of phenotyping addresses this challenge by studying the genotype-phenotype interaction, in order to obtain plants perfectly adapted to specific environments and climate conditions. Entire plant stands are analysed during their growth in so-called high throughput phenotyping platforms. The collected amount of data is large and requires appropriate analysis techniques.

The domain of data mining focuses on the discovery of knowledge in large datasets and already provides well-established tools, which have proven their utility in various domains such as Finance, Biology and Medicine.

This thesis examines the appropriate data mining techniques for the present case of phenotyping, shows the benefit of automatic outlier detection and clustering of agronomic time series to the phenotyping community and addresses promising future directions.

Zusammenfassung

Die Landwirtschaft erfuhr in den letzten Jahren eine große Anzahl technologischer Erneuerungen. Landwirte müssen sich bei ihrer Entscheidungsfindung nicht mehr nur auf eigene Beobachtungen und die langjährige Erfahrung verlassen, sondern können durch Softwaresysteme unterstützt werden. Die sogenannten Decision Support Systems (DSS) berücksichtigen Ergebnisse aus der Bildanalyse der Satellitenbilder der Felder und weitere meist durch Sensoren ermittelte Daten. Diese Verfahrensweise nennt sich Precision Farming und hat oft zum Ziel die optimale Nutzung landwirtschaftlicher Nutzflächen und somit auch einen optimalen Ertrag.

In Anbetracht der wachsenden Weltbevölkerung ist eine gesicherte Nahrungsmittelversorgung von höchster Bedeutung, ein optimaler Ertrag und an die Bedingungen angepasste Pflanzen bilden dafür die Grundbausteine. Um an gegebene Umwelteinflüsse angepasste Pflanzen zu erhalten befasst sich der Bereich der Phänotypisierung mit Studien zu Phänotyp und Genotyp Wechselwirkungen. Dafür werden Merkmale ganzer Pflanzenbestände während ihres Wachstums in Phänotypisierungplattformen gemessen. Die resultierenden Datensätze sind groß und benötigen für die Analyse eine passende Herangehensweise.

Data Mining widmet sich der Analyse großer Datensätze und bietet bereits anerkannte Methoden die bereits in Bereichen wie Finanzwesen, Biologie und Medizin ihren Einsatz fanden.

Diese Arbeit untersucht die Tauglichkeit unterschiedlicher Data Mining Methoden für den Fall der Phänotypisierung. Wir zeigen in diesem Zusammenhang den Nutzen von Techniken wie Ausreißererkennung und Clustering, und diskutieren mögliche künftige Entwicklungen.

Acknowledgement

I want to sincerely thank Chris Biemann for giving me the opportunity to carry out this thesis, despite its deviation in subject and location. Thank you for providing me with the needed motivation, advice and help to face not only technical but also administrative challenges. I really appreciate the time and effort Chris put into the guidance of this work.

I want to thank Florent Masseglia for offering the possibility to write this thesis within his team at INRIA. I am grateful for his effort, support and advice that helped this work succeed.

I thank François Tardieu and Llorenç Cabrera-Bosquet for giving me insight into their work and their time and effort which made this collaboration work.

I want to thank Patrick Valduriez and his team Zenith for welcoming me and providing a great research environment.

I also want to thank my whole family and especially my parents, Tatjana and Viktor Alles, who always encourage me in my actions and offer support.

Finally I want to express my love and gratitude to Guillaume Tartayre, who not only motivated and helped me out during this challenging time, but also made me remember what counts most.

Contents

1 Introduction

> "Actualizing early insights by Lippmann (1922), Allport (1954) and Tajfel (1969a), the basic tenet of the social cognitive approach was that social information is much too complex to be dealt with satisfactorily. As a consequence, human information processors need to simplify the environment. Categorization offers a means to treat individual stimuli as instances of larger groups about which prestored knowledge is available."[Spears et al., 1997]

It is not without reason that the current age is referred to as the *Information Age*. We are constantly exposed to an incredible amount of information. We see on television and hear on the radio about local events and happenings in the world, we can read it on various newspapers, on different information websites, we get personalised updates from microblogging and social networks and we are left alone with the decision of what is important, what is trustworthy and what is worth seeing, listening or reading. Nowadays not only the social information requires a simplification to be satisfactorily dealt with but wee also need in our daily live broader categories in order to maintain an overview and make faster decisions.

Classification is not limited to the human subconsciousness, it is even one of the basic elements of scientific research. In Biology Aristotle created a decision system to classify the different animal species into similar groups. Beginning with the blood colour, red or not and adding the way the young are produced. In plant science Theophratos created the first classification of plants and their structure and in physics the main understandings of the atom were driven by the element classification of Mendeleyev in the 1860s. *E.g.*,[Everitt et al., 2011].

These classifications have been created with much effort and long manual work. Today a manual treatment of the incredible amount of data we are often dealing with is not imaginable. In case we already know the groups we want our data to be assigned to, we use classifications algorithms, which require already assigned example instances to be trained on. Once trained they classify the rest of the data automatically. If the classes are not known in advance or if we want to explore the inherent groups of our dataset, we use clustering algorithms which group elements into clusters, based on their high similarity.

We have seen that information appears in many various formats, as video, text, audio, images or measurements to name a few. All of them require an appropriate handling. In this work we focus on the latter category, named time series. Time series are measurements which where continuously taken in time. They appear in a great variety of fields: the financial sector, the probably mostly known time series is the Dow Jones; the field of seismology where seismograms measure the earth movement at a given time point; in biology, time series may represent the growth of an organism.

This thesis deals with time series in the domain of agronomy, more precisely plant phenotyping. Plant phenotyping is the analysis of the phenetic characteristics of plants. Often this implies the goal to understand the relation between the genetic assets of individuals and their phenotypic traits. Especially today the estimated world population of 7 billions [Weltbevölkerung, 2012] requires not only a stable but an increasing food supply. Therefore phenotyping needs to analyse a large amount of plant species in order to find the best performing or best adapted species to certain environments.

In this thesis we show that techniques well-established in the data mining community, such as clustering and outlier detection, are of great benefit for distant fields like phenotyping.

In Chapter 2 we provide an introduction to the format of time series, possible analysis goals and the applied methods. Also we discuss similarity measures applicable to time series data which are required for many further analysis tasks.

Chapter 3 outlines the context of this work by examining different aspects of phenotyping, its different goals and analysis methods. An important part of this section builds Section 3.2, which introduces the platform the data at hand comes from and describes the characteristics of our data.

In Chapter 4 we underline the importance of outlier detection and present common approaches and techniques used to detect outliers in various datasets. Finally we describe our approach and its evaluation.

Starting from an outlier-free dataset in Chapter 5, we identify different categories of clustering algorithms and present own results and their evaluation. At the end of this chapter we discuss a subspace clustering approach which aims at finding genetic marker relevant for a certain growing behaviour.

Finally in Chapter 6 we summarise and conclude our work giving suggestions for further improvements and possible investigations.

2 Time Series

This chapter serves as introduction to time series and outlines the related tasks and challenges especially in the domain of data mining. Topics of particular interest of the here presented work as time series representation and similarity measures are discussed in broader detail.

A time series is a sequence of points measured successively in time. The most obvious example for a time series is probably the Dow Jones or the development of certain stock prices. An increasingly large part of worlds data is in the form of time series [Maimon and Rokach, 2005]. But not only the economy and financial sector produce a large amount of such data. Social media platforms and messaging services record up to a billion of daily interactions [Piro, 2009] which can be treated as time series. Besides the high dimensions of this data, the medical and biological sector provide a great variety of time series, as gene expression data, electrocardiograms, growth development charts and many more. Example time series for stock prices and search trends are shown in Figure 2.1.

Although statisticians have worked with time series for a century, the increasing use of temporal data and its special nature have attracted the interest of many researchers in the field of data mining [Fu, 2011, Maimon and Rokach, 2005].

2.1 Time Series Data Mining

We have seen that time series can have a variety of sources as weekly sales and stock prices in the financial sector, daily temperature, earth movements, development of organisms in fields such as meteorology, seismology or phenomics, just to name a few. The analysis objectives can be as diverse as the data sources and formats. Time series as seismograms or the Dow Jones are mostly analysed with the goal to predict its evolution for the next days based on previous observations, in order to forecast earthquakes or unprofitable economic trends. But they can also be investigated to extract interesting or surprising trends in order to explore and understand their cause.

Before we dive deeper into this topic it is essential to clarify certain terms which will be used throughout this work. In the following we adopt the definitions stated by [Ding et al., 2008] and [Esling and Agon, 2012]:

Definition Time Series: *A time series T of length n is a sequence of pairs*

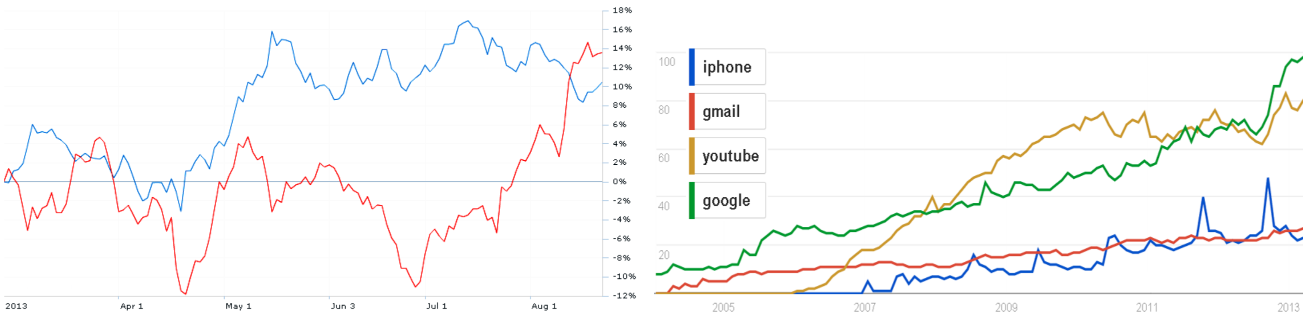
$$T = [(p_1, t_1), (p_2, t_2) \cdots (p_i, t_i) \cdots (p_n, t_n)] \mid\mid (t_1 < t_2 \cdots < t_i < \cdots < t_n)$$

where each p_i is a data point in a d-dimensional space and each t_i represents the point in time when p_i was measured. If the relevant time-series share the same sampling rates, the time stamps can be omitted and the time series can be regarded as an ordered sequence of d-dimensional data points. Nevertheless we will keep the time stamps as the dataset at hand does not provide equal sampling conditions.

Definition Representation: *The representation of a time series T with length n is a model $\overline{T}$ with reduced dimensions, so that $\overline{T}$ approximates T.*

Definition Similarity Measure: *A similarity measure $D(T, U)$ between the series T and U is a function that takes two times series as input and returns their distance d.*

From the data mining perspective time series analysis is often divided in the following categories: [Fu, 2011, Maimon and Rokach, 2005, Keogh and Kasetty, 2003]



(a) (b)

Figure 2.1: Time series examples. Image 2.1a shows the development of stock prices for 6 months, beginning Feb. 25.2013 for Google in blue and Apple in red, taken from `http://www.nasdaq.com/`. Image 2.1b illustrates the development of world wide search trends for the time frame 2004 – 2013, issued from `http://www.google.com/trends/explore`

Indexing Indexing enables the querying of content from a storage pool. Given a query time series Q the goal is to retrieve similar time series from a collection using a similarity measure. This can be either used to only match entire sequences or retrieve similar subsequences. It is used to respond to tasks as 'Find products with similar price patterns' or more complex as 'Find seismic subsequences differing from sequences resulting from geological irregularities' [Agrawal et al., 1995]. Besides its usefulness for exploratory analysis similarity search acts a great part in further data mining tasks as clustering and classification [Chakrabarti et al., 2002]. To speed up sequence retrieval and to deal with the high dimensionality of this data, dimensionality reduction techniques are applied and the results are often stored in index structures adapted for the given representation, such as the R-tree proposed by Guttman [1984] or the F-index from Agrawal et al. [1993]. For more details on different index structures see [Fu, 2011]. Section 2.2 gives an overview of techniques to reduce dimensionality.

Prediction Given a time series T, the prediction or forecasting task aims to predict the next data points based on the evidence extracted from previous points. This is one of the most applied time series task [Esling and Agon, 2012]. Prediction methods can be divided into two classes, linear and non-linear. Linear methods estimate the future values based on a linear combination of past and present values. The parameters for such a combination can be estimated by optimising an error function such as Gaussian Least Squares. Real world data rarely provides a linear relationship, thus requires non-linear models. The task of time series prediction can be considered as a supervised learning approach, taking a range of past and present values of a series as input vector and the future points as target values. This view has enabled the use of supervised learning techniques for time series forecasting in the non-linear case, such as Multilayer Perceptron (MLP) and Support Vector Machines (SVM) [Sapankevych and Sankar, 2009, Esling and Agon, 2012]. Moreover, Barreto [2007] shows that unsupervised techniques can be suitable for this task as well and outlines the use of Self-Organizing Maps (SOM) to tackle this problem.

Clustering Clustering is the most common method applied to the mining task of pattern discovery [Fu, 2011]. It aims to separate the given elements into 'natural' groups so that these elements are similar under a given similarity measure while elements from different groups should be highly different. In other words, those groups should minimise intra cluster variance while maximising inter cluster variance. The obtained clusters can then be used as basis for further investigations.

Classification While clustering aims at finding the *naturally* present groups in a dataset, classification creates a mapping from given time series to classes, which are predefined in advance. This approach requires a training set, a dataset with input time series and their class assignment. It is used to

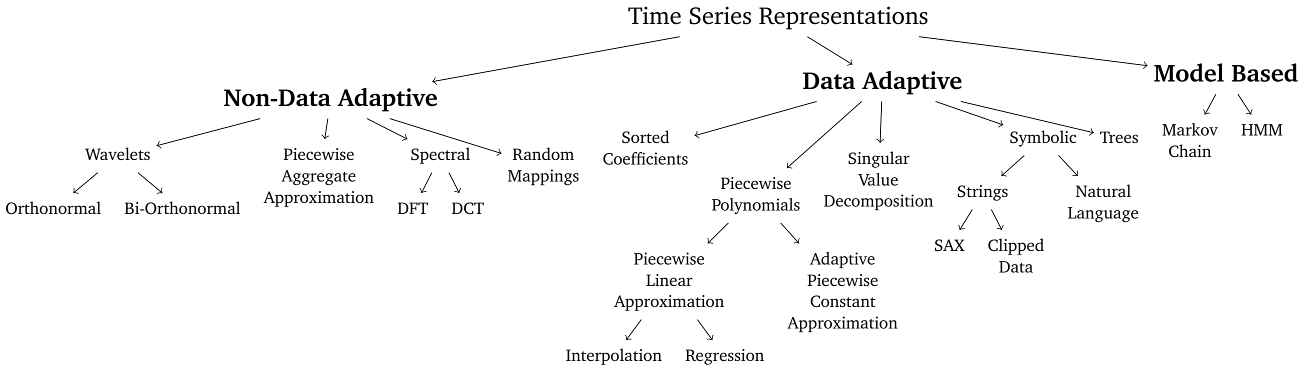


Figure 2.2: Hierarchy of time series representations found in the literature [Lin et al., 2003, Maimon and Rokach, 2005, Ding et al., 2008, Esling and Agon, 2012]

learn the distinctive features which determine the class affiliation in order to assign new unlabelled time series to the most appropriate class, based on those features.

Segmentation The task of time series segmentation, given a time series T, containing n points, is to produce the *best* representation using only K segments, where $K << n$. '*Best*' representation can be defined so that K are internally homogeneous sections [Maimon and Rokach, 2005]. This problem is known as change point detection, Guralnik and Srivastava [1999] describe an algorithm to detect such points which then can be mined for 'interesting episodes', Yamanishi and Takeuchi [2002] present an on-line algorithm for the detection of change points data streams. Another '*best*' representation can be such that K segments reduce dimensionality while retaining the most characteristic features of the time series [Esling and Agon, 2012]. This is also referred to as summarisation, it creates a high level representation of the data which can be beneficial to following mining tasks. Further representations and approaches to cope with high dimensional time series are discussed in Section 2.2.

Sometimes one can find further task groups in the literature, such as *anomaly detection* and *motif discovery* (see [Maimon and Rokach, 2005, Esling and Agon, 2012]). Given a labelled *normal* time series T and another unlabelled time series the goal of anomaly detection is to determine all sections containing unexpected events. The usual approach is similar to the prediction task, which is to build a model from the *normal* series and to flag sequences as anomalies that appear too far from the expectation [Esling and Agon, 2012]. Another option presented by Salvador et al. [2004] is to create time-point clusters representing the normal points of a series. A further task often found in the literature is motif discovery [Patel et al., 2002, Esling and Agon, 2012, Chiu et al., 2003]. It could be regarded as a specification of the segmentation or summarisation task due to their similarities. Its goal is the detection and enumeration of *motifs* in a long time series. Patel et al. [2002] define motifs as reoccurring mutually exclusive subsequences of a time series. They state that motif discovery is in particular useful to summarise and visualise large time series collections. Further Keogh and Lin [2005] have shown that clustering of time series subsequences produces essentially random clusters and thus is meaningless. They propose motif discovery to find meaningful clusters.

2.2 Representations

Many time series datasets are large and high dimensional. Lets take an audio stream as example. Adopting the settings as used by BBC in the UK [Courtice, 2010], sampling rate at 44.1KHz and 16 bits per sample using two channels. A two-hour long audio stream results in more than 1,2 GB of data to be analysed. Accessing each point, in cases of naive approaches even multiple times, is computationally too expensive, therefore analysis are often performed not on the raw data itself, but on a more abstract

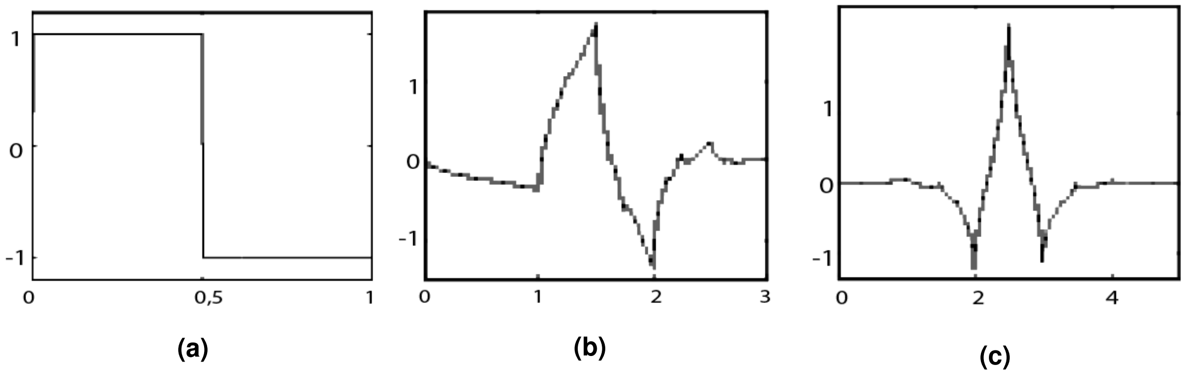


Figure 2.3: The commonly used wavelet shapes. (a) represents one of the first used wavelets, the Haar wavelet. (b) is the Daubechies and (c) the coiflet.

representation. This has the advantage of requiring less space, speeding up calculation procedures and implicit noise reduction [Esling and Agon, 2012]. Nevertheless, we have to keep in mind that despite the listed benefits the reduction of dimensionality of a time series will highly influence the outcome of further processing. We can not expect an analysis performed on a bad approximation to yield results of same quality as one would obtain from an analysis on the original data. Therefore, an dimensionality reduction algorithm has to fulfil certain requirements:

- It has to maintain the local and global shape characteristics of the time series.
- It should be computationally efficient in order to enable its application to large datasets.
- An insensitivity to noise is desirable, but it could also be handled by a preprocessing step.
- And even if it is obvious it should effectively reduce the data's dimensionality.

In the last decade, plenty approaches have been proposed to tackle this problem. Figure 2.2 outlines the different categories proposed by [Ding et al., 2008, Maimon and Rokach, 2005] and some of their methods. Keogh and Kasetty point out that the high number of very different approaches, the lack of comparison and testing on real datasets results in confusion and contradictory claims, such as "*wavelets outperform the DFT*", "*DFT filtering performance is superior to DWT*" and "*DFT-based and DWT-based techniques yield comparable results*", see [Keogh and Kasetty, 2003]. Thus they addressed this issue by re-implementing eight different dimensionality reduction methods and evaluated their performance on different datasets. The results show that the power of reduction and thus the indexing effectiveness of the different methods have about the same performance on the different datasets. However, Chakrabarti et al. [2002] argue that beside the prune power, further traits have to be considered when choosing a representation, such as its suitability for indexing or the supported similarity measures. Therefore, we aim to give an intuition for the variety of representations and will discuss the three main categories of dimensionality reduction techniques outlining the most relevant methods.

2.2.1 Non data-adaptive representation

Non data-adaptive techniques use the same set of parameters for dimensionality reduction regardless of the underlying data. One of the early works on this topic was achieved by Agrawal et al. [1993], who used the Discrete Fourier Transform (DFT).

