LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Similarity Search in Medical Data

Katrin Haegler

München 2011

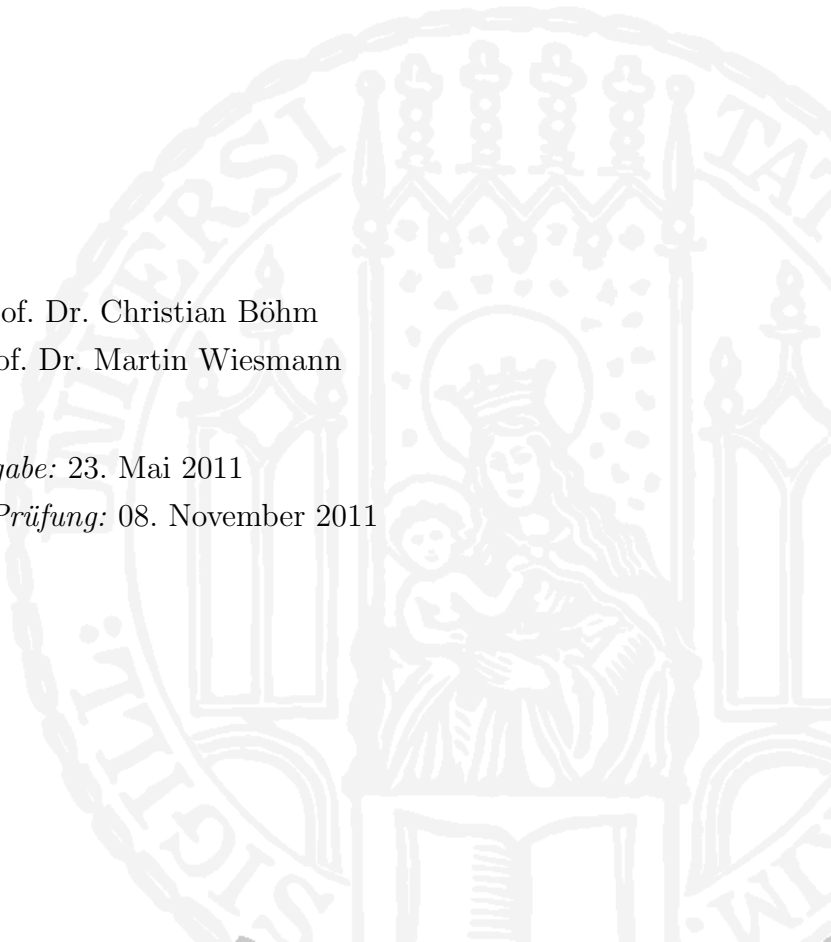
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Similarity Search in Medical Data

Dissertation im Fach Informatik
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt von

Katrin Haegler



Erstgutachter: Prof. Dr. Christian Böhm
Zweitgutachter: Prof. Dr. Martin Wiesmann

Tag der Abgabe: 23. Mai 2011
Tag der mündlichen Prüfung: 08. November 2011

München, den 23. Mai 2011

Contents

Abstract

The ongoing automation in our modern information society leads to a tremendous rise in the amount as well as complexity of collected data. In medical imaging for example the electronic availability of extensive data collected as part of clinical trials provides a remarkable potentiality to detect new relevant features in complex diseases like brain tumors. Using data mining applications for the analysis of the data raises several problems. One problem is the localization of outstanding observations also called outliers in a data set. In this work a technique for parameter-free outlier detection, which is based on data compression and a general data model which combines the Generalized Normal Distribution (GND) with independent components, to cope with existing problems like parameter settings or implicit data distribution assumptions, is proposed.

Another problem in many modern applications amongst others in medical imaging is the efficient similarity search in uncertain data. At present, an adequate therapy planning of newly detected brain tumors assumedly of glial origin needs invasive biopsy due to the fact that prognosis and treatment, both vary strongly for benign, low-grade, and high-grade tumors. To date differentiation of tumor grades is mainly based on the expertise of neuroradiologists examining contrast-enhanced Magnetic Resonance Images (MRI). To assist neuroradiologist experts during the differentiation between tumors of different malignancy we proposed a novel, efficient similarity search technique for uncertain data. The feature vector of an object is thereby not exactly known but is rather defined by a Probability Density Function (PDF) like a Gaussian Mixture Model (GMM). Previous work is limited to axis-parallel Gaussian distributions, hence, correlations between different features are not considered in these similarity searches. In this work a novel, efficient similarity search technique for general GMMs without independence assumption is presented. The actual components of a GMM are approximated in a conservative but tight way. The conservativity of the approach leads to a

filter-refinement architecture, which guarantees no false dismissals and the tightness of the approximations causes good filter selectivity. An extensive experimental evaluation of the approach demonstrates a considerable speed-up of similarity queries on general GMMs. Additionally, promising results for advancing the differentiation between brain tumors of different grades could be obtained by applying the approach to four-dimensional Magnetic Resonance Images of glioma patients.

German Abstract

Die voranschreitende Automatisierung in unserer modernen Informationsgesellschaft führt zu einem gewaltigen Anstieg von Quantität und Komplexität der erfassten Daten. Im Bereich medizinischer Bildgebungsverfahren stellt die elektronische Verfügbarkeit von komplexen und umfangreichen Daten, welche als Teil klinischer Studien erhoben werden, ein außergewöhnliches Potential dar, um wichtige Eigenschaften in komplexen Krankheiten wie z. B. Hirntumoren zu entdecken. Die Verwendung von Datamining-Verfahren zur Analyse der erhobenen Daten führt zur Entstehung verschiedener Probleme. Eine Fragestellung die es zu lösen gilt, ist die Lokalisierung von außerordentlichen bzw. bedeutenden Beobachtungen in Datensätzen, auch bekannt als Ausreißer. Hierfür wird in dieser Arbeit ein technisches Verfahren zur parameterfreien Identifikation von Ausreißern vorgestellt. Dieses Verfahren basiert auf Datenkomprimierung und einem sehr generellen Datenmodell welches die generalisierte Normalverteilung (GND) mit Hilfe der Unabhängigkeitsanalyse (ICA) auf unabhängige Komponenten anwendet.

Ein weiteres Problem welches in vielen modernen Anwendungsbereichen unter anderem in medizinischer Bildgebung auftritt, ist die effiziente Ähnlichkeitssuche in unsicheren Daten. Zur Zeit wird für eine vollwertige Therapieplanung von neu entdeckten Hirntumoren, welche vermutlich glialen Ursprungs sind, eine invasive Biopsie benötigt, da sich sowohl die Prognose als auch die Behandlung von gutartigen, niedrig-gradigen und hoch-gradigen Hirntumoren sehr stark unterscheiden. Desweiteren werden unterschiedliche Hirntumoren hauptsächlich aufgrund der Expertise von Neuroradiologen, welche magnetresonanztomographische, kontrastmittel-verstärkte Bilder untersuchen, differenziert. Zur Unterstützung der Neuroradiologen bei der Unterscheidung von unterschiedlichen Tumorgraden wurde hier ein neues, effizientes Verfahren zur Ähnlichkeitssuche in unsicheren Daten entwickelt. Da der Eigenschaftsvektor eines unsicheren Objektes hierbei nicht exakt bekannt ist, wird er statt dessen als Funktion einer Wahrscheinlichkeitsver-

teilung, in unserem Fall eines Gemischten Gauß Modells (GMM), definiert. Bestehende Verfahren werden durch die Verwendung von achsen-parallelen Gaußverteilungen beschränkt, da sie Korrelationen zwischen verschiedenen Eigenschaften bei der Ähnlichkeitssuche nicht berücksichtigen. In dieser Arbeit wird eine neue, effiziente Technik zur Ähnlichkeitssuche auf generellen Gemischten Gauß Modellen ohne eine Unabhängigkeitsannahme präsentiert. Die eigentlichen Komponenten des GMM werden hierbei in einer konservativen, aber dennoch undurchlässigen Art und Weise approximiert. Die Konservativität des Verfahrens führt zu einer Filter-Verfeinerungsarchitektur, welche dafür garantiert, dass es keine falsch positiven Ergebnisse gibt und die Undurchlässigkeit der Abschätzungen bedingt eine gute Filterselektivität. Eine ausführliche experimentelle Auswertung des Verfahrens zeigt eine beträchtliche Beschleunigung einer Ähnlichkeitsanfrage auf generellen GMM im Vergleich zu einer erschöpfenden Suche. Zusätzlich konnten während der Anwendung des Verfahrens auf vier dimensionale Magnetresonanztomographie Bilder von Gliompatienten vielversprechende Ergebnisse bei der Differenzierung zwischen verschiedenen Hirntumoren unterschiedlichen Grades erzielt werden.

Acknowledgements

This work would not have been possible without the support of various people. I would like to take the opportunity now to thank all of them for their contributions to this thesis.

First of all, I want to thank my doctoral supervisor and first referee, Prof. Dr. Christian Böhm for giving me the great opportunity as an external PhD student to be part of his team. He provided me an insight into ongoing fascinating research and supported this work with his great experience. I am also very thankful to my supervisor and second referee Prof. Dr. Martin Wiesmann who provided great advice from the field of medicine. From the distance he was able to create a great working atmosphere and gave me the opportunity to focus on those aspects that seemed most promising to me.

I owe special thanks to Dr. Jennifer Linn for looking through the medical data with me very patiently and for helping me to extract the most out of the patient data. I would like to also thank Dr. Claudia Plant for her helpful discussions to enhance my work.

Furthermore, I want to express a special thank to Nikola Müller. It was my pleasure to work with you on several projects. I learned a lot from you. I also owe much to my colleagues from the database group in particular Frank Fiedler, Bettina Konte, Bianca Wackersreuther, Annahita Oswald, Peter Wackersreuther, Michael Plavinski, and Andrew Zherdin. I enjoyed the wonderful and interesting discussions with all of you.

A hug and thanks go to all the members of the olfactoric team at the

Department of Neuroradiology. You introduced me to the field of functional imaging and olfactoric. In particular I want to thank Dr. Jessica Albrecht for the guidance and help no matter which day time or continent. Without the coffee sessions with my team the time wouldn't have gone by so fast and I would have definitely had less fun. Thank you Dr. Rebekka Zernecke, Dr. Anna Maria Kleemann, Dr. Rainer Kopietz, and Dr. Veronika Schöpf for so many nice hours.

Finally, I want to express my deepest thanks to my family and friends for comforting me when I needed them. Thank you all for your support and encouragement during the time that I was engaged in this study. The most important thanks go to the most important person in my life. Andi, thank you for all your patience especially in the last months, for always comforting and motivating me. Without you I wouldn't have been able to finish this work. You were there when I needed you the most and you believed in me when I stopped believing in myself.

Katrin Haegler

Bruckmühl, 23. Mai 2011

Chapter 1

Introduction

The proceeding development in medical imaging techniques has accounted to a large amount of high-resolution three-dimensional image data. Especially the high volume of non-invasive measures acquired in clinical routine like structural and functional Magnetic Resonance Imaging (MRI) have revealed new possibilities in getting to know more about the functioning of the human brain. To reveal new relations between imaging and features like diagnosis, prognosis, response to treatment, or genetic profiling elaborated algorithms are required. For the field of brain imaging data mining techniques have proven to be very useful. Even if data mining can be used to discover knowledge from medical data some requirements have to be satisfied so that they can be applied. All applied techniques are application dependent, hence, the utilization of different data mining techniques heavily depends on the different applications, and in order to be processed the data has to have a specific size and format.

In mining medical images several difficulties have to be faced. One important problem is that the majority of data mining approaches like decision trees, nearest neighbor search, or rule-based learning systems need data consisting of certain data objects meaning simple numeric values. Since brain images are multidimensional data arrays the conversion to single numeric val-

ues leads to a large amount of information loss. Hence, it is desirable to find a data structure which approximates the complex image data as accurate as possible. Another problem is that most classification algorithms do not use the raw medical image data but rather the medical record information due to the lack of efficient and effective approaches which learn from the raw images directly. Medical images are complex and large data structures, one patient can be represented by millions of intensity values with multiple dimensions, therefore, the comparison of single intensity values between different patients in order to find patient pairs sharing similarities is impractical and inefficient. In clinical routine medical doctors can not wait hours for data mining algorithms to be finished with their complex calculations. Therefore, algorithms are required which find a compromise between using as much information as possible from the underlying image data but still being computationally efficient due to the clinical practicability. An additional problem which has to be faced working with medical image data is the large heterogeneity of brain image data like the diverse image acquisition modalities, the different formats produced, and the different resolutions which lead to problems in general analysis. The acquired data in the field of brain imaging usually consists of images from various modalities like Magnetic Resonance Imaging, Computer Tomography (CT), or Positron Emission Tomography (PET) which comprise structural and/or functional information of the human brain. In order to be able to include these different data formats produced by diverse modalities accurate preprocessing of the data is of essential importance.

One fundamental step in image preprocessing is the detection of outlying observations. These recorded voxels can either be considered as noise being removed from the data before the actual data mining process starts, or they may incorporate important, outstanding information. The detection and removal of outliers may avoid model misspecification, biased parameter estimation, and hence incorrect results. The identification of outliers prior to modeling and analysis is thus very important. Most existing outlier detection approaches assume Gaussian or Uniform data distributions but for

many real world applications amongst others medical data this cannot be applied. Another problem which has to be addressed when dealing with outlier detection algorithms is the correct setting of parameters. Parameters are often difficult to be distinguished and in order to correctly set these parameters, background knowledge of the underlying data is required. Here we propose a new technique for the identification of outliers. Our approach does not require the selection of any parameter, thus it is a parameter-free outlier identification technique. This can be achieved on account of data compression which we use to distinguish outliers since they cannot be compressed as efficient as the rest of the data. To avoid the assumption of a specific data structure we used a Generalized Normal Distribution (GND) in combination with Independent Components. In addition to the removal of the data several other preprocessing steps have to be executed to address the problem of medical image data heterogeneity including motion correction, standardization, and co-registration in order to obtain image data which has the same format and resolution. Having readily preprocessed three dimensional brain data various data mining techniques can be applied.

The classification of brain tumors using features from structural as well as functional brain images in order to improve diagnosis and therapeutic response has become a considerable research area. The correct assignment of tumor malignancy is important due to the different prognosis and therapy planning of brain lesions with different grading. In most cases a biopsy is necessary to amplify the validity of the diagnosis. However biopsy carries risks, since a single tumor mass can be histologically heterogeneous; extracting parts of the tumor that are not representative (sampling error) would lead to an incorrect diagnosis and an inadequate treatment. Furthermore, biopsy implies risks associated with anesthesia and surgery. Hence, being able to diagnose tumor grades using non-invasive structural and functional Magnetic Resonance Imaging techniques instead of a biopsy would give a tremendous benefit to the patient. To address the problems of finding an accurate data structure loosing as little information as possible while keeping

a high computational efficiency we propose a novel similarity search based classification algorithm. Since one patient is represented by a tremendous amount of voxels we decided to consider each patient as an uncertain object being represented by a Probability Density Function (PDF). In our approach we utilize Gaussian Mixture Models (GMM) as PDF representatives due to the fact that mixture models can capture the different subparts of the images as separate components. Hence, the entire data distribution is considered in our analysis. Furthermore, to accelerate the similarity search in a first step we approximate the Gaussian components to minimize the candidate set for which the time consuming joint probabilities between the query object and the database objects have to be computed. Therefore, our approach is conservative even considering attribute correlations it is still efficient.

1.1 Outline Of The Thesis

The thesis at hand is organized as follows:

Chapter 1 provides an introduction to the general context of this thesis.

Chapter 2 describes the theoretical background information of existing approaches concerning the field of data mining.

Section 2.1 outlines a selection of existing information theoretic measures which can be used to facilitate the differentiation of several given models.

Section 2.2 gives a general overview of uncertain data in the field of data mining.

Section 2.3 provides a brief overview of existing clustering approaches divided into the three major categories. To overcome parametrization a short summary of clustering in combination with information theoretic principles

is also introduced.

Section 2.4 gives an overview of outlier detection introducing the different concepts to identify outstanding objects.

Section 2.5 introduces the field of classification for knowledge discovery in databases. Thereby, different strategies for assigning an unknown object to one of several known classes are presented.

Chapter 3 deals with the identification of outstanding objects in data sets. A new method is proposed which is able to detect outliers without any explicit parameter settings and without requiring the data to be of any specific distribution type.

Section 3.1 gives an introduction to the field of outlier detection and a motivation to our newly proposed outlier approach.

Section 3.2 reviews existing outlier detection approaches and introduces the Minimum Description Length principle in the context of existing data mining methods.

Section 3.3 presents a novel parameter-free outlier detection algorithm which uses ICA to convert data into independent components to reduce redundancy in the data before the broadly applicable General Normal Distribution is applied to the data. The returned outlier score is based on coding costs generated by the Minimum Description Length principle.

Section 3.4 comprehends several different data scenarios of synthetic and real world data to demonstrate the superiority of our proposed outlier detection algorithm.

Section 3.5 concludes the proposed outlier detection approach.

Chapter 4 treats a new efficient similarity search technique being able to handle uncertain data represented by general Gaussian Mixture Models. The method comprises a variety of data mining techniques, including a clustering algorithm adjusted to circular measures, an approximation technique leading to a large runtime reduction, and a classification scheme for finding similar uncertain query objects in a database of uncertain objects. Combining these techniques results in our conservative but tight filter-refinement architecture considering also attribute correlations in the similarity search.

Section 4.1 introduces uncertain objects and the relevance of having an exact representation of these objects for obtaining more accurate similarity search results.

Section 4.2 motivates the need of including correlations in the representation of uncertain objects.

Section 4.3 reviews previous classification approaches on certain and uncertain data including acceleration techniques for handling query processing more efficiently.

Section 4.4 describes our novel similarity search for uncertain data. After the preprocessing of a given database of nGMMs, meaning rotation and approximation of all Gaussian components of the nGMMs, the k objects having the highest probability of being drawn from the same distribution as the query object are returned by the similarity search. Thereby, also feature correlations are considered in the similarity search.

Section 4.5 comprises a variety of different synthetic as well as real world data sets to elaborate the strength of our new similarity search and emphasizes the importance of considering correlations for query processing.

Section 4.6 concludes the novel similarity search technique on uncertain data.

Chapter 5 draws the connection between medical imaging, outlier detection, and similarity search. The complete workflow from image acquisition to query processing of glioma data is described. The algorithm is evaluated using glioma data of low- and high-grade glioma patients leading to a considerable increase in accuracy compared to existing glioma grading methods.

Section 5.1 specifies the need for a new, accurate glioma grading method which is able to integrate the entire tumor ROI information in the grading method.

Section 5.2 delineates our new glioma grading similarity search which is based on non-axis parallel GMMs. Thereby, each nGMM in the data corresponds to one four-dimensional glioma of a brain tumor patient. We present the entire workflow including data preprocessing and query processing and introduce three additional methods which are subsequently used for method comparison.

Section 5.3 comprises glioma grading results on a set of biopsy confirmed glioma patients for our similarity search based glioma grading method as well as for three additional methods highlighting the strength of our method.

Section 5.4 discusses the different aspects of our new glioma grading technique including the advantages and the drawbacks.

Section 5.5 concludes the glioma grading approach.

Chapter 6 concludes this thesis.

Section 6.1 summarizes and discusses the major contributions of this thesis.

Chapter 2

Algorithmic Fundamentals

2.1 Information Theoretic Measures

If for a given model several explaining variables are plausible and the data set is rather small it can be difficult to find a qualified model. For this purpose different model selection approaches have been proposed to assist in finding a good model.

Comparing two alternative models the risk of having a bias due to not considering relevant variables and the risk of using too many irrelevant variables has to be balanced. For this purpose information criterions can be used. An information criterion is a criterion for selecting a model using the goodness of fit of an estimated model to the underlying empirical data. The complexity of the measured model determined by the number of parameters is also included in the decision. The number of parameters is punished to avoid the preference of models with many parameters. All information criteria have in common that they can be formalized in two different ways. The measure for the goodness of fit is either defined by the maximal likelihood or the minimal variance of the residuals. Hence, different interpretation possibilities can be found. For the first criterion using maximal likelihood, the best fitting model

is the model having the highest value, therefore the number of parameters has to be subtracted. Whereas, using the minimal variance the best model is the model with the lowest value, hence the number of parameters has to be added. In the following we will explain different information criteria using solely the maximum likelihood as measure for the goodness of fit.

2.1.1 Akaike Information Criterion

The oldest information criterion is called Akaike Information Criterion (AIC) [Aka74]. It is based on the principle of information entropy, indeed using a given model for characterizing reality AIC provides a relative quantity value of the lost information. AIC does not test a model hypothesis but it offers a possibility to compare different models, hence it can be used for model selection.

Let $L_n(k)$ be the maximum likelihood of a model taking k parameters based on a sample of size n. The likelihood function for $k \leq m$ is denoted by $\hat{L}_n(\theta), \theta \in \Theta \subset \mathbb{R}^m$,

$$L_n(k) = \max_{\theta \in \Theta} \hat{L}_n(\theta), \text{ with } \Theta_k = \left\{ \theta = \binom{\theta_1}{\theta_2} \in \Theta : \theta_2 = 0 \in \mathbb{R}^{m-k} \right\} \tag{2.1}$$

Hence, the Akaike Information Criterion for selecting the best model is

$$AIC = -\frac{2\ln(L_n(k))}{n} + \frac{2k}{n} \tag{2.2}$$

with k being the number of independent adjusted parameters of the model.

Multiple models can thus be ranked by their AIC using the same data set and the model having the minimum AIC can then be selected as the model of choice. The disadvantages of AIC are that the penalty term depends on the size of the sample and that large samples have a better chance to enhance the log likelihood function, hence AIC prefers models with a lot of parameters.

2.1.2 Bayesian Information Criterion

To overcome AIC's disadvantages Schwarz proposed the Bayesian Information Criterion (BIC) [Sch78] also called Schwarz criterion which is closely related to AIC. Here, the penalty factor grows logarithmic with the number of observations n, leading to the following criterion,

$$BIC = -\frac{2\ln(L_n(k))}{n} + \frac{k\ln(n)}{n}. \tag{2.3}$$

Some applications of BIC are model identification in time series and linear regression. But in general it can be applied to a wide range of applications given maximum likelihood-based models. However, in several applications the number of parameters is equal for the different tested models, hence BIC only reduces the maximum likelihood selection.

2.1.3 Minimum Description Length

The Minimum Description Length (MDL) [Ris78] was introduced to describe regularities in the observed data. It is a formalization of Occam's Razor and it is an important concept in information theory. MDL uses the fact that the stronger the data can be compressed the larger is the regularity in the signal. The underlying principle can be described as follows: A sender S wants to transfer a message to a receiver R. Thereby, the receiver knows the characters that could be contained in the message but he does not know the frequency of each character in the message. In order to send the message first the description of the coding schemata is sent followed by the message that was coded by the coding schemata. The sum of the characters which have to be sent can therefore be lower than sending the entire message.

An intuitive example of the MDL principle is shown in Figure 2.1. Suppose we want to transfer data through a transfer channel. The sender wants to transfer the string $X^A Y Z^B$ to the receiver. A naive way would be to

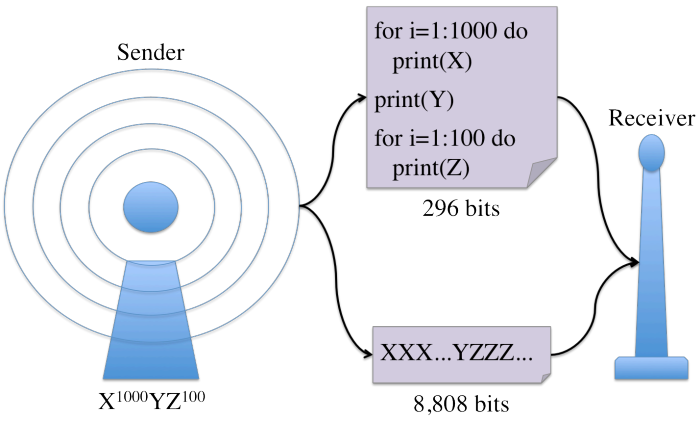


Figure 2.1: An intuitive example of the Minimum Description Length principle. Sending the string message $X^{1000}YZ^{100}$ from a sender to a receiver as simple string would require 8,808 bits. Transferring the same string coded by the coding schemata shown on top only 296 bits are required, therefore, a total of 8,512 bits can be saved.

transfer each single character requiring in total 8,808 bits for $A = 1,000$, $B = 100$ and 8 bits per character. To minimize the communication costs, a smart sender exploits regularities in the data. A little program could generate the string by printing 1,000 times the character X, followed by printing Y finishing with printing 100 times Z. An efficient coding in an arbitrary language requires thus, 296 bits to transfer the string in total. This clever compression reduces the communication cost to 3.36% using MDL.

2.2 Uncertain Data Mining

Most existing databases can only handle exact value representations. But as the amount of data rises also the complexity of the data increases. Many real world applications produce data which imply uncertainty. Data uncertainty in real world applications arises due to measurement inaccuracy, sampling discrepancy, outdated data sources, or other errors. This is especially true for applications that require interaction with the physical world. Hence,

handling of data uncertainty has gained a lot of research interest in the last decade. Thereby, uncertain data can be subdivided into two subtypes, existential uncertainty and value uncertainty. Existential uncertainty implies that it is not certain if the object is existent or not. This can be accounted for by assigning an existence probability value to the object, indicating the confidence of its existence [BGMP92]. An example of existential uncertainty could be a mobil phone tracing system. If a mobil user shuts off his phone or passes through a tunnel the signal is lost hence the data is only partially available. In contrast to that in value uncertainty data instances consist of values with margins of error which can be modeled as areas in combination with a Probability Density Function [CKP04, CKP03]. This is for example the case in face recognition, a person is represented by multiple images taken with different camera angles. Hence, the face specific features, e.g. eye distance or the distance between the nose tip and the chin, which are extracted in order to find the person in a database can include large variability.

In many databases uncertain objects are converted into exact values by assigning a weight to the average or extracting the values of highest occurrence to reduce the complexity and to ease the handling of the data. But the approximation of uncertain data can lead to wrong results in data mining, e.g. due to approximated data, the centroids in clustering tasks are also approximations and hence the deviation of cluster centroids can lead to wrong object assignments.

In order to find an accurate distance measure to handle uncertain objects different strategies can be applied. In general, a set of d dimensional objects o_i each associated with a Probability Density Function, like a Uniform or a Gaussian distribution is given. A naive approach for handling this uncertainty would be to use the least expected distance between two uncertain objects p and q [CCK05]. But the expected distance is a rather unreliable distance indicator. Figure 2.2 shows an example of uncertain objects p, q, and r. Objects p and q are intuitively the closest objects since q_1 has a probability of 0.99 and p_2 has a probability of 0.8 being very close to each other.

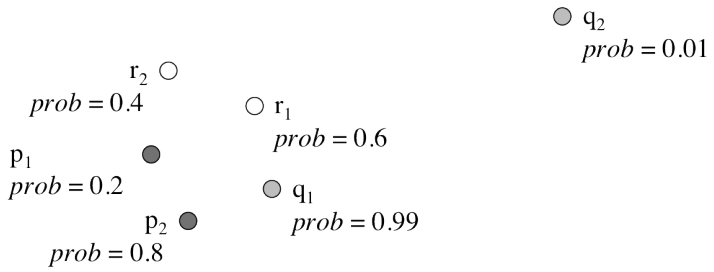


Figure 2.2: An example of three uncertain objects p, q, and r demonstrating that the expected distance is an unreliable distance measure for uncertain objects. Even if objects p and q seem to be most similar since q_1 and p_2 which have very high probabilities are close to each other the expected distance is large due to the distant q_2.

But the expected distance between p and q is very large since q_2 is far away from p.

A more reliable approach would be to report the matching probability between two uncertain objects p and q [BSI08, BPS06]

$$P(p,q) = \int_{\mathbb{R}^d} pdf_p(x) \cdot pdf_q(x)dx \tag{2.4}$$

where $pdf(x)$ denotes the Probability Density Function of p and q. The matching probability represents the relative probability with which a sample x, randomly drawn from the distribution of p equals another sample x randomly drawn from q. Finding an accurate distance measure for uncertain objects is very important because of the various queries and data mining tasks, like Nearest Neighbor queries or clustering which will be introduced in the following sections.

2.3 Clustering

The aim of clustering is the identification of a finite number of categories, classes, or groups in a data set. While objects in the same cluster should be as similar as possible, objects in different clusters should be as dissimilar as possible. Typical application areas for clustering are customer segmentation, clustering of web-logs, structuring of large sets of text documents, or the generation of topical maps from satellite images. Clustering can be subdivided into three fundamental clustering paradigms:

- Partitioning clustering - searches for flat clusterings in k clusters with minimal costs. Parameters are the number of clusters k and the distance function.
- Hierarchical clustering - distinguishes the hierarchy of clusters by combining the most similar clusters. Parameters are the distance function for objects and clusters.
- Density-based clustering - expands clusters by neighboring objects until the density is large enough. Parameters are the minimal density of a cluster and the density function.

In the following these clustering approaches will be explained in more detail, followed by a brief introduction of information theoretic measures which can be used to overcome the problem of parametrization and to evaluate the quality of different clustering approaches.

2.3.1 Partitioning Clustering

The goal of partitioning clustering approaches is to partition a data set into k clusters by minimizing a given cost function. After k initial cluster representatives have been chosen, these representatives are iteratively improved until

they are optimal, followed by assigning each object to its closest, or most similar representative. Thereby, different types of cluster representatives are feasible, e.g. the mean of a cluster (k-Means), a cluster element (k-Medoid), or a Probability Density Function of the cluster (Expectation Maximization).

Objects in partitioning approaches are defined as points $x = (x_1, ..., x_d)$ in an Euclidean vector space, having a centroid μ of all points contained in a cluster C. Cluster costs are measured by

$$cost(C) = \sum_{x \in C} dist(x, \mu)^2 \tag{2.5}$$

and costs for a clustering are measured by

$$cost = \sum_{i=1}^{k} cost(C_i). \tag{2.6}$$

The algorithm starts with a random initialization step, where k objects are randomly chosen as cluster representatives. Subsequently, each data object is assigned to the representative which is closest in space and new representatives, which are the centroids of the cluster objects, are calculated. These steps are repeated until convergence, meaning no objects change their cluster membership anymore.

Existing approaches can be differentiated by their used distance function. In the following, we will address the three most common partitioning clustering approaches k-Means, k-Medoid, and Expectation Maximization (EM).

K-Means The K-Means [DH73] algorithm has the characteristics of the basic partitioning algorithm. Before starting the algorithm the number of clusters k has to be selected. As distance measure k-Means uses measures like the Euclidean norm or the Mahalonobis distance and the cluster representative is located in the center of the cluster.

The algorithm is one of the most frequently used techniques for object clustering. It depends on the order of the objects because of the direct update of the centroids. As soon as an object changes its cluster membership, the affected centroids are updated. The average time complexity of k-Means is $O(n)$ for each iteration and the number of iterations is rather small (average 5-10). A disadvantage of k-Means is its sensitivity to noise and outliers which is caused by the fact that all objects are considered for the calculation of the centroids. Additionally, the number of clusters k is hard to be assigned and k-Means is strongly affected by the initialization step. Therefore, it is useful to try several random initializations and retain the best result.

K-Medoid The K-Medoid algorithm uses a medoid m, instead of the mean, as central element of a cluster. A medoid is a representative object of a cluster whose average dissimilarity to all the objects in the cluster is minimal. Therefore, the cluster costs are calculated by $cost(C) = \sum_{x \in C} dist(x, m)^2$ and the clustering costs by $cost = \sum_{i=1}^{k} cost(C_i)$.

An example of a greedy k-Medoid algorithm is PAM (Partitioning Around Medoids) [KR90] which replaces in each step one medoid with an object not being a medoid (non-medoid), thereby using the medoid - non-medoid pair causing the largest cost reduction. CLARANS [NH94] (Clustering Large Applications based on Randomized Search), another k-Medoid approach has two additional parameters (maxneighbor, numlocal) leading to a time complexity reduction in contrast to the greedy PAM algorithm. At most maxneighbor medoid - non-medoid pairs are considered and the first replacement which leads to a cost reduction is applied. Numlocal is the number of times the search for the k optimal medoids is repeated.

Expectation Maximization The EM algorithm [DLR77] uses, in contrast to k-Means and k-Medoid, a Probability Density Function as cluster representative. The EM algorithm can also be split into two steps, the Maximization step (M-step) which maximizes the log-likelihood of the data and

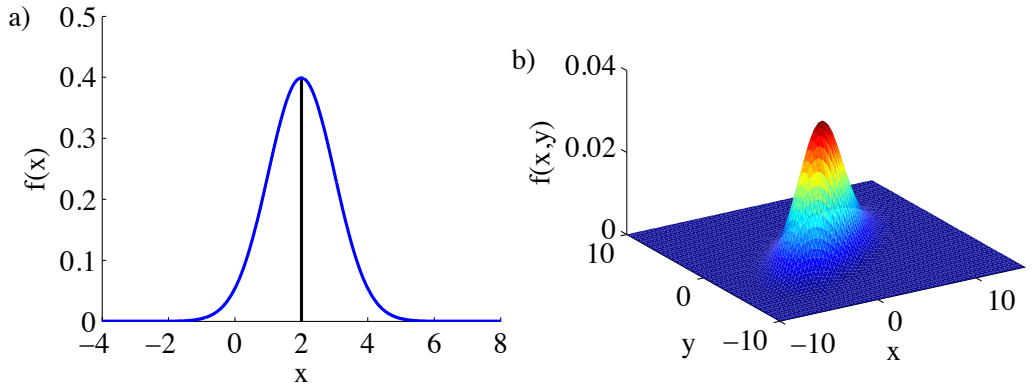


Figure 2.3: Normal distributions. A a) univariate and a b) bivariate normal distribution are depicted.

the Expectation step (E-step) which assigns the objects proportionally to their likelihood to the cluster representatives. A model for a cluster is e.g. a multivariate normal distribution (Figure 2.3). A cluster C is represented by a mean μ and a $d \times d$ covariance matrix Σ of all objects contained in the cluster. The basic idea is that each object belongs to each cluster with a certain probability, which is dependent on $P(x|C_i)$. Like all partitioning algorithms the EM algorithm also consists of two alternating steps, the assignment of the points to the clusters (relative probabilities) and the recalculation of the cluster representative (Gaussian distribution). For obtaining the cluster centers μ_i one has to consider that objects are not assigned with an absolute but rather a relative probability to the clusters.

Each cluster C_i is modeled by a Probability Density Function (Figure 2.3) like

$$P(x|C_i) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_{C_i}|}} \cdot exp^{-\frac{1}{2}(x-\mu_{C_i})^T \Sigma_{C_i}^{-1}(x-\mu_{C_i})}. \tag{2.7}$$

The integral of the density function is unity and the integral of a region R of the density function produces the probability that an arbitrary object of the cluster is located in the region or rather the relative portion (e.g. 20 %)

of objects of the cluster that are located in R.

K clusters are represented by k Gaussian distributions leading to a probability density

$$P(x) = \sum_{i=1}^{k} w_i P(x|C_i), \tag{2.8}$$

where w_i is the relative portion of data objects assigned to C_i.

Bayes' theorem can then be used to obtain the probability of an object x to belong to a cluster C_i. Let $A, B \subseteq \Omega$ and $P(A|B)$ be the conditional probability of A given B then the conditional probability is defined as

$$P(A|B) = \begin{cases} 0, & \text{if } P(B) = 0 \\ \frac{P(A \cap B)}{P(B)}, & \text{else.} \end{cases} \tag{2.9}$$

Furthermore, A and B are independent if $P(A|B) = P(A)$ and $P(B|A) = P(B)$

The theorem of Bayes states that if $A_1, ..., A_k$ is a disjunct partitioning of Ω, so that for at least one i, $1 \leq i \leq k$, it can be applied that $P(A_i) > 0$ and $P(B|A_i) > 0$, than $\forall 1 \leq i \leq k$ it can be applied that

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)}. \tag{2.10}$$

Using the theorem of Bayes we can now obtain the probability with which a given object x belongs to cluster C_i

$$P(C_i|x) = w_i \frac{P(x|C_i)}{P(x)}. \tag{2.11}$$

As k-Means and k-Medoid, EM depends strongly on the initialization and the right choice of the parameter k. The approach finds only local optima, hence it is often necessary to run the EM multiple times choosing the best outcome as final result.

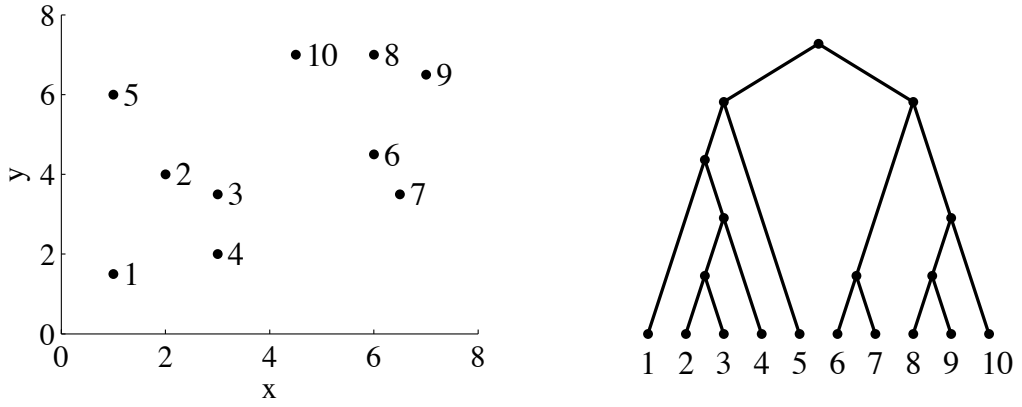


Figure 2.4: An example of a dendrogram which was build using single linkage hierarchical clustering.

2.3.2 Hierarchical Clustering

The goal of hierarchical clustering is the construction of a hierarchy of clusters, called a dendrogram, in which always those clusters having minimal distance are merged. A dendrogram, as depicted in Figure 2.4, is a tree like structure. The root represents the entire data set, the knots of the tree represent clusters, where the inner knots are clusters consisting of objects which are located in the subjacent subtree, and the leaves are the single objects.

There are two different types of hierarchical approaches, the bottom-up construction of the dendrogram (agglomerative) and the top-down construction (divisive). The agglomerative version of the algorithm works as follows:

1. Build singleton clusters, which consist of one object each and determine the distances between all pairs of clusters.

2. Build a new cluster consisting of the two clusters having the smallest distance to each other.

3. Determine the distances between the new cluster and all other clusters.

4. If all objects are located in one single cluster stop, otherwise repeat steps 2 through 4.

Additionally, there are three different hierarchical principles differing in the distance function $dist(x, y)$: single linkage, average linkage, and complete linkage. These principles will be explained in the following in more detail.

Single Linkage The distance function of single linkage can be defined as follows: let X and Y be clusters, that means sets of objects than the distance between those objects is defined by

$$dist_{sl}(X, Y) = \min_{x \in X, y \in Y} dist(x, y). \tag{2.12}$$

Single linkage approaches have a time complexity of $O(n^2)$, like e.g. SLINK [Sib73]. Single linkage results in chain-like clusters, with a large dispersion and an elongated structure. A single linkage variant called CURE (Clustering Using Representatives) [GRS98] is able to adapt the shape of the cluster without a single-link effect by using a larger number of objects instead of only one single reference object as cluster representative.

Average Linkage For average linkage the distance between two clusters X and Y is defined by

$$dist_{al}(X, Y) = \frac{\sum_{x \in X, y \in Y} dist(x, y)}{|X| \cdot |Y|}. \tag{2.13}$$

The time complexity for average linkage is $O(n^2 log(n))$ and it is a compromise between single and complete linkage.

Complete Linkage Complete linkage calculates the distance between two clusters X and Y by using

$$dist_{cl}(X,Y) = \max_{x \in X, y \in Y} dist(x,y). \tag{2.14}$$

Like single linkage the time complexity for complete linkage is $O(n^2)$. An example of a complete linkage algorithm is CLINK [Def77]. Complete linkage leads to small, strongly separated clusters which are convex and equal in size.

2.3.3 Density-based Clustering

Clusters in density-based clustering approaches are areas in a d dimensional space with a high object density. These dense cluster regions are separated by areas with lower object density. Additionally, density-based clusters have to fulfill certain criteria like the local object density of each object contained in a cluster has to exceed a given threshold and all objects in a cluster have to be spatially connected.

An algorithm for density-based clustering is DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [EKSX96]. DBSCAN requires the setting of two parameter, ϵ and $MinPts$ to specify the density threshold of a cluster. An object p is called core object if their are at least $MinPts$-1 (because the core object is also counted) objects for which the distance between the core object p and an object q is smaller or equal to ϵ. There are three fundamental terms for density-based clustering which are depicted in Figure 2.5. An object p is directly density reachable from an object q if the distance between p and q is smaller or equal to ϵ (indicated by the circles around p and q). An object p is density reachable from an object q if there is a chain of objects between p and q that are direct density reachable. Two objects p and q are density-connected if they are both density reachable from a third object r. In order to find a cluster DBSCAN starts with an arbitrary object collecting all objects that are density-connected with this

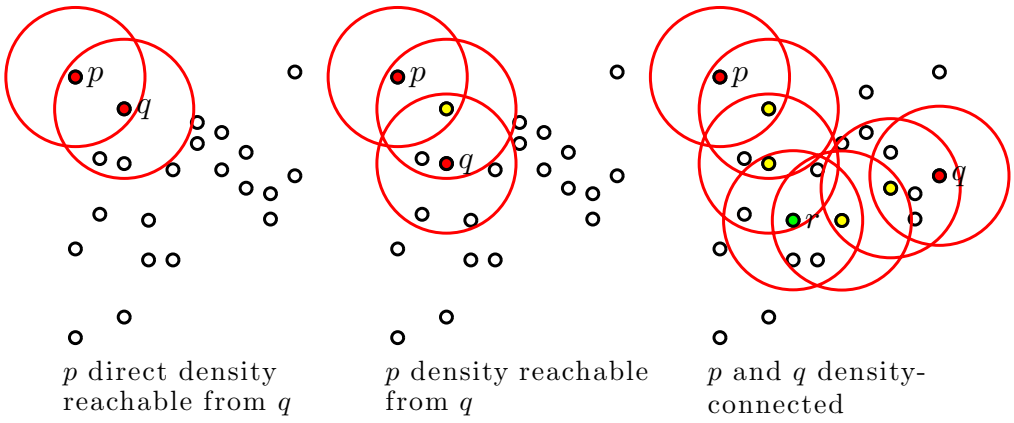


Figure 2.5: Fundamental terms of density-based clustering.

object. That means a density-based cluster is defined by the maximal set of density-connected objects. All objects that are not part of a density-based cluster are considered noise objects.

2.3.4 Parameter-free Clustering

A problem most presented approaches have to overcome is the correct setting of required parameters. In most cases these parameters have to be set by domain experts. To solve this problem some algorithms have been developed coping the parametrization problem by using information theoretic measures.

X-Means Based on k-Means clustering the X-Means [PM00] algorithm tries to solve the parametrization problem of k-Means. For this purpose X-Means does not take one single value k for the number of clusters to be found but rather an interval in which the optimal number of classes is probably located. The X-Means algorithm can be subdivided into two steps, the improve parameter step and the improve structure step. The first step is basically an ordinary k-Means run and the second step determines which centroids are being split to improve the clustering result. Starting from the

lower threshold of the given interval new centroids are added by splitting the parent centroid into two child centroids. To determine if the splitting pays off BIC is determined for the parent cluster as well as for the children clusters on the basis of the contained cluster objects. The centroid getting the more beneficial BIC value is kept the other one is discarded. The clustering having the optimal overall BIC value is returned as the final clustering result.

Robust Information-theoretic Clustering In contrast to X-Means the algorithm RIC (Robust Information-theoretic Clustering) [BFPP06] was not designed for a specific clustering algorithm. It is rather an extension of any arbitrary clustering approach. In particular, RIC is a post-processing step which takes a given clustering of any traditional clustering method as input. Based on the initial clustering, RIC starts by filtering possible noise points from the data followed by cluster modeling. Thereby, given a set of PDFs each cluster model is represented by a rotation matrix from the Principle Component Analysis and a PDF. Clusters possessing similar characteristics are combined and due to the MDL principle RIC is parameter-free. The accomplishment of RIC is that it has the ability to enhance an imperfect clustering which has been generated using suboptimal parameterization.

