Attribute Relationship Analysis in Outlier Mining and Stream Processing

zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Fabian Keller

aus Bühl

Tag der mündlichen Prüfung:	01.07.2015
Erster Gutachter:	Prof. Dr. Klemens Böhm
Zweiter Gutachter:	Prof. Dr. Stefan Wrobel
Dritter Gutachter:	Dr. Emmanuel Müller

DOI 10.5445/IR/1000048790

Contents

Deutsche Zusammenfassung

Gegenstand der Forschung dieser Dissertation ist das Zusammenführen zweier großer Teilbereiche der Datenanalyse. Auf der einen Seite beschäftigt sich das Teilgebiet Outlier Mining mit der Erforschung einer automatisierten Erkennung von Anomalien in Datenbeständen. Auf der anderen Seite gibt es den Teilbereich der Abhängigkeits- oder Korrelationsanalyse, die sich mit der Frage beschäftigt, wie Abhängigkeiten zwischen verschiedenen Datenmerkmalen quantifiziert und erkannt werden können. Beide Forschungsrichtungen wurden bisher isoliert betrachtet. Ein wesentlicher wissenschaftlicher Beitrag meiner Arbeit ist, den Zusammenhang dieser beiden Forschungsrichtungen herzustellen.

Eine große Herausforderung im Bereich Outlier Mining ist es, Anomalien zu detektieren, deren Abweichungen nur bezüglich einer bestimmten Teilmenge der Datenmerkmale (Subspace) sichtbar sind. Diese schwer detektierbaren Anomalien werden daher als Subspace-Outlier bezeichnet. Gegenstand meiner Forschung ist die bisher ungelöste Frage, wie sich algorithmisch Subspaces finden lassen (Subspace-Search), die Subspace-Outlier enthalten. Zur Lösung dieses Problems bedarf es zunächst einer Formalisierung von Subspace-Outliern. Über diese Formalisierung lässt sich schließlich die Verbindung zur Welt der Abhängigkeits- und Korrelationsanalyse herstellen: Die Erkenntnis dabei ist, dass das Vorhandensein von statistischer Abhängigkeit ein notwendiges aber nicht hinreichendes Kriterium für die Existenz von Subspace-Outliern ist. Das primäre Ergebnis dieser Forschung war die Entwicklung des ersten Subspace-Search-Verfahrens, das gezielt für den Kontext Outlier-Erkennung konzipiert wurde. Als Bewertungsfunktion von Subspaces führen wir den sog. Subspace-Kontrast ein, der auf einem Vergleichen von bedingten Wahrscheinlichkeitsdichten mit der zugehörigen Randverteilung basiert. Dadurch ist es möglich die Kontrastberechnung auf traditionelle statistische Tests zum Vergleichen von Wahrscheinlichkeitsdichten zurückzuführen. Das wiederum erlaubt, den Kontrast über das Signifikanzniveau der statistischen Tests zu definieren. Der so definierte Subspace-Kontrast ist daher als Gütemaß sehr robust, anschaulich und ermöglicht aufgrund der impliziten Normalisierung verschiedene Subspaces direkt miteinander vergleichen zu können. Wir haben die Berechnung des Subspace-Kontrasts als Monte-Carlo Algorithmus umgesetzt. Neben dem Vorteil einer sehr effizienten Berechnung lösen wir damit auch die Herausforderung, hoch-dimensionale Subspaces zu bewerten. Die Idee dabei ist, die Größe der Teststatistik über verschiedene Dimensionalitäten konstant zu halten. Dies kann durch eine adaptive Slicing-Technik innerhalb der Rangordnungsstatistiken erreicht werden. Aufbauend auf der Subspace-Kontrast-Funktion

kann nun ein heuristischer Algorithmus implementiert werden, der sich stufenweise von niedrig- zu hochdimensionalen Subspaces vorarbeitet. Abschließend haben wir unseren Subspace-Search-Ansatz in zahlreichen Experimenten empirisch untersucht. Im Vergleich zu existierenden Ansätzen konnte die Erkennungsrate von Outliern deutlich gesteigert werden.

Der nächste Schritt meiner Forschung widmet sich der Fragestellung, wie ein Subspace-Search-Verfahren gezielt für ein vom Anwender gewähltes Outlier-Modell optimiert werden kann. Die Motivation für einen solchen Ansatz ergibt sich aus dem Reichtum und der Variabilität existierender Outlier-Modelle die sich in der wissenschaftlichen Literatur finden lassen. Je nach Modell können Outlier-Definitionen über unterschiedlichste Kriterien erfolgen, beispielsweise basierend auf Distanz, Dichte, Winkelverhältnissen oder dem informationellen Beschreibungsaufwand. Dabei hat in der Praxis jedes Modell gewisse Vor- und Nachteile die anwendungsspezifisch ausgenutzt werden können. Ziel unseres nächsten Subspace-Search-Ansatz ist daher, die Suche der relevanten Subspaces individuell für jedes Objekt in Abhängigkeit eines gegebenen Outlier-Modells durchzuführen. Dies kann über einen stochastischen Ansatz erreicht werden. Im ersten Schritt des Algorithmus wird dazu das Outlier-Modell in zufällig gewählten Subspaces angewandt. Die Dimensionalität dieser Subspaces wird so gewählt, dass sie groß genug ist, um eine große Subspace-Überdeckung zu erreichen, aber gleichzeitig noch nicht zu hoch, um den Effekt des Fluchs der Dimensionalität im Griff zu halten. Im zweiten Schritt erfolgt die Auswertung der so gewonnen Informationen. Dabei wird die Tatsache ausgenutzt, dass ein Subspace-Outlier immer dann einen etwas ausgeprägteren Anomaliegrad zeigen wird, wenn man ihn in einem Subspace betrachtet, der eine Obermenge des relevanten Subspaces ist. Im Umkehrschluss bedeutet das, dass sich der relevante Subspace durch eine kombinatorische Analyse identifizieren lässt. Ergebnis dieses Teilalgorithmus ist eine verfeinerte Menge an Datenmerkmalen, in denen ein jeweiliges Objekt für das gegebene Outlier-Modell anomal erscheint. Im dritten Schritt kann das Outlier-Modell nochmals auf die so erhaltenen verfeinerten Subspaces angewendet werden. Experimentell lässt sich eindeutig feststellen, dass die resultierende Subspace-Suche tatsächlich die jeweiligen Eigenschaften des zugrunde liegenden Outlier-Modells berücksichtigt und damit nur Subspaces ausgibt, die tatsächlich von Relevanz sind.

Abschließend habe ich mich der Frage gewidmet, wie sich eine Subspace-Suche umsetzen lässt in dem Falle, dass Daten nicht statisch vorliegen, sondern dynamisch in Form eines Datenstroms eintreffen. Damit ergibt sich als große Herausforderung, dass sich nun alle Subspace- und Variablenabhängigkeiten selbst dynamisch über die Zeit verändern können. Aufgrund der höheren Komplexität dieses Problem liegt der Fokus dabei auf der einfachsten Ausprägung einer Subspacestruktur, also dem zweidimensionalen Fall, der einer direkten Abhängigkeitsanalyse zweier Datenmerkmale entspricht. Als konkretes Beispiel eines Bewertungsmaß der Abhängigkeit betrachten wir die etablierte Mutual Information. Um Mutual Information in beliebigen Zeitfenstern berechnen zu können wäre es notwendig, den Datenstrom vollständig abzuspeichern, was den Anforderungen einer Online-Technik nicht gerecht wird. Gleichzeitig wäre damit bei jeder

Mutual-Information-Anfrage an das System eine aufwändige Neuberechnung notwendig, selbst dann, wenn sich die Zeitfenster mehrerer Anfragen deutlich überlappen. Daher war das Ziel einen neuartigen Ansatz zu entwickeln, der diese Probleme lösen kann. Die wesentliche Komponente dabei war die Entwicklung einer speziellen Datenstruktur, Query-Anchor genannt, die Zwischenergebnisse der Mutual-Information-Schätzung effizient vorberechnen und zwischenspeichern kann. Dies erlaubt, Mutual-Information-Anfragen auf Basis der Query-Anchor zu beantworten. Für die Verteilung von Query-Anchors über die Zeit wurde eine spezielle Sampling-Technik entwickelt, die gewährleistet, Anfragen mit gleichbleibender Genauigkeit über verschiedene Zeitskalen zu beantworten. In zahlreichen Experimenten konnte gezeigt werden, dass diese Umsetzung eines Online-Mutual-Information-Schätzers beachtliche Verbesserungen bei der Verarbeitungsgeschwindigkeit erzielt. Die Technik stellt damit den ersten Grundstein für eine Subspace-Suche auf Datenströmen dar.

Acknowledgments

This thesis is more than just the work of one person, and I would like to express my gratitude to everyone who has helped me along the way.

First of all, I would like to thank my supervisor, Prof. Klemens Böhm, for his advice and support. This thesis would not have been possible without him. His confidence in me has allowed me to switch from physics to computer science in the first place. Furthermore, I'm grateful for all the freedom and encouragement he has given me over the last four years.

Likewise, I would like to thank Prof. Emmanuel Müller, who has contributed just as much to the supervision of my thesis. He has allowed me to initiate this research direction and has been a great source of inspiration over all the years. In particular I would like to thank you for the immense effort you have put into my supervision and the strong moral support.

I am very grateful to Prof. Stefan Wrobel, for agreeing to act as an external supervisor for this thesis. Thank you very much for your interest in my research and for the very pleasant visit on the occasion of my PhD defense.

In addition to the direct support from my supervisors, I have also received a lot of support from my colleagues at our chair. I've had a great time working with you, and I'd like to thank you all for your big help. In particular I would like to thank Stephan Kessler and Fabian Laforet for the tremendous helpfulness and many fruitful discussions. Similarly, I want to thank Pavel Efros who came to the rescue for the presentation of my last publication, which conflicted with my PhD defense. Last but not least, I would like to give special thanks to Patricia Iglesias – not only for working with me until 5 a.m. if need be, but simply for being the most helpful, altruistic, and entertaining officemate I could imagine.

I also would like to thank my family and all my friends for being supportive as well as diverting. First and foremost, I want to thank my parents for the strong support over many years, and also for sparking my interest in science in the first place. And last but not least, I would like to thank you, Anne, for all your love and support.

Abstract

The main theme of this thesis is to unite two important fields of data analysis. On the on hand, there is the area of outlier mining, which considers the problem of detecting unusual patterns in data. On the other hand, research topics like correlation analysis and subspace search evaluate the relationships of data attributes. Up to now, research on these topics has been conducted independently. The major contribution of this work is to analyze and establish the connection between these fields.

In this thesis we develop several techniques which follow this idea of combining outlier detection with attribute relationship analysis. The main difference of the techniques is how these two aspects are combined. For instance in our first approach, we develop an algorithm which performs attribute relationship analysis as a preprocessing step to outlier mining. Compared to existing techniques, this is the first approach that is optimized to specifically detect attribute relationships that are relevant for outlier mining. In another algorithm, we incorporate the outlier detection process directly into the attribute relationship mining. This allows to quantify attribute relationships individually for each object in dependence of different outlier models.

Apart from our algorithmic contributions, the thesis includes an extensive experimental analysis. We analyze all our algorithms by several different evaluation schemes on a broad range of data sets, including both a large number of real-world and synthetic data sets. Overall, our findings show the synergies of combining the two different worlds outlier mining and attribute relationship analysis: (1) The quality of outlier mining can be increased significantly by exploiting attribute relationships. (2) Outlier detection provides a novel kind of information regarding attribute relationships.

While in the first part of this thesis we focus on traditional databases as data source, we extend the scope in the second part towards data streams. The general goal of this second part is to adapt our approaches to this modified problem. In general, solving data mining problems on data streams is one of the major open challenges in big data applications. In contrast to traditional databases, the nature of the stream requires techniques to operate dynamically – not only with respect to how data is processed, but also regarding the mining results, which become time dependent as well. In this thesis we propose a first technique which allows to perform an attribute relationship analysis that is tailored to operate on data streams. In a broad empirical analysis we can show that this approach has significant advantages in an online stream processing.

Part I.

Introduction

1. Thesis Overview

1.1. Data Mining – from Data to Knowledge

Today's ubiquity of data is probably the most apparent result of the digital revolution. Data is created and collected everywhere – ranging from data recorded by the tiniest sensor devices right up to data of large, complex, and highly coherent systems like the world climate or global economy. This abundance of data raises the key question: How can we extract knowledge from data? The common goal of data mining is to provide scientific solutions to this very question. Hence, it is not a surprise that today data mining is on everyone's mind.

The overall process of extracting knowledge from data has been formalized in different ways in the literature [FPSS96, Sheoo, Azeo8]. Figure 1.1 shows an abstraction of the most commonly used approaches. While there are slight differences in the individual steps of the processes, the structural resemblance is high. Technically data mining appears as an individual step in the processes. However, it is a common understanding to associate data mining with the knowledge extraction process as a whole. Since these methodologies provide generic templates for data mining, they do not explicitly define the notion of data or knowledge. In this dissertation we will focus on the following specific aspects in terms of data and knowledge.

1.1.1. Specification of Data

As indicated in Figure 1.1, "data" can originate from different sources. The distinction of the data source will be reflected in the general structure of this thesis. We will differentiate between *databases* and *data streams* as possible data sources.

The first possible source are traditional databases. The essential property of this kind of data is that the dataset itself can be considered static and finite. The data is stored in its entirety either on disc or in memory. This allows to process the data as a whole, which offers perfect conditions for scientific investigation. Temporal aspects do not play a role in this case. As a consequence, research on static data is often conducted as foundation when first addressing a novel problem statement.

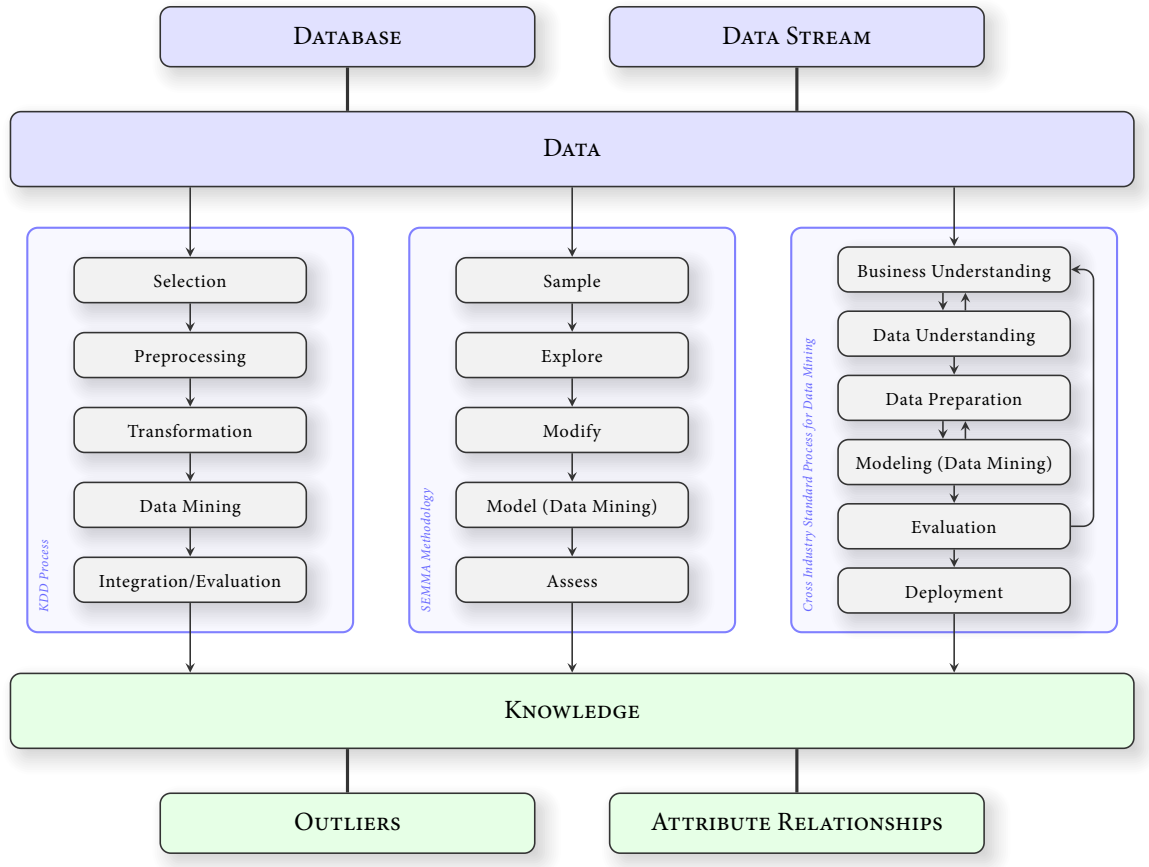


Figure 1.1.: From data to knowledge – abstraction of common approaches to data mining. On the left: The Knowledge Discovery from Databases (KDD) Process [FPSS96]. In the middle: The SEMMA methodology [Sheoo]. On the right: The Cross Industry Standard Process for Data Mining (Crisp-DM) [Azeo8].

The second kind of data sources are data streams. The essential property of data streams is that the notion of time plays an explicit role. This temporal nature of data streams has several implications for a given data mining problem:

- The most obvious property of data derived from a stream is that time inherently appears as a given dimension of the data. This is in contrast to static datasets, where the set of dimensions does not contain a single dimension component with a fixed semantic.

- Time as a dimension is potentially infinite. Furthermore a data stream may be sampled at an almost infinite time resolution – theoretically bounded only by the Plank time 5.4×10^{-44} s, the time it takes light to travel one Planck length. Since infinity potentially appears for both length and resolution, processing data streams poses a huge technical challenge regarding both processing speed and memory complexity.

- The inherent presence of time also applies for "knowledge" in the data mining process. This means that the data mining result, i.e., the extracted knowledge itself, is not static but may change over time. As a result, it is necessary to technically account for potential dynamics in the data mining target.

In this dissertation, the idea is to discuss both sources of data, static and dynamic. The motivation for this is to first utilize the clearness of the static scenario to establish a foundation for further research. As next step we turn to the question of how to transfer our techniques to the world of data streams.

To clarify the following discussions, we briefly introduce our data related terminology and notation used in this thesis: We will refer to the objects of a database or data stream by varying terms like elements, patterns, instances, samples, or simply objects (with standard notion: o). This diversity is only due to linguistic reasons, and there is no difference in the notion of these terms. The set of all objects is denoted as DB. Similarly, we vary the terms used to refer to the features of a data object. The most frequent terms are attributes, dimensions, measured values, or features. We use the notation $\mathcal{A}$ to refer to the set of all attributes.

1.1.2. Specification of Knowledge

The specification of "knowledge" in the data mining process determines the problem statement actually addressed. As with the type of data sources, the scope of this dissertation is two-fold by covering two important information aspects: The knowledge revealed by *outlier mining* on the one hand and *attribute relationship analysis* on the other. Outlier mining, also referred to as anomaly detection, provides information on rare or suspicious elements of a datasets. The type of information provided by attribute relationship analysis addresses the question how data attributes relate to each other, which reveals general structures of the data. In the literature extracting this kind of knowledge is commonly referred to as dependence or correlation analysis. However, we deliberately chose the term "attribute relationship analysis", since it allows for a broader definition of this field. This broader definition will play a key role in this work, which we discuss in Section 1.3.

As a key contribution of this dissertation, we will analyze the connection of the putatively unrelated domains outlier mining and attribute relationship analysis. Before we go into the details of how these topics are related, we will introduce each one individually in the following.

1.2. Introduction to Outlier Mining

Outlier mining is one of the most traditional paradigms in data mining. As a result of its long history and broad application scope, it has been referred to by many synonyms

like anomaly detection, rare-class-, or one-class-classification. In colloquial language the notion of an outlier has no clear definition. In this thesis, we define an outlier according to the following formal definition, which was first introduced by Hawkins in 1980 [Haw80]:

> **Quote:** "An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism."

Outlier mining has a very broad field of possible applications. In general, one has to differentiate between the technical and the conceptual aim of performing outlier detection. When comparing different application domains, the technical aim is the same: Detecting objects which are anomalous or rare. On the other hand, the conceptual motivation behind this can vary largely between different applications. We will take a look at different conceptual motivations in the following.

One of the most traditional motivations of detecting outliers is simply to remove them from a dataset. Such a motivation often arises in cases where outliers represent some sort of noise and do not contain any valuable information at all. This might be the case for instance when data is obtained from a faulty measuring device. In this case, outlier detection is not applied as an end in itself. Often the actual goal might be to apply a different data analysis technique. However, for many data analysis tasks the existence of incorrect data samples is a significant issue. Depending on the technique the issues can range from a severely degraded quality up to completely useless analysis results. Therefore, a common approach is to perform outlier detection as a preprocessing step, and to remove outliers for further processing.

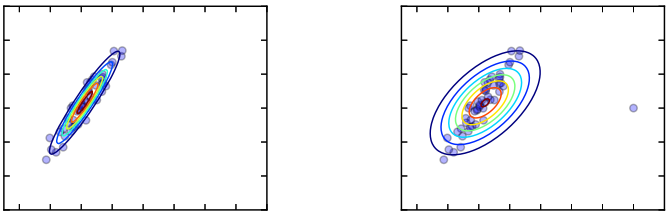


Figure 1.2.: Examples showing the effect of fitting a bivariate Gaussian distribution without (left) and with (right) an outlying object.

One of the most basic examples arises for the traditional statistical problem of fitting a distribution to a data sample. In this case a statistician would chose an appropriate distribution function based on prior knowledge on the data. Given a distribution and a sample, the goal is to obtain the best estimation of the distribution parametrization. Figure 1.2 shows an example of fitting a bivariate Gaussian distribution. The dataset consists of 50 samples drawn from a Gaussian distribution. The profile lines show the resulting parametrization of the Gaussian distribution obtained via estimation. In the left figure, we can see that the estimation indeed captures both the mean and the covariance

matrix of the generating Gaussian very well. On the right, we show the effect of adding a single outlier to the sample. We can see that the estimation of the covariance matrix is immediately completely wrong. Therefore, outlier detection plays an important role for traditional statistics.

Even without such a specific data-analytical goal, outlier mining plays an important role as preprocessing step. As a result of today's ubiquity of complex data sources, storing data has become a challenge in itself. Apart from gathering and integrating data, data cleansing plays a key role in the data storing process. In this case outlier mining can be applied to ensure the validity of data before it is persisted.

Although traditionally outlier mining has played a role as supplementary technique, it has long since become an autonomous problem. The reason for this is that a data anomaly rarely carries no information at all, which would motivate its removal. In may cases it is quite the opposite: A deviating data sample may rather provide *more* information compared to data samples which follow regular patterns. Intuitively this is connected to one of the fundamental observations of information theory: It is more costly to encode a rare pattern compared to frequent patterns. In general, the exceptional behavior and the way how a sample deviates from other objects provides novel knowledge and allows a user to gain insights on the data set at hand. Therefore, outlier mining is often applied as a means to improve the understanding of data, allowing users to search for novel phenomena.

Outlier mining as a means of novel phenomena discovery has applications in a broad range of domains. A typical example is the analysis of health surveillance data. In a medical analysis, an ideal case would be to have large amounts of data of both healthy and diseased patients, allowing to extract a good model of a certain disease. However in practice, such labeled data often is simply not available. In this case, outlier mining allows a physician to analyze the unlabeled data for anomalies. Based on the observed deviation from regular medical conditions, a patient can be examined more specifically resulting in an individually adjusted treatment. Furthermore, the analysis of a whole group of medical anomalies can provide insight of the clinical picture of a disease. This example illustrates that discovering anomalies often provides significant knowledge. Since the challenge of discovering novel phenomena arises in almost all scientific fields, there are many more possible applications. The most notable examples include applications in earth science, economy, electrical and mechanical engineering, fraud detection, surveillance security system, intrusion detection, and law enforcement [Agg13a].

1.3. Introduction to Attribute Relationship Analysis

In contrast to outlier mining, the term attribute relationship analysis has no precise definition in the literature. We deliberately introduce this term to subsume and unify the

concepts of two important paradigms in data mining: Correlation analysis and subspace search. The analysis of the relationship between these two paradigms is one of the key contributions of this dissertations. We will give a brief overview in the following.

Correlation analysis commonly refers to techniques that quantify the dependence of one attribute to another. Note that in this thesis, we use the term correlation in its common, broader sense of "deviation from statistical independence", i.e., correlation refers to any kind of dependence. The key property of a correlation analysis technique is the definition of the correlation measure. A large variety of correlation measures have been studied in traditional statistics and information theory. The spectrum ranges from well-established measures like the Pearson's correlation coefficient, Spearman's rank correlation, or mutual information to more recent measures like the maximal information coefficient [RRF+11]. Each correlation measures has specific properties for instance regarding the sensitivity to certain types of relationships (linear, monotonic, distribution-based) or the interpretation and numeric range of the correlation value. Most commonly, correlation measures focus on quantifying the dependence between exactly two attributes. The knowledge obtained from traditional correlation analysis provides an intuitive meaning and is often highly instructive. As a result, correlation analysis has played a highly influential role in data mining regarding both theoretical developments as well as real-world applications. It has been applied extensively as a means to analyze possible causation – even to the point that the phrase "correlation does not imply causation" has become a ubiquitous reminder of a careful interpretation of correlation analysis results.

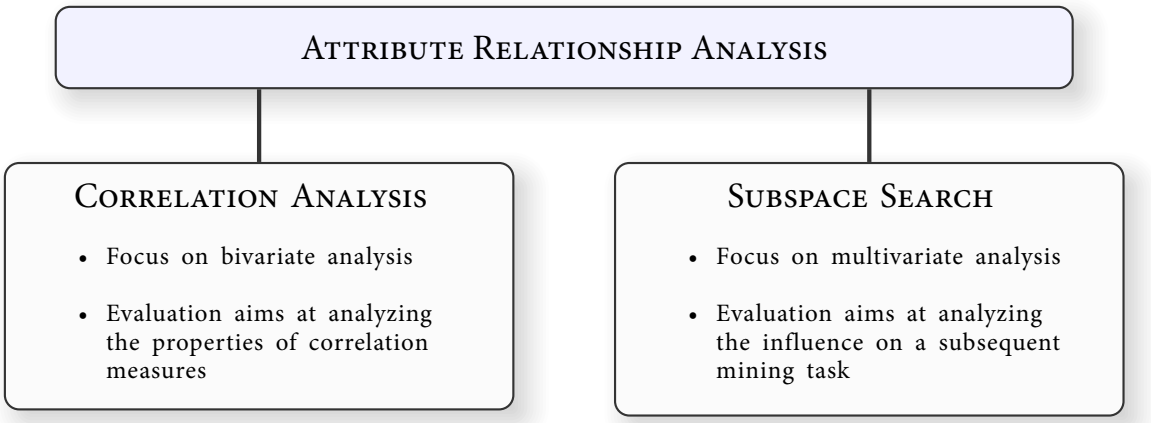


Figure 1.3.: Attribute Relationship Analysis

In contrast to this, subspace search is a rather novel paradigm in data mining. While correlation analysis commonly focuses on the analysis of attribute pairs, subspace search in general performs an analysis of attribute sets of arbitrary cardinality. Subspace search has been introduced in the context of subspace clustering [AGGR98, CFZ99]. The motivation stems from the effect of the infamous "curse of dimensionality" [BGRS99] on clustering: With increasing dimensionality of a data set, all objects of the data set become more and more alike, resulting in meaningless clustering results. Furthermore, clusters

may only manifest themselves in a certain subset of attributes. The idea of subspace search is to introduce a quality function on attribute subsets (subspaces). This quality function evaluates whether a certain subspace may contain meaningful clusters. The overall clustering quality can be significantly improved by applying traditional clustering techniques on the set of the most promising subspaces returned by a subspace search. While originally subspace search has been introduced merely as a preprocessing technique for clustering, it moreover provides knowledge on its own: Similar to correlation analysis, subspace search reveals a relationship between certain attributes. In correlation analysis, the type of relation is defined by the correlation measure. In subspace search, it is the subspace quality function which defines the relationship. In order to emphasize this similarity of these seemingly different topics, we introduce the term attribute relationship analysis to refer to the union of correlation analysis and subspace search. The relation between the terms is summarized in Figure 1.3. The technical connection between these two paradigms will be covered in detail in Chapter 5 and 6.

1.4. Open Challenges

As mentioned in Section 1.1, we will differentiate between the case of static data (databases) and dynamic data (data streams) throughout this thesis. Regarding open challenges, we will see that this division applies as well. Without the notion of time, the challenges of static data lie in the area of discovering the synergies of outlier mining and attribute relationship analysis. In the dynamic case, the challenges shift towards more fundamental problems, due to the increased complexity of the problem itself.

1.4.1. Static Data – Outliers in Subspaces

Recent research on subspaces analysis has been almost exclusively focused on the clustering task. However, a similar issue exists for outlier mining as well. In the literature, this challenge of outlier mining has only been covered superficially and thus is a largely open research topic.

Figure 1.4 illustrates several aspects of this challenge. The data set shows an example of data that is typically gathered by sensor measurements in environmental surveillance. Such data may contain a large number of sensor dimensions including quantities like *temperature, humidity, noise level, air pollution index* etc. The individual figures show examples of two-dimensional subspace projections.

Observation on object level. The first key observation results from tracking a specific object in different subspace views. For instance, we can see that the object highlighted in red ($outlier_1$) only deviates w.r.t. the first subspace $\{$*noise level, air pollution index*$\}$. In the context of all other dimensions, $outlier_1$ shows a fully regular behavior, i.e., it is

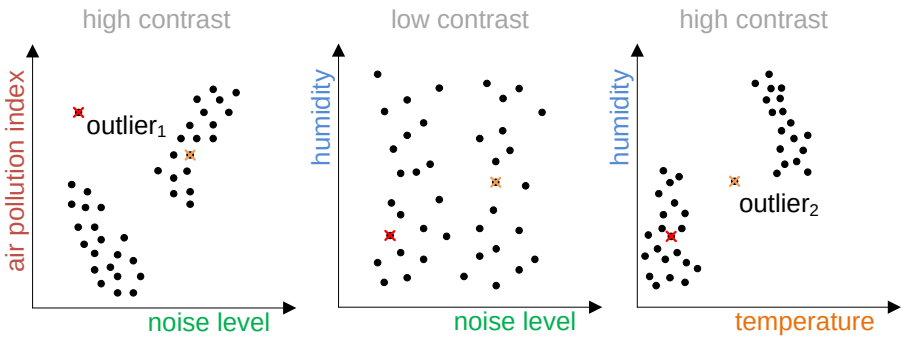


Figure 1.4.: Example of outliers hidden in subspaces

indistinguishable from regular patterns. When it comes to the question whether $outlier_1$ can be detected as outlier, an essential parameter is the dimensionality of the database. As a result of modern data storing facilities, it has become accepted custom to store as much information on a single entity as possible. This means that in many applications one is faced with very high dimensional databases. In the context of the example this implies that there may be a very large number of attributes in which $outlier_1$ is fully regular. Compared to the clear deviation of $outlier_1$ in the given subspace, every attribute that shows regular behavior can be considered irrelevant for the detection of this outlier. In terms of the deviation of $outlier_1$ each of these irrelevant attributes simply adds deviation noise. Therefore, applying an outlier mining approach that operates on the full high-dimensional space has a severe issue: The deviation noise hampers a precise detection of the outlier, potentially up to the point that the anomaly will be missed completely. This makes it obvious that the much-cited curse of dimensionality affects outlier mining as well. Overall, $outlier_1$ is a prime example of one of the major challenges in outlier mining. Throughout this thesis, we will use the term ***subspace outlier*** to refer to such hard-to-detect objects.

Figure 1.4 further illustrates another challenge of subspace outliers. Comparing $outlier_1$ to another anomalous object $outlier_2$ shows that exceptional behavior can be highly individual. In general each outlier may deviate w.r.t. its own set of relevant attributes, e.g., $\{temperature, humidity\}$ in the instance of $outlier_2$. This individuality of the deviating contexts leads to an important challenge: Restricting the analysis to a single, global view on the data does not allow to detect the individual deviations of subspace outliers. Instead, detecting subspace outliers always requires to analyze data from a multi-view perspective. Thus, the detection of subspace outliers poses a computational challenge as a result of the multi-view nature of the problem.

Observation on subspace level. The second observation results from examining the subspaces themselves. We can see that subspaces in general will show a varying degree of structuredness. In the left and the right subspace, the data distributions reveals a distinct pattern. The structures even comprise different clusters of the objects. In terms

of outlier mining, these subspaces show a high contrast, allowing to differentiate clearly between regular and anomalous objects within the subspace. On the other hand, the middle subspace shows a scattered characteristic. In this case it is neither possible to distinguish any outliers or cluster structures. In fact, all objects tend to be similar in terms of their distances to neighboring objects. Regarding outlier mining, this leads to a low contrast for the detection. In the context of the example scenario, this lack of structure is not a surprise: Intuitively, there is no reason to expect any kind of dependence between the local *noise level* and *air humidity*. Thus, assuming the two dimensions are fully independent, a joint examination is of little value, since it is equivalent to combining individual assessments. Overall, we can conclude that on the subspace level there is a general difference in the suitability of a subspace for revealing an outlier. Ideally, when examining outliers in subspaces, one would benefit from knowing which subspaces do (or may) contain potential outliers.

Tackling all these challenges of subspace outliers raises a question that is still an open issue in research: How to perform a subspace search for outlier mining? Clearly, knowing all high contrast subspaces that contain subspace outliers would allow to easily detect them by any traditional outlier detection method constrained to the relevant subspaces. This problem requires to develop novel subspace quality measures specifically designed for outlier detection. The goal for designing such a subspace measure is to make it sensitive to subspaces which potentially contain subspace outliers. This means that the measure must capture the "contrast" of a subspace in the sense whether or not a subspace provides a clear view of the deviation of outliers. This challenge will be covered in Chapter 5 of this dissertation.

A subsequent challenge is to perform a subspace search that adapts automatically to a given outlier definition. In general, there is no universal definition of outliers. In practice each application may require a specific outlier model. As a result of this, the literature provides a large variety of different outlier definitions. It is common practice to choose the outlier definition depending on which model is best suited for a certain data mining problem. This raises the question how to make subspace search adaptive to the model chosen. Given a specific outlier model, the goal for subspace search is to only search for subspaces that are relevant in this case. This requires to adapt the search and the subspace quality measure to the outlier model. The conception of such an adaptive subspace extends the idea of a general-purpose subspace search and will be covered in Chapter 8.

1.4.2. Dynamic Data – Attribute Relationship Analysis

Clearly, the aforementioned challenge of outliers hidden in subspaces is an open issue on data streams as well. Thus, the long term goal is to take the solutions we propose for static data, and make them applicable to data streams as well. However, as a result of the temporal effects and the stream's infinite nature, the problem of detecting subspace outliers in data streams is full of challenges. We will see that in contrast to static data,

we are not only faced with the problem that outlier sensitive subspace search is lacking. On data streams, the challenge is more fundamental: Attribute relationship analysis in general is a non-trivial problem due to the dynamics in the relationships: The dependence between attributes can change over time, and therefore any dependence is tied to a temporal context. Even for traditional pairwise correlation analysis it is an open research question how to deal with these dynamics in general. Therefore, regarding the transition from static to dynamic data, the primary challenge is to develop an attribute relationship analysis technique which can handle the temporal context. Any solution to this challenge paves the way for more advanced subspace search approaches, eventually allowing to solve the challenge of subspace outliers on data streams as well. We address the challenges posed by data streams in the second part of this dissertation.

1.5. Overview of Contributions

By tackling the challenges named above, this dissertations provides the following specific contributions:

Connection of Outlier Mining and Attribute Relationship Analysis. The thesis establishes and analyzes the relation between outlier mining and the analysis of attribute relationships. Section 1.3 already gave a short account on the similarities between correlation analysis and subspace search in the context of outlier mining. To emphasize this relationship we have introduced the term attribute relationship analysis as a subsumption of correlation analysis and subspace search. The technique presented in Chapter 5 is the first approach which exploits this connection. Conceptually, this technique is therefore located right in between the two data mining paradigms outlier mining and attribute relationship analysis.

Proposal of Subspace Contrast. The thesis proposes the first subspace quality measure tailored for outlier mining. The motivation for designing a subspace measure is to enhance the deviation or visibility of outliers which are hidden in subspaces. According to this visibility analogy, we refer to our novel subspace quality measure as "contrast function". The proposed approach focuses on a very efficient computation of the subspace contrast. Furthermore, a key feature is that the contrast measure is designed in a way which allows to easily compare subspaces of different dimensionalities.

Evaluation of Subspace Contrast Regarding Outlier Mining. Our first evaluation of our subspace contrast measure addresses the question how it affects the detection of outliers. We perform a broad range of experiments, showing that subspace contrast leads to significant improvements in terms of outlier detection. Furthermore, we demonstrate how to use subspace contrast as a means to extract outlier descriptions as an additional benefit for manual outlier assessment.

Evaluation of Subspace Contrast Regarding Attribute Relationship Analysis. In a second evaluation of the subspace contrast measure, we analyze its properties from the alternative perspective, i.e., from the perspective of attribute relationship analysis. To this end, we compare our subspace contrast notion against traditional correlation analysis approaches. In a thorough evaluation we can show that our subspace contrast is a highly efficient technique for analyzing attribute relationships, offering several properties which are not provided by existing methods.

Model-Adaptive Subspace Search. The next contribution is a subspace search technique that is capable of adapting the search process to specific outlier models. As a result of the modified search strategy, the subspace contrast is no longer defined universally. Instead the contrast measure is adaptively defined based on the outlier model. This allows to perform subspace search specifically for a broad range of different outlier definitions. Furthermore, it allows us to analyze how different outlier models affect the result of a subspace search.

Model-Specific Subspace Outlier Evaluation. Evaluation of outlier mining results is a challenge in itself. A further key contribution of this thesis is a novel approach to examine the results of outlier mining on real-world data. The idea is based on the observation that the ground truth of a subspace outlier depends on the outlier model being used. In addition to conventional evaluation strategies, we propose to evaluate outlier mining results against a ground truth, which is obtained by a brute force application of each outlier model. This ensures the most meaningful comparison and allows a precise evaluation of the results.

Attribute Relationship Analysis on Data Streams. In the field of data streams this thesis provides a first step towards dynamic subspace search. In order to establish a basis for subspace search, we first transfer the problem of traditional correlation analysis to data streams. Specifically we propose a novel approach for an online estimation of mutual information. Our focus on mutual information is motivated by its favorable properties regarding outlier mining observed in the evaluation on static data. The proposed technique deals with both the dynamics in the dependence itself as well as technical challenges posed by the nature of data streams. We evaluate our technique in a broad range of experiments showing the advantage of an online algorithm in comparison to traditional static computations.

Multiscale Sampling on Data Streams. An important property of an online algorithm is the question of how to store a summary of a data stream over time. The technical challenge is to create a summarization that captures information on various time horizons, for instance ranging from milliseconds to years. As a key component of our solution to the online correlation analysis problem, we propose a novel sampling technique called multiscale sampling. The unique property of multiscale sampling is that it allows to summarize a stream over multiple time scales with an equal summarization quality. This is an important feature for online algorithms, since it allows to operate equally well on

different time horizons. Therefore multiscale sampling could pave the way for other online algorithms that require this property.

2. Traditional Outlier Models

Traditional outlier models will play a key role in the course of this thesis. Therefore, the following chapter will give a brief summary of existing traditional outlier mining techniques. Traditional outlier models are characterized by the fact that they operate on the full-dimensional space. This means that they do not raise the question about relevant or irrelevant attributes at all – they simply use all available attributes of the full space. Due to this characteristic, they are also referred to as fixed-space models.

What differentiates one traditional model from another is the question how regular and irregular objects are defined. At a first glance, these differences appear to be mainly technicalities, and in many cases different outlier models do in fact agree on the question whether a particular object is irregular or not. However, such a consensus is mainly the case when an object is for instance a very apparent anomaly. The particular modeling of outliers becomes much more important when analyzing more interesting cases, i.e, the gray area in between the two opposites. Therefore, we investigate essential differences between outlier models in the following.

2.1. Categorization

Outlier mining techniques can be categorized according to their return types. Some techniques only return a binary information on whether an object is an outlier or not. More advanced approaches instead return a continuous value, which is a measure for the degree of deviation. This degree of deviation is often referred to as *outlier score*, *outlyingness* or *outlierness*. Compared to a binary output, a technique that provides an outlier score for each object has significant advantages: For instance, it allows a data analyst to sort all objects according to their outlier score, and the additional information can be used in subsequent data analyses. Techniques of this kind are therefore often referred to as *outlier ranking* methods. In the course of this thesis, the information content of an outlier detector will play an important role. Therefore, our general focus is on outlier ranking techniques.

2.2. Selected Outlier Models

We will review the main directions of outlier models in the following. We focus on a selection of techniques which are of importance in the scope of this thesis. For a more detailed presentation of traditional methods we refer the reader to surveys like [Agg13a, HA04, CBK07, CBK09].

2.2.1. Distance-Based Paradigm

A large number of techniques models outliers based on the distances to nearest neighbors [KN98, BS03, GPO08, WPT11]. The basic underlying algorithm can be summarized by the following steps:

- For each object, compute the distance to all other objects in the database. Commonly, the distance can be computed by any function satisfying the metric conditions.
- Rank the objects according to their distance and determine the set of k nearest neighbors for each object.
- Define the degree of deviations based on the k nearest neighbor set. The most popular approaches take either the average distance to the k nearest neighbors or the distance to the k-th nearest neighbor itself.

For a distance based approach, the outlier definition has a strong dependence on the parameter k. A possible phenomenon on real-world data is that an outlier is again surrounded by a small number of other outliers. This case is sometimes called a micro cluster or outlier cluster. For a distance based model, the choice of the parameter k has significant influence on whether such outliers are detected or not. In case of a very low k value, the outliers may have a low k nearest neighbor distance to other outliers, and thus will be classified as fully regular objects. Very large k values on the other hand tend to classify whole clusters of moderate size as outliers, since the cluster may have a large distance to other clusters. Therefore, application on real-world data often requires to analyze a broad range of k values to determine the effects of micro clusters.

The complexity of the base algorithm is $O(N^2)$, where N is the number of samples. Techniques like [BS03, GPO08, WPT11] provide runtime improvements, resulting in a sub-quadratic overall complexity. However, the drawback of these techniques is that they do no longer provide a full outlier ranking. The complexity improvements are based on a short-circuit evaluation of the distance. Therefore, many distance-based techniques only provide binary outlier information. Furthermore they often require the user to specify the number of outliers to be detected. Since the number of anomalies is commonly unknown, this significantly reduces applicability. In the context of this thesis a properly

defined outlier score is essential. Therefore, we mainly focus on the base algorithms in our evaluations.

2.2.2. Local Density Paradigm

The local density paradigm was established by Breunig et al. in [BKNSoo] with a technique called *Local Outlier Factor (LOF)*. The key idea is to model outliers according to the deviation from the local object density. In the research community, this model has become one of the most popular reference techniques used in comparative studies. In this thesis it will also play an important role in many experiments. Therefore, we will discuss the model in a bit more detail. In the following we adopt the definitions from [BKNSoo], with only minor modifications to the notation.

Definition 2.1

k-distance of an object p: For any positive integer k, the k-distance of object p, denoted as k-distance(p), is defined as the distance $d(p,o)$ between p and an object $o \in DB$ such that:

(i) for at least k objects $o' \in DB\backslash\{p\}$ it holds that $d(p,o') \leq d(p,o)$, and

(ii) for at most $k-1$ objects $o' \in DB\backslash\{p\}$ it holds that $d(p,o') < d(p,o)$.

In the case that the distances from object p to all other objects $o' \in DB$ are unique, we can simplify this definition: The k-distance is the distance to the k nearest neighbor. Similarly Breunig et al. define the k-neighborhood.