This algorithm is based on the idea that any time sequence can be expressed as a superposition of sine or cosine waves. It projects a time series into the frequency domain, by decomposing the series into sinusoidal waves which are represented by complex coefficients, the Fourier coefficients. [Agrawal et al., 1993] observed that only the first few waves appear to be dominant and therefore the rest can be omitted without any great impact on the reconstruction error. Thus the final time series representation after DFT are the coefficients of the first k waves. Agrawal et al.'s experiments on synthetic data showed that two coefficients are sufficient for good results in the retrieval of similar series. A very important property of DFT for data mining applications is Parseval's Theorem. It states that the total energy of a signal in the time domain is preserved in its projection into frequency space [Shatkay, 1995, Keogh and Pazzani, 2000b]. Disregarding the error introduced by the truncation at k, this means that the euclidean distance will hold the same for the original signal as its transformation [Keogh and Pazzani, 2000b]. This responds greatly to our needs for a good time series representation, the reduction of dimensionality is implied by the usage of k complex coefficients. Further the Parseval's Theorem provides that the relation, hence the distance between series is preserved and DFT can be calculated efficiently with O($n\ log\ n$)[Agrawal et al., 1993]. A concern pronounced by Keogh and Pazzani [2000b] is that the coefficient truncation of positive terms at k causes the distance in the frequency space to be less than the truth distance, resulting in false positives in applications such as similarity search.

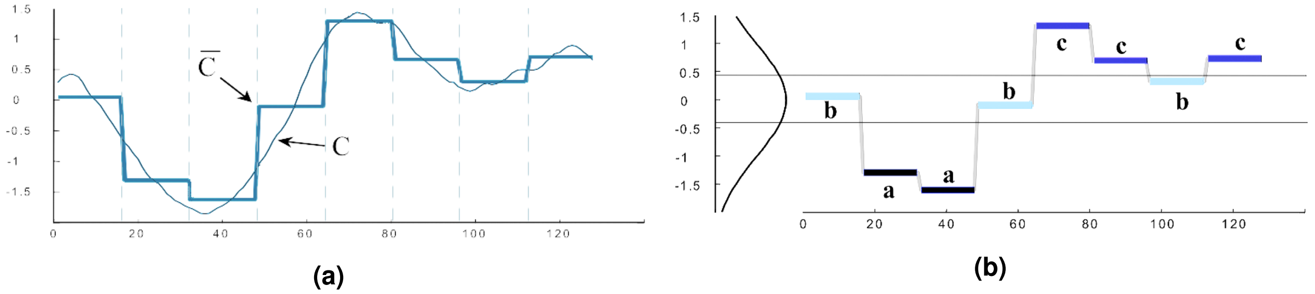
Another very related approach is the Discrete Wavelet Transform (DWT). While DFT uses sinusoidal waves to represent the general shape of a time sequence, DWT processes the series at different scales and resolution. In contrary to DFT, DWT uses localised wavelets of final energy to represent the data. A mother wavelet defines the overall shape and further analysing wavelets derived through shift and scaling add the necessary details to the representation. There is a variety of functions which can be used for DWT. Chan et al. [2003] showed that DWT using a Haar wavelet can be calculated and indexed efficiently. The characteristics of the transform can be controlled by the choice of the mother wavelet as all further wavelets derive from it [Sripath, 2003]. Some common wavelet shapes are shown by Figure 2.3. One drawback is that classical DWT is only defined for sequences with length of powers of two [Maimon and Rokach, 2005], which can be overcome by zero-padding, smooth-padding or periodic extension. Although there are contradictory claims on the performance of DWT (see [Keogh and Kasetty, 2003]), Wu et al. [2000] underlines that DWT's superiority lies in its time complexity of $O(N)$ and the multilevel resolution.

A completely different approach, especially targeting the domain of time series, is the Piecewise Aggregate Approximation (PAA) independently proposed by [Keogh et al., 2001, Yi and Faloutsos, 2000]. The very simple idea appears to be competitive in comparison to the more sophisticated transforms [Maimon and Rokach, 2005]. The idea is to segment a time series T of length n into N consecutive sequences of same length. Then the mean is calculated for each of those sequences resulting in a new representation of N mean value points. Keogh et al. [2001] show that PAA supports comparison of series with different lengths and supports the Euclidean distance measure.

2.2.2 Data-adaptive representation

This category of time series representations assembles techniques which take into account the underlying data and adjust their parameters accordingly. Almost any non-data adaptive approach can become data adaptive by adding a parameter selection step [Esling and Agon, 2012]. Vlachos et al. [2004], Struzik and Siebes [1999] realised this idea for DFT and DWT.

As DFT and DWT, Singular Value Decomposition (SVD) is another transformation-based approach. The important difference to the afore mentioned transforms is that while DFT and DWT apply local transformations, SVD acts globally [Keogh et al., 2001]. This means that the other transforms process one data point at a time. The resulting transformation is independent of the rest of the data. Whereas SVD examines the entire data and rotates the axes to maximise variance along the first few dimensions [Ravi Kanth et al., 1999]. The resulting representation consists of the first few dimensions. Although SVD is an optimal transformation in the sense of minimal reconstruction error [Keogh et al., 2001], it requires the computation of eigenvalues for large data matrices making it computationally very expensive [Esling



(a)

(b)

Figure 2.4: Figure *(a)* illustrates the PAA method. C represents the original time series and $\bar{C}$ is the PAA approximation using the averages of eight subsequences of equal length. Figure *(b)* shows the SAX approximation based on the PAA results from *(a)*. The separation of the distribution space into segments of equal probability, *a,b* and *c*, is depicted on the left of this image. The final SAX representation of the here presented time series is *baabccbc*.

and Agon, 2012, Chakrabarti et al., 2002].

In Chakrabarti et al. [2002], they propose an improved and data adaptive version of PAA, called Adaptive Piecewise Constant Approximation (APCA). While PAA stores the means of consecutive fixed length segments, APCA allows the segments to be of different length, thus more adapting to the data. This means that a region of low activity can be represented by one long segment and regions with high activity are depicted by several short segments. The final representation stores two numbers per segment: its mean and the segment length. Following in terms of dimensionality reduction, PAA with N segments corresponds to APCA with $N/2$ segments. Despite of the coverage of fewer segments, [Chakrabarti et al., 2002] showed that APCA performed at least as good as PAA and often better, in terms of quality measured by the reconstruction error.
This now gives rise to the question of how to determine the best possible segmentation for APCA. Faloutsos and Jagadish [1997] state that finding the optimal piecewise polynomial representation for a time series using dynamic programming requires $O(Mn^2)$ time. Chakrabarti et al. [2002] therefore propose a method achieving an almost optimal representation for APCA using $O(n\ log(n))$ time. Taking into account that Haar wavelet transforms can be obtained in $O(n)$ [Wu et al., 2000], they transform the problem into a wavelet transformation task and convert the resulting coefficients back to APCA representation.

A completely different approaches to dimensionality reduction is the conversion of time series into sequences of symbols. This implies the discretisation of the series, its segmentation and finally a mapping to an alphabet of symbols. Lin et al. [2003] propose an approach called Symbolic Aggregate Approximation (SAX), which is closely related to PAA. The main idea is to use a PAA representation as intermediate step between the raw data and the resulting symbolic sequence. The distribution space (y-axis) is divided into regions of equal probability and each region is associated with a symbol. The PAA sequences falling into those regions are mapped to the corresponding symbol as illustrated by Figure 2.4b. Thus the final representation is the string of successive symbols. This approach have been shown to be competitive in tasks such as time series classification and clustering [Keogh et al., 2004].

2.2.3 Model-based representation

Approaches of the *model based* category assume that a given time series was produced by an underlying model. Dimensionality reduction is obtained by representing the time series by the model's parameters, used to produce the series. As a consequence time series similarity is measured based on the model parameters [Esling and Agon, 2012]. There are several approaches using parametric temporal models such as statistical modeling via feature extraction [Nanopoulos et al., 2001] or the ARMA and ARIMA models, see [Kalpakis et al., 2001]. More sophisticated approaches include Markov Chains or Hidden Markov Models (HMM) [Panuccio et al., 2002]. The objective of those approaches is often not the explicit reduction of dimensionality but rather the improvement of similarity distances for further tasks such as clustering or classification [Kalpakis et al., 2001, Panuccio et al., 2002].

Given this variety of representations, it is a complex task to choose the best approach for a given context, as each approach has its special properties which might be inconvenient in one case but a virtue in another. Although Ding et al. [2008] showed that all eight compared representations yield about the same performance, there are slight differences depending on the dataset. Their findings are that spectral methods such as DFT are good at representing highly periodic datasets and APCA performs significantly better on datasets containing bursts.

2.3 Distance Measures

Almost every data mining task requires a notion of similarity between objects, based on their shape. Such as clustering, classification or indexing. While human intuitively disregard disturbing aspects like amplitude scaling, time shifts, noise and outliers, these factors complicate the task to find a distance measure reflecting the human intuitive perception of similarity. Obviously, which distance measure fits best depends on the context but the following characteristics appear to be desirable in most scenarios, that involve time series:

- It should be consistent with human intuition.
- Perceptually similar objects should be classified correspondingly even if they are not mathematically identical.
- It has to take into account the local and global shape characteristic of a series.
- And finally probably one of the most important points is that it should be almost insensitive to noise, outliers and different transformations between series. Transformations such as shifts in amplitude and scaling.

A further criteria especially important to indexing tasks is the lower bounding lemma introduced by Faloutsos et al. [1994].

$$D_{index}(T,S) \leq D_{true}(T,S) \tag{2.1}$$

This means that in order to avoid false dismissals the distance measure should never overestimate the true distance between some time series T and S. In the following we will present the L_p norms, which despite of their simple nature, especially the Euclidean distance, are very often applied in data mining tasks [Ding et al., 2008]. A further category of similarity measures are elastic measures, elastic in the sense that in contrast to L_p norms which compare point i of a series T with point i of S, they allow a one-to-many comparison [Esling and Agon, 2012].

Besides the different characteristics of each distance measure, Ding et al. [2008] underlines that the reported measure accuracy and speed has to be regarded with care. The size of the dataset the measure is evaluated on plays an important role. They have shown that as datasets get larger, the speed of elastic measures approaches the speed of simple approaches such as the Euclidean distance and the accuracy of the Euclidean distance approaches that of elastic measures. This means that the superior aspects of a measure diminish or even disappear with the growing size of a dataset. With this aspect in mind we will regard the mostly used distance measures in data mining, Euclidean distance and Dynamic Time Warping (DTW) in more detail.

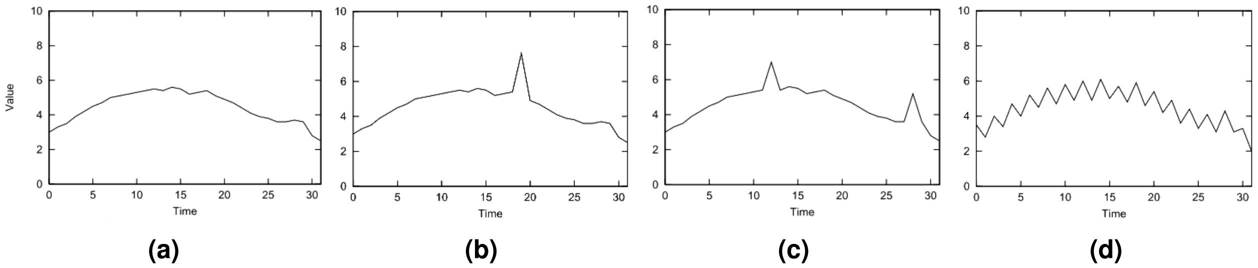


Figure 2.5: These time series were adjusted in order to outline the similarity notions of the different L_p norms. A nearest-neighbour search has been performed for the time series in Figure (a). The use of L_1 obtained time series (b) as nearest-neighbour, L_2 (c) and L_∞ (d).

2.3.1 L_p-Norms

The literature offers a large number of similarity measures, the L_p norm is one of the most popular classes and is defined as

$$L_p(T,S) = (\sum_{i}^{N} |t_i - s_i|^p)^{\frac{1}{p}} \tag{2.2}$$

where T and S are time series of length N and t_i is the measurement of T at time point i. This holds correspondingly for S and s_i. The variable p denotes the norm in use. $p = 1$ is the Manhattan distance, $p = 2$ represents the well-known Euclidean distance and $p = \infty$ the maximum or Chebyshev distance. Yi and Faloutsos [2000] outline the different aspects of this norms very well using an example depicted in Figure 2.5. All series illustrated in this figure are of length 32. Figure 2.5a is the original series, to Figure 2.5b they added a peak of 2.5 units height and to 2.5c two additional bursts of 1.5 units height. In 2.5d they added and subtracted 0.5 units alternately. The nearest-neighbour search for the series in 2.5a using the different norms shows, that L_1 chooses (b), L_2 (c) and L_∞ the series in (d) to be the most similar one. This illustrates [Yi and Faloutsos, 2000] statement that the L_1 norm is optimal when measurement errors are additive Laplacian as it is more robust against outliers. The Euclidean distance is the most widely used distance in similar time series matching [Agrawal et al., 1993]. Powerful feature extraction methods, such DFT and DWT, are only defined for the L_2 norm, due to the projection into frequency space the feature distances are only preserved for the euclidean distance [Keogh and Pazzani, 2000b]. However it does not inherently deal with challenges such as transformations along the y-axis. Two series could fluctuate in the same manner, but at different amplitude levels. This problem can be overcome using normalisation. Another point to consider is that such distances often fail to represent a shape similarity which is obscured by a misalignment in the time axis, see Figure 2.6a. This aspect has given rise to elastic methods discussed in the following section.

2.3.2 Elastic Measures

For most applications, simple distance measures such as L_p norms are sufficient. They are easy to implement, provide a low time complexity, are parameter-free and show good performances [Wang et al., 2012]. Nevertheless there are cases when the overall shape of two time series is similar, but one of them is accelerated or decelerated. Lets take recorded speech as an example of a time series. The same word spoken by two different speakers will produce time series similar in shape, but deformed by the speakers pace and intonation. In order to find the similarity and thus to achieve a better alignment, we have to 'warp' the time axis. This idea is illustrated by Figure 2.6b. The left part of the image shows a simple alignment of two perceptually similar time series using Euclidean distance, which will produce a rather high dissimilarity value due to its sensitivity to irregularities in the time axis. This issue is addressed by elastic distance measures.

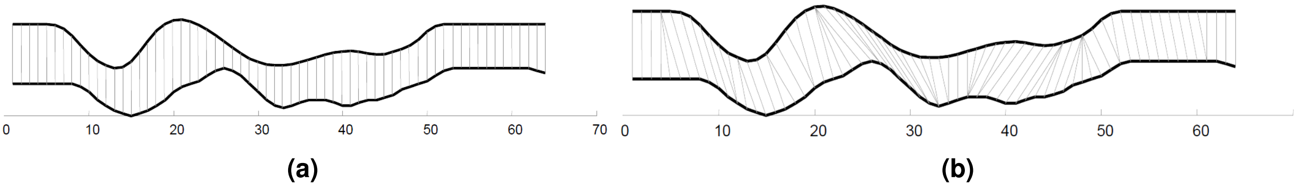


Figure 2.6: The left figure shows an alignment of two time series of similar overall shape. The Euclidean distance will give a high dissimilarity measure as it does not consider the misalignment in the time axis. The right side shows the same time series aligned using DTW. The non-linear alignment allows to provide more representative similarity measure.

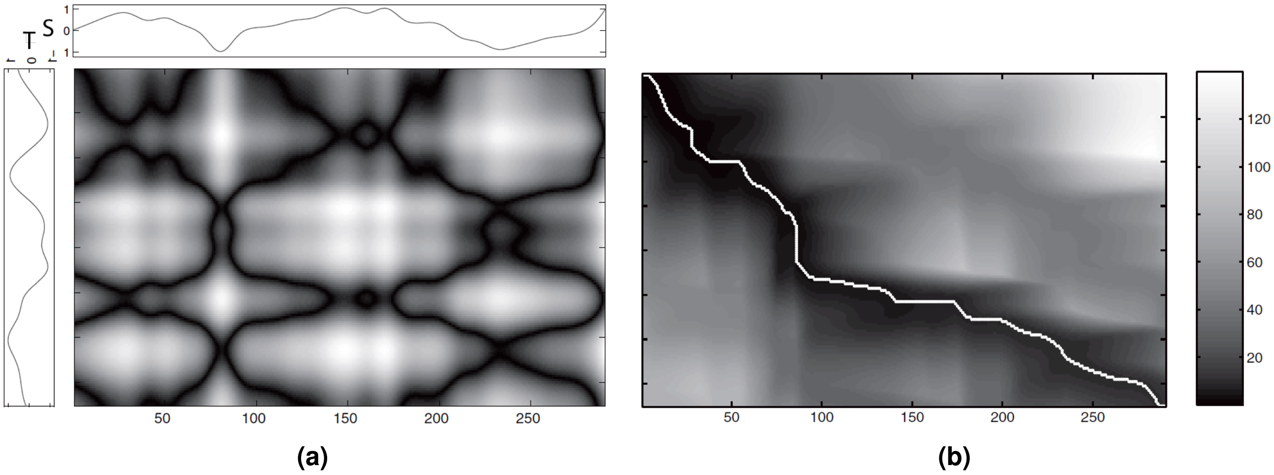


Figure 2.7: The left figure shows the two time series T and S on the left and top of the image. The matrix in the middle is the cost matrix C obtained by evaluating the local cost distance for each element pair of both series. Dark colours represent lower cost and light parts of the image represent high cost respectively. Figure 2.7b is the corresponding accumulated cost matrix. The optimal warping path is plotted in white, note that the path traverses only the darkest regions. These figures are inspired by the example used in [Müller, 2007]

Dynamic Time Warping

Dynamic Time warping (DTW) has originally been used to compare different speech patterns in automatic speech recognition (ASR) tasks [Sakoe and Chiba, 1978, Myers and Rabiner, 1981, Komori and Katagiri, 1992]. It was already a well-established tool in the domain of speech processing when Berndt and Clifford [1994] demonstrated its utility in the data mining domain. Since then it has been used in various tasks such as clustering, classification and anomaly detection [Keogh and Pazzani, 2000a, Tormene et al., 2009, Petitjean et al., 2011]. DTW's main characteristic, the non-linear alignment of time series, overcomes the weakness of the euclidean distance ,which is very sensitive to distortion in the time axis. Despite of its relatively high time complexity of $O(N^2)$ of the classic version it has been successfully applied in domains such as Bioinformatics [Aach and Church, 2001], Medicine [Tormene et al., 2009] and even entertainment [Zhu and Shasha, 2003]. But before we delve any deeper in its assets and disadvantages, we will explain the actual algorithm.

The classical DTW takes two time series T and S of length $N \in \mathbb{N}$ and $M \in \mathbb{N}$ and assumes equidistant samples. In order to compare two samples t_i and s_i, one requires a cost measure. This measure is referred to as *local distance measure*. An important characteristic of this measure is that it should assign a low cost to similar time points and a large cost otherwise. As far as they fulfil this requirement any of the afore mentioned distance methods can be used. Once we compute the cost measure $c(t_i, s_i)$ for each sample pair of both series, we obtain the cost matrix $C \in \mathbb{R}^{N \times M}$. An example for such a matrix is shown by Figure 2.7a, where dark regions represent lower cost. The main goal of DTW is to find the best alignment between the given time series. This now can be achieved by finding the path through the obtained matrix which minimises the overall cost. Such a path is called *warping path*. Müller [2007] defined an alignment as follows:

Definition Warping path: *A warping path for the time series T and S is a sequence $p = (p_i, \cdots, p_L)$ with $p_l = (t_l, m_l) \in [1:N] \times [1:M]$ for $l \in [1:L]$. Typically this path has to fulfill the following constraints*

1. *Boundary condition: $p_1 = (1,1)$ and $p_L = (N,M)$. This simply states that the path has to begin and to end in diagonally opposite corners of the matrix. For the alignment, it means that the first and last points of the two time series have to be aligned to each other.*
2. *Monotonicity condition: $n_1 \leq n_2 \leq \cdots \leq n_L$ and $m_1 \leq m_2 \leq \cdots \leq m_L$. This assures that the elements of the resulting warping path are monotonically increasing in time.*
3. *Step size condition: $p_{l+1} - p_l \in \{(1,0),(0,1),(1,1)\}$ for $l \in [1:L-1]$. This defines that only adjacent cells, including the diagonal cell, of a cost matrix are appropriate for the next step. It also implies that no elements are omitted and that there are no replications in the resulting path p.*

There are numerous possible paths in a cost matrix that satisfy the above stated conditions, but we are particularly interested in the optimal warping path p^* that minimises the total cost. The total cost of a path is defined as

$$c_p(T,S) = \sum_{l=1}^{L} c(t_{n_l}, s_{m_l}) \tag{2.3}$$

and the final DTW distance for the series T and S is the total cost of path p^*:

$$\begin{aligned} DTW(T,S) &= c_{p^*}(X,Y) & (2.4)\\ &= min\{c_p(T,S)|p \ is \ a \ warping \ path \ of \ \ T \ and \ S)\} & (2.5)\end{aligned}$$

In order to obtain the optimal warping path, the naive approach is to try any possible path, leading to an exponential computational complexity. Instead, one makes use of an *accumulated cost matrix D*, see Figure 2.7b. Defined as

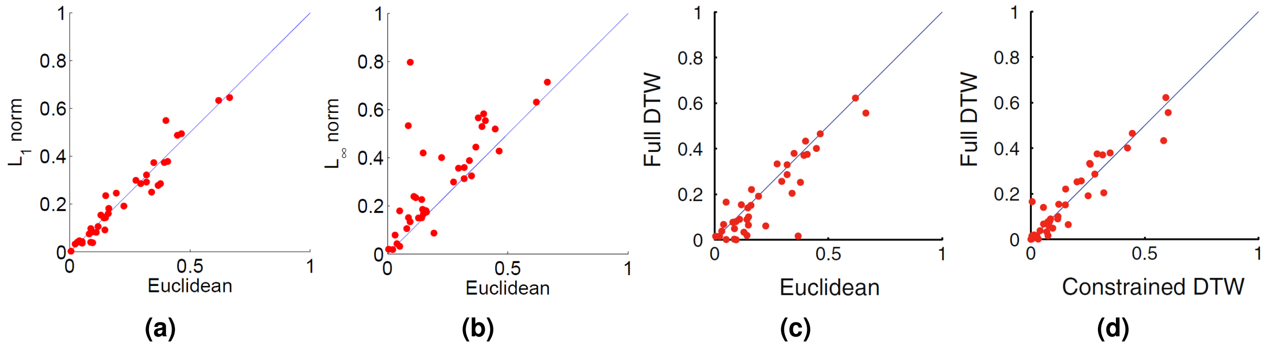
$$D(n,m) = DTW(T_{1\cdots n}S_{1\cdots m}). \tag{2.6}$$

This matrix can be computed efficiently by dividing the task into three sub-problems:

$$D(n,m) = min\{D(n-1,m-1), D(n-1,m), D(n,m-1)\} + c(t_n, s_m) \tag{2.7}$$

where $1 < n \leq N$ and $1 < m \leq M$. We refer to [Müller, 2007] for a proof and further details.