Outlier-robust Clustering using Independent Components [BFP08] OCI is a parameter-free clustering approach using a very general PDF as cluster description, namely the Exponential Power Distribution (EPD). This distribution function implies several different PDFs like the Gaussian, the Uniform, and the Laplace distribution. In a top-down splitting approach OCI combines the EPD with the Independent Component Analysis in order to obtain major orientations in the feature space for each cluster. Additionally, in applying EPD to independent components OCI can obtain adequate planes which can be used for cluster splitting. Due to the MDL principle OCI is a parameter-free approach and due to the general cluster definition it is able to identify clusters of different shape and density not being restricted

to solely Gaussian or Uniform distributions.

2.3.5 Quality Measures

The comparison of different clusterings is a difficult task. Therefore, numerous measures for comparing clusterings have been proposed like pair-counting based, set-matching based, and information theoretic measures. As stated by Vinh *et al.* [VEB10] an adequate clustering comparison should fulfill the following properties: it should be a metric, it should be normalized, and it should have a constant baseline. The metric property requires that a distance measure is positive definite, symmetric and that it has a triangle inequality. The normalization property states that the range of a similarity or distance measure is located within a fixed range like [0,1] and the constant baseline property states that the expected value between independent clustering pairs should be constant. In the ideal case the baseline value should be zero which indicates no similarity between the clusterings. Since the Normalized Mutual Information (NMI) [SG02] and the Adjusted-for-chance Mutual Information (AMI) [VEB09] fulfill all the properties required to be adequate clustering comparison methods they seem to be convincing choices to compare different clusterings.

The Normalized Mutual Information between two clusterings U and V can be defined as

$$NMI(U,V) = \frac{MI(U,V)}{\sqrt{H(U)H(V)}}. \tag{2.15}$$

where the Mutual Information (MI) between two clusterings is normalized by the entropy ($H(U)$, $H(V)$) associated with the clusterings U and V. The Adjusted-for-chance Mutual Information additionally considers the expectation of the mutual information in order to correct for chance leading to

$$AMI(U,V) = \frac{MI(U,V) - E\{MI(M)|a,b\}}{\sqrt{H(U)H(V)} - E\{MI(M)|a,b\}}. \tag{2.16}$$

where a and b are the marginals of the contingency table of U and V and $MI(M)$ denotes the mutual information between any two clusterings associated with the contingency table M.

2.3.6 Discussion

For partitioning clustering approaches, like K-Means, K-Medoid, or EM clustering selecting a suitable k is a major problem. Additionally, these approaches are sensitive to noise. Density-based clustering approaches find clusters with varying shape and size, but they have problems finding clusters with different density. In contrast to partitioning clustering density-based and hierarchical clusterings do not require the number of clusters in advance. Density-based clustering approaches are robust against outliers in the data, but they depend on the setting of the two parameters ϵ and $MinPts$ which can be difficult to determine. Hierarchical clusterings do not only produce one flat clustering but a complete hierarchy including all data points. Nevertheless, it is still possible to extract one single clustering using a dendrogram, e.g. by a horizontal cut through the dendrogram. Disadvantages of hierarchical clustering approaches are that decisions that have been made cannot be withdrawn, they are very sensitive to noise, and the runtime complexity is at least quadratic regarding the number of objects.

To overcome some of the disadvantages of the presented clustering approaches combinations of those techniques have been proposed like e.g. density-based hierarchical clustering. Additionally, approaches trying to solve the problem of parameter selection like X-Means, RIC, or OCI have been proposed. Although, the repeated execution of the underlying algorithm leads to a runtime increase, the handling of the annoying parametrization problem by the use of information theoretic measures like BIC, or MDL implicates a huge advantage.

2.4 Outlier Detection

In the field of clustering, outliers are considered as noise, hence all objects that do not belong to a cluster are considered as outliers. There is no general valid and accepted definition for outliers. In general outliers are a relative phenomenon which always has to be evaluated in connection with the given data set. Hawkins [Haw85] defined outliers as observations that are substantially different from other observations one could even think they have been produced by a different mechanism.

The reason why outlier detection is needed, is because clustering algorithms are normally optimized to find clusters rather than outliers. A group of many untypical data objects which are somehow similar to each other would be classified as a cluster and not as outliers or noise.

If a data set contains outliers there are two different ways to solve the problem. The first variant is adjustment, i.e. one tries to derive those modeling inferences that are biased as little as possible from the number, shape, or occurrence of such an observation, than an outlier is characterized as an observation which could possibly avoid that inference. The second variant of handling outliers is the identification of such observations. The identification solution is used because observations which are considered as outliers could contain additional information about the dataset which are lost using the adjustment solution. The goal of the identification solution is to separate the data set into outliers and non-outliers.

A typical sample application of outlier identification is abuse detection. Lets say a credit card gets stolen, then in general purchasing behavior of the new credit card owner (thief) changes, hence abnormal buying patterns can be an evidence for credit card fraud. Another example would be untypical symptoms and test results which might be an indication for a disease.

In the following we will only concentrate on the identification solution in order to handle outliers. Thereby, outlier identification can be handled with

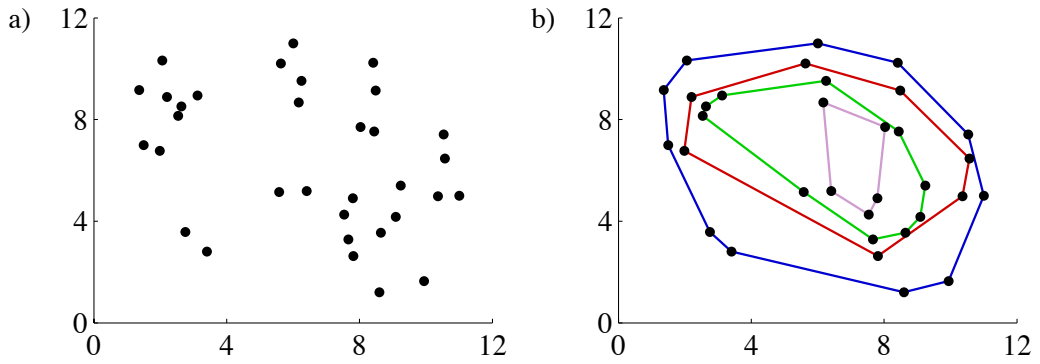


Figure 2.6: The creation of the convex hull for depth-based outlier identification. a) The untreated data set is depicted and b) shows all depths and convex hulls of the data set. The outer hull (blue line) is the convex hull with depth = 1. The hull on the very inside (purple line) is the convex hull with depth = 4.

different principles. In the following we will focus on depth-based, distance-based, and density-based outlier detection approaches.

2.4.1 Depth-based Approach

The general idea of depth-based approaches is to search for outliers at the boundary of the data space not being dependent on statistical distributions. Thereby, data objects are organized in convex hull layers and outliers are those objects being on the outer layers.

An object in depth-based outlier identification methods is represented as a point in a d dimensional space. First the depth of all objects is calculated, thereby objects on the convex hull containing the entire data space have a depth of 1, shown as blue line in Figure 2.6. If the convex hull of depth = 1 is removed, all remaining objects have a depth of 2 (red line Figure 2.6). This can go on until the last convex hull is removed leading to an empty set. Objects are considered as outliers if their depth is smaller or equal to a depth of k. Theoretically this method can also be used for high

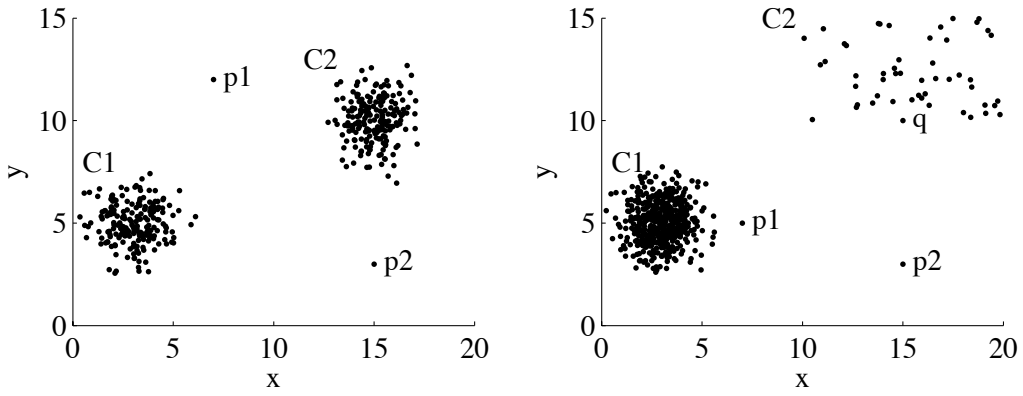


Figure 2.7: Two examples of how to select *pct* and *dmin* in order to identify outliers using a distance-based approach. For the example on the left using $pct = 0.95$ and $dmin = 5$ the outliers p_1 and p_2 can be identified. Whereas the choice of the parameters in the example on the right can not be clearly distinguish without including objects of cluster C_2 in the set of outliers.

dimensional data but in practice it is already inefficient using $d \geq 4$, because of the calculation of the convex hull. Sample algorithms for depth-based approaches are ISODEPTH [RR96] and FDC [JKN98].

2.4.2 Distance-based Approach

In general distance-based approaches judge an object based on the distance to its neighbors. An object in a data set is considered as a distance-based outlier if less than *pct* percent of the objects in the data set have a larger distance then *dmin* to that object. Thereby, *pct* and *dmin* have to be set by an expert.

An example of the selection of *pct* and *dmin* is shown in Figure 2.7 on the left. Selecting $dmin = 5$ and $pct = 0.95$, p_1 and p_2 will be identified as outliers, because 95 % of the objects have a distance which is larger than 5 to those objects. The problem with these approaches is that they use a global outlier factor. Lets say a data set contains two clusters, one very tight one

(C_1) and one cluster with a large dispersion (C_2) depicted in Figure 2.7 on the right. If $q \in C_2$ and p_1 as well as p_2 are both outliers, what would be the correct choice for *pct* and *dmin* in order to identify the outlier without selecting objects of C_2 as outliers?

For distance-based outlier identification different algorithms have been proposed. Knorr and Ng [KN98, KN97, KN99] introduced three different algorithms for this purpose: index-based, nested-loop based, and grid-based outlier identification. The index-based approach computes a distance range join using a spatial index structure. The nested-loop based approach divides the buffer into two parts and uses the second part to compare all points with the points from the first part. Finally, the grid-based approach builds a grid such that any two points from the same grid cell have a distance of at most *dmin* to each other, thereby points only need to be compared with points from the neighboring cells.

2.4.3 Density-based Approach

In the presence of clusters with varying density, distance-based approaches have problems identifying outliers. In those cases density-based approaches would be a better choice. The general idea of density-based approaches is to compare the density around an object with the density of its local neighbors. The relative density of a point compared to its neighbors is thereby computed as an outlier score. Two example approaches of density-based outlier detection are the Local Outlier Factor (LOF) [BKNS00] and the Local Outlier Correlation Integral (LOCI) [PKGF03] which will be explained in more detail in the following.

Local Outlier Factor LOF is a density-based approach which compares the density of an object to the densities of its neighbors. Thereby, an object which is more dense than its neighbors has to be part of a cluster and an object whose density is considerably lower than that of its neighbors is an

outlier. Each object is assigned with a degree of being an outlier, called the Local Outlier Factor.

LOF defines the local neighborhood of an object using its k-Nearest Neighbors. The distance to those k objects is used to approximate a local density. The k-distance of an object p is defined by the distance of p to its k-Nearest Neighbors. The distance of p to an object o in the database DB is called k-distance(p), if at least k objects $q \in DB$ have a distance $dist(p,q) \leq dist(p,o)$ and there are at most k-1 objects $q \in DB$ having a distance $dist(p,q) < dist(p,o)$. The k-distance neighborhood, $N_k(p)$, contains all objects which are closer to p than k-distance(p).

$$N_k(p) = \{q \in DB \setminus \{p\} | dist(p,q) \leq k\text{-distance}(p)\} \tag{2.17}$$

Based on this distance the reachability distance of p with respect to an object o can be defined as

$$\text{reach-dist}_k(p,o) = \max\{k\text{-distance}(o), dist(p,o)\} \tag{2.18}$$

The reachability distance of an object p to o is, therefore either the true distance between p and o or at least the k-distance of o. Figure 2.8 displays the reachability distance of an object p with k = 3. The reachability distance of p with respect to the objects o, p, and s is always the k-distance(o).

The local reachability distance (lrd) of p is defined as

$$lrd_k(p) = 1/\left(\frac{\sum_{o \in N_k(p)} \text{reach-dist}_k(p,o)}{|N_k(p)|}\right) \tag{2.19}$$

This quotient is the mean reachability distance of p to its neighbors. Finally, the local reachability distance is compared to that of the neighbors resulting

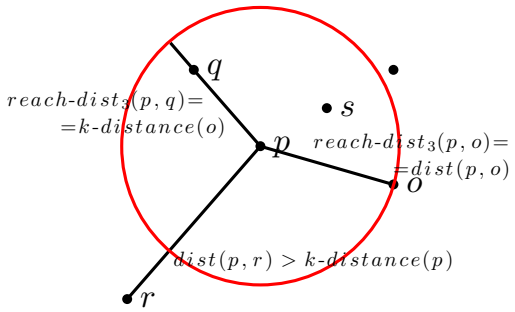


Figure 2.8: An illustration of the reachability distance. Objects o, q, and s have the same reachability distance for $k = 3$, whereas object r is not a k-Nearest Neighbor of objects o, p, q, and s.

in the Local Outlier Factor:

$$LOF_k(p) = \frac{\sum_{o \in N_k(p)} \frac{lrd_k(o)}{lrd_k(p)}}{|N_k(p)|} \tag{2.20}$$

A $LOF_k(p) \approx 1$ means that an object is close to the middle of a cluster and a $LOF_k(p) \gg 1$ denotes that an object is a strong local outlier.

An extension of LOF is the Local Outlier Probability (LoOP) [KKSZ09] which statistically estimates the density in order to be less dependent on the exact value of k. Additionally the results are statistically normalized to be located between 0 and 1 to achieve a better interpretability of the values.

Local Outlier Correlation Integral An approach whose idea is similar to LOF is LOCI. In contrast to LOF, LOCI does not take the k-Nearest Neighbors as reference set but rather the ϵ-neighborhood. An additional difference is that LOCI does not require any input parameters instead it tests multiple granularities of the reference set. In order to obtain the local density of an object p, which is the set of objects in the ϵ-neighborhood,

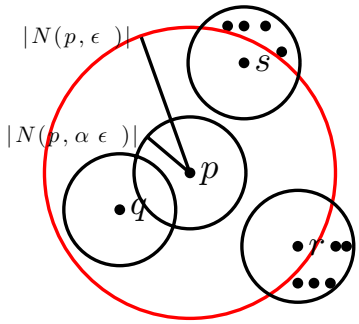


Figure 2.9: An illustration of the cardinality of the local density $|N(p,\epsilon)|$, $|N(p,\alpha\epsilon)|$, and of the average density of the neighborhood $\hat{n}(p,\epsilon,\alpha)$. For example $|N(p,\epsilon)| = 4$ (objects inside red circle), $|N(p,\alpha\epsilon)| = 1$ (objects inside black circle around p, including p), $|N(s,\alpha\epsilon)| = 5$, $|N(r,\alpha\epsilon)| = 6$, and $\hat{n}(p,\epsilon,\alpha) = (1+5+6+1)/4 = 3.25$.

$N(p,\epsilon)$ has to be defined as

$$N(p,\epsilon) = \{p | dist(p,q) \leq \epsilon\}. \tag{2.21}$$

The average number of ϵ-neighbors of p, hence the average density of the neighborhood is

$$\hat{n}(p,\epsilon,\alpha) = \frac{\sum_{q \in N(p,\epsilon)} |N(q,\alpha\epsilon)|}{|N(p,\epsilon)|}. \tag{2.22}$$

An intuitive illustration of the density of the neighborhood and the cardinality of the local density of an object p is shown in Figure 2.9. Finally, the Multi-granularity Deviation Factor (MDEF) is defined as

$$MDEF(p,\epsilon,\alpha) = \frac{\hat{n}(p,\epsilon,\alpha) - |N(p,\alpha\cdot\epsilon)|}{\hat{n}(p,\alpha,\epsilon)} = 1 - \frac{|N(p,\alpha\cdot\epsilon)|}{\hat{n}(p,\alpha,\epsilon)} \tag{2.23}$$

If MDEF equals zero the objects are cluster members, if MDEF is larger than 3 times the normalized standard deviation of the densities of all points from $N(p,\epsilon)$ than the object is considered an outlier.

An additional feature of LOCI is the LOCI plot which displays for a

given point p with respect to ϵ, the cardinality of the ϵ-neighborhood and the average density of the neighborhood with a border of $\pm 3\sigma\hat{n}(p, \epsilon, \alpha)$.

Since the exact solution is very expensive due to the fact that MDEF values have to be calculated for all possible ϵ values, the procedure aLOCI [PKGF03] was proposed which is a fast, approximated solution of LOCI. Thereby, the data space is discretized using a grid with a side length of $2\alpha\epsilon$. The range queries are approximated through grid cells and the ϵ-neighborhood of an object is defined for all cells that are completely covered by the ϵ sphere around the object.

2.4.4 Discussion

In general different models are based on different assumptions to model outliers and provide different types of output like labeling or scoring. They can either consider outliers at global or local resolution, therefore, the different approaches will produce different results.

Depth-based approaches have similar basic concepts as classical statistical approaches using k = 1 distributions, except for the independence of the chosen kind of distribution. They use a global reference set and are usually only efficient in 2 or 3 dimensional spaces due to the calculation of the convex hull. Although the original output is only a label it can be easily extended to a scoring output taking the depth as scoring value. Distance-based approaches can either produce a scoring or a labeling output just like density-based approaches. They use a local reference set where the resolution can be adjusted via *dmin* and *pct*. Density-based approaches have an exponential runtime with respect to the data dimensionality. Outlier detection is an important task to further improve the process of clustering and to identify abnormal or extraordinary objects in data sets. At present, a thorough and comprehensive comparison between the different models and approaches is still missing which would help to further understand the differences and common properties.

2.5 Classification

Classification approaches construct models to assign unknown objects to predefined classes which were build using object and environment specific features. The difference between classification and clustering is that classification has to know the class label of the objects in the database a priori and the goal of clustering is to search for the class labels. The assignment of class labels to unknown query objects allows the association of class specific features with the object to derive the expected characteristics of the object. In order to establish a mathematical model the fixed number of classes present in the database as well as examples of class instances with known class labels have to be known in advance. Auxiliary mathematical methods for the construction of classification models can be based on both classical statistics (e.g. discriminant analysis or k-Nearest Neighbor methods) and on machine learning. Symbolic learning approaches like the decision tree or rule induction constitute approaches which construct understandable classification descriptions for the user. In contrast to that, sub-symbolic approaches like artificial neural networks use the black box principle, for those approaches class descriptions cannot be directly derived from the constructed model.

Using classification, classes or categories are defined which are then allocated to unknown objects. This allocation works based on comparisons between class features and object characteristics. The classifier is thereby the function which assigns the corresponding classes to the objects. In the following we will explain different classification techniques and their quality measures.

2.5.1 Bayes Classifier

A Bayes classifier is a statistical classifier based on the theorem of Bayes.

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} \tag{2.24}$$

with $P(A_i)$ and $P(B)$ being the priori probability of A_i and B and $P(A_i|B)$ being the conditional probability of A_i given B. Bayes' theorem estimates the probability with which a feature vector belongs to a class. Each object is either assigned to the class with the highest probability or to the class whose costs are increased the least when adding the object. To define a Bayes classifier a cost function is needed which allocates certain costs to each possible classification. The goal of the Bayes classifier is to minimize the produced costs which arise from the classifications.

Bayes classifiers have a high classification accuracy in many applications. Additionally, the classifiers are incremental meaning that classifiers can be easily adjusted to new training objects. But since the classifiers need a priori knowledge for the conditional probabilities they are not applicable in many cases.

Naive Bayes Classifier In high dimensional data the estimation of the conditional probabilities is time and memory consuming since all attribute combinations have to be considered. The probabilities can not be saved anymore, hence, the naive Bayes classifier [LIT92] assumes conditional independence.

$$P(A_1 \wedge A_2|B) = P(A_1|B) \cdot P(A_2|B) \tag{2.25}$$

Nevertheless, this assumption can also be wrong leading to unsatisfying results, e.g. if all attributes are equally distributed for several classes. Despite the fact that the independence assumptions are often inaccurate, in reality naive Bayes classifiers often achieve good results in case of uncorrelated data. In the presence of strong attribute correlations naive Bayes classifiers should be extended by a tree between the attributes.

For now the independent attribute model, meaning the naive Bayes probability model, has been derived. To realize a naive Bayes classifier which is a combination of the independent attribute model and decision rules the Maximum A Posteriori (MAP) decision rule can be used, choosing the class with

the most probable hypothesis. The probability of belonging to a class A_i can be described as

$$P(A_i|B_1 \wedge B_2 \wedge ...) = \frac{P(A_i)P(B_1 \wedge B_2 \wedge ...|A_i)}{P(B_1 \wedge B_2 \wedge ...)} = \frac{P(A_i)\prod_j P(B_j|A_i)}{\sum_k \prod_j P(B_j|A_k)} \tag{2.26}$$

and hence

$$A = arg\max_{A_i}\{P(A_i)\prod_j P(B_j|A_i)\} \tag{2.27}$$

due to the fact that the denominator is equal for all classes.

2.5.2 Nearest Neighbor Classifier

Nearest Neighbor classifiers [CH67] are based on instance based learning. The basic Nearest Neighbor classifier is the assignment of an object to the class of its Nearest Neighbor object based on a similarity measure like the Euclidean distance. Thereby, the region which is used to assign a class label to an unknown object can be displayed by Voronoi diagrams as shown in Figure 2.10. Since a basic Nearest Neighbor classifier is very sensitive to noise in most cases it is superior to use a k-Nearest Neighbor (k-NN) classifier for the classification to obtain a more robust outlier handling.

k-Nearest Neighbor Classifier The difference between a basic Nearest Neighbor classifier and a k-NN classifier is that instead of considering solely the most similar object in the database for the decision process the k-NN classifier considering the k closest neighbors of the query object. To ascertain the class assignment based on the k-NNs of the decision set either the majority class is selected (maximum likelihood method) or weighted decision rules can be applied (classes in the decision set are weighted either by their distance or their distribution). Thereby, the correct setting of k in a k-NN search depends on the underlying data. If k is chosen too small the sensitivity for outliers is very high, by contrast if k is set too large an oversized number

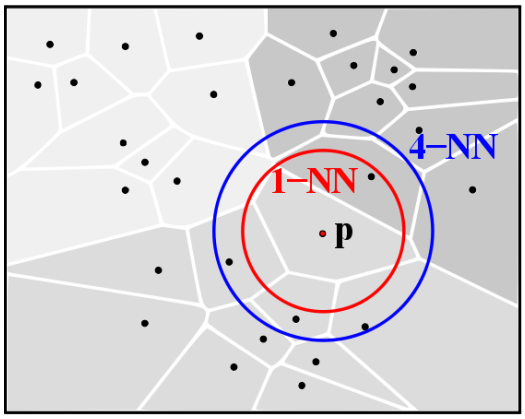


Figure 2.10: A Voronoi diagram of a data set consisting of 3 classes. Voronoi diagrams can be used to illustrate the class assignment of Nearest Neighbor classifiers.

of objects from other classes gets enlisted in the set of Nearest Neighbors. It has been shown that choosing k between $1 \ll k < 10$ seems to be a good choice.

An example of a Nearest Neighbor search and a k-NN search is depicted in Figure 2.10. Using a basic Nearest Neighbor approach (depicted as 1-NN in the Figure), object p would be assigned to the dark gray class since the closest object belongs to the dark gray class indicated by the red circle. Whereas, a k-NN classifier considering the majority class with $k = 4$ would correctly assign object p to the middle gray class since three objects of the decision set (k closest objects) belong to the middle gray class and only one object belongs to the dark gray object (blue circle).

k-Probable Nearest Neighbor Classifier A variant of the k-NN classifier which is able to handle uncertain data objects is the Top k-Probable NN (Top k-PNN) classifier [BSI08]. As mentioned in Section 2.2 uncertain objects consist of values with margins of error which can be described using PDFs. Thereby, the Top k-PNN approach tries to find uncertain objects by selecting those objects having the highest marginal probability to the given

query object.

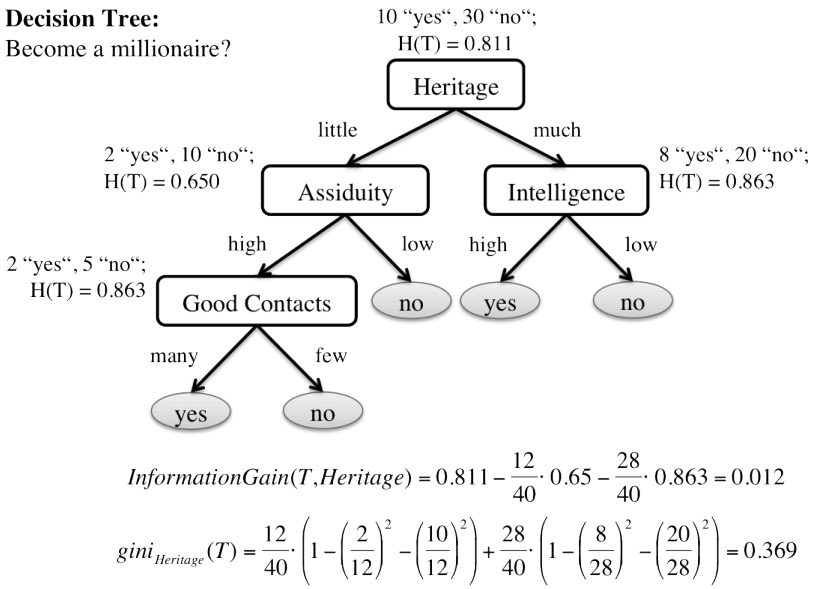
k-Most Likely Identification Query Classifier The k-Most Likely Identification Query (k-MLIQ) [BPS06] was also invented to handle uncertain data queries. A k-MLIQ is very similar to the k-NN approach differing mainly in the choice of the similarity measure. Instead of using distances to determine the most similar objects, the k-MLIQ method computes matching probabilities between the database object and a query object. In contrast to the Top k-PNN classifier the k-MLIQ classifier considers the complete information contained in the PDF of the uncertain objects leading to a more accurate similarity measure.

2.5.3 Decision Tree Classifier

Decision trees are ordered directed trees which are applied to illustrate decision rules [BFSO84]. They hierarchically exemplify successive decisions and they are widely used in application areas where classification is done automatically or where formal rules are deduces from know-how.

The inner knots of a decision tree represent attributes, an edge represents an attribute test on the parent knot, and a leave represents the class to which an object will be assigned. The construction of a decision tree is performed in a top-down manner. For the classification of an object the decision tree has to be traversed from the root to a leave of the tree. At each knot the attribute is tested specifying the edge that leads to the next knot. Hence, there is a distinct classification path through the tree. This procedure is executed until a leaf is reached which specifies the classification. In general, a decision tree contains rules to answer exactly one question.

It is substantial for decision trees to have a representative training data set with reliable experience to the problem which is going to be solved. Starting with the entire training data set being located in the root knot, in each



$$InformationGain(T,Heritage) = 0.811 - \frac{12}{40}\cdot 0.65 - \frac{28}{40}\cdot 0.863 = 0.012$$

$$gini_{Heritage}(T) = \frac{12}{40}\cdot\left(1-\left(\frac{2}{12}\right)^2-\left(\frac{10}{12}\right)^2\right)+\frac{28}{40}\cdot\left(1-\left(\frac{8}{28}\right)^2-\left(\frac{20}{28}\right)^2\right) = 0.369$$

Figure 2.11: A decision tree example illustrating the entropy, the information gain, and the gini index.

induction step the attribute which most reliably classifies the training data with respect to the target attribute is searched for. As stability index for the classification one can use measures like the information gain or the Gini index. An example decision tree of whether or not to become a millionaire is depicted in Figure 2.11 having an information gain for the attribute heritage of 0.012 and a Gini index of 0.369. In the next step the training data is subsequently split according to the identified attribute and the procedure is recursively applied to the newly generated subsets until either each subset only contains objects with the same classification or no further splitting attributes are left.

One large problem when working with decision trees is overfitting. If a decision tree is excessively adjusted to the training data the decision tree tends to overfit the training data leading to a large error rate on the entity of the data. To overcome overfitting erroneous training objects have to be removed and the size of the training set has to be adjusted before building the decision tree. The effect of overfitting can also be diminished by subsequent

pruning of the decision tree. Thereby, after the decision tree has been built on the basis of the training set, the test set can be used to prune the tree by cutting those branches which cause the largest classification error reduction.

2.5.4 Support Vector Machine Classifier

The goal of Support Vector Machines (SVM) [CV95] is to find a linear separation, i.e. an SVM searches for a hyperplane which separates two classes maximally stable. An SVM separates a set of objects into classes so that the area around the class margins which is free of objects is as large as possible.

Given a set of training objects with known class labels each object is represented by a vector in a vector space. The task of a Support Vector Machine is to fit a hyperplane in the vector space which serves as separation plane separating the training data according to its class labels as accurate as possible. Thereby, the distance of those objects being closest to the hyperplane is maximized. This large empty space makes sure that objects that are different from the training objects can still be reliably classified. Building the Maximal Margin Hyperplane (MMH), it is not necessary to consider all training objects, only those objects having the smallest distance to the hyperplane have an influence and are needed to describe the plane mathematically. These objects are called support vectors.

A hyperplane can only linearly separate objects. But since in general real world data cannot be linearly separated, the SVM uses a kernel trick in those cases, to constitute a non-linear class margin. The basic idea of the kernel trick is to transform the vector space and all the training objects into a higher dimensional space. In a space with a sufficient number of dimensions, which might even be an infinite number, even the most interleaved set of objects gets linearly divisible. In this higher dimensional space the MMH is selected. Transforming the linear hyperplane back into the original lower dimensional hyper space the linear hyperplane becomes non linear and might

even consist of several non connected hyperplanes. These hyperplanes are then able to accurately separate the training data into two classes. Since the transformation in the higher dimensional space is computationally very intensive and the delineation of the hyper margin in the lower dimensional space is extremely complex and hence virtually unfeasible, the kernel trick is applied. Kernel functions can be used to characterize the MMH which delineates the hyperplane in the higher dimensions and nevertheless stays well behaved in lower dimensions. Therefore, using kernel functions it is possible to realize the back and forth transformation without actually having to calculate it.

2.5.5 Cross Validation

A major problem of classification approaches is over-fitting. Classifiers are optimized using training data which they can in general handle well but they can still achieve poor results on the entity of the data set, because they have been over-fitted to the training data. In order to prevent over-fitting the entire data set O is split into a training set to generate the classifier (construct the model) and a test set to judge the classifier. This procedure is called train-and-test.

Though, in many application fields the amount of data is very limited and, therefore, a different procedure has to be applied, called m-fold cross validation. As illustrated in Figure 2.12, m-fold cross validation first separates the data set into m equally sized subsets, then m-1 subsets are combined to build the training set and the remaining subset is used to estimate the classifier [Koh95]. Finally, the obtained m classification errors and the m generated models are combined to one final model. A special case of the m-fold cross validation is the leave-one-out cross validation. As the name suggests, it involves using a single object from the data set as test data and the remaining data as training data. This is repeated until each object in the data set has been used once for testing. Therefore, leave-one-out cross

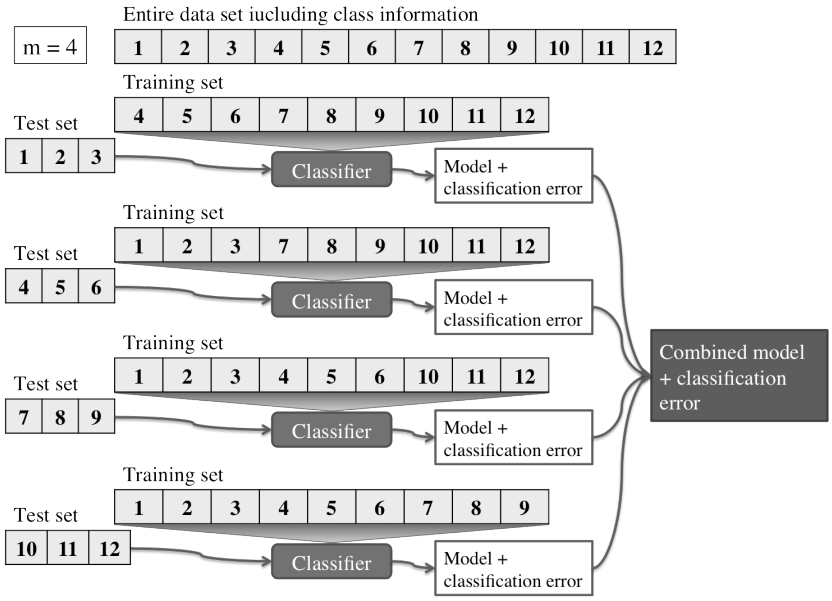


Figure 2.12: The process of m-fold cross validation with m=4. The entire data set shown on top is subdivided into 4 equally large subsets. One subset is used as test set and the remaining 3 subsets are combined to one training set.

validation is the same as an m-fold cross validation with m being the number of objects in the data set.

2.5.6 Quality Measures

In general, classifiers assign a certain amount of objects to wrong classes meaning they make classification errors. Quantitative measures for the evaluation of a classifier can be derived from the relative frequency of these errors. In many cases there are only two possible classes that means the classification is binary. These classifications answer questions like does a patient have a certain disease or not, or is a hurricane going to hit a city or not.

To evaluate a classifier it has to be applied to a certain amount of objects for which class labels are known. Then the resulting classifications obtained

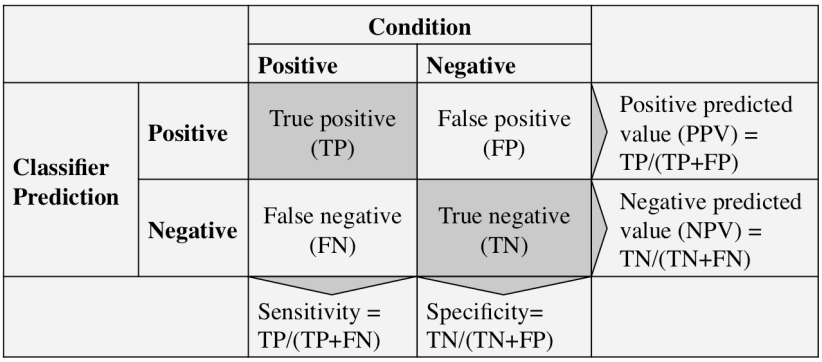


Figure 2.13: Illustration of different classification quality measures.

by the classifier are separated into four groups shown in Figure 2.13. These four groups will be explained on a short example. Let's say the classifier is supposed to distinguish if a person has a disease or not. In order to evaluate the classifier we have to know in advance if the people that are tested really have the disease or not, this is the condition or gold standard. All people that have the disease and are correctly classified as having the disease, are considered True Positives (TP). Those people that do not have the disease and are correctly classified as not having the disease are considered True Negatives (TN). If a person has the disease and was falsely classified as not having the disease this person is a False Negative (FN). The last group consists of those people that do not have the disease but they were falsely predicted to have the disease they are called False Positives (FP).

Based on those four classes (TP, TN, FN, FP) several different classification quality measures can be constituted. The sensitivity also called recall or True Positive rate (TP-rate) indicates the fraction of correctly positive classified objects from the complete set of positive objects. Therefore, the sensitivity is the estimated conditional probability

$$\text{Sensitivity} = \text{Recall} = \text{TP-rate} = \frac{TP}{TP + FN} \tag{2.28}$$

Accordingly, the False Negative rate (FN-rate) denotes the fraction of the falsely negative classified objects from the complete set of positive objects.

$$\text{FN-rate} = \frac{FN}{TP + FN} \tag{2.29}$$

Since both measures refer to the positive category they add up to 1, hence the sensitivity = 1– FN-rate and vice versa.

The specificity also called True Negative rate (TN-rate) declares the fraction of correctly negative classified objects from the entity of real negative objects.

$$\text{Specificity} = \text{TN-rate} = \frac{TN}{TN + FP} \tag{2.30}$$

Accordingly the False Positive rate (FP-rate) corresponds to the fraction of falsely positive classified objects from the complete set of negative objects.

$$\text{FP-rate} = \frac{FP}{TN + FP} \tag{2.31}$$

Also those two measures add up to 1, hence the specificity = 1–FP-rate and vice versa.

In order to detect the fraction of correctly classified objects based on the entity of positive classified objects the Positive Predicted Value (PPV) also called precision is used.

$$\text{PPV} = \text{Precision} = \frac{TP}{TP + FP} \tag{2.32}$$

Accordingly, the Negative Predicted Value (NPV) includes those objects that are correctly negative classified from all negative classified objects.

$$\text{NPV} = \frac{TN}{TN + FN} \tag{2.33}$$

In contrast to all other quality measure pairs those two measures do not add up to 1, because the denominator is different for both cases.

The last quality measure to mention is the accuracy which denotes the fraction of all objects which were correctly classified.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.34}$$

In general it is not enough to only specify one quality measure but rather a combination of several measures to indicate the real quality of a classifier. Though the choice of the measure always depends on the underlying data set.

2.5.7 Discussion

The Nearest Neighbor classifier has a good applicability since it only needs training data as input. It can easily be adapted to new training objects due to the fact that it is incremental. However, the Nearest Neighbor classifier is inefficient because it requires k-NN queries. The interpretation of a decision tree is simple, attributes are implicitly weighted, and the evaluation of the traced model is efficient. Furthermore, decision trees are powerful and effective classifiers which are often used in practical experience. But the tracing of the decision is exponential and therefore very extensive. Besides, heuristic approaches can find only local optima and decision trees are very susceptible to overfitting. In contrast to other classifiers SVMs produce very accurate results while having a rather low tendency to overfitting. The classification of new objects is very efficient and the model is compact. Drawbacks of SVMs are that the training phase can be very time consuming, the implementation is very complex, and the found models are difficult to interpret.

Chapter 3

Parameter-free Outlier Detection Using Data Compression

3.1 Introduction

In many real-world applications a certain amount of exceptional data objects which differ from the rest of the data by some extent is present. For some applications in biology or medicine the removal of those objects is essential in order to achieve accurate data mining results where regularities in the data are searched for. In medicine the advances in medical imaging techniques has lead to the generation of a huge amount of data on a daily basis, hence an automatic detection and removal of unusual data also called noise is essential to guarantee adequate data analysis. For other application fields in medicine, biology, meteorology, or economy, the identification of these exceptional objects is of great importance. For example the identification of patients that differ from the healthy population by a certain amount can help to identify specific features as disease indicators. In meteorology finding extraordinary

combinations of weather elements, e.g. before the occurrence of hurricanes or flood waves can assist in evacuating endangered people betimes and can prevent extreme damages.

Clustering as well as outlier detection are very prominent research areas attracting a lot of attention in recent years. Even if both fields are in principle interested in different aspects of data sets they are still closely related. While clustering is interested in finding similarities between objects, outlier detection looks for dissimilarities in data sets. Nevertheless, without the removal of noise objects, corrupting the clusters being identified by outlier detection, clustering is not able to acceptably handle most real world data sets leading to incorrect clustering results. Hence, the preliminary removal of outliers is essential to guarantee a good clustering quality. Outlier detection algorithms need a clear description of the distribution of cluster objects to be able to separate the outliers from the cluster objects even if cluster objects are not of primary interest. Only if cluster objects are clearly defined outliers can be identified by the algorithm.

There are several different outlier definitions, for our approach we used the definition by Hawkins [Haw80] which states that outliers are observations that deviate so much from other observations as to arise suspicion that they were generated by a different mechanism. To be able to realize this outlier definition, outliers as well as general cluster objects have to be clearly distinguished by means of a discrepancy criterion. This discrepancy criterion has been quantified using either computational geometry by calculating the convex hull [JKN98] or by utilization of a specific distance function [KN99]. Most existing approaches have in common that they strongly rely on adequate parameter settings. Identified outliers are only extraordinary if parameters have been set correctly depending on the unknown data distribution and if the distance function or convex hull fits to the data. Hence, in order to apply existing outlier detection approaches a priori knowledge of the data distribution is required which is not given in most real world data sets.

In the following we will propose a new parameter-free outlier detection algorithm, to overcome the problems of finding a discrepancy criterion and setting parameters which are hard to determine. Since our approach employs data compression we can avoid tedious parameterization. To cope with the definition of a discrepancy criterion we use an adaptable description for the ordinary data. For describing regular data we apply the adjustable Generalized Normal Distribution (GND) which is a generalization of the Gaussian Probability Density Function in combination with Independent Component Analysis (ICA) converting the data into highly compressed independent components. The GND incorporates several different distribution types like e.g. the Uniform, the Gaussian, and the Laplace distribution. In our experiments (Section 3.4) we verify the great adaptability of the GND to different data distribution types.

An object is regarded an outlier if it does not suit well in the estimated PDF of any arbitrarily sized neighborhood of the object. The goodness of a fit is determined by the principle of data compression where an object can be efficiently compressed if it fits well into a distribution function. Applying the concept of Huffman coding [Huf52] an object being a vector in a d-dimensional space is flagged with sparse bits if it has frequent coordinate values while an object having a seldom combination of coordinate values receives many bits. Applying ICA in combination with GND the redundancy in the data can be minimized and determination of frequent and seldom objects can be clearly identified. The used data compression concept is called Minimum Description Length (MDL). In our approach we employ the MDL concept in order to obtain an outlier detection algorithm without the need of any explicit parameter settings. Hence, no prior knowledge of the underlying data is required to apply our approach like the approximate amount of outlier objects present or the density of the clusters. Moreover, our outlier score which arises from the coding costs of an object with respect to the entire data set simplifies the handling and allows to identify outliers by the formation of an outlier ranking.

The following sections are composed as follows: Section 3.2 gives an overview of existing outlier detection approaches and the MDL principle in the field of clustering and other data mining procedures. Section 3.3 characterized our main outlier detection algorithm, including detailed descriptions of the ICA and GND which are applied to generate a very general data description model. Moreover, we utilize the MDL principle to define our outlier score. In Section 3.4 we demonstrate the broad applicability by an extensive experimental evaluation including synthetic and real world data and in Section 3.5 we conclude Chapter 3.

3.2 Related Work

In general outlier detection approaches can be divided into three major categories namely depth-based, distance-based, and density-based approaches. In the following we will give a short summary of different outlier detection approaches. Furthermore, since our approach is based on the information-theoretic MDL principle we will give a short overview of existing information-theory based data mining approaches. For more information concerning anomaly detection, extensive reviews can be found at [CBK09, PP07].

3.2.1 Depth-based Outlier Detection

In order to identify outliers each data point in depth-based approaches is associated with a certain depth. According to the assigned depths the data points can than be arranged in different layers indicated by the convex hulls. Outliers are thereby objects that are expected to be on the shallow layers in contrast to the cluster objects being located on the deep layers. A depth-based outlier detection method which determines 2 dimensional depth contours is called ISODEPTH [RR96]. An extension of ISODEPTH is FDC [JKN98] which constrains the determination of contours to a selected sub-

set of points. The fitting of distributions is avoided and in general multi-dimensional data could be processed. However, depth-based outlier detection approaches have problems with growing dimensionality they only give acceptable performance for $k \leq 2$ since depth-based approaches rely on the calculation of the convex hull [RR96].