Definition 2.2

k-neighborhood of an object p: Given the k-distance of p, the k-neighborhood of p contains every object whose distance from p is not greater than the k-distance, i.e.,

$$N_k(p) = \{q \in DB\backslash\{p\} \mid d(p,q) \leq k\text{-distance}(p)\}$$

For the case of unique distances, the k-neighborhood is simply the set of the k nearest neighbors, and thus, typically we have $|N_k(p)| = k$. The only other possibility is to have $|N_k(p)| > k$ if there are multiple objects with the same k-distance.

Definition 2.3

Reachability distance of an object p w.r.t. object o: Let k be a natural number. The reachability distance of object p with respect to object o is defined as:

$$\text{reach-dist}_k(p,o) = \max\left\{k\text{-distance}(o), d(p,o)\right\}$$

In comparison to the plain metric $d(p,o)$ the reachability distance introduces a regularization: For very low distances $d(p,o)$, the reachability distance instead uses the k-distance of the destination point as lower bound for the distance. Note that the resulting distance function is no longer symmetric. For large distances, the reachability distance is equal to the distance $d(p,o)$.

The following definition performs the transition from distances to densities and introduces $MinPts$, the main parameter of the LOF model. This parameter controls the "locality" of the LOF outlier model. Valid $MinPts$ values are positive integers.

Definition 2.4

The local reachability density of p is defined as:

$$lrd_{MinPts}(p) = 1 \Big/ \frac{\sum_{o \in N_{MinPts}(p)} \text{reach-dist}_{MinPts}(p,o)}{|N_{MinPts}(p)|}$$

In simple terms (again assuming unique distances), the local reachability density of p is the reciprocal of the average reachability distance to the $MinPts$ nearest neighbors of p. This means for instance that for an object with many close-by neighbors (low k-distance), the local reachability density is high.

Definition 2.5

The local outlier factor of p is defined as:

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|}$$

Intuitively, the local outlier factor of a point p compares the lrd of p to the lrd of all its neighbors by taking the average of the lrd ratios. If the neighboring points have high lrd but the point p has a low lrd, the resulting outlier factor is high, which is interpreted as outlier. The interesting property of this approach is illustrated in Figure 2.1. The scatter

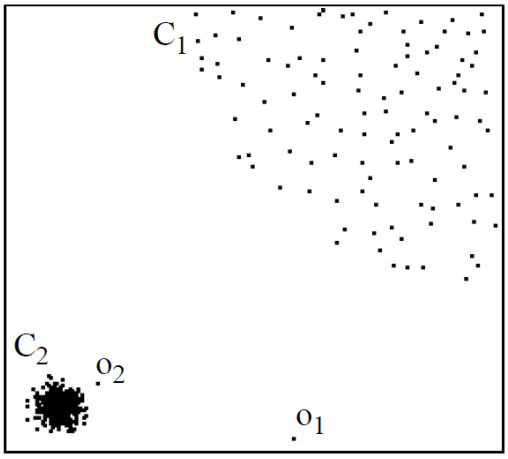


Figure 2.1.: Illustration of LOF. ***Source:*** *[BKNSoo]*

plot features two clusters of different densities and two outliers. With a purely distance-based technique, outlier detection on such a data set can lead to unsatisfactory results. Depending on the distance threshold parameter, the detection would either find:

- For a moderate distance threshold, only object o_1 would exceed the threshold, resulting in a single outlier detection.
- For lower distance thresholds, the points of cluster C_2 would suddenly be below the threshold as well. Therefore, it is possible that all objects in cluster C_2 would be reported as outliers.
- For very low distance thresholds, the same could happen to the points of cluster C_1.

In general, setting the distance threshold is very difficult and it is impossible to detect just o_1 and o_2 without the false detection of clustered objects. With the LOF model this issue does not exist. Since the outlier factor compares an object's density to the density of its direct neighbors, the lrd ratios will yield a value of ≈ 1 for the points in C_1 and C_2. Therefore, the local outlier factor of the points of both clusters will be ≈ 1, and no clustered objects are detected as false-positives. For o_2 on the other hand, the lrd of its direct neighbors is much higher than its own lrd, because the neighbors are part of the very dense cluster C_1. As a result, the local outlier factor of o_2 is $\gg 1$. Note that this result does not have a significant dependence on the parameter $MinPts$. In summary, the main advantages of LOF are its adaption to local densities and the resulting much easier parametrization.

Regarding the complexity, LOF is commonly considered $O(n^2)$, where n is the number of objects, because it relies on nearest neighbor queries. On low-dimensional data, nearest neighbor queries can be speed up, allowing an $O(nlogn)$ implementation. However,

since the required indexing techniques do not scale for high-dimensional spaces, the complexity is $O(n^2)$ in the general case.

The local density idea behind LOF has been pursued by other approaches as well. For instance, LOCI [PKGF03] achieves a local outlier detection by computing the local correlation integral in the object neighborhoods, which for instance facilitates outlier validation by means of so-called LOCI plots. A different work [NOMI10] proposes a modification of LOF which does not need exact nearest neighbor evaluation, resulting in an approximate but faster algorithm.

2.2.3. Angle-Based Outlier Paradigm

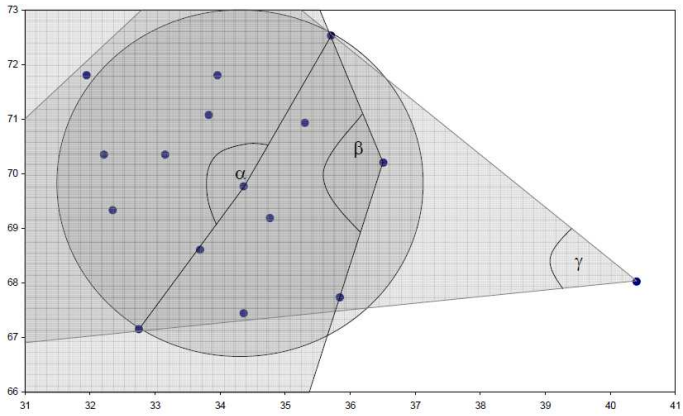


Figure 2.2.: Illustration of the idea behind angle-based outlier models. ***Source:*** *[KShZ08]*

An entirely different paradigm was first introduced in [KShZ08]. Instead of using object distances, this paradigm quantifies an anomaly based on angles to other objects. The idea is illustrated in Figure 2.2. The techniques considers for each individual object the angles that are spanned by all other object pairs. The key observation is that the distributions of these angles are different for clustered objects compared to outliers. For instance for an object right in the center of a cluster, there can be both very small angles – for two objects that lie in a similar direction – and very large angles for object in the opposite direction of the cluster. This means that the variance of the angle distribution is large. For the outlier in Figure 2.2 on the other hand, all possible object pairs will create rather small angles, i.e., the maximum angle γ of this outlier is much smaller than the maximum angle of other objects. Regarding the angle distribution this results in a very low angle variance. The algorithm proposed in [KShZ08], commonly referred to as ABOD, exploits this observation by defining an outlier score based on the variance of the angle distribution. In the most basic version, angle-based outlier detection has a high complexity of $O(n^3)$ with n being the total number of objects, since it has to consider all object pairs for every point. Therefore, [KShZ08] further proposes FASTABOD, which is

an approximate version that restricts the angle evaluations to the k nearest neighbors. This reduces the complexity to $O(n^2 + nk^2)$. In a more recent work [PP12], the complexity of angle-based outlier detection was further reduced to $O(nlogn(d + logn))$ by using AMS sketches (where d is the dimensionality of the data set).

2.2.4. Other Paradigms

We have selected the local-density and angle-based paradigms as examples to illustrate that there are significant formal differences between outlier models. Furthermore, these two paradigms have the highest relevance regarding the scope of this thesis. Overall, there are many more paradigms in the literature, for instance:

- Purely statistical models [RL87]
- Information theoretic models like [SV11].
- Linear projection based approaches like [VCH10].
- Support vector machines can be modified to the case of classifying a single class, resulting in a so-called one-class SVM [SWS+00].
- Another technique [LTZ08] relies on building random decision trees and defines an outlier by the average number of splits required to isolate an object (e.g., an outlier requires less splits compared to inliers).

2.3. Dependence on Application

When applying outlier mining to a given real-world application, the choice of the outlier model is of particular importance. One of the most famous textbooks on outlier mining [Agg13a] summarizes this as: "The data model is everything". To see the importance of the model choice, one has to bear in mind that outlier mining only provides a notion of *unusualness*. This must be differentiated from the notion of *interestingness*: Outlier detection cannot know what is interesting to a data analyst in a particular application domain. In order to redefine interestingness according to application specific goals, additional information would be required. When this additional information on interesting patterns is available, switching to supervised techniques like classification might be beneficial. However, this necessity of high quality training data is exactly the major drawback of supervised techniques. Often training data is sparse, of low quality, or is completely lacking. The big advantage of outlier mining is that it can be applied without any further information requirements, which results in a much broader range of applications. Therefore, the essential question becomes how well the *unusualness* defined by an outlier model matches to the particular *interestingness* in a given problem. There is no definite answer to the question which outlier model is best in each case. Overall the model choice

question is beyond the scope of this thesis. Research on policies to find the ideal model is orthogonal to our research field. For some more details on the proper choice of outlier models, we refer the reader to [Agg13a].

Part II.

Subspace Search for Outlier Mining

3. Challenges

After our summary of traditional outlier models in the previous chapter, we continue with an analysis of open challenges in outlier mining. In modern big data applications, data analysts are often faced with a recurring pattern: Data objects are described by a plethora of attributes. In the past, the amount of information stored per object was often limited by storage space, which was either expensive or inconvenient. As a result of the technical development in recent years, storing a vast amount of data is now feasible, and has become ubiquitous. Therefore, the general challenge in data mining research has become: How to deal with the complexity of high-dimensional data?

In outlier mining, the challenge of high-dimensional data has played an important role in recent years as well [AY01, FMW08, VCH10, MSS11, KKSZ12]. One of the first contributions of this thesis is to summarize all challenges of outlier mining in high-dimensional data by introducing the notion of a ***subspace outlier***. For the discussion of the challenges in this chapter, it suffices to give an informal but general definition of a subspace outlier. In later chapters, we will further refine this notion, resulting in more formal definitions (cf. Chapter 5 and 8).

Definition 3.1

An object $o \in DB$ is a ***subspace outlier*** w.r.t. subspace $S \subseteq \mathcal{A}$, if and only if

- it deviates significantly in subspace S,
- but shows regular behavior in all subspaces $S' \subset S$.

We illustrate this notion by considering an application scenario from medical screening. Typical data attributes in this domain include features like the blood pressure, heart rate, skin conductance, blood glucose, lipoprotein profiles (cholesterol and triglycerides levels), general features like body weight/height, or various features related to a complete blood count. Overall, the resulting data space often has a very high dimensionality in modern medical screenings. However, an outlier in general does not deviate with respect to all attributes at once. With increasing dimensionality, it is more likely that an outlier deviates only w.r.t. a certain subset of the attributes. An example of a typical subspace outlier in the medical screening scenario is illustrated in Figure 3.1 (red marker). The subspace associated with this outlier is the two-dimensional space $S = \{$*systolic blood pressure*,

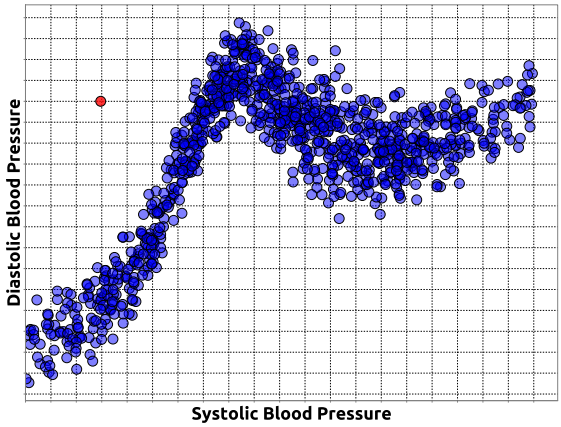


Figure 3.1.: Example of a natural law in health surveillance (fictional data)

diastolic blood pressure}. In this context the object clearly deviates from all other objects, which satisfies the first condition of a subspace outlier. Regarding the second condition, we can see that the object is not anomalous in either $S' = \{$*systolic blood pressure*$\}$ or $S' = \{$*diastolic blood pressure*$\}$ alone, when projecting the object onto the one-dimensional subspaces. In fact, the object even shows very typical values in both dimensions. Thus, it is rather the combination of the two attribute values that is exceptional. This opposing characteristic – abnormality in S, but regularity in $S' \subset S$ – is the main cause of the challenges associated with subspace outliers.

One may also interpret this example from a perspective of natural laws. In general, every data set follows a certain set of natural or domain-specific laws. These laws determine how values of different attributes relate to each other. For instance in medial screening, there is a specific natural law regarding the relation of systolic to diastolic blood pressure (since we are not concerned about the exact law here, the data depicted in Figure 3.1 is fictional). In general, the underlying laws also do not necessarily involve the full set of attributes. Typically, they rather form groups of attributes. In the subspace view, i.e., in the data projection w.r.t. the set of attributes that from a law, the laws manifest in a certain structure. Formally, the data distribution can be described by a manifold of a certain topology. Interpreting the exemplary subspace outlier in Figure 3.1 from the perspective of natural laws yields: The object at hand is an outlier, because it violates the underlying law of systolic and diastolic blood pressure. Later in the thesis (cf. Chapter 5) we will show that there is a general connection between subspace outliers and the underlying domain laws. The takeaway at this point from the example is: In order to deviate from a certain law, it is an intuitive prerequisite that there is some kind of law in the first place.

Challenge 1 – High-Dimensional Invisibility

The first major challenge of subspace outliers is a result from the so-called curse of dimensionality. This effect is observed in many areas of data mining in different manifestations like the insignificance of object distances, meaninglessness of object neighborhoods, and

the empty space phenomenon. Since traditional outlier models are based on concepts that are affected from these issues, the entire detection suffers from the curse of dimensionality. An attempt to detect a subspace outlier in the full-dimensional space is therefore bound to fail. In the context of subspace outliers, a full-dimensional detection means to mix the relevant attributes of the deviating subspace with a potentially huge set of irrelevant attributes. In these remaining attributes, the object may show a fully regular behavior. Therefore, it is possible that the object will appear normal for the most part. Overall, the outlierness assigned to the object will incorporate both the irregularity in a possibly small subspace and the regularity in many other attributes. As a result, the anomalous characteristic may even be balanced by the regular contributions. Based on such a global assessment, full-dimensional outlier detection is prone to overlook the anomaly entirely. In the medical screening example this effect means that whether or not a full-dimensional technique detects the systolic-diastolic-outlier depends on the patient's other attributes: If the patient in addition is malnourished, growth-restricted, and suffering from tachycardia, the detector will surely find the anomaly regarding the blood pressure as well, since irregularities accumulate. However, if the patient is average weight, average height, average age, etc., the anomaly is balanced and the outlier may be missed completely. Clearly, it is preferable to detect any deviation from a natural law, avoiding this balancing influence of high-dimensional data.

To demonstrate the challenge of high-dimensional invisibility, we perform a small experiment on our example data set. As traditional outlier detector we use LOF ($MinPts = 50$), but the results are similar for all traditional models. We illustrate the resulting outlier scores in the form of a scatter plot, in which the size of the markers is proportional to the outlierness – an idea also proposed in [AKR+10]. Figure 3.2 shows the results of LOF calculated w.r.t. three different subspaces: In the first row, we apply LOF directly to the two dimensional subspace {*systolic blood pressure, diastolic blood pressure*}. In this case, the subspace outlier has by far the largest marker. Thus, the object is on rank one in the resulting ranking, allowing a user to spot the anomaly immediately. In the second row of Figure 3.2, we apply LOF to a 7-dimensional subspaces which contains the two relevant plus five irrelevant attributes. In this toy data, we simulate irrelevant attributes by adding uniformly distributed attributes. We also apply a scaling normalization to maintain an equal contribution of each attribute in the underlying distance calculations. We can see that even with only five irrelevant attributes the result has changed significantly: Now, "random" points suddenly have a high outlier score, when examined in the relevant subspace projection. These points simply happen to show a higher global anomalous behavior over the whole 7-dimensional space. The subspace outlier itself has dropped to rank 92 out of 1000 objects. This means that a user now has to examine a rather large result set until the subspace outlier is found. In the third row of Figure 3.2, the same experiment is performed with 10 irrelevant attributes. In this case the subspace outlier drops to rank 394, i.e., users would have to process almost 40% of all objects until they discover the anomaly.

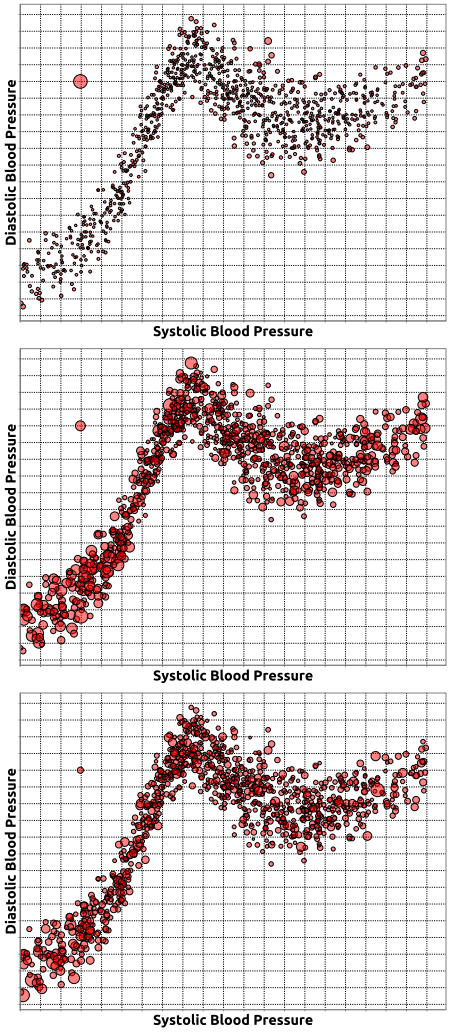


Figure 3.2.: Influence of irrelevant attributes on traditional outlier detection

We can conclude that the visibility of subspace outliers from a high-dimensional view is generally very low and dominated by global effects. Detecting subspace outliers reliably requires to overcome this issue of a global influence. Thus, it is necessary to detect the relevant subspaces themselves.

Challenge 2 – Multivariate Deviation

Another challenging property of subspace outliers is their multivariate nature. Due to visualization limitations, most illustrations of subspace outliers in this thesis will be based on two-dimensional subspace examples. This might suggest that the subspace outlier problem is limited to the bivariate case. However, it is important to note that it in general is a multivariate problem, allowing arbitrary dimensionality of the associated subspaces.

A classic example of this challenge occurs for instance when a database contains percentage values. In the scenario of a medical screening for example, one may measure the composition of blood cells in percentages. The cells that circulate in the bloodstream are generally divided into three types: white blood cells, red blood cells, and platelets. Thus, the respective percentages pct_w, pct_r, and pct_p must each lie in $[0, 1]$ and satisfy $pct_w + pct_r + pct_p = 1$. Figure 3.3 illustrates a data set which follows this law. To show the effect of data errors, we have also added two faulty samples that violate the conditions. In the three-dimensional scatter plot, we can observe these exceptions as clear subspace outliers. In the two- and one-dimensional plots below, we evaluate the second condition in Definition 3.1, which requires that a subspace outlier in S is regular in all subspaces $S' \subset S$. For a three-dimensional subspace outlier this means that it is regular in all six lower-dimensional subspaces. In Figure 3.3 we observe that this is the case for the two objects. Therefore, both objects are indeed multivariate subspace outliers. Such an example can be extended to an arbitrary subspace dimensionality, if the composition has more than three constituents.

Based on this example, we can also discuss the role of the second condition in Definition 3.1, i.e., the requirement that a subspace outlier is regular in all subspaces $S' \subset S$. This condition has two possible interpretations:

- the outlier is not *detectable* in all $S' \subset S$, and
- the outlier is not *describable* in all $S' \subset S$.

Thus, in order to describe the deviation of a subspace outlier, all $|S|$ attributes in the deviating subspace must be considered. For instance in the example from Figure 3.3, it is only possible to explain what is wrong with the outliers by specifying all three dimensions in combination, e.g.: The sum of $pct_w + pct_r + pct_p$ exceeds 1. With respect to any subset of the attributes, there simply is no meaningful description of the anomaly. This duality of detection and description allows us to clarify the motivation behind this second condition of subspace outliers. By requiring regularity in $S' \subset S$ we ensure:

- ***Minimality of Detection:*** Due to the general challenge of high-dimensional spaces, it is always preferable to detect a subspace outlier in the smallest possible subspace. In this minimal subspace, traditional outlier detectors will show the best separation, because no irrelevant attributes are included.
- ***Minimality of Description:*** Similarly, it is our goal to always describe an anomaly by the least number of attributes. Overall, this results in concise descriptions of deviations, without intermixing irrelevant information.

The effect of the minimality condition can also be illustrated based on the blood count example: For instance, an object might as well deviate in all three dimensions at once, e.g., if all three percentages exceed 100%. In this case, it is not necessary to use the combination of the three dimensions – neither for the outlier detection nor for the description of how the object deviates. According to the minimality condition, we do not

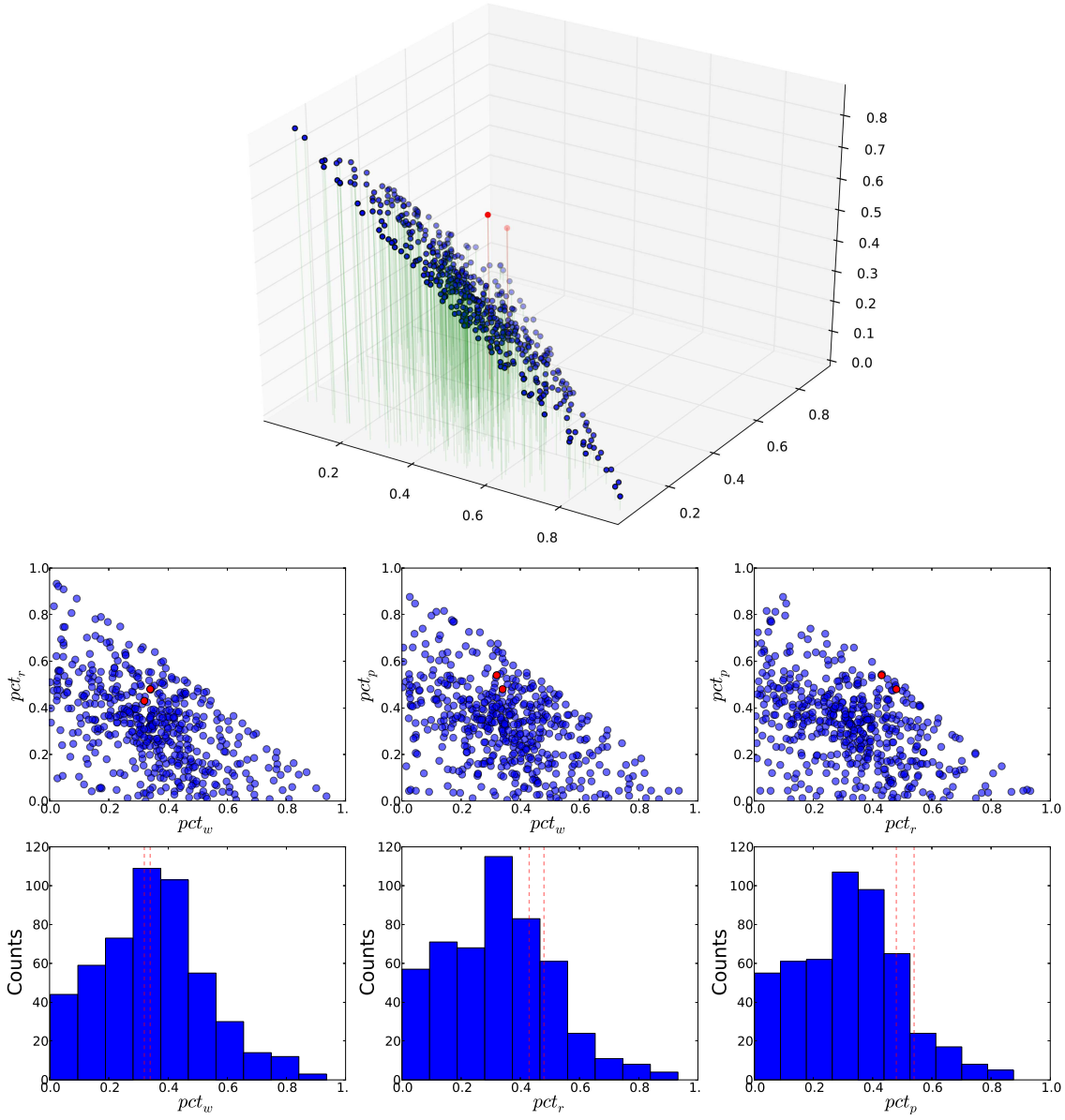


Figure 3.3.: Example of a multivariate subspace outlier; on top: 3D subspace; middle: 2D projections; bottom: 1D projections. Outliers are highlighted by red markers in the scatter plots and red dashed lines in the histograms.

have to consider the more complex three-dimensional subspace here at all. In contrast to the true multivariate outlier from above, the result in this case is more simple: Since the object deviates for instance in *percentage of red blood cells*, it is a subspace outlier in this 1-dimensional subspace. The same reasoning applies to the other two dimensions. Therefore, we can interpret the object as a *multiple* subspace outlier, deviating in three different 1-dimensional spaces. Thus, an important property of the minimality condition

is: It limits the dimensionality of the subspaces to analyze, resulting in a focus on clear, low-dimensional projections. This observation leads to the last challenge of subspace outliers.

Challenge 3 – Multi-View

The discussion regarding Challenge 1 has already shown that the problem of finding subspace outlier cannot be solved by a global view. Overall, the detection of subspace outliers is inherently a multi-view problem because:

- An outlier can have multiple deviating subspaces.
- The set of deviating subspaces for each outlier can be fully independent from deviating spaces of other outliers.

In previous illustrations we have only showed a small number of subspace structures revealing individual outliers. What we cannot illustrate here: In general, such data is governed by a huge number of underlying natural laws and dependencies, and thus features a vast amount of views which may contain subspace outliers. A naive approach to the multi-view challenge is to simply scan all possible subspaces. For a d-dimensional data base, the total number of subspaces is $2^d - 1$. To illustrate: The number of subspace views for a 10-dimensional data base exceeds 1000, for $d = 20$ it exceeds one million, and it is more than one billion for $d = 30$. Thus, searching for subspace outliers is comparable to the proverbial search for a needle in a haystack.

Summary

Overall, the challenges of subspace outliers can be summarized by the observation: Detecting a subspace outlier precisely requires to know its corresponding subspace. Accordingly, the ideal output result of an algorithm is not just a set or a ranking of outliers. Providing the deviating context as well is clearly an additional benefit. This also indicates that outlier mining is connected to finding relationships between attributes: Reporting for instance that the object from Figure 3.1 is a subspace outlier in $\{$*systolic blood pressure, diastolic blood pressure*$\}$ not only tells us something about the anomaly. It also implies that these two quantities follow a certain natural law, which is violated by the outlier. In this thesis we tackle the challenges of subspace outliers by considering possibilities to detect the structures from which an outlier can deviate. In terms of categorizing existing work in the literature, such an approach falls into the category of so-called subspace search techniques. Before proposing our own approaches, we will discuss the relation and differences to related work on subspace search in the following chapter.

4. Related Work

As discussed in the introduction, the first part of this thesis will focus on the key connection between outlier mining and attribute relationship analysis: ***subspace search*** for outlier mining on static data. In the following chapter, we will give an introduction to subspace search in general. We will review existing techniques in this research field with a focus on both the roots of subspace search and techniques that have the highest relevance for this thesis. Furthermore, we will explain the differences of existing paradigms compared to our novel subspace search approaches presented later.

4.1. Subspace Clustering

In the research community, the relation between clustering and outlier mining is commonly considered "conceptually similar, but with opposing goals". Both areas are prime examples for unsupervised learning scenarios. Technically the (dis-)similarity of objects often plays a key role in both paradigms. But while outlier mining is trying to find objects which are unusual, the goal of clustering is to find objects which are highly similar, and thus, form clusters. Overall, clustering has been the slightly more prominent topic in the research community between the two. Accordingly, the notion of a subspace-based[*] data mining approach was first introduced for clustering in [AGGR98]. With the proposed approach, called CLIQUE, a new data mining paradigm emerged: ***subspace clustering***. Technically, the algorithm is a simple grid-based clustering algorithm. However, it is the first clustering technique which has specifically addressed the issues of existing full-space clustering approaches. Figure 4.1 shows the motivating example given in [AGGR98]. In this toy data, it is clear that a meaningful clustering result can only be found w.r.t. the attribute *salary* (clusters C and D). In contrast, the projected density regarding attribute *age* does not have any high-density areas. With a full-space approach, there is no way to distinguish between this difference in the attributes (the clustering results depend entirely on the choice of the density threshold, which therefore is notoriously difficult to chose for full-space clustering techniques). The idea behind CLIQUE is based on a monotonicity of the density of grid cells: If a k-dimensional grid cell is *dense* (density above a certain threshold τ), so are all $(k-1)$-dimensional projections of the grid cell. This allows to

[*] In data mining the term "subspace" has a slightly different meaning from its definition in linear algebra, where it is defined as a subset of a vector space that is closed under addition and scalar multiplication. In data mining and throughout this thesis, a "subspace" refers to a (sub-)set of data attributes.

process all dense grid cells in a bottom-up processing, i.e., starting with all 1-dimensional dense cells, followed by an incremental processing of increasing dimensionality. This processing is similar to the Apriori algorithm [AS94] in frequent itemset mining, where a similar monotonicity property holds for the support of an itemset.

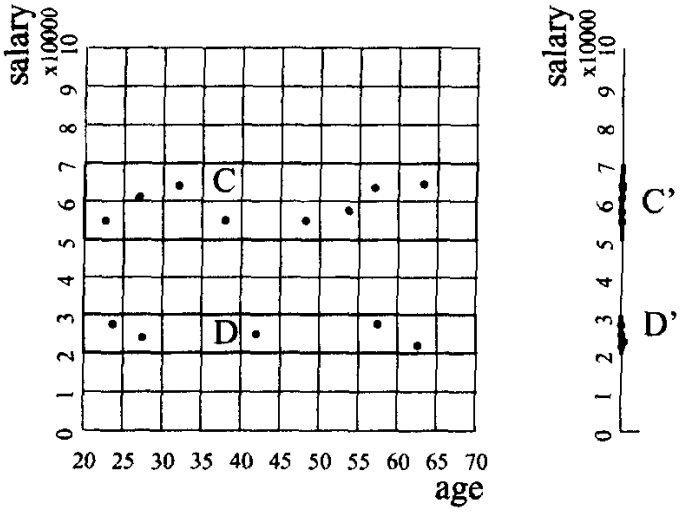


Figure 4.1.: Example of subspace clustering in CLIQUE. ***Source:*** *[AGGR98]*

Overall, CLIQUE is considered a seminal work in data mining, leading to a large number of follow-up techniques on subspace clustering. For instance, MAFIA [GNC99] proposes to use an adaptive-grid to mitigate discretization issues. Another issue, the decreasing density of grid cells for increasing dimensionality, was taken into account in SCHISM [SZ04]: The proposed solution incorporates the Chernoff-Hoeffding bound on the expected density, leading to non-linear monotonically decreasing density threshold function. In order to fully solve the issues of a grid-based clustering, subsequent work focused on applying the density-based clustering paradigm of DBSCAN [EKSX96] to subspace clustering. To this end, SUBCLU [KKK04] introduced the notion of density-connectedness in the subspace context. Later, DUSC [AKMS07] extended this idea by considering the distribution of objects within the neighborhood, leading to a dimensionality-unbiased density measure. In contrast to grid-based approaches, these density-based techniques are able to find clusters of arbitrary shape.

Subspace clustering in general is an example of a ***multi-view*** paradigm: Each object may be part of multiple clusters in multiple subspace projections. Therefore, the computational effort can be high for certain data/parameter combinations, producing an accordingly huge result set of all subspace clusters. This has lead to the modified mining paradigms ***projected clustering*** and ***non-redundant subspace clustering***. In projected clustering, the problem statement is modified towards a single-view perspective: Each object is part of at most one cluster. This simplified problem can lead to a faster processing, at the cost of the very strong limitation on the possible mining results. Examples of projected clustering techniques are PROCLUS [AWY+99], DOC [PJAM02], or P3C [MSE06]. Instead

of restricting the clustering to a single view, the idea behind non-redundant subspace clustering is to prune redundant clustering results. Prominent techniques of this field are INSKY [AKMS08], RESCU [MAG+09], and STATPC [MS08]. For a complete overview of the field of subspace clustering we refer the reader to recent surveys like [KKZ09].

4.2. Subspace Search for Clustering

A major issue of subspace clustering is that each technique comes with it own definition of the notion of "clustered objects". This means that the cluster model is inherently tied to the technique. In many cases these cluster models are very restrictive regarding the detectable cluster shape. For instance, techniques may only be sensitive to grid-like or convex cluster shapes. In this case, subspace clustering technique do not allow to simply exchange the cluster model by a more appropriate cluster definition. This limitation has lead to the development of a new paradigm: ***subspace search for clustering***. The key idea of this paradigm is that the cluster model is generic, i.e., it is possible to plug-in any possible cluster definition. As a result, the search for relevant subspaces becomes a standalone processing step. Figure 4.2 illustrates the conceptual differences of full-space (or fixed-space) clustering, subspace clustering and subspace search for clustering. As a reference, the first row depicts the full-space case, where a clustering method is applied directly to the database. In this case, any conceivable cluster model can be used by changing to a different clustering method. In contrast, this is not possible with subspace search approaches (second row): Here, the search for clusters is coupled to the search for subspaces. Therefore, the cluster model is fixed. Row three depicts the idea of an independent approach for subspace search: Now the two steps are separated, allowing to instantiate the cluster model arbitrarily by any conceivable clustering model. This has a significant benefit, due to the increased flexibility of the processing scheme. For instance, as a result of the decoupling, it is possible to benefit from research progress in either domains: If there is an algorithmic improvement regarding the subspace search step, it can simply be plugged into the system. Or if the future will bring any enhanced cluster models, it is still possible to combine them with existing subspace search approaches.

The first decoupled subspace search approach for clustering is called ENCLUS, published in [CFZ99]. The idea of ENCLUS is to quantify the quality of a subspace based on entropy. The paper shows that a good clustering in general requires a subspace with a low entropy. Furthermore, as a result of basic characteristics of conditional entropy, it is possible to derive a downward closure on the quality criterion. In order to reduce the search space, ENCLUS proposes two variants of removing redundant subspaces during processing. One of the major issues of ENCLUS is that it is grid-based. As a result of this, ENCLUS is strongly affected by the curse of dimensionality in the form of the empty space problem: Using a grid that has a meaningful cell size on the 1-dimensional level, leads to an explosion of the number of high dimensional grid-cell. With increasing dimensionality, the object

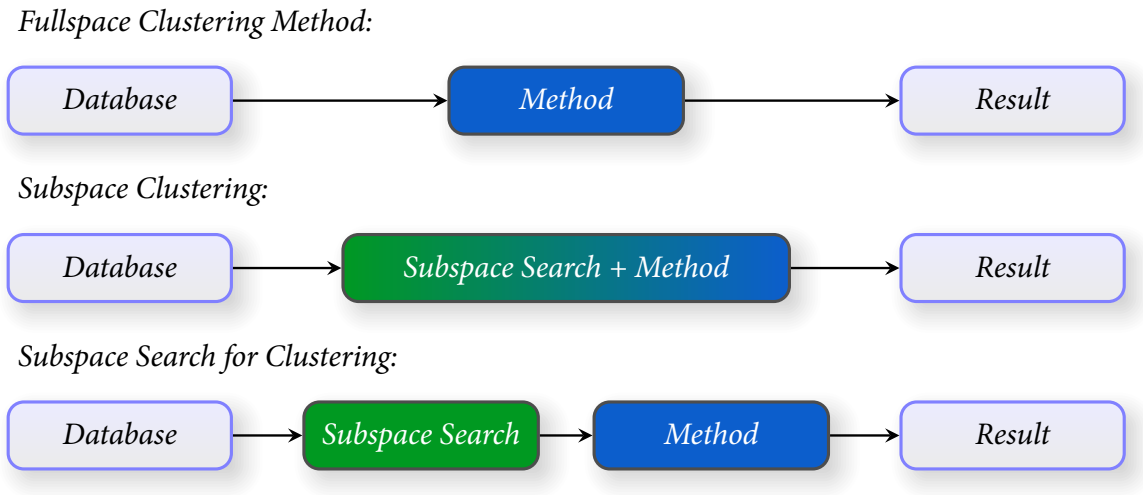


Figure 4.2.: Conceptual difference of subspace mining paradigms

counts in the cells tend to zero, and the object count of populated grid cell will typically be just one. Therefore, ENCLUS struggles to detect high dimensional subspaces for the subsequent clustering step.

Another important subspace search technique for clustering is RIS [KKKW03], acronym for "ranking interesting subspaces". This work defines the interestingness of a subspace based on the number of "core" objects it contains, which are objects with a particularly high neighbor density. Compared to ENCLUS, RIS however ignores the data distribution in the subspace. This approach has been further extended in [BPR+04], proposing a technique called SURFING. In [NDJ10], the idea of subspace search was also extended to spectral clustering.

4.3. Subspace Outlier Mining

We continue with our synopsis of subspace mining, but now changing the subject from the world of clustering to its close sibling outlier mining. Similar to clustering we will differentiate between techniques that tightly couple subspace mining to an outlier model and techniques that perform a generic subspace search. Again we will start with a survey of coupled techniques in this section.

Outlier detection in subspaces has first been proposed by [AY01]. The motivating example transfers the notion of multiple views from clustering to outlier mining. The idea is illustrated by a similar example as our illustration of an "outlier hidden in a subspace view" given in the introduction 1.4.1 (cf. Figure 1.4). The technique is inherently grid-based and tries to find grid cells of exceptional sparsity. The approach is based on a very simple binary

outlier model: An object is considered an outlier if it falls into one of the top m sparse cells returned by their main algorithm. The authors argue that the search for sparse cells cannot be performed systematically. Therefore, they resort to an evolutionary approach to process the search space. While the paper is conceptually groundbreaking, the algorithm itself often does not work very well on real-world data. To some extend, this can be explained by the issues of grid-based techniques and the challenges in implementing reasonable selection/crossover/mutation steps for an evolutionary processing. Furthermore, the technique requires to specify k, the dimensionality of hidden subspace structures. From the user perspective this choice is highly non-trivial: It not only requires to know the dimensionality of the hidden subspace outliers in advance, it also ignores the fact that data may have subspace structures of different dimensionality. However, the approach also has a conceptual issue, since it relies on finding *all* sparse grid cells. This issue becomes obvious when attributes show a strong correlation, which is often the case in real-world data. For instance, consider a two-dimensional subspace, which is 100% correlated and uniformly distributed, resulting in a straight line structure in the subspace. If we use a binning of 10 bins in each dimension, there is a total of 100 grid cells in the subspace. Among these 100 cells, only 10 cells will be populated, and the majority of 90% of the cells will be fully sparse (except for outliers). In general, an outlier can be located in any of these sparse cells. This means that in order to consistently detect outliers, it is necessary to always scan all sparse cells. However, with higher dimensional structures the amount of sparse cells will become huge. After all, the empty space problem is a well known effect in high-dimensional data. Thus, the requirement to iterate over the whole empty space becomes a significant computational burden.

Recent approaches have enhanced subspace outlier mining by ranking objects based on different subspace projections [KKSZ09, KKSZ12, MSS10, MSS11, MASS08, MAIS+12]. These techniques differ in their definition of a relevant subspace and their associated outlier model. For instance, the aim of SOD [KKSZ09] is to detect subspace structures which lie on a linear, axis-aligned manifold. This implies a highly specialized outlier definition, which does not allow to use the technique as a general-purpose outlier detection. The same authors have extended the idea of SOD in [KKSZ12], allowing arbitrarily oriented linear manifolds. However, the resulting outlier model is still not general, due to the strong requirement of linearity.

This has been improved in more advanced techniques like OUTRES [MSS10, MSS11]. OUTRES ranks outliers based on the object's deviation in a statistically selected set of relevant subspace projections. This is achieved by introducing a subspace ranking function, which aggregates the outlierness of an object over all its relevant subspaces. OUTRES considers a subspace relevant, if data is not uniformly distributed in the subspace. This criterion is evaluated by means of a statistical test and on a per-object basis, using the local neighborhood of each object. Finally, OUTRES addresses the challenge of comparing outlier scores obtained from subspaces of arbitrary dimensionality by introducing the notion of an adaptive outlierness. This however means that it is also tightly coupled to its own outlier model.

A unique technique of subspace outlier mining is presented in [MASS08, MAIS+12]. This work differs from others of this field, since it bridges the gap between clustering and outlier mining: The proposed algorithm OutRank relies on an outlier definition that is based on the results of subspace clustering. In fact, the technique does not even access the original data to detect outliers. The notion of an outlier is entirely derived from the membership relation of each object to the sets of clusters in all subspace projections.

4.4. Subspace Search for Outlier Mining

Similar to the situation in clustering, subspace outlier mining relies on an interleaved detection of both subspaces and outliers. This means that these techniques all propose their own outlier criterion that is specific to its subspace processing. They are restricted to this outlier notion, and thus, are not flexible w.r.t. instantiations with different outlier models. In clustering, the decoupling of the cluster model from the subspace search has had obvious advantages like the mutual benefit of both research domains. Therefore, it is natural to ask for the same decoupling for outlier mining as well. This however has been rarely studied in existing research. The only approach that may count as a subspace search for outlier mining is RandSub [LK05]. However, the idea of this technique is to perform a "subspace search" in its most naive form: The algorithm simply creates a set of random subspace projections, and evaluates a given outlier model in these subspaces. This clearly qualifies as a decoupling of the outlier model, and the overall processing can be illustrated similar to the situation in subspace search for clustering from Figure 4.2. Obviously, the random subspace selection cannot guarantee any quality criterion for the selected subspaces. Whether or not a subspace is at all relevant for outlier mining is not considered. Nevertheless, it achieves an improvement regarding outlier detection quality simply by mitigating the curse of dimensionality for the subsequent outlier detection.

In the context of related work, we can describe the major contribution of the first part of this thesis as follows: We will propose subspace search techniques which allow a flexible instantiation of the outlier model. Our techniques thus are the first true subspace search techniques designed for outlier mining. Compared to the naive random subspace selection of [LK05], our aim is to develop enhanced subspace quality measures, resulting in a meaningful subspace search process. In this regard, we consider [LK05] a baseline for any more advanced subspace search technique.

4.5. General Categorization of Subspace Search Approaches

In Figure 4.3, we have summarized related work along with our categorization of the different research fields. Furthermore, it shows how this first part of the thesis is related

	Clustering	*Outlier Mining*
Models	• DBSCAN [EKSX96] • ...	• LOF [BKNS00] • LOCI [PKGF03] • ...
Subspace Mining (Fixed Model)	• CLIQUE [AGGR98] • SUBCLU [KKK04] • MAFIA [GNC99] • SCHISM [SZ04] • DUSC [AKMS07] • ...	• [AY01] • SOD [KKSZ09] • COP [KKSZ12] • OUTRES [MSS11] • OUTRANK [MAIS+12]
Subspace Mining (Flexible Model)	• ENCLUS [CFZ99] • RIS [KKKW03] • SURFING [BPR+04]	• RANDSUB [LK05]

Figure 4.3.: Categorization of related work. Highlighted: Category corresponding to the techniques developed in this thesis.

to these fields: Overall, outlier mining and subspace search are both established topics in the research community. Our concern here will be the novel combination of these fields.

4.6. Remotely Related Work

There is more related work beyond what we have covered in the previous section. Overall, these studies are only loosely related to our work. Moreover, they do not fit into our categorization from Figure 4.3. Therefore, we discuss these topics individually in the following.