There have been many attempts to modify the classical DTW for speed up or to adapt it to the given context. One way to modify the warping path is the modification of the step size condition allowing to omit certain points. In order to favour horizontal or vertical alignments, it is possible to apply local weights. A well-known approach to speed up DTW is to impose global constraints on the admissible region for a warping path. It has been shown that it not only speeds up the computation but improves accuracy for measuring time series similarity by avoiding pathological paths Wang et al. [2012]. The well-known constraint shapes such as *Sakoe-Chiba band* which runs along the main diagonal and allows only usage of cells within a fixed width and the '*Itakura parallelogram*' are reviewed in [Ratanamahatana



(a) (b) (c) (d)

Figure 2.8: These figures are scatter plot representations of the distance measure benchmarks by [Wang et al., 2012]. The x and y coordinates of a dot are the error ratios of the distance measures in comparison. Each dot represents a dataset. In Figure 2.8a the data points above the diagonal line represent that L_2 was more accurate for this dataset than L_1. The further a point is from the line, the greater the improvement in accuracy. The more dots are on one side of the line means that the worse this measure performs in comparison to the other, like for the case in 2.8b where the most dots are on the side of L_∞, speaking for its inferior performance compared to L_2.

and Keogh, 2004]. Thuong and Anh [2012] review three lower bound measures which produce an envelope on the possible warping path and create a lower bound approximation of the final DTW distance. Such a measure should avoid false dismissals by providing measures lower or equal to the actual DTW distance and it should be fast to compute. If the bound is tight enough, it can speed up similarity search. Time series not similar to the lower bound can not be similar to the real series and can be pruned, thus avoiding the computation of the real DTW distance. A related way of speed up is to apply DTW to an approximated version of the data, as shown by Keogh and Pazzani [2000a] who worked on a PAA representation of time series.

In spite of the wide application of DTW, Ratanamahatana and Keogh [2005] claim that there are still myths about this algorithm causing confusion and resulting in papers solving problems that do not exist. They investigated the claim that DTW is particularly good at handling sequences of different lengths, and performed 1-nearest-neighbour classification on datasets with sequences of different length. They computed the classification once on the original different-length time series and once after interpolation of these series in order to get sequences of equal length. Their result is that there is no significant difference in accuracy between DTW on variable and equal-length series. Motivated by various work targeting the speed up of DTW, they report that using a good lower bound, so that the actual distance has to be rarely computed, “DTW is effectively $O(n)$, and not $O(n^2)$, when searching large datasets” [Ratanamahatana and Keogh, 2005]. Wang et al. [2012] performed an exhaustive benchmark of different similarity measures on 38 datasets. The pairwise comparison of DTW and the Euclidean distance is depicted by Figure 2.8. Their results underline that elastic measures outperform the Euclidean distance by a large percentage.

3 Phenotyping

Phenotyping is the phenotypic analysis of organisms. In order to fully capture the extent of this definition, it is important to clarify the term ’phenotype’ first. Already in 1909, Willhelm Johannsen made the distinction between genotype and phenotype [Johannsen, 1909] and further developed his thoughts in [Johannsen, 1911] by giving the following definitions:

Definition Genotype: *“A “genotype” is the sum total of all the "genes" in a gamete or in a zygote.”*[1]

Definition Phenotype: *“All “types” of organisms, distinguishable by direct inspection or only by finer methods of measuring or description, may be characterized as ”phenotypes”.”*

A phenotype is determined by its gens, the environment and a stochastic developmental variation. Developmental variation can favour or block the evolution of certain traits. This means that two organisms of identical genotypes raised in the same environmental conditions can result in different phenotypes [Vogt et al., 2008, Johannsen, 1909]. The biological literature shows a variety of further definitions for these terms. Lewontin (1992) defines phenotype and genotype as classes of organisms satisfying certain genetic or phenetic criteria [Mahner and Kary, 1997], whereas [Futuyma, 1986] refers to genotype as a blueprint of an organism containing the instructions for development and sees the phenotype as the manifestation of this blueprint, influenced by the environment. The most convenient definition seems to depend on the context, for example it might be appropriate for an analysis of DNA sequences to see genotype and phenotype as DNA and proteins whereas in the scope of phenotyping a broader definition is sufficient. For a more detailed discussion see [Mahner and Kary, 1997]. Further we will use the following definitions inspired by [Johannsen, 1911] and [Herskowitz, 1977]:

Definition Genotype: *A genotype is the genetic constitution of an organism.*

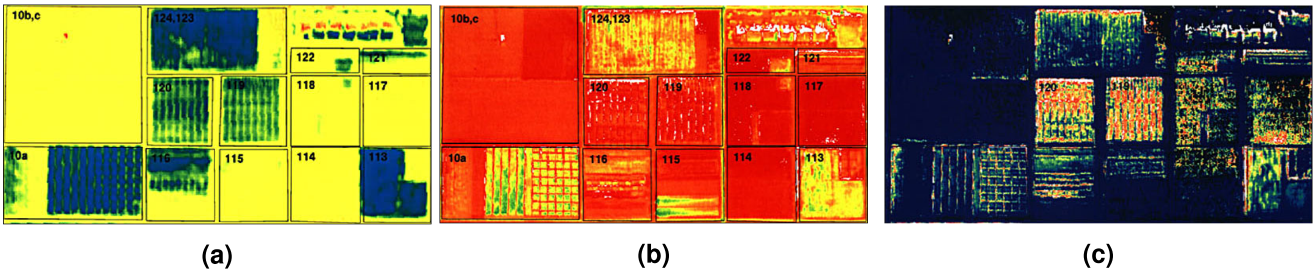
Definition Phenotype: *A phenotype is a collection of traits possessed by an organism that result from the interaction of the genotype and the environment, influenced by the developmental variability.*

As stated before, phenotyping is the analysis of the phenotype of an organism. The goal of this analysis are manifold. Gregor Mendel used observable traits to define and follow units of inheritance [Bochner, 2003]. Phenotyping enables the detection of genetic changes, which confer a growth or an advantage in an observable trait. As important as the advantageous traits are observations of their suppressors, revealing the genes a gene of interest interacts with [Bochner, 2003, Nakazawa et al., 2003]. Thus phenotyping enables the association of genotypes to their possible phenotypes. Phenotyping is used to study developmental variability or organism responses to environmental stimuli [Granier et al., 2006] and reveal genes associated with these responses [Nakazawa et al., 2003].

3.1 Plant phenotyping

This work is realised in the field of agronomy, and therefore, our focus lies on the phenetic analysis of plants. Plant phenotyping has been performed since the beginning of farming, when farmers started to select grasses with more desirable traits for propagation, for example with the goal to increase yield or to obtain more resistant grasses.

[1] A gamete is a cell fusing with another gamete during fertilisation in order to form a union, the zygote.



(a)

(b)

(c)

Figure 3.1: The images were taken by the Daedalus sensor aboard a NASA aircraft flying over the Maricopa Agricultural Center in Arizona. Figure 3.1a shows the colour variations determined by crop density where dark blues and greens indicate lush vegetation and reds show areas of bare soil. The 3.1b image is a map of water deficit, derived from the DaedalusŠ reflectance and temperature measurements. Green and blue points indicate wet soil and red points show dry soil. Figure 3.1c shows where crops are under serious stress, as is particularly (indicated by red and yellow pixels).[Moran, 2001]

Today the growth of the world population of more than 70 Millions yearly [Weltbevölkerung, 2012] requires an adequate increase in food supply, and the additional climate change adds a further challenge. This means we need a continuous improvement of cultivars, optimised for higher yield or stress tolerant plants able to survive in harsh environments. This would allow the cultivation of further land which is rarely used for farming due to its climate or soil conditions. Thus, an efficient and targeted selection is required. Breeding programs face a considerable number of challenges, the number of species to be improved is large and the traits to select for are diverse and often complex. In addition, cultivars are often required to adapt to a broad range of environments. As a consequence, for certain species it requires up to 20 years to produce novel cultivars with the desired traits [Walter et al., 2012]. The improvement of phenotyping techniques by adoption and combination of technologies from different fields as remote sensing, image analysis and spectroscopy aims to speed up this process by several years.
Currently there are two developing tracks: Phenotyping directly on the fields often called, field monitoring or precision agriculture, and phenotyping platforms in the laboratories.

Precision agriculture aims at measuring the inter and intra-field variability. There are two approaches: the map-based approach and the sensor-based approach. The map-based approach uses images of the field taken from a satelitte or an aerial imager. These images in combination with laboratory analyses of soil samples are used to identify the field conditions as shown in Figure 3.1. For the sensor-based approach, sensors are mounted on tractors and the desired properties are measured 'on the go'. Agricultural harvesters equipped with near-infrared spectroscopy devices can capture physical and chemical characteristic of the harvested material [Montes et al., 2007]. Montes et al. [2006] showed that those devices can reveal amount of dry matter, starch and crude protein contents in corn grain. Another non-invasive phenotyping approach is the measurement of spectral reflectance of the plant canopy. Those sensors are mounted on tractors and traits as canopy architecture, water status and nitrogen concentration are captured in the spectra. In order to reveal those phenotypic values, the image undergoes of course a further analysis step using a calibration model [Montes et al., 2007]. Imaging in the field faces several issues. "Variable illumination, dissected, reflecting plant canopies, altered spectral composition of the sunlight in different weather conditions, plant movements due to wind or rain, and many other factors complicate the retrieval of quantitative information from pictures in the field" [Walter et al., 2012]. In order to reveal significant information about the performance of plants in a certain environmental context, environmental parameters have also to be recorded throughout the experiment. Nevertheless, experiments designed for specific environmental conditions optimise statistical relevance of the collected data. Hence, phenotyping experiments are more successful in the laboratory or greenhouse [Walter et al., 2012] as new phenotyping platforms enable the control of most conditions.

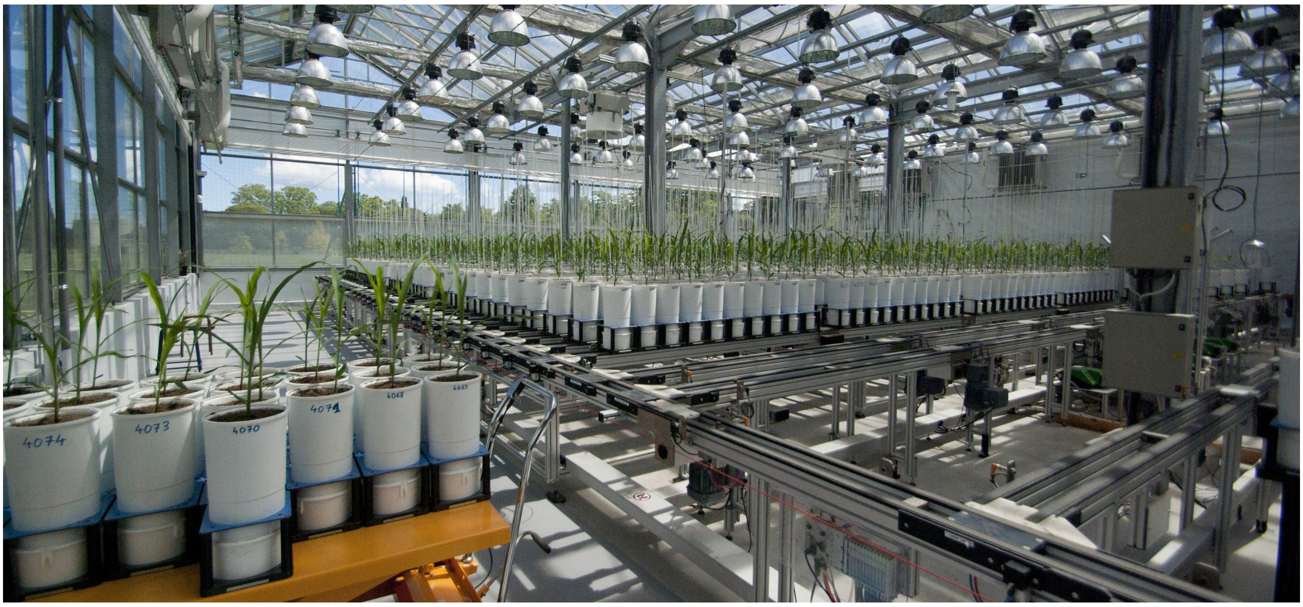


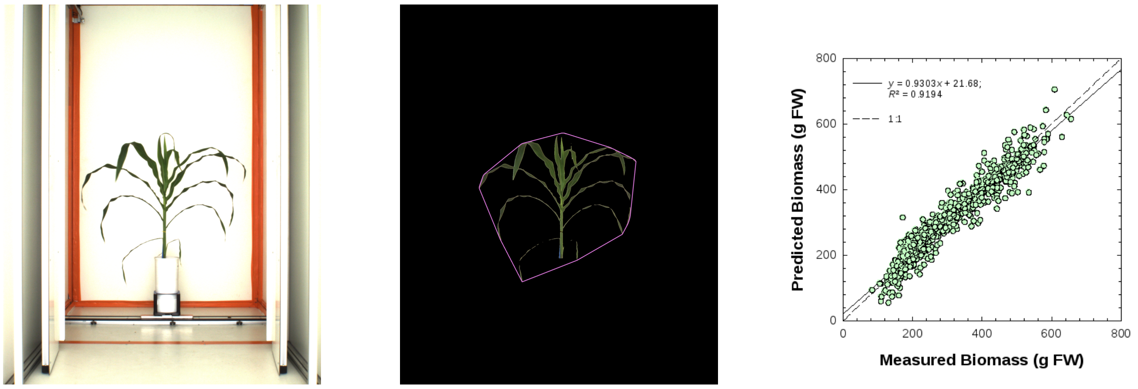
Figure 3.2: The PhenoArch platform

The combination of several techniques, as in the case of sensor-based field monitoring, have brought laboratory phenotyping platforms to a new level. Those techniques now enable the detection of several traits in laboratory grown plants while maintaining a high throughput of more than 1000 plants per day [Granier et al., 2006, Rajendran et al., 2009]. Phenotyping platforms in the greenhouse often have individual pots for each plant and therefore offer a very exact control of most environmental conditions. This is important for the investigation on the genotype-environment interaction. In order to be certain that a plant reaction is due to an environmental change, only this variable of interest should be modified, as for example the amount of watering to analyse different genotype reactions to drought. A widely used concept of advanced phenotyping is to determine phenotypical traits as plant height, total leaf area, leaf number of canopy width from colour pictures of individual plants taken from different angles [Walter et al., 2012]. Therefore, plants are either automatically delivered to a camera system, or the camera is placed at a defined orientation towards the plant. This method depends heavily on further image processing steps, in order to reveal the desired traits, and the quality of these measurements depends on the accuracy of those techniques.

Thus the progress in advanced phenotyping relies on one hand on throughput, the ability to process a large number of genotypes and on the other hand on phenotyping methods, such as automated imaging and near-infrared spectroscopy. Most notably data management and data analysis techniques play a vital role in this domain. Only via automatic and systematic analysis of plant images and spectra the phenetic values can be revealed and used for biologic investigations.

3.2 PhenoArch

PhenoArch is a greenhouse phenotyping platform hosted by the LEPSE group (Laboratoire d'Ecophysiologie des Plantes sous Stress Environnementaux, INRA) in France, Montpellier, shown by Figure 3.2. The platform is designed to analyse genetic characteristics of plant responses to environmental conditions, in particular drought, temperature and light. This platform allows the measurement of plant architecture, leaf area, plant volume/biomass and transpiration rate. It has a throughput of 1650 pots, each placed on its own cart on a conveyor belt, is equipped with two imaging systems for leaves and for roots using near-infrared and contains further sensors to measure temperature, light, humidity and so forth.



(a) (b) (c)

Figure 3.3: Figure 3.3a shows a picture of a corn plant taken by the imaging system from one angle. 3.3b is the resulting image after separation from the background and Figure 3.3c depicts the correlation of predicted biomass verses the measured biomass after harvesting. These images are taken from [Tardieu, 2013]

3.2.1 Environmental measures

To keep track of environmental conditions, light, air temperature and Vapour Pressure Deficit (VPD) are stored every 15 minutes and are measured at 6 positions in the greenhouse at plant level.

Definition VPD: *"Vapour pressure deficit (VPD_{air}) is the difference (deficit) between the amount of moisture in the air (e_a) and how much moisture the air can hold when it is saturated. The maximum water holding capacity also called the dew point, e_{sat}(Ta) increases with temperature. Adding moisture to air beyond it leads to deposition of water (dew)"[Poiré and Tardieu, 2013a].*

The measurement of VPD is particularly useful as not only indicates humidity of the air but takes also into account the influence of temperature on water holding capacity of the air. Higher VPD indicates increase in transpiration, influencing how much plant moisture trails of into the air [Poiré and Tardieu, 2013a]. Further maps of light distribution and VPD, created at several times of a season, allow a precise environmental characterisation. For eight plants distributed according to these maps the organ temperature is measured using thermocouples. For further details and discussion see [Poiré and Tardieu, 2013a,b, Tardieu, 2013].

3.2.2 Soil water status

Especially for experiments on drought resistance, soil water status is one of the most important measures. Therefore plants are weighted up to four times a day and the soil water content is adjusted to the desired amount. [Tardieu, 2013] describes its determination by using the following equations.

$$water\ volume_{soil} = weight_{current} - (weight_{pot} + weight_{dry\ soil} + weight_{plant\ estimated}) \qquad (3.1)$$

$$water\ content = \frac{water\ volume_{soil}}{weight_{dry\ soil}} \qquad (3.2)$$

3.2.3 Plant dimensions

The measurement of plant biomass and leaf area is realised by taking 3 images of each plant at different angles. This images undergo a further image analysis procedure to separate the actual plant from its background. Then the biomass estimation is done via a calibration model, mainly based on the number of green pixels in the images. Once the plants are harvested, this model is evaluated and improved against the measured biomass and leaf area. These steps are illustrated by Figure 3.3

Name	Year	Dent Genotypes	Tropical Genotypes
ZC	2011	30	30
ZA	2012	-	200
ZB	2012	250	-
ZA	2013	250	-

Figure 3.4: Zea mays experiments carried out in the PhenoArch platform between 2011 and 2013

3.2.4 Thermal Time

Montes et al. [2007] argues that phenotypic traits are often treated as static and therefore are only measured once, but for the analysis of genes and gene networks that are active at different development phases and in order to record responses to environmental stress [Wu and Lin, 2006], it is important to keep this dynamic nature. This requires consecutive trait measurements at regular intervals. Unfortunately, when climate conditions are not strictly stable, the resulting time courses have to be analysed individually for each experiment and each day. This is due to the major influence of temperature on development processes. Starting from a minimum temperature threshold, the increase in temperature accelerates enzyme activity, standing for faster growth and development. This proceeds until a maximum temperature where the enzyme coagulates and the new structure is not able to catalyse the reaction [Bonhomme, 2000]. Therefore it is desirable to obtain temperature independent measurements - the thermal time. Thermal time is commonly used to model development of crop species [Granier, 2002]. Sadok et al. [2007] showed that it is also suitable for the analysis of several corn populations at short time steps of 15 minutes. Thermal time is applicable if the rate of the studied process is proportional to the plant organ temperature, than a linear relationship can be integrated over time [Granier, 2002]. Sadok et al. [2007] proposed the equation 3.3 to express temperature independent leaf length at any time (t):

$$L = a \underbrace{\int_0^t [T(t) - T_0]\, dt}_{thermal\ \ time} \tag{3.3}$$

where a is the slope and T_0 x-intercept of the relationship between the rate of leaf growth and temperature, also called termed threshold temperature. For further discussion on this topic see [Sadok et al., 2007, Wu and Lin, 2006, Bonhomme, 2000].

3.3 Data

The here analysed data results from experiments, called 'Manip's, on corn plants (*Zea mays*). They were carried out between 2011 and 2013 using the PhenoArch platform. Analysis mostly used inbred lines of tropical species and dent corn hybrids. The experiments and their configuration are listed by table 3.4. Tropical corn is a short day plant therefore the long summer days in Europe or North America cause it to grow taller delaying or completely blocking its flowering. This results in a higher sugar content making it especially suitable for the production of bio-fuel [Bant, 2007]. Dent corn is a variety of corn with a high soft starch content used as base in the food production. It is one of the most grown varieties in the United States [University of Missouri, 2013].