3.2.2 Distance-based Outlier Detection

The concept of distance-based outlier detection has been introduced and advanced by Knorr and Ng [KN98, KN97, KN99]. In order to detect an outlier object in distance-based outlier detection two parameters have to be set, in fact *pct* which indicates the minimal fraction of points in a database having a larger distance than *dmin* to that object. Distance-based methods depend on the calculation of distance values which are based on a distance metric function. Objects are labeled as outliers and non-outliers in these approaches. In [KN99] a method providing intensional knowledge for the extraordinariness of an outlier is proposed. This intentional knowledge supports semantic interpretation of the identified outliers. Even if proposed distance-based methods are very efficient, the need of a suitable distance metric as well as the correct setting of the parameters *pct* and *dmin* which have to be conducted by a domain expert make an application of distance-based outlier detection very difficult. The data distribution has to be known in advance which is not the case for many real world data sets. Additionally, only global outliers can be identified since the distance threshold *dmin* is fixed to a single value for the entire data set.

3.2.3 Density-based Outlier Detection

The basic concept of density-based outlier detection has been adopted from density-based clustering approaches. In contrast to distance-based outlier detection, density-based approaches are also able to identify local outliers. The

key assumption is that cluster objects are located in dense neighborhoods, whereas outliers do not fit into the objects neighborhood density therefore being far from their closest neighbors.

3.2.3.1 Local Outlier Factor

Breunig *et al.* [BKNS00] were the first to apply the concept of object density to outlier detection. This approach depends on the Local Outlier Factor (LOF) which is assigned to each object. The LOF relies on the local neighborhood density of the objects and indicates the degree of "outlierness". In order to obtain the LOF of an object the parameter $MinPts$ has to be set. $MinPts$ which is the number of Nearest Neighbors of an object is used to define the local neighborhood of the object. To determine the LOF score of an object, the average local density of the $MinPts$-Nearest Neighbors of an object has to be divided by the local density of the object. An object is an outlier if it has a large LOF score (LOF $\gg 1$). Although LOF is not affected by the local density problem, the selection of the global $MinPts$ parameter has a strong impact on the resulting outlier set. If $MinPts$ is chosen too large, objects belonging to small clusters can be regarded as outliers and if $MinPts$ is chosen too small no outliers are detected at all. Additionally, LOF only returns an outlier ranking of the objects, therefore an approximate number of outliers has to be known in advance which is not given for real world data sets. Hence, LOF is only applicable if looking for the k most outlying objects but it is not applicable to remove noise points from a data set.

3.2.3.2 Local Correlation Integral

Another density-based approach called Local Correlation Integral (LOCI) [PKGF03] uses the Multi-granularity Deviation Factor (MDEF) to detect outliers. MDEF is a variation of LOF, thereby an object is an outlier if the

density in its local neighborhood deviates from the local density of the object's nearest neighbors including the object itself. To accelerate the determination of the multi-granularity the counting and the sampling neighborhood are introduced. The sampling neighborhood contains all objects which are in the set of Nearest Neighbors of the object, it is used to calculate the mean neighborhood density of the object. The counting neighborhood is used to estimate the local density of the object. The sampling neighborhood, being larger than the counting neighborhood, is used to collect samples of the counting neighborhood for a more precise estimation of the average neighborhood density. This separation of sampling and counting neighborhood leads to more robust results in some cases as compared to LOF which does not use different neighborhoods. Moreover, the separation makes the algorithm more efficient with respect to the multi-granularity calculation. Nevertheless, the separation leads to more parameters which have to be set, namely α which is required for the counting neighborhood and r_{min} specifying the minimum radius of the sampling neighborhood. In addition to the outlier factor LOCI introduces a specific type of visualization. This so-called LOCI plot can be used to determine further information about the vicinity of the objects, giving information about the outlier objects or about close clusters and micro-clusters. However, LOCI and LOF both use Euclidean distances to determine the density of the objects. Furthermore, to identify outliers using LOCI the authors recommend to use three times the standard deviation of the overall object density of the sampling neighborhood as threshold which presumes Gaussian distributed data.

3.2.4 Data Mining And MDL

The main concept of information theory is to find models which can efficiently find and learn regularities from data. These regularities can then be used to compress the data more powerfully. The concept of information-theory has been applied to a variety of data mining research areas. In particu-

lar, the MDL principle as well as related information-theoretic measures like BIC or AIC have been adapted to different data mining fields like clustering [BFPP06, BFP08, PM00], rule mining [YMW02], classification [KK06], regression [KK06], and anomaly detection [KLR04]. The MDL principle is an important concept in information theory and learning theory, because it draws a connection between learning and data compression. To avoid parameters in clustering and classification MDL can be used to distinguish the best fitting model. Thereby, MDL contrasts different models being able to find a compromise between model quality and complexity. Since MDL has shown to be a good measure for clustering and other data mining techniques we decided to adopt MDL to outlier detection, which as far as we know has not been done yet.

The research field which is most similar to outlier detection with respect to the problem description is clustering. Most outlier detection principles are based on clustering concepts but clustering and outlier detection differ strongly in the output of the algorithms. While clustering searches for subgroups having strong similarities, outlier detection is mainly interested in the dissimilarities. For clustering, outliers or noise can be a big problem since noise is dissimilar to all other objects and would therefore need an own cluster. X-Means [PM00] as a parameter-free extension of the partitioning clustering approach K-Means [DH73] uses the information-theoretic criterion BIC to overcome the parameter k which is responsible for the number of clusters. X-Means is very sensitive to noise since it uses spherical Gaussians as cluster description. Another clustering approach called Robust Information-Theoretic Clustering (RIC) [BFPP06] has been proposed to post-process a clustering. For clustering any conventional clustering algorithm can be used. After primary clustering RIC starts by clarifying the initial clustering from noise followed by a cluster improvement step. The improvement is implemented by the determination of a model for each cluster where each model implies a rotation matrix which has been assigned by Principle Component Analysis and a PDF. Each object is thereby allocated

with one of a number of previously defined PDFs. A further approach which has been introduced lately is called Outlier-robust Clustering using Independent components (OCI) [BFP08]. OCI applies a general PDF, called the Exponential Power Distribution (EPD), in combination with ICA to define a global clustering concept. Some approaches have used the MDL principle to de-noise signals of time series [Ris00, XZX04] with the main intention to remodel the signal as precise as possible.

3.3 Outlier Detection Via Minimum Description Length

We introduce an outlier detection algorithm which uses the Independent Component Analysis to reduce redundancy in the data in combination with the Generalized Normal Distribution to approximate the data. The proposed approach is entirely parameter-free which can be achieved using MDL for data compression based on Huffman coding [Huf52]. Following Hawkin's outlier definition [Haw80], an outlier is defined as an object having extraordinary large coding costs which is in our approach determined by the MDL principle. To identify the set of outliers we use X-Means clustering [PM00]. Thereby, the cluster having the largest coding cost cluster mean is returned as outlier set. Since we use the MDL principle to assign coding costs to the objects we nicely avoid the use of any distance metric, which would require thresholding rendering very difficult without prior knowledge of the neighborhood.

Data sets located in cartesian coordinate systems of real world applications are often spread, rotated, and distorted. The ICA is able to handle data sets which are non-orthogonal. The identification of the independent components can be achieved since the ICA maximizes the statistical independence of the estimated components. As independence definition we used the maximization of non-Gaussianity which is motivated by the central limit

theorem [Fis10]. Before starting with the actual ICA the data has to be preprocessed by centering, whitening, and reduction of the dimensionality. We used Principle Component Analysis (PCA) [Ait84] for whitening and dimensionality reduction of the data. For the actual ICA we used the FastICA algorithm [HKO01], but the idea of ordinary points still needs to be clearly defined.

Most existing outlier detection algorithms assume the underlying data to solely follow a Uniform or Gaussian distribution but since we did not want to restrict our approach to solely one distribution type we decided to incorporate a generalization of the Gaussian PDF called the Generalized Normal Distribution. It includes different distributions, besides the Gaussian PDF, it also includes the Uniform, and the Laplace PDF. The usage of the GND keeps us from requiring any prior knowledge on the type of data distribution. Hence, no bias to models with Gaussian data is created. Finally, in applying a combination of ICA and GND as description of a regular data subset we are able to comprise a diversity of real-world data sets without explicit assumptions on cluster density, shape, and orientation.

In the following we will explain the general algorithm, and the principles of ICA, GND, and data compression. Furthermore, the connection between ICA and GND as well as data compression leading to our outlier approach will be described.

3.3.1 Algorithm

Since outliers have unusually high coding costs in contrast to cluster points having coding costs close to zero, our approach is able to identify outliers completely automatic. Choosing a well fitting compression model for ordinary objects we used a bottom-up approach to detect all irregular data objects. The algorithm (Alg. 3.1) is processed for each data object x separately. Starting with the initialization of a substantial set of Nearest

Algorithm 3.1 Outlier Detection

Input: Database DB
Output: Set of outliers O

$OS = []$; // *outlier scores*
for all objects $x \in DB$ **do**
  // *initial set of Nearest Neighbors determined by Euclidean distances*
  $nn_x = initializeNN$;
  $not\_nn_x = DB \setminus nn_x$;
  $cost_x = []$; // *coding costs of x*
  $cost\_min_{nn_x} = []$; // *minimal coding cost of x*

  // *determine de-mixing matrix and NN mean and transform data using ICA*
  $[M_{nn_x}, m_{nn_x}] = \text{ICA}(nn_x)$;
  $nn_{x,ica} = \text{transform}(nn_x, M_{nn_x}^{-1}, m_{nn_x})$;
  $[\mu_{nn_x}, \sigma_{nn_x}, p_{nn_x}] = \text{estimateGND}(nn_{x,ica})$;// *estimate GND parameters*
  while $not\_nn_x \neq []$ **do**
    $cost_x = cost_x \cup \text{coding\_cost}(x_{ica}; M_{nn_x}^{-1}, \mu_{nn_x}, \sigma_{nn_x}, p_{nn_x})$; // *Eq. 3.19*
    $cost\_min_{nn_x} = cost\_min_{nn_x} \cup$
      $\min(\text{coding\_cost}(nn_{x,ica}; M_{nn_x}^{-1}, \mu_{nn_x}, \sigma_{nn_x}, p_{nn_x}))$;

    $not\_nn_x = \text{transform}(not\_nn_x; M_{nn_x}^{-1}, m_{nn_x})$;
    $cost_{not\_nn_{x,ica}} = \text{coding\_cost}(not\_nn_{x,ica}; M_{nn_x}^{-1}, \mu_{nn_x}, \sigma_{nn_x}, p_{nn_x}))$;
    $nn_x = nn_x \cup \{not\_nn_{o,ica} \text{ with } \min(cost_{not\_nn_{x,ica}})\}$
    $not\_nn_x = DB \setminus nn_x$;

    update M_{nn_x}, m_{nn_x};
    $nn_{x,ica} = \text{transform}(nn_x, M_{nn_x}^{-1}, m_{nn_x})$;
    update $\mu_{nn_x}, \sigma_{nn_x}, p_{nn_x}$;
  end while
  $j = \min(cost\_min_{nn_x})$; // *index of best compressed set of NN*
  $OS = OS \cup (cost_x(j) - \min(cost\_min_{nn_x}))$;
end for
// *the cluster with the lowest cluster mean contains cluster points*
$C = \text{XMeans}(OS)$;
$O = C \setminus \min(\text{mean}(C))$; // *set of outlier objects*

Neighbors nn_x of x based on Euclidean distances (Alg. 3.1, line 8), we center, whiten, and transform the set of Nearest Neighbors resulting in a data set with independent components $nn_{x,ica}$ (Alg. 3.1, line 14f). Then a GND is fitted to the transformed data resulting in the three parameters μ_{nn_x}, σ_{nn_x}, and p_{nn_x} in Alg. 3.1. To determine the coding cost $cost_x$ as compression rate of object x we used the data description of the GND of the set of Nearest Neighbors. Subsequently, the compression rate of each object in the set of Nearest Neighbors is calculated. Having determined the compression rate of all objects in nn_x the cost of the object having the minimum compression rate given the GND model is included in the $cost\_min_x$.

Incrementally, the set of Nearest Neighbors is extended by those residual objects not_nn_x which can be optimally compressed using the present parameters μ_{nn_x}, σ_{nn_x}, and p_{nn_x} of the GND (Alg. 3.1, line 23f). Having updated the set of Nearest Neighbors, the de-mixing matrix M_{nn_x} and the mean m_{nn_x} of the Nearest Neighbors as well as μ_{nn_x}, σ_{nn_x}, and p_{nn_x} are simply adjusted to the new set of Nearest Neighbors since this is less expensive than estimating the parameters completely anew.

The outlier score of object x is calculated by subtracting the minimal overall compression rate $\min(cost\_min_{nn_x})$ from the corresponding compression rate of object x $cost_x(j)$. In the last step the obtained outlier scores are clustered by X-Means. Thereby the cluster having the minimal cluster mean includes all cluster points and the remaining objects build the set of outliers.

3.3.2 Independent Component Analysis

ICA [Com94] provides a basis for the calculation of independent components in a mixture of statistically independent random variables. Let a vector $\vec{x}$ consist of n statistically independent random variables. In order to apply ICA at most one random variable is allowed to follow a Gaussian distribution. It has been determined that the best decomposition of a mixture of

signals can be established by searching for data not following a Gaussian distribution. One reason could be that mixtures of signals from an arbitrary distribution function are always more Gaussian than the original signal. In our approach we apply ICA to maximize non-Gaussianity which is used as a measure of the statistical independence. This can be achieved since ICA favors directions of the data which are not Gaussian distributed. We determine coding costs (measured by entropy) which have to be minimized in order to guarantee a best possible compression efficiency. Since the entropy of Gaussian distributions is maximal and all other distributions have a lower entropy it is desirable to maximize non-Gaussianity.

Most real world data is distorted in the data space, hence the assumption of equally dense data distributions is not applicable in those data sets. To overcome this drawback we applied the ICA to the data. One step in the ICA algorithm is the whitening of the data leading to a de-correlation and a normalization of the data to unit variance. This transformation to the so-called white space makes it possible to implicitly handle data with diverse density.

3.3.2.1 Data Preprocessing

In general ICA needs centered data, meaning data with zero mean, as input. If this is not the case the data has to be centered. In Figure 3.1 the different steps of the ICA algorithm are illustrated. The first step is centering of the data. This can be achieved by subtracting the empirical mean $\vec{m}$ of a data set DB, in the example this is $\vec{m} = \{100, 50\}$, from each data point $\vec{x} \in DB$

$$\vec{c} = \vec{x} - \vec{m} \tag{3.1}$$

whereby the empirical mean is defined as

$$\vec{m} = \frac{\sum_{\vec{x} \in DB} \vec{x}}{n} \tag{3.2}$$

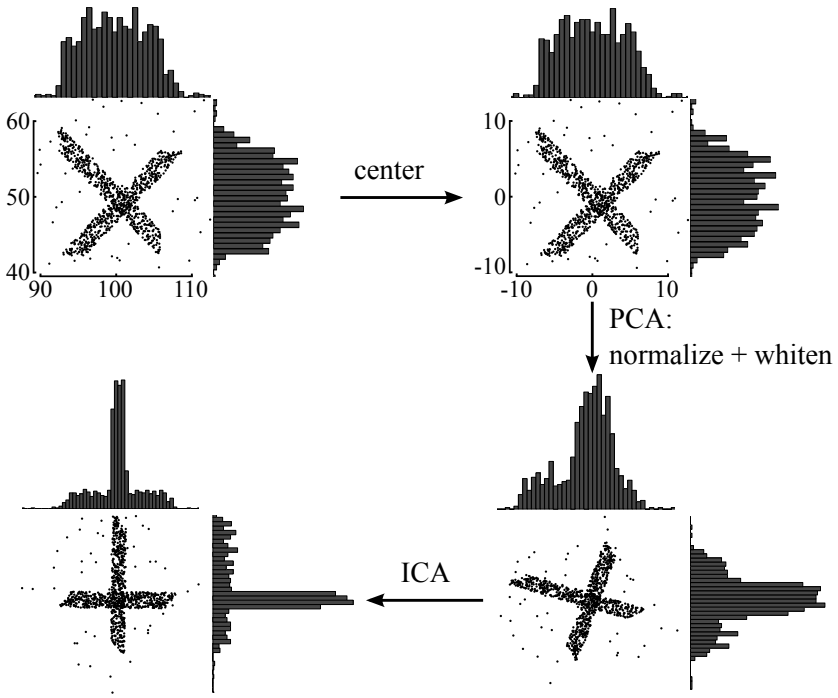


Figure 3.1: The principle of the Independent Component Analysis. Shown are the different steps from the original data until the data is transformed into independent components. First the data is centered, then the data is normalized to unit variance and whitened by PCA, and finally the data is transformed into independent components by ICA.

with $n = |DB|$ being the cardinality of the data set.

Now, since the data is centered the PCA [Ait84] which is a subpart of the ICA can be applied. PCA is used to transform the data loosing as little information as possible while combining existing redundancy in terms of correlations in the data. Thereby, given a set of centered points $\vec{c}$, PCA identifies those directions in a d-dimensional vector space which have maximal variance. For this purpose the centered data $\vec{c}$ need to be normalized to unit variance in all directions. To achieve this, first the covariance matrix Σ has

to be determined by multiplying the centered data vector with its transpose

$$\Sigma = \vec{c} \cdot \vec{c}^{\,T}. \tag{3.3}$$

Then an eigenvalue decomposition of the covariance matrix is conducted

$$\Sigma := VDV^T, \tag{3.4}$$

resulting in the Eigenvectors V and the Eigenvalues D of the covariance matrix. Both Eigenvectors and Eigenvalues are orthogonal matrices. In addition to that the Eigenvalue matrix is a diagonal matrix $D = diag(\lambda_1, ..., \lambda_d)$. The Eigenvectors build a rotation matrix and the square root of the Eigenvalues corresponds to the variance of the main components.

The PCA transform of vector $\vec{x}$ is obtained by

$$\vec{y} := \sqrt{D}^{-1} \times V^T \times \vec{c}. \tag{3.5}$$

Note, that since the Eigenvalue matrix is a diagonal matrix the inverse of the Eigenvalue matrix is simply the inverse of each diagonal entry in the matrix $\sqrt{D}^{-1} = diag(\sqrt{1/\lambda_1}, ..., \sqrt{1/\lambda_d})$. By multiplying the diagonal matrix with the variance the main components are normalized to one. The effect of normalization and whitening of a data set by PCA is depicted in Figure 3.1. The redundancy combination in terms of correlations in the data can be clearly seen.

3.3.2.2 Identification Of Independent Components

For solving the problem of finding independent components which is the major goal of the ICA we used the efficient FastICA [Hyv99, HKO01]. FastICA is based on a fixed-point iteration scheme which solves the determination of the weighting matrix $W = \{\vec{w}_1, ..., \vec{w}_d\}$ to discover the independent components of the transformed data. Until now, we determined projections with

maximal variance by PCA but since we are rather interested in the optimal projection of the data we need to determine the directions of minimal entropy which can be obtained by ICA. Since the iterative optimization of W expects whitened data as input the whitened data produced by PCA can be inserted. In order to optimize W using the fixed-point iteration of the FastICA algorithm the weight vectors $\vec{w}$ of the matrix W are updated by

$$\vec{w} = E\left(\vec{y} \cdot g(\vec{w}^T \cdot \vec{y})\right) - E\left(g'(\vec{w}^T \cdot \vec{y})\right) \cdot \vec{w}. \tag{3.6}$$

Thereby, $E(\cdots)$ is the expected value, $g(\cdots)$ is a non-linear contrast function, and $g'(\cdots)$ is the derivative of the non-linear function g. We decided to use $\tanh(a)$ for $g(a)$, resulting in $g'(a) = \frac{d\tanh(a)}{da}$. The optimization process is finished in case of convergence of W followed by the orthonormalization of W. By now the problem of determining an orthogonal weighting matrix W is reduced, but the random variables are not yet stochastically independent.

In order to project the original data $\vec{x}$ into the independent components we need to determine the de-mixing matrix M^{-1} which is composed of the Eigenvectors V and the Eigenvalues D of the covariance matrix as well as the weighting matrix W. Since the mixing matrix is

$$M = V \times \sqrt{D} \times W \tag{3.7}$$

we can obtain the de-mixing matrix M^{-1} by

$$M^{-1} = W^T \times \frac{1}{\sqrt{D}} \times V^T \tag{3.8}$$

where W and V are both orthonormal matrices. Hence, the determinant of the de-mixing matrix M^{-1} can be written as

$$det(M^{-1}) = \prod_{1 \leq i \leq d} \sqrt{\frac{1}{\lambda_i}}. \tag{3.9}$$

Note that the rotation or weighting matrix W is responsible for the rotation in the white space after the data has been whitened by the scaled Eigenvector matrix of the original data vector.

Finally, to convert the centered data $\vec{c}$, which has been obtained from the original data $\vec{x}$, into independent components $\vec{z}$ we have to project $\vec{c}$ into the independent component space by

$$\vec{z} = M^{-1} \times \vec{c}. \tag{3.10}$$

The last step of Figure 3.1 shows the impact of the transformation of the data into independent components. After ICA the redundancy in the data is minimal.

3.3.3 Generalized Normal Distribution

The Generalized Normal Distribution which is also called power exponential, exponential error, or generalized Gaussian distribution incorporates a large family of symmetric distributions, like the Gaussian, the Laplace, and the Uniform distribution depending on the parameter settings. Given a point x in a d-dimensional space the GND incorporates the following Probability Density Function [MR05]

$$f(x; \mu, \sigma, p) = \frac{1}{2\sigma p^{1/p} \Gamma(1 + 1/p)} \exp\left(-\frac{|x - \mu|^p}{p\sigma^p}\right) \tag{3.11}$$

The three parameters μ, σ, and p represent, the location, scale, and shape of the GND, respectively. Thereby, $-\infty < x < +\infty$, $-\infty < \mu < +\infty$, and σ as well as p are both larger than zero. The gamma function $\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt$ is an extension of the factorial operator for real numbers.

Note, that the parameter p is responsible for the shape of the curve. It is connected to the broadness of the tails and therefore to the kurtosis of the distribution. The GND can describe platikurtic distributions for $p > 2$

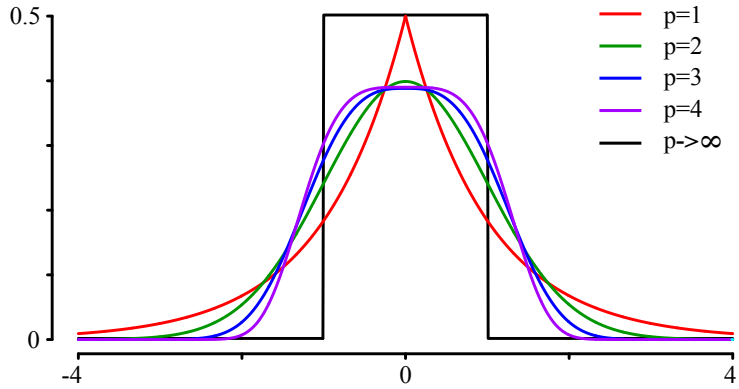


Figure 3.2: Generalized Normal Distribution with different parameter settings of the shape parameter p.

and also leptokurtic distributions $0 < p < 2$. Particularly, using a shape parameter of $p = 1$ leads to a Laplace (or double exponential) distribution (Figure 3.2, red line) and choosing the shape parameter to be 2 results in a Gaussian distribution ($\mathcal{N}(\mu, \sigma^2/2)$; Figure 3.2, green line). As the shape parameter grows towards infinity ($p \to \infty$) the GND coincides with the PDF of the Uniform distribution ($\mathcal{U}(\mu - \sigma, \mu + \sigma)$, Figure 3.2, black line).

3.3.4 ICA With GND Linkage

Now lets combine the result of the ICA with the GND function. During ICA the data was centered, whitened, and normalized to unit variance, followed by the de-correlation of the data. Hence, the resulting data $\vec{z}$ is de-correlated and independent, therefore after ICA we are able to describe each component separately by an own GND. Since we have d different dimensions, d different parameter settings for the GNDs $f(\vec{z}_i; \mu_i, \sigma_i, p_i)$ with $1 \leq i \leq d$ have to be estimated. As described in Subsection 3.3.2, the mixing matrix M which was identified by ICA combines all d PDFs and is used to de-correlate the data $\vec{x}$ by $\vec{z} = M^{-1} \times (\vec{x} - \vec{m}) = M^{-1} \times \vec{c}$, with $\vec{m}$ being the empirical mean of the data and $\vec{c}$ being the centered data. The resulting independent components

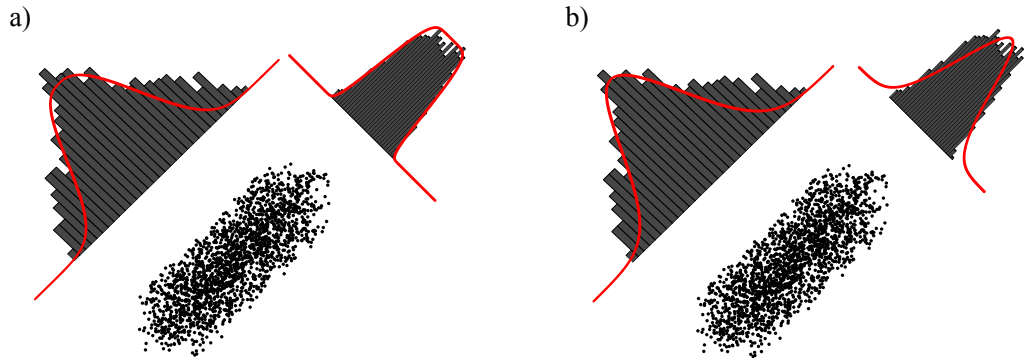


Figure 3.3: a) ICA in combination with GND approximation and b) approximation after ICA using Gaussian distributions.

are not orthogonal anymore due to the mixing matrix M. To combine the independent components with the GND we define the PDF of the GND by

$$f(\vec{z}; M^{-1}, \mu, \sigma, p) = \frac{1}{|det(M^{-1})|} \prod_{1 \leq i \leq d} f(\vec{z}_i; \mu_i, \sigma_i, p_i) \tag{3.12}$$

The advantage of incorporating the GND instead of a Gaussian or Uniform PDF can be seen in Figure 3.3. After ICA each dimension can be modeled separately illustrated by the rotated histograms. A Gaussian distribution (Figure 3.3, b) fails to approximate the different dimensions, since one dimension is uniformly distributed and the other one is close to being Gaussian. In contrast to that the shape parameter p enables the GND to model both dimensions more accurately (Figure 3.3, a).

3.3.5 GND Parameter Estimation

As mentioned earlier the GND has three parameters μ_i accounting for the location, σ_i representing the scale, and p_i which is responsible for the shape of the GND. The determination of these parameters is not a trivial task. After ICA, μ_i and σ_i are no longer identical to the empirical mean and standard

deviation. Given a data set DB, the three parameters μ_i, σ_i, and p_i can be determined by the maximum likelihood estimation method. One condition in order for the derivatives of the likelihood to dissolve with reference to μ_i, σ_i, and p_i is a concurrent estimation of all three parameters.

The log likelihood function of the GND can be formalized as [MR05]

$$l(DB; \mu_i, \sigma_i, p_i) = -n \log[2\sigma_i p_i^{1/p_i} \Gamma(1 + 1/p_i)] - \frac{\sum_{\vec{z} \in DB} |z_i - \mu_i|^{p_i}}{p_i \sigma_i^{p_i}} \quad (3.13)$$

with $n = |DB|$ being the cardinality of the data set DB.

3.3.5.1 Scale Parameter Estimation

Now, lets assume μ_i and p_i to be given, then the parameter σ_i can be computed by deriving the log-likelihood function $l(DB; \mu_i, \sigma_i, p_i)$ regarding σ_i of the GND and setting the resulting expression to zero

$$\frac{d\, l(DB; \mu_i, \sigma_i, p_i)}{d\sigma_i} = -\frac{n}{\sigma_i} + \frac{\sum_{\vec{z} \in DB} |z_i - \mu_i|^{p_i}}{\sigma_i^{p_i+1}} = 0. \quad (3.14)$$

Solving the equation for σ_i the maximum likelihood estimator of σ_i can be obtained by

$$\hat{\sigma}_i = \left(\frac{1}{n} \sum_{\vec{z} \in DB} |z_i - \mu_i|^{1/p_i} \right). \quad (3.15)$$

3.3.5.2 Location Parameter Estimation

For μ_i no explicit solution can be determined, hence we use a nested bisection search to optimize the location parameter in its parameter space. To determine the direction of browsing through the space the derivative of the log-likelihood function with respect to μ_i is used

$$\frac{d\, l(DB; \mu_i, \sigma_i, p_i)}{d\mu_i} = -\frac{1}{\sigma_i^p} \sum_{\vec{z} \in DB} \left(|z_i - \mu_i|^{p_i - 1} sign(z_i - \mu_i) \right). \quad (3.16)$$

3.3.5.3 Shape Parameter Estimation

Just like for the location parameter, no explicit solution can be determined for the shape parameter p_i either. Hence, we also use a nested bisection search for the optimization of the shape parameter in its parameter space. Thus, the derivative of the log-likelihood function with respect to p_i is used to determine the direction of browsing through the parameter space of p_i.

$$\frac{d\, l(DB;\mu_i,\sigma_i,p_i)}{dp_i} = -\frac{n}{p_i^2}\left(\log p_i + \Psi(1+\frac{1}{p_i}) - 1\right) + \tag{3.17}$$

$$+\frac{1}{p_i^2\sigma_i^{p_i}}\left(\sum_{\vec{z}\in DB}|z_i-\mu_i|^{p_i}+p\log\sigma_i\sum_{\vec{z}\in DB}|z_i-\mu_i|^{p_i}-p\sum_{\vec{z}\in DB}(|z_i-\mu_i|^{p_i}\log|z_i-\mu_i|)\right)$$

with $\Psi(\cdots)$ being the digamma function which is the logarithmic derivative of the gamma function $\Psi(a) = \frac{d\ln\Gamma(a)}{da}$. The estimation process is finished in case of convergence of p_i.

3.3.6 Coding Cost Determination

Until now we converted the original data $\vec{x}$ into independent components using ICA and estimated the three parameters of the GND for each dimension separately, in order to determine an accurate representation of the data distribution. Next, we require a quality criterion to evaluate the accuracy of the fit. For this purpose we interlink the principle of data compression with the concept of Probability Density Functions. For data compression we use the Minimum Description Length principle which exploits the fact that the stronger the data can be compressed the larger is the regularity in the signal.

To find a good coding cost representation we used Huffman coding [Huf52] which is an entropy coding. The basic idea of Huffman coding is that different characters have to be coded by a different number of bits in order to save memory. Thus, in our case each object in the data set is tagged by a certain

number of bits. The number of bits is obtained by the inverse logarithm of the probability of the object. Therefore, given an arbitrary PDF $f(\vec{x})$ the coding costs *coding_cost* of an object $\vec{x}$ can be defined using the negative log-likelihood. To represent the coding cost in number of bits the basis of the logarithm is typically 2.

$$coding\_cost(\vec{x}) = \log_2 f(\vec{x})^{-1} = -\log_2 f(\vec{x}). \tag{3.18}$$

Since we have a General Normal Distribution as PDF we can define the relative coding cost of an object $\vec{z}$ which has been transformed by ICA as

$$coding\_cost(\vec{z}; M^{-1}, \mu, \sigma, p) = -\log_2 f(\vec{z}; M^{-1}, \mu, \sigma, p) = \tag{3.19}$$

$$= \log_2(|det(M^{-1})|) - \sum_{1 \leq i \leq d} \log_2 f(\vec{z}_i; \mu_i, \sigma_i, p_i).$$

with $\vec{z} = M^{-1} \times (\vec{x} - \vec{m})$.

We disregard the selection of a grid resolution defining the storage accuracy of the points as well as the parameter costs in our coding costs since we are only interested in the comparison of different GNDs. Hence, we do not determine absolute coding cost values but rather relative ones. Since, optimal data compression is dependent on redundancy in the data it is essential to determine statistically independent major directions of the data with ICA prior of computing the coding costs. This can be clearly seen in Figure 3.1. Starting with disperse data with respect to the x- and y-axis ICA restrains redundancy in the data concerning the axes by transforming it into independent components, providing an optimal basis for data compression.

3.3.7 Outlier Score And Outlier Detection

Putting everything together, for each data object x we screen the entire data set DB by iteratively adding a set of Nearest Neighbors nn_x to x. In order

to assure a stable GND estimation the set of Nearest Neighbors is initialized with the 20 objects having the smallest Euclidean distance to x.

For each set of Nearest Neighbors nn_x the rotation, meaning the demixing matrix of the ICA, and the data description determined by the three parameters μ, σ, and p of the GND are computed. Having the ICA and the GND estimate of nn_x, the data compression rate for each object in the set of centered and whitened Nearest Neighbors of x, $nn_{x,ica}$ can be determined by the coding costs $coding\_cost(nn_{x,ica}; M^{-1}, \mu, \sigma, p)$. Since, we obtain one coding cost value for each object in $nn_{x,ica}$ but we are only interested in the most efficient compression of $nn_{x,ica}$ only the minimal coding cost determined from the set of Nearest Neighbors is kept. This compression rate information $(\min(coding\_cost(nn_{x,ica}; M^{-1}, \mu, \sigma, p)))$ is gathered for each set of Nearest Neighbors. For an optimal compression regarding x it would be necessary to know the correct number of best compressed Nearest Neighbors in advance. But since we do not have this information but we do have information of the objects' coding costs, the minimal coding cost corresponding to the best compression rate in the set of Nearest Neighbors throughout the complete set of Nearest Neighbor sets, $\min(cost\_min_{nn_x})$, represents the best possible GND estimate for any nn_x. As we are interested in the degree of x being an outlier we need to obtain an outlier score. This outlier score is the absolute compression rate increase $cost_x$ with respect to the overall minimal coding cost in the set of Nearest Neighbors throughout the complete set of Nearest Neighbor sets. Hence, the outlier score is $cost_x(j) - min(cost\_min_{nn_x})$ with j being the index of the best compressed set of Nearest Neighbors. Since the screening of Nearest Neighbors is quadratic in the number of points n and cubic in the dimensionality d due to ICA and GND, we decided to increase the size of nn_x exponentially with respect to the size of DB.

By now, outlier scores for all objects have been determined, the next step is to differentiate outlier objects from cluster objects. In general, outliers are expected to have extremely high coding costs in comparison to ordinary data points in the data set. The outlier scores of cluster points are located around

zero because cluster points can be compressed very efficiently, whereas, outliers can have arbitrary large outlier scores. Since the structure of a data set is unknown in most cases, the tagging of outliers is a non-trivial task, because the detection of an adequate threshold is very difficult for unknown data. To overcome the problem of setting this threshold parameter we applied X-Means algorithm [PM00] to the set of n outlier scores. Thereby, X-Means is a parameter-free extension of the partitioning K-Means clustering algorithm. To decide whether a cluster split is advantageous or not, X-Means uses the information theoretic BIC values of the two structures. The outlier scores are one dimensional numeric values which can be efficiently clustered by X-Means. Since all cluster points should have an outlier score close to zero and thus the cluster mean is also close to zero, the cluster having the smallest cluster mean contains the cluster objects and all other clusters contain outlier objects. We are even able to create an outlier order arranging the identified clusters in ascending order of their cluster mean. But in the majority of the tested data sets two clusters were identified by X-Means, whereby the cluster with a cluster mean close to zero contained the cluster points and the other one comprised all outliers.

In our approach we are able to detect outliers without the need of any explicit parameter settings. This can be achieved by combining ICA and GND to estimate the set of Nearest Neighbors, followed by the application of the MDL principle for data compression. Hence there is no need of knowing the number of outliers or the type of data distribution, including shape and density of the data, in advance.

3.4 Experiments

To emphasize the advantages of our approach in contrast to existing outlier detection algorithms we evaluate our outlier detection approach in comparison to two outlier detection approaches, LOF [BKNS00] and LOCI [PKGF03].

For this purpose, two data sets are used, one synthetic data set as well as data from the National Basketball Association (NBA) from the 2007/2008 season. We implemented our approach as well as LOF in Java and obtained the implementation of LOCI from the authors [PKGF03]. The synthetic data set was created to exemplify the strength of our approach. To facilitate the visual comparison and to demonstrate the potency of our approach we use a 2-dimensional synthetic data set.

3.4.1 Synthetic Data

The synthetic data set consists of four clusters, cluster 1 (C1) contains 184 points, cluster 2 (C2) is composed of 154 points, the third cluster (C3) comprises 50 points, and cluster 4 (C4) includes 52 data points shown in Figure 3.4. Each cluster has different cluster properties and a non-orthogonal major orientation. Figure 3.4 illustrates the cluster distributions of all 4 clusters after application of the ICA. As depicted cluster C1 follows a Laplace distribution after ICA in both dimensions, cluster C2 is uniformly distributed, clusters C3 is a mixture of a Uniform and a Gaussian distribution, and C4 underlies Gaussian PDFs. In addition to the cluster points, all together 26 noise points were added to the data set shown in blue.

3.4.2 Outlier Detection Results

We applied our novel outlier detection algorithm to the synthetic data set and compared the outlier results with outliers detected by LOF and LOCI. Figure 3.5 provides the results of our approach, LOF, and LOCI for the synthetic data set.

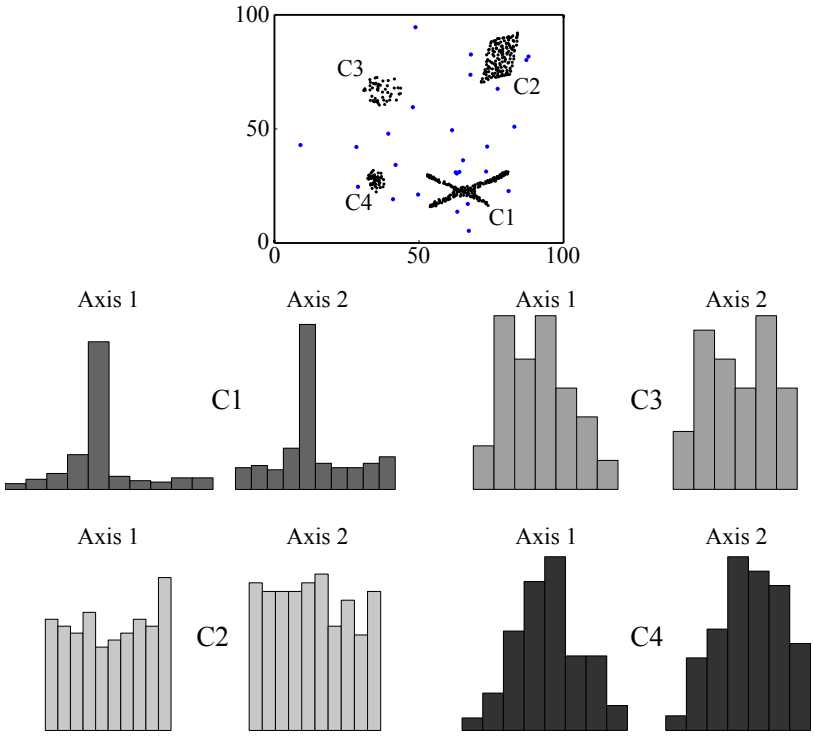


Figure 3.4: The synthetic data set including four clusters and 26 outliers shown on top. All clusters have different shapes and different data distributions. The histograms illustrate the data distributions of the four clusters after application of the ICA. Each dimension is represented by an own histogram. Cluster C1 is mainly Laplacian, C2 uniformly distributed, C3 is a mixture of a uniform and a Gaussian PDF, and C4 has a Gaussian distribution.

3.4.2.1 Outlier Detection Using MDL

We were able to correctly detect all 26 outlier points highlighted in blue (Fig. 3.5, a) with our approach. No outlier order could be detected, hence we obtained 2 clusters from the outlier score clustering by X-Means. All outliers belong to one group, the remaining group constitutes the cluster points shown in black. Note, that our approach did not require any input parameter in order to identify all noise points. It can handle different types of

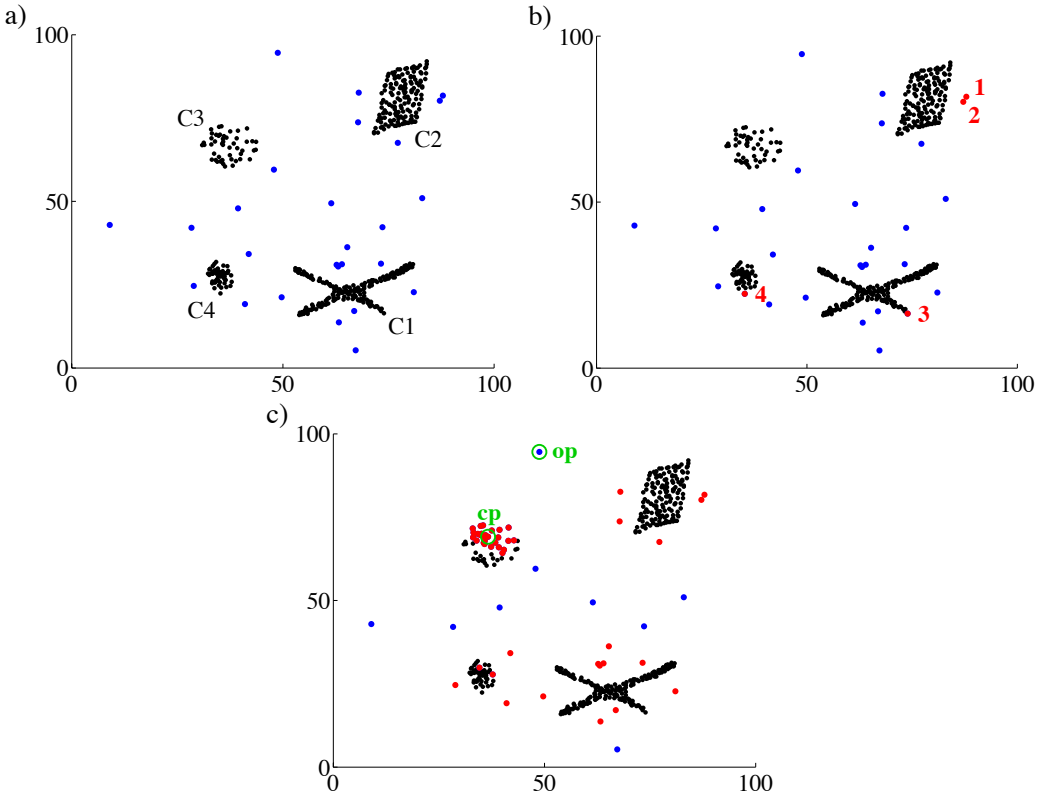


Figure 3.5: Outlier detection results from a) our approach, b) LOF ($MinPts$ = 50 selecting the top 26 outliers), and c) LOCI ($\alpha = 0.5$ and $r_{min} = 10$) for the synthetic data set consisting of four clusters (C1-4) and 26 outliers. Correctly identified outliers are shown in blue, while wrongly identified outliers are highlighted in red. In addition the four wrongly found outliers by LOF are marked by numbers to ease cross referencing in the text. The two points circled in green (cluster point *cp*, outlier point *op*) are illustrated in Figure 3.6 using LOCI plots.

cluster shapes, distributions, and orientations without expecting an explicit description of their distributions.

3.4.2.2 Local Outlier Factor

LOF was applied to identify the outliers based on a $MinPts$ neighborhood

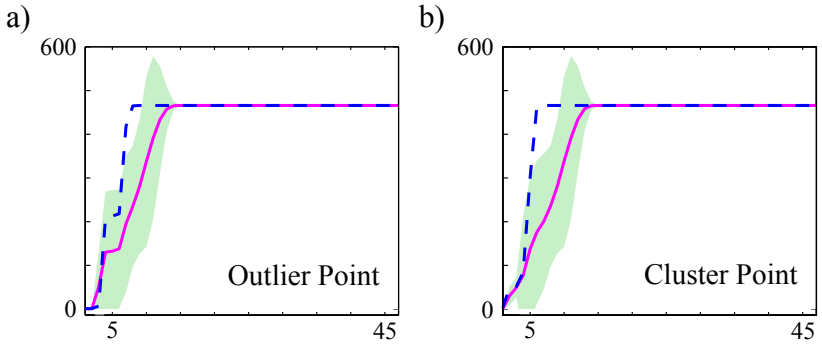


Figure 3.6: LOCI plot for two points detected as outliers. a) shows an outlier which was correctly identified as outlier by LOCI (Figure 3.5, c; labeled by *op*) and b) shows a cluster point which was falsely identified as outlier (Figure 3.5, c; labeled by *cp*).

of 50 determined by the size of the smallest cluster in the set (Fig. 3.5, b). We obtain the top 26 outliers since we know how many outliers are present in the data set. There are 24 out of the 26 noise points assigned correctly by LOF. Two noise points next to cluster C2 (red points 1. and 2.) are not detected, leading to two falsely identified cluster points as outliers (red points 3. and 4.). Note, that we collected the top 26 data points ranked by the LOF score. Setting the parameter $MinPts$ to a value smaller or equal to 10, LOF identifies more cluster points as outliers while leaving many true outliers undetected (data not shown). A $MinPts$ value of 20 to 50 leads to the same result as shown in Figure 3.5. It is important to mention, that even if LOF is able to find the majority of the outliers, for most values of the parameter $MinPts$, an approximate size of the clusters as well as an approximate number of outliers has to be known in advance in order to get a meaningful output. If we have no prior information about the number of outliers, it is only possible to determine an arbitrary number of outliers. In addition, an approximate cluster size needs to be known in advance to set $MinPts$. These assumptions make it difficult to apply LOF to real world data.