4.6.1. Mining Descriptions for Given Outliers

There are several approaches that identify attribute sets as so-called outlier descriptions [KN99, AFP09, LB08]. The obtained outlier descriptions can be interpreted as a special kind of subspace selection. However, these methods extract subspaces only for given outliers, assuming that a faultless outlier detection has taken place in advance. In most real-world scenarios, such proper outlier labels do not exist. Obviously this results in a *chicken and egg dilemma*: (1 → 2) In order to detect outliers hidden in subspaces, traditional outlier detectors require a *prior* subspace selection. (2 → 1) Outlier descriptions would

provide such a subspace selection, but they require the outliers to be detected *in advance*. Therefore, the goal in this thesis is to break this cyclic dependency. The techniques proposed here will not require outlier labels to be known in advance. The relation of outlier description and subspace search will be the focus of Chapter 7.

4.6.2. Dimensionality Reduction

At a first glance, subspace search seems to be related to dimensionality reduction, since both are motivated by the curse of dimensionality. Considering its most popular but basic variant, principal component analysis [Jol86], dimensionality reduction has a long history in research. In general, dimensionality reduction aims at transforming the original space into a new space, while maintaining certain properties of the data. For instance in PCA, the target space is obtained by the orthogonal linear transformation which produces the largest possible variance. Such a linear transform can also be used to detect outliers in the target space, which has been studied for instance in [FMW08]. Since linear transformations only work well for very simple data distributions, the dimensionality reduction has been extended to non-linear embeddings [FL95, TSL00, BN03, SC08, BPV03], which share the key idea behind spectral clustering [Lux07]. Recently, such a non-linear embedding approach was proposed specifically designed for outlier detection [HQYY12]. However, all dimensionality reduction techniques have a fundamental difference to subspace search: In dimensionality reduction the goal is always to find a single target space, in which information on all objects is maintained as far as possible. This single-view restriction always comes with a major drawback: For individual objects, a globally determined space may not be an appropriate embedding. By design, determining a transformation globally must focus on the majority of objects, in order to work well on the data as a whole. Since in outlier mining the focus is always on a few individual objects, the single-view paradigm impedes the detection of these individual contexts [MSS11]. By contrast, subspace search always allows a multi-view perspective on the data. In order to corroborate this reasoning empirically, we have included dimensionality reduction techniques as a competitor in our experiments (cf. for instance Chapter 5.4).

5. Subspace Search for Outlier Mining: High Contrast Subspaces*

5.1. Introduction

In this chapter we will present the first subspace search approach that is tailored to the detection of subspace outliers. The key idea behind this approach is to exploit the connection of outlier mining and attribute relationships. In the chapter on challenges posed by subspace outliers (Chapter 3), we have analyzed subspace outliers from the perspective of natural laws. This has lead to the observation that violations of the underlying principles of the data manifest themselves as subspace outliers. Therefore, we propose to search for subspace outliers based on a detection of the underlying structures in a data set.

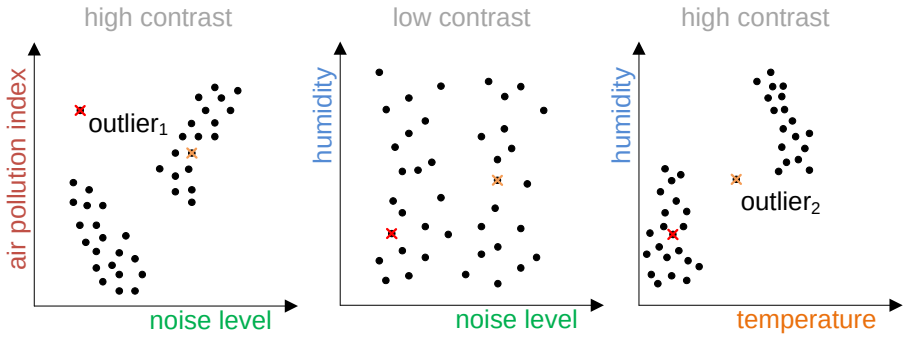


Figure 5.1.: Environmental surveillance example – suspicious sensor readings

We illustrate the idea in Figure 5.1, which shows an example scenario from environmental surveillance. In such a data set, certain groups of attributes will show some kind of relationship. For instance it is likely to observe a relation between measurements of the *noise level* and the *air pollution index*. Since the data distribution in this subspace is governed by an underlying domain-specific law, the corresponding subspace view features a distinctive structure: Low noise levels correspond to less air pollution, and vice versa, more noise often involves higher air pollution. In terms of outlier mining, such a law provides a high contrast for the detection. We can see that one object violates

* This chapter is an extended version of *HiCS: High Contrast Subspaces for Density-Based Outlier Ranking* published in the Proceedings of the International Conference on Data Engineering (ICDE) 2012 [KMB12].

the general pattern, which is manifested in a clear subspace outlier. On the other hand, there are attribute combinations which do not follow any relation. For instance, there is no evidence that *air humidity* is related to the *noise level* in any way. In this case, the subspace view will not show a clear structure. Since there is no pattern to deviate from, the subspace has a low detection contrast for outlier mining. Therefore, it makes sense to exclude meaningless attribute combinations for outlier analysis.

In this chapter, we will propose a technique called HiCS (High Contrast Subspaces), which tackles the challenges of subspace outliers by searching for subspaces which show strong attribute relationships. Such a subspace search approach requires the development of novel quality criteria and processing schemes. Overall, we propose to use a two-step processing:

(1) ***Subspace Search:*** Measures the contrast of subspaces

(2) ***Outlier Detection:*** Evaluates objects in high contrast subspaces

In this work, we focus on the first step. As outlier score for the ranking we rely on the commonly used local outlier factor (LOF) [BKNS00]. However, any other outlier score could be used as instantiation of the second step. Thus, in contrast to existing techniques, we follow the idea of a decoupled processing, which has emerged as superior processing scheme in other domains (cf. Chapter 4). In outlier mining, our approach is the first work that considers subspace search as an individual problem.

Technically, the main idea of our HiCS approach is the statistical selection of high contrast subspaces. We propose a processing based on a series of statistical tests. Each test compares the data distribution in a local subspace region to its marginal distribution. Dependencies between attributes highlight the high contrast of a subspace. Based on these statistical tests and the detected dependence between attributes we derive our contrast measure. Thus, our approach searches for high contrast subspaces with a significant amount of conditional dependence among the selected dimensions, revealing subspaces corresponding to the underlying domain laws. As a result, we enhance the quality of traditional outlier detection by computing outlier scores in high contrast projections only. The evaluation on real and synthetic data shows that our approach outperforms traditional dimensionality reduction techniques [Jol86], naive random projections [LK05] as well as state-of-the-art subspace search techniques [CFZ99, KKKW03] and provides enhanced quality for outlier rankings. In summary, the work in this chapter provides three major contributions:

- The *decoupling of subspace search* as generalized pre-processing step for outlier ranking.
- A *contrast measure* based on the conditional dependence of dimensions in the selected subspaces.
- Two *robust implementations* of our contrast measure based on two different statistical tests.

5.2. High Contrast Subspaces

In the following, we will introduce the necessary notation in Section 5.2.1, and define the general objective for a high contrast subspace selection in Section 5.2.2. We will introduce the notion of subspace slices that specify local subspace regions in Section 5.2.3, and define the contrast measure in Section 5.2.4. In Section 5.2.5 we will show how different statistical tests can be used to instantiate our contrast definition.

5.2.1. Notation

Let DB be a database containing N objects, each described by a D-dimensional real-valued data vector $\vec{x} = (x_1, \ldots, x_D)$. The set $\mathcal{A} = \{1, \ldots, D\}$ denotes the full data space of all given attributes. Any attribute subset $S = \{s_1, \ldots, s_d\} \subseteq \mathcal{A}$ will be called a d-dimensional subspace projection. We denote the distance between objects $\vec{x}$ and $\vec{y}$ as $dist_{\mathcal{A}}(\vec{x}, \vec{y})$, which can be instantiated for instance by the widely used Euclidean distance $dist_{\mathcal{A}}(\vec{x}, \vec{y}) = \sqrt{\sum_{s \in \mathcal{A}} (x_s - y_s)^2}$.

As general property of any outlier ranking method we have to consider the underlying scoring function. It measures the outlierness of an object. Traditionally, each object is sorted according to a single outlier score $score(\vec{x})$ measuring the degree of deviation in all given attributes $\mathcal{A}$. Traditional density-based outlier scores measure the density $p(\vec{x})$ of an object and compare it to the density in the local neighborhood of $\vec{x}$. Local outlier ranking based on density deviation in local neighborhoods has first been proposed by LOF [BKNS00]. In recent years, this outlier mining paradigm has been extended by enhanced scoring functions and efficient outlier ranking algorithms [PKGF03, BS03, GPO08, KShZ08, KKSZ11, MSS10, VCH10].

The problem with all of these full space approaches is introduced by the curse of dimensionality. As pointed out in [BGRS99], the definition of a local neighborhood becomes meaningless for a large number of attributes. Furthermore distances between objects grow more and more alike, thus

$$\lim_{|\mathcal{A}| \to \infty} \max_{\vec{z} \in DB} dist_{\mathcal{A}}(\vec{z}, \vec{x}) - \min_{\vec{z} \in DB} dist_{\mathcal{A}}(\vec{z}, \vec{x}) = 0$$

Since local outlier ranking calculates the density based on the object distances, we observe the same effect for the minimal and maximal value of $score(\vec{x})$. As a result, all mentioned outlier score functions will suffer from a loss of contrast, i.e.:

$$score(\vec{x}) \approx score(\vec{y}) \quad \forall\, \vec{x}, \vec{y} \in DB$$

Any outlier ranking obtained for a sufficiently high dimensional database will degenerate into a random ranking with very similar scores for all objects.

Subspace outlier rankings address this problem by evaluating the score function in lower dimensional subspace projections. They simply restrict the distance computation to a selected subspace S, i.e., compute $dist_S$. Thus, any outlier ranking with $score(\vec{x})$ can be extended to a subspace score $score_S(\vec{x})$. The idea is to aggregate these $score_S(\vec{x})$ values over several subspaces. Each score provides some insights about the deviation of $\vec{x}$ in a lower dimensional projection S. The final ranking is derived from the aggregation of these scores:

Definition 5.1

Outlier Score

$$score(\vec{x}) = \frac{1}{|RS|} \sum_{S \in RS} score_S(\vec{x})$$

In the most basic approach [LK05], RS is a selection of random subspaces that contribute to the overall ranking. A major drawback of this approach is that irrelevant subspaces in RS might blur the overall order of objects. To tackle this challenge, we propose a novel method to select high contrast subspaces only. Our subspace search technique excludes low contrast subspaces, which inhibit a clear distinction between outliers and regular objects.

For our experiments, we instantiate $score_S(\vec{x})$ with the commonly used local outlier factor [BKNS00]. It has been used for the subspace extension based on random projections [LK05] as well. However, our technique is not restricted to LOF only. Any other density-based scoring function could be used for $score_S(\vec{x})$. This flexibility w.r.t. the score function is a main advantage of our method. We only consider the contrast of subspaces and their selection as pre-processing step. Any improvement in the area of outlier scoring can be applied directly to our approach as well. In recent years several extensions of LOF have addressed specific challenges for this local outlier ranking [PKGF03, KShZ08, MSS10, KKSZ11]. While each of these publications proposes an individual score function, they all have an assumption in common: ***An outlier has low density compared to its local neighborhood.*** Our technique relies only on this general assumption.

To derive our criterion for subspace contrast, we treat the attributes in DB as random variables. We use the notion of probability density functions (pdf) to derive the formal background of our contrast criterion. We will adapt the notation for subspaces as follows. For a given subspace $S = \{s_1, \ldots, s_d\}$, we refer to the projected data vectors as $\vec{x}_S = (x_{s_1}, \ldots, x_{s_d})$.

Notation 5.1

The subspace data vector $\vec{x}_S$ is distributed by an unknown ***joint pdf*** of S:

$$p_{s_1,\ldots,s_d}(x_{s_1}, \ldots, x_{s_d})$$

By integration over all attributes $s \in \mathcal{A} \setminus s_i$ we obtain:

NOTATION 5.2

The ***marginal pdf*** of attribute s_i:

$$p_{s_i}(x_{s_i})$$

Please note that the marginal densities are simply one-dimensional projections, independent from any subspace. Furthermore, we can require a condition on the attributes $s \in S \setminus s_i$, which leads to the following notion.

NOTATION 5.3

The ***conditional pdf*** of attribute s_i:

$$p_{s_i \mid s \in S \setminus s_i}\left(x_{s_i} \mid \{x_s \ : \ s \in S \setminus s_i\}\right)$$

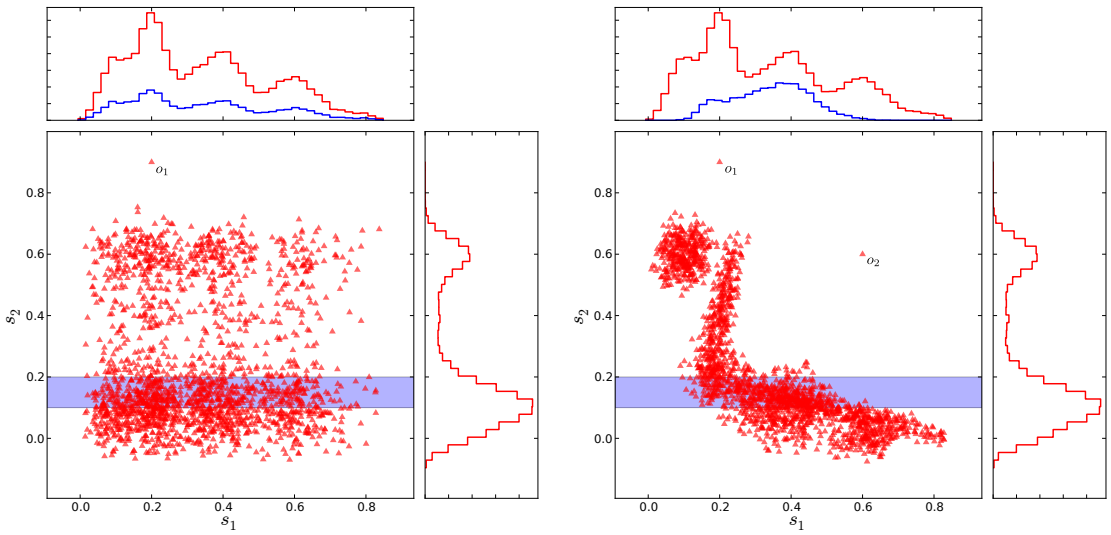
Thus, we express the probability density function of s_i w.r.t. $|S| - 1$ conditions on all other attributes in the subspace.

5.2.2. Objectives

Given the notion of probability density in any subspace S, we can formalize our objectives for subspace search for outlier detection. Our approach is based on a distinction between *trivial* and *non-trivial* outliers, and their relation to what we will call *correlated subspaces*. These notions are new concepts and we will postpone the formal definition for a moment. They are related to our former definition of a subspace outlier in Chapter 3, as we will see in the following.

Motivating Example

We illustrate the relationship between correlated subspaces and trivial/non-trivial outliers by a toy example (cf. Figure 5.2). It consists of two two-dimensional datasets. Both datasets were generated from the same marginal distributions. In dataset A, s_1 and s_2 are completely independent. As a result, this two-dimensional subspace is filled by a random scattering of objects in consistency with the marginal distribution. Nevertheless the dataset contains an outlier object o_1. By considering the one-dimensional projections of this subspace, the existence of o_1 is not a surprise: o_1 could trivially be detected by the examination of the one-dimensional distribution of attribute s_2. We call such an object



(a) *Dataset A – example of an uncorrelated joint pdf* (b) *Dataset B – example of a correlated joint pdf*

Figure 5.2.: high vs. low contrast and the effects on outlier ranking

a *trivial outlier*. In summary, the evaluation of the two-dimensional subspace does not reveal any new information for this dataset.

The other dataset features marginal distributions identical to the ones of dataset A. The difference is that in dataset B there is a distinctive relationship between the attributes s_1 and s_2. This correlation allows the data objects to form regions of varying or unexpected densities over the total possible area that would be consistent with the marginal distribution. We observe (a) cluster-like dense agglomerations of objects and (b) sparse or even empty regions. Besides a trivial outlier o_1, the subspace also features an other outlier o_2. This time the outlier is hidden in all one-dimensional subspace projections, where it even appears to be a clustered object. We will call this type of objects non-trivial outliers. For dataset B the evaluation of the two-dimensional subspace was worthwhile and reveals significant insight regarding the data structure. Accordingly, we have found an example for a high contrast subspace in this case.

Once we have found such a high contrast subspace we can apply any density-based outlier ranking algorithm: For instance in dataset B, o_1 and o_2 both exhibit a much lower density compared to the local neighborhood. Thus, determining the outlierness in the two-dimensional subspace of dataset B would result in a detection of o_1 and o_2, i.e., $score_S(o_{1/2}) \gg score_S(o_i)$ for all other objects o_i in the database.

We can also explain the essential idea of our approach to identify high contrast subspaces using this toy example. Depicted on top of each plot in Figure 5.2, we show two different histograms for the s_1 axis of both datasets. The first one (red) represents the full data

sample, i.e., corresponds to the marginal probability distribution $p_{s_1}(x_{s_1})$. The blue one shows the conditional probability distribution that is generated by the sample according to the selection range w.r.t. the s_2 axis (blue area). The comparison of the blue vs. the red histograms for both datasets show a basic property of correlation: Whereas the histograms for dataset A are in good agreement, we see a significant discrepancy between the two histograms for the high contrast subspace B. The proposed HiCS algorithm is based on the evaluation of this discrepancy.

Please note that we design our contrast measure as a conservative subspace selection criterion. The set of selected subspaces is a proper superset of the subspaces containing non-trivial outliers. We will later show that high contrast is a necessary condition for non-trivial outliers. Still, the result may contain subspaces without any outliers.

In the following we will focus on non-trivial outliers only. The reason is simple: A user might already know about the existence of one-dimensional outliers; One can detect these outliers by existing methods [RL87] without difficulty. Moreover, our subspace search can detect trivial outliers as a by-product of the search for non-trivial outliers. For instance in dataset B, we will always detect o_1 as outlier as soon as attribute s_2 is part of any high contrast subspace. In any case, the detection of non-trivial outliers will provide a much higher information gain to the user. Therefore, we focus on the detection of correlated subspaces containing such non-trivial outliers.

Contrast Based on Dependence of Attributes

In probability theory, two events A and B are called independent and uncorrelated, if and only if the probability of the combined event is given by the product of the individual probabilities, i.e.:

$$p(A \cap B) = p(A) \cdot p(B) \tag{5.1}$$

By putting the notion of correlation in the context of subspaces, we obtain:

Definition 5.2

A subspace S is called an ***uncorrelated subspace*** if and only if:

$$p_{s_1,\ldots,s_d}(x_{s_1},\ldots,x_{s_d}) = \prod_{i=1}^{d} p_{s_i}(x_{s_i}) \tag{5.2}$$

Please note that the formal distinction between statistical dependence and correlation is not important for our purpose. Strictly speaking, the term *set of independent attributes* would be the appropriate expression. Instead we prefer to use the more concise term *uncorrelated subspace* to express the statistical independence within a subspace.

To support the observations regarding Figure 5.2, we want to examine the characteristics of outlier mining in uncorrelated subspaces more formally. The observation of a high value of $score_S(\vec{x})$ implies that the object $\vec{x}$ is located in a region with a low value of the joint pdf $p_{s_1,\ldots,s_d}(x_{s_1},\ldots,x_{s_d})$. On the other hand, we can evaluate the expected density for $\vec{x}$ under the assumption of an uncorrelated subspace:

$$p_{expected}(x_{s_1},\ldots,x_{s_d}) \equiv \prod_{i=1}^{d} p_{s_i}(x_{s_i}) \tag{5.3}$$

We define the notions of trivial/non-trivial outliers over the comparison of the expected density with the joint density:

Definition 5.3

We call an object $\vec{x}_S$ a ***non-trivial outlier*** w.r.t. subspace S if

$$p_{s_1,\ldots,s_d}(x_{s_1},\ldots,x_{s_d}) \ll p_{expected}(x_{s_1},\ldots,x_{s_d}) \tag{5.4}$$

Note that this definition is related to the notion of a subspace outlier from Definition 3.1 in Chapter 3. More specifically, non-trivial outliers are a generalization of subspace outliers, because they relax the invisibility condition to one-dimensional projections. This means that our goal here is more general, i.e., we aim at detecting a superset of true subspace outliers. This simplifies the formalism and obviously still guarantees the detection of all subspace outliers, since they are a subset of non-trivial outliers.

Incorporating the definition of an uncorrelated subspace (Eq. 5.2) into the definition of non-trivial outliers leads to:

Theorem 5.1

An uncorrelated subspace S does not contain any non-trivial outlier.

This follows immediately from Definition 5.3: For an uncorrelated subspace, the joint probability density function $p_{s_1,\ldots,s_d}(x_{s_1},\ldots,x_{s_d})$ is by definition equal to the product of the marginal pdfs and thus, will never fulfill Eq. 5.4. On the other hand, a correlated subspace allows significantly smaller values of $p_{s_1,\ldots,s_d}(x_{s_1},\ldots,x_{s_d})$ compared to the expected density. Thus, we define subspace correlation as objective function for the subspace contrast.

Measuring Correlation

We propose to quantify the subspace contrast by a comparison of different probability density functions. To simplify the notation, we will express all following conditional

probability densities only for s_1 without loss of generality. In the case of an uncorrelated subspace, Eq. 5.2 simplifies the definition of all conditional probability densities within the subspace, i.e.:

$$\begin{aligned} p_{s_1}(x_{s_1}|x_{s_2},\ldots,x_{s_d}) &= \frac{p_{s_1,\ldots,s_d}(x_{s_1},\ldots,x_{s_d})}{p_{s_2,\ldots,s_d}(x_{s_2},\ldots,x_{s_d})} \\ &= p_{s_1}(x_{s_1}) \end{aligned} \tag{5.5}$$

This allows to measure the contrast of a subspace by determining the degree of violation of Eq. 5.5. In other words, we have to compare a conditional pdf of s_1 to the corresponding marginal pdf, and we assign a high contrast to a subspace if we observe a significant deviation between the two pdfs. Please note that the correlation analysis within subspaces goes beyond classical correlation analysis approaches, since we may be faced with high contrast subspaces with more than two dimensions. In contrast to, say, the Pearson or Spearman correlation coefficient [Spe87], the proposed approach is not limited in the subspace dimensionality. Furthermore, it is possible to detect any kind of non-linear correlation. Above all, our approach does not require an evaluation of a high dimensional joint pdf, but is based on one-dimensional densities only. Hence, it does not fall prey to the curse of dimensionality.

In the following sections we will discuss (1) how to empirically analyze the the conditional pdf by introducing the notion of *subspace slices*, (2) how to compare the conditional pdf to the marginal pdf by means of statistical tests, and (3) how to instantiate these statistical tests in our contrast measure.

5.2.3. Evaluation of Conditional Densities

The main challenge for the proposed calculation of the subspace contrast is the empirical analysis of the conditional probability densities $p_{s1|\ldots} \equiv p_{s_1|s_2,\ldots,s_d}(x_{s_1}|x_{s_2},\ldots,x_{s_d})$. Since we do not require any knowledge of the underlying density functions, our goal is to obtain a sample of $p_{s1|\ldots}$ for a specific set of conditions.

Definition 5.4

A set of $|S|-1$ lower and upper conditions $[l_i, r_i]$ is called a ***subspace slice*** w.r.t. subspace S:

$$C = \{x_{s_2} \in [l_2, r_2], \ldots, x_{s_d} \in [l_d, r_d]\} \tag{5.6}$$

The selection of objects that satisfy a subspace slice condition leads to a subsample of DB with a sample size N'. The advantage of these subspace slices over any grid-based density estimation is that we can construct the subspace slices in a way that does not suffer from the curse of dimensionality. The goal is to choose the intervals in the subspace slice C in

such a way that the expectation value for the selection sample size N' is fixed. We derive the construction of the intervals as follows: Each condition in C can be associated with a certain selection of objects. Starting with the full sample of $|DB|$ objects, each selection removes a certain fraction of objects from the current sample. We denote the fraction of objects that will remain in the sample by $\alpha_1 \in (0,1)$. The suffix emphasizes that α_1 is the probability of an object to be selected in a single condition. By assuming an uncorrelated subspace, the selections are independent from each other. In this case the probability for a single object to be selected after $|C|$ equally probable selection steps is $\alpha_1^{|C|}$. Thus, the expectation value of the remaining sample size N' after $|C|$ selections is given by:

$$E\left[N'\right] \;=\; N \cdot \alpha_1^{|C|} \tag{5.7}$$

We can utilize this step-wise selection in the algorithm to generate subspace slices that automatically adapt the selection intervals $[l_i, r_i]$ to provide a desired target statistic size N', independent of the dimensionality of the subspace. The implementation details are given in Section 5.3.1.

5.2.4. Quality Criterion for Subspace Contrast

As mentioned before, our subspace contrast definition is based on the degree of violation of Eq. 5.5. Since we do not require density functions explicitly given, we introduce the following notation to emphasize that we refer to estimated density distributions from a data sample:

- $\hat{p}_s$ refers to the marginal density of some attribute $s \in S$ w.r.t. the full dataset.
- $\hat{p}_{s|C}$ refers to the density of x_s w.r.t. the remaining dataset that fulfills a certain condition set C.

We are now looking for a function $deviation\left(\hat{p}_s,\ \hat{p}_{s|C}\right)$ that compares $\hat{p}_s$ to $\hat{p}_{s|C}$, measures the discrepancy between the two distributions and outputs a value that is proportional to the deviation. There are many ways to define such a function. With HiCS we focus on two different statistical tests, namely Welch's t-test and the Kolmogorov-Smirnov test, which will be described in Section 5.2.5. We will call the two resulting variants HiCS_{WT} and HiCS_{KS}.

In terms of statistical testing, we define the null hypothesis as: *Both samples originate from the same underlying pdf.* In other words, the null hypothesis states that the differences between $\hat{p}_s$ and $\hat{p}_{s|C}$ are within the limits of statistical fluctuations. Due to these fluctuations, the significance of a single statistical test is very limited. In order to achieve a high statistical precision, the HiCS algorithm performs a large number M of different tests. Thus, the definition of our quality criterion of the subspace contrast is given by:

Definition 5.5

Subspace contrast

$$contrast(S) \equiv \frac{1}{M} \sum_{i}^{M} deviation\left(\hat{p}_{s_i},\ \hat{p}_{s_i|C_i}\right) \tag{5.8}$$

HiCS computes the subspace contrast with a Monte Carlo approach. The algorithm performs M iterations. For each iteration, we randomly pick an attribute $s_i \in S$ and generate a random subspace slice C_i. The respective samples are passed to the $deviation$ function, i.e., a function that performs the statistical test. We calculate the final result of the subspace contrast by averaging the deviations of all M statistical tests.

5.2.5. Statistical Tests

Regarding the implementation of the $deviation(\hat{p}_A, \hat{p}_B)$ function, we have employed and examined two different statistical tests.

The first approach uses Welch's t-test, which is a variation of a Student's t-test. The idea of this solution is to first extract estimations of statistical moments from both samples, and then perform a comparison based on these characteristics. The difference between Welch's t-test over the classical Student's t-test is that it utilizes more statistical moments: While the test statistic for Student's t-test only requires the sample means, Welch's t-test also uses information from the estimated sample variances. The test variable is defined as:

$$t = \frac{\hat{\mu}_{s_i} - \hat{\mu}'_{s_i}}{\sqrt{\frac{\hat{\sigma}^2_{s_i}}{N} + \frac{\hat{\sigma}'^2_{s_i}}{N'}}} \tag{5.9}$$

Intuitively, the test variable t will have small absolute values if both samples are taken from the same distribution, i.e., the sample moments are similar. Strong discrepancies between both samples will result in large values for $|t|$. In principle, we could use this test statistic directly as measurement for our deviation, but it has turned out to be preferable to convert the t value into a probability p_t as a means of normalization. This can be achieved by considering the distribution of the t values for a fulfilled null hypothesis. If the null hypothesis is true, i.e., if both samples originate from the same probability density, the test statistic t follows a t-distribution with a degree of freedom which can be obtained by the Welch-Satterthwaite equation [Sat46]. Based on the t-distribution, we can calculate the probability p_t by integration of the t-distribution.

Thus, the detailed steps to calculate the value of the $deviation$ function are:

- First, determine the required statistical moments for both samples: $\hat{\mu}_A, \hat{\sigma}^2_A, \hat{\mu}_B, \hat{\sigma}^2_B$.

- Calculate the test statistic t using Equation 5.9.
- Determine the degree of freedom of the underlying t-distribution $f_t(x)$. The problem of finding the degree of freedom is solved by the Welch-Satterthwaite equation.
- Calculate p_t by evaluating the area of the two-tail integral over $f_t(x)$ for $|x| > t$. This means that p_t is the probability to observe a larger absolute value than $|t|$ by chance if the null hypothesis is fulfilled.
- Finally, we set $deviation(\hat{\mu}_A, \hat{\sigma}_A^2, \hat{\mu}_B, \hat{\sigma}_B^2) = 1 - p_t$.

The second approach uses a two-sample Kolmogorov-Smirnov test to compare the distributions [Ste70]. This test operates on the data samples themselves and does not rely on statistical moments. To calculate the deviation, we first have to build the empirical cumulated distribution functions for both samples. The empirical cumulated distribution function of a sample of x_{s_i} consisting of N objects is defined by:

$$F(x_{s_i}) = \frac{1}{N} \sum_{\vec{y} \in DB} I[y_{s_i} < x_{s_i}] \tag{5.10}$$

where $I[cond]$ is the indicator function, equal to 1 if the condition $[cond]$ is fulfilled and equal to 0 otherwise. In other words, the value of F at a certain point x_{s_i} is the percentage of objects in the sample that have a value less than x_{s_i}. After the construction of F_A and F_B for the two samples, we can calculate the deviation as:

$$deviation(\hat{p}_A, \hat{p}_B) = \sup_{x_{s_i}} |F_A(x_{s_i}) - F_B(x_{s_i})| \tag{5.11}$$

Thus, the deviation value is defined by the maximal difference of the two empirical cumulated distribution functions.

Comparing the two approaches for the statistical test, the Kolmogorov-Smirnov test features two favorable properties from a theoretical point of view. First, it uses the full information of the data samples and does not rely on the indirect calculation of statistical moments. The other problem with all types of t-tests is that the formal derivation requires Gaussian distributed samples. On the other hand, the Kolmogorov-Smirnov test does not make any assumptions on the sample distributions. Nevertheless, our evaluation in Section 5.4 shows that both approaches can achieve good results, even for datasets that differ significantly from a Gaussian distribution.

5.3. HiCS Algorithm

Our algorithm consists of three logically independent parts:

- The calculation of the subspace contrast takes a specific subspace as input, and the output is its contrast.
- The subspace framework is responsible for the generation of subspace candidates that should be evaluated. All results are collected and will be filtered and sorted in a post-processing.
- The application of an outlier ranking on the list of high contrast subspaces.

5.3.1. Contrast Calculation

Overall, we implement the contrast calculation as a Monte Carlo algorithm, operating according to the sampling formalism in 5.2.4. Algorithm 1 shows the overall structure of the algorithm. Each Monte Carlo iteration consists of these two steps: (1) generate a random subspace slice and (2) determine the respective deviation value using a statistical test. Besides the subspace S to use for the contrast calculation, the algorithm has two other input parameters:

- The number of Monte Carlo iterations M, i.e., the number of statistical tests to perform.
- The desired average size of the test statistic. In our implementation we allow to specify the size by a ratio $\alpha \in (0, 1)$ that determines the sample size dynamically in relation to the total size of the database.

As overall output, the algorithm combines all deviation results to obtain a single contrast value for the subspace.

Algorithm 1 calculation of subspace contrast

Input: S, M, α
Output: $contrast(S)$

1: **for** $i = 1 \rightarrow M$ **do**
2: Permute list of subspace attributes $s \in S$
3: Initialize boolean vector *selected_objects* for all objects: *true*
4: **for** $i = 1 \rightarrow |S| - 1$ **do**
5: Select random index block of attribute s_i with a size of $N \cdot \sqrt[|S|-1]{\alpha}$
6: Mask index block with *selected_objects*
7: **end for**
8: Compare distributions:
$deviation\left(\hat{p}_{s_i},\ \hat{p}_{s_i|selected\_objects}\right)$ for the remaining attribute with $i = |S|$.
9: **end for**
10: Combine the results of all statistical test (cf. Equation 5.8).

The generation of the random subspace slice includes to always pick a random attribute that is used for the deviation evaluation. In the pseudo-code, this is denoted as a combination of creating a random permutation of the subspace attributes (line 2) and using the last element of the permutation for the comparison (cf. line 4 and 8). We will refer to this attribute as ***reference attribute***.

Once the reference attribute is chosen for one Monte Carlo iteration, the algorithm has to generate a subspace slice w.r.t. all remaining attributes. The idea of the adaptive subspace slices is implemented as follows: Instead of defining the condition intervals $[l_i, r_i]$ directly in the domain of the underlying variables x_{s_i}, we precalculate one-dimensional index structures for all attributes of the database. This allows to perform the selection over the sorted indices (lines 5 and 6): To generate a condition interval w.r.t. s_i, we first pick a random object rank $j \in [1, N]$, which acts as a centroid of the condition. We then mask a block of objects as "selected" depending on the difference of their rank k in the current attribute s_i compared to the centroid rank j. Overall, we select the indices of the N_1 objects which have the closest rank to j in s_i. If the centroid rank j is not on the border of the distribution, this will typically result in selecting $N_1/2$ objects in both directions from the centroid. If the rank j is close to the maximum or minimum, the selection will still use the closest N_1 elements. In this case the selection will become asymmetric, since we simply select more elements in the direction where objects are available. As discussed in Section 5.2.3, the overall goal of the selection is to always produce a sample of a fixed size $N \cdot \alpha$ (under expectation value). By using the formula for the expectation value (cf. Eq. 5.7), we can express the number of objects N_1 that have to be selected in a single condition by parameter α:

$$E[N] \stackrel{!}{=} N \cdot \alpha = N \cdot \alpha_1^{|S|-1} \quad \Rightarrow \quad \alpha_1 = \sqrt[(|S|-1)]{\alpha} \quad \text{or} \tag{5.12}$$

$$\Rightarrow \quad N_1 = N \cdot \sqrt[(|S|-1)]{\alpha} \tag{5.13}$$

This selection process is repeated for every attribute in S except for the reference attribute. The total result of all selections can be obtained by a conjunctive boolean combination of the selection blocks for the individual attributes. Note that this processing scheme allows a highly efficient implementation, since generating the conditions can be fully implemented based on boolean combination of object indicies. Accessing the original data is not necessary at all for the condition generation. Thus, it is possible to compute and store all index rankings as a preprocessing step, allowing to reuse them in all Monte Carlo iterations even for the contrast calculation of different subspaces S.

Following the adaptive random selection process, Algorithm 1 compares the marginal and the conditional distributions in the reference attribute to obtain a deviation value (line 8). When all Monte Carlo iterations are performed, the algorithm combines all deviations, resulting in the final value for the subspace contrast (line 10).

In Figure 5.3, we illustrate the algorithm based on our previous example data. Each row of Figure 5.3 corresponds to a single Monte Carlo iteration. For a two-dimensional data

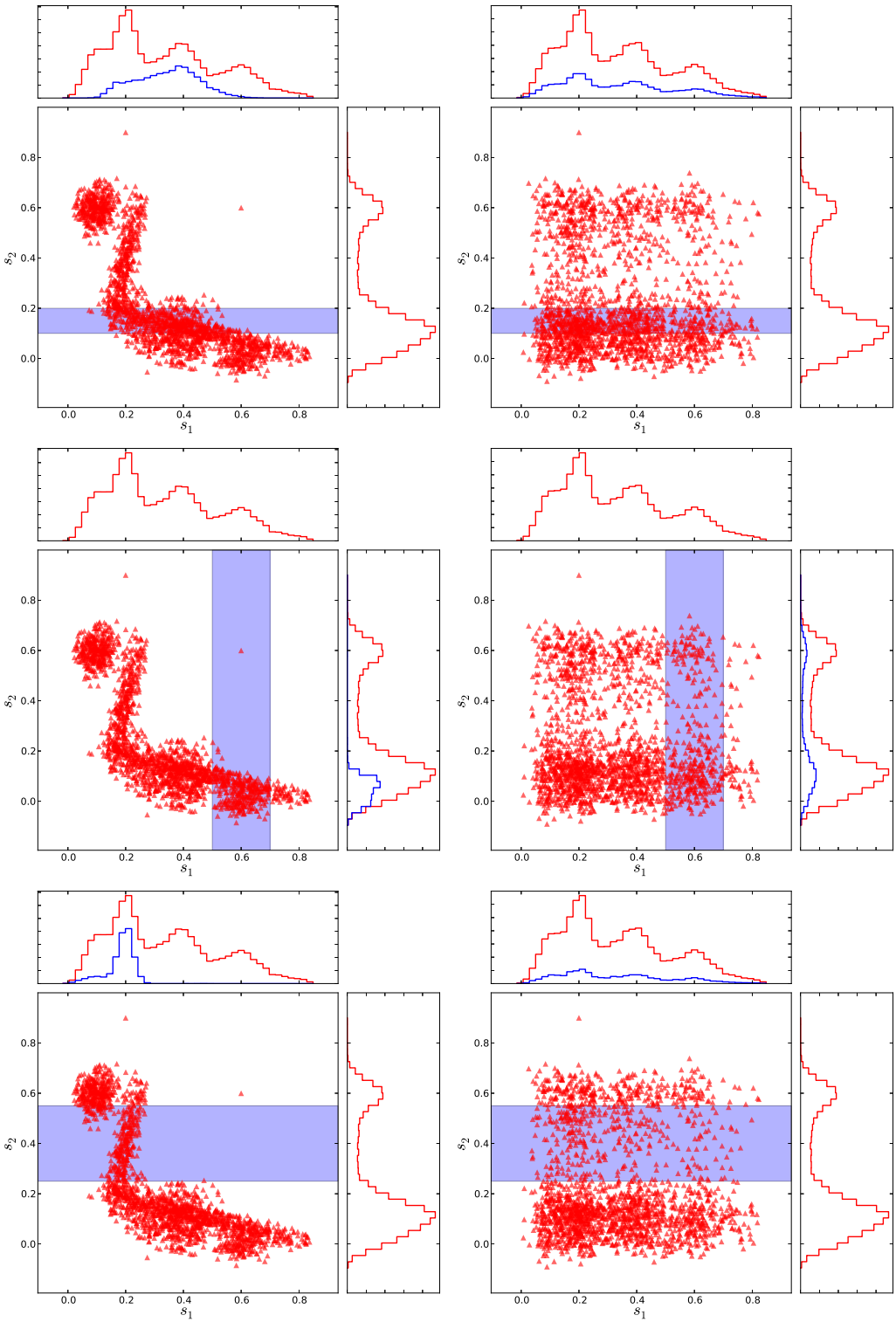


Figure 5.3.: Example results for three random slice evaluations

set, a subspace slice is one-dimensional, i.e., it only contains one condition. We can see that the algorithm picked s_1 as reference attribute in iteration 1 and 3, and s_2 in the second iteration. In each case, a centroid rank is picked from the remaining attribute. In the first iteration, the centroid lies in a region of rather high density. Both datasets have a size of $N = 2000$. Using a typical value of $\alpha = 10\%$ means that we will select 100 objects in both upper and lower direction (blue slice). Since the density is high, the resulting slice is rather narrow in the first iteration. In iteration 3 we can see how the algorithm adapts the width of the slice, in case a centroid lies in a region of lower density. In the margins of each plot, we can see the comparison of the marginal distribution (red) and the conditional distribution (blue) which corresponds to the sample of the subspace slice. In general, there is a significant deviation between the distributions for the left dataset, while they agree very well for the data on the right. In such an obvious case, performing only these three Monte Carlo iterations would suffice to clearly distinguish between the high and the low contrast subspace.

5.3.2. Subspace Framework

The subspace generation for HiCS works as follows: In each step we evaluate the contrast of the current d-dimensional subspaces. The subspaces that have a contrast above a certain threshold will be used for the generation of $(d+1)$-dimensional subspace candidates. This step-wise generation of higher dimensional subspace candidates resembles the principle of the well-known Apriori algorithm [AS94], and is illustrated in Figure 5.4.

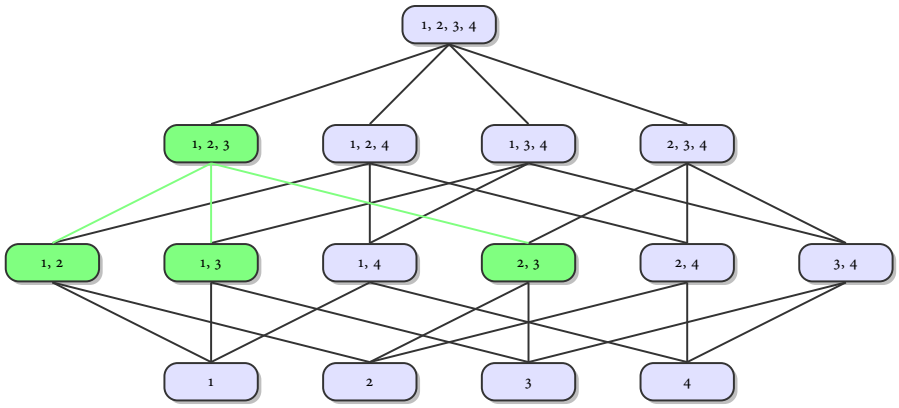


Figure 5.4.: Illustration of the Apriori principle

In contrast to Apriori, the HiCS starts with two-dimensional instead of one-dimensional subspaces, since the definition of a one-dimensional subspace contrast would not be reasonable (no notion of correlation). Another difference to Apriori is that it is not possible to formally derive a monotonicity criterion for the correlation of subspaces. To see this, we can construct a simple counterexample, such as the dataset shown in Figure 5.5. Each box corresponds to a cluster and all four clusters have the same density. In this

example, the three-dimensional joint pdf is not given as the product of the three marginal distributions, i.e., the space is correlated. On the other hand, all two-dimensional subspace projections are equally distributed and, therefore, show no correlation at all. But this example also demonstrates that the construction of such a case requires an extremely specific setup. In real world data, higher dimensional correlation is very likely to be visible in lower dimensional projections. Thus, one can combine lower dimensional subspaces to find correlations in higher dimensional spaces. Based on this heuristic, we can apply the Apriori-like subspace generation to the search of correlated subspaces.

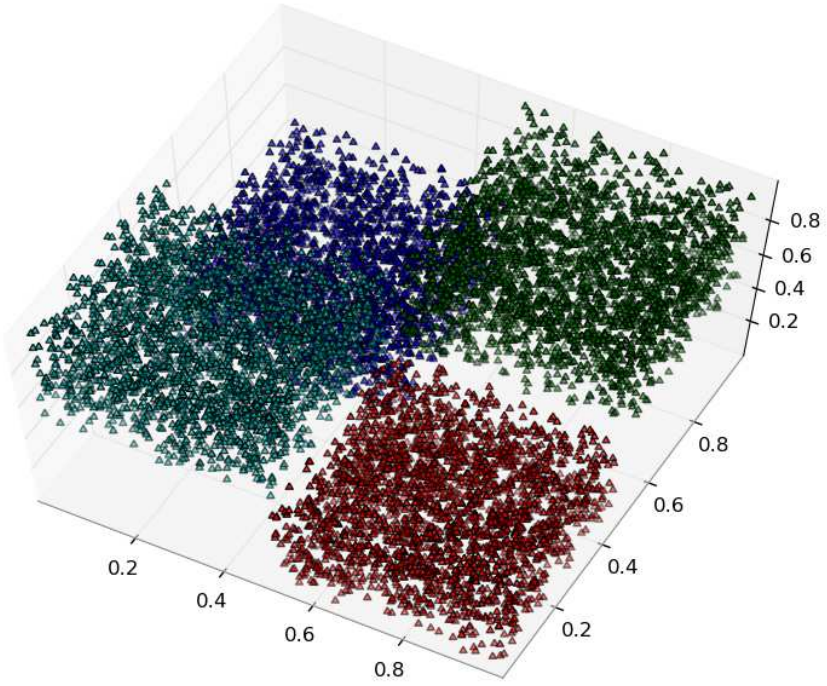


Figure 5.5.: High dimensional correlation

Like with other Apriori algorithms, the threshold for the candidate generation – in our case a lower bound on the contrast value – has a considerable impact on the results. Setting the value too high will result in a very restrictive subspace search, with only low dimensional subspaces or possibly even an empty list of subspaces. In contrast, if the value is much too low, the algorithm will consider almost all possible attribute combinations, resulting in an exponential runtime w.r.t. the total number of attributes.