In order to improve statistical power of the analysis, there are 5 to 6 repetitions for each genotype. A repetition means a plant with an identical genotype. Thus depending on the experiment the data contains measurements for up to 1679 plants, meaning 1679 time series. Time Series are available for Biomass and Leaf Area measured as described in Section 3.2. The here presented analysis mainly used the ZB 2012 dataset, which contains about 250 different corn genotypes. Experiments in the PhenoArch

Pot	Analysis.Time.Stamp	Genotype	Scenario	Manip	Day	Hour	TT	Manip.RatioPHPW	Manip.LeafArea	Manip.Biomass
1	05/06/2012 06:17	F98902H	WW	ARCH2012-05-14	2012-05-25	15:29:31	17.51166016	2.53164556962025	0.0141519751692189	0.91164564746026
1	25/05/2012 15:34	F98902H	WW	ARCH2012-05-14	2012-05-25	15:29:31	17.51166016	2.53164556962025	NA	NA
1	27/05/2012 19:41	F98902H	WW	ARCH2012-05-14	2012-05-27	19:36:42	20.76672389	1.41114982578397	0.0192612130944422	NA
1	05/06/2012 06:18	F98902H	WW	ARCH2012-05-14	2012-05-27	19:36:42	20.76672389	1.41114982578397	0.0192612130944422	NA
1	05/06/2012 06:16	F98902H	WW	ARCH2012-05-14	2012-06-02	20:03:47	30.75733846	0.915492957746479	0.0464625033820251	21.5517678164719
1	02/06/2012 20:08	F98902H	WW	ARCH2012-05-14	2012-06-02	20:03:47	30.75733846	0.915492957746479	0.0315316333832821	NA
1	05/06/2012 06:16	F98902H	WW	ARCH2012-05-14	2012-06-04	20:59:31	34.07325002	1.10897435897436	NA	NA
1	06/06/2012 21:52	F98902H	WW	ARCH2012-05-14	2012-06-06	21:47:54	37.62373219	0.983067729083665	NA	NA
1	09/06/2012 19:51	F98902H	WW	ARCH2012-05-14	2012-06-09	19:46:54	42.68614203	0.959967974379504	0.121457165557066	91.3964570945188
1	13/06/2012 20:37	F98902H	WW	ARCH2012-05-14	2012-06-13	20:32:13	49.31769013	1.09633357296909	0.202595138769	185.941826481552
1	15/06/2012 19:34	F98902H	WW	ARCH2012-05-14	2012-06-15	19:29:31	52.57734602	1.10164141414141	0.276142734010348	244.456926936159
1	25/06/2012 16:26	F98902H	WW	ARCH2012-05-14	2012-06-17	21:46:30	56.06656592	1.10641989589358	0.32219205744672	315.12355613503
1	25/06/2012 21:08	F98902H	WW	ARCH2012-05-14	2012-06-19	22:04:11	59.43565629	1.09155645981689	0.378155445133032	376.985246584148
1	26/06/2012 01:12	F98902H	WW	ARCH2012-05-14	2012-06-22	13:53:03	64.71139101	1.37075718015666	0.468423778797991	471

Figure 3.5: This figure shows data for a F98902H genotype time series of the ZB 2012 experiment. The scenario denotes the water conditions, where WW stands for well-watered, TT is the thermal time and RatioPHPW the ratio of plant hight and plant width.

platform focus on plant responses to different environmental stimuli. In this case, drought. Therefore in the ZB 2012 dataset each genotype has 3-4 well-watered repetitions and 2-3 repetitions, which grew under water deficit. Time series are represented as Biomass or Leaf-Area per Thermal Time as described in section 3.2.4. An example of the relevant data for one plant is shown by Figure 3.5.

3.4 Previous Work

There was a previous different attempt to analyse the data and group it automatically. In order to overcome the problem of unequally sampled data, the time series where interpolated. From a predefined set of functions each series was mapped to a function which represents best its behaviour. The clustering was then performed not on the time series but on their function representation.

This approach was abandoned as the distance between two functions did not sufficiently represent the real difference of the two underlying time series. Therefore this work avoids the use of further abstraction layers and focuses on methods operating on the raw data.

4 Outlier Detection

This chapter serves as introduction to the domain of outlier detection. We discuss the different notions of outlier and the corresponding detection methods. Further we present the task at hand of detecting outlier in plant time series as well as our approach and the final evaluation.

The outlier detection task has many names such as anomaly detection, novelty detection, noise detection, deviation discovery and exception mining, despite the diverse naming they share the same basic approaches to tackle the common problem. While highlighting techniques used by all those categories, we will refer to this task as "*outlier detection*". But before we can detect outliers, we need to know what outliers are. We will follow the suggestion of Hodge and Austin [2004] to use the definition of Grubbs [1974], who introduced statistical rules for the detection of outliers.

Definition Outlier: *"An outlying observation, or "outlier", is one that appears to deviate markedly from other members of the sample in which it occurs. "...*

a *"An outlying observation may be merely an extreme manifestation of the random variability inherent in the data. "*

b *"On the other hand, an outlying observation may be the result of gross deviation from prescribed experimental procedure or an error in calculating or recording the numerical value. "[Grubbs, 1974]*

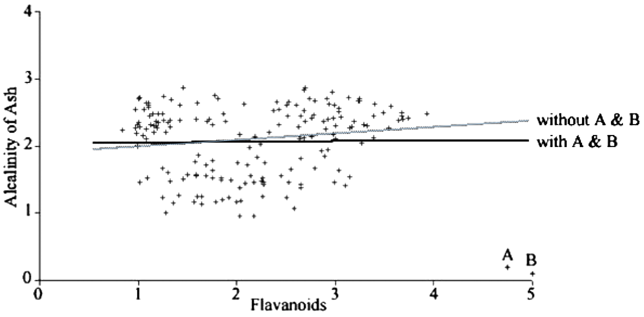
[John, 1995] introduce another notion to the outlier nature. They suggest that a data point located in a "surprising" location can be considered as outlier. Surprising means here for example the detection of a point of class A in between of a cloud of class B instances. Another notion considered by [Aggarwal and Yu, 2001] is the distinction of noise and outliers, where outliers behave differently from the norm and lie outside the noise region.

The detection of outliers can be of high importance in many contexts. Especially in safety critical environments, outliers are anomalous events which might cause significant performance degradation or lead to an accident. As for example, an engine rotation defect in an aircraft or a flow problem in a pipeline [Hodge and Austin, 2004]. The domains and purposes of outlier detection are versatile:

- Network performance monitoring is a vast field where outliers can indicate bottlenecks or detect abnormal behaviour helping to identify network intruders or hostile traffic, see Meratnia and Havinga [2010]
- Fault diagnosis aims to detect faulty products on manufacturing lines or can be used in critical environments such as the analysis of behavioural patterns in motors, pipelines or air space instruments.
- Image analysis offers many applications for outliers, in satellite images they can represent novel features, in surveillance systems they indicate critical events and in a series of images they may be used to distinguish moving objects from their background
- In the medical domain, outlier detection can be used to monitor heart beats and detect unusual events. Or to analyse cells and detect abnormal features in order to identify malicious behaviour.
- Further detection of novelty in text can be used to filter spam mails or undesirable topics.

And this is by far no exhaustive listing, note we haven't even mentioned the economic sector. Some of the common sources of outliers are: human error, error of sensors and instruments, faults in systems, abrupt change in behaviour or just natural deviation. The handling of outliers completely depends on the context. While faulty measurements can be pruned automatically, suspicious network events should raise an alarm and cells with unusual behaviour should be retained for further analysis.

Figure 4.1: Influence of outliers on the placement of a regression line on the wine dataset by Hodge and Austin [2004]. The black line shows a regression line fitted to the data without outlier removal and the grey line represents the same procedure after removal of the outlier points A and B.



4.1 Outliers in phenotyping

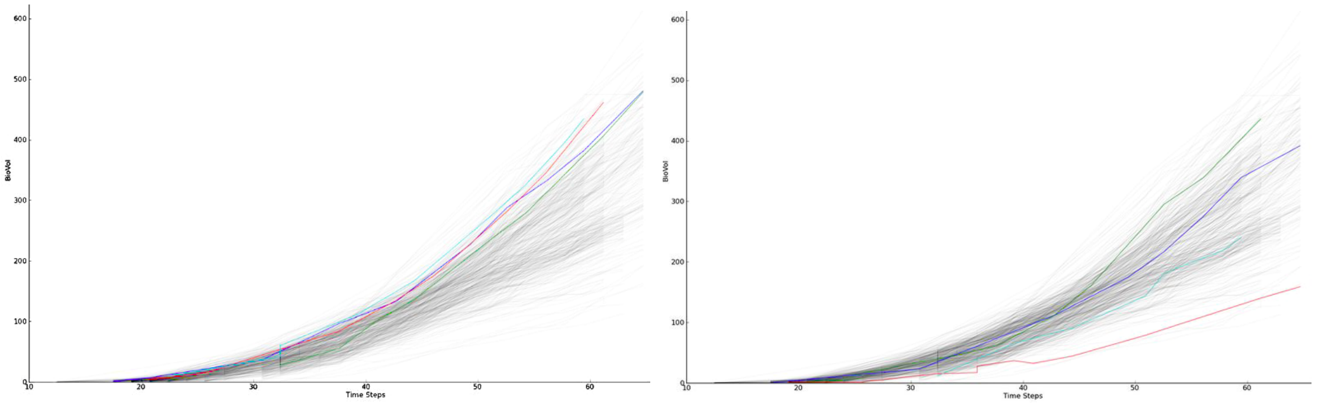
The context of this thesis is the domain of agronomy, more precisely the domain of phenotyping. Thus the considered data is issued from the PhenoArch phenotyping platform described in Section 3.2. The data obtained are measurements of plant traits during their development. Despite the attempt to control as many variables as possible in the experiment, such as temperature, daily amount of water, air humidity and so forth (in order to be able to compare the measured characteristics of different genotypes) this does not prevent the occurrence of outliers. As many sensors are involved into the measurement procedure, a great part of outliers can be introduced by sensor errors, faults in the watering or imaging system, or during post processing. Measures like plant biomass are estimated upon three images of a plant. Wrong lighting conditions or a camera dropout will cause a faulty biomass estimation. While there exist more robust approaches, methods such as regression are highly influenced by outliers as shown by Figure 4.1. In this dataset the presence of only two outlier points causes a remarkable shift in the placement of the regression line.

However, not only outliers caused by sensor errors might influence the analysis, but wrong labelled plants showing completely different characteristics in comparison to their repetitions or broken leafs resulting in a sudden decrease of biomass should get our attention. This leads to a distinction of the following three outlier categories:

- *Point outliers* - outliers within a time series, aberrant points often caused by measurement faults.
- *Plant outliers* - an entire time series is considered as outlier if it differs significantly from the behaviour from further repetitions of the same genotype.
- *Genotype outliers* - an entire genotype is considered as outlier if no common traits can be detected within its repetitions.

The dataset at hand has been already cleaned from the most outlier points using a model-based approach. Thus the main focus lies on the detection of plant-outliers . The outlier definition of Grubbs does fit really well to this case. The sample consists of the repetitions of one single genotype. Thus we deal with four to six time series. A plant outlier is a series differing from the most sample members, often caused by wrong labelling of the pots, an error during planting, accidental damage of parts of the plant or just natural variation. Figure 4.2a shows an example of a normal-behaving genotype while Figure 4.2b demonstrates a genotype containing an outlier, which is highlighted in red. Note that experts are more lenient with plants differing from the other sample members by their exceptionally fast biomass accumulation than with bad performing outliers [F. Tardieu, 2013].

The third category, a genotype outlier, can be considered as a special case of the plant outlier. A genotype is considered as outlier when none of the sample members show similar growing patterns. As the plant outlier, this kind of outlier is due to planting or labelling errors, natural variation or an unfortunate combination of all three. The handling of outliers in the given context is their removal from further analysis. We have seen on the example of regression (Fig. 4.1) that two point outliers can cause an impressive shift in the estimation. Similar to this effect once we analyse growth performance



(a)

(b)

Figure 4.2: The left figure (a) shows the genotype *11430H* of the ZB 2012 dataset, all of its repetitions show similar growing pattern. On the other side figure (b) shows the genotype Lo1180H containing at least one outlier plant, highlighted in red. The greyed out sequences in the background illustrate the distribution of a large part of the remaining dataset.

on genotype level, plant outliers will bias the comparison. Therefore it is necessary to remove such sequences.

4.2 Outlier detection methods

Considering the outlier definition of Grubbs in Section 5, where an outlier is an observation deviating from the samples it appears with, the task seems to be simple. We need to define a range of normal observations and each data point not falling into this range is labelled as outlier. But this approach includes some tiny details which can turn the apparently obvious proceeding into a challenge.

– It is difficult to define a precise range so that it includes all possible normal observations. Further the boundary between outliers and normal behaviour is often fuzzy [Hodge and Austin, 2004], thus an observation close to the boundary but outside the range could be actually a normal observation.
– In many domains, the notion of normal behaviour can change over time, for example in the year 2000 gas prices in France had an upper limit of 1EUR/litres everything above was an exception, and since the year 2010 the normal price is 1.50EUR or above [France-Inflation.com], thus a defined range or model of normal behaviour for the year 2000 will be suboptimal for records since 2010.
– Further the definition of an outlier is highly domain dependent. While peaks in a seismogram represent, outliers and stand for high seismic activity and probably for an earthquake, the same peaks in voice records are completely normal.

Thus it is difficult to find a general outlier detection method fitting to every context. In addition, outliers can have different natures, as we have seen in the previous section for the case of phenotyping, we consider point outliers, plant outliers or genotype outliers. Hodge and Austin [2004], Chandola [2007], Chandola et al. [2009] distinguish the following types based on the nature of outliers:

Point Outlier This category of outliers is similar to our definition for the context of phenotyping. It is an aberrant point in comparison to the rest of the data. The majority of outlier and anomaly detection research focuses on this basic kind of outliers Chandola et al. [2009]

Contextual Outlier A contextual outlier is similar to a point outlier as it is a single aberrant data point, but the difference is that it is considered within its proper context. Regarding the entire dataset this data point might appear in the *normal* range but restraining the data to its context will reveal the deviation. This outlier category has often been investigated in the context of time-series [Salvador et al., 2004, Kou et al., 2006]. Indeed, we can find a high similarity to the here defined class of *plant outliers*. If we put aside the multidimensionality of a time series and consider it as a feature vector, we can treat it as single data point. Restraining the context to the repetitions of a genotype the given time series appeared in, we obtain the problem of *contextual outliers*.

Collective outliers Collective outliers are collections of related data points which are anomalous with respect to the rest of the data. The individual data points might not be anomalous by their own, but their collective appearance is exceptional. This includes cases like surprising subsequences of a long sequence or a sequence of abnormal structure in a set of sequences, like an aberrant genotype repetition.

Note that while point outlier can appear in any dataset, collective outlier require a relation between data points and contextual outlier need a specific context. Chandola et al. [2009] underline that any point or collective outlier can be regarded as contextual outlier by including the given context in the analysis. An important requirement for the successful retrieval of outliers is the knowledge of what an outlier actually looks like in out dataset, preferably in the form of annotated outlier and non-outlier instances. It is of highly importance for the evaluation of a method or can be useful for the creation of a model. Unfortunately this kind of data is very expensive as it has to be human made. Based on the amount of labelled data instances, approaches to outlier detection can operate in the following modes:

Supervised outlier detection Supervised approaches require pre-labelled data for the normal behaviour, and the outlier instances. This data is used to generate a model for each label. If the annotations only have labels for outliers and normal data points, the obtained classifier will only differentiate between these two classes. Each new observation is compared against the models in order to decide where it belongs to. The labelled data has to provide sufficient samples for both classes, the normal but also the outlier class, and cover as many notions of the data as possible in order to obtain a truthful model. Otherwise new observations with previously unseen characteristics might get an incorrect class assignment [Hodge and Austin, 2004]. Chawla et al. [2004] discuss the issues resulting from unbalanced classes.

Semi-supervised outlier detection Semi-supervised approaches model only one class, often the normal behaviour and flag all new observations not corresponding to the obtained model as outlier. This results from the fact that outlier data is often hard to obtain, for example in the fault detection domain, it would require to introduce a system fault to obtain the outliers we are interested in. Whereas in the case of spam detection there are approaches creating a spam model [Mishne et al., 2005], as normal mails can be by far more versatile and therefore hard to represent. However, even if we need labelled data for only one class, we require the full distribution of the class to be modelled in order to permit generalisation [Chandola et al., 2009]. The advantage of this approach is that even a new unseen kind of outlier, as for example a new network intrusion technique, is still handled as outlier as long as it does not completely resemble the normal behaviour (assuming we modelled the normal behaviour). This implies also that a shift in normal behaviour requires the re-learning or shifting of the model [Hodge and Austin, 2004].

Unsupervised outlier detection Unsupervised techniques are widely applicable as they do not require any pre-labelled data. The main approach is to process the data as a static distribution and to flag the most deviant points as outliers, which is similar to unsupervised clustering [Chandola, 2007]. Thus they implicitly make the important assumption that normal data instances are more frequent in the dataset than outliers.

While the aforementioned categories represent modes an outlier detection algorithm can operate in, the subsequent sections will introduce the fundamental techniques.

4.2.1 Classification-based methods

Classification approaches operate in a two-phase mode, *training* and *application*. For training, those techniques require data which is already labelled with its corresponding classes in order to learn a model.

In the subsequent application phase new unlabelled data instances can be labelled based on the learnt class characteristics. The classification-based outlier detection makes the assumption that the provided features are sufficient to distinguish between normal and outlier instances.

Chandola et al. [2009] groups classification-based approaches into multi-class and one-class outlier detection techniques. Multi-class techniques learn to discern multiple *normal* data classes and a new data instance is classified as outlier if it does not fit to any of the available classes. One-class-based approaches learn a discriminating boundary between normal instances and outliers.

There are different classification algorithms which can be used for this task. For example Ryan et al. [1998] used a neural network to detect unusual activity in a computer system. Neural networks can be applied in the multi-class and one-class setting. Further they perform well on unseen data and can learn complex boundaries [Hodge and Austin, 2004].

Another class of classification algorithms are Bayesian networks. The basic idea of the naive Bayes classifier is to use the training data in order to estimate the prior class probabilities and the likelihood of an observation given a class. For the label prediction of a test instance, it estimates the posterior probability of seeing this instance and observing a class label from the set of labels. The label with the highest posterior becomes the predicted instance label. Several variants of this approach have been used for network intrusion detection [Kruegel and Mutz, 2003, García-Teodoro et al., 2009] and Das and Schneider [2007] used a more complex Bayesian network in order to include the conditional dependency between attributes, which is ignored in the classic version.

Classification approaches are a powerful tool, especially when used in the multi-class setting. Once we have obtained the learned model, the classification of new data is considerably fast [Chandola et al., 2009]. But this comes at a relatively high cost. In order to create a truthful data model we need labelled training data for each class, which in most cases has to be labelled manually and therefore is often not available.

4.2.2 Nearest-neighbour-based methods

Nearest-neighbour-based outlier detection approaches make the assumption that *normal* data instances appear in dense groups while outliers are further away from the others [Chandola et al., 2009]. Those approaches require a distance or similarity measure, one of the popular choices is the euclidean distance, see Section 2.3 for further measures. Nearest-neighbour-based techniques can be grouped into two categories:

1. Techniques defining the outlier score by the k^{th} nearest neighbour.
2. Techniques using the relative density of each data instance as outlier score.

The first category is the k-nearest-neighbour classifier (kNN) which classifies points to the class that appears most often among the k-nearest neighbours. Thus the euclidean distance of a point p to its k^{th} neighbour $d_k(p)$ is the outlier degree of p. In order to get the top-n outliers this approach chooses the n greatest $d_k(p)$ scores. While this approach is robust to noise, the parameter k is often difficult to choose in practice [Patcha and Park, 2007]. Liao and Vemuri [2002], Hautamäki et al. [2004] used kNN for outlier detection. Various modifications and improvements have been proposed for the basic technique of kNN, mostly targeting the definition of the anomaly score, the similarity measure or the efficiency [Chandola et al., 2009]. One such modification is to sum over the k-nearest neighbours [Eskin et al., 2002, Zhang and Wang, 2006]. A possible modification of the notion of the outlier score is to count the neighbours n which are within a neighbour boundary defined by the distance d.

An attempt to speed up kNN outlier detection, which has a computationally complexity of $O(N^2)$ in the classical case, was undertaken by Bay and Schwabacher [2003] who noticed that a great part of the time is spend by processing the non-outliers. Thus they use a pruning approach. When calculating the nearest neighbours for a data point p the anomaly threshold is set for any data point to the weakest outlier found so far. They show that it leads to a nearly linear-time complexity in the average case on randomly

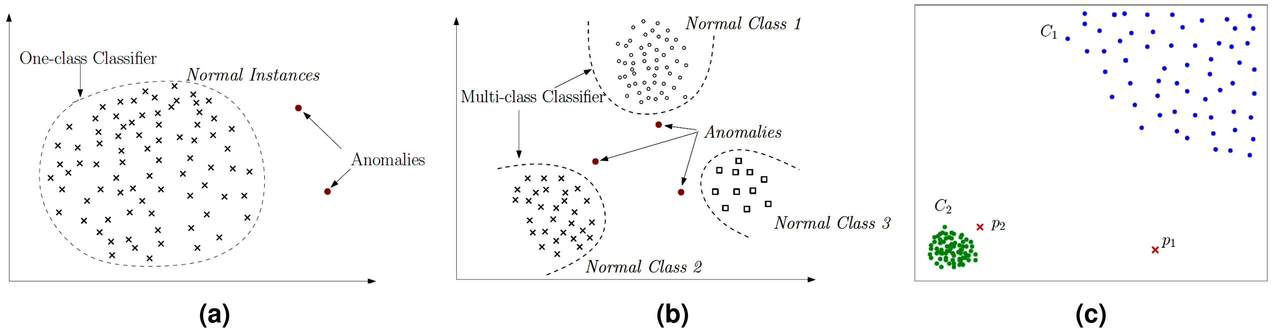


Figure 4.3: Figure a illustrates an one-class classifier which creates a boundary for **normal** behaviour. Image b depicts an multi-class classifier, which not only detects outliers (data points in red) but distinguishes between *normal* classes. And figure c is a representation of a dataset containing two groups, depicted in red and blue, of different densities. This example underlines that a classic nearest-neighbour outlier detection approach will not be able to detect the outlier p_2, due to the high distance between the blue points. These illustrations have been adopted from [Chandola et al., 2009]

ordered datasets.