3.4.2.3 Local Correlation Integral

LOCI was applied to the synthetic data set with $\alpha = 0.5$ and $r_{min} = 10$ (Fig. 3.5, c) and could identify 43 outlier points based on the suggested outlier flagging criteria. Altogether 17 true outliers are missed, while two points from within cluster C3 and 27 points from cluster C4 are labeled as outliers. Different parameter settings of r_{min} may detect more true outliers, but at the same time label more cluster points as outliers. Obviously, LOCI is not able to deal with clusters showing low density, like C4. In Figure 3.6, we have a closer look at the LOCI plot of an outlier point (circled in green as *op* in Figure 3.5, c) and a cluster point (*cp*). The LOCI plots look very similar even though they are supposed to emphasize the difference between a cluster point and an outlier. We have to note, that although we applied the algorithm with the suggested parameter settings, the result was difficult to interpret even after correspondence with the authors.

3.4.3 Outlier Score Visualization

To emphasize the difference and strength of our outlier score in comparison to the LOF score, we introduce a visualization of the "outlierness" (Fig. 3.7). A scatter plot of the data in x-y directions is combined with a bar representation of the outlier factors in the z-dimension. We can clearly show that the utilization of data compression is able to separate the outliers from the cluster points in comparison to the outlier factor of LOF. The majority of the cluster points have an outlier score which is located close to 0.0 which can be seen by the short, dark blue bars in Figure 3.7 a. Outliers are either light blue or even red indicating their extraordinariness, ranging from 6.4 up to 24.2. Due to the large range between cluster points and outliers it is possible to clearly differentiate them using X-Means in our approach.

In contrast, LOF produces values ranging from 0.8 up to 2.3 which makes it almost impossible to clearly differentiate cluster points from outliers ex-

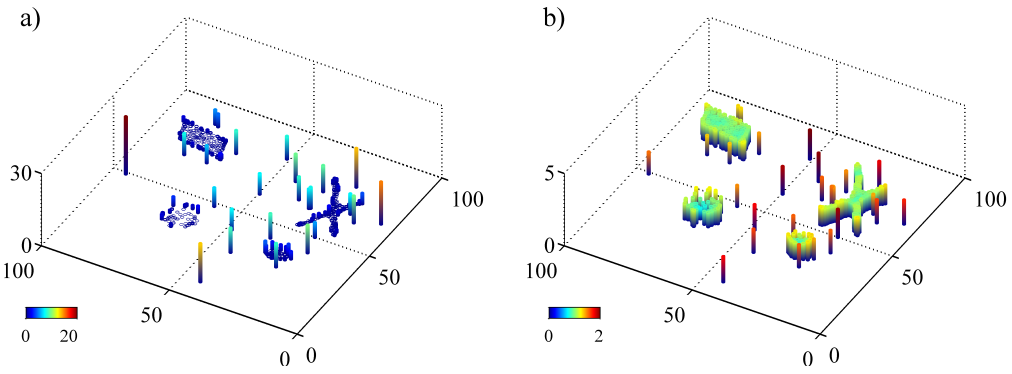


Figure 3.7: Outlier cost of our approach (a) and outlier-factor of LOF (b) for the synthetic data set. Our outlier score (a) and the local outlier factor (b) are shown on the z-axis indicated also by the different coloring of the bars. If a point has a short, dark blue colored bar it has a low score and is a cluster point, if the bar is tall and red it is an outlier.

plicitly. The visualization of the outlier factors of LOF demonstrates, that the cluster structure is based on Euclidean distances: the outlier factors continuously increase circular from the cluster centers to the cluster margins. In contrast to LOF, our outlier scores are equally low throughout the entire cluster except for the cluster edge points. It is based on the flexible cluster structure description using ICA and EPD.

3.4.4 Experimental Data

After extensive evaluation of our approach on synthetic data sets, we want to apply our novel parameter-free outlier detection method to experimental data. We used the National Basketball Association (NBA) data available at the NBA website `http://www.nba.com`. In Season 2007/08, 450 players are described with four attributes: the number of games played (GP), and the number of points (PPG), rebounds (RPG), and assists (APG) per game. Our approach was applied to this NBA data detecting 105 outliers. Figure 3.8

Outlier Score	Name	GP	PPG	RPG	APG
19.6	Stephon Marbury	24	13.9	2.5	4.7
17.9	Jamaal Tinsley	39	11.9	3.6	8.4
16.1	Gilbert Arenas	13	19.4	3.9	5.1
15.4	Andrew Bynum	35	13.1	10.2	1.7
13.6	Elton Brand	8	17.6	8	2
12.9	Ronald Muray	73	9.1	4.5	1.3
12.8	Jason Kidd	80	10.8	7.5	10.1
12.5	Chris Kaman	56	15.7	12.7	1.9
12.3	Ramon Sessions	17	8.1	3.4	7.5
12.0	Randy Foye	39	13.1	3.3	4.2

Table 3.1: Top 10 outliers identified with our approach on NBA data.

displays scatter plots of the NBA data. For simplicity reasons, we highlight only the top 10 outliers in red as listed in Table 3.1. Obviously, the data distribution is non-Gaussian.

The top 10 outliers identified by our approach, include outstanding players like Stephon Marbury with a coding cost of 19.6 being 12 times higher than the average coding costs. Marbury is an outstanding player with respect to all attributes. He played only 24 games out of 82 and was still able to achieve 13.9 points and additionally assisted in 4.7 points, resulting in being involved in 18.6 points per game. Jamaal Tinsley, has played 39 games in this season but was still able to assist in 8.4 game points. He was involved in 20.3 points and played more games than Marbury. Gilbert Arenas exhibits a rare combination of playing 13 games while achieving 19.4 points per game. Jason Kidd is outstanding in the number of rebounds having played in 80 out of 82 games. Elton Brand has played only few games but was still able to achieve an outstanding number of points. As evident from Figure 3.8, outstanding players such as Kidd or Brand are best characterized with the most general model with only one component.

To put our outlier detection method into a context, we applied LOF and LOCI to the NBA data set, as well. Table 3.2 displays the top 10 outliers identified by LOF. Highlighted in bold are all players that were identified as

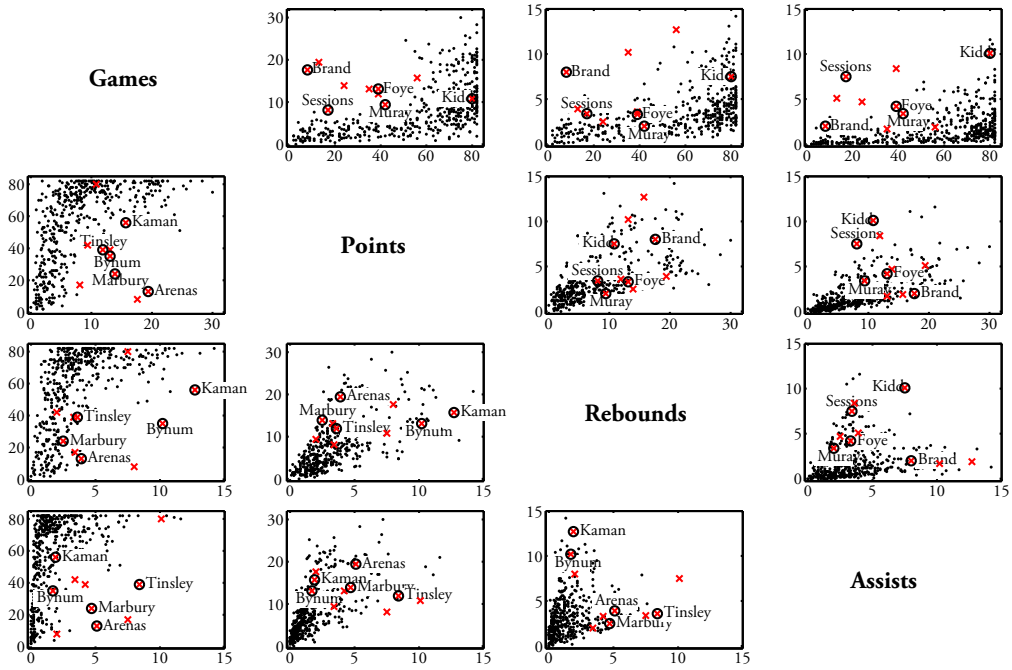


Figure 3.8: NBA data of the 2007/2008 season. Shown with red crosses are the top 10 outliers identified with our approach. For clarity reasons 5 outliers are labeled and marked with circles in the upper triangle and the remaining 5 outliers are labeled and marked in the lower triangle.

top 10 outliers with our approach, like Marbury, Arenas, or Brand. Except for one, all players from the set of top 10 outliers of our approach are at least under the top 20 of LOF. However, the outstanding player Kidd was missed by LOF ranked at the 50th position with a LOF score of 1.16. In addition, as observed for synthetic data, the result of LOF strongly depends on its parameterization. Only seven players are reproducibly detected as top 10 for a $MinPts = 40$ (players are marked with an asterisk). All five players which were included in the Top 10 of our approach were also included in the intersect of $MinPts = 40$ and $MinPts = 50$ which strikes that they are strongly outstanding. The top 10 outliers found by LOCI are shown in Table 3.3. The intersect between LOCI and our approach is again highlighted.

LOF	Name	GP	PPG	RPG	APG
1.43	**Elton Brand***	8	17.6	8	2
1.32	Steve Francis	10	5.5	2.3	3
1.31	Kasib Powell	11	7.6	4	1.6
1.28	**Gilbert Arenas***	13	19.4	3.9	5.1
1.28	Chris Webber*	9	3.9	3.6	2
1.27	**Stephon Marbury***	24	13.9	2.5	4.7
1.26	Dwyane Wade*	51	24.6	4.2	6.9
1.25	LeBron James	75	30	7.9	7.2
1.24	**Andrew Bynum***	35	13.1	10.2	1.7
1.24	**Chris Kaman***	56	15.7	12.7	1.9

Table 3.2: Top 10 outliers identified by LOF with $MinPts = 50$ on NBA data sorted by the Local Outlier Factor. Players also among the top 10 of our approach are marked in bold font. The asterisk indicates players which are also among the top 10 using $MinPts = 40$. Note that all players found to be in the Top 10 of our approach and LOF using $MinPts = 50$ are also found using LOF and $MinPts = 40$.

Name	GP	PPG	RPG	APG
LeBron James	75	30	7.9	7.2
Kobe Bryant	82	28.3	6.3	5.4
Dwyane Wade	51	24.6	4.2	6.9
Chris Kaman	56	15.7	12.7	1.9
Elton Brand	8	17.6	8	2
Andrew Bynum	35	13.1	10.2	1.7
Jamaal Tinsley	39	11.9	3.6	8.4
Mike Bibby	48	13.9	3.3	6
Jermaine O'Neal	42	13.6	6.7	2.2
Udonis Haslem	49	12	9	1.4

Table 3.3: Top 10 outliers identified by LOCI on NBA data. Players also among the top 10 of our approach are marked in bold font.

3.5 Conclusion

Outlier detection is an important research area for detecting outstanding observations in data sets or to remove noise prior to data analysis. Here a complete parameter-free outlier detection approach was presented. Starting

with converting the data into independent components we used a Generalized Normal Distribution which is a generalization of the Gaussian distribution to describe each independent component separately. To determine the degree of "outlierness" of an object we used data compression which is very intuitive to interpret and does not need any user defined input parameters.

Our approach includes three desirable properties. First of all it is parameter-free which simplifies the handling of the algorithm since no complicated parameters like the number of outliers present in the data have to be known in advance. Second, the utilization of the ICA prior to approximating the data enables us to describe even non-orthogonal data very accurate. Combining the independent components produced by the ICA with the flexible GND model which comprehends the Uniform, the Gaussian, and the Laplace distribution for describing the data leads to a wide applicability of the algorithm. Third, we incorporated the MDL principle which relates learning and data compression for learning regularities in the data in order to achieve a natural balance between goodness of fit and model complexity.

Many approaches explicitly or implicitly assume Gaussian distributions or do not provide a model of the data which is essential in many applications, including selectivity estimation, indexing, and classification. As shown by our experimental section our approach does not depend on any specific data distribution like Gaussian or Uniform data due to the general data model of the GND. Therefore, our method is applicable to a variety of different real world data sets.

Chapter 4

Similarity Search In Uncertain Data

4.1 Introduction

In several application fields e.g. biometric identification, sensor networks, medical imaging, or video retrieval, data can contain a certain amount of uncertainty. Uncertainty can arise by measurement error or if single objects are represented by a huge amount of values. Conventionally available database systems are not able to handle these uncertain data objects since they can only model certain data associated with a feature vector. To overcome this shortcoming a considerable amount of research has been conducted in the field of uncertain data handling.

An uncertain object is not represented by an exact position like a certain object but it is rather assigned a Probability Density Function. This data representation enables the object to imply an infinite amount of values. Thereby, objects can be defined by different PDFs, like Uniform, Gaussian, or other PDFs. A very general distribution function for representing uncertain objects is the Gaussian Mixture Model (GMM). A GMM is a probabilistic

model which consists of a mixture of several weighted Gaussian distributions. A GMM combines m Gaussian distributions by a weighted sum, with m being the cardinality of the model. Each of the m Gaussian components of a GMM can be modeled by three parameters: a weight w, a d-dimensional location vector μ, and a $d \times d$ dimensional covariance matrix Σ. In Figure 4.2 a one dimensional (left) and a two dimensional (right) example of a GMM are depicted, both consisting of four Gaussian components. Each Gaussian component comprehends a weight, a mean vector, and a covariance matrix. In some cases GMMs are restricted to model solely axis-parallel Gaussian distributions due to efficiency reasons and to reduce the complexity of the data. In other words, in those cases correlations between different features like those shown on the left in Figure 4.2 (ellipsoids build angles close to 45°) are ignored leading to a loss of precision. Including those correlations in the data model would lead to a more general representation of the respective data leading to a higher accuracy in finding the most similar objects to a given uncertain query object.

In addition to a correct representation of the uncertain data, it is also important to provide efficient query processing when dealing with highly complex uncertain data. To accelerate the processing of certain queries several indexing structure techniques have been proposed e.g. the R-tree [Gut84], the X-tree [BKK96], or the iDistance [JOT+05]. For low dimensional data these indexing techniques arrange the data hierarchically. Hence, query processing requires only logarithmic time complexity, but for moderate and high dimensional data efficiency decreases. This led to the invention of advanced indexing methods like e.g. the VA-file [WSB98]. The VA-file accelerates the sequential scan of the data using a lax data compression. Thereby, the certain data points are approximated in a conservative and tight way. For low dimensional data the VA-file is as efficient as the hierarchical structures but in higher dimensional data especially in large data sets it is often superior to tree-like indexing structures.

Since uncertain data is even more complex than certain data leading to

high computation time, indexing structures are essential in order to be efficiently handled. For this purpose tree-like methods like the U-tree [TCX+05] or the Gauss-tree [BPS06] have been proposed to accelerate the handling of uncertain data. However, these structures handle only axis-parallel PDFs disregarding possible correlation between different features. This leads to false dismissals when answering queries on general GMMs. To the best of our knowledge until now no indexing method has been introduced which is able to process exact GMM representations including correlations between attributes. Furthermore, no method has yet been proposed for speeding up the sequential scan of uncertain data, similar to the VA-file for certain data.

In our proposed similarity search method we try to overcome these drawbacks by using a conservative approximation approach for accelerating the similarity search on general Gaussian Mixture Models. Our approach does not only use GMMs but also considers correlations between different features, hence being able to handle so-called non-axis parallel GMMs. Thereby, the similarity between a database object and a query object is calculated by computing the probability of an uncertain database object to be randomly drawn from the PDF of the query object. Thus, for object comparison we use relative matching probabilities, being a probabilistic equivalent of the similarity measure for certain data objects. To accelerate the very expensive matching probabilities between non-axis parallel GMMs we introduce a conservative approximation method which guarantees no false dismissals, since the approximated probability density values are an over-estimation of the absolute matching probabilities. To minimize the approximation error which arises by approximating Gaussian distributions with strongly correlated attributes we introduce a clustering procedure which clusters those Gaussian components having similar orientation in space. Thus, Gaussian distributions implying strong correlations are grouped together leading to a more accurate approximation. This clustering method particularly supports the tightness of our conservative approximation. The tightness is responsible for the filtering of a large amount of candidates filtered that are dissimilar to the

uncertain query object. Thus only very few possible candidate objects pass the filtering step being transformed to the computationally expensive refinement step. Therefore, our filter-refinement architecture leads to a substantial reduction in runtime. The major contributions of our approach are:

- *General uncertain data representation:* Uncertain database objects and uncertain query objects are represented by so-called non-axis parallel GMMs considering correlations between features.
- *Cluster similarly oriented Gaussian components:* To reduce the approximation error Gaussian distributions with comparable main orientation in space are clustered using a circular partitioning clustering approach.
- *Filter-refinement architecture:* The combination of our conservative and tight approximation technique filtering a large amount of candidates and the expensive refinement step which only has to calculate few absolute probabilities ensures no false dismissals, a good filter selectivity, and a large runtime reduction.

In the following we will start by giving a few motivating examples in Section 4.2, followed by briefly overviewing related work in the field of uncertain data and similarity search in Section 4.3. In Section 4.4 we will introduce our filter-refinement architecture. Starting with an overview of our algorithm we will then explain the representation of uncertain objects by non-axis parallel GMM and their conservative approximation. For improving the approximation of Gaussian distributions with a similar major orientation in space the circular clustering is described. Then, the actual similarity search including the filtering step and the refinement step are described. A detailed and elaborated experimental evaluation is given in Section 4.5, including several synthetic and real world data sets. Thereby, we perform parameter benchmarking and performance measures. In Section 4.6 we conclude this Chapter.

4.2 Motivation

Since the attacks on September 11^{th}, 2001, the idea of more effective security technology arose. Biometric identification implies methods for uniquely recognizing humans based upon one or more intrinsic physical or behavioral traits. One area of biometric identification is face recognition where uncertainty arises from limitations of feature measurements like eye distance, mouth corner distance, or length, width, and curvature of the nose. These uncertainties among different features may well be correlated due to the fact that the aspect of the camera influences some of the feature measurements in the same way and, therefore, it is complex enough to require non-axis parallel GMMs. It is obvious that we want to find distributions that considerably overlap with the query distribution, not those having a small Euclidean distance. Another area in biometrics is speaker identification. Thereby, short-time segments of speech are used to identify a suspicious subject. Auditory signals are decomposed into Mel-Frequency Cepstral Coefficients (MFCC) which are then used to build GMMs [RR95]. These models can have full covariance matrices because some features are statistically dependent.

For sensor networks like for example weather data including measurements like temperature, humidity, barometric pressure, etc. the exact time period is crucial to the accuracy, as well as the technical equipment. Furthermore, additional complexity is added by the dependencies between the obtained measurements like e.g. an increase in barometric pressure and humidity results in most mesothermal climate zones in increasing temperature. In many application fields the dependency of these weather criteria is not considered at all or only represented in a very imprecise way. With the use of non-axis parallel GMMs the imprecise weather data can be modeled more accurately. To find the most alike date in a weather data history given a weather phenomena of one day is of high importance e.g. to predict floods, hurricanes, or other weather catastrophes.

In medical imaging neuroradiologists use medical images as a “second

opinion" in both detecting tumors and making diagnostic decisions. Extracting and analyzing the characteristics of benign and malignant tumors produced by different Magnetic Resonance Imaging techniques like e.g. perfusion or diffusion weighted Magnetic Resonance Imaging can aid neuroradiologists in their diagnosis and, therefore, save lives. Several perfusion features like Cerebral Blood Volume (CBV), Cerebral Blood Flow (CBF), and Mean Transit Time (MTT) can be extracted from the tumor region and fitted by a non-axis parallel GMM. Since the CBF is highly dependent on the CBV and the MTT, non-axis parallel GMM would be a perfect choice to represent the tumor data. Building a database which contains perfusion GMMs of affirmed tumors with different grades could then aid neuroradiologists in grading tumors of new patients.

4.3 Related Work

In many data mining and information retrieval systems the identification of objects which are similar to a given query object is of great relevance. The most commonly used method to compare diverging feature vectors is to use a distance metric measure like e.g. the Euclidean distance. Since the importance of some features can be larger compared to others, weighted Euclidean queries or general ellipsoid queries can be utilized instead of ordinary metric distance measures. Since efficiency is always an important aspect in information retrieval several different indexing structures for accelerating similarity queries in high-dimensional feature spaces have been proposed (for a survey see [BBK01]). However, individual object uncertainty is a problem these approaches cannot adequately deal with.

Several applications like e.g. biometric identification, sensor networks, or medical image data include uncertainty. Hence, the different features of the objects cannot be determined exactly but with some degree of insecurity which can arise from measurement errors in sensor networks. Usually

uncertain objects are modeled using Probability Density Functions, while a large amount of queries specifically k-Nearest Neighbor and range queries have been extended to probabilistic cases. As for the certain data also uncertain data has to be processed efficiently, hence several research groups have proposed indexing and query processing methods for uncertain data [DS05, DSBHW06, CKP03, CXP+04, TCX+05, BPS06].

Uncertain data can be handled in different ways. One way to is to use probabilistic queries which have lately attained increasing attention. The model proposed in [CKP03] can deal with different types of queries which enable the processing of uncertain data. Each object is assigned a feature value interval, comprising the exact values. Thereby, the feature value is defined by a specific PDF modeling the distribution of the values. This PDF is located in the aforementioned interval. Another group [CXP+04] introduced methods which were based on the R-tree [Gut84] to productively handle probabilistic threshold queries. Thereby, if the probability of a data object being located in the query interval is larger than the assigned probability threshold value of the probabilistic threshold query it is returned by the algorithm as hit. The Top-K Probability Nearest Neighbor query introduced by Bescales *et al* [BSI08] intends to identify those database objects which have the highest probability of being a Nearest Neighbor to the given query object. The applied general uncertainty model can handle uncertain database objects and uncertain query objects. But the model only regards the PDF as black box without actually employing the information contained in the probability distribution itself. In a related approach the uncertainty of the objects was described by an uncertainty region which was connected to a PDF [TCX+05]. To efficiently handle those uncertainty regions the authors introduce the U-tree which is a hierarchical multi-dimensional index structure on uncertain data. A method also considering the PDF distribution information was proposed in [BPS06]. They used Gaussian distributions to model uncertain objects. Thereby those objects which are most likely represented by the same object as the query object are returned by the probabilistic

query. This approach can be used to find those images in a given database of facial images showing a suspect portrayed on the query image. The Gauss-tree which is included in the approach to efficiently handle queries is again a hierarchical index structure. The minimal bounding rectangle thereby enclosed those Gaussian distributions possessing similar mean and variance. The disadvantage of the proposed approach is that it can only handle single axis-parallel Gaussian distributions assuming that the underlying data follows a Gaussian distribution. For querying a database of uncertain video clip objects Böhm *et al.* [BGK+07] introduced a video retrieval system. Just like the Gauss-tree this model also used axis-parallel PDFs to represent the data objects (video clips in a color histogram space), but instead of Gaussian distributions they use Gaussian Mixture Models. All mentioned tree-like structure (Gauss-tree, U-tree) as well as most other index structures for efficient uncertain data management are only able to process uncertain objects with non-correlated feature vectors, hence resulting in constrained overall accuracy.

4.4 Searching Uncertain Data Using GMMs

This chapter will start up with the description of our algorithm. In the following Gaussian Mixture Models will be introduced with a focus on the non-axis parallel version of Gaussian Mixture Models (nGMM). Since our approach also considers the correlations between features, the determination of absolute similarity probabilities between a query object and the database objects is very time consuming and highly complex. To accelerate the similarity search while maintaining the same accuracy as the extensive search we introduce a filter-refinement architecture using conservative and tight approximations in combination with a clustering procedure to achieve a more accurate approximation of highly correlated features.

// Each object comprising the three parameters w, μ, Σ
Given: Database DB, Query object $\mathcal{G}'$, Number of hits k

// Preprocess (rotate and approximate) DB and obtain rotation angle representatives
(DB_{ra}, θ_{uv}) = preprocessDB(DB); *// Alg. 4.1*

// Identify k-Most Likely Identification Queries and their absolute probabilities using the four-step procedure
(k-MLIQ, k-P) = identifyMLIQs($k, DB, \mathcal{G}', DB_{ra}, \theta_{uv}$); *// Alg. 4.3*

Figure 4.1: Overview of the different steps of our algorithm. The preprocessing has to be execute only once. As soon as a rotated and approximated database has been generated several k-MLIQs can be executed.

4.4.1 Algorithm

As input the algorithm needs a database DB of n uncertain objects as well as an uncertain query object. The database objects $\mathcal{G}^*$ and the query object $\mathcal{G}'$ are all non-axis parallel GMMs meaning they can contain correlations between different features. In addition the algorithm needs the parameter k being the number of most similar objects returned by the algorithm. An overview of the main steps of our algorithm including the preprocessing and the k-Most Likely Identification Query (k-MLIQ) search is depicted in the Figure 4.1. Before being able to search for the most similar objects to a given query object the database has to be prepared (Alg. 4.1). Note, that this procedure has to be executed only once for a given database.

The preprocessing of the data in the database as formalized in Alg. 4.1 is done separately for each Gaussian distribution, hence each of the m Gaussian distributions of the n GMMs in the database has to be processed separately following the same workflow. We start the preprocessing by clustering all Gaussian distributions according to their main orientation in space followed by the approximation of each Gaussian distribution. The clustering has to be done to minimize the approximation error, evolving from strongly correlated

Algorithm 4.1 preprocessDB

Input: Database DB
//Rotate and approximate DB and obtain rotation angle representatives θ_{uv}
Output: DB_{ra}, θ_{uv}

$l = (d^2 - d)/2$; *//Number of rotation angle dimensions*
$\theta = []$; *// Set of* $n \cdot m$ *l-dimensional rotation angles*
//The procedure is done for each Gaussian distribution separately
for all $\mathcal{G}_i^* \in \mathcal{G}^* \in DB$ **do**
  $\Sigma_{i,tmp}^* = \Sigma_i^*$; *//covariance matrix of Gaussian distribution* $\mathcal{G}_i^*$
  $\theta_{\mathcal{G}_i^*} = []$; *// l rotation angles of Gaussian distribution* $\mathcal{G}_i^*$
  for all l coordinate axis pairs x_i, x_j **do**
    //Get largest Eigenvector of covariance matrix $\Sigma_{i,tmp}^*$
    $V_{max} = \text{EigenvalueDecomposition } (\Sigma_{i,tmp}^*)$;
    $\theta_v = \text{shift}(\arctan2(V_{max}(x_i, x_j)))$; *// Eqs. 4.27, 4.29*
    $\theta_{\mathcal{G}_i^*} = \theta_{\mathcal{G}_i^*} \cup \theta_v$;
    $\Sigma_{i,tmp}^* = R(\theta_v) \cdot \Sigma_{i,tmp}^* \cdot R(\theta_v)^T$;
  end for
  $\theta = \theta \cup \theta_{\mathcal{G}_i^*}$;
end for
//Obtain c l-dimensional cluster representatives
$\theta_{uv} = X\text{-Means } (\theta)$;

//Rotate and approximate all $n \cdot m$ *Gaussian distributions in the database*
$DB_{ra} = []$;
for all $\mathcal{G}_i^* \in \mathcal{G}^* \in DB$ **do**
  //Rotate and approximate $\mathcal{G}_i^*$ using its cluster representative $\theta_{uv,\mathcal{G}_i^*}$
  $\mathcal{G}_{i,ra}^* = \text{prepareGaussian}(\theta_{uv,\mathcal{G}_i^*}, \mathcal{G}_i^*)$; *//Alg. 4.2*
  $DB_{ra} = DB_{ra} \cup \mathcal{G}_{i,ra}^*$;
end for

features. Firstly, after having obtained the Eigenvector V_{max} of the largest Eigenvalue generated by the Eigenvalue decomposition of the covariance matrix Σ_i^* of a Gaussian distribution $\mathcal{G}_i^*$, one rotation angle at a time is obtained followed by a rotation of the covariance matrix. The rotation is executed by

Algorithm 4.2 prepareGaussian

Input: Angle representative θ_{uv}, Gaussian distribution $\mathcal{G}_i$
Output: $\mathcal{G}_{ra}$ *//Rotated and approximated Gaussian distribution*

for all θ_{uv} **do**
//Rotation matrix of the jth rotation angle
$\mu_{i,r} = \text{rotateMean}(\mu_i, \theta_{uv})$; *// Eq. 4.32*
$\Sigma_{i,r} = \text{rotateSigma}(\Sigma_i, \theta_{uv})$; *// Eq. 4.33*
end for
$\chi_{i,r} = \text{approxSigma}(\Sigma_{i,r})$; *// Eq. 4.23*
$\psi_{i,r} = \text{approxW}(w_i, \chi_{i,r}, \Sigma_{i,r})$; *// Eq. 4.22*
$\mathcal{G}_{ra} = \psi_{i,r} \cdot N(x_r; \mu_{i,r}, \chi_{i,r})$;

inserting the currently calculated rotation angle in the corresponding Givens rotation matrix. Secondly, all $m \cdot n$ l-dimensional rotation angles are then clustered using an X-Means algorithm which has been adjusted to cyclic distance values. Thirdly, all Gaussian distributions are first rotated using the currently calculated cluster representatives to be as axis-parallel as possible in the new coordinate system followed by the approximation of the Gaussian distribution (Alg. 4.2). Finally, these rotated and approximated Gaussian distributions are stored in a separate database DB_{ra}.

This newly generated database DB_{ra} in combination with the original database DB can then be used as input for the actual k-MLIQ search (Alg. 4.3). The k-MLIQ search is a four-step procedure. In the first step the query object has to be converted to be able to compare the query object with the rotated and approximated objects of the database DB_{ra}. Contrary to the database object the Gaussian distributions have to be rotated into all coordinate systems followed by the approximation to calculate approximated Probability Density Values (PDV_a) with a preferably small approximation error. The actual rotation and approximation is done the same way as for the database objects using Alg. 4.2. Now the database objects and the query object are preprocessed and the actual k-MLIQ search can start (Alg. 4.3

Algorithm 4.3 identifyMLIQs

Input: Number of MLIQ k, Database DB, Preprocessed database DB_{ra}, Query $\mathcal{G}'$, Rotation angle cluster representatives θ_{uv}
Output: k-MLIQ; k-P;

//Step 1) Rotate and approximate all Gaussian distributions of query object $\mathcal{G}'$ in all rotation angle clusters θ_{uv} leading to c representations of $\mathcal{G}'$
$\mathcal{G}'_{ra} = []$;
for all $\mathcal{G}'_i \in \mathcal{G}'$ **do**
  $\mathcal{G}'_{ra} = \mathcal{G}'_{ra} \cup$ prepareGaussian$(\theta_{uv}, \mathcal{G}'_i)$; *//Alg. 4.2*
end for

//Step 2) Calculate approximated PDVs
$PDV_a = []$; *//Probability density values of all objects in the database*
for all $\mathcal{G}^*_{ra} \in DB_{ra}$ **do**
  $PDV_a = PDV_a \cup PDV_a(\mathcal{G}^*_{ra}, \mathcal{G}'_{ra})$; *//Eq. 4.44*
end for
//Sort approximated PDVs in descending order
sort(PDV_a);

//Step 3) Obtain k largest PDVs
k-MLIQ $= []$; *//k-most likely identification queries*
k-PDV $= []$; *//Probability density values of k most likely identification queries*
$PDV_{ub} = -\infty$; *//PDV upper bound*
while $PDV_{ub} < getFirst(PDV_a)$ **do**
  //$\mathcal{G}^*_1$ being the object on the first position of PDV_a with largest PDV_a
  $PDV = PDV(\mathcal{G}^*_1, \mathcal{G}')$; *//Eq. 4.37*
  $indexI = \min(k\text{-PDV})$;
  if $PDV_{ub} < PDV$ **then**
    k-PDV $= \{k\text{-PDV} \setminus k\text{-PDV}(indexI)\} \cup PDV$;
    k-MLIQ $= \{k\text{-MLIQ} \setminus k\text{-MLIQ}(indexI)\} \cup \mathcal{G}^*$;
  end if
  $PDV_{ub} = \min(k\text{-PDV})$;
  $PDV_a = PDV_a \setminus getFirst(PDV_a)$
end while

//Step 4) Calculate absolute probabilities of k-MLIQ
k-P $= []$;
for all $\mathcal{G}^* \in k$-MLIQ **do**
  k-P$= P(\mathcal{G}^*, \mathcal{G}')$; *//Eq. 4.45*
end for

Step 2-4). The search starts by calculating and sorting all approximated PDV_a values between the preprocessed query object $\mathcal{G}'_{ra}$ and each preprocessed database object $\mathcal{G}^*_{ra}$ (Step 2). Then, subsequently the Probability Density Value (PDV) between the original database object $\mathcal{G}^*_1$ momentarily having the largest PDV_a and the original query object is calculated, until a PDV_a having a smaller value then the upper bound PDV_{ub} is found (Step 3). Thereby, the present object is added to the list of k-MLIQ if and only if it has a PDV which is larger than the smallest PDV in the list of k-MLIQs. The upper bound PDV_{ub} is updated to the smallest PDV in the list of k-MLIQs. In the last step (Step 4) absolute probabilities (k-P) for the k objects being most similar to the query object are computed and the algorithm stops.

Using the four-step procedure we can keep an accuracy of 100 % while saving computation time due to the lesser absolute probability computations. A detailed formalization of the single parts of the algorithm are given in the following sections.

4.4.2 Non-axis Parallel Gaussian Mixture Model

Mixture Models are a specific type of Probability Density Functions which comprehend a certain amount of component functions. In the case of Gaussian Mixture Models the component functions are Gaussians. The combination of these Gaussian component functions build a multimodal density. In contrast to non-parametric histograms, Gaussian Mixture Models offer a higher flexibility and are more precise when modeling complex data distributions. For uncertain data, where values are not exactly known, Gaussian Mixture Models can provide a very accurate estimation of the uncertain objects.

The general form of a Gaussian distribution for a single variable x can be formalized as

$$N(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4.1}$$

where μ corresponds to the mean and σ to the standard deviation of the distribution. Hence, $N(x;\mu,\sigma)$ indicates the density function characterizing the measurements of the variable x.

A Gaussian distribution in a d-dimensional space is defined by two parameters, the location parameter μ and the covariance matrix Σ. The location parameter is a d-dimensional vector $\mu = (\mu_1, \cdots, \mu_d)^T$ and the covariance matrix is a quadratic, positive-definite, and symmetric $d \times d$ dimensional matrix which contains the covariances of the random vector. The Gaussian distribution function of one d-dimensional Gaussian can be defined as [DMJRM00]:

$$N(x;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\left(-\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{2}\right) \tag{4.2}$$

thereby $|\Sigma|$ corresponds to the determinant of Σ and Σ^{-1} denotes the matrix inverse of Σ. Note, that we consider the entire covariance matrix not only the diagonal of the matrix containing the variances. Hence, also feature correlations are included in our model leading to non-axis parallel Gaussians.

As mentioned, a GMM consists of several Gaussian distributions which build one non-axis parallel GMM $\mathcal{G}$. Let $\mathcal{G}$ comprise $m = |\mathcal{G}|$ different Gaussian distributions, m being the cardinality of the model. Thereby, each Gaussian component is additionally assigned by a weight w and the weights of a GMM sum up to identity

$$\sum_{\mathcal{G}} w = 1. \tag{4.3}$$

Let furthermore, each Gaussian PDF be modeled by μ and Σ then the complete Probability Density Function of a GMM $\mathcal{G}$ can be defined as

$$f_{\mathcal{G}}(x;w,\mu,\Sigma) = \sum_{i=1}^{m} (w_i \cdot N(x;\mu_i,\Sigma_i)) = \tag{4.4}$$

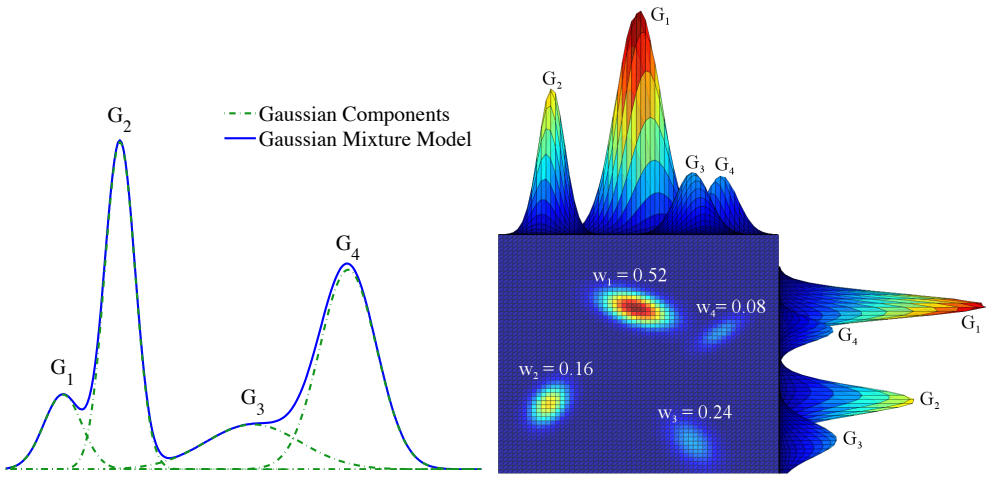


Figure 4.2: A 1-dimensional GMM object with m=4 components is depicted on the left and a 2-dimensional nGMM object consisting of m=4 components $G_1, \ldots, G_4$ is illustrated on the right. Thereby, the objects are comprised of three parameters, a weighting vector $w = (w_1, \ldots, w_4)$, a covariance matrix Σ, and a location vector μ. In case of the 1-dimensional GMM the covariance matrix is not a matrix but rather the standard deviation of the single components.

$$= \sum_{i=1}^{m} \left(\frac{w_i}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left(-\frac{(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)}{2} \right) \right)$$

Due to the fact that the weights of each GMM add to unity we can additionally deduce that

$$\int_{\mathbb{R}^d} f_{\mathcal{G}}(x; w, \mu, \Sigma)\, \mathbf{d}x = 1. \tag{4.5}$$

Two examples of GMMs are depicted in Figure 4.2. Both GMMs consist of $m = 4$ Gaussian components $G_1, \ldots, G_4$. The GMM on the left comprises four 1-dimensional Gaussian components (cf. Eq. 4.1) which are connected using Eq. 4.4 by replacing the covariance matrix with the standard deviation. Hence, the GMM consists of the weights $w = (w_1, \ldots, w_4) = (0.35, 0.4, 0.15, 0.1)$ summing up to unity, the location vector $\mu = (\mu_1, \ldots, \mu_4) = (1, 13,$

8, -2), and the standard deviation $\sigma = (\sigma_1, \ldots, \sigma_4) = (0.8, 1.5, 2.5, 1)$. In the 2-dimensional example on the right Eq. 4.4 can be applied to obtain the GMM with the 4 components. The weights $w_1, \ldots, w_4$ are depicted on top of each component again summing up to unity and the location vector is a 4×2 vector $\mu = (\mu_{11}, \mu_{12}; \ldots; \mu_{41}, \mu_{42})$. Since we have a 2 dimensional GMM we also have covariances between the two features leading to one 2×2 covariance matrix for each component $\Sigma_1, \ldots, \Sigma_4$. Hence, as soon as more than one dimension is existent correlations between features (indicated by the rotated ellipsoids in Figure 4.2 left) can occur leading to non-axis parallel Gaussian distributions.

4.4.3 Non-axis Parallel GMM Approximation

To accelerate the computation of our similarity measure for achieving the most similar objects to a query object we propose an approximation technique. This is required since our similarity measure is very expensive due to the consideration of the entire covariance matrix information including correlations between different features. In this approximation technique each weighted non-axis parallel Gaussian component $\mathcal{G}_i$ of the nGMM with the three parameters w_i, μ_i, and Σ_i is approximated. The goal is to replace each non-axis parallel Gaussian component with an axis parallel Gaussian component having the parameters ψ_i, μ_i, and $\chi_i = \phi_i D_i$. Thereby, ψ_i and ϕ_i are scalar values and D_i is a diagonal matrix leading to an axis parallel matrix and hence an axis parallel Gaussian representation. ψ_i represents the new weighting factor and ϕ_i in combination with D_i corresponds to the new axis parallel covariance matrix χ_i. To put it in other words, the conservative approximation of a non-axis parallel Gaussian curve leads to an axis parallel Gaussian curve.

The aim is to conservatively enclose the original non-axis parallel Gaussian $\mathcal{G}_i$ in the new axis parallel Gaussian which can be achieved by a specific setting of the new weighting factor ψ_i. In the following we will explain how

to determine the axis parallel representation of a non-axis parallel Gaussian.

We start up with the non-axis parallel Gaussian curve which can be described by the three original model components w_i, μ_i, and Σ_i

$$w_i \cdot N(x; \mu_i, \Sigma_i) = \frac{w_i}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left(-\frac{(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)}{2}\right) \tag{4.6}$$

The notion in the exponent not considering the $-\frac{1}{2}$ is the Mahalonobis Distance (MD) or generalized squared interpoint distance [Mah36, DMJRM00] between x and μ_i of the same distribution with the covariance matrix Σ_i

$$MD(x; \mu_i, \Sigma_i)^2 = (x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i). \tag{4.7}$$

The Mahalonobis distance uses correlations in the data to specify the similarity of an unknown and a known variable because the MD is obtained by inverting the covariance matrix.

Note, that we intent to replace the covariance matrix Σ_i by an axis parallel matrix χ_i. Hence, we have to look for a distance function $MD(x; \mu_i, \chi_i)^2$ for the axis parallel matrix χ_i satisfying the following characteristic

$$MD(x; \mu_i, \chi_i)^2 \leq MD(x; \mu_i, \Sigma_i)^2, \qquad \forall x, \mu_i \in \mathbb{R}^d. \tag{4.8}$$

Since the exponential function is a monotonous function, in our case it is a steeply and monotonously downward-sloping curve on account of the $-\frac{1}{2}$ in the exponent, we have to find a real lower bound of $MD(x; \mu_i, \chi_i)^2$ to guarantee that $MD(x; \mu_i, \chi_i)^2$ is smaller or equal in contrast to the original Mahalonobis distance.