Since our goal has been to design the algorithm in a way that allows a direct application to unknown datasets, we have solved this problem by means of an adaptive threshold. In contrast to conventional Apriori-like approaches, we postpone the decision whether to keep a candidate or not to the point when the contrast of all d-dimensional candidates is available. This allows to sort all current candidates and to keep only a certain number. We use the number of maximally retained candidates as parameter. Setting this

candidate_cutoff parameter allows a much more precise prediction of the runtime than specifying a reasonable minimum contrast threshold for a specific dataset.

The subspace generation process terminates when the Apriori merge step produces an empty list for the $(d+1)$-dimensional subspace candidates. In the HiCS algorithm, the subspace generation is followed by a pruning step. The idea is to remove redundant subspaces from the output to ensure that the final subspace ranking contains only important subspaces [MAG+09]. We remove a redundant d-dimensional subspace T if the subspace list contains a $(d+1)$-dimensional subspace S that has a higher contrast score than T.

5.3.3. Subspace Outlier Ranking

As final step, HiCS has to apply an external outlier ranking algorithm to the list of detected subspaces and aggregate all results. For our evaluation we use LOF as outlier score [BKNS00]. As aggregation functions we considered maximum and average. In practice taking the maximum is very sensitive to fluctuations of the outlierness and will lead to poor results, especially if the number of detected subspaces is large. Therefore we have used the average of the outlier ranking values throughout our experiments (cf. Definition 5.1). This also ensures that the outlierness is cumulative: If an object deviates in several subspaces, its total outlierness will increase compared to objects that only appear as outlier in a single subspace.

5.4. Experiments

To evaluate the quality of our HiCS approach we perform experiments on synthetic and real world datasets. We treat the problem of outlier ranking independently from the selection of high contrast subspaces. Thus, we evaluate HiCS against a series of other subspace search algorithms as pre-processing to a common outlier ranking algorithm. We focus on LOF [BKNS00] as a widely used reference algorithm for full-space outlier mining. We abstract from any enhancements by recent or future techniques [PKGF03, KShZ08, MSS10, KKSZ11], which can be used as instantiations of the outlier ranking as well. We compare HiCS against the following competitors:

- the full-space LOF outlier ranking [BKNS00]
- dimensionality reduction PCA [Jol86] + LOF [BKNS00]
- the baseline approach using random subspaces [LK05], referred to as RANDSUB
- state-of-the-art subspace search: ENCLUS [CFZ99] and RIS [KKKW03]

For all subspace methods, we adapted LOF to measure object distances only w.r.t. the given subspace, as proposed by [LK05]. To ensure comparability, we applied the same LOF outlier model with identical parameter settings (i.e., the *MinPts* value) for all competitors. We use only the best 100 subspaces from the results of all subspace search methods, to enforce a concise subspace selection.

We quantify the quality of the obtained outlier rankings by calculating the *area under curve* (AUC) of the ROC curve. To ensure comparability for runtime evaluation, we implemented all algorithms in C++ and performed all experiments on an Intel® i3-550 Processor with 4 GB RAM. In addition, we provide all datasets and parameter settings online†, to ensure repeatability of our experiments.

5.4.1. Experiments on Synthetic Data

For scalability experiments, we have generated synthetic datasets of different size and dimensionality. We randomly selected 2-5 dimensional subspaces out of the full data space and generated high density clusters in these subspaces. In each subspace we picked 5 objects and modified them to deviate from all clusters in the selected subspace. To ensure the challenge of non-trivial outlier detection, this deviation was done in a way that the object will not be visible as outlier in any lower dimensional projection. Please note that this generation allows an object to be an outlier in multiple subspaces independently. This fulfills the real world observation of outliers hidden in multiple subspace projections (cf. Section 5.1).

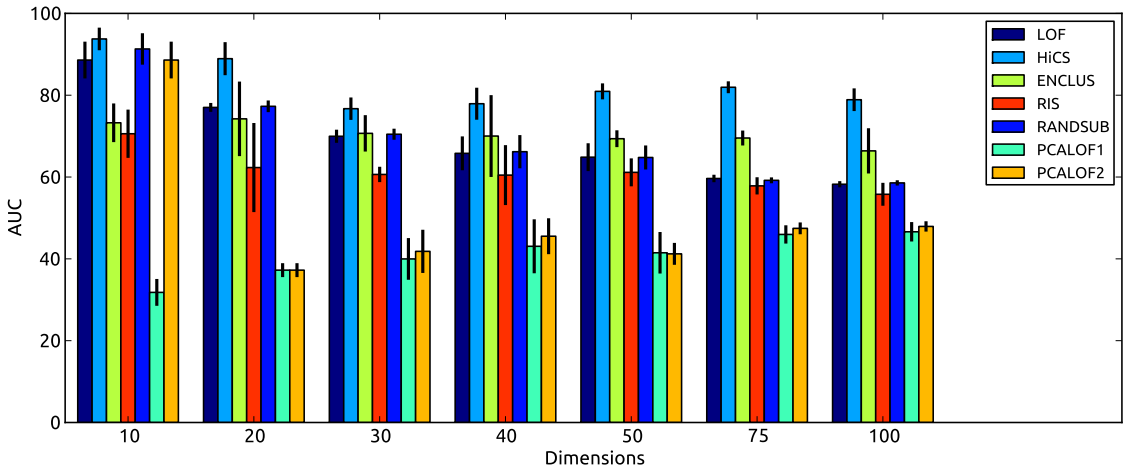


Figure 5.6.: Quality (AUC) of outlier rankings w.r.t. increasing dimensionality

† http://www.ipd.kit.edu/~muellere/HiCS/

Quality Evaluation

To evaluate the quality of HiCS we compare it with the competing algorithms in a series of experiments based on AUC. We focus on the scalability of all competitors w.r.t. the dimensionality of the data space. In Fig. 5.6 we depict the average AUC and its standard deviation for each algorithm (derived out of three randomly generated databases). HiCS outperforms the competing approaches. In particular, it scales with increasing dimensionality and shows high quality results even for high dimensional databases. Only Enclus shows similar scalability but with lower overall quality. However, a detailed examination of the subspaces selected by Enclus shows that it mainly found all two and some of the three-dimensional subspaces. This is expected because the grid based entropy measure is likely to fail for higher dimensional subspaces. In contrast, HiCS is able to detect even a high contrast in most of the five-dimensional subspaces. On the other hand, full-space runs of Lof show a degradation with increasing dimensionality, due to the curse of dimensionality. Traditional dimensionality reduction techniques such as Pca, should cope with the curse of dimensionality. However, as shown, Pca fails as preprocessing technique for outlier ranking. Please note that we have evaluated two strategies for dimensionality reduction: PcaLof1 (reduction to 50% of the total dimensionality) and PcaLof2 (constant reduction to 10 Pca-attributes). For the 10-dimensional datasets, the second strategy does not reduce the dimensionality, hence it shows the same quality as Lof. For all other cases Pca shows the worst performance (with AUC values close to 50%). This means that the resulting outlier ranking is equivalent to random guessing. We exclude Pca from further consideration, as preliminary experiments had indicated similar bad results for the following experiments as well.

Runtime Evaluation

In addition to the quality evaluation, we depict the runtime w.r.t. increasing dimensionality in Fig. 5.7. All experiment runs are identical to the previous experiment on quality evaluation, but we consider only the competitors that are based on subspace rankings. We always specify the total processing time, i.e., the time for both the subspace search and the outlier detection. Overall, the results show the scalable processing of HiCS. In particular we observe almost no increase in runtime for more than 30 dimensions. This results in a runtime comparable to the simple grid-based processing of Enclus, which is the fastest algorithm in this test but with drawbacks in terms of quality. This scalability effect of HiCS is due to our *candidate_cutoff* parameter in the subspace generation framework. It is set to 400 in this experiment. For the experiments with a dimensionality below 30, HiCS never generated more than 400 candidates. Thus, the runtime increases with more dimensions and more possible combinations of attributes. When we reach 40 dimensions, the cutoff is applied for the first time. It ensures both high quality, by maintaining the top-400 highest contrast subspaces, and low runtime, by pruning low contrast subspaces.

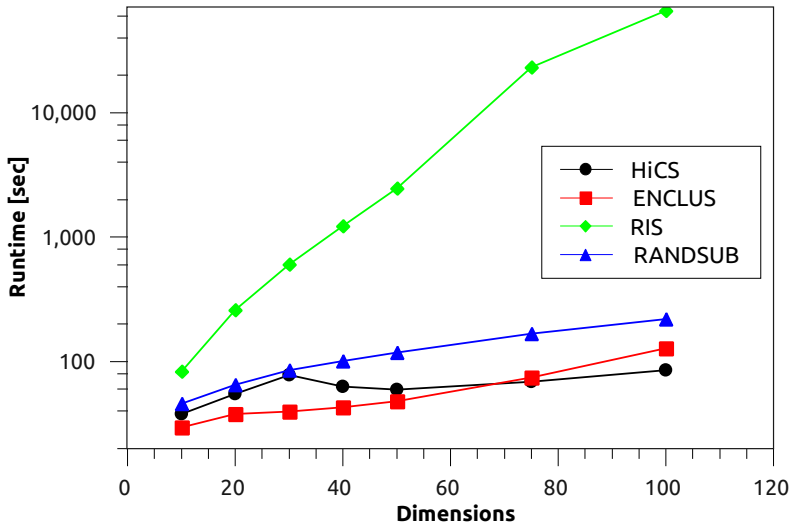


Figure 5.7.: Runtime w.r.t. dimensionality D, with fixed DB-size 1000

Besides the scalability w.r.t. data dimensionality, we have been interested in scalability w.r.t. the database size. The experimental results are shown in Fig. 5.8. The minimum runtime of all competitors is determined by the runtime of LOF and the number of selected subspaces. The latter one is fixed for all algorithms to the 100 most promising subspaces. Due to the quadratic complexity of the LOF algorithm, we expect at least a quadratic total processing time for all competitors. For RIS we observe a cubic complexity w.r.t. the database size, and accordingly this technique does not scale very well. For HICS and ENCLUS, the overhead for the subspaces detection is almost negligible if the database is sufficiently large. If we compare these two subspace search algorithms to the naive random selection, we observe that RANDSUB actually consumes more time. This is because it generates much larger subspaces on average. This seems to have a bigger impact on the runtime than the execution of a subspace search algorithm.

Parameters

In our comprehensive quality experiment (cf. Fig. 5.6), we have noticed a high sensitivity w.r.t. parametrization for our competitors. For RIS and ENCLUS in particular, we have observed that finding good parameter settings is difficult. Therefore we had run the whole experiment with a large number of configurations for these two algorithms. We have shown only the best values in the previous graphs. To evaluate the robustness of our parameter settings, we describe more detailed experiments in the following. We evaluate both variants of our statistical instantiation HICS $_{WT}$ and HICS $_{KS}$ as defined in Section 5.2.5, and we used HICS $_{WT}$ as default setting in all other experiments.

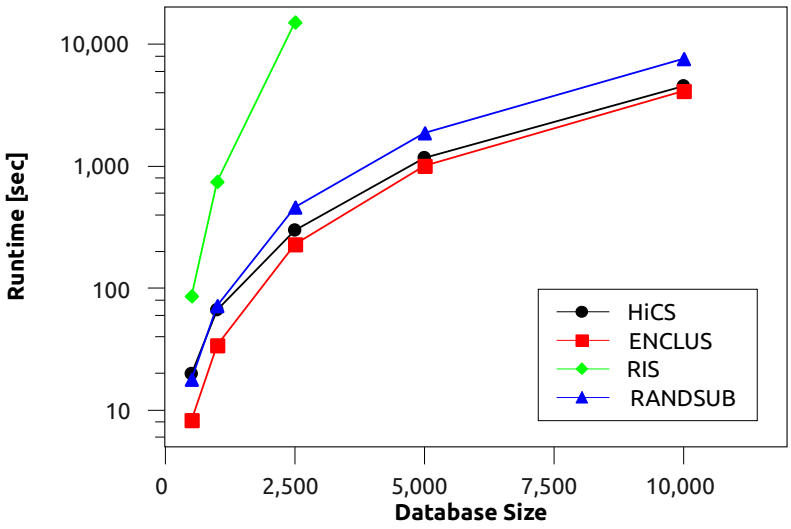


Figure 5.8.: Runtime w.r.t the DB-size, with fixed dimensionality 25

The first parameter is the number of statistical tests M that are performed for each subspace or, in other words, the number of iterations of the Monte Carlo algorithm. This trade-off between runtime and the influence of statistical fluctuations does not have a critical impact on the results. Fig. 5.9 shows the AUC quality measure contingent on the number of statistical tests. We recommend to use 50 as a default value for this parameter, as used in all other experiments.

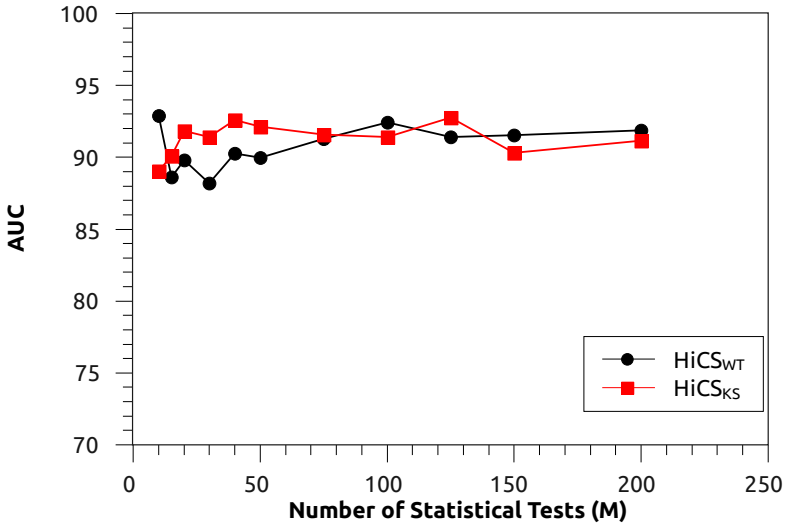


Figure 5.9.: Dependence on the number of statistical tests (M)

Furthermore, we evaluated the influence of the test statistic size α as depicted in Fig. 5.10. The experiment shows that the resulting quality is fairly robust w.r.t. the parameter α. For very low values ($\alpha < 5\%$, i.e., less than 50 objects in this experiment) we noticed a slightly

increased fluctuation of the quality. This effect becomes more important when we also reduce the number of statistical tests. Thus, having more statistical tests helps to decrease the influence of α. For larger α values, the statistical tests are less sensitive, resulting in a minor quality reduction.

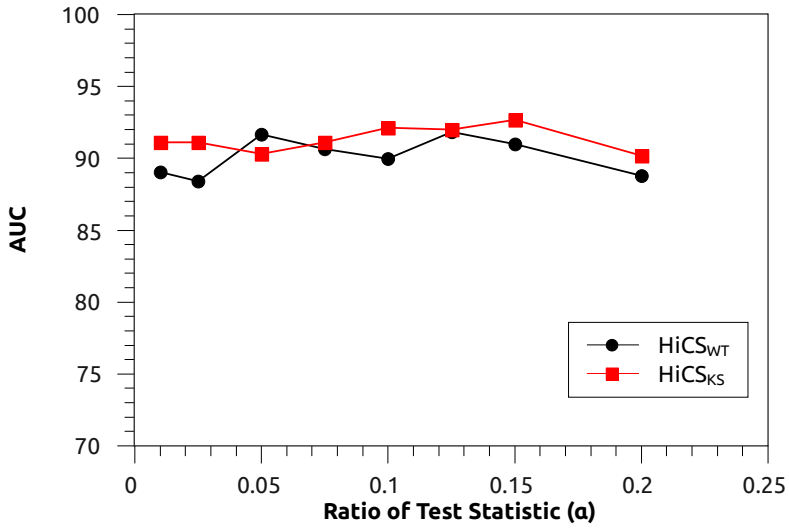


Figure 5.10.: Dependence on the size of the test statistic (α)

The last parameter *candidate_cutoff* limits the number of candidates in the bottom-up subspace processing. Thus, it influences the total runtime and the maximal dimensionality in the subspace ranking. To avoid any dataset dependence of this parameter, we have evaluated the qualities on several synthetic datasets. The following graphs always show average values. In Fig. 5.11 we can see a peak in the quality at around 500. For lower values, the quality is reduced, since the cutoff may remove some good candidates from the subspace list. The reason for this quality decrease can be found by analyzing the resulting subspace ranking: We observed that the selection starts to contain many redundant subspaces. This redundancy leads to a slight quality loss in the resulting outlier score. Please note that the fluctuations introduced by this parameter still are relatively small if we compare them to the results in Fig. 5.6. In addition to the quality evaluation we depict the influence of the cutoff parameter on the runtime in the lower part of Fig. 5.11. We see that the *candidate_cutoff* parameter allows to control the total runtime precisely. In combination with the previous quality experiments we conclude that not all candidates are required and can be pruned without a significant quality loss.

5.4.2. Experiments on Real World Data

To evaluate HiCS in a real life situation, we chose eight real world benchmark datasets from the UCI ML Repository [FA10]: Thyroid (ANN version), Arrhythmia, Breast Cancer, Breast Cancer Wisconsin (Diagnostic), Diabetes, Glass, Ionosphere and Pendigits.

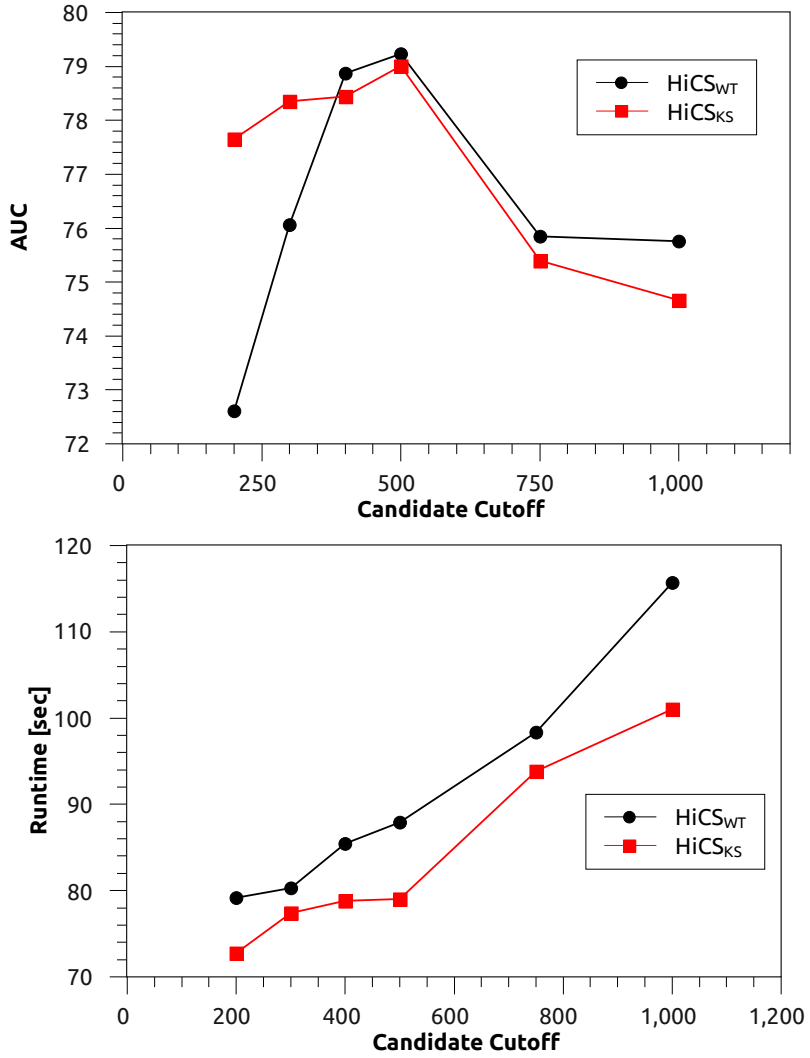


Figure 5.11.: Quality and Runtime w.r.t. candidate cutoff parameter

The main dataset characteristics are summarized in Table 5.12. Since outlier mining is conceptually similar to detecting objects that belong to a rare class, we focused on datasets where the class definitions featured a clear minority class. We assume this class to contain the outliers in these datasets. For the Pendigits dataset, all classes have equal frequencies. In this case we reduced the number of objects for one class (corresponding to the digit "0") by a factor of 10%.

Before we applied any subspace outlier algorithm, all datasets were prepared with the same preprocessing strategy: We removed nominal attributes and rescaled each attribute to the unit interval $[0, 1]$. Furthermore, we excluded attributes that show a strong discretization characteristic. The problem with such attributes is that LOF requires a non-zero k-neighbor distance for all objects, where k is equal to the utilized $MinPts$ parameter.

Experiment	*Objects*	*Dimensions*	*Outlier Ratio*
Ann-Thyroid	3772	6	2.47 %
Arrhythmia	420	129	4.29 %
Breast	198	33	23.74 %
Breast (diagnostic)	569	30	37.26 %
Diabetes	768	8	34.90 %
Glass	214	7	4.21 %
Ionosphere	351	32	35.90 %
Pendigits	6870	16	2.27 %

Table 5.12.: Real-world datasets

Therefore, we excluded attributes that have less than 10 different values or attributes in which more than 50% of all objects share the same numerical value.

	LOF	HICS	ENCLUS	RIS	RANDSUB
Ann-Thyroid	86.16	95.11	94.32	**95.16**	93.32
Arrhythmia	62.92	62.29	62.11	**63.61**	63.52
Breast	56.42	59.31	**59.55**	-	56.98
Breast (diagnostic)	86.94	**94.23**	94.19	90.77	87.07
Diabetes	70.98	**72.47**	71.15	71.63	71.70
Glass	76.86	80.05	79.73	**80.65**	78.48
Ionosphere	77.97	82.34	**82.37**	80.93	79.02
Pendigits	93.54	**95.04**	94.29	90.74	93.22

Table 5.13.: AUC results on real-world datasets

	LOF	HICS	ENCLUS	RIS	RANDSUB
Ann-Thyroid	7.1	37.2	68.1	574.0	674.0
Arrhythmia	0.5	26.4	7.9	2216.1	48.2
Breast	0.1	2.4	1.5	-	3.5
Breast (diagnostic)	0.3	15.8	11.8	14.3	28.2
Diabetes	0.3	3.3	5.9	4.0	26.2
Glass	0.0	0.2	0.3	0.1	1.7
Ionosphere	0.1	6.1	4.2	668.2	11.0
Pendigits	34.1	1194.5	2195.6	11282.7	3326.2

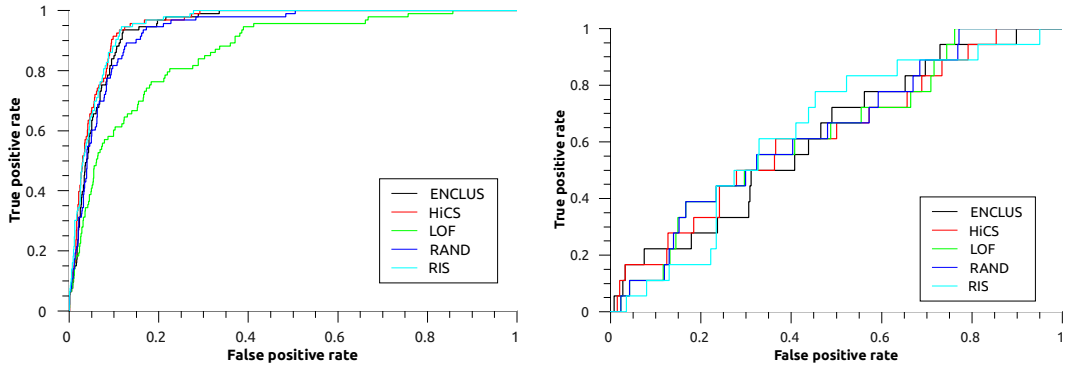
Table 5.14.: Runtime results on real-world datasets

The results of all real world experiments are shown in Fig. 5.13 and 5.14. The best AUC values are highlighted in bold, and high quality results that are within 1% of the best are not grayed out. The results demonstrate that HiCS achieves a very good overall performance. It is the best algorithm for three datasets and is close to the best result in four other experiments. Other approaches achieve high quality only for a small subset of the datasets and show a higher quality variation depending on the dataset used. HiCS is the only algorithm with high quality on most of the datasets. Considering runtime, HiCS is among the fastest subspace search algorithms. Only Enclus shows similar runtimes.

In addition, we show the individual ROC curves for all datasets in Fig. 5.15. It is interesting to note that the HiCS algorithm shows a tendency to reach the maximal true positive rate earlier than other methods. Thus, it is perfect for applications that require a high recall of outliers with best precision of the outlier ranking. On the other hand, we observe a minor weakness of HiCS if one is interested in very low false positive rates: In the Ionosphere dataset for example, the outlier ranking seems to miss some full space outliers. This results in a reduced steepness of the ROC curve for low false positive values. The reason for this might be the focus on multi-dimensional subspaces. After all, we did not remove any outliers that are trivially visible in one-dimensional projections. Therefore it might be possible to improve the quality of HiCS even further by applying a pre-processing step that takes care of the detection of trivial outliers. This would result in even higher quality, while the overall results of all AUC values show that we already obtain very good quality without such a pre-processing. Overall, HiCS shows excellent results on a broad variety of datasets, with robust and easy-to-use parameters, and a scalable processing w.r.t. the dimensionality of databases.

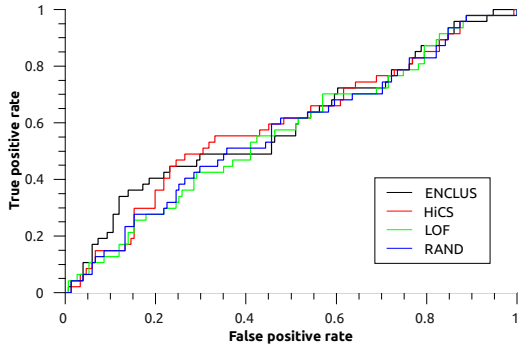
5.5. Conclusions

In this chapter, we developed an approach that is able to detect subspaces for outlier mining in high dimensional databases. We proposed the first subspace search method that selects high contrast subspaces for density-based outlier ranking. We focus on the detection of outliers that are neither visible in the full space nor in a single attribute. These non-trivial outliers show up in high contrast subspace with a strong correlation in the selected dimensions. In our two-step approach, we measure the contrast of subspaces and select the most promising ones for outlier ranking. In this decoupled processing, we propose a first contrast measure based on correlation analysis. It uses the difference between marginal and conditional pdfs of a subspace as a criterion for high contrast. The extensive set of experiments shows that our HiCS approach outperforms existing subspace search techniques, both on synthetic and on real world datasets.

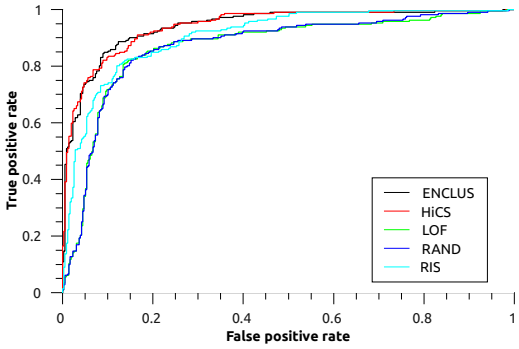


(a) *Ann-Thyroid*

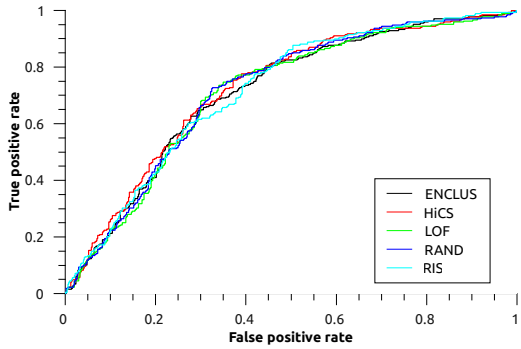
(b) *Arrhythmia*



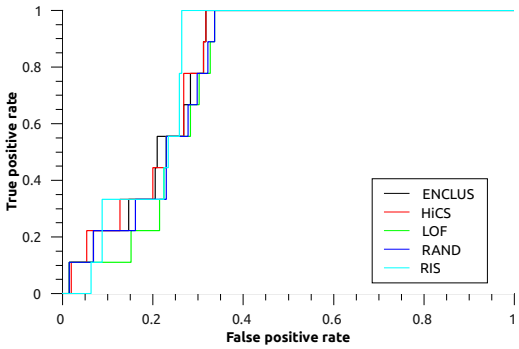
(c) *Breast*



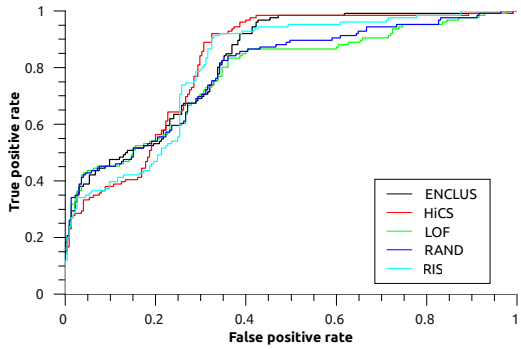
(d) *Breast (diagnostic)*



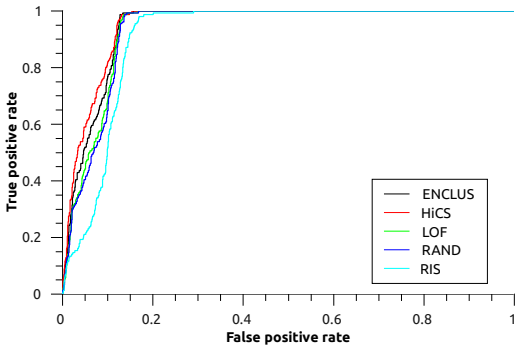
(e) *Diabetes*



(f) *Glass*



(g) *Ionosphere*



(h) *Pendigits*

Figure 5.15.: ROC plots on real world data sets

6. Subspace Contrast as a Correlation Measure

In Chapter 5, we have developed HiCS motivated by the final goal of detecting outliers. In the following chapter we want to look at the contrast measure of HiCS from an entirely different perspective – ***correlation*** **analysis***. Analyzing correlations is one of the most fundamental problems in data analysis. Given a high dimensional data set, often one of the first questions is: What exactly are the relations between all quantities of the system? In this chapter, we investigate the potential of the subspace contrast to answer this question. Therefore, we will compare HiCS against common approaches of correlation analysis. We will see that HiCS's contrast measure offers excellent properties for correlation analysis.

6.1. Bivariate Correlation Measures

For high dimensional data, the common approach in correlation analysis is to break the problem down into multiple pairwise analyses, i.e., one considers all possible variable pair combinations of a data sets [RRF+11]. For each pair, a bivariate correlation measure is used to quantify the amount of correlation. This allows to rank all variable pairs depending on the correlation, which facilitates a manual assessment of very strong or weak dependencies. We summarize the most important bivariate correlation measures in the following.

Pearson correlation coefficient. The Pearson coefficient is arguably the most popular correlation measure. However, it is only sensitive to a linear dependence between variables. The possible numeric range is $[-1, +1]$, with zero indicating neither positive nor negative correlation. A value of zero however does not imply statistical independence: In many cases the data distribution contains a superposition of positive and negative trends which cancel each other. Therefore, many non-linear dependencies are missed by linear methods. On the other hand, the Pearson coefficient allows for a very efficient computation due to its simplicity.

* Note that we again use the term correlation in its common, broader sense of "*deviation from statistical independence*", i.e., correlation refers to any kind of dependence. This must not be confused with the simpler notion of correlation corresponding to the formal definition of linear correlation coefficients.

Spearman correlation coefficient. The Spearman coefficient is a minor modification of the Pearson coefficient: Instead of operating in the original domain of the variables, it computes the dependence based on rank orders. As a result, it is not only sensitive to linear relations, but to any monotonic dependence in general. Apart from that, it shares the same basic properties of the Pearson coefficient.

Mutual Information (MI). In information theory, mutual information is a ubiquitous measure for the mutual dependence of two random variables. Intuitively, mutual information $I(X, Y)$ is equal to the reduction of uncertainty on one random variable X given knowledge of another variable Y. It is a symmetric measure, i.e., $I(X, Y) = I(Y, X)$. Most commonly, mutual information is measured in the unit *bits*, which facilitates interpretation. A high mutual information indicates a large reduction of uncertainty, i.e., the variable pair shows a strong mutual dependence. There is however no upper limit for the value of mutual information. For a perfect functional dependence – like for instance the identity function $y = x$ – mutual information is infinite, since all uncertainty is removed, i.e., one can specify y to arbitrarily many digits of accuracy by knowing x. Regarding the lowest possible value, mutual information is zero if and only if the two variables are independent. Compared to other dependence measures, like for instance Pearson or Spearman correlation, mutual information is not limited to specific kinds of dependence, e.g., linear or monotonous, but captures every possible type of dependence in the distribution function. On the other hand, the generality of mutual information leads to a significant challenge regarding estimation: Obtaining an unbiased estimate of mutual information from empiric data is a non-trivial problem. For instance, in the case of a perfect functional dependence, no estimation scheme can actually return the proper result of an infinite mutual information. Due to the limited statistics, they will instead return arbitrary finite (but large) values. In Chapter 9 we will take a more detailed look at the problem of estimating mutual information. In the following evaluation of mutual information as a correlation measure, we will use an implementation of the Kraskov estimator [KSG04], which is a general-purpose state-of-the-art estimation algorithm for mutual information [KA13].

Maximal Information Coefficient (MIC). In a recent work [RRF+11], Reshef et al. have proposed a new correlation measure called *maximal information coefficient* (MIC). It is basically a normalized variant of mutual information. Algorithmically, MIC is defined by maximizing the following expression over all possible binning schemes in the XY-plane:

$$MIC[X;Y] = \max_{|X||Y|<B} \frac{I[X;Y]}{\log_2(\min(|X|,|Y|))} \tag{6.1}$$

where $I[X;Y]$ is the grid-based estimate of mutual information, and $|X|$ and $|Y|$ refer to the number of bins in X and Y respectively. The denominator in the equation corresponds to the theoretical maximum of $I[X;Y]$ for the specific binning. Therefore, MIC is normalized in the interval $[0,1]$. A perfect functional dependence would have $MIC = 1$, since both the numerator and the denominator will have the maximal mutual information

value that is possible with a certain binning. Thus, in contrast to traditional mutual information, *MIC* does no longer have the numerical issues of mutual information in case of full dependence. However, the advantage of the normalization comes at the prize of a very high computational complexity: The formal definition requires to iterate over *all* possible two dimensional binnings, where the total number of bins $|X| \cdot |Y|$ is below a certain threshold B. The leads to an exponential complexity with respect to the total number of data points N. Therefore, Reshef et al. propose an approximate algorithm which avoids exponential complexity, but cannot guarantee to find the binning that maximizes Equation 6.1. In the experiments we present in this chapter, we use the official *C* implementation of this approximate algorithm provided by MINEPY [AFV+13].

Overall, the publication of *MIC* has lead to much controversy in the research community. One of the essential aspects in the original publication [RRF+11] is that the authors claim that *MIC* has a property which they call *equitability*. Intuitively, this property means that the *MIC* value should be the same for the same level of noise independent of the specific dependency. However, Reshef et al. do not provide a formal definition of equitability, and try to show that the property is fulfilled by simulations only. Later, Kinney et al. have provided a formalization of equitability [KA13]. Based on this formalization it is possible to prove formally that *MIC* does not satisfy equitability, or rather, that there cannot be any correlation measure satisfying the equitability condition. Furthermore, they also show that the equitability observed in the simulations is an artificial result of small sample sizes. In response to this, Murrel et al. further investigate the formal definition of equitability in [MMM14]. They find that whether or not equitability is satisfiable depends on the noise model. With the noise model in [KA13], equitability is indeed never satisfiable – however, when the noise model is not allowed to have a trend depending on $f(x)$, equitability can theoretically be fulfilled. However, *MIC* also does not satisfy this formalization of equitability. But despite the fact that this key property of *MIC* is not fulfilled formally, one can observe empirically that the *MIC* values are similar for different dependencies with the same noise level. In the following we motivate why an entirely different kind of equitability is required for our purposes.

6.2. Requirements for our Contrast Measure

Before evaluating our contrast measure in terms of its properties regarding correlation analysis, we point out essential differences to the approaches mentioned above. As a result of HICS' actual purpose – a subspace search for outlier analysis – there is key difference to traditional correlation measures: Subspace contrast is inherently a *multivariate measure*. In contrast to this, traditional measures are commonly limited to a bivarite analysis. This is because the extension of traditional correlations measures to the multivariate case is non-trivial either on a technical level or in terms of the interpretation of the multivariate measure. For instance, mutual information can be specified for the ternary

case [Kri09]. However, this definition of mutual information no longer has a minimum value of zero, i.e., mutual information can suddenly take negative values as well. There has been many attempts to come up with a meaningful interpretation of such a correlation measure. Unfortunately, independence can also no longer be detected by observing a mutual information of zero, i.e., a mutual information of zero can actually have a strong deviations from independence. While the estimation problem is also more difficult for the multivariate case, this issue is not an artifact of estimation but already arises formally. A comprehensive analysis of the interpretation of such multivariate mutual information [Kri09] comes to the conclusion that they have little practical use. Due to the lack/issues of other multivariate correlation measures, analyzing the multivariate case would not be very interesting. Instead, we focus on a comparison of the bivariate case in the following, where we can compare our contrast measure against the large number of important competitors mentioned before.

In the light of these issues of traditional correlation measures, we want to summarize the necessity of a novel contrast measure regarding our design of the HiCS framework. The requirements for a contrast measure to perform a subspace search within the HiCS framework are:

Subspace Equitability. We adopt the term equitability in the sense of a general notion of comparability. However, from the point of view of subspace search, our concern is not comparability w.r.t. similar noise levels. What is important for the evaluation of subspaces though, is an equitability w.r.t. different subspaces, especially subspaces of different dimensionality. In fact, this requirement was the primary motivation behind reducing the contrast evaluation to the one-dimensional comparison of marginal and conditional distributions. In combination with our dimensionality-adaptive slicing scheme, we obtain subspace equitability by design: The comparison mechanism is the same for every subspace of every possible dimensionality. For instance when using a Kolmogorov-Smirnov test, the contrast measure always has the same semantic – the maximal difference in the cumulative empirical distributions. Therefore, the resulting contrast measure of different subspaces are immediately comparable. From the point of view of the subspace search framework, this allows to filter out promising subspace candidates and obtain a meaningful ranking of all subspaces as final output.

Generality. We require that the measure is general, i.e., it can detect any kind of dependence and is not limited to very specific relationships like the linear correlation coefficients.

Computational Efficiency. Since the subspace search framework has to evaluate the contrast of a large number of subspaces, computational efficiency is mandatory. We will see that the complexity of our contrast measure is split into a constant pre-processing part (only has to be calculated once for a data set) and another part, which is necessary to actually compute the contrast of a specific subspace. This means that once the pre-processing is done, a batch evaluation of a large number of subspaces can be performed very efficiently.

Robust Parametrization. Regarding the parameters of the contrast calculation, our goal was to provide easy-to-use parameters which allow a robust computation of the subspace contrast.

The development of our novel contrast measure is a result of the fact that no existing correlation measure satisfies all these requirements.

6.3. Properties of the Contrast Measure

Since our contrast measure is realized as a Monte Carlo algorithm, the possibilities of a theoretical analysis are limited. Instead we will provide a thorough empirical analysis of its properties in the following sections. In general, subspace contrast is the average result of all underlying statistical tests. In this chapter, our focus is on the HiCS variant which uses the Kolmogorov-Smirnov test. In this case, the quantity obtained in a single test is the maximal difference in the cumulative empirical distributions:

$$D = \sup_x |F_{marginal}(x) - F_{conditional}(x)|$$

By definition this difference is bounded by $0 \leq D \leq 1$. Accordingly, the overall contrast measure is bounded by the range of D as well. Furthermore, we can fully calibrate the contrast measure to satisfy $contrast \simeq 0$ for independence and $contrast \simeq 1$ for a perfect functional dependence. To achieve this, we determine the expected values of D for these two extreme cases. Specifically:

- To determine D_{min}, the lowest possible expectation value of the KS-test, we draw random samples of size $N' = \alpha \cdot N$. By drawing random samples, we can exactly simulate what happens in the case of constructing condition slices w.r.t. a fully independent variable. N' is exactly the same sample size that the conditional samples obtained from the slicing will have. We repeat this random sampling with a fixed number of Monte Carlo iterations. The resulting average deviation D_{min} will reflect the deviations that one observes in a two-sample KS-test with the given distribution and sample sizes N, N' under the assumption of full independence.
- To determine D_{max}, the largest possible average deviation of the KS-test, we simulate the case of a perfect functional dependence, e.g., $y = x$. In case of this perfect correlation, all nearest neighbors in X are exactly the same in Y. Therefore, when we select a slice (i.e., a block of neighboring points) in X, we get the exact same block of points in Y. The cumulative distribution of this conditional sample will therefore be maximally steep. In the extreme case of $N'/N \to 0$, i.e., the size N' of the conditional sample is negligible w.r.t. the total sample size N, the conditional distribution will approximate a Dirac delta, since an infinitely small slice in X is as well infinitely narrow in Y. Thus, the resulting cumulative distribution will

approximate a Θ function. The average of placing the slices randomly in this case will result in $D_{max} = 0.75$, since we average over a maximal deviation, which is 1 at the left border of the distribution, 0.5 in the center of the distribution and again 1 at the right border. For a non vanishing N'/N we can simply perform the same calculation by iterating over all possible selection blocks and taking the average. This will provide the exact value of the expectation value of D_{max}, because the random selection of the index blocks itself is uniform, and therefore we can obtain the expectation value analytically.

By considering these two extremes, we can calibrate all subsequent results of the statistical tests used for the contrast calculation:

$$D' = \frac{D - D_{min}}{D_{max} - D_{min}}$$

In our experiments we will use the calibrated version of the statistical test.

Note that the above procedure is not limited to the KS-test. We only wanted to show that for some tests either D_{min} or D_{max} can even be obtained analytically. For other statistical tests where there is no analytic result for D_{max}, one could always fall back to a Monte Carlo simulation as well. Thus, the contrast measure can always be calibrated to $[0, 1]$, as long as the test statistics are bounded.

6.4. Correlation Analysis Results with HiCS

In this section we illustrate the properties of HiCS when utilized as a traditional bivariate correlation measure. We compare the properties to those of Pearson/Spearman correlation, mutual information (MI), and MIC. To this end, we generate a variety of data sets from various distributions showing different types of dependence or independence. For all data generators, we generate a total of 100 data sets, each with a sample size of 1000. We compute all correlation measures for each data sets, allowing to determine both the mean and the standard deviation of each measure for a particular distribution. Regarding the parameters of HiCS we use $\alpha = 0.05$ and 1000 Monte Carlo iterations in the following.