The second category of nearest-neighbour algorithms uses a neighbourhood density estimation to identify outliers. Instances in a neighbourhood with high density are assumed to be normal, while data instances in a low density region are flagged as outliers. The previously described basic k^{th} nearest-neighbour approach can also be regarded as a density-based outlier detection approach when we take into account the following points. The k^{th}-nearest-neighbour distance of point p, $d_k(p)$ corresponds to a hypersphere centred at p and containing k further data points. Thus, $d_k(p)$ can be seen as the estimated inverse density of p in its dataset. Unfortunately, density-based techniques perform bad on datasets with variable density as in Figure 4.3c. In this example, the distance of p_2 to its nearest neighbour from the green cluster is lower than the nearest-neighbour distance of points within the blue cloud. Thus, p_2 won't be considered as an outlier.

In order to overcome the difficulties arising with variable density datasets, Breunig et al. [2000] assigned a local outlier score to any data point, known as *Local Outlier Factor (LOF)*. LOF represents the extent to which a point p will be considered as outlier. LOF is the average local density of the k nearest neighbours of p and its local density itself Chandola et al. [2009]. To compute the local density of p, we need to find the radius of the smallest hypersphere, centred at p and containing its k nearest neighbours. The local density is than k divided by the volume of the hypersphere. It follows that if p is one of the green points in Figure 4.3c, it will have the same local density as its neighbours, but if we consider p_2, its local density is higher than that of its neighbours, it results in a higher LOF.
Several variants have been proposed to improve certain characteristics. Tang et al. [2002] propose a variation, called *Connectivity-based Outlier Factor (COF)*, which computes the neighbourhood of a node in an incremental manner. It is designed to better caption certain regions, such as straight lines. Further work has been done to handle different data types such as streams [Pokrajac, 2007] and to reduce computational complexity [Chiu et al., 2003] which is $O(N^2)$ for the classical case.

The main advantage of nearest-neighbour outlier detection approaches is that they are unsupervised and purely data driven. They can also be adapted to different data types just by picking the appropriate distance measure. However they rely on the fact that normal instances lie in dense regions. If this is not always the case, this will result in false outliers. And a very important point to recall is that this technique relies heavily on the truthful representation of distance between data instances by the chosen distance measure.

4.2.3 Clustering-based methods

Clustering aims at grouping similar data instances into clusters. As for the case of nearest-neighbour approaches, the similarity of objects is defined by a context and data dependent distance measure. Outlier detection via clustering is mainly a unsupervised approach, but can be also performed in a semi-supervised manner, for example by using labelled data to obtain an outlier threshold [Chandola et al., 2009]. While we will introduce here the main ideas used for clustering-based outlier detection we won't go into detail of general clustering. A more detailed discussion on this topic and the presentation of basic clustering techniques is covered by Section 5. There are basically three assumptions a clustering-based outlier detection approach may make use of:

1. Normal data instances appear in groups forming clusters while outliers do not belong to any group
2. Normal data instances are close to a cluster centroid and the closest cluster centroid of an outlier is *far* away.
3. Normal data instances are part of a large and dense cluster while outliers are in clusters which are small and/or sparse.

The first category of approaches applies a well known clustering algorithm and declares all data instances as outliers that were not assigned to a cluster. This requires an algorithm which does not necessarily assign all data instances to a cluster, such as DBSCAN [Ester et al., 1996] or SNN [Ertoz et al., 2002], a nearest-neighbour clustering approach. A concern of these approaches is that they are optimised to find meaningful clusters and not to pick out outliers.
The second category calculates an outlier degree based on the distance to the nearest cluster of a data instance p. For this purpose k-means and in particular Self-Organizing Maps (SOM) [Kohonen, 1997] are widely applied, see Hodge and Austin [2004], Chandola [2007]. These techniques assume that outliers do not form clusters by themselves, and do not work for cases where outliers appear in small groups. Approaches of the third category overcome this issue by defining that normal instances have to come in dense and large groups. He et al. [2003] incorporate this notion and introduce the *Cluster-Based Outlier Factor*, which is assigned to each data instance in order to represent the outlier degree. This factor captures the size of the cluster an instance p belongs to and its distance to the centroid.
Several cluster-based approaches appear to be similar to the nearest-neighbour techniques discussed in the previous subsection. Both approaches rely heavily on the performance of a distance measure between data instance and incorporate the notion of density. However the main difference of these approaches is that while nearest-neighbour outlier detection techniques determine the outlier degree of instance p based on the k-nearest neighbours, clustering approaches operate on the cluster the instance p is assigned to.

The benefit of the clustering approach to outlier detection is that these technique can operate in the unsupervised mode and the outlier detection phase is fast, since new data points have to be compared against a relatively small number of clusters. Nevertheless, the initial clustering of the data appears often to be the bottleneck [Chandola et al., 2009]. Further the detection performance depends on whether the clustering algorithm in use is adapted to capture the notion of 'normality' in the data.

4.2.4 Complex outliers

At the beginning of Section 4.2, we introduced that outlier detection can be divided into groups to handle best the three different kinds of outliers: *Point outlier, contextual outlier* and *collective outlier*. The methods presented above mainly focus on the simple case of point outliers but there also methods to handle complex outliers such as contextual or collective outliers.
Contextual outliers are data instances which are only outliers within their context, they might be considered as normal when regarding the whole data. The context of such data can be of a completely different

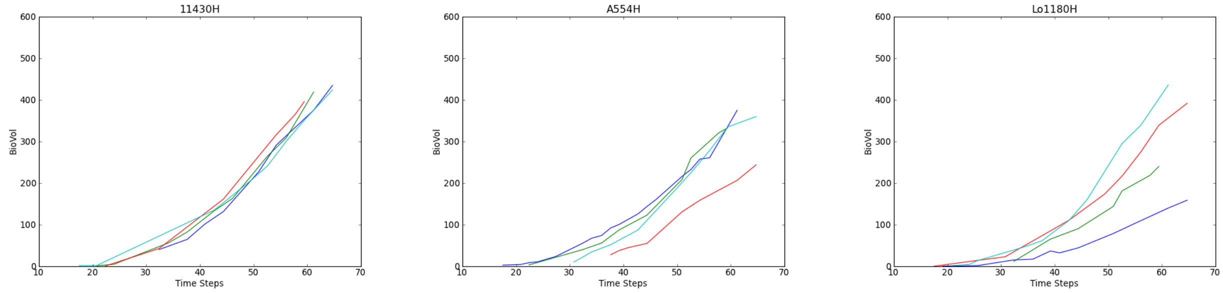
nature, such as spatial, where the location of an instance is of high relevance, sequential, where the context is a sequence of points or it might be of a completely different nature such as a category derived from the attributes of a data instance. There are mainly two approaches to handle those outliers, either by reducing them to point outliers or by making use of the given data structure. As these outliers are only anomalous in their context, the reduction to point outliers is achieved by applying the aforementioned techniques only within a given context. The second option is to model the data structure and to use the model for outlier detection. This is used in cases where it is difficult to define the actual context. The model is learned on training data and used to predict the normal behaviour. If the encountered behaviour differs significantly from the prediction, it is considered as outlier. A simple example for this technique is regression where contextual data is used to fit a regression line [Chandola, 2007, Chandola et al., 2009]

Another kind of complex outliers are collective outliers, which are not exceptional by themselves, it is their co-occurrence that makes them exceptional. These outliers can be grouped by the type of their relationships: sequential, spatial, graph-based. Given the context of this work, we are mostly interested in sequential outliers. Further information on spatial and graph-based collective outliers, and their detection methods are provided by Chandola et al. [2009], Chandola [2007].
Dealing with sequential collective outliers one application is to detect outlier sequences from a set of sequences. This implies two challenges. The sequences are not necessarily of equal length and they are not necessarily properly aligned. This means that the first point in sequence T might correspond to the third data instance in sequence S. This is a major challenge when dealing with biological sequences [Gusfield, 1997]. As for the case of contextual outliers one of the classical approaches is to reduce the problem to a point outlier detection task. Provided that the series are of equal length, we can transform the sequence into finite feature space. This means that a time series T of length N can be regarded as a single data instance with N attributes. Then the usual outlier detection techniques can be applied. Blender et al. [1997] used this approach to detect cyclone regimes in North Atlantic weather data. In order to deal with the variable length of sequences and the misalignment the easiest way is to use an appropriate distance measure capable of serving these conditions [Chandola et al., 2009]. For a discussion on distance measures see Section 2.3. There are also model-based approaches to this task using Finite State Automatons [Sekar et al., 2002], Marcov Chains [Ye and Li, 2000] and Hidden Markov Models (HMM) [Warrender et al.].

A variation of the here described task can be the detection of outlier subsequences within a long sequence, such as the detection of anomalous patterns in an EEG. This version entails a different challenge, the length of outlier subsequences is not defined and can vary within a single sequence. This makes it difficult to create a model for normal behaviour [Chandola et al., 2009]. Chakrabarti et al. [1998] regard this problem from the information theoretic perspective. They create subsequences of the sequence which minimises entropy. The sequences with highest entropy are considered as outliers. Further Keogh et al. [2004] use a sliding window to detect outliers. The outlier score is obtained by comparing each resulting subsequence against the original sequence.

4.3 Outlier detection for alarm generation

In subsection 4.1 we introduced three outlier notions appearing in the context of phenotyping: *point*, *plant* and *genotype outlier*. The literature differentiates between *point outliers*, like in our case, and complex outliers as *contextual* and *collective* outliers. We are mostly interested in the detection of plant outliers. Regarding the categorisation proposed in the literature, we can assign them definitively to the class of complex outliers as plant outliers are more than just an anomalous data point. Plant outliers are entire time series, which differ significantly from their genotype equivalents, making it a collective outlier. Further we are not interested in outliers of the entire dataset but rather on aberrant series of

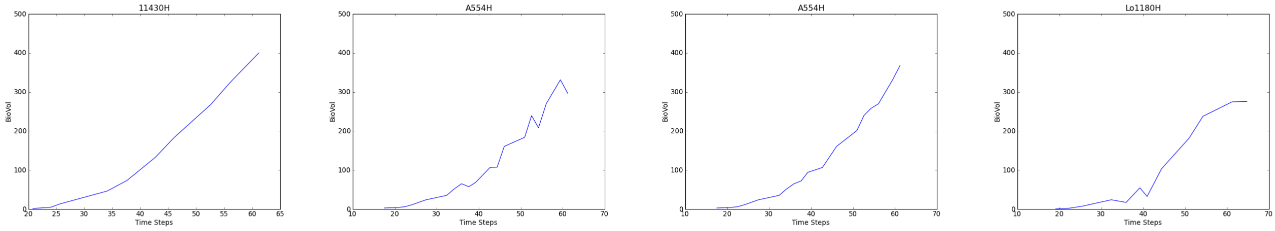


(a)

(b)

(c)

Figure 4.4: These figures show the performance of biomass accumulation for three genotypes, $11430H, A554H, Lo1180H$. The genotypes are represented by their repetitions which grew in the well-watered condition. $11430H$, in figure (a) serves as an example for a good performing genotype where all repetitions follow the same course. $A554H$ in (b), follows about the same course but contains one outlier, highlighted in red. Figure (c) represents the genotype $Lo1180H$ showing very variable performances.



(a)

(b)

(c)

(d)

Figure 4.5: This figure represents the averaged performance of the genotypes depicted by 4.4. (a) is the averaged performance of $11430H$, (d) is the averaged performance of $Lo1180H$ and the images (b) and (c) represent $A554H$, whereas (b) is the average of all its repetitions, including the outlier and (c) was obtained after the outlier removal. We can see that only after outlier removal we get a more appropriate performance representation.

a genotype repetition. This means that they have to be retrieved in the context of their genotype and therefore can be considered as contextual outliers. To put it briefly, the goal is to detect *contextual collective outliers*.

The importance of their retrieval can be illustrated by their effect on further analysis of the data. As already shown in Figure 4.1 the presence of outliers in a dataset can have a remarkable effect on analysis such as regression. In the given context further analysis aim to study plant characteristics based on the genotype variety. In order to ease this work genotypes will be grouped into categories of similar growing patterns. A genotype is represented by its four to six repetitions. If we do not remove any aberrant repetitions they will influence the estimated average performance of the genotype which might lead to a wrong group assignment. Figure 4.4 shows three genotypes represented by their repetitions. Figure 4.4a is a well-performing genotype with its repetitions following the main course, Figure 4.4b is similar to (a) but contains an outlier and Figure 4.4c shows a large variability in its performance. In Figure 4.5 we illustrate the average performance of the genotypes in Figure 4.4(a), (b) and (c). We can see that if we do not remove the aberrant series as depicted by Figure 4.5b, its performance appears to be more similar to the variable genotype than to the more appropriate good performing genotype. This illustrates the necessity to remove plant outliers.

More specific requirements for an outlier detection solution are defined by two experts, who work with the PhenoArch platform and are familiar with the data. The outlier detection should be able to be integrated into the current workflow in an ’expert-agreement’ mode. The outlier detection component will propose genotypes containing possible plant outliers which will be reviewed and if appropriate removed by the expert. Therefore we have to focus on the recall, see Equation 4.2. This means that the algorithm has to retrieve preferably all outliers available in the dataset (high recall), and in order to reduce the amount of instances, to be checked by the expert, it should avoid false matches (keep precision high). Further it has been noted that low performing outliers are more crucial and we can be more lenient with well-performing anomalies. Keeping these aspects in mind, the following subsections discuss the similarity distance which has been used and three approaches. The performance of these approaches is then presented in Section 4.4.

4.3.1 Similarity distance

We refer to the statement that most data mining tasks require a notion of similarity, this is especially the case for the task of outlier detection. In order to declare a data instance p as outlier, it is necessary to define by which means p is less similar to the rest of the considered data. In Section 2.3 we have reviewed different distance measures and seen that the classic approach is to use the Euclidean distance which is fast, easy to implement and meets most requirements. Further we want to recall the observation of Ding et al. [2008], stating that the performance of simple methods approaches the performance of more sophisticated elastic measures in big datasets. Therefore we will reconsider the available amount of data.

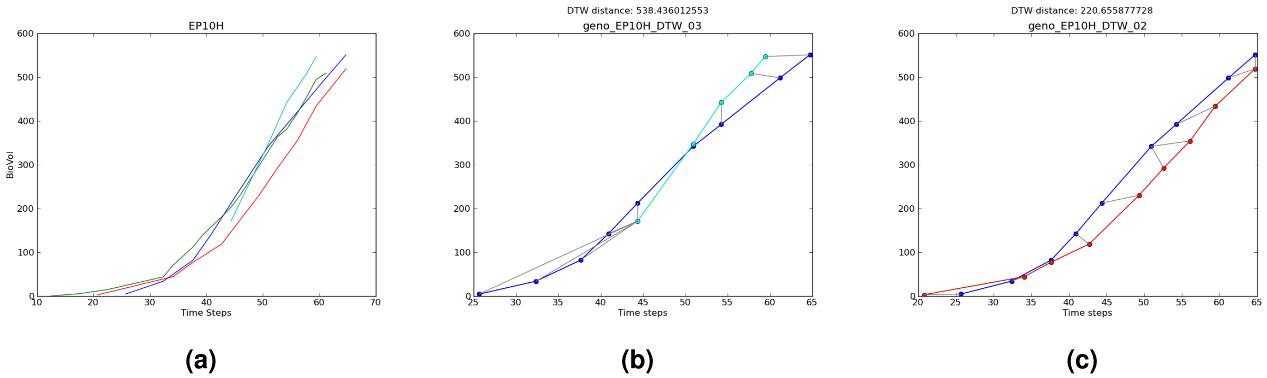
The available data consists of several phenotyping experiments, which will be analysed separately as the experiments where performed under different conditions. An experiment can contain about 1690 plants resulting in about 300 genotypes if we count four to six repetitions. As we are mostly interested in the contextual plant outliers, we will regard them in their context which is the genotype. Thus we are targeting the detection of outliers in a very small set of four to six time series. This small amount of time series induces the use of a more sophisticated measure as Dynamic Time Warping (DTW). We use DTW with the Euclidean distance as local distance measure. DTW has shown to be more accurate and more robust in many applications in comparison to the Euclidean distance, see Section 2.3.

As stated by Gusfield [1997], time series in the domain of biology often have the inherent properties of non-aligned sequences and differences in sequence length. The data at hand is no exception. Even though DTW is able to deal with sequences of different lengths, we observe that shorter sequences get a higher dissimilarity estimation as equal length sequences, see Figure 4.6. This might be a valid behaviour for the general case, but considering the given context this is not desirable. A time series starting at later point in time does not necessarily mean that the measured plant started growing at this time point but that the conditions were favourable for its detection by the camera system. This might be due to the plant architecture like a small size, very thin structure, difficult growing angle for detection or just a camera fault. We can also find earlier ending time series in the dataset, which could be caused by an accident requiring earlier harvesting or a difficulty in the imaging system. Therefore a low similarity score should not only be based on the length difference of two series.

To overcome these problems, one option is to use interpolation to obtain series of equal length, this would avoid the kind of alignment shown by Figure 4.6b. We have chosen to work only on the available data and therefore, when calculating the DTW distance for a long series T and a short series S, we remove the elements of T which are missing at the beginning or end of S.

4.3.2 Lower bound approach

The first approach bases on the fact that bad performing outliers are considered more crucial than good performing ones and on the observation that those outliers have a high distance from the rest of



(a) **(b)** **(c)**

Figure 4.6: Figure (a) illustrates the repetitions of the $EP10H$ genotype. We will have a closer look on the cyan, blue and red graph repetition and their distance estimation using DTW. Figure (b) shows the DTW alignment between the short cyan coloured and blue coloured series. As DTW requires every point to be aligned, this results in the alignment of the first cyan point with every preceding point of the blue time series, resulting in a high distance measure of 538,43. Whereas the DTW distance of a same length series of subjectively equal similarity, illustrated by (c), is much lower (220,65)

the repetition. Thus, given a genotype G and a distance threshold d, we retrieve its worst performing repetition. If its closest neighbour has a higher distance than d, the repetition is flagged as outlier. In order to cover cases with two bad performing outliers, we apply the same procedure to the second worst performing repetition and flag both series as outliers if the neighbour of the second series is further away than d. This is illustrated by Algorithm 1. Note that this approach does completely ignore good performing outliers.

```
Algorithm detect_lower_bound_outlier(repetitions, dist)
    sorted_reps = repetitions.sort(order='avg');
    outlier_list = [];
    if dtw_distance(sorted_reps[0], sorted_reps[1])>dist then
        outlier_list.append(sorted_reps[0])
    else, if sorted_reps[2] != null & dtw_distance(sorted_reps[1], sorted_reps[2]) then
        outlier_list.append(sorted_reps[0], sorted_reps[1])
    return outlier_list;
```

Algorithm 1: Lower bound outlier detection

4.3.3 Neighbourhood approach

This approach belongs to the nearest-neighbour approach family as described in Subsection 4.2.2. The idea is not to directly detect outlying time series but rather to find 'normal' similar sequences and define the rest as outliers. This assumes that normal time series appear in groups and outliers are isolated sequences. Thus we consider every given time series T as 'normal' if it has at least k neighbours at a predefined distance d. This approach is illustrated by Algorithm 2. It is a costly procedure as we need to determine the nearest neighbours for each time series.

4.3.4 Combined approach

The combination of several approaches is often referred to as "Ensemble Method", Dietterich [2000] discusses ensemble methods in the domain of machine learning and claims that ensemble classifier yield

```
Algorithm detect_outlying_neighbours(repetitions, dist, k)
    outlier_list = [];
    for rep in repetitions do
        for neighbour in get_nearest_neighbours(k, rep) do
            if dtw_distance(rep, neighbour)>dist then
                outlier_list.append(rep)
        end
        return outlier_list;
    end
```

Algorithm 2: Neighbourhood-based outlier detection

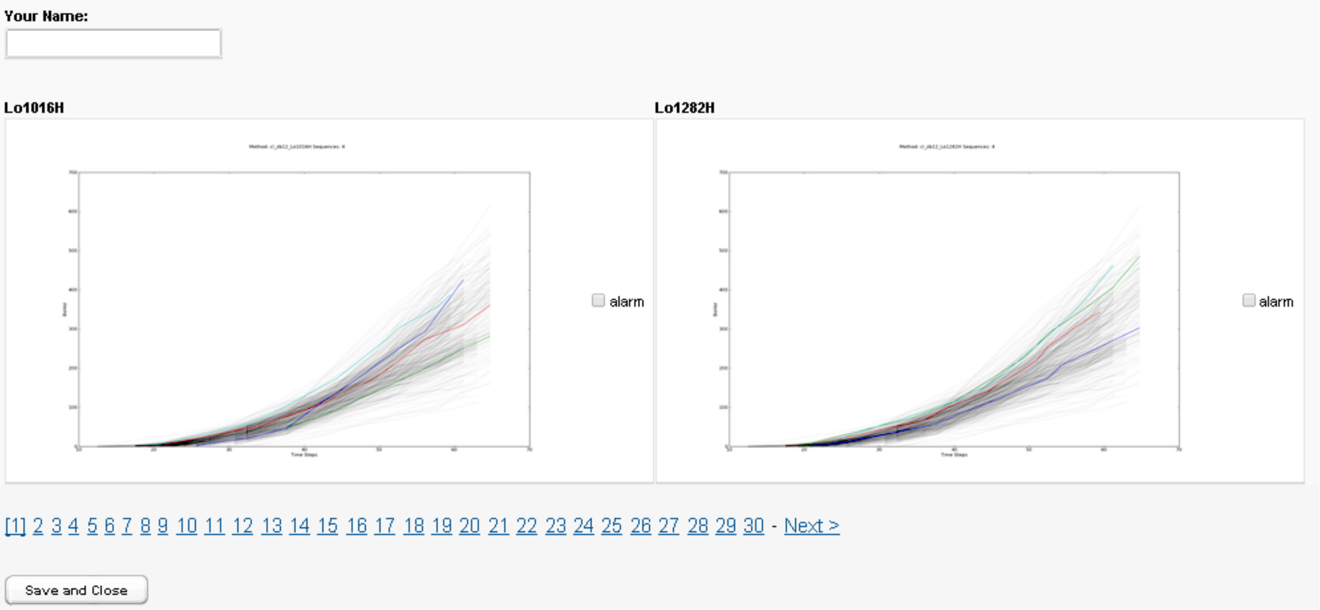

Your Name:
Lo1016H
□ alarm
Lo1282H
□ alarm
[1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 - Next >
Save and Close

Figure 4.7: User interface for the annotation of aberrant repetitions.

better performance than a single classifier. Depending on the number of available approaches, their combination can be organised in different ways. If there are more than three techniques available, they can be arranged as a majority voting. Otherwise, in case of an outlier detection application, one can either use only the results where all participants agree or set a flag where at least one approach detects an anomaly. In order to be sure to get the low performance outliers and not to miss extreme high performing anomalies, we combined the two approaches defined in Algorithm 1 and 2. Considering that we have only two approaches, the majority voting does not make any sense. Thus we realised the two latter combinations. The first one was to mark those genotypes, where both algorithms detect an outlier. The other version marks the genotypes where one or both of the algorithms raised an alarm. In order to get the best performance, the combined approach was executed with various distance parameters for both underlying algorithms. The final performance is reported in Section 4.4.2.