The covariance matrix Σ_i consists of the covariances on the off diagonal and the squared variances $\sigma_{i1}^2, \ldots, \sigma_{id}^2$ on the diagonal. In order to obtain an axis parallel matrix representation χ_i we need a diagonal matrix. Therefore, we fill D_i, being part of the axis parallel matrix χ_i, with the squared variances

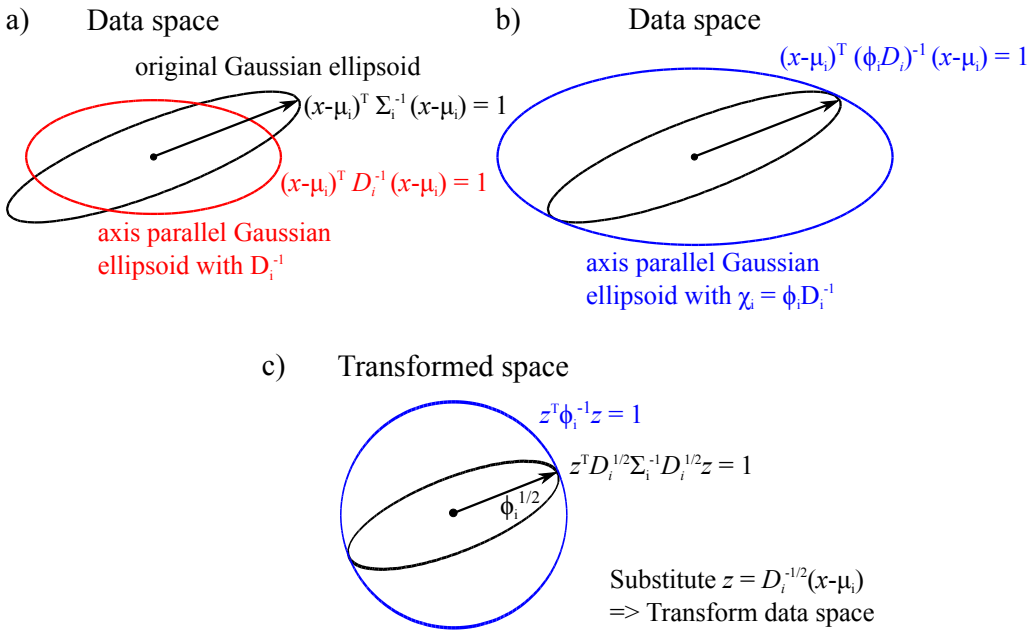


Figure 4.3: Illustration of the approximation using ϕ_i and D_i^{-1}. a) Without the scaling factor ϕ_i the evolving axis parallel ellipsoid (red ellipsoid) is not a conservative approximation of the original Gaussian ellipsoid (black ellipsoid). b) Including the scaling factor in the Mahalonobis distance leads to the desired conservative approximation (blue ellipsoid). c) In order to achieve the scalar value ϕ_i we transform both ellipsoids by substituting $z = D_i^{-1/2}(x-\mu_i)$.

of the original covariance matrix Σ_i

$$D_i = \text{diag}(\sigma_{i1}^2, \ldots, \sigma_{id}^2). \tag{4.9}$$

If we would now simply insert the diagonal matrix D_i in the Mahalonobis distance $MD(x;\mu,D)^2$ we would not receive a lower bound of $MD(x;\mu,\Sigma)^2$ as shown in Figure 4.3 a. Hence, we need to multiply the matrix D_i with a scalar value ϕ_i also called the scaling factor to obtain the final axis parallel matrix χ_i illustrated in Figure 4.3 b.

In order to determine this scaling factor, we use a transformation which leads to a spherical representation of the axis parallel ellipsoid. An ellipse can

be described by the equation of center $x^2/a^2 + y^2/b^2 = 1$ with a and b being positive real numbers. In case of a sphere $a = b$ and since the denominator of the spherical equation of center is the squared radius of the sphere, the radius can be easily determined by taking the squared root of the denominator.

Now, to determine a spherical representation of the ellipsoid we convert the data space. In other words, we multiply the original data $(x - \mu_i)$ with the inverted square root of the diagonal matrix D_i leading to a spherical view of the axis parallel ellipsoid depicted in Figure 4.3 c. To transform the ellipsoid we first have to obtain $\sqrt{D_i^{-1}}$. The inverse and the square root of D_i can be calculated for each element $\sigma_{1i}^2, \ldots, \sigma_{id}^2$ separately, due to the fact that the matrix D_i is a diagonal matrix:

$$\sqrt{D_i^{-1}} = \text{diag}\left(\frac{1}{\sqrt{\sigma_{i1}^2}}, \ldots, \frac{1}{\sqrt{\sigma_{id}^2}}\right). \tag{4.10}$$

Now, to determine the transformation we substitute $z = \sqrt{D_i^{-1}}(x - \mu_i)$, which can be reformulated as $(x - \mu_i) = z\sqrt{D_i}$. Hence, the axis-parallel ellipsoid

$$(x - \mu_i)^T \chi_i^{-1} (x - \mu_i) = 1 \tag{4.11}$$

becomes with $\chi_i = \phi_i D_i$

$$(z\sqrt{D_i})^T (\phi_i D_i)^{-1} (\sqrt{D_i} z) = \frac{z^T z}{\phi_i} = 1 \tag{4.12}$$

while the non axis-parallel ellipsoid becomes

$$(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) = z^T \sqrt{D_i} \Sigma_i^{-1} \sqrt{D_i} z = 1. \tag{4.13}$$

Since, the radius of the sphere is the square root of the denominator, it is $\sqrt{\phi_i}$. This radius corresponds to the largest semiaxis of the inner ellipsoid which can be determined by taking the inverse of the smallest Eigenvalue of the inverse correlation matrix C_i^{-1}.

The correlation matrix C_i with the elements $corr_{r,s}$ ($1 \leq r \leq d$, $1 \leq s \leq d$) of the Gaussian $\mathcal{G}_i$ can be obtained by dividing each element $cov_{r,s}$ of the covariance matrix by the corresponding square root of the variance of $cov_{r,r} = \sigma_r^2$ and $cov_{s,s} = \sigma_s^2$, hence, $corr_{r,s} = \frac{cov_{r,s}}{\sqrt{cov_{r,r}cov_{s,s}}} = \frac{cov_{r,s}}{\sqrt{\sigma_r^2\sigma_s^2}}$. Since the matrix D_i contains the diagonal elements of the covariance matrix $D_i = \text{diag}(\sigma_{i1}^2, \ldots, \sigma_{id}^2)$ we can also obtain the correlation matrix by

$$C_i = \sqrt{D_i^{-1}}\Sigma_i\sqrt{D_i^{-1}}. \tag{4.14}$$

Looking at equation 4.13 we have the inverse of the correlation matrix

$$C_i^{-1} = \sqrt{D_i}\Sigma_i^{-1}\sqrt{D_i}. \tag{4.15}$$

The aim is to determine the largest semi-axis of the inner ellipsoid which corresponds to the transformed ellipsoid of the original non-axis parallel Gaussian distribution. To determine the largest semi-axis of the inner ellipsoid we have to calculate the inverse of the smallest Eigenvalue of the inverse correlation matrix C_i^{-1}. Thereby, the Eigenvalues Λ' as well as the Eigenvectors V' can be obtained by the Eigenvalue decomposition of C_i^{-1}.

$$C_i^{-1} = \sqrt{D_i}\Sigma_i^{-1}\sqrt{D_i} = V'\Lambda'V'^T \tag{4.16}$$

Note, that the Eigenvalue matrix is a diagonal matrix $\Lambda' = \text{diag}(\lambda_1', \ldots, \lambda_d')$, therefore, the inverse of the smallest Eigenvalue can be simply obtained by

$$\phi_i = \left(\min(\text{diag}(\lambda_1', \ldots, \lambda_d'))\right)^{-1}. \tag{4.17}$$

Using the correlation matrix C_i instead of the inverse of the correlation matrix C_i^{-1} we can equivalently determine ϕ_i by using Eigenvalue decomposition of the correlation matrix C_i

$$C_i = \sqrt{D_i^{-1}}\Sigma_i\sqrt{D_i^{-1}} = V\Lambda V^T \tag{4.18}$$

but instead of taking the inverse of the smallest Eigenvalue we now have to take the largest Eigenvalue of the set of Eigenvalues $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_d)$ leading to

$$\phi_i = \max(\text{diag}(\lambda_1, \ldots, \lambda_d)) \tag{4.19}$$

and subsequently

$$\chi_i = \phi_i D_i = \max(\text{diag}(\lambda_1, \ldots, \lambda_d)) \cdot D_i. \tag{4.20}$$

Now that we have obtained the new axis parallel matrix χ_i we only have to determine the new weighting factor in order to receive a complete Gaussian component with the three parameters ψ_i, μ_i, and χ_i. For identifying the new scalar value ψ_i we have to restructure the original Gaussian component as follows:

$$\begin{aligned} w_i \cdot N(x; \mu_i, \Sigma_i) &= \frac{w_i}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left(-\frac{(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)}{2}\right) \\ &\leq \frac{w_i}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left(-\frac{(x-\mu_i)^T \chi_i^{-1} (x-\mu_i)}{2}\right) \\ &= \frac{w_i \sqrt{|\chi_i|}}{\sqrt{|\Sigma_i|}} \frac{1}{\sqrt{(2\pi)^d |\chi_i|}} \exp\left(-\frac{(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)}{2}\right) \\ &= w_i \sqrt{\frac{|\chi_i|}{|\Sigma_i|}} N(x; \mu_i, \chi_i) \end{aligned} \tag{4.21}$$

Taken together, the new model component is composed of the weighting factor ψ_i which can be written as

$$\psi_i = \text{approxW}(w_i, \chi_i, \Sigma_i) = w_i \sqrt{\frac{|\chi_i|}{|\Sigma_i|}} \tag{4.22}$$

and the new matrix

$$\chi_i = \phi_i D_i = \text{approxSigma}(\Sigma_i) = \phi_i \cdot \text{diag}(\sigma_{i1}^2, \ldots, \sigma_{id}^2). \tag{4.23}$$

We have to mention that the new weighting factors of the newly emerged axis parallel nGMM approximation do not sum up to unity anymore. But it is not necessary that the new axis parallel conservative approximation of the original PDF is a PDF for itself since it is only an approximation which is used as an upper bound to save absolute probability calculations.

4.4.4 Clustering Of Non-axis Parallel Gaussians

By now each component of the nGMMs can be represented by an axis parallel Gaussian approximation. If the original Gaussians have only weak correlations between the features the Gaussian approximation is very similar to the original Gaussian but in case of very strong correlations (45° angle) of two or more coordinate axis the original Gaussian can not be approximated well.

A solution for this problem might be to transform the Euclidean coordinate system in case of strong correlations. For a better understanding Figure 4.4 depicts an example of two GMMs G and H which are composed of four Gaussian components each $G = (G_1, \ldots, G_4)$ and $H = (H_1, \ldots, H_4)$. In the Euclidean coordinate system (Figure 4.4 a) four of the Gaussian components (G_1, G_2, H_1, H_2) can be approximated very well since their features are almost uncorrelated. But the remaining four Gaussians (G_3, G_4, H_3, H_4) having really strong correlations (close to a bisector) can only be badly approximated. Note, that even though e.g. G_1 and G_2 have different rotation angles they can still be approximated equally well in the same coordinate system. The rotation of the coordinate system by 45° (Figure 4.4 b) causes Gaussians G_3, G_4, H_3, and H_4 to be approximated well having a low approximation error while the approximation error of G_1, G_2, H_1, and H_2 is very high. Thus, separating the set of Gaussian distributions according to their

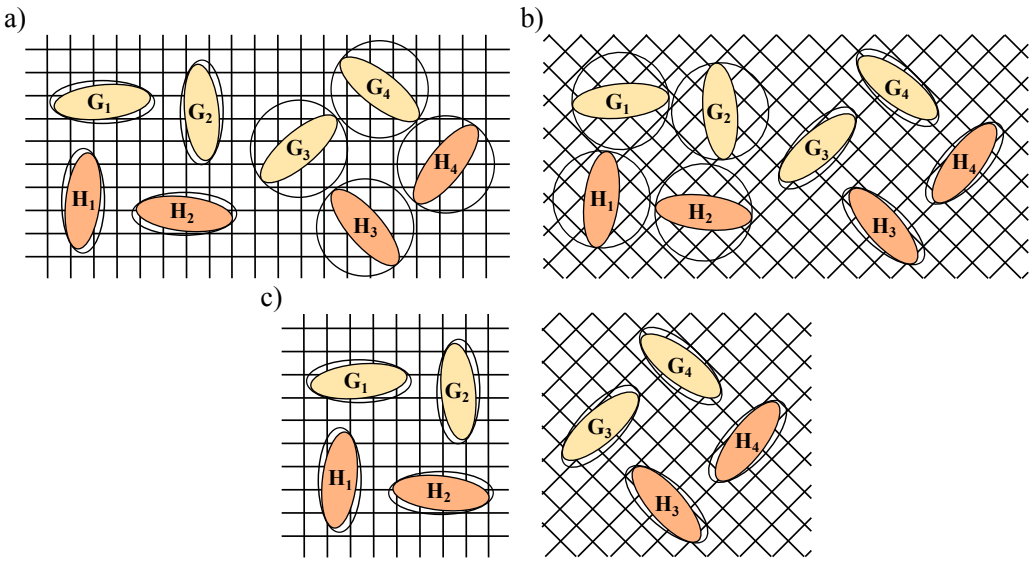


Figure 4.4: Two 2-dimensional Gaussian Mixture Models G and H each comprising four Gaussian components $G = (G_1, \ldots, G_4)$ and $H = (H_1, \ldots, H_4)$. a) Normal coordinate system where components G_1, G_2, H_1, and H_2 can be approximated very well but components G_3, G_4, H_3, and H_4 can only be badly approximated. b) Using a coordinate system which has been rotated by 45° G_3, G_4, H_3, and H_4 can be approximated well, whereas G_1, G_2, H_1, and H_2 can only be badly approximated. c) Hence, separating the set of Gaussians according to their rotation angle in space, all ellipsoids can be approximated equally well.

rotation angle in space (Figure 4.4 c) independent of the GMM they belong to would significantly decrease the approximation error of all Gaussian distributions.

We decided to use a clustering method in order to group those Gaussian distributions together having a similar main extension (rotation angle) in space. Thereby, the rotation angle can be determined by means of the Eigenvector matrix arising from Eigenvalue decomposition of the covariance matrix. Since, for clustering we need cluster representatives, we have to find typical Eigenvector matrices (prototype matrices) summarizing a set of Gaussian distributions according to their rotation angle. To obtain those pro-

totype matrices we require a similarity measure for combining matrices with a small approximation error in the same coordinate system. Additionally, we need a procedure to determine all rotation angles of a Gaussian distribution using the Eigenvector matrix.

For calculating the rotation angles we decided to use the idea of Givens rotation matrices [PFTV92]. A Givens rotation also called Jacobi-Rotation is a rotation in one plane which is spanned by two coordinate axis. Since we want to obtain c new coordinate systems all being represented by $l = (d^2 - d)/2$ Givens rotation matrices B_{uv} $(1 \leq u \leq c, 1 \leq v \leq l)$. Consequently, the components $\mathcal{G}_i$ $(1 \leq i \leq m)$ of an object $\mathcal{G} \in DB$ defined by the parameter w_i, μ_i, and Σ_i are not saved by their original coordinates μ_i but rather by the rotated version of their coordinates $B_{uv} \cdot \mu_i$. The Mahalonobis distance in the Gaussian PDFs thus has to be adjusted as follows

$$MD(x; \mu_i, \Sigma_i)^2 = (B_{uv}x - B_{uv}\mu_i)^T B_{uv}\Sigma_i^{-1}B_{uv}^T(B_{uv}x - B_{uv}\mu_i). \tag{4.24}$$

If the rotated covariance matrix $B_{uv}\Sigma_i^{-1}B_{uv}^T$ is more axis parallel then the original covariance matrix Σ_i^{-1} indicated by a smaller approximation error

$$\sqrt{\frac{|B_{uv}\chi_i B_{uv}^T|}{|B_{uv}\Sigma_i B_{uv}^T|}} < \sqrt{\frac{|\chi_i|}{|\Sigma_i|}} \tag{4.25}$$

then it is useful to rotate the Gaussian PDF.

4.4.4.1 Rotation Angles By Givens Rotations

Givens rotation matrices in combination with Eigenvalue decomposition can be applied to determine all $l = (d^2-d)/2$ rotation angles θ_v $(1 \leq v \leq l)$ of one Gaussian distribution. These l rotation angles θ_v will then be used to cluster the Gaussian distributions according to their orientation in space. For each coordinate axis pair combination the rotation angle has to be determined separately. In other words, after having obtained the first rotation angle θ_1

of an ellipsoid, the ellipsoid has to be rotated by that angle before being able to achieve the next angle.

For a Gaussian distribution $\mathcal{G}_i$ rotation angles can be determined by the Eigenvalue decomposition of the Gaussian's covariance matrix Σ_i

$$\Sigma_i = V \Lambda V^T \tag{4.26}$$

using the Eigenvector $V_{max} = \begin{pmatrix} x_1 \\ \cdots \\ x_d \end{pmatrix}$ of the largest Eigenvalue ($\max(\Lambda)$). Since the arctangent function for two real arguments x_1 and x_2 ($\arctan2(x_1, x_2)$) can be used to obtain the angle between the positive x_1 axis of a plane and the point given by the coordinates x_1, x_2 we can adopt

$$\theta_v = \arctan 2(x_1, x_2) = \begin{cases} \arctan(x_1/x_2) & x_2 > 0 \\ \arctan(x_1/x_2) + \pi & x_2 < 0 \\ \pi/2 & x_2 = 0, x_1 \geq 0 \\ -\pi/2 & x_2 = 0, x_1 < 0 \end{cases} \tag{4.27}$$

to compute the l rotation angles θ_v based on the Eigenvector V_{max}.

As mentioned before, each angle, corresponding to two coordinate axis has to be calculated separately. In order to achieve all l rotation angles we have to rotate the covariance matrix after each rotation angle calculation. The rotation of the covariance matrix Σ_i can be achieve by a Givens rotation matrix of the form

$$R(\theta_v) = \begin{pmatrix} Id & & & & \\ & \cos(\theta_v) & & \sin(\theta_v) & \\ & & Id & & \\ & -\sin(\theta_v) & & \cos(\theta_v) & \\ & & & & Id \end{pmatrix}. \tag{4.28}$$

Thereby, the recently computed rotation angle θ_v is inserted in the corre-

sponding Givens rotation matrix. Note, that the position of the $\cos(\theta_v)$ and $\sin(\theta_v)$ corresponds to the two axis examined at present. This Givens rotation matrix can then be used to rotate Σ_i using $R(\theta_v) \cdot \Sigma_i \cdot R(\theta_v)^T$. After having rotated Σ_i the next rotation angle can be calculated applying again an Eigenvalue decomposition on the newly determined covariance matrix and consequently calculating arctan2 of the newly evolved V_{max} of the subsequent coordinate axis pair followed by the rotation of the covariance matrix. This has to be done l times in total until all l rotation angles have been determined.

For a better understanding of the rotation angle determination we will clarify the procedure using a 3D example. In a 3D space there are a total of $l = (3^2 - 3)/2 = 3$ rotation angles which have to be computed. Given a covariance matrix Σ, we first have to apply an Eigenvalue decomposition for obtaining the Eigenvector $V_{max} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ of the largest Eigenvalue. To determine the first rotation angle we now have to apply arctan2 to one coordinate axis pair. We will start with the axis pair x_1, x_2 leading to $\theta_1 = \text{arctan2}(x_1, x_2)$. Now, the covariance matrix has to be rotated by the Givens rotation matrix

$$R(\theta_1) = \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) & 1 \\ -\sin(\theta_1) & \cos(\theta_1) & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

leading to the new rotated covariance matrix $\Sigma' = R(\theta_1) \cdot \Sigma \cdot R(\theta_1)^T$. Based on the new covariance matrix Σ' a new Eigenvector $V'_{max} = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix}$ has to be calculated by Eigenvalue decomposition. Now we can compute the second rotation angle by inserting the next axis pair in the two-argument arctangent function $\theta_2 = \text{arctan2}(x'_1, x'_3)$. The corresponding rotation matrix

$$R(\theta_2) = \begin{pmatrix} \cos(\theta_2) & 1 & \sin(\theta_2) \\ 1 & 1 & 1 \\ -\sin(\theta_2) & 1 & \cos(\theta_2) \end{pmatrix}$$

is used to obtain $\Sigma'' = R(\theta_2) \cdot \Sigma' \cdot R(\theta_2)^T$. The third and last rotation angle θ_3 can be determined analogously leading to $\theta_3 = \arctan2(x_2'', x_3'')$.

Until now we have obtained all l rotation angles for each Gaussian distribution $\mathcal{G}_i$. These angles can capture arbitrarily large negative as well as positive degree values. However, since a Gaussian distribution having a 0° angle has practically the same orientation in space as a Gaussian distribution with a ±180° angle and orthogonal angles should also be clustered together (cf. Figure 4.4) we have to shift the obtained angles before determining the angle cluster representatives. The goal is to shift all angles so that they are located between 0° and 90°. Therefore, angles have to be shifted as follows

$$\text{shift}(\theta_v) = \begin{cases} 90° + (\theta_v \bmod 90°), & \theta_v < 0° \\ \theta_v \bmod 90°, & \theta_v \geq 90° \\ \theta_v, & \text{otherwise} \end{cases} \tag{4.29}$$

Finally, all angles are located in the interval between 0° and 90°, hence we are ready to determine cluster representatives.

4.4.4.2 Rotation Angle Cluster Representative

By now we have obtained l rotation angles for each Gaussian distribution $\mathcal{G}_i$. Since each Gaussian Mixture Model contains m Gaussian distributions (m being different for each GMM) and the database DB consists of n GMMs we have a total of $m \times n$ l-dimensional rotation angle sets. The goal is to cluster the Gaussian distributions according to their orientation in space, hence our next step is to find a cluster representative θ_{uv} $(1 \leq u \leq c, 1 \leq v \leq l)$ for each angle cluster C. As input the clustering algorithm needs the recently calculated $m \cdot n$ l-dimensional rotation angle sets. As clustering algorithm we decided to use the parameter-free clustering algorithm X-Means [PM00] which is a parameter-free extension of the partitioning clustering algorithm K-Means.

Typically X-Means uses Euclidean distances for clustering but as mentioned in the previous Subsection we want to insert angles, being cyclic values, into the algorithm. Note, that the angles contain values between 0° and 90°, while 0° and 90° should be clustered together. In other words, the distance between 0° and 90° equals zero. Therefore, we first have to adjust the basic X-Means algorithm to the cyclic measures. Hence, instead of using Euclidean distance measures in order to obtain the similarity between two sets of l rotation angles θ_q and θ_p of two Gaussian distributions q and p

$$d_E(\theta_q, \theta_p) = \sqrt{\sum_{i=1}^{l} (\theta_{qi} - \theta_{pi})^2} \tag{4.30}$$

we adjusted the Euclidean distance to cyclic values by

$$d_C(\theta_q, \theta_p) = \sqrt{\sum_{i=1}^{l} \left(\min\{|\theta_{qi} - \theta_{pi}|; 90 - |\theta_{qi} - \theta_{pi}|\}\right)^2}. \tag{4.31}$$

To clarify the impact of the cyclic distance measure we will give a short example. Given are the three $l = (2^2 - 2)/2 = 1$-dimensional rotation angles $\theta_q = (89°)$, $\theta_p = (1°)$, and $\theta_r = (47°)$. Using the Euclidean distance measure would lead to the distances $d_E(\theta_q, \theta_p) = 88°$, $d_E(\theta_q, \theta_r) = 42°$, and $d_E(\theta_p, \theta_r) = 46°$ while the cyclic Euclidean distance measure leads to the distances $d_C(\theta_q, \theta_p) = 2°$, $d_C(\theta_q, \theta_r) = 42°$, and $d_C(\theta_p, \theta_r) = 44°$. Since the two angles 89° and 1° are almost orthogonal the desired distance for our algorithm is 2° rather than 88°.

Subsequently, we also have to change the update of the cluster representatives θ_{uv} ($1 \leq u \leq c, 1 \leq v \leq l$; with c being the number of clusters) to cyclic measures. For each of the l rotation angle dimensions the following procedure is done separately. All angles contained in a cluster are sorted in ascending order for each of the l dimensions using the HeapSort sorting algorithm [BFF96]. The sorting is done to obtain a ring-like data structure of the

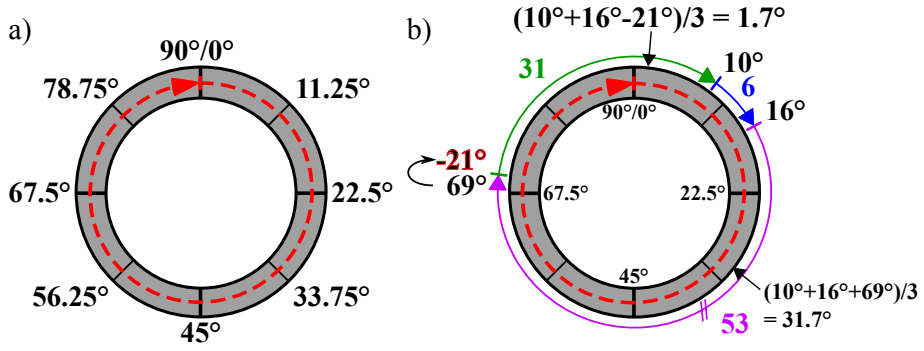


Figure 4.5: Update of a cluster center. a) All angles of each dimension are sorted separately in an ascending order resulting in a ring-like data structure where 90° is equal to 0°. b) An example of a cluster update. The illustrated cluster comprises the three angles 10°, 16°, and 69°. After sorting the angles in ascending order distances between all θ_i and θ_{i+1} are calculated indicated by the colored numbers 6, 53, and 31. The largest distance is then used as cut-point resulting in a flip of -90° of all angles which are larger than 16°. Hence, 69° has to be converted to -21° leading to a cluster representative $\theta_{uv} = 1.7°$.

angles contained in the cluster as depicted in Figure 4.5 a. Then in the list of sorted angles all distances between angle θ_i and the successive angle θ_{i+1} are obtained. The largest obtained distance is then used as cut-point leading to a flip of -90° of all angles that are larger than θ_i. In other words, we subtract 90° from all angles which are larger than θ_i. In the last step of obtaining cluster representatives the mean of all cluster members is calculated.

An example of how to determine a cluster representative is depicted in Figure 4.5 b. Thereby, the three angles 10°, 16°, and 69° are sorted in ascending order and distances (indicated by the colored values) are calculated. The two red dashes mark the largest distance (53) which is used as cut-point. Therefore, all angles larger than 16° have to be converted by subtracting 90° (69°-90° = -21°). The cluster representative can then be calculated by building a mean of all cluster members leading to $\theta_{uv} = 1.7° = (10° + 16° + -21°)/3$ instead of $\theta_{uv} = 31.7° = (10° + 16° + 69°)/3$ using the original

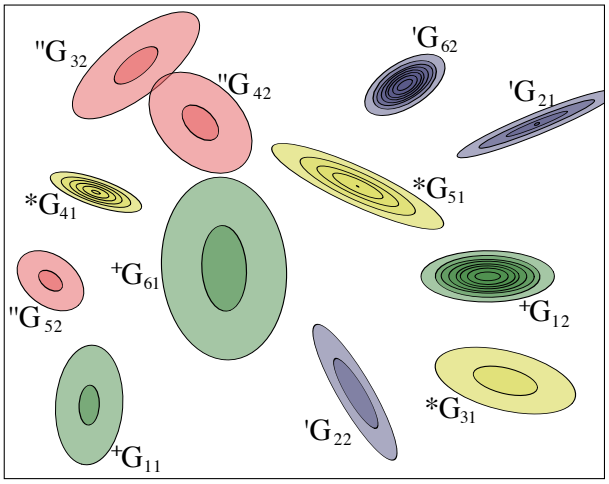


Figure 4.6: Clustering example of the X-Means clustering adjusted to cyclic distances. Six GMMs $G_1, \ldots, G_6$ each comprising $m = 2$ Gaussian distributions $G_{11}, G_{12}, \ldots, G_{61}, G_{62}$ are depicted. The 12 Gaussian distributions are clustered according to their main orientation in space resulting in 4 clusters containing 3 Gaussian distributions each. The different coloring as well as the signs in front of the distribution name constitute the cluster belonging of the Gaussian distributions (cluster 1: green, +; cluster 2: blue, '; cluster 3: red,"; cluster 4: yellow, *).

angles.

To illustrate the effect of clustering Gaussian distributions according to their orientation in space we depicted a clustering example of 12 Gaussian distributions belonging to a total of six GMM $(G_1, \ldots, G_6)$ in Figure 4.6. Each GMM consists of $m = 2$ Gaussian distributions indicated by the second subscripted number, e.g. G_{11} and G_{12} both belong to the same GMM G_1. Note, that the amount of Gaussian distributions m of the GMMs can vary between GMMs but for simplicity reasons we decided to use an equal amount in the example. The clustering algorithm X-Means which has been adjusted to cyclic values could identify a total of 4 clusters (indicated by the different coloring of the distributions and also by the signs ", ', +, and *). The cluster representatives for the four clusters are $\theta_{11} = 1.6°$ (green, +), $\theta_{21} = 31.6°$

(blue, '), $\theta_{31} = 40°$ (red, "), and $\theta_{41} = 71.2°$ (yellow, *). We want to point out that the clustering does not take into account the mean μ_i of the Gaussian distributions, therefore, Gaussian distributions do not need to have a similar mean in order to be clustered together, since the clustering considers solely the rotation angles of the distributions.

4.4.4.3 Coordinate System Update

As a reminder, the main goal of the rotation angle clustering was to rotate those Gaussian distributions which are highly correlated (45°) in order to determine a more meaningful approximation of the original Gaussian distribution when applying the nGMM approximation from Subsection 4.4.3. Hence, after having completed the actual rotation angle clustering by the cyclic X-Means, all Gaussian distributions belonging to one cluster have to be rotated l times in order to be located in the new coordinate system.

The rotation is accomplished once more by inserting rotation angles in Givens rotation matrices. The l rotation angles of the cluster representatives θ_{uv} $(1 \leq u \leq c, 1 \leq v \leq l)$ are inserted one at a time in the corresponding Givens rotation matrices $B_{uv} = R(\theta_{uv})$. These matrices are then used to transform the mean μ_i and the covariance matrix Σ_i of each Gaussian distribution before being inserted in the Mahalonobis distance. Thus each μ_i as well as each Σ_i has to be rotated l times using the corresponding cluster representative θ_{uv} by

$$\text{rotateMean}(\mu_i, \theta_{uv}) = B_{uv} \cdot \mu_i = R(\theta_{uv}) \cdot \mu_i \tag{4.32}$$

and

$$\text{rotateSigma}(\Sigma_i, \theta_{uv}) = B_{uv} \cdot \Sigma_i \cdot B_{uv}^T = R(\theta_{uv}) \cdot \Sigma_i \cdot R(\theta_{uv})^T. \tag{4.33}$$

The rotated mean and covariance matrix can then be inserted in the

Mahalonobis distance as follows

$$MD(x;\mu_i,\Sigma_i)^2 = (B_{uv}x - B_{uv}\mu_i)^T B_{uv}\Sigma_i^{-1}B_{uv}^T(B_{uv}x - B_{uv}\mu_i) \tag{4.34}$$

before the approximation takes place leading to a smaller approximation error in comparison to using the original covariance matrix and mean.

4.4.5 Object Identification

Until now we have introduced an approximation procedure for non-axis parallel Gaussian Mixture Models and a clustering algorithm to minimize the approximation error. In the following we will define a similarity measure to compare two nGMMs. Since this similarity measure is very time consuming we will subsequently draw the connection to the approximation introduced earlier to accelerate our similarity search while producing no false dismissals due to the upper bound quality of the approximation.

4.4.5.1 Object Identification Using nGMMs

In order to identify the most similar objects in a database DB consisting of n uncertain objects to a given query object $\mathcal{G}'$ we need to define a similarity measure. The database is represented by $DB = \{\mathcal{G}_1, \ldots, \mathcal{G}_n\}$. Each object in the database as well as the query object itself are defined by a non-axis parallel GMM using the parameters w, μ, and Σ

$$\mathcal{G} = \{(w_1, \mu_1, \Sigma_1), \ldots, (w_m, \mu_m, \Sigma_m)\}. \tag{4.35}$$

Note, that the query object does not need to be part of the database.

Since we also take feature correlations into account we have to consider the entire covariance matrix information for the similarity search. Now, to obtain the similarity between an object $\mathcal{G}^* \in DB$ and the query object

$\mathcal{G}'$ we use the joint Probability Density Value PDV of two objects. This Probability Density Value constitutes the joint probability with which a random variable x drawn from the nGMM distribution of a database object $\mathcal{G}^*$, $f_{\mathcal{G}^*}(x; w^*, \mu^*, \Sigma^*)$, coincides with a random variable x' which was drawn from the distribution of the query object $f_{\mathcal{G}'}(x'; w', \mu', \Sigma')$. This joint Probability Density Value can be defined as

$$PDV(\mathcal{G}^*, \mathcal{G}') = \int_{\mathbb{R}^d} f_{\mathcal{G}^*}(x; w^*, \mu^*, \Sigma^*) \cdot f_{\mathcal{G}'}(x; w', \mu', \Sigma')\mathbf{d}x. \tag{4.36}$$

We expect the query object to always be statistically independent of the database objects and hence we can rephrase the formula of the PDV by

$$PDV(\mathcal{G}^*, \mathcal{G}') = \sum_{i=1}^{m^*} \left(\sum_{j=1}^{m'} \left(w_i^* w_j' \int_{\mathbb{R}^d} N(x; \mu_i^*, \Sigma_i^*) \cdot N(x; \mu_j', \Sigma_j')\mathbf{d}x \right) \right). \tag{4.37}$$

Based on the work of [BPS06] we will demonstrate in the following that the assumption

$$\begin{aligned} PDV(\mathcal{G}^*, \mathcal{G}') &= \sum_{i=1}^{m^*} \left(\sum_{j=1}^{m'} \left(w_i^* w_j' \int_{\mathbb{R}^d} N(x; \mu_i^*, \Sigma_i^*) \cdot N(x; \mu_j', \Sigma_j')\mathbf{d}x \right) \right) = \\ &= \sum_{i=1}^{m^*} \left(\sum_{j=1}^{m'} w_i^* w_j' \cdot N(\mu_i^*, \Sigma_i^* + \Sigma_j', \mu_j') \right) \end{aligned} \tag{4.38}$$

can be applied since variances and covariances of independent stochastic variables can be simply summed up.

Using substitution of the definition of the normal distribution

$$N(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \cdot \exp\left(-\frac{(x-\mu)^T \Sigma^{-1} (x-\mu)}{2} \right) \tag{4.39}$$

the probability density of the product $N(x; \mu_i^*, \Sigma_i^*) \cdot N(x; \mu_j', \Sigma_j')$ can be rewrit-

ten as follows:

$$N(x;\mu_i^*,\Sigma_i^*)\cdot N(x;\mu_j',\Sigma_j') = \frac{1}{\sqrt{(2\pi)^d|\Sigma_i^*+\Sigma_j'|}}\cdot \tag{4.40}$$

$$\cdot\exp\left(-\frac{(\mu_i^*-\mu_j')^T(\Sigma_i^*+\Sigma_j')^{-1}(\mu_i^*-\mu_j')}{2}\right)\cdot N\left(x;\frac{\Sigma_j'\mu_i^*+\Sigma_i^*\mu_j'}{\Sigma_i^*+\Sigma_j'},\frac{\Sigma_i^*\Sigma_j'}{\Sigma_i^*+\Sigma_j'}\right).$$

The first term corresponds to the normal distribution $N(\mu_i^*,\Sigma_i^*+\Sigma_j',\mu_j')$ and is independent from the integration variable x. Hence, the first term can safely be written in front of the integral, since it is a constant. The second term is the PDF of a normal distribution which always integrates to 1 when integrating from $-\infty$ to $+\infty$ independent of μ and Σ. Hence we obtain

$$PDV(\mathcal{G}^*,\mathcal{G}') = \sum_{i=1}^{m^*}\left(\sum_{j=1}^{m'}\left(w_i^*w_j'\int_{\mathbb{R}^d}N(x;\mu_i^*,\Sigma_i^*)\cdot N(x;\mu_j',\Sigma_j')\mathbf{d}x\right)\right) =$$

$$=\sum_{i=1}^{m^*}\left(\sum_{j=1}^{m'}\left(w_i^*w_j'N(\mu_i^*,\Sigma_i^*+\Sigma_j',\mu_j')\int_{\mathbb{R}^d}N\left(x;\frac{\Sigma_j'\mu_i^*+\Sigma_i^*\mu_j'}{\Sigma_i^*+\Sigma_j'},\frac{\Sigma_i^*\Sigma_j'}{\Sigma_i^*+\Sigma_j'}\right)\mathbf{d}x\right)\right)=$$

$$=\sum_{i=1}^{m^*}\left(\sum_{j=1}^{m'}w_i^*w_j'N(\mu_i^*,\Sigma_i^*+\Sigma_j',\mu_j')\right). \tag{4.41}$$

Until now we have obtained the PDV which indicates the relative probability with which a sample x drawn from the distribution $f_{\mathcal{G}^*}(x;w^*,\mu^*,\Sigma^*)$ equals a sample x' drawn from the distribution $f_{\mathcal{G}'}(x;w',\mu',\Sigma')$. Hence, the Probability Density Value denotes the probability with which $\mathcal{G}^*$ and $\mathcal{G}'$ overlap being the relative matching probability of a database object $\mathcal{G}^*\in DB$ and a query object $\mathcal{G}'$.

To obtain absolute probabilities for each object in the database to coincide with the query object we use the theorem of Bayes which leads to the

following

$$P(\mathcal{G}^*, \mathcal{G}') = \frac{PDV(\mathcal{G}^*, \mathcal{G}')}{\sum_{i=1}^{n} PDV(\mathcal{G}_i^*, \mathcal{G}')} \tag{4.42}$$

with n being the cardinality of the dataset $n = |DB|$.

The time complexity for object identification using absolute probabilities is $O(d^3)$. This can be ascribed to the calculation of the inverse of the covariance matrix (Σ^{-1}), the eigenvalue decomposition, and the calculation of the determinant ($|\Sigma|$). Since the calculation of the absolute probabilities for every object in the database is very time consuming and, therefore inefficient for object identification it is amenable to minimize these expensive computations.

4.4.5.2 Acceleration By Rotation And Approximation

To speed up the determination of absolute probabilities we use the conservative nGMM approximation step in combination with the rotation angle clustering introduced in Sections 4.4.3 and 4.4.4. Finding similar objects using the conservative approximation involves only a time complexity of $O(d)$ in contrast to the complex computation of the absolute probabilities which involves a time complexity of $O(d^3)$.

We first have to rotate and approximate each original Gaussian distribution $\mathcal{G}_i^*$ in the original database DB to build a rotated and approximated database DB_{ra}. Therefore, all Gaussian distributions $\mathcal{G}_i^*$ of all GMMs $\mathcal{G}^* \in DB$ have to be rotated according to the rotation angle cluster representative of the cluster they belong to. These rotated Gaussians can then be approximated following the procedure described in Subsection 4.4.3. By now we have a database of rotated and approximated Gaussian distributions defined by

$$\mathcal{G}_{i,ra}^* = \frac{\psi_i^*}{\sqrt{(2\pi)^d |\chi_i^*|}} \exp\left(-\frac{(B_{uv}x - B_{uv}\mu_i^*)^T B_{uv} \chi_i^{*-1} B_{uv}^T (B_{uv}x - B_{uv}\mu_i^*)}{2} \right). \tag{4.43}$$

4.4.5.3 k-Most Likely Identification Query

Putting everything together, we are now ready to search for a query object. The main goal of our approach is to identify those k objects in the database which are most similar to the query object. For this purpose we used a k-Most Likely Identification Query [BPS06]. Similar to a k-Nearest Neighbor (k-NN) search a k-MLIQ searches for the objects being most similar to a given query object but in contrast to a k-NN search probabilities are used for comparing two objects. Hence, not the k objects having the smallest distance to the query object but rather those k objects having the largest probability to be drawn from the same distribution are searched for.

Thereby, our k-MLIQ is a four-step approach (cf. Alg. 4.3):

1. Before being able to search a query object $\mathcal{G}'$ in the database of rotated and approximated GMMs $\mathcal{G}^*_{ra} \in DB_{ra}$, the query object also has to be rotated and approximated.

2. Approximated Probability Density Values PDV_a between the rotated and approximated query object and all rotated and approximated objects in the database DB_{ra} are calculated using

$$PDV_a(\mathcal{G}^*_{ra}, \mathcal{G}'_{ra}) = \sum_{i=1}^{m^*} \left(\sum_{i=1}^{m'} \psi^* \psi' N\left(B_{uv}\mu^*, B_{uv}\chi^* B_{uv}^T + B_{uv}\chi' B_{uv}^T, B_{uv}\mu'\right) \right). \tag{4.44}$$

3. Probability Density Values between those objects in the database having the highest approximated PDV_a and the query object are calculated until the abortion criterium is met.

4. PDVs are converted to absolute probabilities.

This four-step procedure will now be explained in more detail.

Figure 4.7 illustrates the complete four-step procedure of Alg. 4.3 including the determination of the upper bound PDV_{ub} for two different values

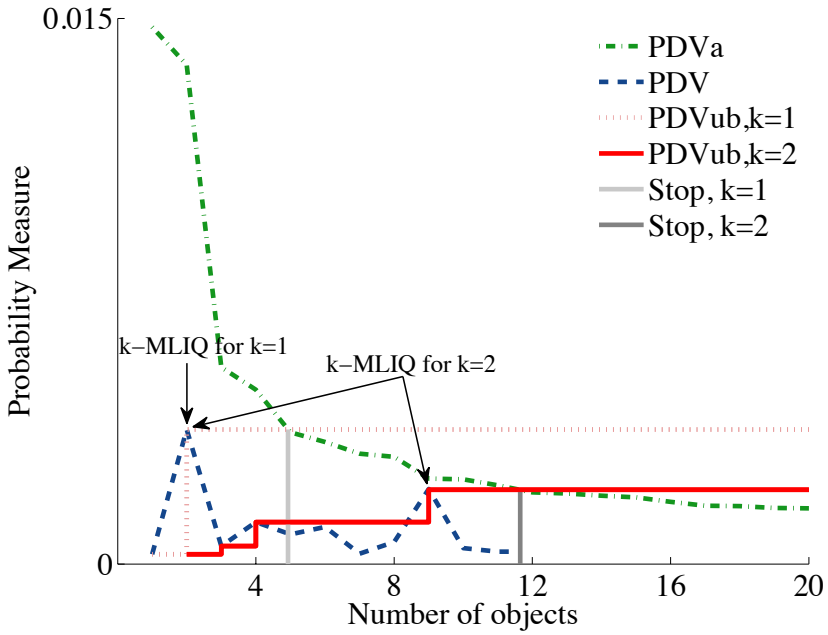


Figure 4.7: The abortion criterion and the determination of the upper bound PDV_{ub} are illustrated using a k-MLIQ search with $n = 100$ for $k = 1$ and $k = 2$. Shown are only the first 20 objects until the abortion criterion is met. Illustrated are steps 2 to 4 from the four-step procedure. Step 2: Approximated PDV_as for all 100 objects are computed and sorted (green line). Step 3: Then starting with the object on the very left (object 1) having the largest PDV_a (green line), PDV values between the original query object and the original data objects are computed from left to right until 1 for $k = 1$ or 2 objects for $k = 2$ are identified having a PDV larger than PDV_{ub} (light red line for $k = 1$, red line for $k = 2$). The upper bound is updated, indicated by a step of the threshold line, if a subsequent object has a PDV which is than PDV_{ub} of the smallest object in the set of k-MLIQs. The abortion criterion is met (light grey line for $k = 1$, grey line for $k = 2$) if one object has a PDV_a smaller than the upper bound PDV_{ub} at present. Step 4: Absolute probabilities of all objects in the set of k-MLIQ are then calculated and sorted.

of k. Before we can start the search of a query object $\mathcal{G}'$ in the database $\mathcal{G}^*_{ra} \in DB_{ra}$ we have to preprocess the query object (Step 1, not shown in Figure 4.7). In contrast to the database objects the query object has to be rotated into all coordinate systems in order to be able to calculate the similarity between the database objects $\mathcal{G}^*_{ra}$ and the query $\mathcal{G}'_{ra}$. Hence, we have a total of c (the number of rotation angle clusters) query object representations. Since we have a database of n objects after completion of Step 2 we have n approximated PDV_as. In Step 3 these values are sorted in descending order since we are interested in those objects having the largest PDV_a (green line). As mentioned earlier, approximated Gaussian distributions are an overestimation of the original Gaussians, subsequently approximated PDV_as are also an overestimation of the PDV between the original query object $\mathcal{G}'$ and the original database objects $\mathcal{G}^*$. The total number of PDV calculations necessary until convergence of the algorithm are not known in advance they are rather determined on the run. Starting with the object $\mathcal{G}^*$ in the database having the largest PDV_a (object number 1 on the left), subsequently, Probability Density Values $PDVs$ between $\mathcal{G}'$ and $\mathcal{G}^*$ are calculated. If k PDVs have been obtained having a larger PDV value (PDV_{ub}; light red line, $k = 1$; red line, $k = 2$) than the PDV_a regarded at present the procedure stops (light gray and gray line). In other words, if the smallest PDV (PDV_{ub}) in the set of k-MLIQs is larger then the PDV_a examined momentarily, the abortion criterion is met and the procedure stops.