We begin by the most basic type of dependence: Invertible functional dependencies. Figure 6.1 shows various functions of this type, ranging from linear relationships over cubics right up to exponential/logarithmic functions. All functions are injective w.r.t. the defined domains, i.e., there is a one-to-one mapping for each x to each y value and vice versa. In summary, we observe:

- Traditional correlation measures work reasonable well on these data sets. In case of a strong non-linear characteristic, the Pearson coefficient drops to about $|0.86|$. Since Spearman correlation only requires monotonicity, it captures all cases with ± 1.

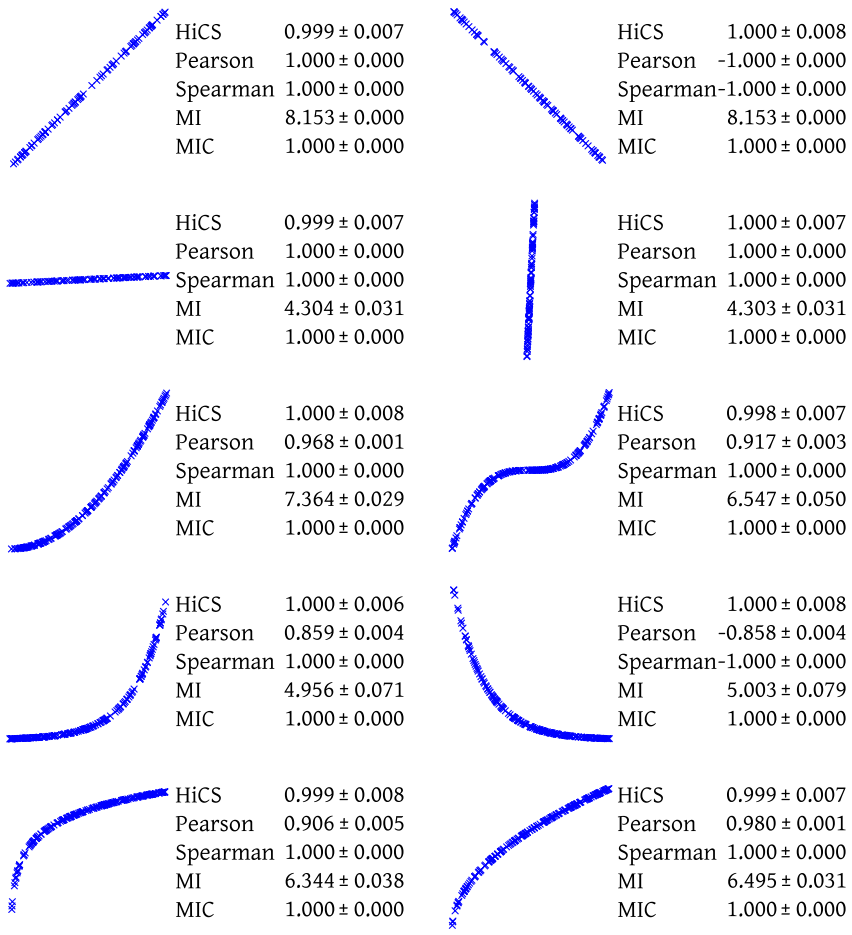


Figure 6.1.: Invertible functional dependencies; The functions $y = f(x)$ are (in default writing direction): x, $-x$, $1/20x$, $20x$, x^2, x^3, e^x, e^{-x}, $\log x$, $\sqrt{x}$. The domain of x has been restricted to ensure invertibility.

- Estimating mutual information on noiseless invertible functions is challenging: The theoretical value of mutual information is ∞ in all cases. We can see that from a sample size of 1000, the estimation result is still only a few bits. Furthermore, the estimation result depends on the slope of the functions. For instance the estimated mutual information for a linear relation with slope ± 1 is almost double the result for the linear relations with very steep/shallow slopes (to make the slopes visible in Figure 6.1 we do not have normalized the coordinate axis of the top four plots – all other plots are normalized).

- MIC is exactly 1 in all cases, since there always is a perfect binning in the examples.

- For HiCS we can see that the result is also ≈ 1 for all invertible functional dependencies. Due to its Monte Carlo nature, the result is not exactly equal to 1. However,

the fluctuations are very low, which is reflected by very low standard deviations compared to the absolute magnitude of the contrast value.

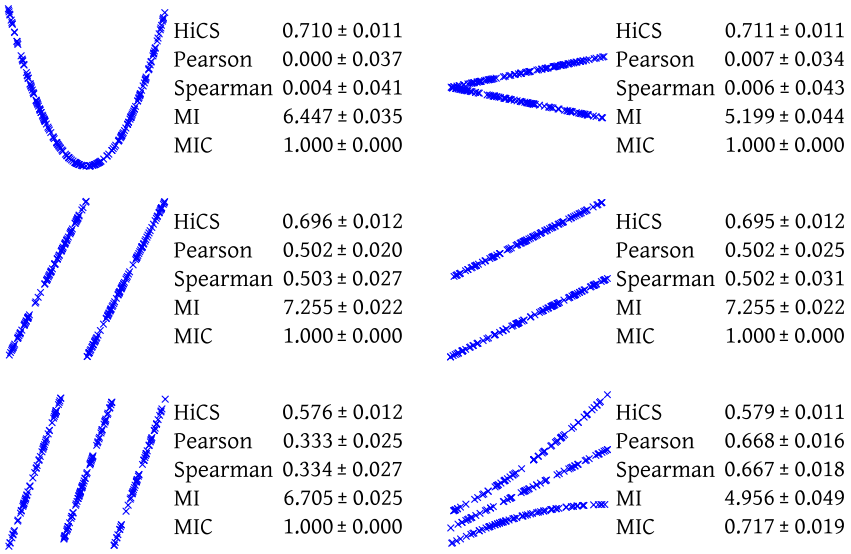


Figure 6.2.: Non-injective dependencies

In the next set of dependencies we take a look at functional dependencies which are not injective. This means that there are either x or y values, which map to multiple values in the other dimension. Such dependencies can come from either non-invertible functions (e.g., symmetric functions like a parabola), or from dependencies which are a superposition of several functional dependencies. Figure 6.2 shows examples of such cases. The observations for such relationships are:

Regarding traditional correlation coefficients, the result depends on the question whether the relation maintains a linear trend. For symmetric relations like in the first row of Figure 6.2, the trend vanishes, and thus no dependence is detected by traditional methods. This observation will apply to all the following examples, and we will therefore not repeat it explicitly.

The most fundamental difference when comparing HiCS to both MI/MIC is that HiCS can distinguish between injective and non-injective relationships: In comparison to the results for the invertible functions from Figure 6.1, we can see that both MI and MIC provide results, which are numerically identical in both cases. For MI the values are again in the area of > 4 bits. For MIC, the theoretically correct result has to be exactly 1 in all cases, since there always is a binning where all points fall into one bin, while all other bins are fully empty. In fact, the only result where the empirically determined MIC value is not exactly 1 is a result of estimation limitations, which we will see in more detail in the later examples.

In contrast to this, HiCS does distinguish between the cases of a one-to-one or a one-to-many relationship – a property which we will call *multiplicity sensitivity*. This is a result of the fact that the contrast function compares the differences in distribution in both directions, and averages over all deviations. For instance in the case of a parabola, a selection slice corresponding to e.g. large y values yields a conditional sample which has two peaks in x. Therefore, the deviation in distribution is less pronounced compared to the case of a single peak. This is in contrast to invertible functions, where a selection slice always produces a single peak in the other dimension. Numerically we can see that HiCS reflects the non-invertibility by returning a subspace contrast of about 0.7. We can also see that the value is similar if the topology of the dependence is identical: For instance, we observe similar subspace contrasts for a parabola and a superposition of two linear functions. Furthermore, we can see that the subspace contrast also reflects multiplicity, i.e, the number of possible mappings between x and y. For instance for the examples in Figure 6.2, the cases with a 3:1 multiplicity have a significantly lower contrast (≈ 0.58) compared to cases of 2:1 multiplicity (contrast ≈ 0.7). Overall, this multiplicity sensitivity is a very important property of HiCS, since other techniques are not able to capture the degree of multiplicity in functional relationships. Therefore, our subspace contrast provides a very interesting alternative to existing measures. In our opinion it makes sense for a correlation measure to incorporate this degree of determination in the correlation model, since it allows further differentiation between dependencies.

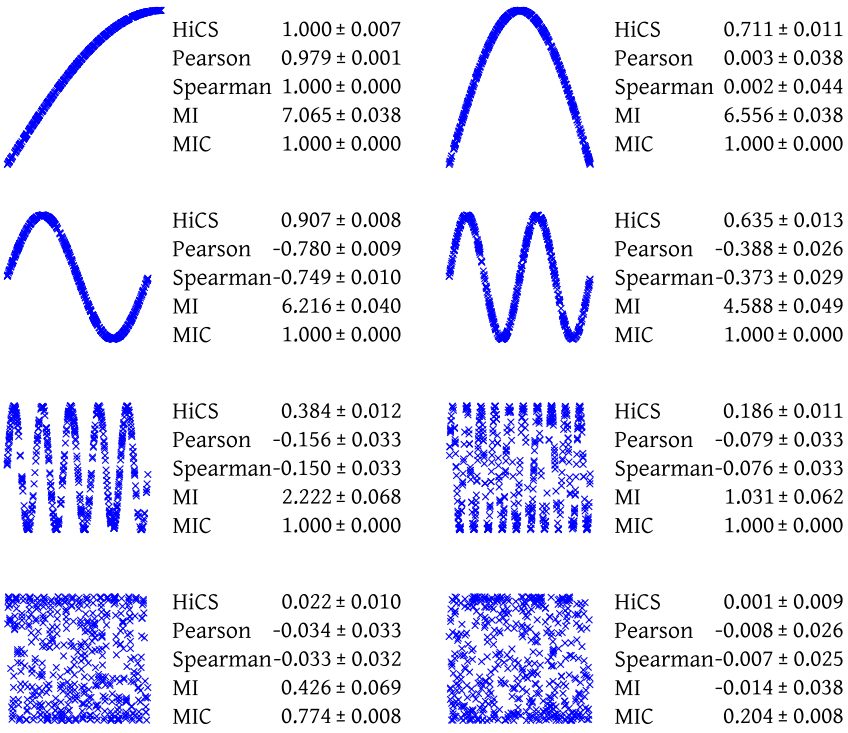


Figure 6.3.: Examples for dependencies with varying functional multiplicity

Another example where we can see multiplicity sensitivity is to consider functions with very large multiplicity. The most common case of functions with high multiplicity are periodic functions like a sine, where e.g. one y value can map to a very large number of possible x values. Figure 6.3 shows examples of sine waves of different lengths resulting in different multiplicities. If we restrict the domain to one quarter of a period, the resulting relation is invertible, and we can see that HiCS is again ≈ 1 as expected. In case of a half period, the resulting function is topologically equivalent to a parabola, and HiCS does indeed return an identical contrast value. An interesting observation is that the contrast of a full sine period is in fact larger than the contrast of the half-period. However, this is not an effect of multiplicity, since in both cases, the majority of all y values have two possible x values. Still, it makes sense to observe a higher contrast for the full period, since knowing the y value does provide more global information on x: Positive y values restrict the two possible x values to the lower half of the x domain, while negative y values are only possible for $x > x_{middle}$. Considering that the resulting relation even shows a significant linear trend, it is reasonable to assign a higher contrast to the full period. For any further increment of the number of periods, we can see the effect of multiplicity sensitivity again: For two periods the contrast drops to ≈ 0.6, for five periods it is ≈ 0.4, and ≈ 0.18 for ten periods. In the second to last example the number of periods is 20. In this case the sample size is far to low to fully represent the high frequency oscillations in the relationship. Around the center, the resulting distribution looks more like an independent uniform distribution. The only slight hint for a deviation from independence is the remaining discretization of x values at the upper and lower borders. The assigned contrast in this case is slightly above zero, at the edge of a statistically significant deviation from independence.

This example also illustrates one of the major issues of the main design principle of MIC – the requirement to obtain $MIC = 1$ for any possible noiseless functional dependence. The problem with this assertion is that the theoretical value of MIC is in fact always 1 when all x and y values are distinct, which has also be pointed out by [KA13]. The effect of this is visible in the last example of Figure 6.3, where the number of periods has been increased to 100. Given a sample of size 1000, it is impossible to properly infer the underlying relation due to lack of statistics. All correlation measures except MIC reflect this by a result corresponding to full independence. MIC on the other hand should return exactly 1 in this case as well. The reason for the actual result of ≈ 0.2 lies in the parameter B of MIC, which limits the total number of bins $|X| \cdot |Y|$ (cf. Equation 6.1). Modifying this parameter yields entirely different MIC values, from exactly 1 down to ≈ 0. Similarly, the parameter has significant effects for any data set where x and y values are distinct, because the results depend entirely on the maximally allowed complexity of the binning. Since for continuous quantities it is common to have distinct values, MIC is often affected by this parameter choice in practice. We will see in Section 6.5 that the parameters of HiCS are very robust, and do not have this issue of fluctuating from full dependence to full independence.

In the next example, we take a look at relationships described by step functions. Figure 6.4 shows a few exemplary dependencies. An interesting property of step functions is that

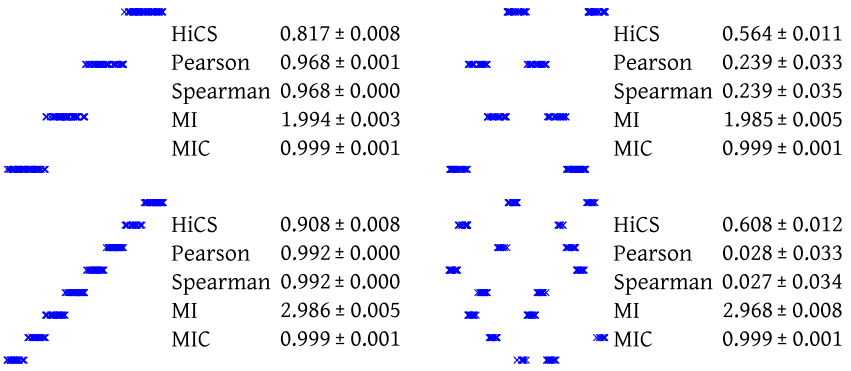


Figure 6.4.: Step functions

the interpretation of mutual information is straightforward in this case. This is because knowing the y value removes the uncertainty about x exactly corresponding to the total number of steps, while the precise position in x still remains uncertain within the step range. For instance in Figure 6.4, the total number of possible y values is 4 in the first row and 8 in the second row. Therefore, the theoretical mutual information value is 2 bits and 3 bits respectively. We can see that the MI estimation algorithm is very close to these theoretical results. Comparing the first row with the second row, the general results of MI make sense: In the second row, knowing y provides more information about x, i.e., there is a stronger dependence, and thus, MI is increased. In contrast, MIC does not detect this increase in information: Since all functions in Figure 6.4 are noiseless, MIC is always exactly 1. The subspace contrast of HiCS in turn does capture the stronger dependence of step functions with higher resolutions: The contrast increases from ≈ 0.82 for a 4-step resolution to a contrast of ≈ 0.9 for 8 steps. By increasing the resolution further the contrast would converge to 1 in the limit of an infinite step resolution, as we can conclude from the linear case.

A comparison of the columns in Figure 6.4 again reveals HiCS's multiplicity sensitivity: MI and MIC both allow the individual steps to be split into arbitrarily complex patterns which are resolvable by their estimation principles. In contrast to this, HiCS can detect that for instance the lowest y value has a simpler connection to x in the left column compared to the figures in the right column. We can see that multiplicity sensitivity also makes sense in these examples, since the functional dependencies of the examples on the right are obviously more complex. For these more complex relationships, HiCS still detects a strong dependence. At the same time, the subspace contrast reflects the difference in complexity of the patterns. A similar behavior can be found in relationships involving block-uniform dependencies as shown in Figure 6.5. In this illustrations the relationships in the top row have a theoretical MI of 1 bit; the lower examples have 2 bits. Again the multiplicity sensitivity allows HiCS to clearly distinguish between simpler (left) and more complex cases (right).

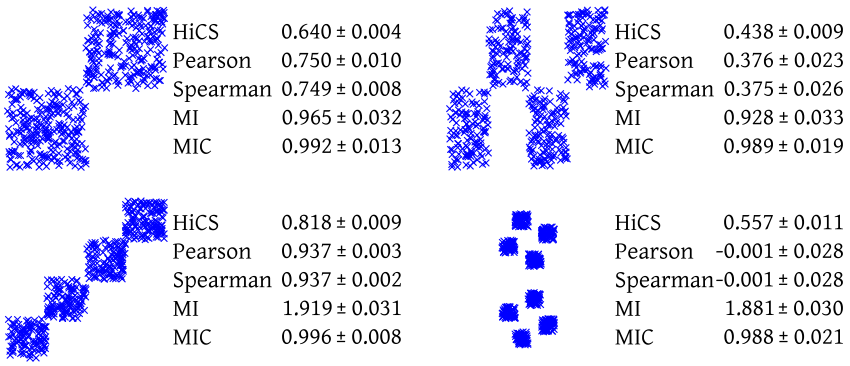


Figure 6.5.: Relationships involving block-uniform dependencies

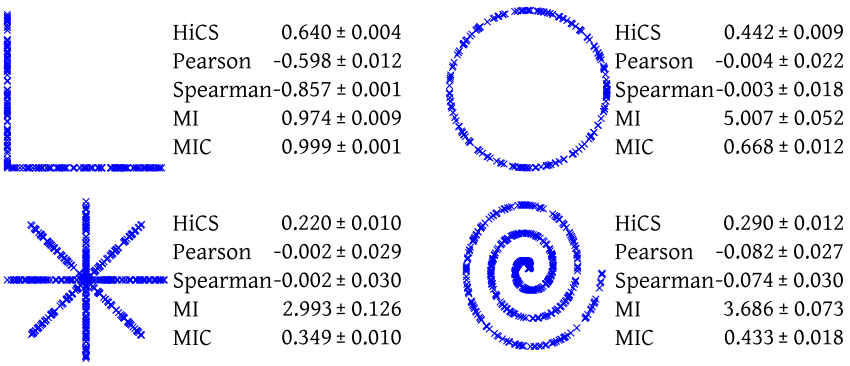


Figure 6.6.: Noiseless manifolds

Figure 6.6 and Figure 6.7 show further examples commonly used to compare correlation measures. In Figure 6.6 we focus on noiseless, one-dimensional manifolds. We can see that all non-linear correlation measures can clearly detect the dependence in these shapes. However there are differences in how the shapes are ranked: For HiCS and *MIC*, the strongest correlation is observed for the upper left example. This pattern is commonly called a non-coexistence relationship, referring to the fact that a non-zero value can only occur in either x or y but never in both (which is a fairly common pattern in real-world data, e.g., also found in [RRF+11]). Both HiCS and *MIC* observe the least correlation for the star manifold. It is interesting to see how similar HiCS and *MIC* behave in these examples despite their different nature. *MI* on the other hand shows an almost opposite ranking in these examples.

In Figure 6.7 we show examples of noisy distributions. The first two rows contains examples from previous relations now with added noise. The noise model used is as follows: For both x and y, we add Gaussian noise with a standard deviation corresponding to 5% of the standard deviations in the original distribution. We can see that such noise levels lead to reductions of the contrast by an intuitively appropriate degree. In the last two rows we also have included Gaussian distributions with different covariances, and a

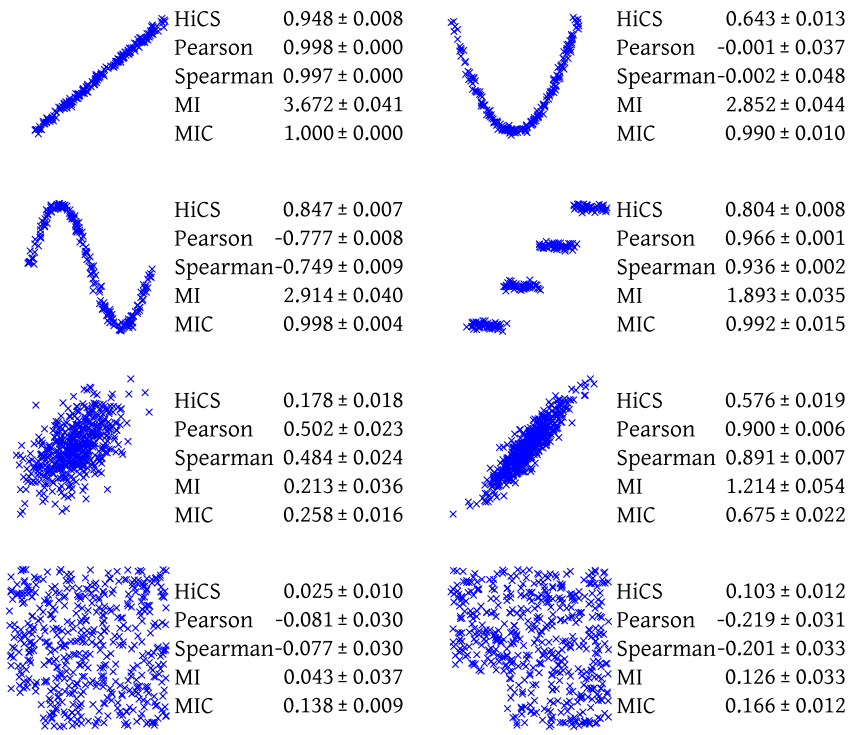


Figure 6.7.: Examples of noisy dependencies

distribution which is almost uniform – it deviates from full independence since a small area at the edge has been spared out. Overall, we can see that HiCS and *MIC* again show very similar behavior in these cases.

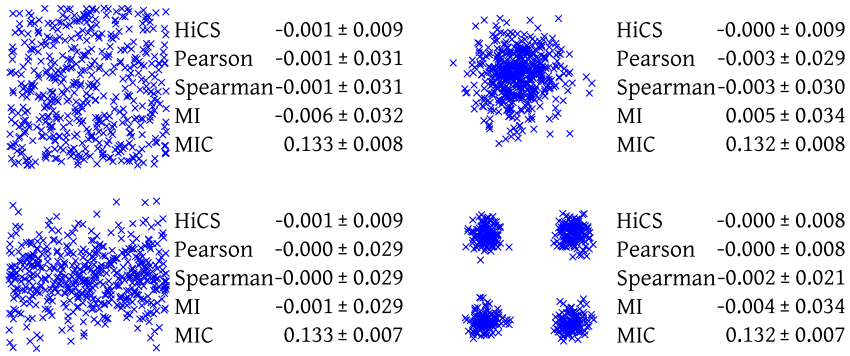


Figure 6.8.: Fully independent variables; in default writing direction: Both variables uniform, both variables Gaussians, uniform in x and Gaussian in y, 4 Gaussian clusters.

In the last experiment, we take a look at the case of fully independent variables. Figure 6.8 shows examples of distributions which are entirely independent. We can see that HiCS shows a very good zero calibration, i.e., full independence can be clearly detected by a

contrast value of 0. This is in contrast to *MIC*, which still reports an artificial dependence in these examples. This overestimation of fully independent relationships has also been visible in examples in the original publication [RRF+11]. Thus, detecting independence is more reliable with HiCS due to the possibility of a precise calibration of the subspace contrast.

6.5. Parameter Evaluation

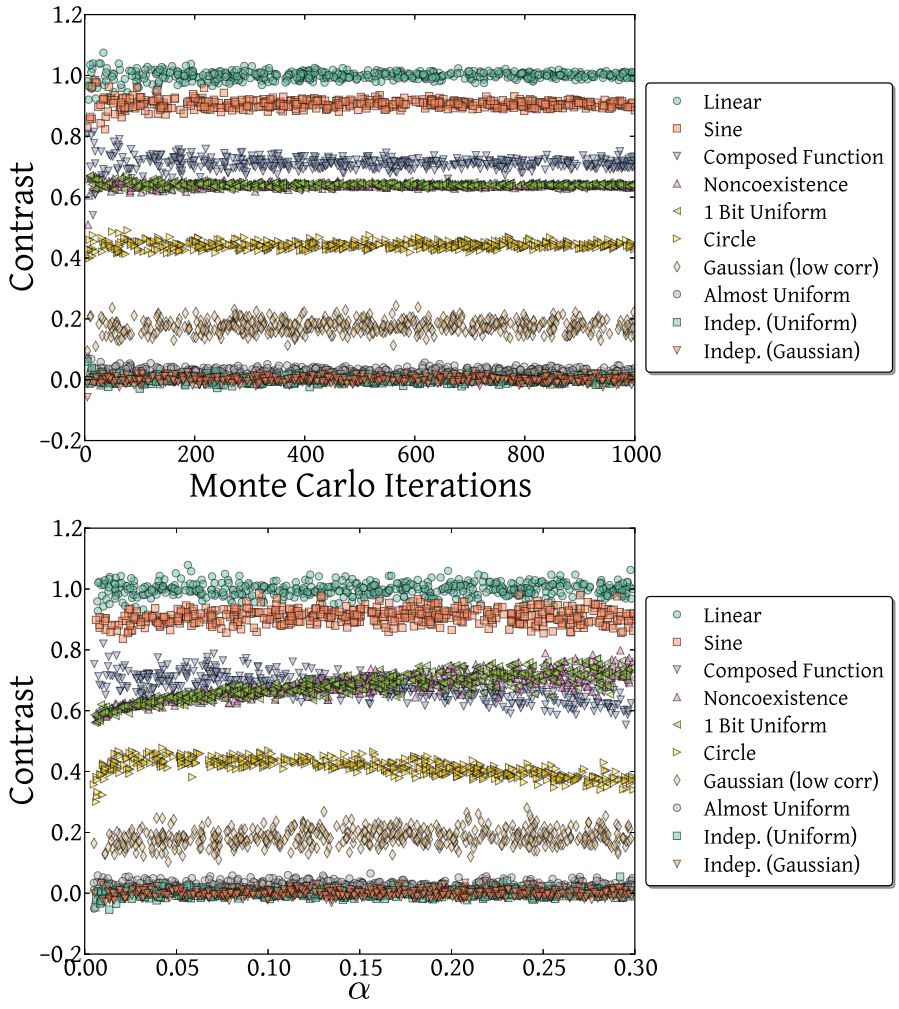


Figure 6.9.: Parameter evaluation: Monte Carlo iterations and α

In this section we want to investigate how the parameters of HiCS affect correlation evaluation. Furthermore, we will also evaluate how difficult it is to estimate subspace contrast from small samples, where measuring correlation is most challenging. To this end we propose the following experimental setup: First, we have selected 10 exemplary

data generation schemes from the previous section including relationships like linear, sines, Gaussians, independent variables etc. For each of these data generators we then perform 500 experiments consisting of:

- generate a random sample from the generator,
- pick a random value of the parameter to evaluate,
- and determine the contrast for this parameter/data set combinations.

Regarding the range of the parameters we analyze the following combinations:

- The parameter α is varied from 0.005 to 0.3 (i.e., slice widths are between 0.5% and 30% of the data size) – in this case with a fixed number of Monte Carlo iterations of 100, and a fixed sample size of 1000.
- The number of Monte Carlo iterations is varied from 5 to 1000 – with a fixed α of 0.05, and a fixed sample size of 1000.
- The sample size is varied from 20 to 500, now with fixed number of Monte Carlo iteration of 100, and fixed α of 0.05.

Figure 6.9 shows the results of this sensitivity analysis for α and the number of Monte Carlo iterations. Overall, we can see that the subspace contrast is very robust w.r.t. the parameter choice. Regarding the number of Monte Carlo iterations we can see that ≈ 50 iterations already provide very stable results for most distributions. The most challenging data distributions seem to be the composed function and the sine generator – here 100 Monte Carlo iterations are required for precise estimation.

Regarding the parameter α, we can see that most data sets do not show any dependence on this parameters. The only distributions where a slight effect is visible are the circle manifold and the 1-bit block-uniform distribution. The subspace contrast has exactly opposite dependence on α in these two cases, which is plausible: For the circle manifolds, broader slices hampers to see the fine-grained details of the noiseless manifold, and thus, increasing α reduces the visible contrast. For the block-uniform distribution on the other hand, details do not play a role in the dependence and the deviation actually becomes clearer by using broader slices. This means that the parameter α can be used to slightly shift the focus from global to local details and vice versa.

In Figure 6.10 we show the results for varying sample sizes. Overall, there is almost no bias of the subspace contrast for very low sample sizes. The most challenging distribution in terms of the sample size seems to be the circle manifold. This also makes sense since it requires a larger statistic to capture the complexity of this manifold compared to simpler structures like linear functions. Overall we can conclude that HiCS shows very robust estimation results regarding both its parameters and in case of low statistics.

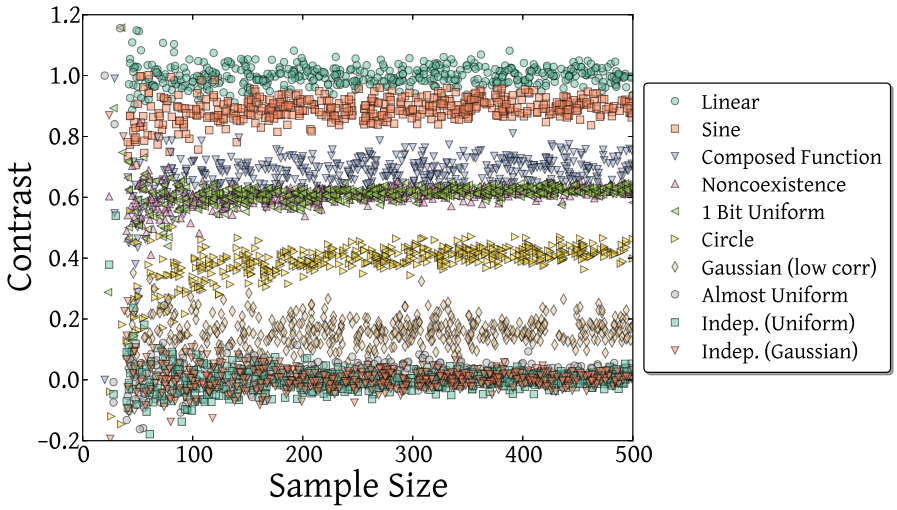


Figure 6.10.: Evaluation on small samples

6.6. Performance

We conclude our experiments by turning to the question of the scalability of the correlation measures w.r.t. the data size N. Regarding the complexity of our contrast measure, it is important to keep in mind its primary purpose – the batch evaluation of the contrast of a huge number of subspaces within HiCS. To speed up the overall subspace processing, the algorithm performs the following two steps as a preprocessing for every dimension:

- Generate the index data structure, which requires to sort the dimension. Thus, this step amounts to $O(N \log N)$ and it is the only step which does not have linear complexity.
- Perform the calibration of the minimal/maximal result of the statistical test in that dimensions. This involves a Monte Carlo simulation with the same number of iterations as used later for the actual contrast calculation.

In terms of the overall structure of HiCS this means that there is a small constant overhead for the preprocessing of all dimensions. During the actual subspace search, this precomputed information per dimension can be reused. This leads to a tremendous speed up, since most dimensions will be involved in a large number of subspace evaluations. On the other hand, this preprocessing strategy is slightly out of proportion if we use the contrast measure only to evaluate a single attribute pair. Therefore, we evaluate both runtimes for HiCS, the isolated runtime of the contrast computation alone, and the total runtime including the preprocessing of the two dimensions plus contrast computation. Figure 6.11 shows how the runtimes depend on the size of the data set. We can see that HiCS shows excellent scalability w.r.t. the sample size. Including preprocessing, it outperforms *MIC* when the sample size exceeds 1000. Note that the algorithm for *MIC*

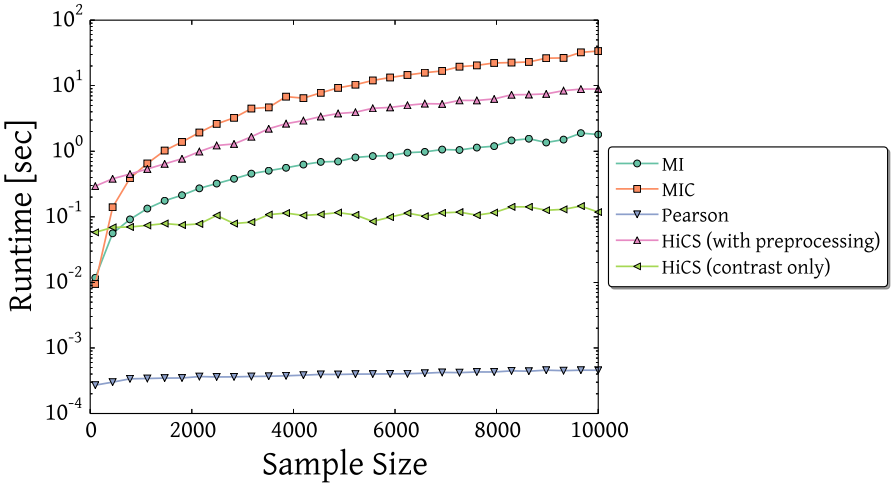


Figure 6.11.: Runtime evaluation

already is the approximate version which avoids exponential complexity. Still, MIC does not scale to large data sets. The much lower algorithmic complexity of MI and Pearson correlation results in runtimes which are orders of magnitude below MIC. An interesting result is the runtime of HiCS when only considering the contrast computation. In this case, HiCS is even much faster than estimating mutual information. In practice this is a very useful result, because in many cases the complexity of the preprocessing is negligible. For instance, a typical use case is to use a correlation measure on *all* $D(D-1)/2$ variable pairs of a D-dimensional data set. In this case, it is possible to fully benefit from HiCS preprocessing scheme: We can preprocess each dimension once, and reuse the results in all $(D-1)$ pairwise combinations. In contrast to this, all other correlation measures do not allow any speed up when dimensions are evaluated multiple times. Note that this benefit of the preprocessing scheme is especially pronounced for very high dimensional data. Thus, compared to other state-of-the-art correlation measures, HiCS has the potential to deal with both large data sets and a large number of variable pairs.

7. Knowledge Discovery: From High Contrast Subspaces to Outlier Rules*

7.1. Introduction

In general, outlier mining has two aspects: (1) ***identification*** and (2) ***description*** of outliers. So far, our main concern was the problem of identifying outliers. In this section, we now to turn to the question of describing outliers. In general, mining outlier descriptions is still an open issue in the research community, as discussed in Chapter 4.6.1. The common goal in description mining is to encode or visualize the mining results in a way that provides humans an immediate understanding of outlier reasons. Thus, these techniques assist humans in verifying the outlier characteristics. Without such outlier descriptions, humans may be overwhelmed by outlier mining results that cannot be verified manually due to large and high dimensional databases. Humans might miss outlier reasons, especially if outliers are deviating w.r.t. multiple contexts. Therefore, humans depend on appropriate descriptions. This situation enforces the development of novel outlier description algorithms and their comparison in a unified framework.

In this thesis, we propose an approach which is based on the results of mining high contrast subspace structures. In other words, our goal is to use the information of high contrast subspaces to explain why a certain object is anomalous. Since the main motivation for mining high contrast subspaces was based on their intuitive interpretation, it is now the natural step to utilize these results as a basis for the descriptions. This will also highlight the general connection of outlier mining and attribute relationships: The descriptions we generate will semantically represent relationships between attributes. The meaning of these relationships however is coupled to the results of outlier mining.

In the following, we will present OutRules†, a framework for mining outlier descriptions that enable an easy understanding of multiple outlier reasons in different contexts. We introduce the notion of *outlier rules* as a novel outlier description model. A rule illustrates the deviation of an outlier in contrast to its context in which the object shows normal

* This chapter is an extended version of *OutRules: A Framework for Outlier Descriptions in Multiple Context Spaces* published in the Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD) 2012 [MKBB12].

† Project website: http://www.ipd.kit.edu/~muellere/OutRules/

behavior. Our framework highlights the practical use of outlier rules and provides the basis for future development of outlier description models.

With OutRules we extend the outlier mining framework *SOREX* [MSG+10], which is based on the popular *WEKA* toolkit. The idea behind OutRules is to extract both regular and deviating attribute sets for each outlier and present them as so-called outlier rules. We utilize the cognitive abilities of humans by allowing a comparison of the outlier object within its regular context. This comparison enables an easy understanding of the individual outlier characteristics. For instance, in a health-care example with attributes *age*, *height*, and *weight* (cf. Fig. 7.1), a description for the marked outlier could be "the outlier deviates w.r.t. (1) *height* and *weight*, and (2) *height* and *age*". However, this first description provides the deviating attribute combinations only. In addition, we present groups of clustered objects (e.g., in attributes *weight* and *age*) as the regular contexts of the outlier. Overall, we present multiple contexts as regular neighborhoods from which the outlier is deviating. Reasoning is then enabled by manual comparison and exploration of these context spaces.

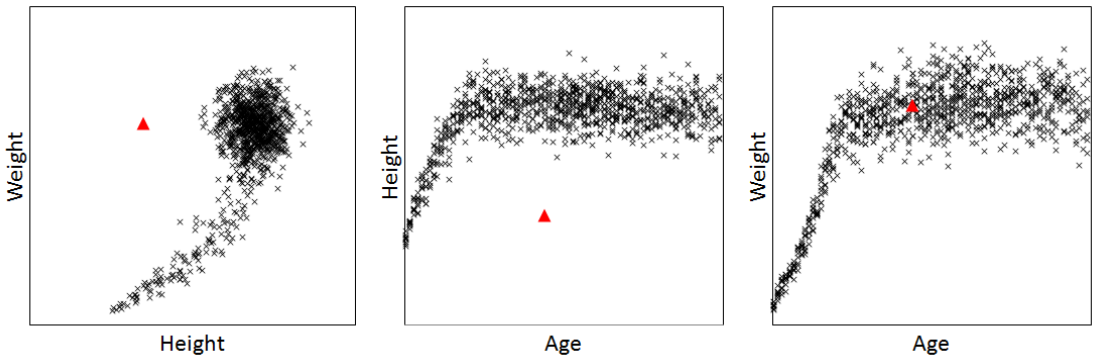


Figure 7.1.: Example of an outlier deviating w.r.t. multiple contexts

7.2. Describing Outliers by Outlier Rules

Our description model is based on the intuitive observation that each outlier deviates from other objects that are considered to be normal. Outlier rules accordingly represent these antagonistic properties of regularity on the one side and irregularity on the other side. As depicted in our example, there are multiple attribute combinations in which the object is an outlier, and there are multiple contexts in which it is regular. This multiplicity of context spaces is a general pattern of subspace-based approaches. OutRules is the first framework that exploits these multiple context spaces for outlier rules. It illustrates the similarity among clustered objects and the deviation of the individual outlier. Therefore, it provides information about multiple contexts and highlights the differences to its local neighborhoods in these context spaces.

We consider each outlier individually and compute multiple outlier rules for each object. Each outlier rule is a set of attributes that show highly clustered objects on the one side, and on the other side, an extended set of attributes in which one of these objects is highly deviating. For instance in our previous example the outlier occurs under the attributes *age* and *height*. A first rule could be "the age is normal but the person is significantly too short". In this case the description might lead to the casual explanation that the represented person suffers from impaired growth. This outlier rule can be represented as $\{age\} \Rightarrow \{height\}$. Formally, an outlier rule is defined as follows:

Definition 7.1

Outlier Rule $A \Rightarrow B$
For an object o, the rule $A \Rightarrow B$ describes the *cluster membership* of o in attribute set $A \subseteq \mathcal{A}$ and the *deviating behavior* in $A \cup B \subseteq \mathcal{A}$, where $\mathcal{A}$ is the set of all attributes.

The notion of *clustered* and *deviating* behavior can be instantiated arbitrarily, e.g., by an underlying outlier score.

Syntactically, an outlier rule is composed of a left hand side A, and a right hand side B. We call A the *context* of o in which it shows regular behavior. Note that this definition of an outlier rule does not aim at providing all attributes in which an outlier is regular. This is because for a high dimensional database an outlier often is regular in the majority of the attributes. Returning this information would not lead to deeper insight. Instead, an outlier rule alludes to the very interesting property of subspace outliers that we have discussed in Chapter 3: If an object is a strong outlier in the attribute set $A \cup B$, it is a surprise to observe a fully regular or even strongly clustered behavior in a subset A of these attribute. This indicates that the anomaly can only be explained by the combined structure that is hidden in the attribute set. In other words, we have found a striking example of a non-trivial subspace outlier.

As depicted in our example, there might be multiple reasons for an outlier deviation. Hence, our algorithm has to detect multiple contexts in which o is clustered. Since the actual reason for an outlier is highly application-dependent, it is hard to make a binary decision of relevant and irrelevant rules. Therefore we propose to output a ranking of all extracted rules. This requires to introduce a criterion, which evaluates and quantifies the quality of a rule based on the data distribution in A and $A \cup B$.

Definition 7.2

Strength of an Outlier Rule $A \Rightarrow B$
We define the *strength* of an outlier rule $A \Rightarrow B$ as an abstract quality criterion that, based on the data distribution, simultaneously quantifies both aspects:

- the degree of regularity to other objects in the left hand side A, and
- the degree of outlierness in $A \cup B$.

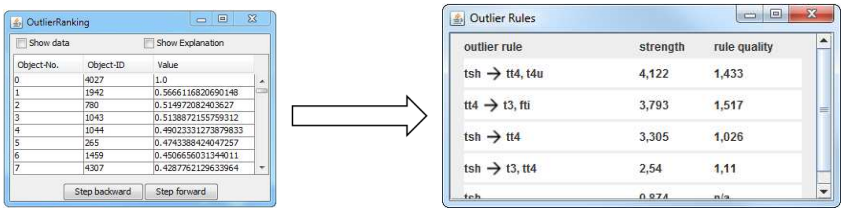
In our framework we have instantiated the strength criterion by a combination of kernel density estimation for the clustered aspect, and LOF for the deviating property. Regarding the kernel density, our approach follows the idea of a dimensionality-dependent normalization as presented in [MSS11]. We use a parameter that allows to weight the two aspects individually, defaulting to equal weight for both. In general the framework is open for any other instantiation of the strength quality criterion, e.g., for outlier rules in a specific application scenario.

OUTRULES' algorithm to mine all outlier rules is implemented as follows. As a first step, the framework will run HICS on the data set to extract subspaces that may contain subspace outliers. In the second step, we apply an outlier model (LOF) to the top ranked subspaces. The number of subspaces used is a free parameter of the framework. In general, processing more subspaces increases the number of analyzed outlier rules, thus leading to more output information, at the cost of an increased runtime. However, since the subspaces are ranked according to their contrast, it often suffices to process only a few subspaces to detect high quality rules. Finally, the combination of outliers and subspaces is utilized to generate the possible outlier rules corresponding to each subspace. This step can be performed interactively for specific objects selected by the user (cf. visualization example in Section 7.3). For each pair of object and subspace, we iterate over all possible left hand sides A given a subspace $S = A \cup B$. For each possible left hand side, we evaluate the strength of the corresponding outlier rule, by computing a kernel density estimate in A. Overall, this leads to a list of all rules for the specific outlier. We rank the list according to the strength, allowing a user to investigate the rules depending on the quality. In summary, we can see that the algorithmic aspects are straightforward, given we have solved the problem of detecting high contrast subspaces. This shows that HICS is the key component for mining outlier rules.