4.4 Evaluation and results

Section 4.2 shows that many outlier detection methods require pre-labelled data for training and testing. Even though we do not require training data we need to fix the free distance parameters and labelled data is indispensable for evaluation. Therefore we started a venture to annotate outlying plant time series, which is discussed in the following section. The resulting data was used to measure and compare the performance of the here proposed approaches, see Section 4.4.2.

The aforementioned approaches both require the definition of a distance parameter that has to be adapted to the kind of data at hand. This is best done with annotated data, where we can evaluate which settings generate too many false dismissals or false matches. Further and most importantly only labelled data can provide an objective measure on the performance of our approaches and their suitability for this task. At this point we have to admit that 'objective' is a big word and we will mention certain concerns on the objectivity of this measure, considering the given situation, in the course of this section.

Often, labelled data is not available and has to be obtained via a manual data annotation by domain experts, making it very expensive. This also corresponds to our situation. In order to facilitate and speed up this procedure, we developed a web interface representing the genotypes which had to be marked in case of an outlier. Figure 4.7 shows a screen shot of this interface. The genotype representation has been realised as in Figure 4.2b. The experts had to choose between three different representations, shown in Figure 4.8. Next to each figure is a checkbox which has to be checked in order to report an outlier.

Two domain experts familiar with the data at hand annotated all genotypes of the $ZB2012$ phenotyping experiment, resulting in 300 annotated images. The instructions were to mark images containing aberrant time series which should be removed from further analysis. This seems like an obvious task but it turns out that it is not as straightforward as expected. In order to get more insight into the difficulties of the outlier detection task and to obtain an upper bound we can reach with automatic methods we computed the annotator agreement on the annotated series using recall, precision and the F_1 score. The F_1 score is the harmonic mean between recall and precision defined as, *e.g.*, [D, 2000],

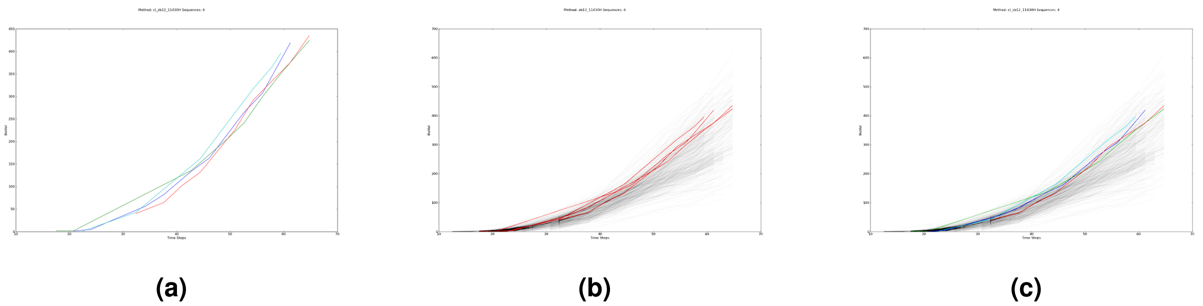
$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}. \tag{4.1}$$

There are two possibilities how we can look at precision and recall. In the classical case, precision is the fraction of correctly classified data instances, whereas recall measures the percentage of outliers present in the data which were actually retrieved by our system. Another version is to divide the data into two classes, a positive and a negative one, and to constrain the precision to the positive class. This means that precision will measure the percentage of the correctly classified positive instances [D, 2000], see Equation 4.2. This can also be applied in the given case as we have two classes: outliers and non-outliers, and we are mostly interested in the performance of the retrieval of outlier instances.

$$Precision = \frac{pos_{retrieved}}{pos_{retrieved} + neg_{retrieved}} \qquad Recall = \frac{pos_{retrieved}}{pos_{retrieved} + pos_{rejected}} \tag{4.2}$$

While the annotations shows a high overall annotator agreement with a F_1 score of 0.918 (prec = 0.962 rec = 0.879), the comparison of the outlier class reveals a substantial disagreement: F_1 = 0.262 (rec = 0.26, prec = 0.44), which was masked by the comparably large number of non-outlier instances.

The low F_1 score hints at the high subjectivity of this task and illustrates the difficulty to obtain an objective evaluation measure. In order to assure accurate outlier annotations, we conducted another annotation iteration, this time based on alarm confirmations. We fine-tuned the free parameters of our combined approach on one of the annotations and asked the annotators to confirm or reject the automatically detected outliers. Then we merged the annotation from the first annotation round with the alarm confirmations for each annotator. This approach leads to a small increase of overall agreement to F_1 0.934 (prec = 0.932, rec = 0.935) but results in a considerably improved outlier agreement of F_1 = 0.62 (rec = 0.64, prec = 0.62). Remember, as our focus lies on outliers, we are especially interested in a high outlier agreement.



(a) (b) (c)

Figure 4.8: Three different genotype representations. Figure (a) is a simple multicoloured plot of the repetitions. Figure (b) and (c) additionally show a large part of the rest of the dataset in light grey, enabling to see the repetitions in relation to the general sequence distribution. Note that (b) is uni coloured and (c) multicoloured as (a) .

4.4.2 Performance

We have fine-tuned and evaluated all three approaches presented in the previous section on the manually labelled data set. Each approach was run with different parameters in order to find an optimal setting. For the evaluation we combined the annotations obtained from the two annotators in two different fashions. The first can be regarded as a conjunction of the two sets, we consider a data instance as an outlier only if both annotators agree. The second version is a union, meaning that an instance is considered as outlier if at least one of the annotators flagged it correspondingly. The evaluation was performed using the F_1 score as defined in equation 4.1 focusing on the outlier class, for reasons of completeness we also computed the overall F_1 score for certain experiments.

The results are reported by Table 4.1. The table shows multiple parameter settings for each approach and the corresponding performance. We can observe that a very simple idea such as the lower bound approach can lead to impressive performance as an F_1 score of 0.676 on the union annotation set comparable to the more sophisticated neighbourhood approach which obtained an F_1 score of 0.666 on this test set. Apart from their similar performance on this dataset, we expect the two approaches to capture different notions of outliers. Where the lower bound approach considers only isolated and bad performing time series as outliers, the neighbourhood approach finds isolated series regardless of the direction. Indeed, our expectations are confirmed by an increase of the F_1 score for the combination approach.

The two base approaches have also been combined in a union and conjunction manner, where $Ensemble_{conj}$ means that only data instances, where both approaches raise an alarm are flagged as outlier and $Ensemble_{union}$ raises an alarm if at least one of the approaches reports an outlier match. Note that for the ensemble methods we also evaluated several settings and it is not necessarily the best setting of the individual approach which achieves best results in a combined manner, as can be seen for the case of $Ensemble_{conj}$ and the individual performance of the single approaches with the corresponding settings.

Method	Parameter	Dataset						
		Union				Conjunction		
		Prec	Rec	F_{pos}	F_{all}	Prec	Rec	F_{pos}
Lower bound	d_l: 285	0.657	0.695	**0.676**	0.916	0.410	0.937	**0.571**
	d_l: 250	0.486	0.768	0.595		0.293	1.0	0.453
Neighbourhood	d: 295 k:2	0.677	0.608	0.641		0.435	0.843	**0.574**
	d_n: 280 k:2	0.652	0.681	**0.666**	0.914	0.402	0.906	0.557
	d_n: 275 k:2	0.618	0.681	0.648		0.381	0.906	0.537
$Ensemble_{conj}$	d_n: 275, k:2, d_l: 250	0.934	0.623	**0.747**	0.930	0.630	0.906	**0.743**
$Ensemble_{union}$	d_n: 365, k:1, d_l: 285	0.64	0.695	0.666	0.912	-	-	

Table 4.1: Outlier detection performance table, the three different approaches have been evaluated on two variants of the obtained annotation sets, union and conjunction. We present recall, precision and F_1 score for both cases. F_{pos} represents the F-measure for the outlier class, whereas F_{all} is the measure for all classes, thus takes correctly classified non-outlier into account. Further d_n is the distance for the neighbourhood approach and d_l the distance for the lower bound approach.

5 Time series clustering

Clustering aims at finding the '*naturally*' appearing groups in a dataset. Unlike classification it does not require the prior definition of classes and thus is considered as an unsupervised method. The groups are chosen so that their elements are similar under a predefined similarity measure, meaning that they should minimise intra cluster variance while maximising inter cluster variance. It is the most commonly used method for pattern discovery [Fu, 2011] but this is not its only purpose, it can be used to get a better overview of the data, to reduce high dimensionality or it can serve as a preprocessing step, which structures the data and makes further analysis easier. The applications are manifold. For a more formal definition we rely on the propositions in [Hansen and Jaumard, 1997]:
A clustering C is the division of dataset D into M clusters fulfilling the following constraints:

$$C = \{c_1, c_2, \dots, c_M\}$$

$$c_j \neq \emptyset \qquad c_i \cap c_j = \emptyset \; | i, j = 1, 2, \dots M \wedge i \neq j \qquad \cup_{i=1}^{M} c_j = D \tag{5.1}$$

In the given context we use clustering to group similar genotype time series in order to ease further analysis, like the investigation on the origin of certain growing patterns or a faster detection of genotypes with a *surprising* behaviour.

As already discussed in Section 4.2.4 in the context of outlier detection, most techniques assume static data and therefore time series need a special treatment. A common approach is to consider a time series T of length N as a single data instance with N attributes and to apply the classical techniques. This is also applicable in the case of clustering. Another approach is the modification of classical clustering techniques by switching the similarity measure to a measure capable of handling time series data. Most clustering algorithms require a measure of proximity between the data instances, for a discussion on distance measures suitable for time series ,see Section 2.3.

5.1 Clustering methods

Classical clustering techniques can be separated into five categories: partitioning, hierarchical, density-based, grid-based and model-based methods [J, 2006, Liao, 2005]:

- Partitioning cluster algorithms divide a dataset D with N data instances into k partitions where $k \leq N$ and each partition contains at least one data instance. The partitions may be defined as hard, meaning that an object can be part of only one partition, or fuzzy, where the partition membership is defined by a degree of affiliation.
- Hierarchical clustering creates a tree of clusters from its data. The agglomerative version at first considers every element as a single cluster and merges them consecutively based on a merging criteria.
- Density-based approaches create clusters from data regions, which exceed a predefined density threshold. An interesting characteristic is that data points not exceeding this threshold are considered as noise.
- Grid-based clustering performs the clustering routines on a grid structure which is obtained by transforming the object space into the desired representation.
- And finally model-based approaches assume a model for each cluster and fit the model to the data.

This is just a listing of the prominent characteristics of the cluster categories. We will focus on hierarchical and density-based approaches, refer to [Liao, 2005, Xu and Wunsch, 2005, Fu, 2011] for a more in-depth discussion of the remaining categories.

5.1.1 Hierarchical clustering

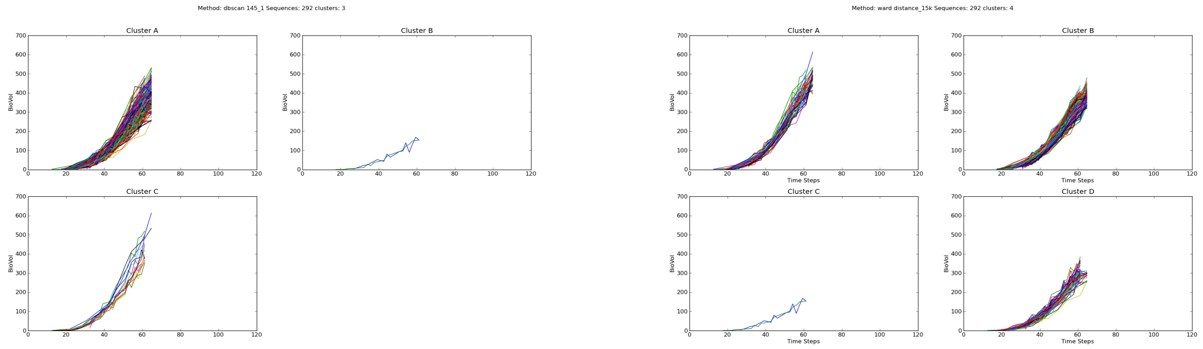
Hierarchical clustering is one of the oldest clustering methods, see survey of [Hansen and Jaumard, 1997], but its use is still well-established [Shumway, 2003, Rodrigues, 2008]. Hierarchical clustering methods create a tree of clusters from the given data. We distinguish between two different versions of this algorithm: a bottom-up approach, the agglomerative hierarchical clustering (HAC) and a top-down procedure, named divisive hierarchical clustering (DHC). Divisive hierarchical clustering (DHC) starts with one initial cluster containing all elements and proceeds by successively splitting the clusters in two until a given criteria is reached (for example when each element has its own cluster). The bottom-up version of this method, hierarchical agglomerative clustering (HAC), is used more frequently [Hansen and Jaumard, 1997]. It initially assigns each data instance to its own cluster and successively merges clusters until the reach of a predefined stop condition. The resulting data tree is shown in form of a dendrogram in Figure 5.3. Hierarchical clustering is useful to visualise the inherent structure of a dataset and it is a good method to subjectively evaluate the performance of a distance measure between data instances [Maimon and Rokach, 2005].

The cluster selection for a merge or a division can be adapted to individual needs. *Single linkage* measures the distance between clusters based on the closest elements belonging to different clusters. Then the clusters with minimal distance are merged. *Average linkage* computes the average distance between all elements of two clusters and the *centroid linkage* considers the cluster distance to be the Euclidean distance between the cluster centroids. All three approaches choose clusters with minimal distance for merging. A slightly different approach is the Ward's method. In this case the two clusters are selected for merging, whose combination will result in the smallest increase of the sum-of-squares variance [Liao, 2005]. Therefore the resulting variance value is computed for every possible merge and we finally execute the merge with a minimal increase of variance. This is very useful if a given context defines a low sum-of-squares as a criteria for a good clustering, as Ward's method will keep it as low as possible with each merge. Unfortunately theses approaches are computationally expensive, $O(N^2)$ complexity in time and space, and therefore are not suitable to handle high dimensional data [Xu and Wunsch, 2005]. Hierarchical clustering is a greedy method, thus suffering from the fact that already executed merges or splits can not be undone, therefore there is a trend to combine them with further clustering methods [Liao, 2005].

5.1.2 Density-based clustering

Density-based clustering methods assume that clusters appear as dense regions in a metric space. These methods search for highly dense regions in the dataset and consider them as separate clusters. This is similar to the single linkage method but cope with the phenomenon of chaining, where a chain of points can extend the cluster for a long distance, by avoiding the addition of data instances causing a notable drop in average cluster similarity [Everitt et al., 2011].

A relatively well-known density-based clustering algorithm was introduced by Ester et al. [1996], DBSCAN, which assumes that clusters appear in dense and concentrated regions and is designed to find clusters of arbitrary shape. An interesting property of this algorithm is that it inherently copes with noise in the dataset, by declaring dense regions as clusters and regions of low-density as noise. This approach requires the user to define two parameters: a minimum distance d and a minimum number of neighbours n. Correspondingly a point p requires at least n neighbours in the radius of d in order to form a cluster. We distinguish between core and non-core objects of a cluster. A core object is, as explained for point p, an object with at least n elements in its neighbourhood. When the neighbourhoods of two core objects overlap they are simply merged to one cluster, enabling the detection of clusters with arbitrary shapes. Further objects are considered as 'direct density reachable' from point p if they are within the



(a)

(b)

Figure 5.1: Clustering results for DBSCAN (a) with the parameters d 145, n 1 and HAC Ward cluster result in (b) with the stop condition at a cluster distance of 15 000.

distance d and they are just 'density reachable' if we can establish a chain of core objects to the object in question. The border of a cluster contains the 'border objects' which do not have the necessary number of neighbours in their neighbourhood but are part of the cluster if they are density reachable. Elements not assigned to any cluster are considered as noise. Further, it finds the number of clusters automatically, which distinguishes it from for example k-means.

DBSCAN is not only useful as a pure clustering algorithm but also for the detection of noise. Unfortunately it does not perform well on sets of varying density and in high-dimensional space where the data is often sparse [Everitt et al., 2011]. Birant and Kut [2007] adapted the algorithm to handle spatial and temporal data and to cope with variable density by using a spatial information as an additional clustering criterion. The acceptable radius for a cluster is then adjusted based on the spatial information.

5.2 Genotype clustering

The goal of genotype clustering is to find similar behaving genotypes in order speed up further phenotypic analysis and to ease the study of genotype-phenotype interaction. In more detail we cluster average biomass accumulation time series for each genotype, but the here presented approaches can be applied to any other phenotypic time series, like the evolution of leaf area or the plants transpiration rate. As the clustering results will be the base for further analysis, the expert's expectation of an optimal clustering outcome is: 4 to 6 more or less balanced clusters, some clusters representing the mediocre performing plants and two for the really good and really bad performers. The following subsections present the actual approach of biomass time series clustering and the evaluation of our results.

5.2.1 Clustering approach

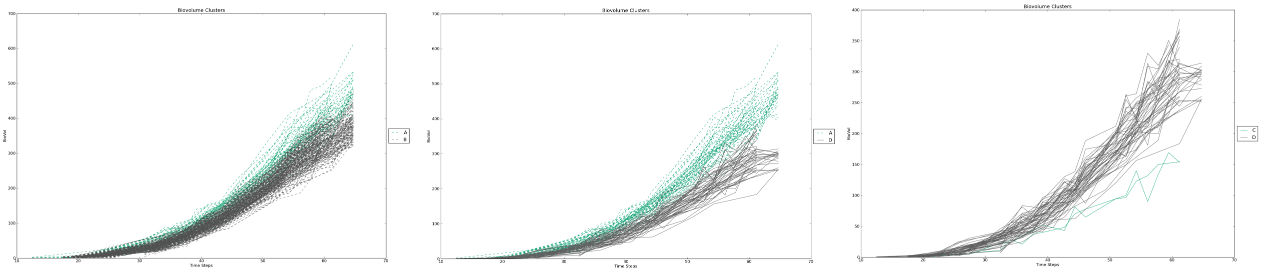
The clustering of genotype time series is the next step after the removal of outliers discussed in Section 4.1. So far a genotype has been represented by its repetitions, as we are mostly interested in the performance of an entire genotype and not in individual plants. We derive an average genotype performance based on its repetitions. Recall that the time series at hand are not uniformly sampled, therefore we can not simply iterate through the time steps and average the time series values for each point. Thus in order to obtain the average time series we slightly adapted the basic proceeding using a nearest-neighbour approach to determine the corresponding time steps within the repetitions. This is illustrated by the Algorithm 3.

```
**Algorithm** *calc_avg_time_series(repetitions)*
    longest_rep = get_longest_rep(repetitions);
    avg_rep = [], to_avg=[];
    **for** *time_step* ***in*** *longest_rep.time* **do**
        **for** *rep* ***in*** *repetitions* **do**
            time_match = get_nearest_neighbour(time_step, rep.time);
            to_avg.add(rep.valueAt(time_match)) ;
        **end**
        avg_rep.append(avg(to_avg), time_step) ;
        to_avg.clear();
    **end**
    **return** avg_rep;
```

Algorithm 3: Calculation of average genotype performance

Keeping in mind the expectation of four to six balanced clusters, we clustered the 300 average genotype time series using the density-based algorithm DBSCAN and four versions of the HAC approach, Single Linkage, Average Linkage, Centroid Linkage and Ward. Each clustering algorithm was run several times with varying parameters in order to approximate the expected clustering results. Both types of algorithms require a time series similarity measure to decide on cluster affiliation. In Section 2.3 we reviewed commonly used similarity measures for time series and in Chapter 4 we showed that Dynamic Time Warping proves to be suitable to represent the similarity of the given time series. Therefore we used DTW for all further approaches requiring time series similarity.

Due to the very high density of the entire dataset, the best clustering we could achieve using DBSCAN is shown in Figure 5.1a. When the distance parameter d was too high, almost all time series were assigned to a single cluster and a lower d value caused too many time series to be classified as noise. Therefore DBSCAN was not retained for further investigations.
Some notable observations in the results of HAC are that Single Linkage and Average Linkage show results comparable to DBSCAN. Beside the fact that they do not classify any time series as noise, they show difficulties to create balanced clusters. Unlike Single Linkage the Ward algorithm manages to create surprisingly balanced data clusters. This striking difference is depicted in Figure 5.3. We can see that the Average Linkage dendrogram is drawn to the right whereas the Ward dendrogram shows a comparably balanced cluster tree.