In the last step, the PDV of the objects included in the set of k-MLIQs have to be converted to absolute probabilities. For this purpose the PDV of each object has to be opposed to the sum of all PDVs.

$$P(\mathcal{G}^*, \mathcal{G}') = \frac{PDV(\mathcal{G}^*, \mathcal{G}')}{\sum_{i=1}^{n} PDV(\mathcal{G}^*_i, \mathcal{G}')}. \tag{4.45}$$

As explained in Step 3 our aim is to compute as little PDVs as possible, hence a complete summation of the denominator is not possible. But since the remaining uncounted PDVs are smaller then their corresponding PDV_as

and since the PDV_as are sorted in descending order and are all smaller then the ones examined so far, the summation of the calculated PDVs is sufficient for calculating absolute probabilities. In other words, since those PDVs corresponding to PDV_as smaller than PDV_{up} belong to objects which are very dissimilar to the query object their PDVs are close to zero and are therefore irrelevant for the computation of the absolute probabilities.

Summing all up, after the preprocessing of the database (rotation and approximation) the k-MLIQ can start by first rotating the query object c times to be represented in all rotation angle clusters followed by the approximation of all query object representations. Then the four-step procedure for determining the k-MLIQs in the database is executed and those k objects in the database having the highest absolute probabilities of being drawn from the same distribution as the query object are returned.

4.5 Experiments

For evaluating the performance of our k-MLIQ algorithm we conducted several experiments including synthetic as well as real world data. The main goal was to demonstrate that our approach is superior to existing similarity and Nearest Neighbor searches not considering correlations in the data while reducing computation time due to the conservative approximation of the nGMMs. Hence, by adding the approximation filtering architecture to correlated nGMMs the overall runtime can be reduced compared to computing the absolute probability for every nGMM, while guaranteeing no false dismissals due to the conservativity of our filter.

As a runtime reference point for our approximation k-MLIQ search we used the k-MLIQ approach computing Probability Density Values PDVs between the query object and every single data object in the database, named complete k-MLIQ search, in the following. Furthermore, we compared our approximation k-MLIQ search to two additional methods not considering cor-

relations between different features of the Gaussian distributions. As quality comparison method we ran a k-MLIQ search considering axis-parallel Gaussian distributions of the nGMMs, meaning the weights, the location vector, and the standard deviations but not the complete covariance matrix was considered for searching the most similar objects to a query object. This method will be called axis-parallel k-MLIQ search. As additional comparison method, a k-Nearest Neighbor approach, using Euclidean distances of the weighted mean values of the nGMMs as a distance measure was implemented, which will be called k-NN search in the following. For runtime comparison we produced several synthetic data sets, varying four properties of the data, namely the dimension d, the number of objects n in the database, the number of Most Likely Identification Queries or alternatively the number of Nearest Neighbors k, as well as the number of Gaussian distributions per GMM m. For performance comparison we varied the number of MLIQs or NNs k and reported the Area Under the Curve (AUC) for all approaches. As real world examples we used two datasets, one 6 dimensional biometric data set from the BioID Face DB [JKF01] and one 4 dimensional weather data set from Freiburg, Germany. All algorithms are implemented in Java and were run on a 2.4 GHz Intel Core 2 Duo Macintosh computer with 4 GB RAM.

4.5.1 Synthetic Data

Uncertain data for the query and the database objects was generated randomly by choosing mean values to be located between 0 and 100 for each Gaussian distribution of each GMM object. The weights were assigned summing up to 1 within each GMM object. For building the non-axis parallel covariance matrices, we started to choose the standard deviation randomly within a range of 0 to 10. These covariance matrices were rotated in each dimension d using Givens rotations matrices (cf. Eq. 4.28), where the angle of the rotation depended on the cluster the Gaussian distribution was randomly assigned to. Note, that the rotation of the Gaussian distributions

was independent of the GMM it belonged to. The number of angle clusters l was also adjusted depending on the experimental design. Additionally, a randomly chosen variance between 1 and -1 was added to each rotation angle before transforming the covariance matrices as described in Subsection 4.4.4.3.

We generated four data scenarios demonstrating the advantages of our approximation k-MLIQ search compared to the complete k-MLIQ search indicated by a tremendous runtime reduction of the approximation k-MLIQ in comparison to the complete k-MLIQ search. Varying one of the four data properties d, n, k, or m all other properties were kept constant. The default values were $d = 2$, $n = 1,000$, $k = 1$, and $m = 2$. In order to obtain stable results we conducted 10 rounds of k-MLIQ/k-NN queries for each condition with equal property settings. We benchmarked the number of PDV computations in percent as well as the overall runtime for each round and averaged over the 10 rounds.

4.5.1.1 Data Property Benchmarks

We started by varying the dimension ($d = 2, ..., 20$) while keeping n, k, and m constant (Fig. 4.8 a, b). All four methods show increased runtime with increasing dimensionality, while the runtime of the complete k-MLIQ search is 3 to 15 times larger compared to the approximation k-MLIQ search. On average the approximation k-MLIQ search, the axis-parallel k-MLIQ search, and the k-NN search have a subordinate runtime of 7, 40, and 75 fold, respectively, compared to the complete k-MLIQ search. For the approximation k-MLIQ search this is in accordance with the percentage of avoided PDV calculations shown in Fig. 4.8 (b). On average 96 % of the PDV calculations could be saved.

Additional experiments were conducted diversifying the number of objects in the data sets ($n = 1\ 000, ..., 500\ 000$) (Fig. 4.8 c, d). A 13 fold

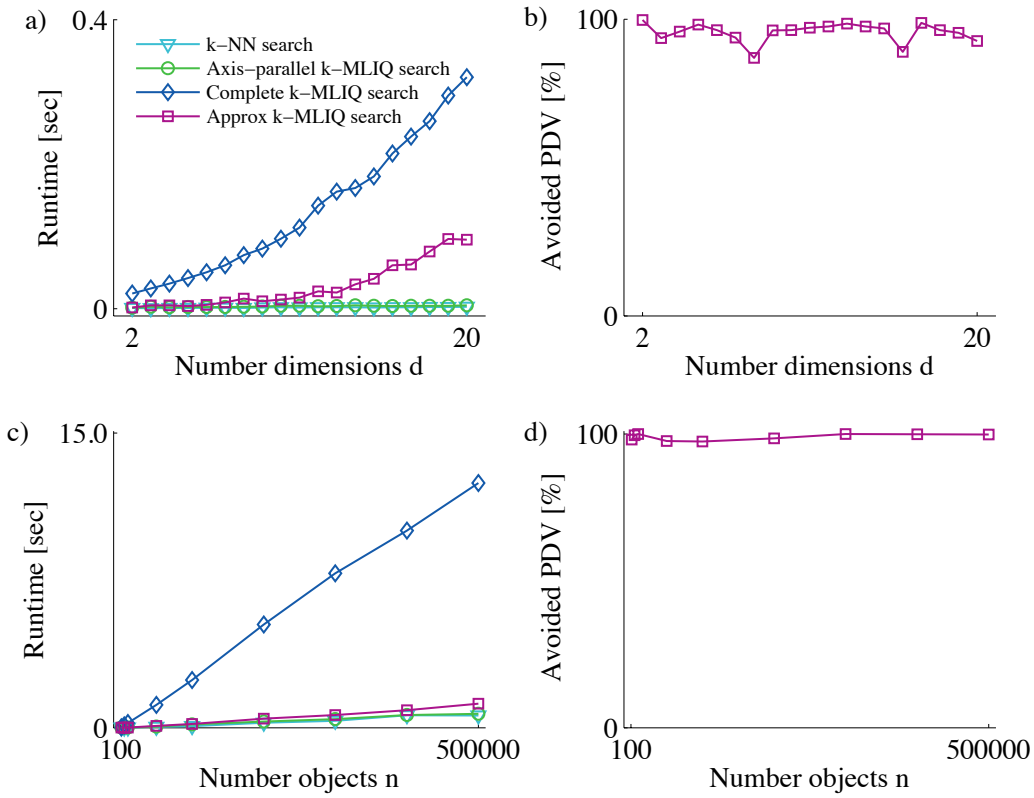


Figure 4.8: Runtime comparison of the approximation k-MLIQ search, the complete k-MLIQ search, an axis-parallel k-MLIQ search, and a k-NN search as well as the percentage of Probability Density Value calculations PDV that could be saved using the approximation k-MLIQ search compared to the complete k-MLIQ search. a), b) Varying the dimension d and c), d) varying the number of objects n.

speed-up in runtime was achieved using the approximation k-MLIQ search, due to the saved comparisons.

Varying the number of Most Likely Identification Queries or Nearest Neighbors $k = 1, ..., 1\ 000$, (Fig. 4.9 e) led to an increase in runtime using the complete k-MLIQ search, the axis-parallel k-MLIQ search, and the k-NN search until approximately two-thirds of the objects were processed, followed by a decrease until $k = n = 1\ 000$. This can be easily explained by

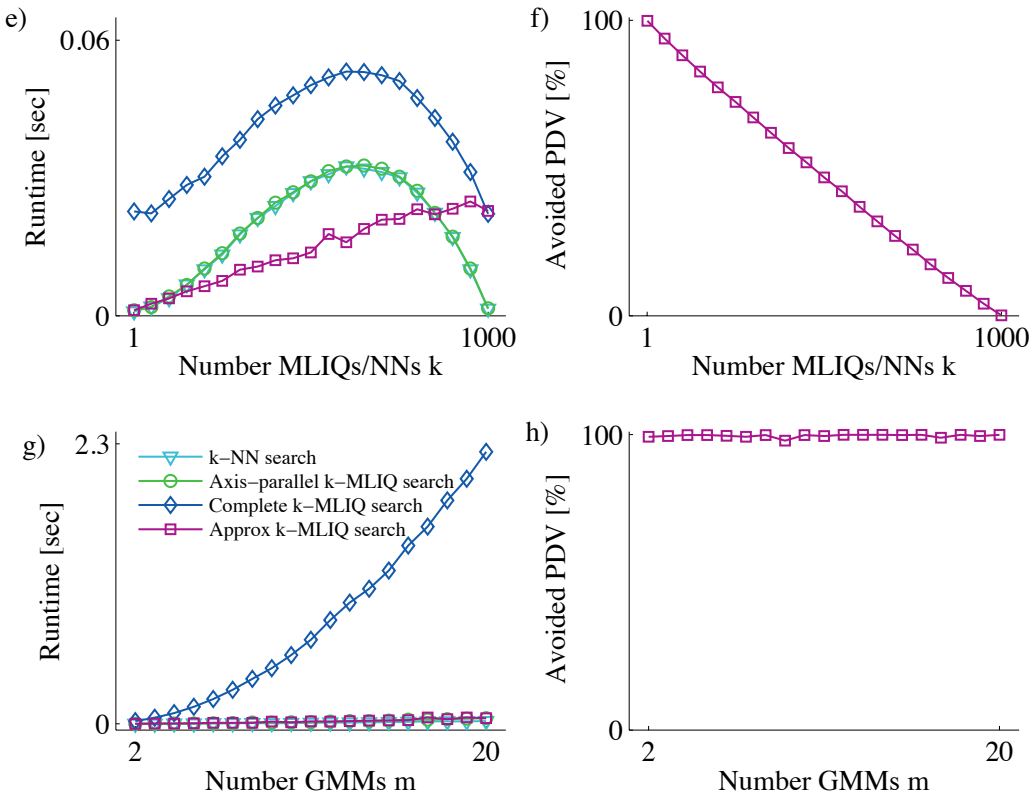


Figure 4.9: Runtime comparison of the approximation k-MLIQ search, the complete k-MLIQ search, an axis-parallel k-MLIQ search, and a k-NN search as well as the percentage of Probability Density Value calculations PDV that could be saved using the approximation k-MLIQ search compared to the complete k-MLIQ search. e), f) Varying the number of MLIQs/NNs k and g), h) varying the number of Gaussians m per nGMM.

the number of sorting operations that had to be executed, each time a new object was detected having a larger probability than the smallest probability in the list of k-MLIQs/k-NNs so far. Once the set of MLIQs/NNs comes closer to the size of the database the sorting has to be executed only rarely, resulting in a runtime decrease. The curve progression of the approximation k-MLIQ search is different, it has an almost constantly rising curve with a smaller slope than all other methods until $k = n = 1\ 000$ where all PDV

calculations have to be executed and, therefore, ending up with the same runtime as the complete k-MLIQ search. With increasing k more PDV calculations are needed as can be seen in Fig. 4.9 (f), therefore, the runtime also increases. On average the approximation k-MLIQ search, the axis-parallel k-MLIQ search, and the k-NN search process the objects with a speed-up factor of 4 compared to the complete k-MLIQ search.

Finally, m the number of Gaussian distributions in a GMM was varied from 2 to 20. The curves of the complete k-MLIQ search compared to the approximation k-MLIQ search, the axis-parallel k-MLIQ search, and the k-NN search in Fig. 4.9 (g, h) are even further apart than in Fig. 4.8 (a, c) and can be explained the same way, having even a larger runtime increase of almost 40 fold using the approximation k-MLIQ search, and the axis-parallel k-MLIQ search, and 70 fold using the k-NN search compared to the complete k-MLIQ search.

4.5.1.2 Performance

Furthermore, we ran a performance comparison between our approximation k-MLIQ search, the axis-parallel k-MLIQ search, and the k-NN search. For method comparison we used Receiver Operator Characteristic (ROC) curves on k-MLIQ and k-NN queries displaying the False Positive (FP) rate on the x-axis and the True Positive (TP) rate on the y-axis. In the context of Most Likely Identification Queries as well as Nearest Neighbor queries, FP rate is defined as the percentage of objects which are predicted positive but actually are negative (FP) among all objects in the database that are actually negative. The TP rate is the percentage of objects that are predicted positive and actually are positive (TP) among all objects in the database that are actually positive.

For data generation we used the same generator as described at the beginning of this section with the default values $d = 2$, $n = 1\,000$, and $m = 2$, varying k between 1 and 1 000 and altogether 10 rounds of k-MLIQ and

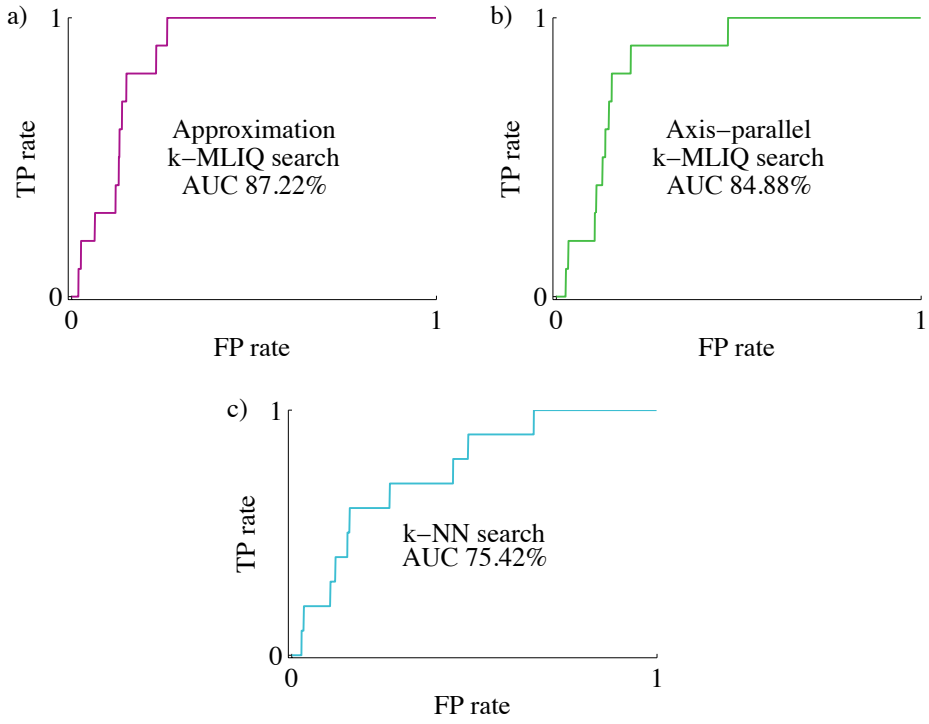


Figure 4.10: Receiver Operating Characteristic curves of a) the approximation k-MLIQ search (magenta), b) the axis-parallel k-MLIQ search (green), and c) the k-NN search (light-blue) using different k values for the MLIQ and NN search. Displayed in the center of the ROC curves are the Area Under the Curve values for each ROC plot.

k-NN queries were averaged in order to obtain stable ROC curves. The AUC of our approximation k-MLIQ search with 87.22% (Fig. 4.10 a) was larger than the AUC of the axis-parallel k-MLIQ search with 84.88 % and the k-NN search with only 75.42 % (Fig. 4.10 b, c) suggesting that the quality of the approximation k-MLIQ search's results is better than the quality of the axis-parallel k-MLIQ search as well as the quality of the k-NN search.

4.5.2 Real World Data

To demonstrate the advantages of our approximation k-MLIQ search in comparison to existing similarity search methods not considering correlation between different features we used two different real world data sets. The first data set was a BioID Face DB [JKF01] which was used for a face recognition task and the second one was meteorologic data [Rad10] which was used for weather forecasting.

4.5.2.1 Biometric Identification

Face recognition is a major area of research within biometric signal processing. In the process of face recognition the localization and identification of human faces in digital images is a fundamental area. Given a series of images for one person with different camera angles, each person can be represented by a variety of correlated distance measures which can be modeled as non-axis parallel GMMs.

We used the BioID Face DB [JKF01] which contains a total of 1521 gray level images to demonstrate the performance of our approximation k-MLIQ search. Thereby, we extracted 6 different distance features of each image in the BioID Face DB. For the 6 dimensional nGMM database we extracted all distances between the left and the right pupil, the right pupil and the nose tip, the left pupil and the nose tip, the right mouth corner and the nose tip, the left mouth corner and the nose tip, and the right and the left mouth corner of each image (cf. Fig. 4.11). To generate nGMMs of the image data we employed the Expectation Maximization algorithm [MP00]. The EM algorithm is a two-step algorithm starting in the first step with a randomly chosen model, and then alternately assigning the data to the individual parts of the model (Expectation step). In the second step the parameters of the model are improved according to the current model assignment (Maximization step). The EM algorithm is a partitioning algorithm with one parameter,

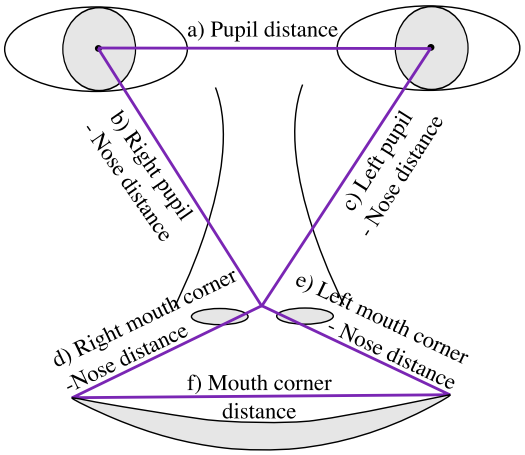


Figure 4.11: Illustration of the features extracted from the BioID Face DB [JKF01]. The 6 different features that we extracted were a) the distances between the left and the right pupil, b) the distances between the right pupil and the nose tip, c) the distances between the left pupil and the nose tip, d) the distances between the right mouth corner and the nose tip, e) the distances between the left mouth corner and the nose tip, and f) the distances between the right and the left mouth corner.

the number of clusters k. To automatically estimate k (in our case number of Gaussian components m) we performed a 10-fold cross validation as implemented in the WEKA package [HFH+09]. Now to generate nGMMs, we used half of the images of each person as input for the EM algorithm and the other half of the images of each person were used to generate a set of query nGMMs.

We performed k-MLIQ/NN searches with our approximation k-MLIQ search, the axis-parallel k-MLIQ search, and the k-NN search varying k from 1 to 5. The number of correctly identified queries is depicted in Fig. 4.12. For all k-MLIQ searches our approximation k-MLIQ search was able to identify more queries correctly than both other methods. The approximation k-MLIQ search was able to identify 100.0% of all queries using a k-MLIQ with $k = 4$, while the axis-parallel k-MLIQ search could only correctly identify 81.0% of the input queries, and the k-NN search could only correctly identify 61.2%

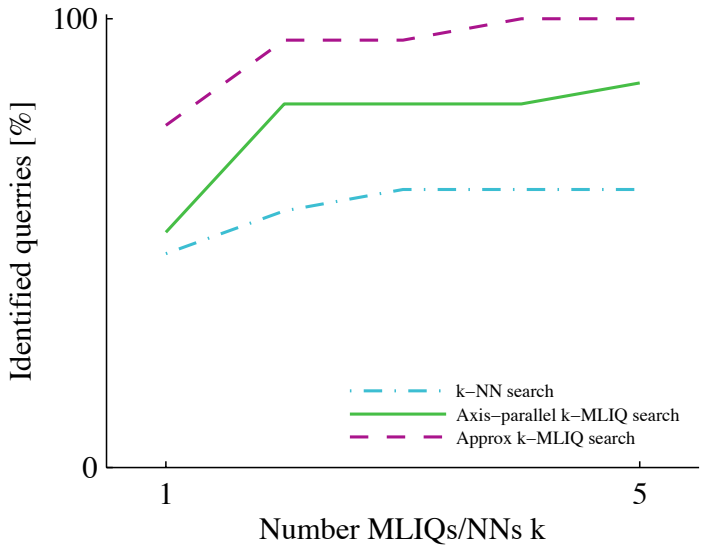


Figure 4.12: Correctly identified queries in % applying the approximation k-MLIQ search, the axis-parallel k-MLIQ search, and the k-NN search on the BioID Face DB [JKF01]. The approximation k-MLIQ search (magenta, dashes) obtained 100.0% accuracy, meaning all objects were found, using an k-MLIQ search with $k = 4$, while the axis-parallel k-MLIQ search (green, solid) as well as the k-NN search (light blue, dash-dots) could only obtain an accuracy of 81.0% and 61.2% with $k = 4$, respectively.

of the queries for $k = 4$. This clearly demonstrates that our approximation k-MLIQ search outperforms the axis-parallel k-MLIQ search and the k-NN search if data features are strongly correlated.

4.5.2.2 Meteorologic Data

Lately the number of natural catastrophes increased all over the world. The natural catastrophes which are most dangerous having the highest number of death counts are hurricanes and tornados but also avalanches, forest fires, flood waves, or floods can cause epidemic plagues, lead to crop loss, or threaten human life. Therefore, weather forecast is an important system for early detection of hazards.

For our weather forecast system we obtained weather data from the website [Rad10] which contains freely available weather data from the city of Freiburg in Germany ranging from January 1^{st}, 2005 until December 31^{st}, 2009. Therefore, the database consisted of 1826 days. The data of one day corresponds to one object in the database, each being represented by a 4-dimensional nGMM, containing the temperature, humidity, barometric pressure, and the wind speed of one day in Freiburg. We obtained the nGMMs of the weather data using again the EM algorithm as described in the previous Subsection. EM is a two-step algorithm consisting of the Expectation and the Maximization step. The output of the EM algorithm was one nGMM for each day.

The goal was to find the MLIQ/NN for one specific day given the weather data of the past 5 years in order to predict the weather of the following day as precise as possible. We randomly selected one day from the year 2010 (March 2^{nd}, 2010) for which we performed a weather forecast. Then, we performed a 1-MLIQ/NN search for the previous day (March 1^{st}, 2010) using the approximation k-MLIQ search, the axis-parallel k-MLIQ search, and the k-NN search. Based on the results we built our weather prediction always using the following day of each hit as source point for predicting the weather of March 2^{nd}, 2010.

The probability P and the distances between March 1^{st}, 2010 and the

Method	Hit	Probability/Distance
Approximation k-MLIQ search	March 6^{th}, 2009	0.372
Axis-parallel k-MLIQ search	Feb. 6^{th}, 2009	0.144
k-NN search	March 11^{th}, 2008	28.386

Table 4.1: The k-MLIQ hit and probability of March 1^{st}, 2010 with $k = 1$ for our approximation k-MLIQ search and the axis-parallel k-MLIQ search and the k-NN hit and distance of March 1^{st}, 2010 with $k = 1$ for the k-NN search.

		Mean (SD)	Min	Max
March 2^{nd}, 2010	Temp. (°C)	2.7 (4.1)	-2.4	11.0
	Humidity (%)	71.7 (14.1)	42.0	86.0
	Barometric p. (hPa)	1019.2 (2.5)	1014.0	1023.0
	Wind speed (km/h)	2.7 (3.7)	0.0	16.0
Approximation k-MLIQ search	Temp. (°C)	5.2 (2.9)	1.9	11.4
	Humidity (%)	69.7 (13.5)	43.0	87.0
	Barometric p. (hPa)	1009.4 (3.1)	1003.0	1013.0
	Wind speed (km/h)	11.2 (6.2)	0.0	30.3
Axis-parallel k-MLIQ search	Temp. (°C)	1.8 (1.1)	-0.3	3.2
	Humidity (%)	90.3 (1.5)	84.0	92.0
	Barometric p. (hPa)	988.6 (1.8)	987.0	994.0
	Wind speed (km/h)	1.3 (2.6)	0.0	12.5
k-NN search	Temp. (°C)	8.4 (1.7)	5.6	12.0
	Humidity (%)	69.0 (11.2)	44.0	91.0
	Barometric p. (hPa)	997.1 (5.6)	991.0	1008.0
	Wind speed (km/h)	37.3 (11.3)	10.5	72.1

Table 4.2: Real mean, standard deviation, minimum, and maximum values of the temperature, humidity, barometric pressure, and wind speed of March 2^{nd}, 2010 as well as the predicted values using the following day of the 1-MLIQ/NN search of the approximation k-MLIQ, the axis-parallel k-MLIQ, and the k-NN search as source point.

MLIQ/NN hit of all three methods are shown in Table 4.1. For the weather prediction of March 2^{nd}, 2010 always the day after the identified MLIQ/NN was chosen. Figure 4.13 (gray bars) shows the frequency of each feature value of March 2^{nd}, 2010, meaning how often each temperature, humidity, barometric pressure, and wind speed value occurred on March 2^{nd}, 2010. To visually compare the weather prediction quality of the approximation k-MLIQ search (magenta), the axis-parallel k-MLIQ search (green), and the k-NN search (light-blue) we plotted their predicted feature value distributions on top of the real distribution. Furthermore, Table 4.2 contains the mean

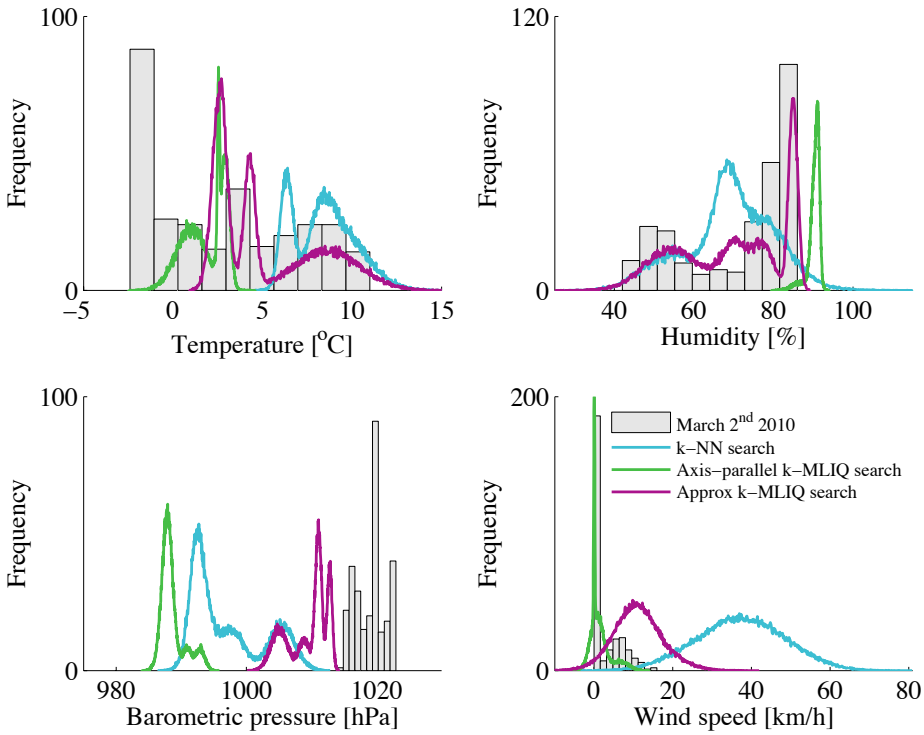


Figure 4.13: Weather prediction for March 2^{nd}, 2010. The gray bars show the real frequency distributions of the temperature, humidity, barometric pressure, and the wind speed of March 2^{nd}, 2010. The colored distributions are the predicted weather distributions of March 2^{nd}, 2010 by the approximation k-MLIQ search (magenta), the axis-parallel k-MLIQ search (green), and the k-NN search (light-blue).

values, standard deviations, minimum, and maximum values of March 2^{nd}, 2010 and the predictions of the three methods.

Based on the data distributions (Figure 4.13) and the mean values (Table 4.2) the approximation k-MLIQ method obtained the best overall results, meaning the mean values and the data distributions were closer to the real values than the predicted values of the other two methods. The approximation k-MLIQ method had a mean temperature difference of 5.7° C, a mean humidity difference of 2 %, a mean barometric pressure difference of 10 hPa,

and a mean wind speed difference of 8.5 km/h. In contrast to that the axis-parallel k-MLIQ search and the k-NN search obtained the following values: mean temperature difference 0.9° C, 5.7° C; mean humidity difference 18.6 %, 2.7 %; mean barometric pressure difference 30.6 hPa, 22.1 hPa, and mean wind speed difference 1.4 km/h, 34.6 km/h. Even though the axis-parallel k-MLIQ search produced slightly better results considering the temperature and the wind speed than the approximation k-MLIQ search, our method was able to score well in all 4 dimensions. Hence, the approximation k-MLIQ search produced the best overall results and would, therefore, be the method of choice for further weather predictions.

4.6 Conclusion

We proposed a new efficient and accurate similarity search for uncertain data. Existing approaches do not consider correlations between different features leading to a loss of information and, therefore, introducing inaccuracy in the search. To overcome this problem we extended the similarity measure to handle very precise Probability Density Functions consisting of non-axis parallel Gaussian Mixture Models. To our knowledge this has not been done so far. Since the calculation of the Mahalonobis distance is very time consuming we introduced a combination of GMM approximation and angle clustering to speed up the procedure while keeping 100 % filter selectivity. Thereby, the angle clustering step minimizes the approximation error by clustering those Gaussian distributions with a similar orientation in space. The newly determined coordinate systems are then used to rotate Gaussians according to their major orientation in space. These rotated, more axis-parallel distributions are subsequently approximated using our filter-refinement architecture.

The conservative approximation in combination with the accurate Mahalonobis distance considering correlations in the similarity search let to more accurate results of our approach compared with methods which ig-

nored correlations thoroughly. This could be demonstrated in our very detailed experimental section including various synthetic and real world data sets. Furthermore, due to our filter-refinement architecture we demonstrated the 100 % filter selectivity resulting in no false dismissals while on average a runtime reduction of 10 fold in comparison with the complete calculation of all exact Mahalonobis distances could be achieved.

Chapter 5

Similarity Search Based Glioma Grading

5.1 Introduction

The proceeding development in medical imaging techniques has accounted to a large amount of high-resolution three-dimensional image data. Especially the high volume of non-invasive measures acquired during clinical routine like structural and functional Magnetic Resonance Imaging (MRI) have revealed new possibilities to study the functioning of the human brain. For the field of brain imaging, data mining techniques have proven to be very useful since the large amount of image data cannot be processed directly due to efficiency reasons. Especially for brain tumors, the use of different structural as well as functional MRI techniques has become a considerable research area, in order to improve non-invasive diagnosis, grading, and post-therapeutic follow-up.

The correct assignment of tumor malignancy is important due to the different prognosis and therapy planning of brain tumors of different histological grades. To date, histological evidence provided by biopsy, being the gold standard for glioma grading, is necessary to amplify the validity of

the diagnosis. However, biopsy requires non-invasive determination of tumor hot spots, since a single tumor mass can be histologically heterogeneous; extracting parts of the tumor that are not representative (sampling error) would hence lead to an incorrect diagnosis and an inadequate treatment [KTE+11]. Furthermore, biopsy implies risks associated with anesthesia and surgery.

Standard MRI protocols for diagnosis of glioma patients mainly rely on the interpretation of contrast enhanced T1-weighted images for tumor grading. Thereby, contrast enhancement is used as an indicator for tumor malignancy. Since some low-grade gliomas show contrast enhancement while a considerable subgroup of high-grade gliomas does not [LYW+03], more sophisticated techniques for glioma grading are needed. Many studies have extracted single features of the structural tumor images like location, volume, size, shape, etc. in order to find relevant features for tumor grading [BJS10, KKK+10, LKK+01, MFS+00]. Some techniques have been proposed trying to classify tumors by considering spatial information of three dimensional tumor Regions Of Interest (ROI) [MDH99, PML+05]. Others have used non-invasive functional dynamic MRI techniques like perfusion MR, Diffusion Tensor Imaging (DTI), or MR spectroscopy in order to find meaningful criteria to improve non-invasive glioma grading [KIN+01, MAA+03, MJSA+04, PMB06, ZWC+09].

Several research groups have demonstrated that perfusion MRI can be used to distinguish between different tumor grades [BJS10, BSW06, LYB+04, LKK+01, PMB06] mainly differentiating between grade II and a combined group of grade III and IV brain tumors. It has been shown that Cerebral Blood Volume reliably correlates with tumor grade and histological findings of increased tumor vascularity [BJS10, BSW06, PP00, SKK+98, WJH+98]. Nevertheless, until now the non-invasive grading of low-grade versus anaplastic glioma has remained very difficult.

The goal of this work was the development of a semi-automatic classifier-based method for differentiating between low-grade (grade I/II) and anaplas-

tic (grade III) gliomas using perfusion-weighted MR imaging in combination with post contrast T1-weighted imaging (T1CE). For data preprocessing we included the outlier detection algorithm described in Chapter 3 and for the similarity search we utilized the algorithm introduced in Chapter 4 considering amongst others also feature correlations for the grading of the tumors. The database used for the similarity search consisted of four-dimensional non-axis parallel Gaussian Mixture Models (GMM), whereat the four dimensions were comprised of three perfusion parameters Cerebral Blood Volume, Cerebral Blood Flow, and Mean Transit Time, as well as the T1CE image. In our approach we considered the entire intensity distribution information embedded in the tumor ROIs in order to render our methodological approach more accurate.

5.2 Grading Of Brain Tumors

Data analysis was performed offline using a combination of existing preprocessing MRI software like Statistical Parametric Mapping (SPM8) as well as self-made software which was combined to a semi-automated workflow (cf. Figure 5.1).

5.2.1 Glioma Grading Based On Similarity Search

In the following, we will give a short overview of the data preprocessing and the classifier-based algorithm, which will be described in more detail in the subsequent subsections. First, a database consisting of perfusion maps and T1CE images of patients with histologically proven grad I through grade III gliomas was build. The general 7-step workflow of the database generation is depicted in Figure 5.1. (1) Perfusion images were corrected for patient's head movement [FWH$^+$96], (2) followed by the generation of perfusion maps (CBF, CBV, MTT) while accounting for contrast agent leakage [BSW06].

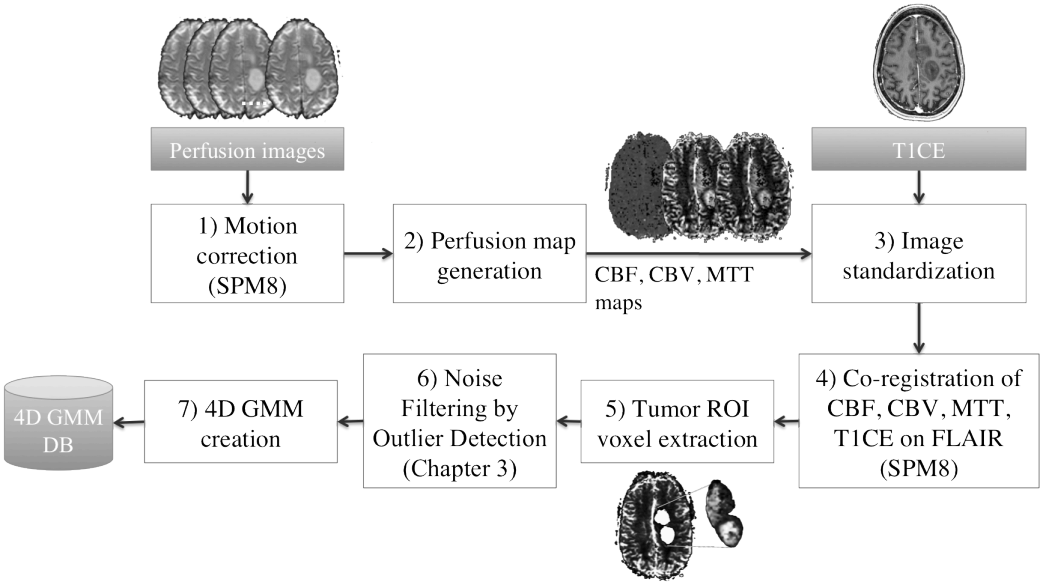


Figure 5.1: Seven-step workflow of the database generation. At first perfusion images have to be corrected for patient motion using the Matlab based SPM8 Toolbox. Based on the functional images perfusion maps (Cerebral Blood Flow, Cerebral Blood Volume, and Mean Transit Time) are generated using self-made software. Perfusion maps and original contrast-enhanced T1-weighted images (T1CE) are standardized in order to be able to compare intensity value between different patients since MRI produces only relative intensity values. To be able to compare different voxels of the different image modalities the perfusion maps (CBF, CBV, and MTT) as well as the T1CE image are co-registered to the FLAIR image using the SPM8 toolbox. After tumor ROIs have been drawn on the FLAIR images the voxel intensity of those tumor ROIs can be extracted. To reduce the noise in the data the tumor intensities are filtered using the Outlier Detection algorithm introduced in Chapter 3. In the last step the 4D GMM are created using an Expectation Maximization algorithm. The generated non-axis parallel GMMs build the database for the k-Most Likely Identification Query search.

(3) Then perfusion maps and T1CE images were standardized in order to obtain comparable intensity scales [NU99]. (4) These standardized images (CBF, CBV, MTT, T1CE) were registered to the FLAIR images using co-registration [CMD+95]. Afterwards, all images were in the same space having comparable intensity values. (5) After drawing tumor ROIs on the FLAIR images, the voxel intensity values contained in the tumor ROI of each image were extracted and (6) subsequently filtered from noise/outliers using our previously introduced outlier detection algorithm (cf. Chapter 3). (7) The four-dimensional intensity values of each patient were converted to a non-axis parallel GMM, using an Expectation-Maximization algorithm [MP00]. This workflow was executed for each patient in the database. It required no user intervention except for the Arterial Input Function (AIF) selection and the tracing of the tumor ROI. The resulting GMMs of all patients constituted the database.

This database built the basis for the similarity search leading to the tumor grading. In order to grade a glioma patient with unknown diagnosis the image data of the new, non biopsy confirmed patient had to be transformed into a non-axis parallel GMM by applying the 7-step architecture mentioned earlier. Next, a k-Most Likely Identification Query [BPS06] using the non-axis parallel GMM of the new patient as input was executed (cf. Chapter 4). The algorithm returned the k most similar patients and their joint probabilities to the unknown patient similar to the k-Nearest Neighbor approach using joint probabilities instead of distances as similarity measure; subsequently, the $TScores$ were calculated giving evidence of the new patients tumor grade. For more details concerning individual parts of the algorithm see the following Subsections.

5.2.1.1 Motion Correction (Step 1)

Since we do not only acquire one single three dimensional brain volume in functional imaging but a series of 3D brain volumes are acquired over time

we have to start by correcting for possible head movement of the patient. The low resolution of the images which has been optimized to highlight contrast agent passage leads to limited anatomical contrast. In order to extract perfusion information from these images it is assumed that the voxel position in the brain is stable meaning the location of the voxels does not change over time. Since there is always some degree of head movement during an acquisition series we have to estimate and correct for head movement. Hence, we used the motion correction routine included in SPM8 implemented in Matlab 7.7 (The MathWorks Inc., Sherborn, MA, USA) [FWH+96] for motion correction. In general one reference image is chosen from within the series of 3D images and all remaining images are registered to this stationary reference image. For motion correction we used the first 3D brain volume of each scanning series to realign all other volumes of the perfusion series to.

5.2.1.2 Perfusion Map Generation (Step 2)

After perfusion data was correctly realigned the parameter maps for CBV, CBF, and MTT had to be obtained based on the work of Ostergaard *et al.* [OWC+96, OSK+96] using self-made software implemented in Matlab 7.7.

For obtaining perfusion parameters we used dynamic susceptibility contrast imaging which can be used to measure intracranial hemodynamics. Using this perfusion weighted MR sequence, perfusion parameters like CBV, CBF, and MTT can be determined. After an intravenous bolus injection of a paramagnetic, non-diffusible contrast agent (gadobenate dimeglumine) the signal intensity change has to be traced over a certain time period using fast MR sequences [Har08]. Our goal was to determine the perfusion parameter values of each voxel in the brain based on the time course illustrating the tracer passage over time. Since the single tracer particles follow a certain path starting from a feeding artery we were able to obtain the transit time, the flow, and the vascular structure of each voxel given the tracer volume of

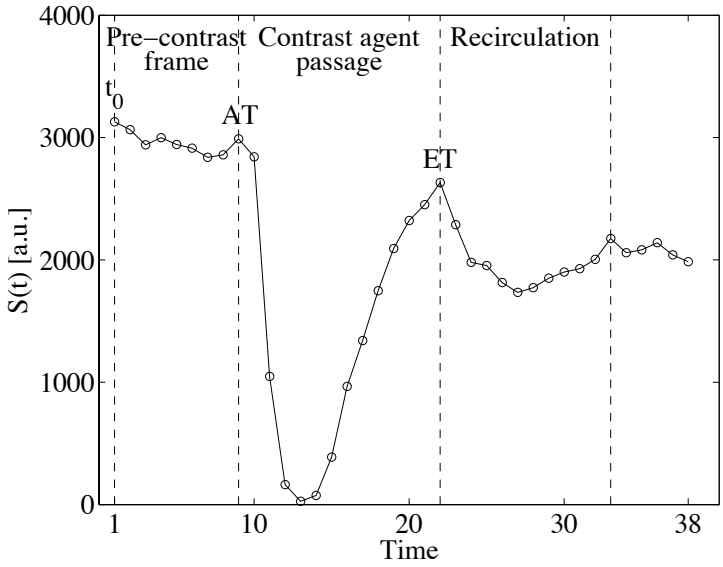


Figure 5.2: Typical dynamic susceptibility contrast imaging voxel intensity time curve. The time from the start of the imaging (t_0) until the arrival time (AT) of the contrast agent (pre-contrast frame) is used to relate the signal intensity $S(t)$ to the contrast agent $C(t)$. The contrast agent passage is characterized by a signal drop in the T2*-weighted image due to the paramagnetic properties of the contrast agent followed by a signal increase. *ET* indicates the last time point before recirculation of the contrast agent takes place indicated by a repeated signal drop. Hence, the signal does not return to the pre-contrast level due to recirculation.

an artery at a starting time point (pre-contrast frame).

Figure 5.2 shows a typical signal intensity curve of a contract agent passage of one voxel from a T2*-weighted MR sequence. Given an intact Blood-Brain-Barrier (BBB) the signal reduction is caused by the contrast agent concentration gradient between the intra- and extra-vascular space. Consequently, when the contrast agent passes a certain brain tissue a decrease in the T2*-weighted images of the perfusion sequences can be observed.