7.3. Outlier Rules Visualization

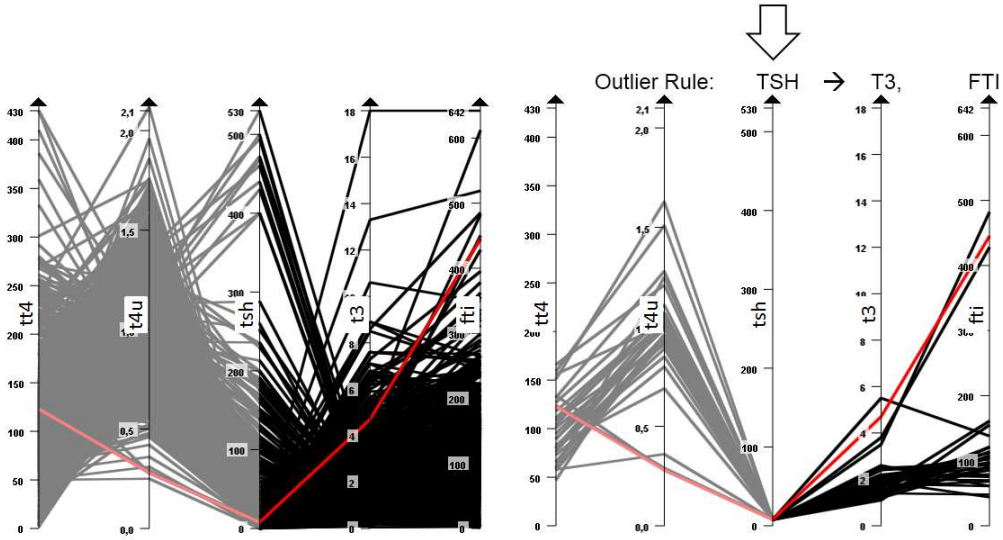
The visualization of outlier rules in OUTRULES consists of three components.

- An overview of all outliers is presented in the outlier ranking component (cf. Figure 7.2(a)). The outliers are ranked according to their average outlierness over all subspaces considered. Thus, the top ranked outliers are either strongly deviating in only one (or a low number of) subspaces or they may be ranked high as a result of a moderate deviation in many subspaces. Switching to a different aggregation



(a) *all outliers (ranked)*

(b) *all outlier rules for one outlier (ranked)*



(c) *parallel coordinates plots as an example visualization of an outlier rule; left: no context; right: neighborhood in TSH; in gray: irrelevant attributes for the outlier rule*

Figure 7.2.: One exemplary outlier from the Thyroid data set [UCI ML repository]

like the maximum or the median is obviously straightforward, but we observed that in practice the average is often a very good starting point for a manual evaluation.

- Individual outliers can be chosen from this ranking for further exploration. For each outlier, we show a list of all possible outlier rules. The outlier rules are sorted by their respective strength (cf. Figure 7.2(b)), allowing to examine the most plausible explanations first.

- The last component is the visualization of individual outlier rules; each outlier rule can be explored in more detail by looking at the underlying data distribution. For example, we have implemented scatter plots, distribution statistics, density-distributions in individual attributes, and more enhanced visual representations such as well-established parallel coordinate plots (cf. Figure 7.2(c)).

In the following we will give an example of the visualization capabilities of the OutRules framework. In the example, we use the *Thyroid (ANN version)* dataset from the UCI ML repository [FA10], which consists of various thyroid function test measurements of a

total of 3772 patients. Typically interpreting mining results of such a dataset requires a considerable domain knowledge. However, we will see that the properties of an outlier rule and the nature of an outlier become clearer by the comparison with similar objects provided by OutRules.

In Figure 7.2(c), we show the visualization of the first outlier rule (highest strength) of one of the highly ranked outliers. If we consider all objects in the database (left plot), we observe that the outlier (red line) is quite regular for all attributes from a global point of view. We now examine the underlying outlier rule, which is $\{TSH\} \Rightarrow \{T3, FTI\}$. In our parallel coordinate plots we show the attributes belonging to the rule as fully visible, while other attributes are grayed out (in Figure 7.2(c) we only show 5 out of all 21 attributes due to space constraints). To analyze the meaning of the outlier rule $\{TSH\} \Rightarrow \{T3, FTI\}$, we restrict the visualization (right plot) to the local neighborhood in attribute TSH (the patient's level of thyroid-stimulating hormone). We can see that there is a clear cluster with similar *TSH* levels containing the outlier. However, we now can also see that these patients with similar *TSH* levels typically show a very different characteristic in terms of their level of triiodothyronine (*T3*) and their free thyroxine index (*FTI*). In this context, the outlier shows a high deviation from the local neighborhood. Apparently, the low *TSH* level by itself is not exceptional. However, it is exceptional to have such a low *TSH* level in combination with a moderate *T3* and a high *FTI* level. We can see that the visualization of outlier rules greatly simplifies to find such a pattern. This is especially useful if the underlying patterns are unknown, or as in our example, are only obvious to true domain experts. But even for a domain expert the visualization capabilities of OutRules generates added value. For instance, a domain expert might be aware of the above pattern behind the rule $\{TSH\} \Rightarrow \{T3, FTI\}$. In this case, the ranking of the outlier rules according to their strength allows to swiftly skim the top ranked rules until a rule appears that may even be non-trivial from the point of view of an expert.

8. Adaptive Subspace Search*

8.1. Introduction

In this chapter, we deal with one of the most basic challenges of outlier mining in general: Depending on the application domain, the notion of anomalousness, i.e., what actually defines an anomaly, can be highly individual. To tackle this problem, many different outlier models have been proposed, as discussed in Chapter 4. Each model considers different outlier properties. For instance, some models are sensitive to distance deviations [KN98]; others capture deviation in the local density [BKNS00]; yet other models prefer angle-based [KShZ08] or statistical deviation [RL87]. Examples of three different notions are given in Figure 8.1, showing an example from energy consumption measuring, e.g., smart meter data. In these subspaces, the local density model would for instance clearly detect the green object (o_1) in the left figure, due to its high local density variation. The red object (o_2) in the middle figure would also be detected by local density- or distance-based models, while an angle-based model would not focus on this anomaly. Furthermore, depending on the parametrization the yellow object (o_3) in the left figure would either be detected as outlier or as part of a microcluster. As each outlier model is meaningful for different application domains, we do not want to discuss the pros and cons or even parametrization aspects of the different models. Instead, we focus on the orthogonal problem: How can we incorporate any possible outlier model into subspace search?

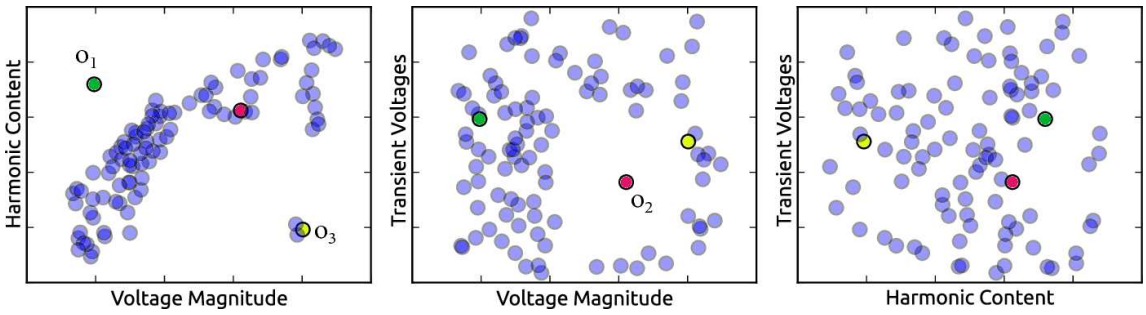


Figure 8.1.: Example of different outliers in subspaces

* This chapter is an extended version of *Flexible and Adaptive Subspace Search for Outlier Analysis* published in the Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM) 2013 [KMWB13].

To solve this problem, the approach envisioned must have the following two properties: It must be both

- *flexible* – the method allows to exchange the outlier model at all, and
- *adaptive* – the method furthermore performs the search tailored to the outlier model.

Adaptiveness is a stronger condition and implies flexibility. Figure 8.2 shows the idea behind the processing we propose in comparison to related work. Existing techniques from the field of subspace outlier mining [AY01, KKSZ09, MSS11, KKSZ12] rely on a fixed outlier model. The model cannot be exchanged depending on the application domain. Therefore, these techniques are neither flexible nor adaptive. On the other hand, our HiCS approach proposed in Chapter 5 is agnostic w.r.t. the outlier model, which only is applied as a post-processing step. Therefore, HiCS is flexible, but it is not adaptive to the model. In order to make subspace search adaptive, the search results must be tailored to the outlier characteristics of individual outliers. A subspace search scheme with this property is applicable to a broad range of application domains. Furthermore, adaptiveness allows to search for relevant subspaces individually for each outlier, and hence also improves outlier description by revealing specific outlier properties.

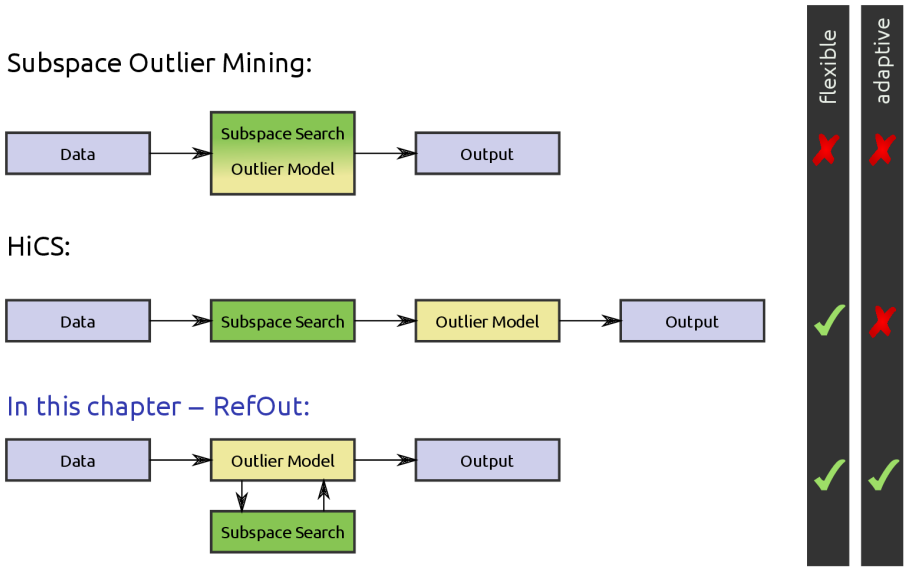


Figure 8.2.: Processing scheme in comparison to related work

As the main contribution of this chapter we propose RefOut, a flexible and adaptive subspace search framework for outlier mining. It finds relevant subspaces by a refinement process that adapts to the given outlier model. The key idea is based on the observation that traditional outlier detection methods (applied to subspaces) do capture at least small deviations of an outlier even though some irrelevant attributes are included. In the distance-based outlier model for instance, o_2 is a clear outlier in subspace S_2 =

{*Voltage Magnitude*, *Transient Voltage*}. In a high dimensional database it is hard to detect this subspace directly. But when considering random subspaces T with $|T| \geq |S_2|$, some of these random spaces will contain S_2. When applying the distance-based model to evaluate o_2 in such a space $T \supseteq S_2$ the model will report a relatively high outlier score. In contrast to this, we measure relatively low outlier scores in all other spaces $T \not\supseteq S_2$ in which o_2 shows no irregular behavior w.r.t. the distance-based model. Our main idea is to detect these *score discrepancies* of high outlier scores in $T \supseteq S_2$ over low scores in $T \not\supseteq S_2$ for individual objects. We extract information hidden in outlier scores to make a conclusion which subspace induces a high outlier score for the given outlier model. We use this information to refine a pool of random subspaces according to the discrepancies in the outlier scores. This means that if we for instance perform RefOut with an angle-based outlier model in our example, it would ignore o_2 and S_2 and instead focus on angle-based outliers and their respective subspaces.

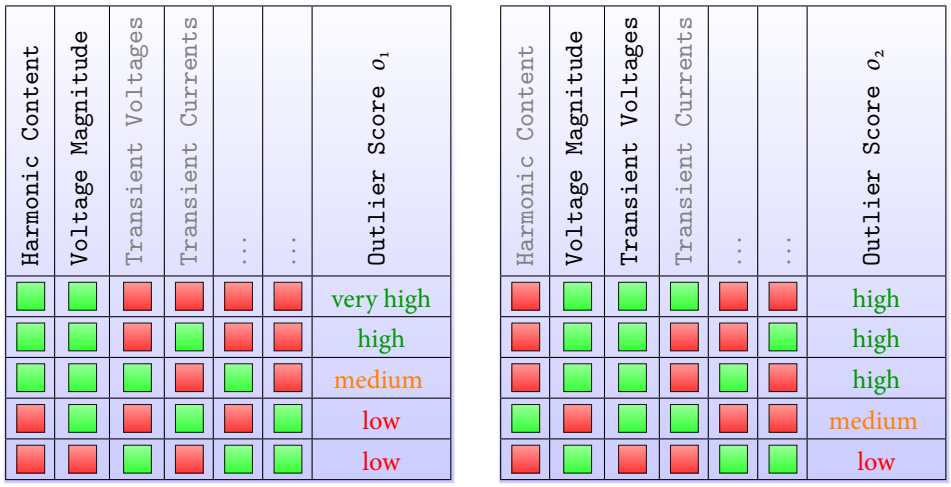


Figure 8.3.: Combinatorial problem for outliers of Figure 8.1

Figure 8.3 illustrates the idea behind the proposed algorithm. Our illustration shows the result of evaluating both o_1 (left) and o_2 (right) by a density-based outlier model in a pool of random subspaces. A binary attribute vector represents each subspace (green = attribute is part of the subspace, red = attribute not used). The subspaces are sorted by the outlier score of the object in the respective subspace. We can see that the two objects have different relevant subspaces in the top rows. Considering o_1, there is a discrepancy in the outlier scores for the different subspaces: The outlier score is high for the top three subspaces, which are supersets of S_1, and low for other spaces which are not. This discrepancy of the outlier score serves as an indication of the relevance of S_1. In this simple case, we can conclude that there is a dependency between a high outlier score and the attribute combination {*Harmonic Content, Voltage Magnitude*}. On the other hand, for the outlier o_2 we observe a dependency between the outlier score and other subspace attributes. The idea is to extract such discrepancies and the corresponding dependencies to iteratively refine the pool of subspaces. Since the approach directly operates on the

outlier scores, the refinement results can be different depending on the underlying outlier model. Therefore, RefOut fulfills both flexibility and adaptiveness.

With RefOut, we make the following contributions:

- We formalize outlier characteristics in different subspaces resulting in a so-called *score profiles* and use their properties in our adaptive search.
- We derive the *score discrepancy problem*, which provides a new theoretical perspective on subspace search.
- We propose the first subspace search approach based on the *score discrepancy problem* providing outlier descriptions for individual objects.

To the best of our knowledge RefOut is the first subspace search technique that is both flexible and adaptive w.r.t. different outlier models. In our experiments we show that this adaptivity leads to an enhanced quality for various outlier models.

8.2. Basic Notions

Let DB be a database consisting of N objects, each described by a D-dimensional real-valued data vector $\vec{x} = (x_1, \ldots, x_D)$. The set $\mathcal{A} = \{1, \ldots, D\}$ denotes the full data space of all given attributes. Any attribute subset $S = \{s_1, \ldots, s_d\} \subseteq \mathcal{A}$ will be called a d-dimensional subspace projection. For calculations in specific subspaces we constrain the vectors to the respective attributes, i.e., $\vec{x}_S = (x_{s_1}, \ldots, x_{s_d})$. This allows to deploy notions such as distance, density, and outlierness directly at the subspace level.

To define an adaptive outlier detection framework, we formalize the notion of an outlier model:

Definition 8.1

An ***outlier model*** is a function that maps every object of the database to a real-valued ***outlier score*** w.r.t. a given subspace S:

$$score(\vec{x}_S) \in \mathbb{R} \quad \forall \vec{x} \in DB$$

8.2.1. Pre-processing Outlier Scores

Since our framework evaluates individual objects in different subspaces, the only necessary requirement is that the outlier scores are comparable among different subspaces. Most outlier models do not immanently provide this comparability among subspaces. However, comparability can always be enforced by applying a normalization scheme. We assume

that the normalization ensures that the outlierness distribution of the majority of regular objects has *(1)* a mean of $default_{out}$ and *(2)* a variance of 1 independent of S. For examples of such normalization schemes for arbitrary outlier models we refer to unification techniques [KKSZ11]. For the outlier models used in this work we obtain the required properties by applying the following transformation:

$$\overline{score}_S = \frac{1}{N} \sum_{\vec{x} \in DB} score(\vec{x}_S) \tag{8.1}$$

$$Var(score_S) = \frac{1}{N-1} \sum_{\vec{x} \in DB} (score(\vec{x}_S) - \overline{score}_S)^2 \tag{8.2}$$

$$score'(\vec{x}_S) = (score(\vec{x}_S) - \overline{score}_S) / \sqrt{Var(score_S)} \tag{8.3}$$

In the remainder of this work, we apply this transformation to all outlier models utilized. For the sake of presentation, we also assume an increasing sort order of $score(\vec{x}_S)$, i.e., higher values correspond to stronger outlier characteristics. Finding alternative normalization schemes is orthogonal to our work. We focus on the selection of subspaces only and use this existing pre-processing scheme.

8.2.2. Formalization of Outliers in Subspaces

In Chapter 3, we have discussed the challenges posed by outliers hidden in subspaces. In the following, we want to formalize these observations based on the notion of *outlierness profiles*. In our formalism we focus on one individual object $\vec{x}$ and consider the outlier score properties evaluated over different subspaces by keeping one subspace S fixed for comparison. This allows to define:

Definition 8.2

The ***outlierness profile*** of an individual object $\vec{x}$ w.r.t. subspace S is a function over random subspaces T with $|T| = d'$ defined as

$$profile_{\vec{x},S}(d') = \begin{cases} E\left[score(\vec{x}_T)\right] \text{ with } T \subset S, \text{ for } d' < |S| \\ score(\vec{x}_S) \qquad\qquad\qquad\quad , \text{for } d' = |S| \\ E\left[score(\vec{x}_T)\right] \text{ with } T \supset S, \text{ for } d' > |S| \end{cases}$$

Based on this outlier profile, we are able to compare the outlier score of $\vec{x}$ in subspace S with all of its super- and sub-spaces T. Considering various spaces T with different dimensionality d' we derive the definition of a true subspace outlier as follows:

Definition 8.3

An object $\vec{x}$ is a ***true subspace outlier*** with respect to subspace S if and only if

$$profile_{\vec{x},S}(|S|) = \max_{d' \in 1 \ldots D} profile_{\vec{x},S}(d') \gg default_{out}$$

We call this maximum value the **peak** of $\vec{x}$ in subspace S and we further require:

$$\begin{aligned} profile_{\vec{x},S}(d') &\ll peak && \forall d' < |S| \\ profile_{\vec{x},S}(d') &> default_{out} && \forall d' > |S| \\ profile_{\vec{x},S}(d') &< profile_{\vec{x},S}(d'-1) && \forall d' > |S| \end{aligned}$$

We will also refer to ***true subspace outliers*** as d-***dimensional outlier*** with $d = |S|$, the dimensionality of the subspace.

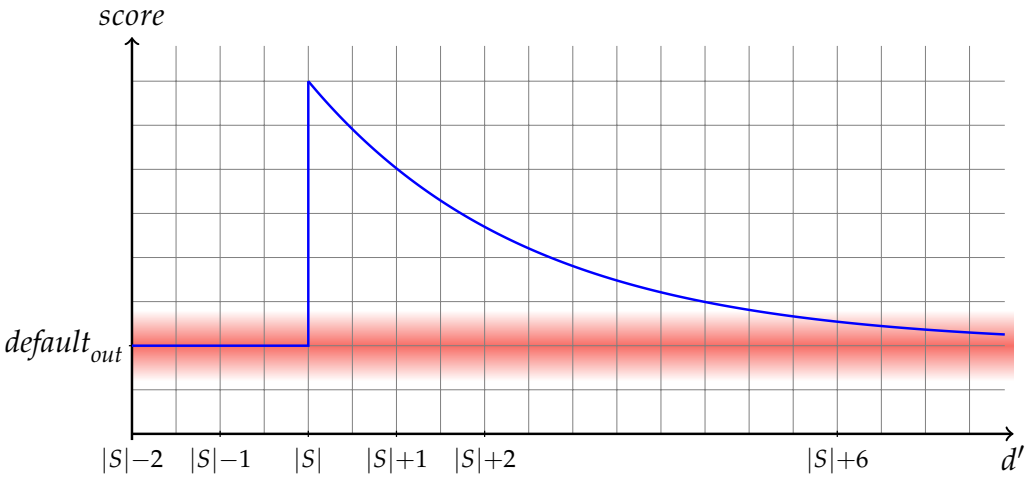


Figure 8.4.: Ideal profile of a true subspace outlier

Figure 8.4 illustrates these definitions. The plot shows an idealized outlierness profile of an individual object (blue line) that fulfills the true subspace outlier conditions. The red area shows the distribution of regular objects (unit variance as a result of normalization). At $d' = |S|$, we can see the clear outlier score peak, deviating by several standard deviations.

When considering random superspaces of S ($T \supset S$), the expectation value of the outlier score decreases monotonically. It is precisely this manifestation of the curse of dimensionality [BGRS99] that is commonly observed in reality: Adding irrelevant attributes hampers the outlier detection. Thus, the measured outlier score decreases with increasing dimensionality since all objects become more and more alike. Comparing the blue curve of an individual outlier with the red distribution of regular objects shows that at some

point the deviation of our true subspace outlier is comparable with the average deviation of regular objects. Thus, it is no longer possible to detect the true subspace outlier.

For lower dimensional subspaces $d' < |S|$ the object is projected in random subspaces of S. The defining property $profile_{\vec{x},S}(d') \ll peak$ for these spaces means that the true subspace outlier is projected into regions of regular densities in these subspace projections. This effect is also very common in reality. Think of o_1 from our example in Figure 5.1. This object clearly has a $peak \equiv profile_{\vec{o}_1,S_1}(2)$. Projecting the two dimensional subspace $S_1 = \{$*Voltage Magnitude*, *Harmonic Content*$\}$ to its one-dimensional subspaces will project o_1 into regions of high density. In none of these subspaces the object shows an exceptional outlier score, thus, $profile_{\vec{o}_1,S_1}(1) \ll peak$. By assuming that no other attribute contributes to the deviation of o_1, all properties are fulfilled and o_1 is a true subspace outlier in S_1.

Regarding higher dimensional true subspace outliers (i.e. large $|S|$), the condition

$$profile_{\vec{x},S}(d') \ll peak \; \forall d' > |S|$$

implies that the object is not exceptional in *all* lower dimensional projections. For instance, a true subspace outlier in a 4-dimensional subspace S appears to be regular in all 3-, 2-, and 1-dimensional projections of S. Only the joint consideration of all attributes makes the object exceptional, and no single attribute of S is responsible for the anomalousness alone. This property of true subspace outliers makes their detection exceptionally hard. Note that, if an object deviates in for instance two attributes s_1 and s_2, this object is not a true subspace outlier in $S = \{s_1, s_2\}$ since it suffices to clearly detect the outlier by considering the attributes separately. Thus, we would consider this object to be a true (1-dimensional) subspace outlier in both $S_1 = \{s_1\}$ and $S_2 = \{s_2\}$.

Please note that our definition of true subspace outliers is not a binary definition. For our detection framework we output the size of the peak as final outlier score for each object. Thus, we provide an *outlier ranking* with the most prominent true subspaces outliers ranked first. Also note that Definition 8.3 is a formalized version of our informal definition of a "subspace outlier" in Chapter 3 (cf. Definition 3.1). We therefore use the term "true subspace outlier" here to differentiate the formal version from our previous generic definition.

To corroborate our model of outlierness profiles and of true subspace outliers, Figure 8.5 shows examples of real outlierness profiles. For the sake of illustrating outlierness profiles we introduce profile instantiations: We draw a single line corresponding to one specific sequence of random subspaces T over the dimensionality range (each point corresponds to the outlier score in a random subset/superset of S; $T = S$ at the peak). This allows to visualize outlier score distributions and expectation values by plotting a large number of these instantiations. The first figure shows a real world outlier from the Breast dataset, detected as 4-dimensional true subspace outlier in our evaluation. The outlierness profile was generated based on the local density outlier model. As a reference we show profiles of

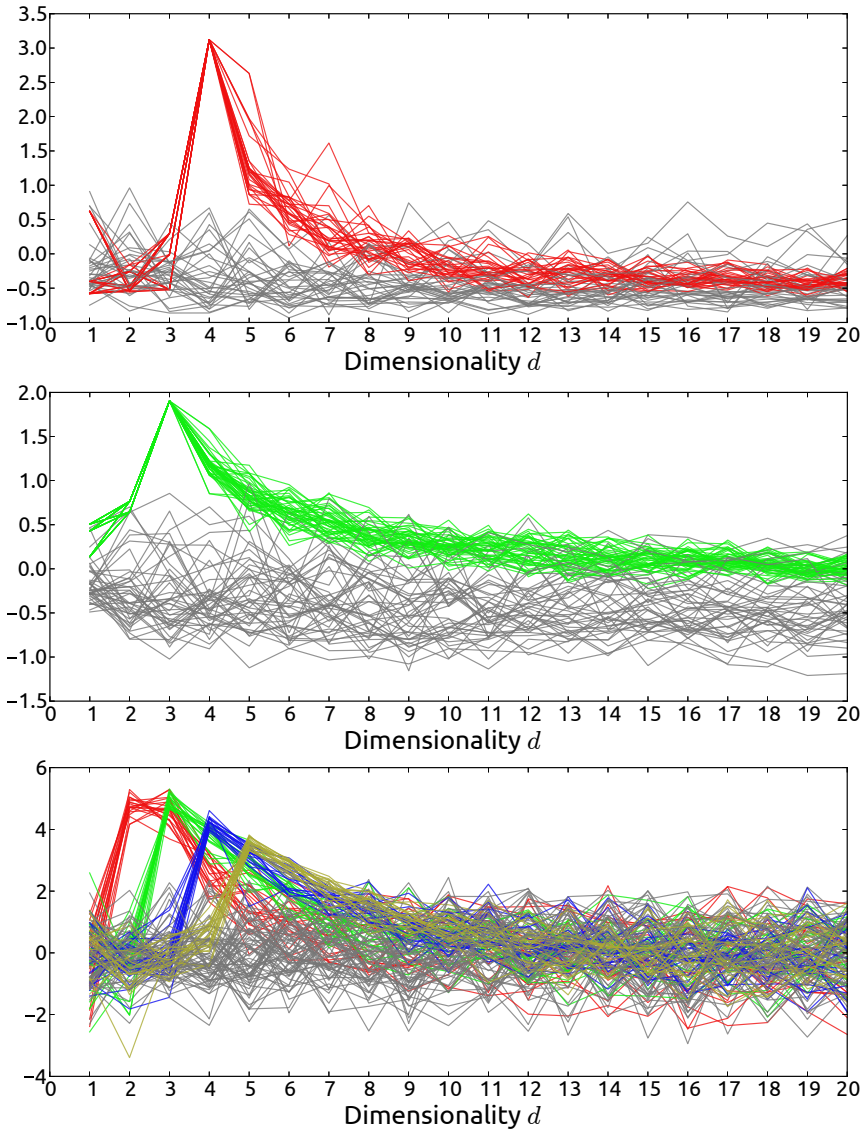


Figure 8.5.: Examples of outlierness profiles

regular objects in gray. The second figure shows the outlierness profile of a different object with a 3-dimensional peak, this time evaluated with a distance-based model. Overall, the observed outlierness profiles are in good agreement with our model. In fact, such observations of true subspace outliers on real world data were the primary motivation for the development of RefOut. We also generate our synthetic data according to these observations (cf. Sec. 8.4) and include hidden outliers of different subspace dimensionalities. The third figure shows examples from our synthetic data; this time evaluated with an angle-based model. Note that the three examples are generated based on outlier models with vast differences in their raw outlier score distributions, but the general shape of outlier profiles is preserved after normalization.

All these examples illustrate the need for subspace selection: Outliers can be clearly detected in the peaking subspace. In addition, this subspace is a valuable description of the individual outlier characteristics.

8.3. RefOut Algorithm

Our RefOut approach consists of two building blocks. The first one is the definition of a general framework for an adaptive subspace analysis based on traditional outlier scores. The underlying idea is based on the transformation of the subspace search problem into a score discrepancy analysis problem (Section 8.3.1 and Section 8.3.2). The second building block of RefOut deals with the question of how to solve this novel score discrepancy problem. We will propose our solution in Sec. 8.3.3.

8.3.1. The Score Discrepancy Problem

Identifying outlier in subspaces is computationally expensive. In principle, an exhaustive search for true subspace outliers requires scanning through all possible subspaces $2^{\mathcal{A}}$ for each object in the database. Due to the exponential number of subspaces, this would only be feasible for very low dimensional databases. To achieve a scalable subspace outlier detection it is necessary to drastically reduce the search space. To this end, we follow the idea of random subspace sampling [LK05] as a basis for our adaptive subspace search.

In order to take a new perspective on the subspace search problem, we look at the effects of applying a given outlier model in subspaces selected randomly. In the following, we focus on a single object $\vec{x}$ that is a true subspace outlier in subspace S under the outlier model. To simplify the analysis, we further assume that the object $\vec{x}$ is not a true subspace outlier in any other subspace. We denote the set of irrelevant attributes as $I = \mathcal{A} \setminus S$.

Let T be a random variable of subspaces, i.e., T is drawn uniformly from $2^{\mathcal{A}}$. We refer to the sample over these random subspaces as *subspace pool* $\mathcal{P} = \{T \mid T \text{ drawn iidfrom } 2^{\mathcal{A}}\}$. By applying the given outlier model to the random subspaces T, we obtain a sample of outlier scores:

$$\mathcal{O} = \{score(\vec{x}_T) \mid T \in \mathcal{P}\}$$

The subspace S of $\vec{x}$ plays an important role in the random sampling process: It partitions both the subspace pool $\mathcal{P}$ and the outlier scores $\mathcal{O}$ depending on whether the random subspace T is a superset of S or not. We denote the split of the subspace pool $\mathcal{P}$ as

$$\mathcal{P}_S^+ = \{T \mid T \supset S \wedge T \in \mathcal{P}\}$$

$$\mathcal{P}_S^- = \{T \mid T \not\supset S \wedge T \in \mathcal{P}\}$$

and the partition of the outlier scores $\mathcal{O}$ as:

$$\mathcal{O}_S^+ = \{score(\vec{x}_T) \mid T \supset S \wedge T \in \mathcal{P}\}$$

$$\mathcal{O}_S^- = \{score(\vec{x}_T) \mid T \not\supset S \wedge T \in \mathcal{P}\}$$

We now examine the two outlier score populations $\mathcal{O}_S^+$ and $\mathcal{O}_S^-$ by considering our observations w.r.t. the outlierness profiles. We know that for the spaces in $\mathcal{P}_S^+$, the outlier score is described by the outlierness profile (cf. Fig. 8.4), since they are supersets of the true subspace S. This means that for score $o \in \mathcal{O}_S^+$ we have $E[o] > default_{out}$, i.e., the expectation value of the score is increased over $default_{out}$. Note that this observation only applies for the expectation value of the score; in reality one can obtain an $o < default_{out}$ by chance.

For the spaces $T \in \mathcal{P}_S^-$ the true subspace S is never completely covered. We have to consider two cases when analyzing the population $\mathcal{O}_S^-$. The first case is that T partially covers S, i.e., T includes some but not all attributes of S. This means that we obtain a subspace which projects the true subspace outlier into a region of regular density. Regarding the outlierness profile, this corresponds to the left side of the peak. Thus, in this case we have $E[o] \approx default_{out}$ for $o \in \mathcal{O}_S^-$. The second case is that the random subspace T and true subspace S are completely disjunct. Thus $T \subseteq I$, i.e., T exclusively consists of attributes that are irrelevant for this true subspace outlier. In these attributes $\vec{x}$ is completely regular, thus, $E[o] \approx default_{out}$.

Combining these observations implies that we observe a discrepancy between the expectation values of the outlier score populations $\mathcal{O}_S^+$ and $\mathcal{O}_S^-$, namely:

$$E[\mathcal{O}_S^+] \; > \; E[\mathcal{O}_S^-] \tag{8.4}$$

The main idea behind our framework is to exploit this discrepancy.

Effects of random sampling: Before we reformulate the problem statement, we analyze how the random sampling of T influences this discrepancy. The general goal is to keep the total number of analyzed subspaces $|\mathcal{P}|$ low to ensure a feasible processing, i.e., $|\mathcal{P}| \ll 2^{\mathcal{A}}$. This means that in practice we have to deal with the limited size of the populations $\mathcal{O}_S^+$ and $\mathcal{O}_S^-$. It is reflected in the statistical uncertainty when comparing $\mathcal{O}_S^+$ and $\mathcal{O}_S^-$ as in Eq. 8.4. This statistical uncertainty is influenced by the dimensionality $|T|$ of the subspaces $T \in \mathcal{P}$. We have to consider the effects of both high and low dimensional T:

- *Low $|T|$:* Considering the dimensionality dependence of the outlierness profile (cf. Fig. 8.4), it is obvious that the observed outlierness difference becomes statistically more significant when the subspace T is more similar to S, i.e., when the superset T contains only a small number of additional irrelevant attributes. In Fig. 8.4, this corresponds to subspaces with a dimensionality close to the outlierness peak. This means that we can maximize the discrepancy in Eq. 8.4 by reducing the dimensionality of the subspaces in $\mathcal{P}$ to a dimensionality that is only slightly larger than $|S|$.

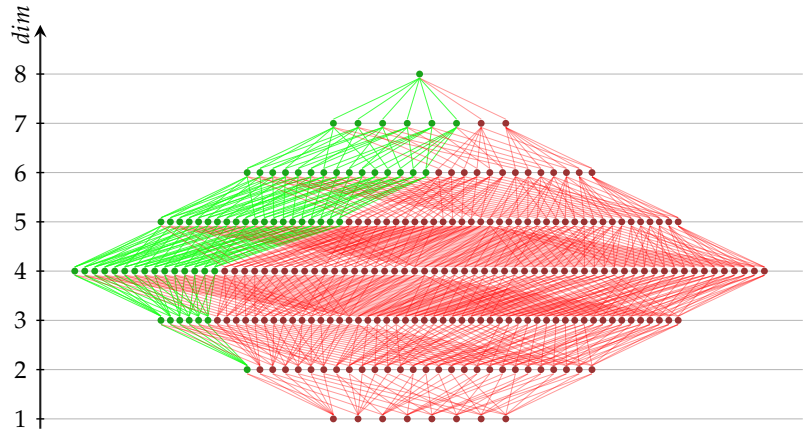


Figure 8.6.: Superspaces of a 2-dimensional subspace

• *High* $|T|$: On the other hand, we have to consider the underlying combinatorial problem: What is the probability that a random subspace T is a superset of S? Since the subspaces are drawn independently, we can use the hypergeometric distribution to quantify the probability that a space $T \in \mathcal{P}$ is a superset of subspace S. For a database consisting of D attributes, we obtain the ***coverage probability***:

$$P(T \supseteq S) = \frac{\binom{D-|S|}{|T|-|S|}}{\binom{D}{|T|}}$$

Figure 8.6 illustrates this relation. The rhombus visualizes the possible subspaces per dimensionality, and the connections represent inclusion of subspaces. As an example, the superspaces of the 2-dimensional space $S = \{1, 2\}$ are highlighted in green. The relative coverage of these subspaces clearly increases with the number of dimensions. Figure 8.7 gives an example of the coverage probability in relation to the relative dimensionality of T. Intuitively, the coverage probability increases if either $|T|$ is large (large covering subspace) or $|S|$ is small (small subspace to cover). For instance, in a database with $D = 100$ attributes and $|T| = 25$ the coverage probability is 6.06% for a two-dimensional subspace and 0.07% for a five-dimensional one. Increasing the size of the sampled subspaces to $|T| = 75$ increases these probabilities to 56.1% and 22.9% respectively. As we can see, if the subspaces in $\mathcal{P}$ are low-dimensional, it becomes more and more likely that $\mathcal{P}$ does not contain any superspaces of S. For a limited subspace pool sample $\mathcal{P}$, the superset samples $\mathcal{P}_S^+$ and $\mathcal{O}_S^+$ become very small or even empty. This means that the comparison $\mathcal{O}_S^+$ and $\mathcal{O}_S^-$ is affected by a high statistical uncertainty. Thus, we require high dimensional subspaces T to ensure that the superset populations $\mathcal{P}_S^+$ and $\mathcal{O}_S^+$ are large enough to allow a statistical inference with a high significance level.

Problem Statement: To finally transform the problem of searching for relevant subspaces into a new formulation of the problem statement, we reverse the interpretation of Eq. 8.4 in the following. So far, we have assumed a given true subspace S and analyzed

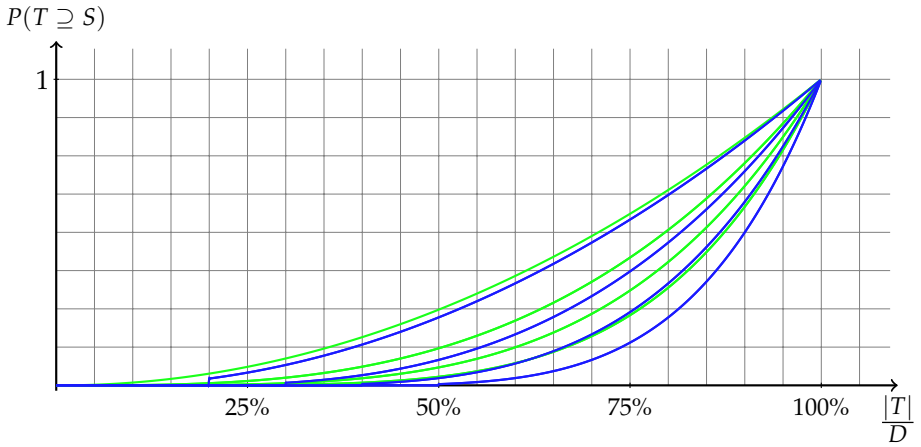


Figure 8.7.: Coverage illustration and coverage probability for $|S| \in \{2, 3, 4, 5\}$, $D = 10$ *(blue),* $D = 100$ *(green)*

its influence on $\mathcal{P}$ and $\mathcal{O}$. We now turn to the question of searching for an S' given a subspace pool $\mathcal{P}$ and outlier scores $\mathcal{O}$. We have found that for a true subspace outlier the corresponding true subspace S causes a partition of subspaces and outlier scores. For this partition we observe the discrepancy of $E[\mathcal{O}_S^+]$ and $E[\mathcal{O}_S^-]$. The reversal yields our problem statement: *Given a subspace pool* $\mathcal{P}$ *and outlier scores* $\mathcal{O}$*, which refinement* S' *causes a partitioning that maximizes the discrepancy of the outlier score populations* $\mathcal{O}_{S'}^+$ *and* $\mathcal{O}_{S'}^-$*?* For the given object, this S' is the best possible approximation of the underlying true subspace S given the limited sample size of $\mathcal{P}$ and $\mathcal{O}$. For the construction of our adaptive framework, we consider this to be a stand-alone problem and only require a subspace refinement function of the form:

$$\texttt{Refine}(\mathcal{P},\ \mathcal{O},\ d') \rightarrow S'$$

This function takes a subspace pool $\mathcal{P}$ and outlierness scores $\mathcal{O}$ of the considered object as input. The third parameter d' determines the dimensionality of the output candidate, i.e., $|S'| = d'$. The output S' is the refined subspace candidate. Formally, this refined candidate is the subspace maximizing the discrepancy, i.e.:

$$\arg\max_{S'} \left(E[\mathcal{O}_{S'}^+] - E[\mathcal{O}_{S'}^-]\right)$$

Intuitively, this S' is the best possible d'-dimensional subspace that lets the given object appear anomalous. In other words, we can use `Refine` to get the best lower dimensional attribute explanation why the considered object is an outlier for the given outlier model. The `Refine` function is the key component of our adaptive framework and is used to refine the subspaces adaptively to the outlier score of an individual object. We postpone the discussion of an instantiation of the `Refine` function to Section 8.3.3 and continue with the overview of our framework in the following.

8.3.2. Adaptive Subspace Search Framework

At a glance, the RefOut framework consists of three steps: (1) perform outlier mining on the subspaces of an *initial subspace pool* $\mathcal{P}_1$ consisting of random subspaces; (2) refine $\mathcal{P}_1$ resulting in a *refined subspace pool* $\mathcal{P}_2$ that contains subspaces tailored to the given outlier model; (3) perform outlier mining on $\mathcal{P}_2$ to obtain the final output. The first step of the framework can be considered a modified version of the random feature bagging approach proposed in [LK05]. However, our approach goes beyond this random guessing by performing an adaptive refinement in the second step.

Step 1: The objective of the first step is to collect as much information about objects and subspaces as possible. We randomly draw subspaces of dimensionality d_1 without replacement and add them to $\mathcal{P}_1$ until $|\mathcal{P}_1|$ reaches a threshold $psize$. Note that this allows RefOut to perform an exhaustive search on dimensionality level d_1 for very low dimensional databases or large $psize$, but in general $\binom{D}{d_1} \gg psize$. The dimensionality parameter d_1 controls the trade-off between a good subspace coverage probability (large d_1) or a less severe curse of dimensionality (low d_1). The framework then applies the given traditional outlier model to all subspaces $T \in \mathcal{P}_1$. To ensure the desired property of comparable outlier scores amongst different subspaces, we apply the normalization (Eqs. 8.1-8.3) to the outlierness distribution in every subspace. The framework stores these normalized outlier scores for every object in every subspace.

Step 2: The goal of the second step is to exploit the information collected in Step 1 by refining the subspaces adaptively to the outlier scores resulting in the refined subspace pool $\mathcal{P}_2$. Note that the subspace refinement operates per object, i.e., every object has an individually refined subspace. In principle it would be possible to produce a refined subspace for every object in the database, resulting in $|\mathcal{P}_2| = N$. However, if an object does not show anomalous behavior in any of the subspace projections of $\mathcal{P}_1$, it is very likely that this object simply is regular. Thus, to speed up the processing, the framework excludes these inliers for subspace refinement. Instead of processing all objects, the framework ranks all objects according to their maximum outlier score over all subspaces in $\mathcal{P}_1$. A parameter $opct$ controls the number of objects (expressed as ratio of the database size) that are considered for subspace refinement, i.e., we consider the top $\lfloor opct \cdot N \rfloor$ objects from this ranking. Since each subspace refinement adds one subspace to the refined pool, this also determines the size $|\mathcal{P}_2|$. The target dimensionality of the subspace refinement is given by parameter d_2, i.e., $|T| = d_2 \quad \forall T \in \mathcal{P}_2$.

Step 3: The third step applies the outlier model again – this time to the refined pool $\mathcal{P}_2$. As in Step 1, we normalize the outlier scores of each subspace to ensure comparability. The final outlier score of an object is the maximal normalized outlier score observed over all subspaces in $|\mathcal{P}_2|$. Algorithm 2 summarizes the steps of the RefOut framework.

To analyze the complexity of this algorithm, we look at the search space processed. A naive algorithm would check all $2^{\mathcal{A}}$ subspaces, which clearly does not scale. In contrast,

Algorithm 2 Adaptive Subspace Search

Input: DB, outlier model $score(.)$, d_1, d_2, $psize$, $opct$
Output: score and best subspace description for each object

```
1: P_1 = random subspaces of dimensionality d_1
2: Apply score(.) to all T ∈ P_1 and normalize outlier scores
3: Rank objects according to maximal outlier score
4: for x⃗ ∈ ⌊opct · N⌋ top ranked objects do
5:     Extract O for individual object x⃗
6:     S' = Refine(P_1, O, d_2)
7:     Insert S' in P_2
8: end for
9: Apply score(.) to all T ∈ P_2 and normalize outlier scores
10: Output maximum score and subspace for each object
```

we only look at a limited set of subspaces. The search space is limited by the parameters $psize$ and $opct$. Furthermore, the subspace candidate refinement requires only a small number of subspaces considered in the pool. The total number of subspaces processed is $(psize + \lfloor opct \cdot N \rfloor)$. Thus, the complexity of the framework itself is $\mathcal{O}(N)$. In terms of the underlying outlier model to check these subspaces, we depend on the complexity of the detection algorithm, which range from $\mathcal{O}(D \cdot N)$ for efficient distance-based [GPO08], $\mathcal{O}(D \cdot N^2)$ for density-based methods [BKNS00], up to $\mathcal{O}(D \cdot N^3)$ for the basic version of angle-based methods [KShZ08].