(a) (b) (c)

Figure 5.2: Three cluster pairs of the Ward clustering result in Figure 5.1b. For each clustering result we created such cluster pair plots which were judged by the experts on their meaningfulness.

Algorithm	Parameter	Clusters	Agreement
ward	15k	4	0.93
	10k	6	0.83
	7k	8	0.75
centroid	7k	3	1
	5k	2	-
	4k	4	0.66
average	4k	4	0.83
	3k	4	0.75
	2k	7	0.66

Table 5.1: Expert evaluation of clustering results

5.2.2 Evaluation

Clustering evaluation can be divided into two fields, intrinsic and extrinsic evaluation. Intrinsic evaluation evaluates the cluster based on internal criteria, like inter and intra cluster variance assuming that objects within a cluster should be highly similar and objects of different clusters should not. The problem with this kind of evaluation is that while a clustering might be intrinsically optimal it is not necessarily meaningful in a given context. Regarding the clustering results we obtained with DBSCAN (in Figure 5.1a), as the data is very dense it puts a large part of the sequences into one single cluster, this cluster might be intrinsically optimal, but does not help us separate the rather good and rather bad growing genotypes. Therefore this method is not necessarily meaningful in our context.

Extrinsic evaluation uses additional information, not used for the clustering itself, for evaluation, often in the form of labelled data or expert judgement. Based on the vantage of clustering, not requiring any training data, it is used in fields where almost no or only few labelled data is available. As a result extrinsic evaluation of clustering turns out to be difficult.

Despite the lack of available labelled clustering data, we want to decide on the best performing clustering algorithm in the given context. Considering that a purely intrinsic evaluation will not give any insight of the utility of our result, we conducted an expert interview in order to obtain the best clustering outcome.

Two experts, familiar with the data, who will use the clustering outcome as base for future analysis, were presented several clustering results of the different clustering algorithms. The results were represented by a plot for each cluster. Additionally we provided a further plot for each pair of clusters to highlight the cluster difference and to ease the expert decision whether this data separation is meaningful or not, as shown in Figure 5.2. Thus, for a clustering with four clusters we obtained six expert judgments, stating to which degree they agree on this data separation. The average expert agreement on each clustering result is shown in Table 5.1. Note that the centroid linkage clustering outcome earned full expert agreement. This is due to a special case, it created reasonable clusters by providing one large cluster for the average performers a cluster for the above average performers and a cluster for the few badly performing genotypes. The expert opinion at this point is that while the cluster separation is entirely reasonable these clusters do not necessarily offer the best basis for future analysis. Further analysis will focus not only on the exceptionally good and bad performers but is also interested in the above average performing genotypes; a higher granularity is desirable in the average performers cluster. The desired granularity is achieved by the clustering result using Ward's method, which splits the moderate performing plants in two while maintaining separate clusters for outstanding genotypes. Thus this clustering is considered as being suitable to serve further analysis.

5.3 Subspace clustering

especially When performing classical clustering in high dimensional data, it appears that certain elements have uncorrelated features. This can impede the effective cluster detection[Madeira and Oliveira, 2004]. Therefore clustering algorithms simultaneously clustering the rows and columns of a data matrix have been proposed. The obtained clusters consist of a subset of the rows and columns of the original data matrix. These algorithms are called *Biclustering* and are often applied to gene expression data. Similar algorithms can be found in the field of document clustering referred to as *Co-clustering*. Tanay et al. [2005], Madeira and Oliveira [2004] introduce the challenges addressed by biclustering and discuss different approaches. Subspace clustering is another extension to the normal clustering approach. It addresses the issue of uncorrelated features in high dimensional data by the localisation of the cluster search to only relevant dimensions [Parsons et al., 2004].

Given a matrix with phenetic characteristics and the corresponding genetic asset, subspace clustering can be of interest as it can contribute to the study of genotype-phenotype interaction. We would like to answer the question of, apart from the environmental influences and the developmental variability, which genetic elements are responsible for certain phenotypic behaviour.

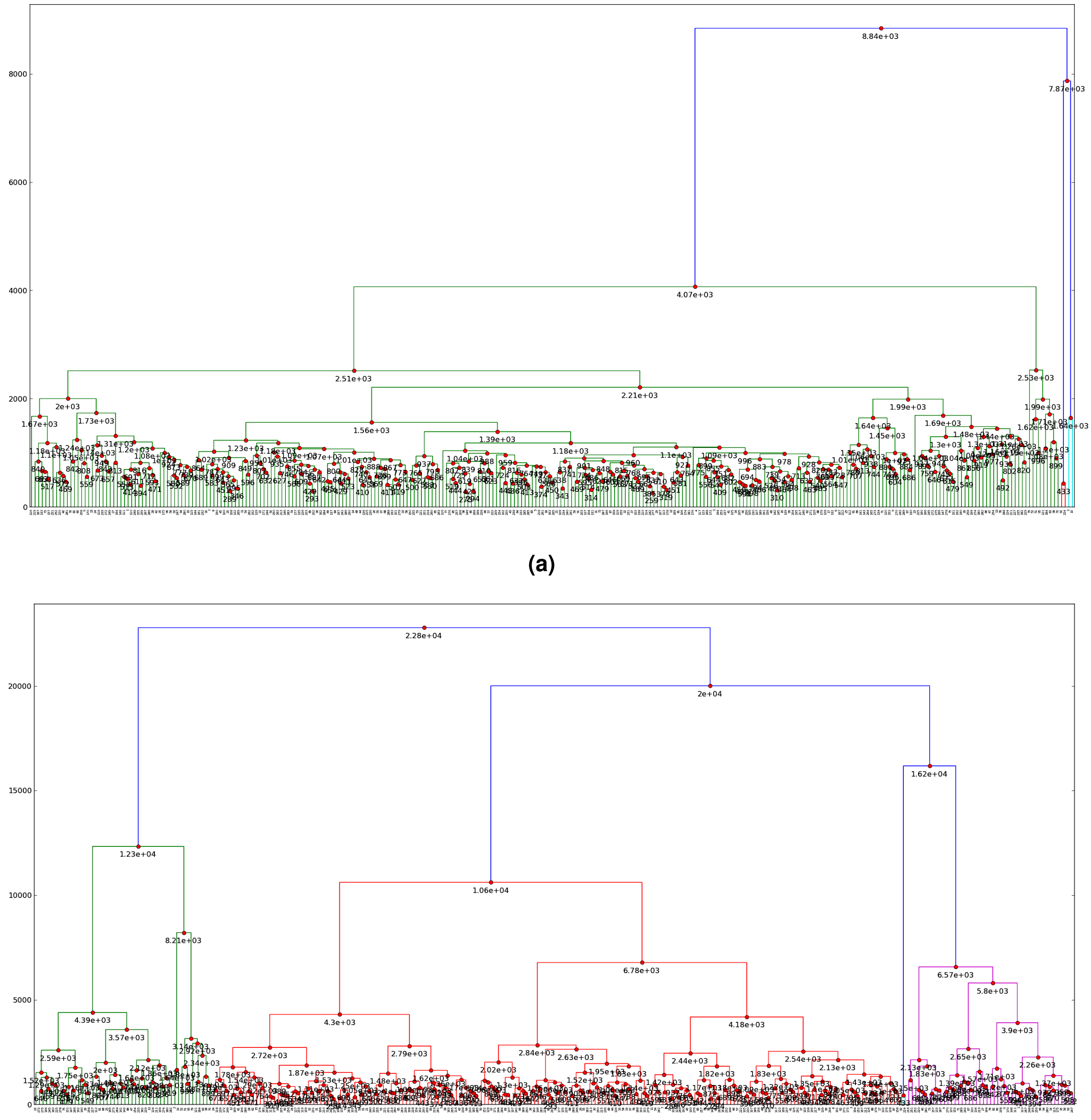
5.3.1 Data

We obtained a genotype marker matrix containing marker for most of the genotypes used in the ZB 2012 experiment. The matrix contains 50 000 marker columns for each of its genotypes.
A genetic marker is a gene with two or more alternative forms, in our case just two. One version to express this aspect is via probabilities that a marker develops in one or the other way. This has been done in the format at hand. A “0” corresponds to the probability that a marker will be manifested as one state and “2” as the other. A “1” means that the probability is about equal for both states. Thus the marker representation of a genotype is a vector of length 50 000 consisting of $\{0, 1, 2\}$.

5.3.2 Marker clustering

The main goal is to reveal genotypic elements most probably responsible for a certain phenotypic trait. The genotype markers embody the gentotypic elements, where we need to find the most relevant ones with regard to a phenotype trait. Therefore we need to create an association of phenotype traits and markers. In the previous part of this section we revealed groups of genotypes showing similar characteristics of biomass accumulation. The obtained clusters can be used to represent a phenotypic trait - for example the cluster containing very high climbing biomass graph represent the trait of fast and well growth. Thus the clusters can be used to divide the marker matrix into subspaces. This means that we will have a part of the matrix corresponding to the well-growing genotypes, not so well-growing genotypes and so forth.

Once obtained, the subdivided marker matrix use information theoretic analysis and clustering to obtain the most relevant marker of a submatrix. For the evaluation of these results we need expert judgement. Of course we can perform an intrinsic evaluation of the clusters, but an intrinsicly optimal cluster is possibly not of interest for the phenotypic studies. Note that the initial marker matrix contains 50 000 columns, thus the resulting clusters may contain 10 000 columns or more, additionally complicating the expert evaluation. Therefore, this part requires further effort and time for the adequate visualisation and the evaluation of results, which goes beyond the scope.



(a)

(b)

Figure 5.3: Dendrogram representation of HAC with Average Linkage (a) and Ward (b) on the ZB 2012 dataset. We can see that Ward's method manages to keep the clusters balanced, whereas the average linkage method is drawn to the right and shows a bulk of time series, depicted in green, which will appear in one single cluster.

6 Conclusion

The main asset of this work is its interdisciplinary nature. A cooperation between researchers of entirely different fields is not always easy. Each domain has its own customs, an individual way to approach problems and an appropriate set of tools. In the domain of phenotyping, a common practice and even a basic procedure in order to make sense of one's data is to apply pure Statistics and draw corresponding conclusions. Therefore in domains where all proceedings are based on statistical analysis and results of statistical models purely data driven data mining approaches might appear not sound and inappropriate. Therefore to make such a cooperation work, it requires a lot of patience and openness to new approaches on both sides. This thesis is a result of such a cooperation and "makes proof" of the fact that well-established techniques in one domain can be of great benefit even in distant fields.

Similar to the introduction of DTW to the data mining community by Berndt and Clifford [1994] which caused it to become an integral part of this field, DTW appears to be also a great asset in the domain of phenotyping. It shows good results as the similarity measure for phenomic time series in applications such as outlier detection and clustering. We observed that while DTW is able to deal with series of different lengths, an approximate length adjustment improved the similarity performance. DTW was originally designed to deal with acceleration and deceleration in time series but in our case the difference in time series lengths often result from the lack of measurements at the beginning or the end of an experiment. Thus, as the classical DTW approach requires the alignment of the beginning and the end of two series we obtained not necessarily meaningful similarity measures. This problem does not always have to be solved by the cut off of some measurements but could also be approached via a relaxation of DTW's alignment criterion.

We cover the task of outlier detection in time series, essential for meaningful outcomes of further analysis. Note that we focus on complex contextual outliers, thus addressing aberrant genotype repetitions instead of single points in time series. We show that with simple threshold and nearest-neighbour-based approaches we achieve an F_1 score of 0.93 for the classification of outliers and normal instances. For now this is just an algorithm applied to a set of data. A very useful next step is based on engineering work to expand this approach to an outlier detection system with a graphical user interface to facilitate the configuration. Due to the lack of annotated data we evaluated the outlier detection approach only on one dataset. Thus we do not know how well it generalises to other data. A key feature of such a platform would be a slider to control the sensitivity of the outlier detection control by changing the thresholds, enabling its adjustment to datasets of plants with different growing patterns.
A different approach to the task of outlier detection is to exploit the setup of the system usage. The outliers detected by the system will be reviewed by an expert prior to their removal. The confirmation or rejection of detected outliers can be used to train an online classifier. The classifier can be initially trained on some labelled examples and fine tuned via user feedback during its usage. This might result in a more precise adjustment to a dataset than just a change of threshold and enables self-improving of the classifier over time.

Probably the most important part of this work is the grouping of similar growing patterns. To date, for an analysis of best performing genotypes in a given environmental condition, it was necessary to verify each repetition growing in this condition for every genotype. Based on the observed growing patterns, the genotypes had to be manually sorted into appropriate performance groups. This proceeding can now be automatised. We extract a genotype's average performance based on its repetitions, and detect similar behaviour, present in the dataset using clustering. Four different hierarchical agglomerative clustering methods were compared in order to find the best suited algorithm. We observed, that despite the high density of the given dataset, Ward's method manages best to keep the clusters balanced. A following expert interview revealed as well, that the clusters obtained by Ward's method are the most reasonable and suit best for further analysis. Moreover, we noticed that the density-based approach DBSCAN has difficulties to obtain balanced clusters which led to its exclusion from following investigations. Therefore, it is of interest if a density-based algorithm able to handle varying densities, such as OPTICS [Ankerst et al., 1999], will yield any better results. Despite the good results of Ward's method, we have to take into account its high computational complexity which limits its usage to comparatively small datasets, which has however not posed a problem in the scope of this work. Hence, a natural next step should be the comparison of less computationally complex clustering algorithms such as Self-organizing maps (SOM) [Kohonen, 1982].

Subspace clustering is another very promising subject of this thesis. Recall, that one of the goals of phenotyping, is to study the genotype-phenotype interaction. For each previously obtained genotype cluster, we clustered the corresponding genetic marker matrix with the goal to reveal the most relevant marker groups responsible for a certain phenotypic behaviour. This seems to be a very promising track to shed light on the interaction of genotypes and phenotypes. Due to the time frame of a thesis, the presented approach could barely be touched upon and hence still requires an evaluation of the retrieved marker clusters. Approaches considering this topic, may contain an information theoretic analysis of the clusters to obtain the most relevant one, and the elaboration of an evaluation procedure to assure the significance of our findings. An evaluation approach should especially consider knowledge from the domain of plant genetics, such as common marker locations or marker families.

We have shown that DTW is a well-suited similarity measure for time series, that Hierarchical clustering can be applied on time series and that especially Ward's method shows good performance on a dense dataset. But this is not necessarily the main contribution of this thesis. This work highlights, that while there are domains penetrated by techniques based on data mining, like the financial sector, elections, genetics, there are still fields offering interesting challenges where the analysis of large amounts of data can be of great benefit, but is still in its infancy.

List of Figures

List of Tables

List of Algorithms

Bibliography

J Aach and G M Church. Aligning gene expression time series with time warping algorithms. *Bioinformatics (Oxford, England)*, 17(6):495–508, July 2001. ISSN 1367-4803. URL http://www.ncbi.nlm.nih.gov/pubmed/11395426.

C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. *ACM SIGMOD Record*, 30(2):37–46, June 2001. ISSN 01635808. doi: 10.1145/376284.375668. URL http://portal.acm.org/citation.cfm?doid=376284.375668.

R Agrawal, C Faloutsos, and A Swami. Efficient similarity search in sequence databases. *Proceedings of the Fourth International Conference on Foundations of Data Organization and Algorithms*, 1993. URL http://link.springer.com/chapter/10.1007/3-540-57301-1_5.

R Agrawal, K Lin, H. S. Sawhney, and K Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. *Proceedings of the 21st International Conference on Very Large Databases*, pages 490–501, 1995. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.40.4034\&rep=rep1\&type=pdf.

M Ankerst, M Breunig, H. P Kriegel, and J Sander. OPTICS: ordering points to identify the clustering structure. *ACM SIGMOD Record*, pages 49–60, 1999. URL http://dl.acm.org/citation.cfm?id=304187.

M.U. Bant. If corn is biofuels king, tropical maize may be emperor, 2007. URL http://www.eurekalert.org/pub_releases/2007-10/uoia-ici101507.php.

G. A Barreto. Time series prediction with the self-organizing map: A review. *Perspectives of neural-symbolic integration*, 2007. URL http://link.springer.com/chapter/10.1007/978-3-540-73954-8_6.

S. D Bay and M Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. *Conference on Knowledge discovery and data mining*, 2003. URL http://dl.acm.org/citation.cfm?id=956758.

DJ Berndt and J Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. *KDD workshop*, pages 359–370, 1994. URL http://www.aaai.org/Library/Workshops/1994/ws94-03-031.php.

D Birant and A Kut. ST-DBSCAN: An algorithm for clustering spatialâĂŞtemporal data. *Data & Knowledge Engineering*, 60(1):208–221, January 2007. ISSN 0169023X. doi: 10.1016/j.datak.2006.01.013. URL http://linkinghub.elsevier.com/retrieve/pii/S0169023X06000218.

R Blender, K Fraedrich, and F Lunkeit. Identification of cyclone track regimes in the North Atlantic. *Quarterly Journal of the . . .*, 1997. URL http://onlinelibrary.wiley.com/doi/10.1002/qj.49712353910/abstract.

B. R Bochner. New technologies to assess genotype-phenotype relationships. *Nature reviews. Genetics*, 4(4):309–14, April 2003. ISSN 1471-0056. doi: 10.1038/nrg1046. URL http://www.ncbi.nlm.nih.gov/pubmed/12671661.

R Bonhomme. Bases and limits to using ’degree.day’ units. *European Journal of Agronomy*, 13(1):1–10, July 2000. ISSN 11610301. doi: 10.1016/S1161-0301(00)00058-7. URL http://linkinghub.elsevier.com/retrieve/pii/S1161030100000587.

MM Breunig, H. P Kriegel, RT Ng, and Jörg Sander. LOF: identifying density-based local outliers. *ACM Sigmod Record*, pages 93–104, 2000. URL http://dl.acm.org/citation.cfm?id=335388.

K Chakrabarti, E. J Keogh, S Mehrotra, and M. J Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. *ACM Transactions on Database Systems*, 27(2):188–228, June 2002. ISSN 03625915. doi: 10.1145/568518.568520. URL http://portal.acm.org/citation.cfm?doid=568518.568520.

S Chakrabarti, S Sarawagi, and B Dom. Mining surprising patterns using temporal description length. *VLDB*, pages 1–12, 1998. URL http://pdf.aminer.org/000/642/238/mining_surprising_patterns_using_temporal_description_length.pdf.

F. K Chan, A. W Fu, and C Yu. Haar wavelets for efficient similarity search of time-series: with and without time warping. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):686–705, May 2003. ISSN 1041-4347. doi: 10.1109/TKDE.2003.1198399. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1198399.

V Chandola. Outlier detection: A survey. *... Surveys, to ...*, 2007. URL http://www.bradblock.com.s3-website-us-west-1.amazonaws.com/Outlier_Detection_A_Survey.pdf.

V Chandola, A Banerjee, and V Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 2009. URL http://dl.acm.org/citation.cfm?id=1541882.

N. V Chawla, N Japkowicz, and A Kotcz. Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations ...*, 6(1):2000–2004, 2004. URL http://dl.acm.org/citation.cfm?id=1007733.

B Chiu, E. J Keogh, and S Lonardi. Probabilistic discovery of time series motifs. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*, page 493, 2003. doi: 10.1145/956804.956808. URL http://portal.acm.org/citation.cfm?doid=956750.956808.

R Courtice. Radio Technical Standards : BWAV Specification. (March 2010):1–11, 2010.

J. H Martin A-Kehler K Vander Linden N Ward D, Jurafsky. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*, volume 2. MIT Press, 2000.

Kaustav Das and Jeff Schneider. Detecting anomalous records in categorical datasets. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, page 220, 2007. doi: 10.1145/1281192.1281219. URL http://portal.acm.org/citation.cfm?doid=1281192.1281219.

TG Dietterich. Ensemble methods in machine learning. *Multiple classifier systems*, 2000. URL http://link.springer.com/chapter/10.1007/3-540-45014-9_1.

H Ding, G Trajcevski, and P Scheuermann. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008. URL http://dl.acm.org/citation.cfm?id=1454226.

L Ertoz, M Steinbach, and V Kumar. A new shared nearest neighbor clustering algorithm and its applications. *... Data and its Applications ...*, pages 1–15, 2002. URL http://www-users.cs.umn.edu/~kumar/papers/siam_hd_snn_cluster.pdf.

E Eskin, A Arnold, P Michael, P Leonid, and S Sal. A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data. *Applications of data mining in computer security*, 3(40):77–101, 2002.

P. Esling and C. Agon. Time-series data mining. *ACM Computing Surveys*, 45(1):1–34, November 2012. ISSN 03600300. doi: 10.1145/2379776.2379788. URL http://dl.acm.org/citation.cfm?doid=2379776.2379788.

M Ester, H. P Kriegel, J Sander, and Xi Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD*, 1996. URL http://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf.

B S. Everitt, S Landau, M Leese, and D Stahl. *Cluster Analysis*. Wiley Series in Probability and Statistics. John Wiley & Sons, Ltd, Chichester, UK, January 2011. ISBN 9780470977811. doi: 10.1002/9780470977811. URL http://doi.wiley.com/10.1002/9780470977811.

L. Cabrera-Bosquet I. Alles F. Tardieu, F. Masseglia. Phenoarch data analysis discussion notes, June 2013.

C Faloutsos and H. V. Jagadish. A signature technique for similarity-based queries. *IEEE*, 1997. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=666899.

C Faloutsos, M Ranganathan, and Y Manolopoulos. Fast subsequence matching in time-series databases. 1994. URL http://dl.acm.org/citation.cfm?id=191925.

France-Inflation.com. Evolution du prix de l'essence en France. URL http://france-inflation.com/graph_carburants.php.

T Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24 (1):164–181, February 2011. ISSN 09521976. doi: 10.1016/j.engappai.2010.09.007. URL http://linkinghub.elsevier.com/retrieve/pii/S0952197610001727http://www.sciencedirect.com/science/article/pii/S0952197610001727.

D.J. Futuyma. *Evolutionary Biology SEC.Ed.* Sinauer Associates, Incorporated, 1986. ISBN 9780878931835. URL http://books.google.de/books?id=zaKSJQAACAAJ.

P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1-2):18–28, February 2009. ISSN 01674048. doi: 10.1016/j.cose.2008.08.003. URL http://linkinghub.elsevier.com/retrieve/pii/S0167404808000692.

C Granier. Individual Leaf Development in Arabidopsis thaliana: a Stable Thermal-time-based Programme. *Annals of Botany*, 89(5):595–604, May 2002. ISSN 03057364. doi: 10.1093/aob/mcf085. URL http://aob.oupjournals.org/cgi/doi/10.1093/aob/mcf085.

C Granier, L Aguirrezabal, K Chenu, S. J. Cookson, M. Dauzat, P Hamard, J Thioux, G Rolland, S Bouchier-Combaud, A Lebaudy, B Muller, T Simonneau, and F Tardieu. PHENOPSIS, an automated platform for reproducible phenotyping of plant responses to soil water deficit in Arabidopsis thaliana permitted the identification of an accession with low sensitivity to soil water deficit. *The New phytologist*, 169(3):623–35, January 2006. ISSN 0028-646X. doi: 10.1111/j.1469-8137.2005.01609.x. URL http://www.ncbi.nlm.nih.gov/pubmed/16411964.

F. E Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 1974. URL http://www.tandfonline.com/doi/abs/10.1080/00401706.1969.10490657.

V Guralnik and J Srivastava. Event detection from time series data. *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '99*, pages 33–42, 1999. doi: 10.1145/312129.312190. URL http://portal.acm.org/citation.cfm?doid=312129.312190.

D Gusfield. *Algorithms on strings, trees and sequences: computer science and computational biology*. 1997. URL http://books.google.com/books?hl=en\&lr=\&id=Ofw5w1yuD8kC\&oi=fnd\&pg=PP1\&dq=Algorithms+on+Strings,+Trees,+and+Sequences:+Computer+science+and+computational+biology\&ots=k1oCGAt7F7\&sig=AEK7nIyQq1n_THB0RUR4TJuXhgE.

A Guttman. R-trees: A dynamic index structure for spatial searching. *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, 1984. URL http://dl.acm.org/citation.cfm?id=602266.

P Hansen and B Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79(1-3): 191–215, October 1997. ISSN 0025-5610. doi: 10.1007/BF02614317. URL http://link.springer.com/10.1007/BF02614317.

V Hautamäki, I Kärkkäinen, and P Fränti. Outlier detection using k-nearest neighbour graph. *Pattern Recognition, 2004. ...*, pages 4–7, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1334558.

Z He, X Xu, and S Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650, June 2003. ISSN 01678655. doi: 10.1016/S0167-8655(03)00003-5. URL http://linkinghub.elsevier.com/retrieve/pii/S0167865503000035.

I.H. Herskowitz. *Principles of genetics*. Macmillan, 1977. ISBN 9780023539305.

VJ Hodge and J Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, (1969):85–126, 2004. URL http://link.springer.com/article/10.1007/s10462-004-4304-y.

M Kamber J Pei J, Han. *Data mining: concepts and techniques*. Morgan kaufmann, 2006.

W Johannsen. *Elemente der exakten Erblichkeitslehre*. Fischer Jena, 1909. URL http://publikationen.stub.uni-frankfurt.de/files/9149/johannsen--elemente_schnupper.pdf.

W Johannsen. The genotype conception of heredity. *The American Naturalist*, 45(531):129–159, 1911. URL http://www.jstor.org/stable/10.2307/2455747.

G. H John. Robust Decision Trees: Removing Outliers from Databases. *KDD*, 1995. URL http://www.aaai.org/Papers/KDD/1995/KDD95-044.pdf.

K Kalpakis, D Gada, and V Puttagunta. Distance measures for effective clustering of ARIMA time-series. *Data Mining, 2001. ICDM ...*, 2001. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=989529.

E. J Keogh and S Kasetty. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and Knowledge Discovery*, 7(4):349–371, 2003. URL http://www.springerlink.com/index/G7535342U0781722.pdf.

E. J Keogh and J Lin. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and information systems*, 8(2):115–122, August 2005. ISSN 0219-1377. doi: 10.1007/s10115-004-0172-7. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1250910.

E. J Keogh and M. J Pazzani. Scaling up dynamic time warping for datamining applications. *Knowledge discovery and data mining*, In 6th ACM:285–289, 2000a. URL http://portal.acm.org/citation.cfm?doid=347090.347153http://dl.acm.org/citation.cfm?id=347153.

E. J Keogh and M. J Pazzani. A simple dimensionality reduction technique for fast similarity search in large time series databases. *Knowledge Discovery and Data Mining*, 2000b. URL http://link.springer.com/chapter/10.1007/3-540-45571-X_14.

E. J Keogh, K Chakrabarti, M. J Pazzani, and S Mehrotra. Locally adaptive dimensionality reduction for indexing large time series databases. *Proceedings of the 2001 ACM SIGMOD international conference on Management of data - SIGMOD '01*, pages 151–162, 2001. doi: 10.1145/375663.375680. URL http://portal.acm.org/citation.cfm?doid=375663.375680.

E. J Keogh, Selina Chu, D Hart, and M. J Pazzani. Segmenting time series: A survey and novel approach. *Data mining in time series ...*, 2004. URL http://www.asianscientist.com/books/wp-content/uploads/2013/06/5210_chap01.pdf.

T Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 69:59–69, 1982. URL http://link.springer.com/article/10.1007/BF00337288.

T. Kohonen. Exploration of very large databases by self-organizing maps. *Proceedings of International Conference on Neural Networks (ICNN'97)*, 1:PL1–PL6, 1997. doi: 10.1109/ICNN.1997.611622. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=611622.

T Komori and S Katagiri. Application of a generalized probabilistic descent method to dynamic time warping-based speech recognition. *Speech, and Signal Processing, 1992*, (4):497–500, 1992. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=225863.

Y Kou, C. T Lu, and D Chen. Spatial Weighted Outlier Detection. *SDM*, pages 613–617, 2006. URL http://www.siam.org/meetings/sdm06/proceedings/072kouy.pdf.

C Kruegel and D Mutz. Bayesian event classification for intrusion detection. *Computer Security Applications Conference*, Proceeding(Acsac):14–23, 2003. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1254306.

T W Liao. Clustering of time series dataâĂŤa survey. *Pattern Recognition*, 38(11):1857–1874, November 2005. ISSN 00313203. doi: 10.1016/j.patcog.2005.01.025. URL http://linkinghub.elsevier.com/retrieve/pii/S0031320305001305http://www.sciencedirect.com/science/article/pii/S0031320305001305.

Y Liao and VR Vemuri. Use of K-nearest neighbor classifier for intrusion detection. *Computers & Security*, 21(5): 439–448, 2002. URL http://www.sciencedirect.com/science/article/pii/S016740480200514X.

J Lin, E. J Keogh, Stefano Lonardi, and B Chiu. A symbolic representation of time series, with implications for streaming algorithms. *Proceedings of the 8th ACM SIGMOD*, pages 2–11, 2003. URL http://dl.acm.org/citation.cfm?id=882086.

S. C Madeira and A. L Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, 1(1):24–45, 2004. ISSN 1545-5963. doi: 10.1109/TCBB.2004.2. URL http://www.ncbi.nlm.nih.gov/pubmed/17048406.

M Mahner and M Kary. What exactly are genomes, genotypes and phenotypes? And what about phenomes? *Journal of Theoretical Biology*, 186(1):55–63, 1997. URL http://www.ncbi.nlm.nih.gov/pubmed/9176637.

O. Z Maimon and L Rokach. *Data mining and knowledge discovery handbook*. Springer US, Boston, MA, 2005. ISBN 978-0-387-09822-7. doi: 10.1007/978-0-387-09823-4. URL http://www.springerlink.com/index/10.1007/978-0-387-09823-4http://books.google.com/books?hl=en\&lr=\&id=S-XvEQWABeUC\&oi=fnd\&pg=PR21\&dq=Data+Mining+and+Knowledge+Discovery+Handbook\&ots=LBUncoDz0K\&sig=_v8iiDJiTokuYyWGyGjjuUkk0Ls.

N Meratnia and P Havinga. Outlier Detection Techniques for Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials*, 12(2):159–170, 2010. ISSN 1553-877X. doi: 10.1109/SURV.2010.021510.00088. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5451757.

Gi Mishne, D Carmel, and R Lempel. Blocking Blog Spam with Language Model Disagreement. *AIRWeb*, 2005. URL http://www.ra.ethz.ch/CDStore/www2008/airweb.cse.lehigh.edu/2005/proceedings.pdf\#page=11.

J. M Montes, H. F Utz, W Schipprack, B Kusterer, J Muminovic, C Paul, and A. E Melchinger. Near-infrared spectroscopy on combine harvesters to measure maize grain dry matter content and quality parameters. *Plant Breeding*, 125(6):591–595, 2006. ISSN 01799541. doi: 10.1111/j.1439-0523.2006.01298.x. URL http://doi.wiley.com/10.1111/j.1439-0523.2006.01298.x.

J. M Montes, A. E Melchinger, and J. C Reif. Novel throughput phenotyping platforms in plant genetic studies. *Trends in plant science*, 12(10):433–6, October 2007. ISSN 1360-1385. doi: 10.1016/j.tplants.2007.08.006. URL http://www.ncbi.nlm.nih.gov/pubmed/17719833.

S Moran. Precision Farming, 2001. URL http://earthobservatory.nasa.gov/IOTD/view.php?id=1139.

M Müller. Dynamic Time Warping. In *Information retrieval for music and motion*, pages 70–84. 2007. URL http://books.google.com/books?hl=en\&lr=\&id=kSzeZWR2yDsC\&oi=fnd\&pg=PA1\&dq=Information+Retrieval+for+Music+and+Motion\&ots=GqCpxgo71E\&sig=aV1DmD6-l9aPPmkWzoKEgPnjv-I.

C. S Myers and L. R Rabiner. A Comparative Study of Several Dynamic Time-Warping Algorithms for Connected-Word. *Bell System Technical Journal*, 1981. URL http://www3.alcatel-lucent.com/bstj/vol60-1981/articles/bstj60-7-1389.pdf.

M Nakazawa, T Ichikawa, A Ishikawa, H Kobayashi, Y Tsuhara, M Kawashima, K Suzuki, S Muto, and M Matsui. Activation tagging, a novel tool to dissect the functions of a gene family. *The Plant Journal*, 34(5):741–750, 2003. URL http://www.ncbi.nlm.nih.gov/pubmed/12787254.

A Nanopoulos, R Alcock, and Y Manolopoulos. Feature-based classification of time-series data. *Information processing and ...*, 0056, 2001. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.73.9555\&rep=rep1\&type=pdf.

A Panuccio, M Bicego, and V Murino. A Hidden Markov Model-based approach to sequential data clustering. *Structural, Syntactic, and Statistical Pattern Recognition*, 2002. URL http://link.springer.com/chapter/10.1007/3-540-70659-3_77.

Lance Parsons, Ehtesham Haque, and Huan Liu. Subspace clustering for high dimensional data. *ACM SIGKDD Explorations Newsletter*, 6(1):90–105, 2004. ISSN 19310145. doi: 10.1145/1007730.1007731. URL http://portal.acm.org/citation.cfm?doid=1007730.1007731.

A Patcha and J Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448–3470, August 2007. ISSN 13891286. doi: 10.1016/j.comnet.2007.02.001. URL http://linkinghub.elsevier.com/retrieve/pii/S138912860700062X.

P Patel, E. J Keogh, L Jessica, and S Lonardi. Finding motifs in time series. *ICDM*, 2002. URL http://cs.gmu.edu/~jessica/Lin_motif.pdf.

F Petitjean, A Ketterlin, and P Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, March 2011. ISSN 00313203. doi: 10.1016/j.patcog.2010.09.013. URL http://linkinghub.elsevier.com/retrieve/pii/S003132031000453X.

Chris Piro. Chat reaches 1 billion messages sent per day, 2009. URL http://www.facebook.com/note.php?note_id=91351698919.

R Poiré and F Tardieu. Vapour Pressure Deficit, 2013a. URL http://bioweb.supagro.inra.fr/phenoarch/images/sensor5.pdf.

R Poiré and F Tardieu. Temperature measured with thermocouples, 2013b. URL http://bioweb.supagro.inra.fr/phenoarch/images/sensor1.pdf.

D Pokrajac. Incremental local outlier detection for data streams. *... Data Mining, 2007. CIDM ...*, (April), 2007. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4221341.

K Rajendran, M Tester, and S. J Roy. Quantifying the three main components of salinity tolerance in cereals. *Plant, cell & environment*, 32(3):237–49, March 2009. ISSN 1365-3040. doi: 10.1111/j.1365-3040.2008.01916.x. URL http://www.ncbi.nlm.nih.gov/pubmed/19054352.

C. A Ratanamahatana and E. J Keogh. Making time-series classification more accurate using learned constraints. *Proceedings of SIAM ...*, 2004. URL http://books.google.com/books?hl=en\&lr=\&id=gcJVK9a9RR0C\&oi=fnd\&pg=PA11\&dq=Making+Time-series+Classification+More+Accurate+Using+Learned+Constraints\&ots=mOuhYVtm2o\&sig=54CjgFJS3Z2tK7otMecc1LjEh4g.

C. A Ratanamahatana and E. J Keogh. Three myths about dynamic time warping data mining. *International Conference on Data Mining*, pages 506–510, 2005. URL http://www.siam.org/proceedings/datamining/2005/dm05_50ratanamahatanac.pdf?q=wordspotting.

K.V Ravi Kanth, D Agrawal, A Abbadi, and A Singh. Dimensionality Reduction for Similarity Searching in Dynamic Databases. *Computer Vision and Image Understanding*, 75(1-2):59–72, July 1999. ISSN 10773142. doi: 10.1006/cviu.1999.0762. URL http://linkinghub.elsevier.com/retrieve/pii/S1077314299907622.

PP Rodrigues. Hierarchical clustering of time-series data streams. *Knowledge and Data ...*, X(X):1–12, 2008. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4407702.

J Ryan, M Lin, and R Miikkulainen. Intrusion detection with neural networks. *... in neural information processing systems*, pages 72–77, 1998. URL http://www.aaai.org/Papers/Workshops/1997/WS-97-07/WS97-07-013.pdf.

W Sadok, P Naudin, B Boussuge, B Muller, C Welcker, and F Tardieu. Leaf growth rate per unit thermal time follows QTL-dependent daily patterns in hundreds of maize lines under naturally fluctuating conditions. *Plant cell environment*, 30(2):135–146, 2007. URL http://www.ncbi.nlm.nih.gov/pubmed/17238905.

H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, February 1978. ISSN 0096-3518. doi: 10.1109/TASSP.1978.1163055. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1163055.

S Salvador, F. K Chan, and J Brodie. Learning States and Rules for Time Series Anomaly Detection. *FLAIRS Conference*, 2004. URL http://www.aaai.org/Papers/FLAIRS/2004/Flairs04-055.pdf.

Nicholas I Sapankevych and Ravi Sankar. Time series prediction using support vector machines: a survey. *IEEE Computational Intelligence Magazine, IEEE*, (May):24–38, 2009. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4840324.

R Sekar, A Gupta, and J Frullo. Specification-based anomaly detection: a new approach for detecting network intrusions. *Proceedings of the 9th . . .* , 2002. URL http://dl.acm.org/citation.cfm?id=586146.

Hagit Shatkay. The Fourier transform-A primer. *Brown University*, (November), 1995. URL https://intranet.dcc.ufba.br/pastas/gaudi/biometrica/papers/id/murilo/fourier_intro_Shatkay.pdf.

RH Shumway. Time-frequency clustering and discriminant analysis. *Statistics & probability letters*, 63(3):307–314, July 2003. ISSN 01677152. doi: 10.1016/S0167-7152(03)00095-6. URL http://linkinghub.elsevier.com/retrieve/pii/S0167715203000956http://www.sciencedirect.com/science/article/pii/S0167715203000956.

R. E Spears, P. J Oakes, N. E Ellemers, and S Haslam. *The social psychology of stereotyping and group life.* 1997. URL http://psycnet.apa.org/psycinfo/1997-97264-000.

Deepika Sripath. Efficient Implementations of Discrete Wavelet Transforms Using FPGAs. *Electronic Theses, Treatises and Dissertations*, Paper 1599, 2003. URL http://diginole.lib.fsu.edu/etd/1599/.

ZR Struzik and Arno Siebes. Measuring time series similarity through large singular features revealed with wavelet transformation. *Database and Expert Systems . . .* , 1999. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=795160.

Amos Tanay, R Sharan, and Ron Shamir. Biclustering algorithms: A survey. *Handbook of computational molecular . . .* , (May):1–20, 2005. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.118.8302\&rep=rep1\&type=pdf.

J Tang, Z Chen, A. W Fu, and D. W Cheung. Enhancing effectiveness of outlier detections for low density patterns. *Advances in Knowledge . . .* , 2002. URL http://link.springer.com/chapter/10.1007/3-540-47887-6_53.

F Tardieu. PhenoArch, 2013. URL http://bioweb.supagro.inra.fr/phenoarch.

NC Thuong and DT Anh. Comparing three lower bounding methods for DTW in time series classification. *Proceedings of the Third Symposium on . . .* , pages 200–206, 2012. URL http://dl.acm.org/citation.cfm?id=2350747.

P Tormene, T Giorgino, S Quaglini, and M Stefanelli. Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. *Artificial intelligence in medicine*, 45(1):11–34, January 2009. ISSN 1873-2860. doi: 10.1016/j.artmed.2008.11.007. URL http://www.ncbi.nlm.nih.gov/pubmed/19111449.

CAFNR University of Missouri. Corn Extension, 2013. URL http://plantsci.missouri.edu/grains/corn/.

M. Vlachos, D Gunopulos, and G. Das. Indexing Time-Series under Conditions of Noise. *Data mining in time series databases*, 57, 2004.

G Vogt, M Huber, M Thiemann, G van den Boogaart, O. J Schmitz, and C. D Schubart. Production of different phenotypes from the same genotype in the same environment by developmental variation. *The Journal of experimental biology*, 211(Pt 4):510–23, February 2008. ISSN 0022-0949. doi: 10.1242/jeb.008755. URL http://www.ncbi.nlm.nih.gov/pubmed/18245627.

A Walter, B Studer, and R Kölliker. Advanced phenotyping offers opportunities for improved breeding of forage and turf species. *Annals of botany*, 110(6):1271–9, November 2012. ISSN 1095-8290. doi: 10.1093/aob/mcs026. URL http://www.ncbi.nlm.nih.gov/pubmed/22362662.

X Wang, A Mueen, H Ding, G Trajcevski, P Scheuermann, and E. J Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309, February 2012. ISSN 1384-5810. doi: 10.1007/s10618-012-0250-5. URL http://link.springer.com/10.1007/s10618-012-0250-5.

C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: alternative data models. *Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No.99CB36344)*, pages 133–145. doi: 10.1109/SECPRI.1999.766910. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=766910`.

Stiftung Weltbevölkerung. Weltbevölkerung zum jahreswechsel 2012/2013. http://www.weltbevoelkerung.de/oberes-menue/presse/presse/presseinformationen/news-ansicht/display/weltbevoelkerung-zum-jahreswechsel-20122013.html, December 2012.

R Wu and M Lin. Functional mapping - how to map and study the genetic architecture of dynamic complex traits. *Nature reviews. Genetics*, 7(3):229–37, March 2006. ISSN 1471-0056. doi: 10.1038/nrg1804. URL `http://www.ncbi.nlm.nih.gov/pubmed/16485021`.

Y Wu, D Agrawal, and A El Abbadi. A comparison of DFT and DWT based similarity search in time-series databases. *Proceedings of the ninth international conference on Information and knowledge management - CIKM '00*, pages 488–495, 2000. doi: 10.1145/354756.354857. URL `http://portal.acm.org/citation.cfm?doid=354756.354857`.

R Xu and D Wunsch. Survey of clustering algorithms. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 16(3):645–78, May 2005. ISSN 1045-9227. doi: 10.1109/TNN.2005.845141. URL `http://www.ncbi.nlm.nih.gov/pubmed/18252358`.

Kenji Yamanishi and Jun-ichi Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*, page 676, 2002. doi: 10.1145/775107.775148. URL `http://portal.acm.org/citation.cfm?doid=775047.775148`.

Nong Ye and X Li. A markov chain model of temporal behavior for anomaly detection. *Proceedings of the 2000 IEEE Systems, Man, and . . .* , (4):6–7, 2000. URL `http://homepages.laas.fr/owe/METROSEC/DOC/WA1\_1.pdf`.

BK Yi and C Faloutsos. Fast time sequence indexing for arbitrary Lp norms. *VLDB*, pages 385–394, 2000. URL `http://repository.cmu.edu/compsci/553/`.

Ji Zhang and Hai Wang. Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance. *Knowledge and Information Systems*, 10(3):333–355, March 2006. ISSN 0219-1377. doi: 10.1007/s10115-006-0020-z. URL `http://link.springer.com/10.1007/s10115-006-0020-z`.

Yunyue Zhu and Dennis Shasha. Warping indexes with envelope transforms for query by humming. *Proceedings of the 2003 ACM SIGMOD international conference on on Management of data - SIGMOD '03*, page 181, 2003. doi: 10.1145/872773.872780. URL `http://portal.acm.org/citation.cfm?doid=872757.872780`.