In order to obtain the perfusion parameters we need to determine the intravascular contrast agent time course $C(t)$ for each voxel based on the

original signal time course. Thereby, the relation between the MR signal $S(t)$ and the intravascular contrast agent concentration $C(t)$ at time point t is given by

$$C(t) = -\frac{1}{TE} \log \left(\frac{S(t)}{S(0)} \right) \tag{5.1}$$

where TE corresponds to the echo time and S(0) corresponds to the mean baseline signal

$$S(0) = \frac{\sum_{i=0}^{AT} S(t_i)}{t_{AT} - t_0} \tag{5.2}$$

with t_0 being the start time point of the baseline signal before the presence of the contrast agent but after the magnetization has reached its steady-state. AT being the contrast agent arrival time point [MFH+06]. Hence, the denominator of Eq. 5.1 corresponds to the mean of the pre-contrast frame in Figure 5.2. The resulting contrast agent concentration time signal of the signal intensity time curve of Figure 5.2 is depicted in Figure 5.3.

Since the perfusion parameters are dependent on the contrast agent curve we first had to extract the probability density function describing the passage of the contrast agent before being able to calculate perfusion parameters. For this purpose we used the adaptive total least square gamma-variate fitting method [LTL+03] where the start and stop time points of the contrast agent concentration curve were adaptively determined.

Given the contrast agent passage curve, perfusion parameters can now be determined. Thereby, CBV can be calculated using the following equation [OWC+96, Ost05]

$$CBV = \frac{1}{\rho} \frac{(1 - h_{LV})}{(1 - h_{SV})} \frac{\int C(t)dt}{\int C_{AIF}(t)dt} \tag{5.3}$$

with $h_{LV} = 0.45$ being the hematocrit value for large vessels, $h_{SV} = 0.25$ being the hematocrit value for small vessels and $\rho = 1.04$ g/ml being the brain density. $C_{AIF}(t)$ corresponds to the concentration curve of an arterial input function which was estimated from the signal change of a major artery,

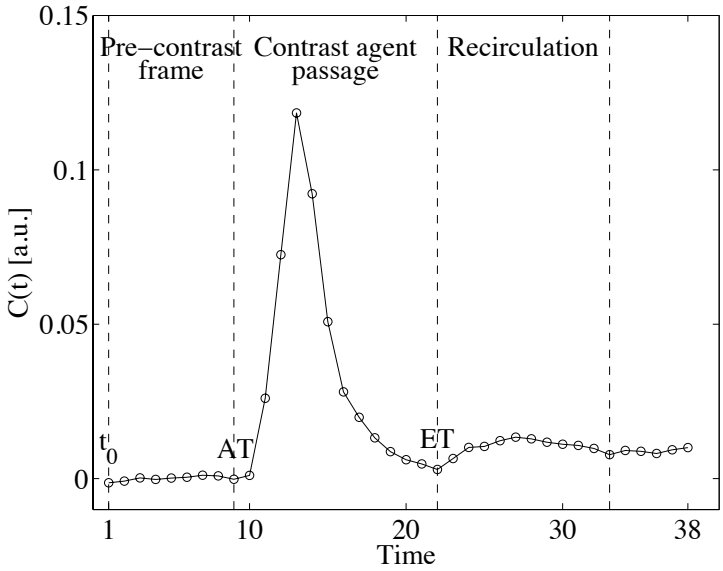


Figure 5.3: The contrast agent concentration $C(t)$ of the signal intensity time course $S(t)$ from Figure 5.2. In contrast to the signal intensity time course the contrast agent passage causes the contrast agent signal to increase. The recirculation can also be seen disabling the signal to return to the pre-contrast baseline level. AT corresponds to the contrast agent arrival time and ET corresponds to the last time point before the recirculation takes place.

which had to be selected manually.

Since we know the time interval between the single time points t given by the repetition time TR we can reformulate Equation 5.3 as follows using trapezoidal integration

$$CBV = \frac{1}{\rho}\frac{(1-h_{LV})}{(1-h_{SV})}\frac{\sum_{i=AT}^{ET}(C(t_i)\cdot TR)}{\sum_{j=AT_a}^{ET_a}(C_{AIF}(t_j)\cdot TR)} \tag{5.4}$$

with AT and ET being the first and last time points of the contrast agent passage of the voxel of interest and AT_a and ET_a being the first and last time points of the contrast agent passage of the AIF. Note that these time point do not have to be equal.

The Cerebral Blood Flow can be determined using a model-independent deconvolution technique since the contrast agent concentration is related to the CBF by the following formula:

$$C(t) = CBF \cdot C_{AIF}(t) \otimes R(t) \tag{5.5}$$

with $R(t)$ being the residue function and $\otimes$ indicating convolution. $R(t)$ is thereby the tissue retention of tracer at time t reflecting the portion of contrast agent in the vasculature after the bolus has been injected. Since the residue function is a decreasing function over time $R(t = 0) = 1$ and $R(t = \infty) = 0$ [Ost05]. To solve equation 5.5 we applied a model-independent linear algebraic deconvolution approach. Therefore, we have to solve the matrix equation

$$C = CBF \cdot \Delta t \cdot A \cdot R \quad \Rightarrow \quad \begin{pmatrix} C(t_{AT}) \\ C(t_{AT+1}) \\ ... \\ C(t_{ET}) \end{pmatrix} =$$

$$= CBF\Delta t \begin{pmatrix} C_{AIF}(t_{AT}) & 0 & \cdots & 0 \\ C_{AIF}(t_{AT+1}) & C_{AIF}(t_{AT}) & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ C_{AIF}(t_{ET}) & C_{AIF}(t_{ET-1}) & \cdots & C_{AIF}(t_{AT}) \end{pmatrix} \begin{pmatrix} R(t_{AT}) \\ R(t_{AT+1}) \\ ... \\ R(t_{ET}) \end{pmatrix} \tag{5.6}$$

in order to obtain CBF. To determine a stable solution for CBF we have to correct for noise present in the signal [Ost05]. This is done by applying Singular Value Decomposition (SVD) to matrix A by

$$A = ULV', \tag{5.7}$$

where the diagonal matrix L contains the singular values of matrix A. Before deconvolution takes place we have to regularize those elements (set to zero) of

matrix L below 20 % of the maximum element in L [LPL+99]. After the noise has been eliminated, we can apply deconvolution. Thereby, CBF corresponds to the maximum of the scaled residue function $R'(t) = CBF \cdot R(t)$ [Ost05].

Since, the standard SVD is strongly dependent on the effects of algorithmic artifacts arising from arterial-tissue delay we calculated the reformulated SVD rather than the standard SVD to correct for those effects [SLTF04]. Thereby, a time shift of the contrast agent signal curve is applied in case of an arterial-tissue delay.

Having obtained the CBV as well as the CBF, we can easily calculate the MTT by means of the central volume theorem [OWC+96]

$$MTT = \frac{CBV}{CBF}. \tag{5.8}$$

The equation for determining the CBV assumes an intact BBB but since we use data from brain tumor patients which do not necessarily have an intact BBB we have to account for possible contrast agent extravasation while calculating the CBV. Thereby, we used the method described by Boxerman *et al.* [BSW06]. The effect of leakage was removed separately for each voxel by adding the estimate of K_2 (reflecting the effects of leakage, being different for each voxel) multiplied by a constant value C, to the CBV_{uncorr} estimate calculated using Eq. 5.4 [BSW06]

$$CBV = CBV_{uncorr} + K_2 \cdot C. \tag{5.9}$$

Subsequently the MTT calculation was also adjusted using the new CBV instead of the CBV_{uncorr}. An example of perfusion maps of one grade III glioma patient can be seen in Figure 5.4.

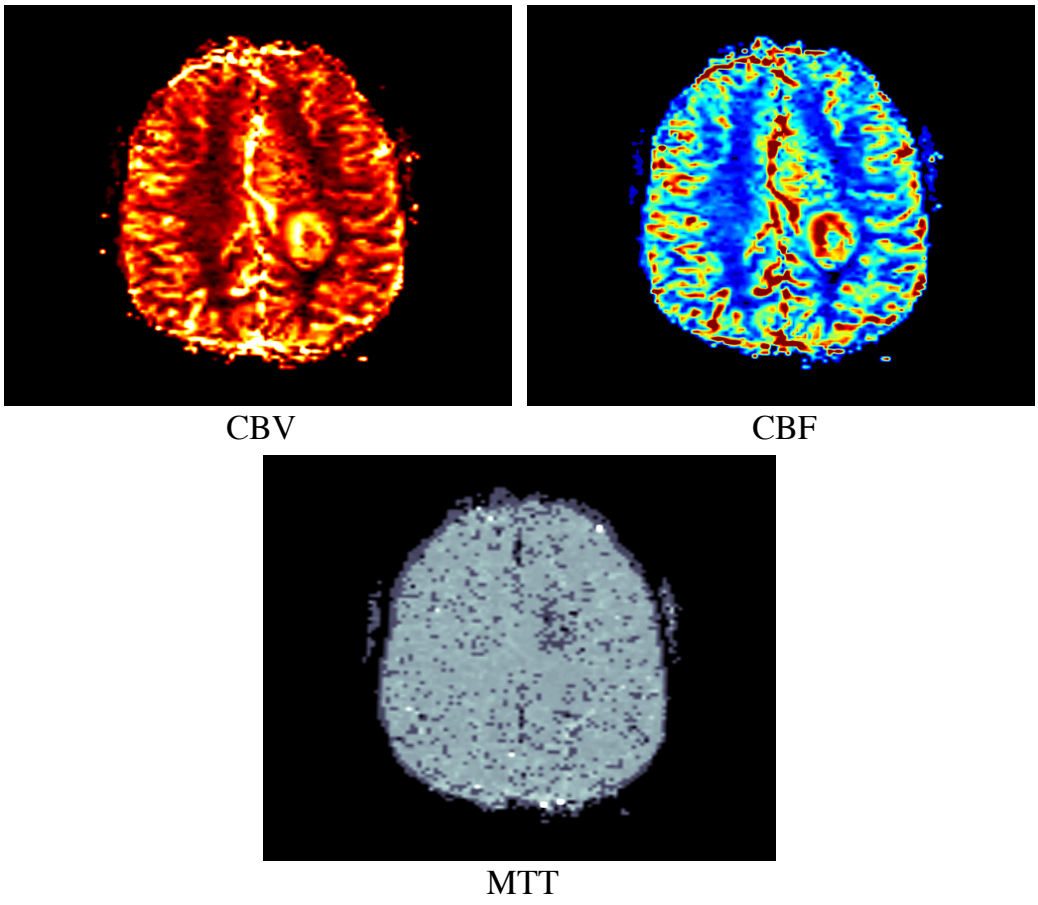


Figure 5.4: An example of the perfusion maps CBV, CBF, and MTT of a grade III brain tumor patient. The tumor is located in the right hemisphere indicated by the bright yellow (CBV image) or the dark red (CBF image) voxels.

5.2.1.3 Image Standardization (Step 3)

Since MR imaging produces only relative signal intensity values leading to inter-patient and inter-study variability, direct intensity value comparison is impossible. To overcome this problem most studies normalize the tumor Regions Of Interest by a contra lateral white matter ROI, but since this method introduces additional operator-dependent errors because contra lateral white

matter ROIs have to be drawn manually we used image standardization instead. This method enables us to directly compare intensity values of different patients of the same MR protocol and body region without the need of a contra lateral white matter ROI. Hence, after standardization similar intensities have similar tissue meaning.

For image standardization a two-step (training and transformation step) piecewise linear transformation method, described by Nyul and colleagues [NU99, NUZ00, MU06, BUB10] was applied. For each of the perfusion maps as well as for the T1CE images, we used 11 landmarks, corresponding to the 0, 10, 20, ..., 80, 90, and 99.8 percentiles $(p_1, ..., p_{11})$ of the intensity distribution of the images and the standardized intensity range was set to $s_1 = 0$ and $s_2 = 50{,}000$ as described by Bedekar *et al.* [BJS10].

In the training step all images of the CBV, CBF, MTT, or T1CE were used, respectively, to determine the landmarks for a mean histogram of unnormalized intensity values. Thereby, the 11 percentile landmarks were determined for each of the images in the training data set, separately and subsequently the mean of these percentile representatives was build.

Starting with the creation of a histogram for an image, intensities at the predefined landmarks were extracted. Then the intensity values between $p_1 = 0^{th}$ percentile and $p_{11} = 99.8^{th}$ percentile were linearly mapped to the minimum and maximum values (s_1 and s_2) of the standardized image range using the following equation [BJS10]

$$p_{x'} = s_1 + \frac{(p_x - p_1)}{(p_2 - p_{11})}(s_2 - s_1), \qquad p_x \in \{p_2, ..., p_{10}\} \tag{5.10}$$

In order to obtain meaningful percentile representatives for each percentile the mean over all images in the training data set is obtained.

In the next step (transformation step) these mean percentile representatives were used to standardize all image scales by mapping the percentiles of the images to the percentile representatives. This was done by mapping

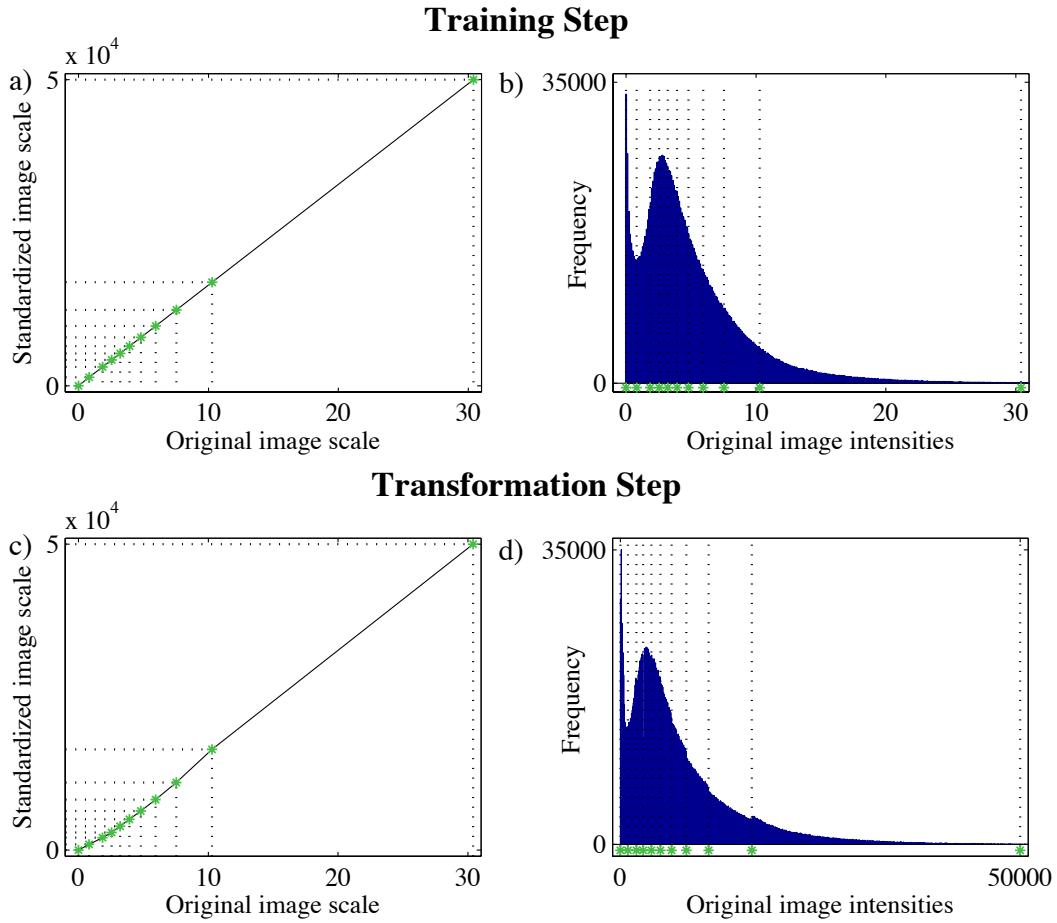


Figure 5.5: Two-step piecewise standardization process example. b) An intensity histogram of an original CBV image. a) The mapping of the 11 percentiles to the standard image scale. This is done for all images of the training set. c) The resulting mean of the standard image scale from the training set is then used to standardize all images in the transformation step. d) The standardized intensity histogram of the image from b. Note that the intensities on the x-axis now range from p_1=0 to p_{11}=50,000. Green stars and dotted lines indicate the 11 percentiles of the intensity histogram.

each interval between the subsequent percentile representatives separately.

Figure 5.5 shows an example of the training and the transformation step using one CBV image to determine the percentiles representatives (a and b).

Then using the mean percentiles obtained by the training of all CBV images the same CBV image is transformed to the standard scale (c and d). Note that the bends in Figure 5.5 c correspond to the adjusted intensity intervals.

5.2.1.4 Co-registration (Step 4)

Our goal was to extract intensity values from a tumor ROI coming from the CBV, CBF, MTT maps, and the T1CE images of each patient. But these images were not produced by the same modalities. For example the post-contrast T1 weighted image has a matrix size of 512 x 512 and a voxel size of 0.5 x 0.5 x 0.6 mm^3, whereas the perfusion sequence produced image volumes with a matrix size of 128 x 128 and a voxel size of 1.8 x 1.8 x 6.5 mm^3. Hence, the number of voxels and the size of the voxels varies rendering direct extraction of intensity values at the same brain location very difficult. To enable voxel extraction at the same location in the brain of different image modalities co-registration was used. Thereby, all image volumes had to be registered to one single target volume.

Since, in the following tumor ROIs will be drawn on Fluid-Attenuated Inversion Recovery (FLAIR) images we decided to co-register all image volumes (CBV, CBF, MTT, and T1CE) to the corresponding FLAIR image of each patient using the co-registration method of SPM8 [CMD+95]. After co-registration, all image volumes had a comparable matrix size of 512 x 512 and a voxel size of 0.4 x 0.4 x 5.5 mm^3 corresponding to the matrix and voxel size of the FLAIR images. Thus by now a voxel in all images corresponded to the same brain region enabling the extraction of tumor intensity values.

5.2.1.5 Tumor ROI Voxel Extraction (Step 5)

All tumor ROIs were drawn by experienced neuroradiologists on FLAIR images using the lesion drawing tool of MRIcron [RKB07], including all voxels identified as solid tumor tissue, but not the voxels interpreted as edema.

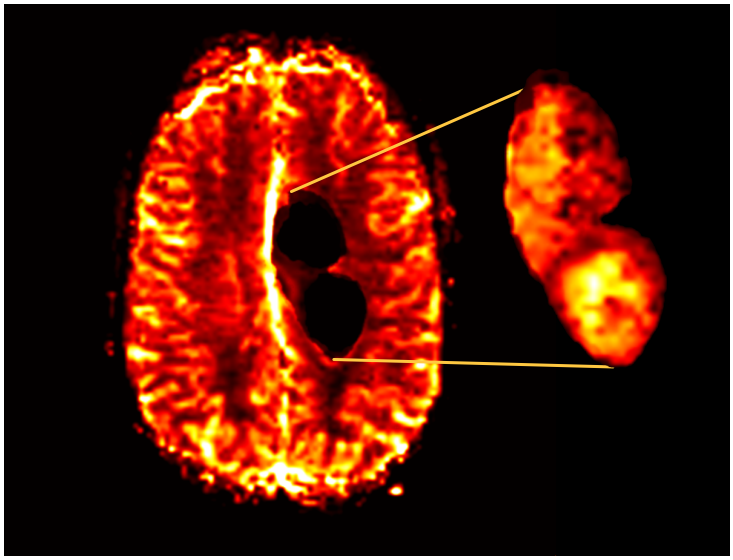


Figure 5.6: An example of a three dimensional tumor ROI extraction of a grade III glioma. Depicted is the CBV image with the extracted tumor ROI.

In unclear cases the T1CE images were additionally visually inspected to further clarify the borders of the tumor ROI. The intensities of the three dimensional tumor ROIs were then extracted from the CBV, CBF, MTT, and T1CE images. An example of a grade III tumor ROI from a CBV image is shown in Figure 5.6.

5.2.1.6 Noise Filtering By Outlier Detection (Step 6)

After tumor ROI extraction each patient was represented by a variety of single voxel values for each of the 4 dimensions. To correct for possible outliers which would disrupt the creation of nGMMs we applied our outlier detection algorithm from Chapter 3, filtering all noise points from the data without the need of any parameter settings.

5.2.1.7 GMM Creation (Step 7)

Our goal was to model the tumor tissue by a Probability Density Function using data of the CBV, CBF, MTT, and T1CE (4 dimensions of our feature space) since it is our hypothesis that tumors of varying grades differ in these distribution functions. In particular, we believe that these distribution functions involve different correlations among the four dimensions. Therefore, the tumor tissue of each patient was modeled as a GMM over the 4-dimensional feature space, because a GMM can fit any finite set of empirical data, and GMMs are simple and general.

A GMM $\mathcal{G}$ is composed of a set of $m = |\mathcal{G}|$ Gaussian distributions. Each Gaussian distribution is defined using the parameters w_i, μ_i, and Σ_i ($1 \leq i \leq m$), with $0 \leq w_i \leq 1$ being the weight of the Gaussian component, $\mu_i = (\mu_{i1}, \ldots, \mu_{id})^T$ being the location parameter vector of a d-dimensional space (in our case composed of the $d = 4$ components CBV, CBF, MTT, and T1CE), and Σ_i being a quadratic $d \times d$ dimensional covariance matrix. The Gaussian distributions can be defined by the following formula:

$$w_i \cdot N(x; \mu_i, \Sigma_i) = w_i \cdot \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left(-\frac{(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)}{2}\right). \quad (5.11)$$

Particularly if two or more of the dimensions exhibit a dependency (which is according to our hypothesis eventually characteristic for the grade of a tumor), Σ_i is not a diagonal matrix but some of the off-diagonal elements are different from zero. Therefore, we will refer to those GMMs as non-axis parallel GMMs (nGMM) in the following to emphasize eventual dependencies. To illustrate a typical distribution function exhibiting correlations, we have depicted a two-dimensional nGMM in the space of MTT and T1CE in Figure 5.7.

The overall PDF of an nGMM $\mathcal{G}$ is defined as a weighted sum of the m

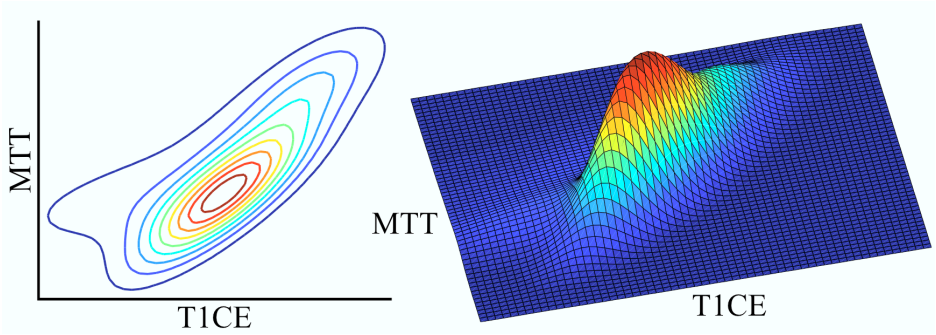


Figure 5.7: Example of a two-dimensional nGMM in the MTT and T1CE space. Correlations are represented by the almost 45° angle of the nGMM in the given example. On the left, the nGMM is depicted as contour plot and on the right the intensities of the two-dimensional nGMM are additionally drawn on the z-axis.

Gaussian distributions

$$f_{\mathcal{G}}(x; w, \mu, \Sigma) = \sum_{i=1}^{m} (w_i \cdot N(x; \mu_i, \Sigma_i)) = \tag{5.12}$$

$$= \sum_{i=1}^{m} \left(\frac{w_i}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left(-\frac{(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)}{2} \right) \right).$$

The weights w_i of each model $\mathcal{G}$ sum up to identity ($\sum_{i=1}^{m} w_i = 1$).

For the creation of the four-dimensional nGMMs the preprocessed, standardized, and co-registered three-dimensional perfusion maps (CBV, CBF, MTT) and the three-dimensional T1CE image of each patient were filtered using the corresponding tumor ROI and the outlier detection algorithm. The resulting intensities of each patient were then transformed into four-dimensional nGMMs. An example of the single channel intensity histograms of one patient in order to see how the data could be distributed is given in Figure 5.8. The mixture model can capture the different subparts of the tumor model as separate components.

For estimating the three parameters w, μ, and Σ of the parametric mix-

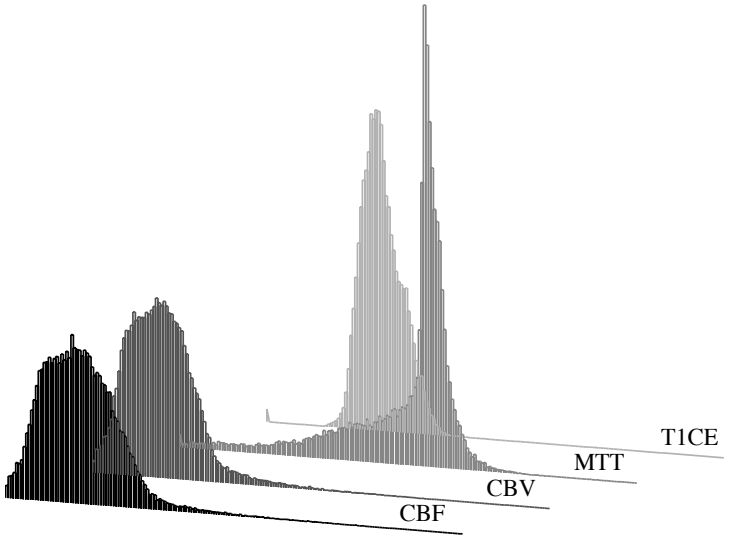


Figure 5.8: Example of single channel intensity histograms of a patients tumor ROI. Each of the single channels builds one dimension in the four-dimensional nGMM.

ture model distribution we employed the Expectation-Maximization algorithm by Geoffrey and David [MP00]. This algorithm is an iterative two-step algorithm. It starts with a randomly chosen model, then alternately the data is assigned to the individual parts of the model (Expectation step) and the parameters of the model are improved according to the current model assignment (Maximization step). To automatically estimate the number of clusters, in our case the number of Gaussian components m of the nGMM, required for the EM algorithm, we performed a 10-fold cross validation as implemented in the WEKA package [HFH+09]. Thereby, starting with all data located in one cluster the number of clusters was increased if the average log-likelihood over all 10 results increased. In case of a log-likelihood decrease or if m was equal to the number of instances in the data set, the model having the maximal log-likelihood was returned. The resulting n nGMMs (one for each patient) were inserted in a database DB, which served as basis for the k-MLIQ search. A result of an EM clustering of one glioma

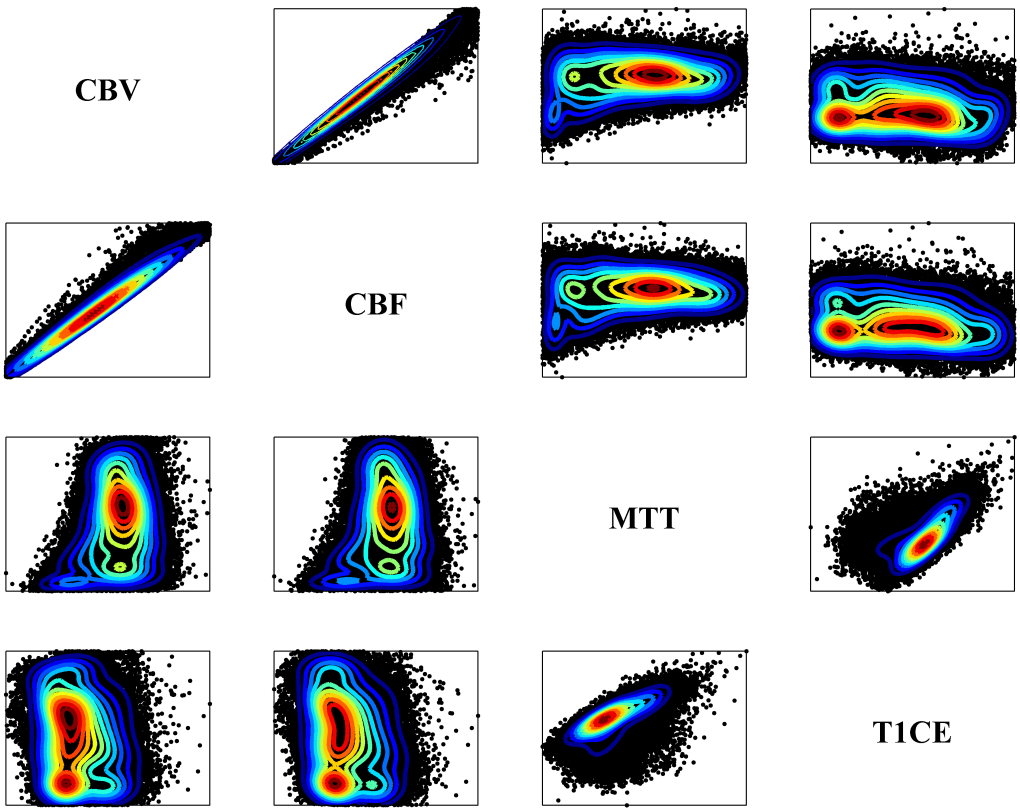


Figure 5.9: Example of a four-dimensional nGMM produced by the Expectation Maximization algorithm. The underlying tumor intensity values (black dots) of the six possible combinations of the four-dimensional data are overlaid by the contour lines of the nGMM function which was returned by the EM algorithm. In this example the CBV and CBF are perfectly correlated shown by the almost 45° angle in the upper left corner image.

patient is shown in Figure 5.9. The contour lines of the four-dimensional nGMM are thereby superposed on the tumor intensity values (black dots). In this example the CBV and the CBF are strongly correlated shown by the almost 45° angle of the contour lines.

5.2.1.8 k-MLIQ Of Non-axis Parallel GMMs

For the k-MLIQ we used the same algorithm as described in Subsection 4.4.5. The original database DB consisted of n nGMMs, one for each patient. We conducted the k-MLIQ to identify the most similar patients in the database to an unknown, non biopsy confirmed query patient. These k-MLIQs are based on the probability that a query object and a database object describe the same object [BPS06]. In general a k-MLIQ search is similar to a k-Nearest Neighbor search, except for the similarity measure, which is used to identify the most similar objects. Nearest Neighbor searches use distance measures like the Euclidean distance whereas an MLIQ search uses probabilities to measure the similarity between two objects.

In order to find a plausible prediction of the tumor grade of a query object we needed a measure of how similar two objects were, called the matching probability. To find the joint probability between one query object $\mathcal{G}'$ and the database objects $\mathcal{G}^* \in DB$ the similarity between $\mathcal{G}'$ and the objects $\mathcal{G}^*$ is measured using the joint Probability Density Value

$$PDV(\mathcal{G}^*, \mathcal{G}') = \int_{\mathbb{R}^d} f_{\mathcal{G}^*}(x; w^*, \mu^*, \Sigma^*) \cdot f_{\mathcal{G}'}(x; w', \mu', \Sigma')\mathbf{d}x. \tag{5.13}$$

This PDV measures how likely it is for a sample x, randomly drawn from the distribution $f_{\mathcal{G}^*}$, to be equal to another sample $x \sim f_{\mathcal{G}'}$. Since all $\mathcal{G}^* \in DB$ are statistically independent from $\mathcal{G}'$ and variances as well as covariances of independent stochastic variables can be summed up [BPS06], PDV can be reformulated as

$$PDV(\mathcal{G}^*, \mathcal{G}') = \sum_{i=1}^{m^*} \left(\sum_{j=1}^{m'} w_i^* w_j' \cdot N(\mu_i^*, \Sigma_i^* + \Sigma_j', \mu_j') \right). \tag{5.14}$$

For a proof of the formula please refer to Subsection 4.4.5.

To find the most likely grading of a new glioma patient $\mathcal{G}'$, we performed

a k-MLIQ in our database of biopsy confirmed pre-classified glioma patients. Using the theorem of Bayes, a k-MLIQ identifies those k glioma patients in the database, which match the PDF of the query patient with the highest probability

$$P(\mathcal{G}^*, \mathcal{G}') = \frac{PDV(\mathcal{G}^*, \mathcal{G}')}{\sum_{i=1}^{n} PDV(\mathcal{G}_i^*, \mathcal{G}')} \tag{5.15}$$

with $|DB| = n$ being the cardinality of the database.

To accelerate the k-MLIQ search we used the conservative nGMM approximation step in combination with the rotation angle clustering as described in Sections 4.4.3 and 4.4.4. We therefore start be rotating and approximating each original Gaussian distribution $\mathcal{G}_i^* \in DB$ leading to a rotated and approximated database DB_{ra}. For the actual k-MLIQ search we used the four-step procedure as described in Subsection 4.4.5.3. Thereby, before searching a query object $\mathcal{G}'$, this object also has to be rotated and approximated as the database objects. Then approximated Probability Density Values PDV_a between the query object $\mathcal{G}'_{ra}$ and all database objects $\mathcal{G}_{ra}^*$ are calculated. Those objects in the database having the largest approximated PDV_a are then used to compute PDVs until an abortion criterium is reached. In the last step absolute probabilities for the objects located in the set of k-MLIQs are calculated.

To obtain a suitable glioma score, which expresses the degree of certainty to predict a glioma of grade I/II vs. a glioma of grade III called $TScore$, we contrast the probabilities of grade I/II glioma patients in the query result $\mathcal{G}^* \in k\text{-MLIQ}_{I/II}$ to those of grade III tumors in the query result $\mathcal{G}^* \in k\text{-MLIQ}_{III}$, according to the formula

$$TScore(\mathcal{G}^*, k-\text{MLIQ}) = \sum_{\mathcal{G}^* \in k-\text{MLIQ}_{I/II} \subseteq k-\text{MLIQ}} P(\mathcal{G}^*, \mathcal{G}') - \sum_{\mathcal{G}^* \in k-\text{MLIQ}_{III} \subseteq k-\text{MLIQ}} P(\mathcal{G}^*, \mathcal{G}'). \tag{5.16}$$

The $TScore$ ranges from 1 (low-grade glioma [WHO grade I/II]) to -1 (high-grade glioma [WHO grade III]). In the following we will call our method

approximation k-MLIQ search.

5.2.2 Comparison Methods

To compare the quality of the approximation k-MLIQ search to existing methods and therefore to demonstrate the superiority of considering correlations between features, we conducted three additional methods. The first method was glioma grading which was based on the presence of contrast enhancement executed by neuroradiologist experts. The second method was a k-MLIQ search not considering correlations which we will call axis-parallel k-MLIQ search and the third one was a k-NN approach using Euclidean distances of the weighted mean values of the nGMMs as a distance measure, which will be called k-NN search in the following.

5.2.2.1 Glioma Grading Based On Contrast Enhancement

Glioma grading based on conventional MRI sequences was performed by two independent neuroradiologists in consensus who were blinded to the histological information. They visually inspected FLAIR as well as pre- and post contrast T1-weighted images. Tumors that showed any pathological contrast enhancement in the contrast-enhanced T1-weighted sequence were diagnosed as high-grade gliomas. In the absence of any contrast enhancement a low-grade glioma was diagnosed.

5.2.2.2 Axis-parallel k-MLIQ Search

To test, whether the non-axis parallel k-MLIQ method is superior to a k-MLIQ method not considering feature correlations a k-MLIQ of axis parallel GMMs was carried out, not considering feature correlations of the GMMs.

Instead of using the entire covariance matrix Σ_i of the Gaussian components, the axis-parallel k-MLIQ method used only the variances σ_i of the

covariance matrices. Therefore, the joint Probability Density Value for the axis-parallel k-MLIQ search can be defined by the following formula:

$$PDV_{ap}(\mathcal{G}^*, \mathcal{G}') = \sum_{i=1}^{m^*} \left(\sum_{j=1}^{m'} w_i^* w_j' \cdot N(\mu_i^*, \sigma_i^* + \sigma_j', \mu_j') \right). \tag{5.17}$$

As for the non-axis parallel method, the theorem of Bayes is used to convert joint PDV_{ap}s to absolute probabilities with $|DB| = n$.

$$P_{ap}(\mathcal{G}^*, \mathcal{G}') = \frac{PDV_{ap}(\mathcal{G}^*, \mathcal{G}')}{\sum_{i=1}^{n} PDV_{ap}(\mathcal{G}_i^*, \mathcal{G}')} \tag{5.18}$$

The $TScore$ is obtained the same way as for the approximation k-MLIQ search using

$$TScore(\mathcal{G}^*, k-\text{MLIQ}) = \sum_{\mathcal{G}^* \in k-\text{MLIQ}_{I/II} \subseteq k-\text{MLIQ}} P_{ap}(\mathcal{G}^*, \mathcal{G}') - \sum_{\mathcal{G}^* \in k-\text{MLIQ}_{III} \subseteq k-\text{MLIQ}} P_{ap}(\mathcal{G}^*, \mathcal{G}'). \tag{5.19}$$

5.2.2.3 k-Nearest Neighbor Search

The k-NN search approach was solely based on the weighted mean values of the Gaussian components. Euclidean distances of these weighted means were used to determine the distances between the tumor patients. In contrast to the k-MLIQ results the k-NN search did not provide any probabilities but rather distances corresponding to the dissimilarity between two objects. Hence, the k objects having the smallest distance were returned by the k-NN search.

To distinguish the predicted grade returned by the k-NN search the most frequently predicted grade in the set of k-NN was chosen.

5.2.3 Quality Measures

In order to receive quality measures for the three presented methods (approximation k-MLIQ search, axis-parallel k-MLIQ search, and k-NN search) we used leave-one-out cross validation, which is a special case of the X-fold cross validation [Koh95] with X = n which is the number of observations (= patients in the database). The validation was performed for each of the n patients by completely excluding the respective patient from the database (resulting in a new database size of (n-1)). The test patient p was then used as query object in the k-MLIQ/k-NN. In total, n = 37 (number of patients in our database) MLIQs/NNs were executed for obtaining quality measures of the methods.

Since the k-MLIQ and the k-NN methods include a parameter k which had to be assigned we executed nested leave-one-out cross validation to adequately evaluate the methods [RHPM04]. Thereby, having excluded the test patient p from the database resulting in a database size of (n-1), each remaining patient had to be excluded to determine the best fitting value for the parameter k. This particular value for k was then assigned to the k-MLIQ/k-NN search for grading patient p. Thus, using nested leave-one-out cross validation we obtained an independent validation set, since the k-MLIQ/k-NN only considered those patients which were presently located in the database. All algorithms concerning the k-MLIQ/k-NN search were implemented in Java and run on a 2.4 GHz Intel Core 2 Duo Macintosh computer.

The accuracy, sensitivity, specificity, positive predicted value (PPV), and negative predicted value (NPV) were calculated for the grading results of all four methods. Tumors which were histologically grade III and subsequently found as grade III tumors, were considered as True Positive findings; grade I/II tumors which were identified as low-grade gliomas and found at histological examination to be grade I/II, were considered as True Negative findings.

For statistical analysis SPSS 18.0 for Macintosh (SPSS Inc, Chicago, IL, USA) was used. To compare the quality of all glioma grading methods, the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve was carried out. After drawing the ROC curves (one for each method) the AUC can be obtained. Thereby, the AUC values can range from 0 to 1, where a diagonal ROC curve (lower left corner to upper right corner) having an AUC of 0.5 (worst possible value) corresponds to random guessing and a AUC of 1 (most desired value) corresponds to a perfect grading result. The considered method had discriminatory power if the curve is significantly different from the diagonal.

5.3 Glioma Grading Results

5.3.1 Patient Population

A total of 37 consecutive adult patients (15 women, 22 men; ranging in age from 22 to 74 years; mean age 44.8 years, SD 14.0 years) with histologically proven grad I to grade III gliomas were included in the study (n = 2 pilocytic astrocytomas, WHO grade I; n = 15 diffuse astrocytomas, WHO grade II; n = 1 ganglioglioma, WHO grade II; n = 1 oligodendroglioma, WHO grade II; n = 4 oligoastrocytomas, WHO grade II; n = 11 anaplastic astrocytomas, WHO grade III; and n = 3 anaplastic oligoastrocytomas, WHO grade III). Tumors were classified and graded according to the World Health Organization classification [DHOW07]. All patients gave their written informed consent to participate in the study prior to the beginning of the MR examination. The institutional review board of the Ludwig-Maximilians University Munich, Germany approved the study protocol.

5.3.2 MR Imaging

MR imaging was performed on a 3 T standard clinical MR scanner (Signa HDx, GE Healthcare, Milwaukee, WI, USA) with 8 receiving channels. The standardized MRI protocol included a pre-contrast T1-weighted sequence (Echo time [TE] = 3.1 ms, repetition time [TR] = 6.9 ms, Field Of View (FOV) = 23 x 18 mm, voxel size = 0.5 x 0.5 x 0.6 mm^3, matrix size = 512 x 512), and a T2 proton density-weighted sequence (TE = 15 ms, TE2 = 130 ms, TR = 3840 ms, FOV = 23 x 18 mm, voxel size = 0.9 x 0.9 x 4.8 mm^3, matrix size = 256 x 256). Axial T2-weighted Fluid-Attenuated Inversion Recovery images were collected with a fast spin-echo readout (TE = 120 ms, TR = 8502 ms, FOV = 22 x 22 mm, voxel size = 0.4 x 0.4 x 5.5 mm^3, matrix size = 512 x 512).

A T2*-weighted Echoplanar Imaging (EPI) multislice sequence was applied for perfusion imaging (TE = 40 ms, TR = 1675 ms, FOV = 23 x 23 mm, voxel size = 1.8 x 1.8 x 6.5 mm^3, matrix size = 128 x 128). Functional perfusion images were obtained approximately 12 seconds before and 50 seconds after a bolus injection of the contrast agent gadobenate dimeglumine (0.2 mmol/kg, Multihance; Bracco Diagnostics, Inc., Princeton, NJ) at an injection rate of 5 ml/s. Post-contrast T1-weighted images were acquired after completion of the perfusion sequence (TE = 3.1 ms, TR = 6.9 ms, FOV = 23 x 18 mm, voxel size = 0.5 x 0.5 x 0.6 mm^3, matrix size = 512 x 512) followed by a T2-weighted navigation sequence (TE = 102 ms, TR = 11 860 ms, FOV = 22 x 18 mm, voxel size = 0.4 x 0.4 x 2.0 mm^3, matrix size = 512 x 512).

In order to run the k-MLIQ and k-NN search algorithms we first had to preprocess the perfusion images of all 37 patients as described above. After preprocessing was finished intensities of tumor ROI voxels were extracted and converted to four-dimensional (CBV, CBF, MTT, T1CE) nGMMs. These nGMMs formed the database for the k-MLIQ/k-NN searches.

5.3.3 Approximation k-MLIQ Search

Our approximation k-MLIQ technique achieved an overall tumor grade prediction accuracy of 83.8 %, which was obtained by comparing the true tumor grade with the predicted tumor grade indicated by a positive or negative TScore. Thereby, the parameter k was chosen by nested leave-one-out cross validation as described in the Methods Section. The detailed results of the 37 approximation k-MLIQ searches are illustrated in Figure 5.10 a. Six out of the 37 patients were falsely graded by the approximation k-MLIQ search indicated by red circles. In detail, 2 anaplasic astrocytomas (WHO grade III) and 1 anaplastic oligoastrocytoma (WHO grade III) were falsely graded as grade II tumors and two diffuse astrocytomas (WHO grade II) as well as one oligoastrocytoma (WHO grade II) were falsely graded as grade III tumors. Hence, the approximation k-MLIQ method showed a sensitivity, specificity, PPV, and NPV of 78.6 %, 87.0 %, 78.6 %, and 87.0 %.