8.3.3. Instantiation of the Refinement Function

The goal of the refinement function `Refine` is to obtain the d'-dimensional subspace S' that maximizes the discrepancy of the populations $\mathcal{O}^+_{S'}$ and $\mathcal{O}^-_{S'}$. The input of `Refine` is the set of subspaces $\mathcal{P}$ and the corresponding outlier scores $\mathcal{O}$ of an individual object $\vec{x}$. To simplify the notation we treat both input sets $\mathcal{P}$ and $\mathcal{O}$ as sequences with an arbitrary but fixed order. Since there is an outlierness value for every subspace $T \in \mathcal{P}$, we define the order $\mathcal{P} \equiv (T_1, T_2, \ldots, T_M)$ and $\mathcal{O} \equiv (o_1, o_2, \ldots, o_M)$ such that $o_i = score(\vec{x}_{T_i})$. We will use the notation (T_i, o_i) to refer to a pair of subspace and corresponding outlier score.

To illustrate the problem to solve, we introduce a running example in Figure 8.8. The table shows the measured outlier scores of an outlier with true subspace $S = \{1, 2, 3, 4\}$ evaluated in random subspaces of dimensionality 9 within a database of dimensionality 12. A green box indicates that an attribute is included in the random subspace. To ease presentation, we have ordered the (T_i, o_i) tuples according to the outlier score of the object in the respective subspaces. If we partition the rows according to $T \supset S$ vs $T \not\supset S$, we obtain the rows with the ranks 1, 2, 3, 4, and 7 as population $\mathcal{P}^+_S$. Considering the corresponding outlier score populations $\mathcal{O}^+_S$ and $\mathcal{O}^-_S$ clearly shows that $\mathcal{O}^+_S$ is stochastically

greater than $\mathcal{O}_S^-$. Ideally, for any $d' \geq 4$ the goal of the `Refine` function is to detect this discrepancy and return a refined subspace $S' \supseteq S$.

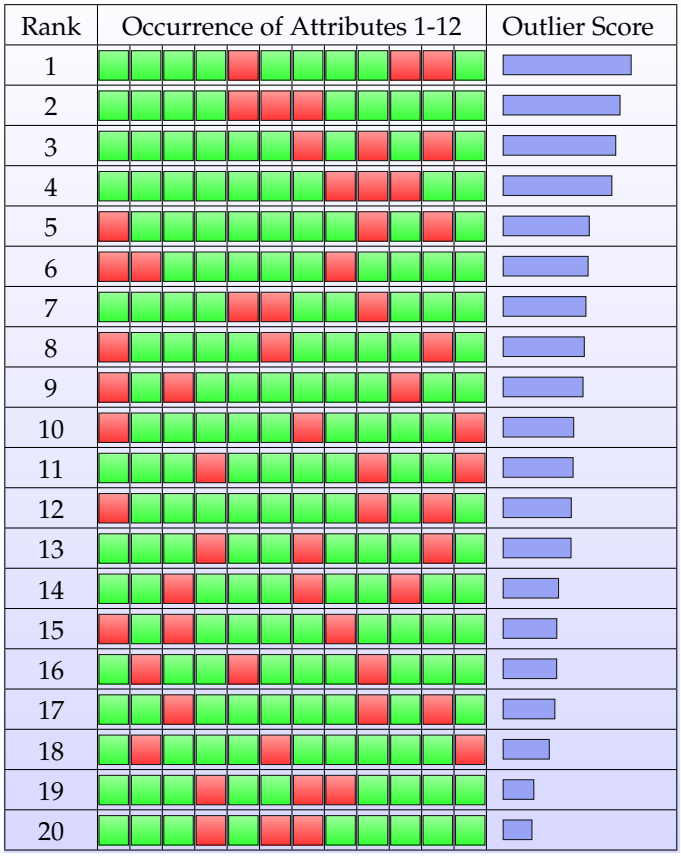


Figure 8.8.: Score discrepancy for $S = \{1, 2, 3, 4\}$

In the following we point to the three major challenges of the refinement problem and explain how we deal with them in our proposed solution.

Uncertainty of populations: This challenge refers to the general problem of comparing populations. For instance, the example demonstrates that the two populations are not strictly separable in general due to statistical fluctuations: We observe that the subspace on rank 7, which is a superset of the true subspace, is ranked below two irrelevant subspaces that coincidentally show a high outlierness for the object. Hence, any solution of the refinement problem must handle uncertainty in outlier score distributions. Another issue is that for a high dimensional S', the partition may yield a very small sample $\mathcal{P}_{S'}^+$ due to the low coverage probability of high dimensional subspaces. In this case the size of the outlier score populations becomes unbalanced, i.e., $\mathcal{O}_{S'}^+$ is much smaller than $\mathcal{O}_{S'}^-$. For instance, if we consider an S' that corresponds exactly to the top ranked subspace in the example, the statistical significance of comparing $\mathcal{O}_{S'}^+$ and $\mathcal{O}_{S'}^-$ is low since $|\mathcal{O}_{S'}^+| = 1$. Therefore, we propose to rely on statistical tests that are designed for comparing populations and

properly handle uncertainty. To quantify the separation power for a given candidate C, our approach requires an instantiation of the following function:

$$\begin{aligned} discrepancy\,(\mathcal{O}_C^+, \mathcal{O}_C^-) \;\equiv\; & \text{p-value of a statistical test} \\ & \text{that is sensitive to} \\ & E\,[\mathcal{O}_C^+] \;>\; E\,[\mathcal{O}_C^-] \end{aligned}$$

By using the p-value we leave the question of the statistical significance to the underlying test: In case of a very small population $\mathcal{O}_C^+$, any reasonable test will report lower p-values, since it is not possible to reject the null-hypothesis of identical populations with a high certainty. There are many possibilities to instantiate the statistical test. For instance, we can use the one-sided versions of the Mann-Whitney-Wilcoxon test or the Student's t-test. We evaluated several instantiations in our experiments. Although we observed only minor differences, we obtained the overall best results with Welch's t-test (a Student's t-test without assuming equal variances of the samples). The reason could be that a t-test is more sensitive to outliers compared to the Mann-Whitney-Wilcoxon test, which only considers the ranks of the populations. While the t-test's sensitivity to outliers is an issue in other domains, it actually is useful in our case: For a high dimensional true subspace S the coverage probability is low. Thus, we might only have a few matching subspaces in the subspace pool. Fortunately, the t-test captures this discrepancy well compared to a rank test. According to our experiments, this property seems to outweigh the fact that the Gaussian assumption of a t-test does not necessarily apply to the outlier score distributions.

Joint occurrence property: We know from the outlierness profiles that only the joint occurrence of the attributes S causes an increased outlier score of a true subspace outlier. In projections of S, the object falls in regions of regular density. In the given example, we observe that the individual occurrences of attributes $\{1, 2, 3, 4\}$ below Rank 7 are completely random and independent from each other since the complete set is never included in these subspaces. Detecting joint occurrences highlights the set-like property of the problem and its exponential characteristic: An exhaustive search to find the exact d'-dimensional subspace S' that maximizes the discrepancy of $\mathcal{O}_{S'}^+$ and $\mathcal{O}_{S'}^-$ would require to evaluate the $discrepancy$ of all possible $\binom{D}{d'}$ partitions. Thus, it is not feasible to search for an exact solution. Instead we propose a heuristic search for a subspace S' that approximately maximizes the discrepancy. We define the quality of a candidate subspace C according to the discrepancy of the corresponding partition:

$$quality(C) = discrepancy\,(\mathcal{O}_C^+, \mathcal{O}_C^-)$$

Based on this quality function we perform a beam search of the candidates in a bottom-up processing. A parameter $beamSize$ determines the number of candidates that we keep on each dimensionality level. We start with all possible one-dimensional candidates. In each iteration we calculate the quality $quality(C)$ of all candidates C. We rank the candidates depending on their quality and discard all candidates that have low quality, i.e.,

we only keep the top-$beamSize$ ones. These top candidates are used to construct higher dimensional candidates. This construction is similar to constructing higher dimensional candidates in frequent itemset mining [AS94]: We form a $(d+1)$-dimensional candidate in case our candidate set contains all its d-dimensional projections. If it is not possible to construct a higher dimensional candidate, the processing stops.

To highlight the rationale of such a processing we discuss the question whether there is some kind of monotonicity in the candidate generation. In frequent itemset mining, monotonicity refers to the fact that when the quality criterion of a candidate C (in this case the itemset support) is above a certain threshold, so it is for all subsets of S. In our score discrepancy problem, we are faced with a quality criterion which is more complex than a simple count of items, and monotonicity does not hold. However, we observe that our problem has a property which we would call *per-level-monotonicity*. On a fixed dimensionality level d, we have

$$quality(C_{true}) > quality(C_{rand}) \tag{8.5}$$

where C_{true} are d-dimensional subsets of S and C_{rand} are random d-dimensional candidates which do not share attributes with S. We can see this by noting that $\mathcal{O}^+_{C_{true}} \supseteq \mathcal{O}^+_S$. Thus, the population $\mathcal{O}^+_{C_{true}}$ contains all increased scores of the true population $\mathcal{O}^+_S$ plus a random sample of $\mathcal{O}^-_S$. When taking expectation values, we still have:

$$E\left[\mathcal{O}^+_{C_{true}}\right] > E\left[\mathcal{O}^-_{C_{true}}\right]$$

For random candidates C_{rand} the expectation values of the samples $\mathcal{O}^+_{C_{rand}}$ and $\mathcal{O}^-_{C_{rand}}$ are the same, and thus, Eq. 8.5 holds. This per-level-monotonicity ensures that by keeping the top candidates on each level in the beam search, we maximize the likelihood of finding the correct S in each step.

To finally obtain the refined d'-dimensional output subspace, we proceed as follows: During the bottom-up beam search we keep a record of all candidate qualities ever evaluated. We rank all candidates according to their $quality(C)$, i.e., their p-values expressing how well they separate the outlier score populations. To collect exactly d' attributes for the output candidate, we iterate over this list, starting with the top ranked candidates. We add the attributes of the candidates in the ranking to the output candidate S' until $|S'| = d'$. In case of adding a candidate C completely would yield $|S'| > d'$, we rank the attributes $a \in C$ according to their one-dimensional qualities $quality(\{a\})$ and only add the best attributes until $|S'| = d'$.

Limited size of subspace pool: Another challenge is introduced by the limited size of the subspace pool. If this number is low, combinatorial interferences are likely to occur. For instance, the last attribute in Figure 8.8 is not part of the relevant subspace. But since it was never excluded from the top ranked subspaces, there is no way to detect that it is an irrelevant attribute for the given object. Due to the limited number of combinations, the attribute must be added to the set of relevant attributes as a false positive. In order

to completely avoid false positives, it would be necessary to evaluate all $\binom{D}{d}$ possible d-dimensional subspaces on each level. Clearly this is not feasible. However, we can reduce the issue of false positives by relaxing the general goal of the subspace refinement. After all, any reduction of irrelevant attributes already improves outlier detection. Thus, detecting the true S precisely is unlikely unless we construct a huge subspace pool. Instead, the framework increases outlier detection quality by refining the subspace to a dimensionality level d_2. This allows the refinement step to output an $S' \supseteq S$ which may include some false positive attributes. From the framework's point of view, the main goal is achieved: It has been possible to remove $(d_1 - d_2)$ irrelevant attributes, adaptively on the underlying outlier model, allowing enhanced outlier detecting by scoring an object in its individually best subspace S'.

We conclude this section with a brief summary of our solution: The proposed `Refine` function extracts a refined subspace individually for each object based on the outlier scores according to the underlying outlier model. These properties, per-object processing and adaptiveness, distinguish our approach from existing subspace search techniques [CFZ99, KKKW03, KMB12, NMV+13]. The refined subspace is obtained by maximizing the discrepancy in outlier score distributions. Our algorithm performs a beam search that exploits the per-level-monotonicity. Exploiting this special property of our problem distinguishes our approach from approaches e.g. in subgroup detection [Wro97], where such a property does not hold. Furthermore, we have proposed a construction of the output subspace which allows $S' \supseteq S$, and thus, is tailored to the idea of refining subspaces within the enclosing RefOut framework.

8.4. Experiments

Our experiments focus on the interplay of traditional outlier models with subspace search approaches. From the field of outlier models we chose three representative techniques: (1) Local Outlier Factor (LOF) [BKNS00], (2) distance-based outlier detection (DB) [KN98], and (3) angle-based outlier mining (ABOD) [KShZ08]. Our general evaluation scheme is to combine these three models with the following subspace selection schemes: (1) random subspace selection (RS) and (2) the full attribute space (FS) as two baselines; (3) HiCS [KMB12] as representative of subspace search techniques; (4) RefOut. For HiCS and RS we always use the maximum outlier score of all subspaces. To ensure repeatability, we provide details on our experiments online.[†]

Our main focus is to analyze outlier detection quality on real world data. We use the area under the ROC curve (AUC) as quality criterion. To perform scalability experiments and to evaluate all RefOut parameters, we utilize synthetic data. Our synthetic data generator injects true subspace outliers in a database as follows: We partition the attributes of the

[†] http://www.ipd.kit.edu/~muellere/RefOut/

database of dimensionality D in subspace components of dimensionality d randomly between 2 and 8 with equal probability. To create a structure of regular objects in each subspace component, we draw random values satisfying $x_{s_1} + \ldots + x_{s_d} = 1$. We inject a true subspace outlier by deviating one object slightly from this hyperplane, satisfying that all its lower dimensional projections are in a region of regular density. This special type of true subspace outlier can be detected clearly by all three outlier models in the subspace components.

Dataset (size x dim)	**Ground Truth**	**Peaks in Dim**				
		1	2	3	4	5
Breast (198 x 31)	ABOD	0	139	40	16	3
	DB	58	81	44	15	0
	LOF	36	67	52	29	14
Breast Diagnostic (569 x 30)	ABOD	0	284	187	98	-
	DB	101	268	155	45	-
	LOF	94	177	177	121	-
Electricity Meter (1205 x 23)	ABOD	6	217	405	577	-
	DB	99	537	393	176	-
	LOF	197	374	413	221	-

Table 8.9.: Datasets and dimensionality of peaks

8.4.1. Adaptiveness on Real World Data

As already illustrated in our toy example in the introduction, it is clear that a LOF outlier is not necessarily an ABOD outlier. Since the true subspace outliers are individual to each model, it would be desirable to have a ground truth of true subspace outliers of each type. To this end, we introduce a novel evaluation approach for detection quality of true subspace outliers in dependence on the outlier model. We propose to perform an *exhaustive search* to obtain a ground truth of true subspace outliers for each model. That is, we scan all subspaces of a dataset exhaustively with each model up to an upper dimensionality level. This is obviously a very time-consuming operation. Therefore, we have to focus on datasets of moderate size and dimensionality to reach a reasonable upper dimensionality level. We chose the datasets Breast, Breast Diagnostic [FA10] and a larger Electricity Meter dataset from a collaboration partner. Note that we had to drop two discrete attributes from the Breast dataset to ensure a well defined local outlier factor. We further normalized all attributes to a unit interval. We scanned up to a dimensionality of 4 for Breast Diagnostics (31,930 subspaces for each model) and Electricity Meter (5,488

subspaces), and up to level 5 for Breast (206,367 subspaces). The overall scanning took several days, mainly spent on running ABOD (using the FastABOD version [KShZ08]).

Since in Sec. 8.2 we defined the target function to quantify true subspace outliers to be the height of the peak, we store the maximal peak for each object and the corresponding subspace during our exhaustive scan. A first insight is that the three models show very different distributions regarding the dimensionality in which each object showed its maximal subspace outlierness. These results are given in Table 8.9. For instance, we can see that for Breast and Breast Diagnostic LOF tends to see more high dimensional peaks, while for Electricity Meter ABOD detects more high dimensional peaks. Note that for ABOD the outlierness rarely peaks in 1-dimensional subspaces, since the ABOD score degenerates to a (still meaningful) variance over reciprocal distance products in one dimension.

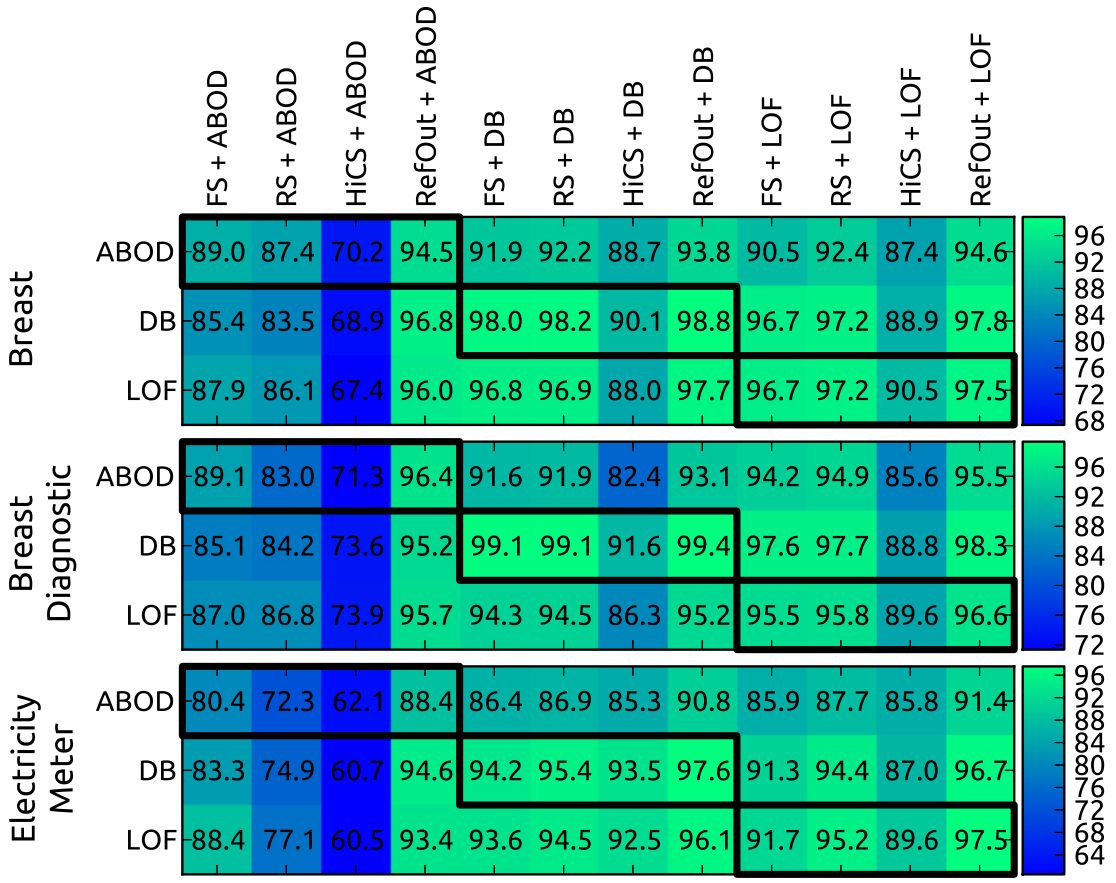


Figure 8.10.: True subspace outlier detection quality (AUC) on real world data

For the following experiments we rank the peaks (for each model and dataset) and extract three different true subspace outlier ground truths for each model corresponding to the top 2%, 5%, and 10% of the peaks. This allows us to investigate interesting cross evaluations and analyze questions like how well does LOF detect ABOD outliers, or which one of the

true subspace models is the hardest to detect in the full space? To this end, we evaluate all 12 combinations of {FS (full-space), RS (random-subspaces), HiCS, RefOut} × {ABOD, DB, LOF} on all ground truths. The average AUC values of these experiments are shown in Fig. 8.10. Each row corresponds to a certain ground truth model ABOD/DB/LOF. We highlight the blocks where a subspace approach uses the same outlier model as the ground truth, and intuitively we expect the best results in this case. We can see that this for instance is strongly pronounced for Breast Diagnostic with the DB model. On the other hand, we were surprised to find that the ABOD ground truth is sometimes better detected using DB/LOF instead of ABOD itself as detection model.

Regarding the adaptiveness of the subspace search models, we can see that the static selection scheme of HiCS does not perform well in general, especially in combination with ABOD. Using random subspaces shows better overall adaptation simply by making no assumption for the selection at all. In most cases RS improves over a full-space detection, but not when combined with ABOD. Regarding RefOut, we can see that its adaptive design clearly improves the subspace selection for all models. We observe the most pronounced improvement over the other subspace techniques in combinations with ABOD. The systematic quality improvement of RefOut comes along with a slightly increased runtime: The average runtimes over all models and datasets were: 41.6 sec for RS, 49.0 sec for HiCS, and 76.2 sec for RefOut, which is still several orders of magnitudes below the runtime for exhaustive searching and is worth to be invested when looking at the improved detection and description of individual outliers.

8.4.2. Scalability with Dimensionality

To analyze the dependence of the detection quality with the database dimensionality we performed experiments on different dimensionality levels. We generated 5 random datasets on each dimensionality level 25, 50, 75, and 100 with subspace outliers of a random dimensionality up to 8. For this experiment we focus on a single outlier model to keep the number of results manageable. We chose the LOF outlier model due to its high popularity. We kept the LOF parameter $MinPts = 10$ constant for all approaches. For the random subspace detection we chose the same dimensionality level as the dimensionality of the initial pool of RefOut (75% of D) to highlight the improvement due to subspace refinement. We keep the total number of evaluated subspaces equal for RS, HiCS, and RefOut. Fig. 8.11 shows the results. Regarding quality, we can see that even the random subspace approach consistently outperforms a fullspace subspace detection. Regarding HiCS we can see that it can improve over random subspaces on average. But we also see the effect of its non-adaptiveness: Sometimes the subspaces detected by HiCS match quite well (on the 50 dimensional level); other times HiCS outputs subspaces that are of no use to the outlier model (on $D = 75$). For RefOut we observe a very good scalability with respect to the dimensionality: The subspace selection consistently outperforms the other subspace approaches. The price for the increased quality is a slightly increased

runtime. However, we can see that the increase over the runtime baseline defined by RS is rather low: This means that the majority of the runtime is spent on applying the outlier model itself and not on the subspace refinement framework. Overall RefOut shows a linear scalability w.r.t. the number of dimensions, making it capable of handling high dimensional databases.

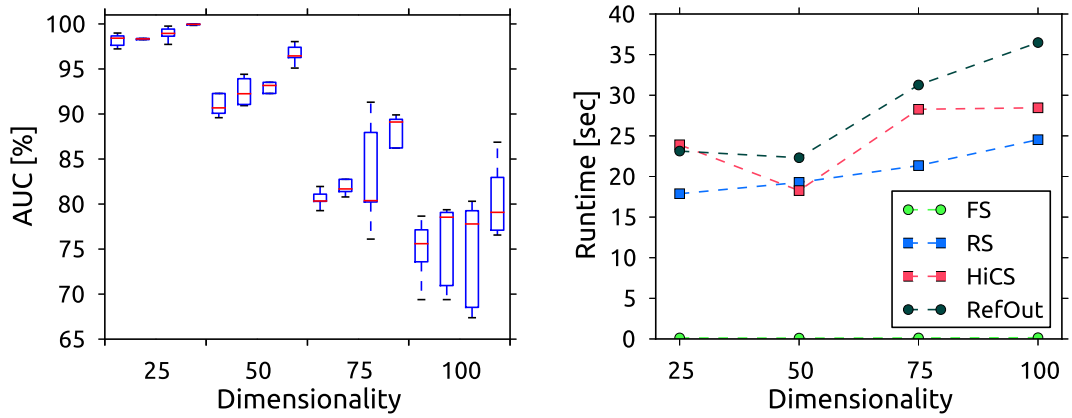


Figure 8.11.: Scalability w.r.t. increasing dimensionality on synthetic data (from left to right in each group: FS, RS, HiCS, RefOut)

8.4.3. Parameter Evaluation

We performed a thorough analysis of all parameters in RefOut, again based on the LOF model. We evaluated each parameter configuration on the pool of 20 datasets for Sec. 8.4.2. This means that the dataset pool contains both difficult and more easier datasets. In our opinion this is important to ensure that we do not analyze the influence of a parameter for a single database dimensionality. In order not to use absolute values for d_1 and d_2, we set these parameters as percentage of D. Our default parameters were $psize$=100, $opct$=20%, d_1=75%, d_2=30%, and a $beamSize$=100. Starting from this configuration we performed a sensitivity analysis by varying each parameter individually. The results are shown in Fig. 8.12. We can see that in general the parameters are robust and slight variations of a parameter do not harm the results significantly. Note that the main fluctuations in the results are caused by the broad spectrum in difficulty of the datasets. As expected, increasing the pool size has a positive influence on the results, although we did not observe further improvements above a pool size of 125. The $opct$ parameter that controls how many objects are considered for subspace refinement is also straightforward to set up: Higher values produce better results since the detection quality of the high dimensional subspace scan is less relevant. Our primary choice of 75% for d_1 was motivated by the idea that we wanted both good subspace coverage while keeping the number of irrelevant attributes low. The results show that this choice was still a bit too high: Checking subspaces of a dimensionality of 60% gave slightly better results. This indicates that RefOut works

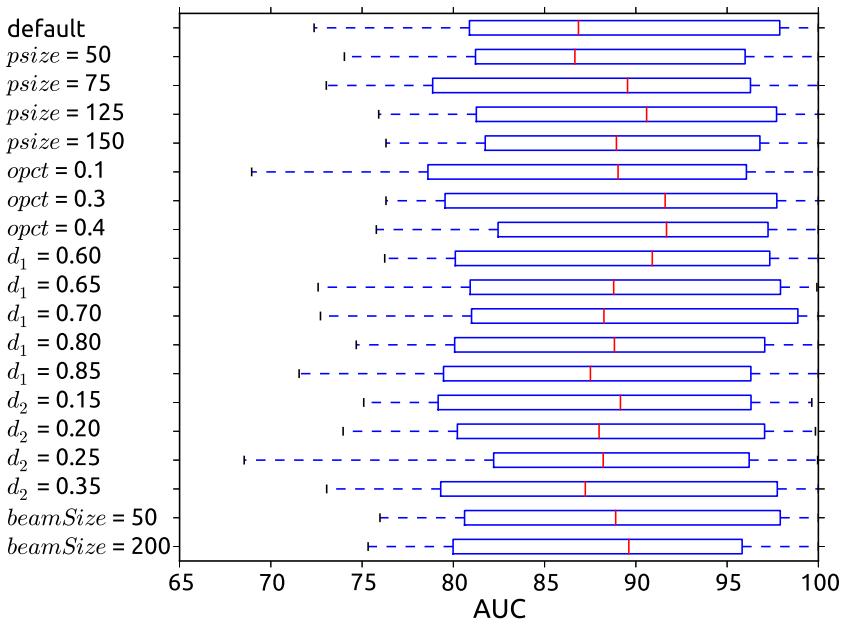


Figure 8.12.: Parameter evaluation

well with a low subspace coverage; the influence of irrelevant attributes is the bigger issue. We did not observe a significant influence of the $beamSize$ in our bottom-up subspace refinement on the results, which shows that even low values in the beam search can find reasonably good refinement candidates.

8.4.4. Study of Descriptive Power

In addition to the quality and runtime results, we study the descriptive power of RefOut in a real-world scenario. Therefore, we conclude our experiments with an evaluation of the individual outlier subspaces provided by RefOut for the Breast Cancer (diagnostic) dataset. We interpret the subspaces as individual anomaly descriptions in the context of breast cancer diagnostics. The dataset features patient records, and each patient is described by 30 real-valued attributes. The attributes are computed from a digitized image of a fine needle aspirate of a breast mass [FA10]. This includes for instance the radius of a cell nucleus, its symmetry and its concaveness. In Table 8.13 we present an excerpt of RefOut's results for patients with a malignant tumor.

To illustrate, the top ranked anomalous patient shows a deviation in a specific subspace, including different properties such as the deviation of the cell nucleus area or its fractal dimension. This specific anomaly is detected with a very high outlierness value. This is a result of selecting the subspace which maximizes the deviation of this patient compared with the majority of healthy patients. A full-space analysis of this patient is likely to show a much lower deviation due to irrelevant attributes. It might even miss the anomaly.

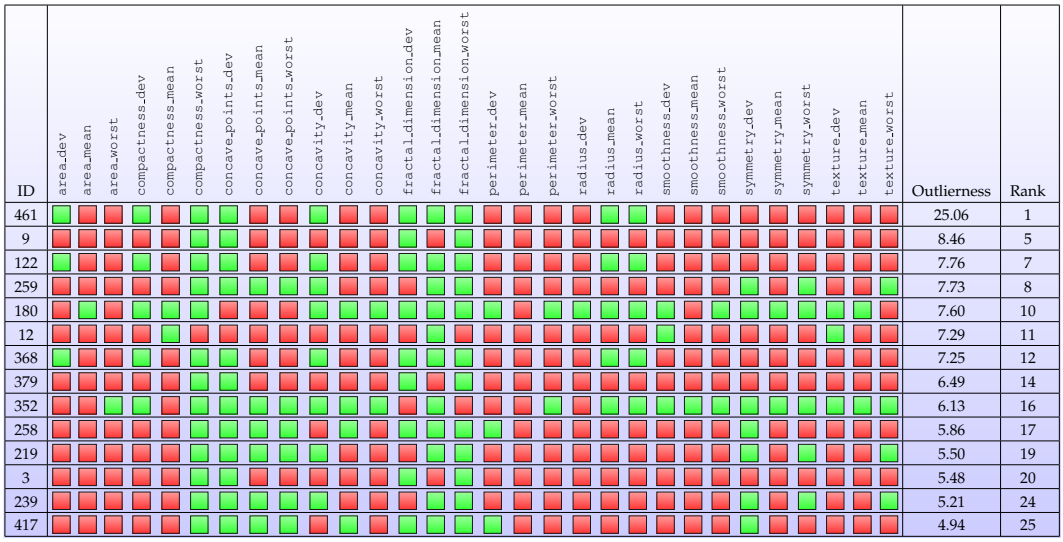


Figure 8.13.: Individual subspace outliers for the Breast Cancer (diagnostic) dataset

We now look at another diseased patient, for instance the one with ID 12. We can see that the disease manifests itself in a completely different anomaly this time. For this patient, the anomalous context contains the deviation of the cell nucleus texture, next to other attributes. Since this anomaly context is rare, it might be possible to identify a different disease pattern from a medical point of view.

Overall, we can see that the anomalous context varies considerably for different diseased patients. The individual subspace search of RefOut is the first decoupled approach that identifies anomalous objects and their *individual* subspace context at the same time. This example also points to two possible contributions of the individual subspace search: First, it helps physicians to detect *and* describe an anomalous health status of a patient. Second, it allows to develop new insights regarding different clinical pictures of a disease by analyzing individual anomalous contexts.

8.5. Conclusions

In this chapter, we present a flexible and adaptive subspace search technique for outlier mining. It refines a pool of random subspaces by exploiting the score discrepancy in different subspaces. Based on the statistical comparison of outlier scores, we achieve an adaptive search tailored to the underlying outlier model. This allows us to inherit the properties (quality, performance, etc.) of various well-established outlier definitions for the subspace search. This results in an improved outlier detection but also in individual outlier descriptions for each object.

Part III.

Attribute Relationship Analysis on Data Streams

9. Estimating Mutual Information on Data Streams*

In the following part of the thesis we will turn our attention to attribute relationships in the context of data streams. Clearly, the long term goal for data streams is the same as for static data, i.e., to solve the challenges of subspace outliers by means of studying attribute relationships. However, as a result of the dynamics of data streams, the problem of analyzing attribute relationships changes fundamentally: For data derived from a stream, the attribute relationships themselves can vary over time. Therefore, it is no longer enough to consider the question "what is the relationship between a certain set of attributes?" Due to the dynamics of data streams, every statement regarding attribute relationships must be tied to a temporal context, i.e., a time frame extending from a certain time t_1 to t_2. Thus, the question becomes: "What is the relationship between a set of attributes within a certain time frame?" This means that now analyzing attribute relationships is no longer only a problem of combining attributes, but also a problem of considering all temporal contexts. Thus, the resulting problem has a significantly higher complexity compared to the static case. Accordingly, the full problem of analyzing attribute relationships on data streams has not been studied thoroughly in the scientific literature. Even for the case of pairwise attribute relationships, the scientific literature is scarce for a strictly online scenario. Therefore, as a first step towards a subspace search on data streams we propose to consider the problem in its most basic form. Specifically, we will focus on the case of

- bivariate attribute relationships, and
- temporal contexts which are externally specified by the user.

This simplifies subspace search to a two-dimensional subspace query problem, i.e., the aim is to enable answering questions like "what is the relationship between a certain attribute pair in a certain time frame". Since even this problem does not have a general solution on data streams, we think it is an important step to address this basic, unsolved problem first. In the long term, such a simplified subspace search can provide a basis for future extensions. Such extensions can either move from bivariate to multivariate subspaces, or include an automatic mining of temporal contexts as well.

* This chapter is an extended version of *Estimating Mutual Information on Data Streams* published in the Proceedings of the 27th International Conference on Scientific and Statistical Database Management (SSDBM) 2015 [KMB15].

Another motivation for focusing on bivariate attribute relationships results from the observation: *Multivariate relationships on static data are often the result of not incorporating temporal effects.* To illustrate this observation, think of a scenario where one performs a daily measurement of *soil moisture* and the *amount of rainfall*. For most times of the year, these two measurements will show a strong correlation. But even with such a simple and strong relation, there are possibilities for exceptions. For instance, the relation might change in midsummer or winter in case of artificial watering or freezing effects. Thus, the bivariate relation itself might change over time. Storing the measurements in a static database without temporal information does not allow to see such changes directly, since the different periods would be mixed up in the database. However, they may be visible indirectly in the form of a multivariate relationship. For instance, consider we store an additional measurement *average daily temperature*. In this case, there will be a multivariate dependence in the three dimensions, because the average daily temperature will serve as a temporal hint: For very high and very low temperatures (hint for midsummer/winter), the dependence of moisture/rainfall will be lower than for moderate temperatures (hint for spring/autumn), resulting in a typical example of a ternary relationship. Note that this is a result of the fact that one dimension encodes temporal information, regardless of the impreciseness of this information. On static data, this effect can always be observed when a bivariate relationship changes over time, and a third quantity allows to roughly distinguish between the different time periods. Thus, many multivariate relationships which are visible in a static view on the data are often artifacts of dynamic effects. Such an indirect reconstruction of temporal effects from multivariate dependencies must obviously be less precise compared to considering the temporal effects explicitly. The reverse conclusion from these observations thus becomes: By incorporating temporal effects directly, many static multivariate relationships can break down into temporary bivariate relationships. This further motivates our focus on the bivariate case as the first step on data streams.

9.1. Overview

In information theory, mutual information is a ubiquitous measure for the mutual dependence of two random variables. First introduced in 1948 as part of Claude Shannon's fundamental contributions to information theory, mutual information has a long history in both theory and application [DH07, Qiu12, JYX13, LLZG10, HS10, SS12]. Intuitively, mutual information $I(X, Y)$ is equal to the reduction of uncertainty on one random variable X given knowledge of another variable Y. It is a symmetric measure, i.e., $I(X, Y) = I(Y, X)$. Most commonly, mutual information is measured in the unit *bits*, which facilitates interpretation: A high mutual information indicates a large reduction of uncertainty, i.e., the variable pair shows a strong mutual dependence. Compared to other dependence measures, like for instance Pearson or Spearman correlation, mutual information is not limited to specific kinds of dependence, e.g., linear or monotonous,

but captures every possible type of dependence. As a result of these properties, mutual information has furthermore played an important role in the development of specialized mutual dependence measures [RRF+11].

Calculating the theoretical mutual information value is straightforward when the underlying probability density functions are known. On real-world data however, the distributions are commonly unknown. Proper estimation of mutual information from real-valued data is a non-trivial problem and has been covered in the literature extensively. Amongst the traditional mutual information estimators, Kraskov estimation has emerged as a leading approach [KSG04, KBG+07, WWL09, KA13]. Consequently, we build upon this estimation principle in this work. Traditional estimation algorithms however focus on the case of a fixed, static data sample. The notion of time is not considered explicitly. This is a fundamental difference to data originating from a stream. By its nature, a data stream is evolving and changing over time, is infinite, and comprises multiple time scales. Given these properties, the analysis of data streams has become a challenging task in the database research community.

To illustrate, think of analyzing the mutual dependence in a stream of stock prices. Detecting a mutual dependence of stocks provides important information for financial analysis, investment management, or return prediction. In general, the mutual dependence between stocks fluctuates over time, and one may observe periods of high or low mutual information. Furthermore, the changes in mutual dependence may occur on a broad range of time horizons: In some cases a mutual (in-)dependence lasts for decades, while in other cases a dependence appears and disappears within seconds. An analyst might for instance find a dependence of a pair of stocks in July. This leads to questions like whether the dependence did also exist in June, when it has appeared first, or whether it also exists on different time scales like a yearly time horizon or when looking at hours or minutes. Overall, we make the following key observation: Since the dependencies are dynamic, each analyst may be interested in a different time window. That is, analysts want to estimate mutual information based on an arbitrary window size, and the window may be shifted arbitrarily into the past.

Challenges. This observation has a direct implication when designing a data stream management system (DSMS) that supports mutual information queries: The DSMS must allow a user to explicitly specify the query window boundaries individually for each query. In general, such queries are so-called ad-hoc one-time queries [BBD+02], and they are most challenging since this is the most general type of query. Supporting such queries even raises the question: Is it possible at all to answer mutual information queries in *any* window without storing the entire data stream? Naively, one could approach the problem by (1) storing the entire stream and (2) running a static mutual information estimator for every incoming query. Clearly, this naive approach has severe limitations and is not in line with the so-called "streaming model" [BBD+02]: First, storing the stream obviously contradicts the idea of stream processing. The second issue affects query performance; we illustrate it using our example scenario: If there are many analysts working simultaneously,

the DSMS has to answer many queries as well; the rate of incoming queries may even exceed the data rate of the stream. In such a case, the naive approach will collapse since it has to expensively recompute a mutual information estimate for every single query – even if the windows of two queries have a significant overlap. Therefore, a challenge is to develop a summarization data structure which provides aggregated information that is useful for many queries.

Our Contributions. In this work, we propose the framework Mise (Mutual Information Stream Estimation) which tackles the challenge of answering mutual information queries in arbitrary time windows. To avoid storing the whole data stream, we exploit the *multiscale* nature of time. We illustrate the idea in our example scenario: For financial analyses, time scales can vary significantly, ranging from seconds right up to years or decades. In such analyses, the query window size and the amount the query window is shifted into the past often show a certain relationship. We exploit this by dividing the space of all possible queries into multiscale equivalence classes depending on the ratio of the window size w and the offset o into the past. For instance, the following two queries are equivalent: (I) a query with $w = 1$ second and an offset of $o = 5$ seconds, and (II) a query with $w = 5$ years and an offset of $o = 25$ years. In this work we will tackle the essential question that arises with multiscale equivalence: How can a DSMS answer equivalent queries with equal quality? As a key contribution we provide a solution to this question by deriving the proper sampling distribution out of this requirement. We will see that the common principle of *more detail on more recent data* emerges naturally as a result. Based on the sampling distribution required, we develop two different *multiscale sampling schemes* which have either constant or logarithmic complexity over time. They are the first sampling schemes that inherently provide equal quality over multiple time scales.

As another important contribution, we introduce the notion of a *query anchor*, which is a novel dynamic data structure for mutual information estimation. In a nutshell, a query anchor keeps track of quantities that allow to estimate mutual information according to the Kraskov principle. These quantities include nearest neighbor relationships and counts of data points in the marginal distributions. While the computation of these quantities is straightforward on static data, the challenge with data streams becomes: To obtain an efficient estimation, it is necessary to keep track of all changes in these quantities over time. By proposing the query anchor data structure, we solve this problem and enable an incremental computation of these quantities. Consequently, query anchors provide aggregates that can be used for different queries. This leads to a significant speed-up of query execution time.

Summing up, our contributions to deal with the challenges mentioned are as follows: We deal with

- ***the stream's dynamic nature*** by design, i.e., by allowing the user to query the stream in arbitrary windows,

- ***the stream's infinite*** and ***multiscale nature***, by introducing the novel multiscale sampling paradigm,
- ***a large number of online queries***, by efficient incremental computations within the query anchor data structure.

Furthermore, we provide a detailed analysis of both our multiscale sampling schemes and the query anchor data structure. To complement our analysis, we demonstrate the high quality of Mise in a broad range of experiments, including several real-world scenarios.

This chapter is structured as follows: We first review estimation of mutual information on static data in Section 9.2. In the following Section 9.3, we give an overview of related stream summarization techniques. We proceed by introducing the Mise framework in Section 9.4, and provide a formal analysis of the complexity in Section 9.5. The empirical analysis of the Mise framework follows in Section 9.6. Section 9.7 provides a concluding summarization and takes a look at future work.

9.2. Static Estimation Paradigms

Estimation of mutual information on static data has been studied in many publications, including several surveys [PSMP07, WWL09, KBG+07]. Estimators can be categorized according to the underlying formula of the estimation. The first estimation paradigm is based on the integral definition of mutual information:

$$I(X,Y) = \iint p(x,y) \log \frac{p(x,y)}{p_X(x)p_Y(y)}\, dy\, dx \tag{9.1}$$

where $p(x,y)$ is the joint probability density function, and marginal distributions are denoted as $p_X(x)$ and $p_Y(y)$. Estimators of this type replace these theoretical functions by density estimates; they are hence called plug-in estimates [PSMP07]. A common problem of such estimators is that, since the underlying distributions are unknown, they are prone to underestimating the variability of the distributions based on a finite sample. This leads to a heavily biased estimate of mutual information, which has been studied extensively [Pan03, Sch04, DV99, DSSK04]. Furthermore, to the best of our knowledge, there is no general purpose density estimation technique that allows to query density estimates in arbitrary windows over a data stream. Therefore, we will focus on the second estimation paradigm in this work. Estimators of this kind are based on the entropic definition of mutual information:

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \tag{9.2}$$

where $H(.)$ denotes entropy. Therefore, estimating mutual information can be achieved by an estimation of entropy. The Kozachenko-Leonenko-Estimator [KL87] is a famous non-parametric approach to estimate entropy based on nearest neighbor information.

The problem regarding estimation of mutual information is that the errors of estimating the marginal and the joint entropies do not cancel. This was solved by Kraskov et al [KSG04], leading to a mutual information estimator with excellent estimation properties: Comparative studies [KBG+07, WWL09, KA13, KSG04] have shown that the Kraskov estimator (1) shows a very fast convergence, (2) is unbiased in case of independent variables, and (3) shows very low bias in general compared to estimators of the first paradigm. However, it is an open research question how to incorporate its principles into the estimation process for data streams. Our goal is to make these favorable estimation properties available for online processing.

9.3. Related Work

Analyzing related work shows that certain issues recur. Therefore, we first summarize recurring limitations before analyzing related work in detail.

• FixedWindow: Many data stream techniques do not allow the user to specify arbitrary query windows. Summarization techniques typically maintain a synopsis aggregated either over the whole stream, a sliding window, or – as generalization of these paradigms – aggregated based on a certain (smooth) time decay function. In either case the scope in time is fixed, i.e., the "query window" is inherently bound to the aggregate computation. In particular, most techniques focus on keeping track of the *most recent* aggregate value. This prevents the user from querying the aggregate in any window that is strictly in the past.

• LimitedDomain: Many data stream summarization techniques are exclusively designed to operate on data streams consisting of discrete items or integer values within a limited range. Compared to real-valued attributes, the finite attribute domain simplifies any summarization task since it allows to operate on item frequencies, which again allows to make use of various sketching techniques [CGHJ12]. From an estimation theoretic perspective, the major challenge of estimating mutual information on continuous data is a result of the infiniteness of the attribute domain. Therefore, making any assumptions regarding the domain is not feasible when constructing a general purpose estimator.