5.3.4 Glioma Grading Based On Conventional MRI Sequences

Based on contrast-enhancement on conventional MRI sequences 13 out of the 37 patients were graded incorrectly: 10 histologically proven low-grade tumors showed contrast enhancement and were graded as grade III gliomas (1 pilocytic astrocytomas, WHO grade I; 4 diffuse astrocytomas, WHO grade II; 1 ganglioglioma, WHO grade II; 1 oligodendroglioma, WHO grade II; and 3 oligoastrocytomas, WHO grade II) and 3 grade III tumors (2 anaplastic astrocytomas, WHO grade III; and 1 anaplastic oligoastrocytoma, WHO grade III) were falsely graded as low-grade gliomas due to the absence of contrast enhancement. Thus, based on contrast enhancement alone the accuracy, sensitivity, specificity, PPV, and NPV were 64.9 %, 78.6 %, 56.5 %, 52.4 %, and 81.3 %, respectively. Three out of the six patients, who were falsely graded by the approximation k-MLIQ search method overlapped with

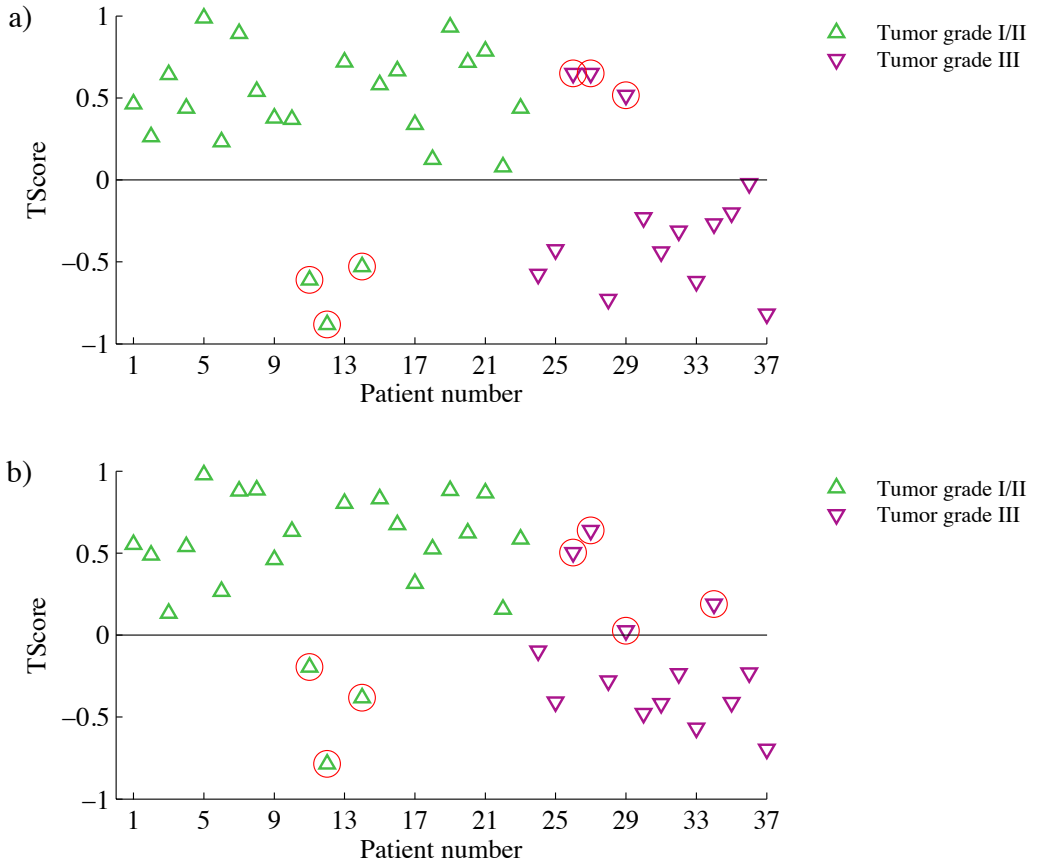


Figure 5.10: $TScore$ results of the 37 tumor patients processed with a) the approximation k-MLIQ search and b) the axis-parallel k-MLIQ search using nested leave-one-out cross validation. Green upward triangles indicate patients with a tumor grade I/II diagnosis and should therefore have a positive $TScore$ and magenta downward triangles denote patients with a tumor grade III diagnosis preferably having a negative $TScore$. Red circles imply wrong graded patients.

the patients falsely graded by the manual method which was solely based on contrast-enhancement (2 diffuse astrocytomas, WHO grade II; 1 anaplasic astrocytoma, WHO grade III).

5.3.5 Axis-parallel k-MLIQ Search

The axis-parallel k-MLIQ search used the same database as the approximation k-MLIQ while ignoring correlations between different features. It achieved an accuracy, sensitivity, specificity, PPV, and NPV of 81.1 %, 71.4 %, 87.0 %, 76.9 %, and 83.3 %, respectively. In total 7 gliomas were falsely classified, including the 6 falsely classified gliomas of the approximation k-MLIQ search, one additional grade III glioma (anaplastic astrocytoma, WHO grade III) was falsely graded as grade II glioma. The $TScore$ results of the axis-parallel k-MLIQ search are shown in Figure 5.10 b.

5.3.6 k-Nearest Neighbor Search

The k-NN search which only considered the weighted means of the database objects obtained an accuracy, sensitivity, specificity, PPV, and NPV of 78.4 %, 66.7 %, 87.0 %, 76.9 %, and 80.0 %, respectively. The k-NN search mis-graded 8 out of the 37 gliomas in the database. Three grade II gliomas (3 diffuse astrocytomas, WHO grade II) were falsely classified as grade III gliomas by the k-NN search and 5 grade III gliomas (5 anaplastic oligoastrocytomas, WHO grade II) were falsely grade as grade II gliomas.

5.3.7 ROC Plot Results

The ROC plots of the k-MLIQ methods' and the k-NN method's grading results as well as the grading by conventional MRI sequences based solely on contrast-enhancement are shown in Figure 5.11. Thereby, the AUC of the grading results produced by the approximation k-MLIQ search were significantly different from random guessing with $p = 0.001$ ($AUC = 0.828$) whereas not being significant using the grading by conventional MRI sequences considering contrast-enhancement alone ($AUC = 0.675$, $p = 0.077$). The grading results of the axis-parallel k-MLIQ search ($AUC = 0.792$, $p = 0.003$) and of

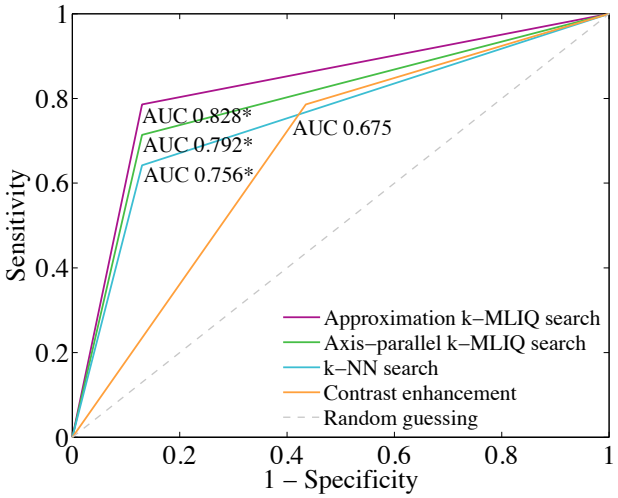


Figure 5.11: The ROC plot of the k-MLIQ method results as well as of the results achieved based on the presence of contrast-enhancement alone. Additionally shown is the Area Under the ROC curve (AUC) of both methods. * AUC significantly different from random guessing with $p \leq 0.01$.

the k-NN search (AUC = 0.756, p = 0.010) were also significantly different from random guessing.

5.4 Discussion

Until now most approaches for glioma grading used only single MRI characteristics of the tumor images like location, size, mean CBV value, etc. [BJS10, KKK+10, LKK+01, MFS+00]. Most existing perfusion MRI studies considered only the CBV of the perfusion parameters for tumor grading. In a study by Zacharaki *et al.* [ZWC+09] 161 features including different shape, intensity, and gabor texture features of tumor ROIs were used to classify brain tumors. Authors applied feature selection to reduce the number of classification dimensions. Relative CBV (rCBV) and T1CE were located in the set of top-ranked features to separate grade II from grade III patients leading to an accuracy of 75 %. Shin *et al.* [SLK+02] examined relative CBF

(rCBF) in addition to the rCBV of glioma patients, achieving promising results. Using an rCBF ratio cutoff of 3.57 they obtained a sensitivity of 72.7 % and a specificity of 100 % for separating low-grade from high-grade gliomas. Since, CBV, CBF, and T1CE already demonstrated to be good indicators for different tumor grades, we decided to integrate the complete perfusion information including CBV, CBF, and MTT in combination with the T1CE information in our classifier-based method, which has to our knowledge not yet been done.

Opposed to most existing studies, our classifier-based method integrates the information of the whole tumor volumes. There are three major advantages of this approach: 1) It considers the heterogeneity of an individual glioma by using the distribution of the different perfusion parameters, and not only the mean value measured in a single ROI. 2) It does not depend on a cutoff value for perfusion parameters. In existing studies, which used rCBV ratios for separating low-grade from high-grade gliomas, the rCBV ratio cutoff values ranged from 1.5 to 2.9 rendering a clear separation without prior knowledge of the data very difficult [CRL04, LYW+03, LKK+01, SLK+02]. 3) The consideration of feature correlations between the CBV, CBF, MTT, and T1CE in the nGMMs resulted in a more accurate method to determine the tumor grade without the need of prior knowledge of the underlying data. The performance of our method with an accuracy, sensitivity, and specificity of 83.8 %, 78.6 %, and 87.0 %, respectively without the need of a cutoff value, which would probably change using a different set of patients, is a major advantage of our approach.

One reason for the large variability in previous studies considering the rCBV cutoff value could be the lack of a standard image intensity scale in MRI. MRI produces only relative signal intensity values, hence images of a patient acquired at different time points or images of different patients cannot be compared directly. A common approach to handle this problem is to compare signal intensity values in a given ROI with those measured in a contra lateral white matter ROI. However, using a reference ROI in con-

tra lateral tissue in order to produce rCBV ratios introduces an operator dependent variability. To overcome this problem we applied image intensity standardization [BJS10, NU99], which standardizes the image intensity scale of one specific MR protocol and body region in a two-step architecture. Hence, a contra lateral reference ROI is not required anymore, enabling the direct inter-patient ROI comparison and, thus eliminating a possible error source. Furthermore, since in our approach nGMMs are automatically determined by EM clustering using exclusively the underlying intensity values of the tumor ROIs, even subtle intensity changes, which are undetectable by visual inspection, will be included in the automatic analysis.

The extraction and approximation of the complete tumor ROI leads to the loss of spatial information. Thus, with the current input information, our approach does not allow the identification of tumor hot spots, i.e. the part with the highest malignancy, for biopsy planning [KTE+11]. However, in further studies the spatial aspect could be incorporated in the grading procedure for example by matching hot spot ROIs of Positron Emission Tomography (PET) images to MRI data. Another drawback of the study is the small and heterogeneous patient population with several histological diagnoses. One might wonder how e.g. pilocytic astrocytomas can be correctly graded as low-grade gliomas by our approach even though they strongly differ from other glioma types present in our database both histologically as well as on conventional MRI sequences. As mentioned earlier our method is solely based on similarities between a respective test glioma and the tumors building the database. Hence, it seems plausible that perfusion characteristics of pilocytic astrocytomas are more similar to grade II gliomas than to grade III gliomas thus leading to a correct grading by our method. Nevertheless, the lack of a larger dataset disables the algorithm to differentiate between different histological diagnoses like between oligodendroglioma (1 patient) and diffuse astrocytoma (15 patients). Thus, since the prediction of our algorithm is based on the underlying information of the database, a larger dataset would most probably further increase the prediction accuracy of our algorithm. If more

data (patients with biopsy confirmed diagnosis) is included in the database the chances of finding more accurate and suitable hits to an unknown patient also rise which leads to a higher grading accuracy. By including additional grade I and also glioblastoma (GBM) patients in the database our method could be extended to also distinguish between gliomas of grade I, II, III, and IV. Furthermore, the incorporation of Diffusion Tensor Imaging and MR spectroscopy data, as well as PET data which have been shown to be reliable tumor grade indicators [HOK+09, LYW+03, MAA+03, PKM+07], as additional dimensions in the nGMM should further increase the accuracy of our approach.

5.5 Conclusion

In this Chapter we combined the outlier detection and the similarity search in uncertain data a propose our classifier-based technique for glioma grading, which is based on a k-MLIQ search. The approach is almost fully automatic, except for tracing the tumor ROI and selecting the AIF for the generation of the perfusion maps. The database for the k-MLIQ search was composed of four-dimensional nGMMs comprised of the entire ROI intensity distribution of the CBV, CBF, MTT, and T1CE images also considering attribute correlations. Our approach had a glioma grading accuracy of 83.8 %, which was significantly better than random guessing, whereas glioma grading based on contrast-enhancement alone (accuracy of 64.9 %) was not significantly different from random guessing. Even though the axis-parallel k-MLIQ search and the k-NN search also achieved significantly better results than random guessing they were not able to reach equally good grading results as our approach. Therefore, incorporating feature correlations and the entire perfusion information in our similarity search, which is exclusively based on finding similarities between different gliomas, gives a great benefit to our method.

Chapter 6

Conclusion

6.1 Summary

In the field of data mining especially in mining medical images there are many difficulties that have to be faced. Starting with the preprocessing of the data including the removal of noise or outliers the image data has to be adequately prepared in order to be able to gain as much information as possible. After the preprocessing has been finished several different data mining techniques can be applied for knowledge discovery. The methods and concepts presented in this work contribute to the field of knowledge discovery in medical imaging. This chapter provides a summary of the major contributions of this work.

6.1.1 Introduction And Algorithmic Fundamentals

Chapter 1 starts with a motivation for this work, constituting the major topic that is going to be dealt with. After the motivation the algorithmic fundamentals to this work are delineated in Chapter 2 starting with an overview to existing information theoretic measures which can be used to facilitate the abolishment of parameters in several data mining fields. Furthermore, the

prominence of uncertain data in data mining is described since the remainder of the work is heavily based on the handling of uncertain data. Then the field of clustering, outlier detection, and classification are introduced since in this work we do not only propose a novel outlier detection method which is based on the cluster definition and other useful principles of existing clustering algorithms but we also introduce a new classification method including parts of a cyclic clustering algorithm.

6.1.2 Parameter-Free Outlier Detection Using Data Compression

Various application fields like e.g. economy, biology, or medicine have to deal with the automatic identification of outstanding observations in data sets also known as outliers. In Chapter 3 we introduced our new parameter-free outlier detection approach which is based on a very general data model. This new approach thereby overcomes two of the major problems most existing outlier detection approaches suffer from. The first problem that has to be dealt with is parametrization. The outcome of most outlier detection approaches strongly depends on the accurate selection of parameters like the number of outliers present in the data or the minimum size of clusters. In order to adequately choose these parameters, for most real world data sets, the expertise of domain experts as well as background knowledge on the underlying data is required. The second problem that has to be faced is the restriction to a specific data distribution like a Gaussian or Uniform distribution. In our presented parameter-free outlier detection approach we were able cope with these problems.

The basic idea of our technique relates outlier detection to data compression, thereby outliers are objects which can not be effectively compressed given the rest of the data set. To avoid the assumption of a certain data distribution e.g. a Gaussian or Uniform distribution, our outlier approach relies

on a very general data model combining the Generalized Normal Distribution with Independent Components. We define an intuitive outlier factor based on the principle of the Minimum Description Length together with a novel algorithm for outlier detection. An extensive experimental evaluation on synthetic and real world data demonstrates the benefits of our technique.

6.1.3 Similarity Search In Uncertain Data

Efficient similarity search in uncertain data is a central problem in many modern applications such as biometric identification, stock market analysis, or medical imaging. In such applications, the feature vector of an object is not exactly known but is rather defined by a Probability Density Function like a Gaussian Mixture Model. Previous work is limited to axis-parallel Gaussian distributions, hence, correlations between different features are not considered in the similarity search.

In Chapter 4, we proposed our novel, efficient similarity search technique for general GMMs without independence assumption for the attributes, which approximates the actual components of a GMM in a conservative but tight way. For minimizing the approximation error we clustered Gaussian distributions with the same major orientation in space before approximating the Gaussian distributions. The filter-refinement architecture guarantees no false dismissals, due to conservativity, as well as a good filter selectivity, due to the tightness of our approximations. An extensive experimental evaluation of our approach demonstrates a considerable speed-up of similarity queries on general GMMs and an increase in accuracy compared to existing approaches.

6.1.4 Similarity Search Based Glioma Grading

Adequate therapy planning of gliomas needs histological determination of invasive biopsy due to the fact that both treatment and prognosis of glial neo-

plasms vary strongly depending on their histological grading. Magnetic Resonance Imaging based glioma grading is currently mainly based on contrast-enhanced T1-weighted images. To additionally gain information on tumor physiology for glioma grading functional Magnetic Resonance Imaging techniques like e.g. perfusion MR have also been considered.

In Chapter 5, we presented our novel technique for glioma grading, which combines our outlier detection approach with the similarity search for uncertain data. After preprocessing the tumor data using a combination of existing and self-made image processing software (e.g. perfusion map generation, outlier detection, etc.) we performed a similarity search on the tumor data. In order to perform the search as accurate as possible we used four different features of the tumors as input for the similarity search: the Cerebral Blood Volume, the Cerebral Blood Flow, the Mean Transit Time, and the post-contrast T1-weighted image. For each patient a so-called tumor feature vector was defined by a four-dimensional Probability Density Function, more precisely a Gaussian Mixture Model. In contrast to existing similarity searches we also considered correlations between different features in the similarity search. Applying our approach to MRI data sets of 37 glioma patients (23 grade I/II, 14 grade III gliomas), which were preprocessed and converted to four-dimensional GMMs, we achieved an accuracy, sensitivity, and specificity of 83.8 %, 78.6 %, and 87.0 % while grading based solely on contrast-enhancement could only achieve an accuracy, sensitivity, and specificity of 64.9 %, 78.6 %, and 56.5 %, respectively. Hence, our proposed similarity search based grading technique is of great value for supporting non-invasive tumor grading since it integrates the information of different MRI sequences and perfusion maps in one semi-automatic analysis.

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

AIC	Akaike Information Criterion
AIF	Arterial Input Function
AMI	Adjusted-for-chance Mutual Information
APG	Assists Per Game
AUC	Area Under the Curve
BBB	Blood-Brain-Barrier
BIC	Bayesian Information Criterion
CBF	Cerebral Blood Flow
CBV	Cerebral Blood Volume
CLARANS	Clustering Large Applications based on Randomized Search
CT	Computer Tomography
CURE	Clustering Using Representatives
DB	Database
DBSCAN	Density-Based Spatial Clustering of Applications with Noise

DTI	Diffusion Tensor Imaging
EM	Expectation Maximization
EPD	Exponential Power Distribution
EPI	Echoplanar Imaging
FLAIR	Fluid Attenuated Inversion Recovery
FN	False Negative
FOV	Field Of View
FP	False Positive
GBM	Glioblastoma
GMM	Gaussian Mixture Model
GND	Generalized Normal Distribution
GP	Games Played
ICA	Independent Component Analysis
k-MLIQ	k-Most Likely Identification Query
k-NN	k-Nearest Neighbor
LOCI	Local Outlier Correlation Integral
LOF	Local Outlier Factore
LoOP	Local Outlier Probability
MAP	Maximum A Posteriori
MD	Mahalonobis Distance

MDEF	Multi-granularity Deviation Factor
MDL	Minimum Description Length
MFCC	Mel-Frequency Cepstral Coefficient
MI	Mutual Information
MMH	Maximal Margin Hyperplane
MRI	Magnetic Resonance Imaging
MTT	Mean Transit Time
NBA	National Basketball Association
nGMM	non-axis parallel Gaussian Mixture Model
NMI	Normalized Mutual Information
NPV	Negative Predicted Value
OCI	Outlier-robust Clustering using Independent components
PAM	Partitioning Around Medoids
PCA	Principle Component Analysis
PDF	Probability Density Function
PDV	Probability Density Value
PET	Positron Emission Tomography
PPG	Points Per Game
PPV	Positive Predicted Value
rCBF	relative CBF

rCBV	relative CBV
RIC	Robust Information-Theoretic Clustering
ROC curve	Receiver Operator Characteristic curve
ROI	Region Of Interest
RPG	Rebounds Per Game
SPM	Statistical Parametric Mapping
SVD	Singular Value Decomposition
SVM	Support Vector Machines
T1CE	Post contrast T1-weighted imaging
TE	Echo time
TN	True Negative
Top k-PNN	Top k-Probable Nearest Neighbor
TP	True Positive
TR	Repetition Time
WHO	World Health Organization

Bibliography

[Ait84] J. Aitchison, *Reducing the dimensionality of compositional data sets*, Mathematical Geology **16** (1984), no. 6, 617–635.

[Aka74] H. Akaike, *A new look at the statistical model identification*, IEEE Transactions on Automatic Control **19** (1974), no. 6, 716–723.

[BBK01] C. Böhm, S. Berchtold, and D.A. Keim, *Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases*, ACM Comput. Surv. **33** (2001), no. 3, 322–373.

[BFF96] B. Bollobás, T.I. Fenner, and A.M. Frieze, *On the best case of heapsort*, Journal of Algorithms **20** (1996), no. 2, 205–217.

[BFP08] C. Böhm, C. Faloutsos, and C. Plant, *Outlier-robust clustering using independent components*, Proceedings of the ACM SIGMOD Conference on Management of Data, 2008, pp. 185–198.

[BFPP06] C. Böhm, C. Faloutsos, J.Y. Pan, and C. Plant, *Robust information-theoretic clustering*, Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), 2006, pp. 65–75.

[BFSO84] L. Breiman, J. Friedman, C. Stone, and R.A. Olshen, *Classification and regression trees*, 1 ed., Wadsworth & Brooks/Cole Advanced Books & Software, 1984.

[BGK+07] C. Böhm, M. Gruber, P. Kunath, A. Pryakhin, and M. Schubert, *Prover: Probabilistic video retrieval using the gauss-tree*, Proceedings of the International Conference on Data Engineering (ICDE), 2007, pp. 1521–1522.

[BGMP92] D. Barbara, H. Garcia-Molina, and D. Porter, *The management of probabilistic data*, IEEE Transactions on Knowledge and Data Engineering **4** (1992), no. 5, 487–502.

[BJS10] D. Bedekar, T. Jensen, and K.M. Schmainda, *Standardization of relative cerebral blood volume (rcbv) image maps for ease of both inter- and intrapatient comparisons*, Magn Reson Med **64** (2010), no. 3, 907–913.

[BKK96] S. Berchtold, D.A. Keim, and H.P. Kriegel, *The X-tree: An index structure for high-dimensional data*, Proceedings of the International Conference on Very Large Databases (VLDB) (San Francisco, U.S.A.) (T. M. Vijayaraman, Alejandro P. Buchmann, C. Mohan, and Nandlal L. Sarda, eds.), Morgan Kaufmann Publishers, 1996, pp. 28–39.

[BKNS00] M. Breunig, H.P. Kriegel, R. Ng, and J. Sander, *Lof: Identifying density-based local outliers*, Proceedings of the ACM SIGMOD Conference on Management of Data, 2000, pp. 93–104.

[BPS06] C. Böhm, A. Pryakhin, and M. Schubert, *The gauss-tree: Efficient object identification in databases of probabilistic feature vectors*, Proceedings of the International Conference on Data Engineering (ICDE), 2006, p. 9.

[BSI08] G. Beskales, M. Soliman, and I. IIyas, *Efficient search for the top-k probable nearest neighbors in uncertain databases*, Proc. VLDB Endow. **1** (2008), no. 1, 326–339.

[BSW06] J.L. Boxerman, K.M. Schmainda, and R.M. Weisskoff, *Relative cerebral blood volume maps corrected for contrast agent extravasation significantly correlate with glioma tumor grade, whereas uncorrected maps do not*, AJNR Am J Neuroradiol **27** (2006), no. 4, 859–867.

[BUB10] U. Bagci, J. Udupa, and L. Bai, *The role of intensity standardization in medical image registration*, Pattern Recognition Letters **31** (2010), no. 4, 315–323.

[CBK09] V. Chandola, A. Banerjee, and V. Kumar, *Anomaly detection: A survey*, ACM Computing Surveys **41** (2009), no. 3, 1–58.

[CCK05] M. Chau, R. Cheng, and B. Kao, *Uncertain data mining: A new research direction*, Proceedings of the Workshop on the Sciences of The Artificial (WSA) (Hualien, Taiwan), 2005.

[CH67] T. Cover and P. Hart, *Nearest neighbor pattern classification*, IEEE Transactions on Information Theory **13** (1967), no. 1, 21–27.

[CKP03] R. Cheng, D. Kalashnikov, and S. Prabhakar, *Evaluating probabilistic queries over imprecise data*, Proceedings of the ACM SIGMOD Conference on Management of Data, 2003, pp. 551–562.

[CKP04] R. Cheng, D.V. Kalashnikov, and S. Prabhakar, *Querying imprecise data in moving object environments*, IEEE Transactions on Knowledge and Data Engineering **16** (2004), no. 9, 1112–1127.

[CMD+95] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal, *Automated multi-modality image registration based on information theory*, Information Processing in Medical Imaging **3** (1995), 263–274.

[Com94] P. Comon, *Independent component analysis, a new concept?*, Signal Processing **36** (1994), no. 3, 287–314.

[CRL04] D.J. Covarrubias, B.R. Rosen, and M.H. Lev, *Dynamic magnetic resonance perfusion imaging of brain tumors*, Oncologist **9** (2004), no. 5, 528–537.

[CV95] C. Cortes and V. Vapnik, *Support-vector networks*, Machine Learning **20** (1995), no. 3, 273–297.

[CXP+04] R. Cheng, Y. Xia, S. Prabhakar, R Shah, and J.S. Vitter, *Efficient indexing methods for probabilistic threshold queries over uncertain data*, Proceedings of the International Conference on Very Large Databases (VLDB), 2004, pp. 876–887.

[Def77] D. Defays, *An efficient algorithm for a complete link method*, The Computer Journal **20** (1977), no. 4, 346–366.

[DH73] R.O. Duda and P.E. Hart, *Pattern classification and scene analysis*, John Wiley & Sons Inc, 1973.

[DHOW07] L. David, O. Hiroko, W. Otmar, and C. Webster, *Who classification of tumours of the central nervous system*, 4 ed., International Agency for Research on Cancer (IARC), Lyon, 2007.

[DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin, *Maximum likelihood from incomplete data via the em algorithm*, Journal of the royal statistical society **39** (1977), no. 1, 1–38.

[DMJRM00] R. De Maesschalck, D. Jouan-Rimbaud, and D. Massart, *The mahalanobis distance*, Chemometrics and Intelligent Laboratory Systems **50** (2000), no. 1, 1–18.

[DS05] N.N. Dalvi and D. Suciu, *Answering queries from statistics and probabilistic views*, Proceedings of the International Conference on Very Large Databases (VLDB), 2005, pp. 805–816.

[DSBHW06] A. Das Sarma, O. Benjelloun, A.Y. Halevy, and J. Widom, *Working models for uncertain data*, Proceedings of the International Conference on Data Engineering (ICDE), 2006, p. 7.

[EKSX96] M. Ester, H.P. Kriegel, J. Sander, and X. Xu, *A density-based algorithm for discovering clusters in large spatial databases with noise.*, Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), 1996, pp. 226–231.

[Fis10] H. Fischer, *A history of the central limit theorem: From classical to modern probability theory*, 1st edition. ed., Springer, 2010.

[FWH+96] K.J. Friston, S. Williams, R. Howard, R.S. Frackowiak, and R. Turner, *Movement-related effects in fmri time-series*, Magn Reson Med **35** (1996), no. 3, 346–355.

[GRS98] S. Guha, R. Rastogi, and K. Shim, *Cure: an efficient clustering algorithm for large databases*, Proceedings of the ACM SIGMOD Conference on Management of Data, vol. 27, 1998, pp. 73–84.

[Gut84] A. Guttman, *R-trees: A dynamic index structure for spatial searching*, Proceedings of the ACM SIGMOD Conference on Management of Data, 1984, pp. 47–57.

[Har08] M. Hartmann, *Moderne schnittbildgebung bei gliomen*, Onkologie heute (2008), no. 4, 8–17.

[Haw80] D. Hawkins, *Identification of outliers*, Chapman and Hall, London, 1980.

[Haw85] D Hawkins, *Outliers*, vol. 6, Kotz, S., Johnson, N. L. (Hg.), Encyclopedia of Statistical Sciences, New York, 1985.

[HFH+09] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten, *The weka data mining software: An update*, ACM SIGKDD Explorations Newsletter **11** (2009), no. 1, 10–18.

[HKO01] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*, 1 ed., Wiley-Interscience, 2001.

[HOK+09] J. Huo, K. Okada, H.J. Kim, W.B. Pope, J.G. Goldin, J.R. Alger, and M.S. Brown, *Cadrx for gbm brain tumors: Predicting treatment response from changes in diffusion-weighted mri*, Algorithms **2** (2009), 1350–1367.

[Huf52] D. Huffman, *A method for the construction of minimum-redundancy codes*, Proceedings of the Institute of Radio Engineers **40** (1952), no. 9, 1098–1101.

[Hyv99] A. Hyvärinen, *Fast and robust fixed-point algorithms for independent component analysis*, IEEE Transactions on Neural Networks **10** (1999), no. 3, 626–634.

[JKF01] O. Jesorsky, K.J. Kirchberg, and R. Frischholz, *Robust face detection using the hausdorff distance*, Proceedings of the International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA) (London, UK), Springer-Verlag, 2001, pp. 90–95.

[JKN98] T.J. Johnson, I. Kwok, and R.T. Ng, *Fast computation of 2-dimensional depth contours*, Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD) (New York, NY, USA), 1998, pp. 224–228.

[JOT+05] H.V. Jagadish, B.C. Ooi, K.L. Tan, C. Yu, and R. Zhang, *idistance: An adaptive b^+-tree based indexing method for nearest neighbor search*, ACM Transactions on Database Systems (New York, NY, USA), vol. 30, 2005, pp. 364–397.

[KIN+01] K. Kono, Y. Inoue, K. Nakayama, M. Shakudo, M. Morino, K. Ohata, K. Wakasa, and R. Yamada, *The role of diffusion-weighted imaging in patients with brain tumors*, AJNR Am J Neuroradiol **22** (2001), no. 6, 1081–1088.

[KK06] S. Kim and I.S. Kweon, *Simultaneous classification and visual-word selection using entropy-based minimum description length*, Proceedings of the International Conference on Pattern Recognition (ICPR), vol. 1, 2006, pp. 650–653.

[KKK+10] H.S. Kim, J.H. Kim, S.H. Kim, K.G. Cho, and S.Y. Kim, *Post-treatment high-grade glioma: usefulness of peak height position with semiquantitative mr perfusion histogram analysis in an entire contrast-enhanced lesion for predicting volume fraction of recurrence*, Radiology **256** (2010), no. 3, 906–915.

[KKSZ09] H.P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, *Loop: local outlier probabilities*, Proceedings of the Conference on Information and knowledge management (CIKM) (New York, NY, USA), 2009, pp. 1649–1652.

[KLR04] E. Keogh, S. Lonardi, and C. Ratanamahatana, *Towards parameter-free data mining*, Proceedings of the International

Conference on Knowledge Discovery and Data Mining (KDD), 2004, pp. 206–215.

[KN97] E. Knorr and R.T. Ng, *A unified notion of outliers: Properties and computation*, Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), 1997, pp. 219–222.

[KN98] E. Knorr and R. Ng, *Algorithms for mining distance-based outliers in large datasets*, Proceedings of the International Conference on Very Large Databases (VLDB) (New York, NY, USA), 1998, pp. 392–403.

[KN99] E. Knorr and R.T. Ng, *Finding intensional knowledge of distance-based outliers*, Proceedings of the International Conference on Very Large Databases (VLDB), 1999, pp. 211–222.

[Koh95] R. Kohavi, *A study of cross-validation and bootstrap for accuracy estimation and model selection*, Proceedings of the International Joint Conference on Articial Intelligence, vol. 2, Morgan Kaufmann, 1995, pp. 1137–1143.

[KR90] L. Kaufman and P.J. Rousseeuw, *Finding groups in data: An introduction to cluster analysis*, Wiley-Interscience, 1990.

[KTE+11] M. Kunz, N. Thon, S. Eigenbrod, C. Hartmann, R. Egensperger, J. Herms, J. Geisler, C. la Fougere, J. Lutz, J. Linn, S. Kreth, A. von Deimling, J.C. Tonn, H.A. Kretzschmar, G. Pöpperl, and F.W. Kreth, *Hot spots in dynamic18fet-pet delineate malignant tumor parts within suspected who grade ii gliomas*, Neuro Oncology **13** (2011), no. 3, 307–316.

[LIT92] P. Langley, W. Iba, and K. Thompson, *An analysis of bayesian classifiers*, Proceedings of the National Conference on Artificial Intelligence, 1992, pp. 223–228.

[LKK+01] S.J. Lee, J.H. Kim, Y.M. Kim, G.K. Lee, E.J. Lee, I.S. Park, J.M. Jung, K.H. Kang, and T. Shin, *Perfusion mr imaging in gliomas: comparison with histologic tumor grade*, Korean J Radiol **2** (2001), no. 1, 1–7.

[LPL+99] H.L. Liu, Y. Pu, Y. Liu, L. Nickerson, T. Andrews, P.T. Fox, and J.H. Gao, *Cerebral blood flow measurement by dynamic contrast mri using singular value decomposition with an adaptive threshold*, Magn Reson Med **42** (1999), no. 1, 167–172.

[LTL+03] X. Li, J. Tian, E. Li, X. Wang, J. Dai, and L. Ai, *Adaptive total linear least square method for quantification of mean transit time in brain perfusion mri*, Magn Reson Imaging **21** (2003), no. 5, 503–510.

[LYB+04] M. Law, S. Yang, J.S. Babb, E.A. Knopp, J.G. Golfinos, D. Zagzag, and G. Johnson, *Comparison of cerebral blood volume and vascular permeability from dynamic susceptibility contrast-enhanced perfusion mr imaging with glioma grade*, American Journal of Neuroradiology **25** (2004), no. 5, 746–755.

[LYW+03] M. Law, S. Yang, H. Wang, J.S. Babb, G. Johnson, S. Cha, E.A. Knopp, and D. Zagzag, *Glioma grading: sensitivity, specificity, and predictive values of perfusion mr imaging and proton mr spectroscopic imaging compared with conventional mr imaging*, AJNR Am J Neuroradiol **24** (2003), no. 10, 1989–1998.

[MAA+03] C. Majos, J. Alonso, C. Aguilera, M. Serrallonga, J. Perez-Martin, J. J. Acebes, C. Arus, and J. Gili, *Proton magnetic resonance spectroscopy ((1)h mrs) of human brain tumours:*

assessment of differences between tumour types and its applicability in brain tumour categorization, Eur Radiol **13** (2003), no. 3, 582–591.

[Mah36] P. Mahalanobis, *On the generalised distance in statistics*, Proceedings of the National Institute of Sciences of India **2** (1936), no. 1, 49–55.

[MDH99] V. Megalooikonomou, C. Davatzikos, and E.H. Herskovits, *Mining lesion-deficit associations in a brain image database*, 1999, pp. 347–351.

[MFH+06] K. Mouridsen, K. Friston, N. Hjort, L. Gyldensted, L. Østergaard, and S. Kiebel, *Bayesian estimation of cerebral perfusion using a physiological model of microvasculature*, Neuroimage **33** (2006), no. 2, 570–579.

[MFS+00] V. Megalooikonomou, J. Ford, L. Shen, F. Makedon, and A. Saykin, *Data mining in brain imaging*, Stat Methods Med Res **9** (2000), no. 4, 359–394.

[MJSA+04] C. Majos, M. Julia-Sape, J. Alonso, M. Serrallonga, C. Aguilera, J. J. Acebes, C. Arus, and J. Gili, *Brain tumor classification by proton mr spectroscopy: comparison of diagnostic accuracy at short and long te*, AJNR Am J Neuroradiol **25** (2004), no. 10, 1696–1704.

[MP00] G. McLachlan and D. Peel, *Finite mixture models*, 1 ed., John Wiley & Sons, Inc., Hoboken, NJ, 2000.

[MR05] A. Mineo and M. Ruggieri, *A software tool for the exponential power distribution: The normalp package*, Journal of Statistical Software **12** (2005), no. 4, 1–24.

[MU06] A. Madabhushi and J.K. Udupa, *New methods of mr image intensity standardization via generalized scale*, Med Phys **33** (2006), no. 9, 3426–3434.

[NH94] R.T. Ng and J. Han, *Efficient and effective clustering methods for spatial data mining*, Tech. report, University of British Columbia, Vancouver, BC, Canada, Canada, 1994.

[NU99] L.G. Nyul and J.K. Udupa, *On standardizing the mr image intensity scale*, Magn Reson Med **42** (1999), no. 6, 1072–1081.

[NUZ00] L.G. Nyul, J.K. Udupa, and X. Zhang, *New variants of a method of mri scale standardization*, IEEE Trans Med Imaging **19** (2000), no. 2, 143–150.

[OSK+96] L Ostergaard, AG Sorensen, KK Kwong, RM Weisskoff, C Gyldensted, and BR Rosen, *High resolution measurement of cerebral blood flow using intravascular tracer bolus passages. part ii: Experimental comparison and preliminary results*, Magn Reson Med **36** (1996), no. 5, 726–736.

[Ost05] L. Ostergaard, *Principles of cerebral perfusion imaging by bolus tracking*, Journal of Magnetic Resonance Imaging **22** (2005), no. 6, 710–717.

[OWC+96] L. Ostergaard, R.M. Weisskoff, D.A. Chesler, C. Gyldensted, and B.R. Rosen, *High resolution measurement of cerebral blood flow using intravascular tracer bolus passages. part i: Mathematical approach and statistical analysis*, Magn Reson Med **36** (1996), no. 5, 715–725.

[PFTV92] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical recipes in c: The art of scientific computing*, 2 ed., Cambridge University Press, 1992.

[PKGF03] S. Papadimitriou, H. Kitagawa, P.B. Gibbons, and C. Faloutsos, *Loci: Fast outlier detection using the local correlation integral*, Proceedings of the International Conference on Data Engineering (ICDE), 2003, pp. 315–326.

[PKM+07] G. Pöpperl, F.W. Kreth, J.H. Mehrkens, J. Herms, K. Seelos, W. Koch, F.J. Gildehaus, H.A. Kretzschmar, J.C. Tonn, and K. Tatsch, *Fet pet for the evaluation of untreated gliomas: correlation of fet uptake and uptake kinetics with tumour grading*, Eur J Nucl Med Mol Imaging **34** (2007), no. 12, 1933–1942.

[PM00] D. Pelleg and A. Moore, *X-means: Extending K-means with efficient estimation of the number of clusters*, Proceedings of the International Conference on Machine Learning (ICML), 2000, pp. 727–734.

[PMB06] J.M. Provenzale, S. Mukundan, and D.P. Barboriak, *Diffusion-weighted and perfusion mr imaging for brain tumor characterization and assessment of treatment response*, Radiology **239** (2006), no. 3, 632–649.

[PML+05] D. Pokrajac, V. Megalooikonomou, A. Lazarevic, D. Kontos, and Z. Obradovic, *Applying spatial distribution analysis techniques to classification of 3d medical images*, Artif Intell Med **33** (2005), no. 3, 261–280.

[PP00] J.R. Petrella and J.M. Provenzale, *Mr perfusion imaging of the brain: Techniques and applications*, American Journal of Roentgenology **175** (2000), no. 1, 207–219.

[PP07] A. Patcha and J.M. Park, *An overview of anomaly detection techniques: Existing solutions and latest technological trends*, Computer Networks **51** (2007), no. 12, 3448–3470.

[Rad10] P. Radl, Website, 2010, Available online at `http://wetter61169.de/`; visited on April 1st 2010.

[RHPM04] M. Ruschhaupt, W. Huber, A. Poustka, and U. Mansmann, *A compendium to ensure computational reproducibility in high-dimensional classification tasks*, Stat Appl Genet Mol Biol **3** (2004), Article37.

[Ris78] J. Rissanen, *Modeling by shortest data description*, Automatica **14** (1978), no. 5, 465–471.

[Ris00] J Rissanen, *Mdl denoising*, IEEE Transactions on Information Theory **46** (2000), no. 7, 2537–2543.

[RKB07] C. Rorden, H.O. Karnath, and L. Bonilha, *Improving lesion-symptom mapping*, J Cogn Neurosci **19** (2007), no. 7, 1081–1088.

[RR95] D.A. Reynolds and R.C. Rose, *Robust text-independent speaker identification using gaussian mixture speaker models*, IEEE Transactions on Speech and Audio Processing **3** (1995), no. 1, 72–83.

[RR96] I. Ruts and P.J. Rousseeuw, *Computing depth contours of bivariate point clouds*, Computational Statistics and Data Analysis **23** (1996), no. 1, 153–168.

[Sch78] G. Schwarz, *Estimating the dimension of a model*, The Annals of Statistics **6** (1978), no. 2, 461–464.

[SG02] A. Strehl and J. Ghosh, *Cluster ensembles - a knowledge reuse framework for combining multiple partitions*, Journal of Machine Learning Research **3** (2002), 583–617.

[Sib73] R. Sibson, *Slink: An optimally efficient algorithm for the single-link cluster method*, The Computer Journal **16** (1973), no. 1, 30–34.

[SKK+98] T. Sugahara, Y. Korogi, M. Kochi, I. Ikushima, T. Hirai, T. Okuda, Y. Shigematsu, L.X. Liang, Y.L. Ge, Y. Ushio, and M. Takahashi, *Correlation of mr imaging-determined cerebral blood volume maps with histologic and angiographic determination of vascularity of gliomas*, American Journal of Roentgenology **171** (1998), no. 6, 1479–1486.

[SLK+02] J.H. Shin, H.K. Lee, B.D. Kwun, J.S. Kim, W. Kang, C.G. Choi, and D.C. Suh, *Using relative cerebral blood flow and volume to evaluate the histopathologic grade of cerebral gliomas: preliminary results*, AJR Am J Roentgenol **179** (2002), no. 3, 783–789.

[SLTF04] M.R. Smith, H. Lu, S. Trochet, and R. Frayne, *Removing the effect of svd algorithmic artifacts present in quantitative mr perfusion studies*, Magn Reson Med **51** (2004), no. 3, 631–634.

[TCX+05] Y. Tao, R. Cheng, X. Xiao, W.K. Ngai, B. Kao, and S. Prabhakar, *Indexing multi-dimensional uncertain data with arbitrary probability density functions*, Proceedings of the International Conference on Very Large Databases (VLDB), 2005, pp. 922–933.

[VEB09] Nguyen Vinh, Julien Epps, and James Bailey, *Information theoretic measures for clusterings comparison: is a correction for chance necessary?*, Proceedings of the International Conference on Machine Learning (ICML) (New York, NY, USA), 2009, pp. 1073–1080.

[VEB10] Nguyen V., J. Epps, and J. Bailey, *Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance*, Journal of Machine Learning Research **11** (2010), 2837–2854.

[WJH+98] E.T. Wong, E.F. Jackson, K.R. Hess, D.F. Schomer, J.D. Hazle, A.P. Kyritsis, K.A. Jaeckle, W.K.A. Yung, V.A. Levin, and N.E. Leeds, *Correlation between dynamic mri and outcome in patients with malignant gliomas*, Neurology **50** (1998), no. 3, 777–781.

[WSB98] R. Weber, H.J. Schek, and S. Blott, *A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces*, Proceedings of the International Conference on Very Large Databases (VLDB), 1998, pp. 194–205.

[XZX04] J. Xie, D. Zhang, and W. Xu, *Spatially adaptive wavelet denoising using the minimum description length principle*, IEEE Transactions on Image Processing **13** (2004), no. 2, 179–187.

[YMW02] T. Yoshida, H. Motoda, and T. Washio, *Adaptive ripple down rules method based on minimum description length principle*, Proceedings of the International Conference on Data Mining (ICDM), 2002, pp. 530–537.

[ZWC+09] E.I. Zacharaki, S. Wang, S. Chawla, D. Soo Yoo, R. Wolf, E.R. Melhem, and C. Davatzikos, *Classification of brain tumor type and grade using mri texture and shape in a machine learning scheme*, Magn Reson Med **62** (2009), no. 6, 1609–1618.