• Univariate: Many of the techniques discussed below are designed for summarizing a single univariate stream. In order to leverage them to estimate mutual information, it would be necessary to modify them to the bivariate case. In many cases such a modification is non-trivial or impossible.

• Biased: Many ideas discussed below would result in a mutual information estimator based on Equation 9.1, and would come with all the issues discussed in Section 9.2.

Estimation Foundations. We now review stream summarization techniques as proposed in the scientific literature that may serve as a foundation for computing a mutual information estimate. For instance, one might be tempted to leverage techniques which summarize quantiles to estimate mutual information. This problem of summarizing ϵ-approximate quantiles has been solved for both the single pass [GK01] and the sliding window [AM04] paradigms. Such an approach would suffer from FixedWindow (inability to specify arbitrary queries) and Biased (due to the binning characteristic of quantiles), and most notably it remains a non-trivial problem to extend the notion of summarizing one dimensional quantiles to the bivariate case (Univariate). Similarly, maintaining histograms as a summary [DGIM02, GP06] suffers from FixedWindow and Biased as well. Another problem that has been addressed is estimating entropy over data streams [LSO+06, BG06, CCM07]. Even if there might be (non-trivial) solutions to issues FixedWindow and Univariate for these techniques, a severe problem remains: The techniques heavily rely on the assumption of a limited attribute domain (LimitedDomain). Overall we can see that all existing summarization techniques are affected by several issues. This highlights the necessity to develop a novel summarization data structure.

Correlation Analysis. Mutual information in the broader context of (pairwise) dependence measures in general is related to work on online correlation tracking. In particular the so-called all-strong-pairs correlation query problem [XSTK04] has been solved for data streams [ZX08, ZX11]. While issues FixedWindow and LimitedDomain apply for these techniques as well, the major difference is the problem statement itself: Compared to linear binary correlations, mutual information can capture much more complex dependence types.

Sampling. The principle of *more detail on recent data* plays an important role in many approaches on data streams. This concept has been applied for instance to the problem of maintaining specific aggregates according to a time-decay weight function [CTX07, CKT08, CSSX09], and to sampling with a weighted reservoir [ES06, Agg06]. However, none of these approaches derives the weight function from quality requirements on the queries. As a major contribution we will derive the particular weight function that is required to ensure the equal treatment of queries over multiple time scales. We will see that this basic requirement results in a unique dynamic weight function, which does not allow a straightforward application of existing weighted sampling schemes.

Nearest Neighbor Querying. Section 9.4.2 will show that estimating mutual information according to the Kraskov principle requires knowing the nearest neighbors. Thus, our approach is remotely related to work on nearest neighbor (NN) monitoring in spatio-temporal databases. Given a set of objects and a set of query points, continuous k-NN querying addresses the case that both the objects and query points move over time. Techniques like [ISS03] make strong assumptions on the trajectories of the objects, e.g., they must move with a constant velocity. Later work [MPH05, YPK05, XMA05] relaxes these assumptions, but does not explicitly consider the appearance/disappearance of objects or

queries. To some degree this has been addressed in [MMPP07] (by incorporating a sliding window model into continuous queries) and [BOPY07] (by allowing objects/queries to expire after a certain time). While these concepts are somewhat similar with what is needed here, fundamental differences remain. We will discuss these differences after we have proposed an exact problem statement in Section 9.4.1.

9.4. Proposed Approach

In what follows, we will structure the presentation of our approach into three steps. First, we introduce a summarization data structure, a so-called query anchor, which is responsible for collecting the dynamics of nearest neighbor relationships in the data stream (Section 9.4.1). As the next step, we move to the bigger picture, by explaining how the overall algorithm makes use of these query anchors (Section 9.4.2). We will see that, once we are able to keep track of the changes in nearest neighbors over time, we can extract an online mutual information estimation from the query anchors. Finally we turn to the question of how to solve the challenges introduced by the infinite and multiscale nature of the stream (Section 9.4.3). Our solution to this problem will exploit the equivalence of multiple time scales for sampling.

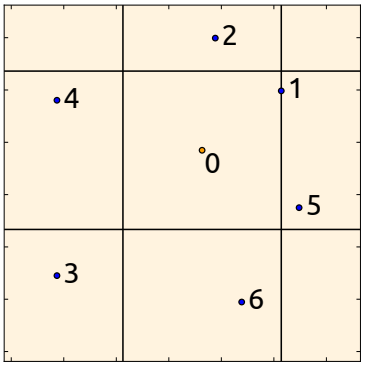
Subject of our analysis is data streams formed by a pair of one-dimensional continuous random variables X and Y. We do not make any assumption on the underlying distributions of X and Y. We assume a fixed sampling rate of the data stream, i.e., samples arrive after a fixed time interval. Extending our approach to variable-rate data streams or more than two variables is part of future work. We denote the pair of realizations at time t as $Q_t = \langle X_t, Y_t \rangle$, where X_t and Y_t are the samples at time t. We will refer to subsequences of the data stream with the notation $\mathcal{Q} = \{Q_{t_1}, Q_{t_2}, \ldots\}$. In general a subsequence $\mathcal{Q}$ can be sparse, i.e., does not necessarily contain consecutive data samples. In order to constrain a subsequence to a certain time window starting at t_s and ending at t_e, we use the following notation:

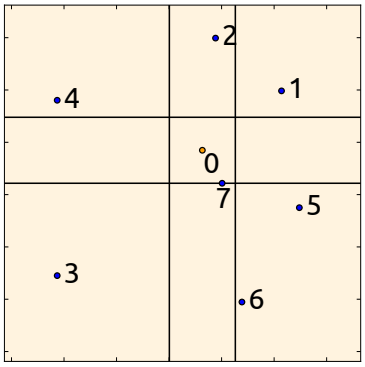
$$\mathcal{Q}_{t_e}^{t_s} \equiv \{Q_t \in \mathcal{Q} \mid t_s \leq t \leq t_e\}$$

Regarding time points, our convention is to use the time t_0 to refer to the present time, i.e., the current or most recent time point available.

9.4.1. Query Anchors

We now introduce the summarization data structure that we use to collect information from the data stream. The computation of a mutual information estimate according to the Kraskov principle requires knowledge of nearest neighbor relationships in both the joint and the marginal spaces. While the computation is straightforward in the case of static data, it becomes a challenge in the online case: Nearest neighbor relationships are





t	MC_x	MC_y
0	0	0
1	0	1
2	1	1
3	1	1
4	1	2
5	1	3
6	2	3
7	2	0

Figure 9.1.: Example showing incremental effects on nearest neighbors, marginal points, and marginal counts

no longer static but inherently change over time, making it necessary to incrementally track changes over time. We will model these dynamics in the following.

Definition 9.1

Distance: We define the distance between two data points Q_t and $Q_{t'}$ according to the maximum norm denoted as:

$$dist(Q_t, Q_{t'}) \equiv max\left(|X_t - X_{t'}|, |Y_t - Y_{t'}|\right)$$

Using the maximum norm is in line with Kraskov and ensures that estimation errors cancel each other out [KSG04].

Definition 9.2

k **Nearest Neighbor Distance:** We define the k nearest neighbor distance of Q_t for a subsequence $\mathcal{Q}$ as the distance to the k nearest neighbor of Q_t in $\mathcal{Q}$. The k nearest neighbor is a point $Q_{t^\star} \in \mathcal{Q}$ satisfying:

$$\left|\{Q_{t'} \in \mathcal{Q} \setminus \{Q_t\} \mid dist(Q_t, Q_{t'}) < dist(Q_t, Q_{t^\star})\}\right| = k - 1$$

We denote the k nearest neighbor distance as follows:

$$kNND(Q_t, \mathcal{Q}) \equiv dist(Q_t, Q_{t^\star})$$

When the subsequence does not have a length of k, the k nearest neighbor distance is undefined. Please note that for the definition of $kNND$ it is irrelevant whether the k nearest neighbor is unique.

We illustrate these definitions by an example given in Figure 9.1. It shows how a nearest neighbor relationship can evolve over time. For simplicity we consider the $k = 1$ nearest neighbor. Our point of reference is the point at time $t = 0$, located near the center in the plots. The points are labeled according to their time of occurrence in the stream. The left plot shows the subsequence $\mathcal{Q}_6^0$, i.e., contains all points Q_t with $0 \leq t \leq 6$. We can see that the $k = 1$ nearest neighbor up to time $t = 6$ is the data point Q_1. The square centered on our point of reference corresponds to $kNND(Q_0, \mathcal{Q}_6^0)$. The next plot shows the subsequence extended by one data point. This leads to an update of the nearest neighbor, reducing $kNND$ given $\mathcal{Q}_7^0$.

Based on the nearest neighbor information, we now define the notion of marginal points, which plays a key role in the estimation process:

Definition 9.3

Marginal Points: Given a point of reference Q_t, we call a data point $Q_{t'} \neq Q_t$ an X-marginal point of Q_t if and only if

$$|X_t - X_{t'}| < kNND(Q_t, \mathcal{Q})$$

where $\mathcal{Q}$ is a subsequence containing both Q_t and Q_t'. We define the marginal points w.r.t. Y correspondingly.

We illustrate this notion using Figure 9.1. Intuitively, marginal points are points that fall into the slices corresponding to the $kNND$ box. For the subsequence $\mathcal{Q}_6^0$ (left plot) and our point of reference Q_0, we identify Q_1, Q_4, and Q_5 as marginal points in Y. With respect to X, only Q_2 and Q_6 are marginal points – Q_1, which defines the k nearest neighbor distance itself, is not included due to the "less than" condition. In the second plot corresponding to subsequence $\mathcal{Q}_7^0$ we observe that some points have lost their marginal point property due to the update of $kNND$. In the Y direction for instance, all three former marginal points are no longer located within the slice.

Definition 9.4

Marginal Counts: Given a point of reference Q_t and a subsequence $\mathcal{Q}$ containing Q_t, we define the marginal counts as the number of marginal points in the subsequence, i.e.:

$$MC_x(Q_t, \mathcal{Q}) = |\{Q_{t'} \in \mathcal{Q} \mid Q_{t'} \text{ is } X\text{-marginal point of } Q_t\}|$$

Accordingly, we refer to the number of Y-marginal points in $\mathcal{Q}$ as MC_y.

The table on the right in Figure 9.1 shows the development of the marginal counts over time. In the following discussion we focus on only one dimension (Y w.l.o.g.). When a

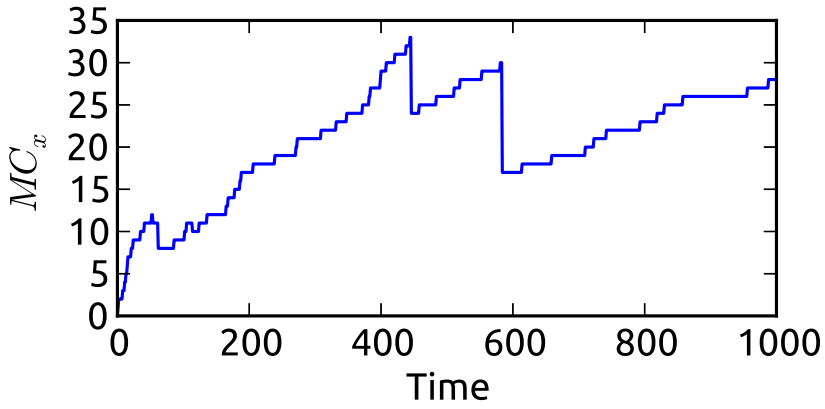


Figure 9.2.: Evolvement of marginal counts over time

new data point arrives, there are in general three possibilities: (1) The data point does not fall into the current Y slice, leaving MC_y unchanged. (2) The data point falls into the slice but has no influence on the current $kNND$. This increments MC_y by one. (3) The data point leads to an update of the $kNND$. Note that for $k > 1$, the new point does not have to be the new best nearest neighbor itself; it can take any position within the top-k ranked neighbors. In general this will result in a new distance of the neighbor on rank k. After such a $kNND$-update, the marginal points have to be re-evaluated. In general the decrease in the k nearest neighbor distance means that MC_y may drop to a lower value. Figure 9.2 shows an exemplary plot of MC_y over time, summarizing these dynamics of the marginal count: As long as $kNND$ is unchanged, MC_y increases monotonically; updates of $kNND$ lead to sudden drops of MC_y. We will further analyze the growth rate of marginal counts over time in our complexity analysis in Section 9.5.

In order to handle these dynamics of marginal counts, we now define the notion of a query anchor:

Definition 9.5

Query Anchor: We define a query anchor as a data structure that precomputes and stores marginal counts. It is associated with a certain data point Q_t, i.e., is located at time t and has knowledge on X_t and Y_t. A query anchor provides

- a method insertRight($Q_{t'}$) which adds a data point $Q_{t'}$ in forward time direction, i.e., $t' > t$,
- a method insertLeft($Q_{t'}$) which adds a data point $Q_{t'}$ in backward time direction, i.e., $t' < t$,

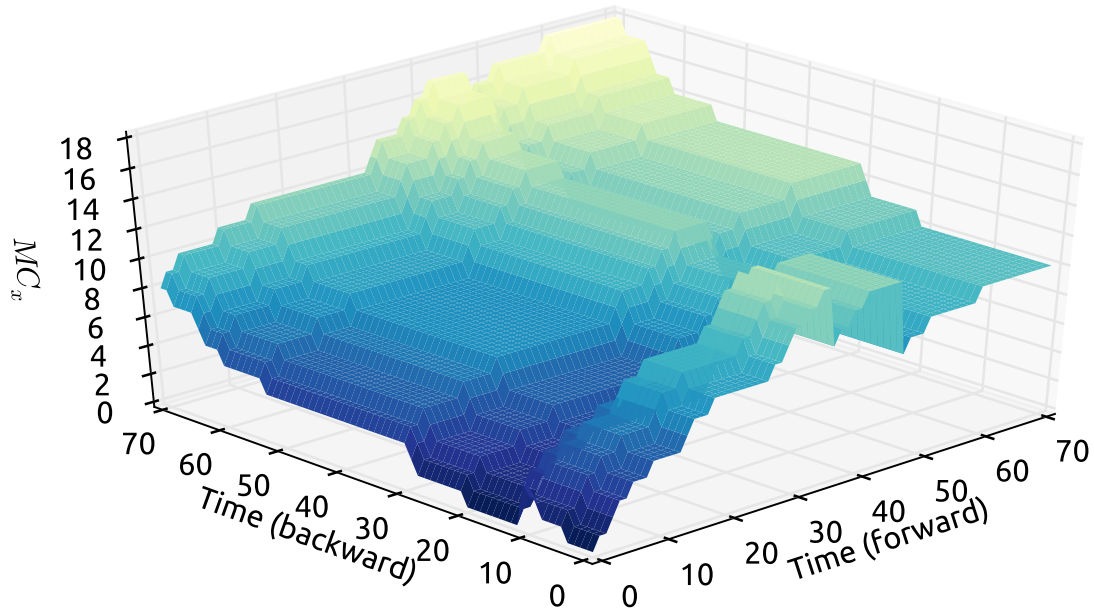


Figure 9.3.: Information on marginal counts stored in a query anchor in forward and backward time direction

- a method QUERY(t_1, t_2) which returns the marginal counts $MC_x(Q_t, \mathcal{Q}_{t_2}^{t_1})$ and $MC_y(Q_t, \mathcal{Q}_{t_2}^{t_1})$, and the total number $N \equiv |\mathcal{Q}_{t_2}^{t_1}|$ of data points that have been shown to the query anchor by its insert operations.

Note that in general the number N can be smaller than the window size $t_2 - t_1$ if the query anchor has only seen a sparse subsequence of the data. Compared to our example from Figure 9.1, a query anchor differs in the sense that it has to keep track of the marginal counts in both time forward and time backward direction. Therefore, by proposing query anchors we allow to incrementally add data from the future or the past by means of the INSERTLEFT and INSERTRIGHT functions. Figure 9.3 shows a two-dimensional illustration of the marginal count MC_y for both time directions: The "increase and drop" behavior from Figure 9.2 can now be observed in forward and backward time direction. We will turn to the question of implementing query anchors in Section 9.5, providing a solution to efficiently store the information contained in Figure 9.3.

Differences to Continuous k-NN Queries. Having formulated the problem statement, it becomes clear that the problem has fundamental differences to work on continuous k-NN queries:

- There, a continuous query always targets at the *current* state. Here in turn, a query anchor has to evaluate queries w.r.t. *any* time window containing the query anchor.

- Symmetry of time directions: For a query anchor, the notion of time splits into a time forward and backward component. Work on continuous queries does not consider this issue. We will show that it is possible to exploit this symmetry of both time directions in an implementation (Section 9.5).
- On the other hand, an issue not explicitly studied here, but addressed in related work on spatio-temporal databases is the mobility of objects/queries. Our specific work does not need to take it into account, since both data objects and query points are simply measured values, which cannot change in retrospect.

Overall, these differences highlight that our concept of query anchors is orthogonal to work on continuous k-NN queries.

9.4.2. MISE Framework

Our query anchor data structure provides an abstraction over the dynamics of marginal counts observed in a data stream. This abstraction allows to formulate the Kraskov estimation principle [KSG04, KL87] in the online context:

Definition 9.6

Mutual information estimate: Given a query anchor for Q_t and a subsequence $\mathcal{Q}_{t_2}^{t_1}$ with $t_1 \leq t \leq t_2$, the mutual information estimate is defined as follows:

$$\hat{I} = \psi(k) - \psi(MC_x + 1) - \psi(MC_y + 1) + \psi(N) \tag{9.3}$$

where ψ is the digamma function, N is the length of the subsequence that the query anchor has seen, and MC_x, MC_y are the marginal counts returned by $\text{QUERY}(t_1, t_2)$.

For the theoretical background behind Equation 9.3 we refer to [KSG04]. Briefly sketched, the idea of the Kraskov principle is to formalize the probability that there are $k-1$ objects with a distance lower then $kNND$ and $N-k-1$ objects with a distance exceeding $kNND$. This probability can then be plugged into the integral definition of entropy, leading to a mutual information estimate via Equation 9.2.

Obviously an estimation based on a single query anchor has a large statistical uncertainty. This statistical error can be reduced significantly by taking the average of the estimates from several query anchors. We will exploit this idea in our MISE framework, which we describe in the following.

The MISE framework (cf. Algorithm 3) provides two operations: (1) an INSERT operation to add data from the stream into the system, and (2) a QUERY operation which retrieves a mutual information value for a certain query window. Internally MISE stores a sample of

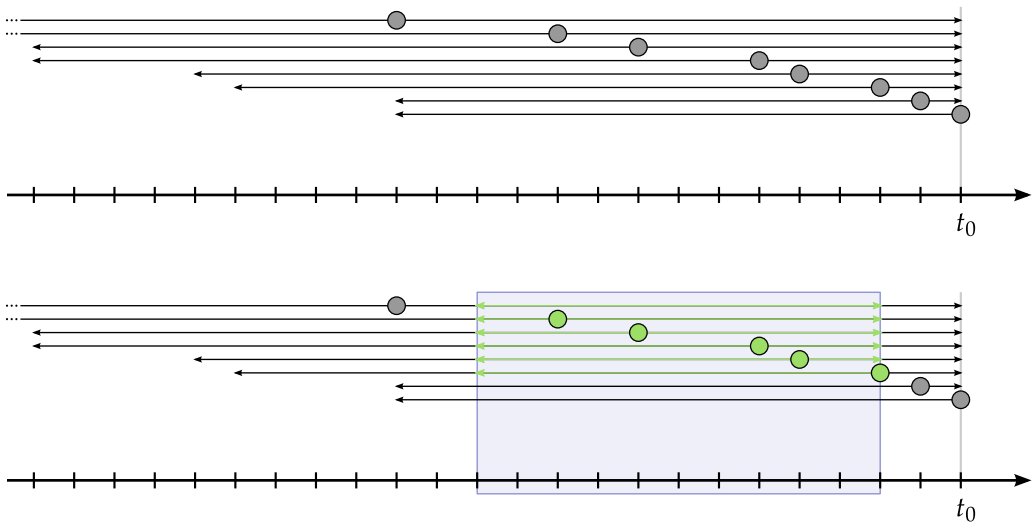


Figure 9.4.: Illustration of MISE

query anchors. This query anchor sample is modified by a SAMPLING function responsible for the deletion of query anchors. We will discuss the instantiation of this SAMPLING function in Section 9.4.3 and continue with an explanation of the INSERT and QUERY operations.

The INSERT operation first creates a new query anchor which corresponds to the data point Q_t just received. We then perform a forward and reverse initialization: The forward initialization performs an INSERTRIGHT operation on all existing query anchors for the new element Q_t. In other words, we show the new element to all existing query anchors in the current sample. The reverse initialization on the other hand adds data points corresponding to the existing anchors to the new anchor by using the INSERTLEFT operation. Finally, we add the query anchor to the sample and invoke a SAMPLING function (cf. Section 9.4.3). In general, the SAMPLING function modifies the current anchor sample by deleting certain anchors. An exemplary result after performing several insert operations is illustrated in Figure 9.4. Each circle corresponds to a query anchor, and the positioning shows the distribution of the query anchor sample over time. The black arrows indicate how much information was added to a query anchor by either INSERTLEFT or INSERTRIGHT. Note that in reverse time direction (INSERTLEFT) the data points are filled sparsely, i.e., not every data point covered by the arrow was actually inserted into the query anchor. In time forward direction on the other hand, all data points can be inserted. In terms of this illustration, the INSERT operation (1) adds a new query anchor at t_o, (2) extends the arrows of existing anchors by one step to the right, (3) extends the arrow of the new query anchor to the left, up to the position of the oldest query anchor, and (4) modifies the sample.

The QUERY operation first determines the query anchors that are contained in the query window. We query each anchor in the window for the marginal counts MC_x and MC_y, and the number of data points N that a query anchor has seen in the given window. Overall, we obtain a mutual information estimate from each anchor by Equation 9.3 and return the arithmetic sample mean of these estimates. Figure 9.4 illustrates the query operation: The blue shaded area corresponds to an exemplary query window. The green arrows show the query ranges that are used to obtain the marginal counts of the anchors within the query window. Note that for different query windows or a different query anchor distribution it is possible that the green arrows do not extend fully over the query window. In this case the query anchor has only seen a subsample of the whole window, which can still contribute valuable information to the estimation. The overall algorithmic structure of the MISE framework is shown in Algorithm 3.

Algorithm 3 MISE framework

```
1:  anchors ← {}
2:  procedure INSERT(Q_t)                          ▷ interface to add data
3:      a ← new query anchor at Q_t
4:      for all o ∈ anchors do
5:          o.INSERTRIGHT(Q_t)                     ▷ forward initialization
6:          a.INSERTLEFT(o)                        ▷ reverse initialization
7:      end for
8:      anchors ← anchors ∪ {a}
9:      anchors ← SAMPLING(anchors)                ▷ cf. Section 9.4.3
10: end procedure
11: function QUERY(t_1, t_2)                       ▷ interface to query MI
12:     inWindow ← {a ∈ anchors | t_1 ≤ a ≤ t_2}
13:     estimates ← ()                             ▷ empty sequence
14:     for all a ∈ inWindow do
15:         MC_x, MC_y, N ← a.QUERY(t_1, t_2)
16:         Î ← ψ(k) − ψ(MC_x + 1) − ψ(MC_y + 1) + ψ(N)
17:         estimates.APPEND(Î)
18:     end for
19:     return MEAN(estimates)
20: end function
```

Estimation Quality. An analytic analysis of the estimation variance and bias would require strong assumptions on the data. Since the overall mutual information estimate is based on taking a sample mean of individual estimates, the standard deviation of MISE can be expressed by the standard deviation of the mean: Assuming that the data distribution is static over the query window leads to a standard deviation of $\sigma = \sigma_{\hat{I}}/\sqrt{M}$, where M is the number of query anchors in the window and $\sigma_{\hat{I}}$ is the standard deviation

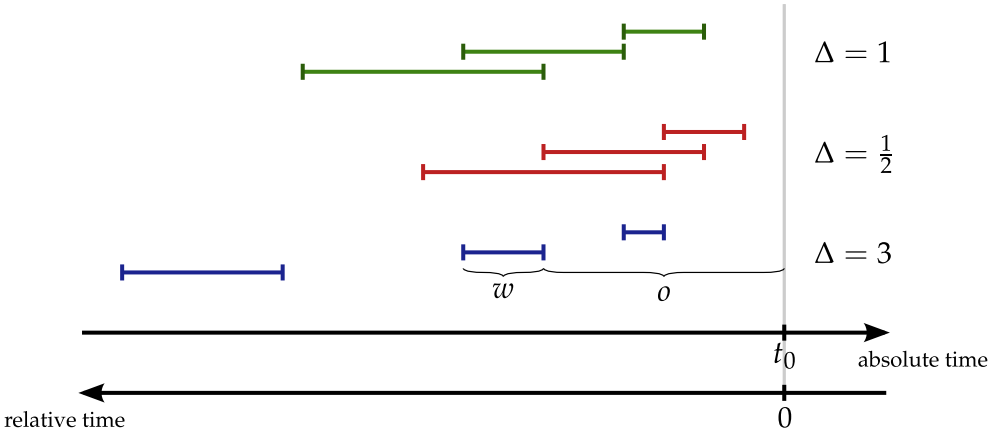


Figure 9.5.: Examples of multiscale equivalence classes

of the individual estimates. $\sigma_{\hat{I}}$ obviously depends on the distribution of the data. Instead of deriving $\sigma_{\hat{I}}$ only for specific distributions, we focus on a very broad empirical analysis of the estimation characteristics in Section 5.4, featuring a large number of real-world data streams.

9.4.3. Multiscale Sampling of Query Anchors

Our goal regarding the SAMPLING function is to exploit the multiscale nature of time. In general, any query window can be specified by its width w and the offset o, which denotes how much the query window is shifted into the past (cf. Figure 9.5). Intuitively, the motivation behind our multiscale sampling follows the general equivalence of time scales. Think of a query with a window size of 1 second shifted by 1 second into the past, a query with w = 1 hour and o = 1 hour, or even a query with w = 1 year and o = 1 year. Though the queries are defined over vastly different time scales, they are structurally equivalent. In many application a user might want to obtain answers of equal quality for these queries. Traditional sampling approaches like sliding window (SW) or reservoir sampling (RS) have significant issues with queries comprising multiple time scales. In general, we expect SW to fail for queries with a large window size in the distant past and RS to fail for queries with a very small window size in the most recent past. The key question is: How is it possible to answer these queries with equal quality? We will see that this simple requirement automatically generalizes for arbitrary o / w values, and that the *more detail on recent data* principle emerges naturally as a result. To formalize equivalence of time scales, we will denote the ratio of the query offset o to the window size w as a unit-free quantity $\Delta \equiv \frac{o}{w}$. Based on this quantity, we can partition the space of all possible queries into equivalence classes.

Definition 9.7

Multiscale Query Equivalence: We define the multiscale query equivalence relation ~ between queries A and B by: $A \sim B$ iff $\Delta_A = \Delta_B$.

We call the groupings formed by ~ multiscale equivalence classes. Based on the multiscale equivalence of queries, we formalize the key idea behind our approach to operate on various time scales:

Definition 9.8

Multiscale Sampling: A multiscale sampling is a sampling scheme which provides an equal expected number of sampling elements for all queries which belong to the same equivalence class.

Thus, for a multiscale sampling of query anchors the expected number of query anchors in a window is constant for all queries with the same Δ. Figure 9.5 shows examples of equivalent queries for different Δ values. When the multiscale property is fulfilled, the queries with the same color have the same expected number of query anchors.

We will now propose a novel sampling scheme that fulfills the multiscale property. More specifically, we derive the sample distribution that is required for multiscale sampling. To simplify the presentation, we temporarily assume a continuous time domain and switch to a discrete time in a second step. Since the sample distribution is only defined for $t < t_0$, we will change to a time domain that is relative to t_0 and extends into the past (cf. Figure 9.5). This allows us to use the notion of probability densities to express the expected number of anchors in a query window. We refer to the probability density of our query anchors as $f(t)$. Thus, we are looking for an $f(t)$ which is a probability density function that satisfies the multiscale property. The expected number of query anchors in a query window $[o, o + w]$ is equal to the integral $\int_o^{o+w} f(t)dt$ times the total number of query anchors. By using $o = w \cdot \Delta$, we can write the integral bounds as $[w\Delta, w(\Delta + 1)]$. Definition 9.4.3 requires that, for a fixed Δ, this integral (the expected number of sampling elements) is invariant of the time scale, i.e., it is constant for all w. Thus, $f(t)$ must fulfill:

$$\int_{w\Delta}^{w(\Delta+1)} f(t)dt \stackrel{!}{=} const \tag{9.4}$$

Lemma 1. *Sampling according to a reciprocal distribution $f(t) = \frac{C}{t}$ fulfills the multiscale property (with appropriate normalization C corresponding to a finite positive support).*

Proof. Equation 9.4 requires $\frac{d}{dw}\int_{w\Delta}^{w(\Delta+1)} f(t)dt \stackrel{!}{=} 0$. Differentiation under the integral according to the generalized Leibniz integral rule yields:

$$(\Delta + 1)\, f\left(w(\Delta + 1)\right) \stackrel{!}{=} \Delta\, f\left(w\Delta\right) \tag{9.5}$$

By plugging $f(t) = \frac{C}{t}$ into Equation 9.5, one can see that all Δ terms cancel each other out, i.e., the reciprocal distribution satisfies the multiscale property for any $\Delta > 0$. □

A striking property of Equation 9.5 is that it is an instance of the famous refinement equation in wavelet theory [SN96]. This is a result of the fact that the general idea of the equivalence of multiple scales plays a central role in wavelet theory as well.

We now turn to the question of how to transform this result to a discrete time domain. Obviously the support of a reciprocal distribution is only defined for $t > 0$ due to the singularity at $t = 0$. This directly reflects the general issue of estimation from a very small window size w: The smaller the window size, the larger the necessary density of sample points in order to maintain a sample of a fixed size. In a real-world system there commonly are domain specific constraints on the sampling frequency, i.e., the sampling resolution cannot be arbitrarily high. We deal with this issue by allowing a *saturation* of the discrete distribution in the region where the theoretically necessary sample density exceeds what is physically possible. To formalize the discretization, we will highlight the discretized time domain by using n as a counter of time steps in the past, starting with $n = 1$ as the most recent time point. We discretize the reciprocal distribution at these time points, each resulting in a probability P_n. Each P_n is equal to the probability that our sample contains the query anchor which is n time steps in the past:

$$P_n = \begin{cases} 1 & \text{if } n \leq \alpha \\ \frac{\alpha}{n} & \text{otherwise} \end{cases} \tag{9.6}$$

The resulting distribution[†] is illustrated in Figure 9.6. The (negative) x-axis corresponds to the discretized time steps n, and the y-axis shows the probabilities P_n. The parameter α controls the decay of the reciprocal distribution, and it will serve as the parameter to control the overall quality of the sampling. Equation 9.6 implies that we must keep the $\lfloor \alpha \rfloor$ most recent query anchors with a probability of 1, as a result of the shortage of available sampling points. For older query anchors with $n > \alpha$, the probability to have a certain query anchor in the sample follows the reciprocal function.

So far Equation 9.6 only specifies the necessary probabilities in the query anchor sample at a fixed time t_0. The essential question now becomes: How do we have to delete existing query anchors when going from one time step to the next in order to maintain an overall distribution according to Equation 9.6? This requires to convert the sampling probabilities P_n of Equation 9.6 into an incremental deletion scheme.

[†] The particular shape of this function – a piecewise composition of a reciprocal and a uniform function – makes a direct application of sampling schemes that specify weights in time forward direction [ES06, CSSX09] impossible.

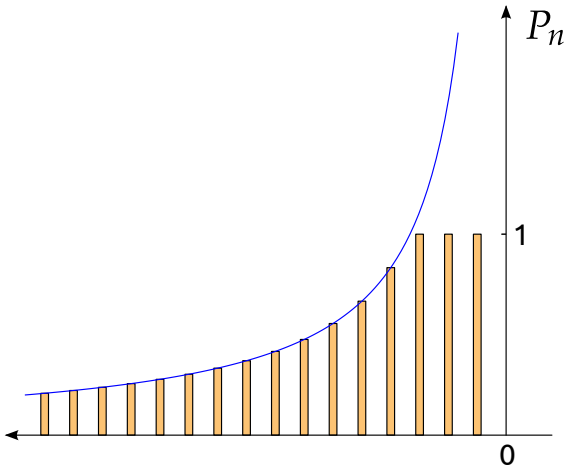


Figure 9.6.: Discretized distribution with saturation

Definition 9.9

SAMPLING **function:** Based on the notion of *stepwise sampling probabilities*

$$SP_n = \begin{cases} 1 & \text{if } n \leq \alpha \\ \frac{P_n}{P_{n-1}} & \text{otherwise} \end{cases} \tag{9.7}$$

we define the SAMPLING function as follows: We keep a query anchor of age n with a probability of SP_n.

This means that we generate a random value $rand \in [0, 1]$ for every anchor. If $rand < SP_n$ we keep the anchor; otherwise the anchor is deleted immediately.

Lemma 2. *Modifying the query anchor sample with the* SAMPLING *function results in the sampling probabilities P_n after each time step.*

Proof. In order to show this, we have to link the stepwise sampling probabilities to the probabilities P_n. Intuitively, the stepwise probabilities slide over a certain time point when time evolves. Since the decisions whether to keep a given query anchor are independent, the final probability that a query anchor still exists after k time steps is simply the product of the first k stepwise probabilities. Therefore we have to prove that the product of the stepwise probabilities indeed gives the desired probability P_n, i.e.,

$$P_n \stackrel{?}{=} \prod_{k=1}^{n} SP_k \tag{9.8}$$

Let n^* be the smallest $n > \alpha$. Obviously, Equation 9.8 is fulfilled for all $n < n^*$ since both sides are 1. For $n = n^*$ we have $P_{n-1} = 1$, and thus, $SP_n = P_n$, which again satisfies Equation 9.8 given $\prod_{k=1}^{n-1} SP_n = 1$. For $n > n^*$ we conclude by induction:

$$P_{n+1} \overset{!}{=} \prod_{k=1}^{n+1} SP_k = \prod_{k=1}^{n} SP_k \cdot SP_{n+1} = P_n \cdot \frac{P_{n+1}}{P_n} \qquad \square$$

Combining Lemmas 1 and 2 leads us to our final conclusion: It is possible to construct an iterative sampling scheme which always maintains the multiscale property in the query anchor sample.

9.5. Implementation and Analysis

In the following we will discuss implementation details of our approach. To ensure repeatability, we provide both pseudo-code and a ready-to-use executable on a supplementary website,[‡] and focus on the essentials in the following.

Query Anchor Complexity. An important aspect of our proposed approach is that it is possible to implement query anchors very efficiently. Our solution is based on the fact that a query anchor can treat the forward and backward time directions independently. For both time directions, we can use dynamic arrays to store the marginal points and changes to the set of the k nearest neighbors. The insertion of new data works as follows: insertRight first checks whether the new data point leads to a change of the k nearest neighbors in time forward direction. If this is the case, the new set of k nearest neighbors is appended to the dynamic array storing the neighborhood changes. Estimation theory shows that Kraskov estimation in general requires a very low k settings (i.e., $k \leq 4$, cf. Section 5.4), thus, the $O(\log k)$ complexity of the set operations are negligible. Next, insertRight checks whether the new data point is a marginal point in either X or Y, and appends the point to the respective dynamic arrays. insertLeft is implemented accordingly, operating in time backward direction. Overall, an insert operation comes down to extending the dynamic arrays, i.e., the amortized insert complexity is $O(1)$.

The query operation has two substeps: (1) reconstruction of the proper $kNND$ for the given query window and (2) counting of marginal points. Step (1) can be implemented efficiently, since the two dynamic arrays storing the changes of the k nearest neighbor sets in both time directions are sorted by construction. This allows to perform a binary search to retrieve the k nearest neighbor sets in each time direction. To get the $kNND$ over the whole query window, it is simply possible to create the union of both sets and determine the k-th element. Step (2) counts the marginal points which are within the window boundaries and have a marginal distance lower than the just determined $kNND$.

[‡] http://www.ipd.kit.edu/~muellere/MISE/

Due to the intrinsic sorting of our two-sided insert scheme, a binary search can again solve this efficiently.

Regarding memory complexity, a query anchor obviously requires $O(M)$, where M is the number of marginal points. The question is how the number of marginal points M grows over time. Unfortunately, a respective formal analysis would require assumptions regarding both the data distribution itself and how it changes over time. Even under the assumption of a static data distribution, there is no general result. However, it is possible to derive the general spectrum of possible growth rates. This follows from the findings of extreme value theory [LLR83], which we explain in the following. As illustrated in Figure 9.1 and 9.2, there are two opposing effects: On the one hand, if the size of a slice was fixed, the number of marginal points would simply increase linearly, assuming a static data distribution. On the other hand, the width of the slice can only decrease monotonically over time. Thus, the question is how fast the k-NN distance decreases over time. In general the distance distribution of each query anchor is highly individual. Determining the minimum (or the k-th smallest element) of a sample drawn from this distance distribution is a standard problem of extreme value theory [LLR83]. Since the metric is bounded by the minimum distance of zero, the domain of attraction is limited to a specific category, the Type III or Weibull family. However, the convergence rate of the minimum does not have a general result in this category. Hence, the overall growth rate of our marginal count can vary; there are the following cases:

- The minimum distance may decrease $\propto \frac{1}{N}$. For instance, this is the case if the distance distribution is an exponential or uniform distribution [LLR83]. In this case the complexity of marginal counts is $O(1)$, due to the rapid decrease of the slice width.
- For some data distributions the growth of marginal counts is $O(N^{\alpha})$ with $\alpha < 1$. For example, when a query anchor is placed within a uniform distribution, the growth is $O(N^{\frac{1}{2}})$. We derive this result in the following.
- For (rare) outlier objects the distance distribution mainly produces large distances, and therefore, the rate of convergence of the minimum is low. This yields the worst case complexity of $O(N)$.

In light of these findings, our expectation for the general case of arbitrary data distributions which may change over time is to obtain a mixture of these three cases. Therefore, we perform a thorough empirical analysis of the growth rate in our evaluation (cf. Section 9.6.4).

Case Study: Uniform Distribution.

Lemma 3. *Let $k = 1$ and let Q be a query anchor centered in an independent uniform distribution. The number of points M that lie in a slice of Q grows according to $\frac{\sqrt{\pi}}{2} n^{\frac{1}{2}} + O(n^{-\frac{1}{2}})$.*

Proof. Let X and Y be independent one dimensional uniform distributions on $[0, 1]$ (w.l.o.g.). Placing the reference point in the middle of the uniform distribution results in one dimensional distance distributions $D_X = |X - 0.5|$ and $D_Y = |Y - 0.5|$, which are

again uniform distributions but on $[0, 0.5]$. Let $D = max(D_X, D_Y)$. The CDF of D, i.e. $F_D(d)$, is given by:

$$\begin{aligned} F_D(d) &\equiv P(D \leq d) = P(X \leq d \cap Y \leq d) \\ &= P(X \leq d)P(Y \leq d) = F_X(d)F_Y(d) \\ &= \begin{cases} 0, & \text{if } d < 0 \\ 4d^2, & \text{if } 0 \leq d \leq 0.5 \\ 1, & \text{if } d > 0.5 \end{cases} \end{aligned}$$

where $F_X(d)$ and $F_Y(d)$ are the (linear) CDFs of X and Y. The PDF of D is therefore given by:

$$f_D(d) = \begin{cases} 8d, & \text{if } 0 \leq d \leq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

Let R_n be the size of the box after the query anchor has seen n data points. This means that $R_n = min(D_1, \ldots, D_n)$, where each D_i is distributed according to $f_D(d)$. The CDF of R is given by:

$$\begin{aligned} F_{R_n}(r) &= 1 - P(R_n > r) \\ &= 1 - P(D_1 > r \cap \ldots \cap D_n > r) \\ &= 1 - \prod_{i=1}^{n} P(D_i > r) = 1 - \prod_{i=1}^{n} (1 - F_{D_i}(r)) \\ &= 1 - (1 - F_D(r))^n = 1 - (1 - 4r^2)^n \end{aligned}$$

which gives the following PDF for R_n:

$$f_{R_n}(r) = 8nr(1 - 4r^2)^{n-1}$$

We can now obtain P, the number of points in the slice in dependence of R_n. In order to do so, we have to divide the width of the slice ($2R_n$) by the two dimensional area, which contains the remaining uniform distribution (the unit square minus the volume of the box, i.e., $1 - 4R_n^2$). Thus:

$$P_n = \frac{2R_n}{1 - 4R_n^2}(n - 1)$$

We can get the expectation value $E[P_n]$ by expressing the transformation from R_n to P_n via $g(r) = \frac{2r}{1-4r^2}(n-1)$ and integrating $g(r)$ weighted by $f_{R_n}(r)$, i.e.:

$$\begin{aligned} E[P_n] &= \int_{-\infty}^{+\infty} g(r) f_{R_n}(r)\, dr \\ &= \int_0^{0.5} \frac{2r}{1 - 4r^2}(n-1)\left(8nr(1 - 4r^2)^{n-1}\right) dr \\ &= \int_0^{0.5} 16r^2 n(n-1)(1 - 4r^2)^{n-2}\, dr \end{aligned}$$

$$
\begin{aligned}
&= \int_0^{0.5} 16r^2 n(n-1) \sum_{k=0}^{n-2} \binom{n-2}{k} 1^{n-2-k} (-4r^2)^k \, dr \\
&= 16n(n-1) \sum_{k=0}^{n-2} \binom{n-2}{k} (-4)^k \frac{\left(\frac{1}{2}\right)^{2k+3}}{2k+3} \, dr \\
&= \frac{\sqrt{\pi} n!}{2(n-\frac{1}{2})!}
\end{aligned} \tag{9.9}
$$

Expansion of this expression leads to:

$$
E[P_n] = \frac{\sqrt{\pi}}{2} n^{\frac{1}{2}} + \frac{\sqrt{\pi}}{16} n^{-\frac{1}{2}} + \frac{\sqrt{\pi}}{256} n^{-\frac{3}{2}} + O(n^{-2})
$$

□

MISE Complexity. Naturally, the most important complexity factor of the MISE framework is the size S of the query anchor sample. The sample size S not only determines the overall memory consumption; it also defines the complexity of the INSERT operation since the INSERT operation has to connect each incoming data sample to the existing query anchors and vice-versa. Therefore the insert complexity is $O(S)$. We can express the expectation value of S after processing T data points as follows:

$$
E[S] = \lfloor \alpha \rfloor + \alpha \sum_{k=\lfloor \alpha \rfloor + 1}^{T} \frac{1}{k} = \lfloor \alpha \rfloor + \alpha \left(H_T - H_{\lfloor \alpha \rfloor} \right) \tag{9.10}
$$

where H_i is the i-th harmonic number. Asymptotic expansion of H_T reveals a complexity of $O(\log T)$. We use this result to construct two different versions of our algorithm. The first version MISE$_D$ works with this slowly growing dynamic query anchor sample. For a second version MISE$_F$, we fix the sample size S and instead operate with slow changes of α over time. This means we modify α in each step by solving Equation 9.10. This has to be done numerically since the equation has no analytic solutions. In Figure 9.6 this would correspond to a slight change of the decay rate. Obviously each modification to α introduces a small error since the current query anchors in the sample have been sampled with a probability that has been slightly too large. To account for the accumulation of these slight errors, we delete query anchors with a probability equal to the ratio of the P_n values calculated once with the old and once with the new α. This exactly corrects the error and maintains a proper reciprocal distribution over time. Overall, the two versions of MISE have insert complexities of $O(\log T)$ for MISE$_D$ and $O(1)$ for MISE$_F$.

9.6. Experiments

The focus of our experimental evaluation is to analyze MISE regarding both performance and estimation quality. In particular, we will analyze how MISE performs overall in a

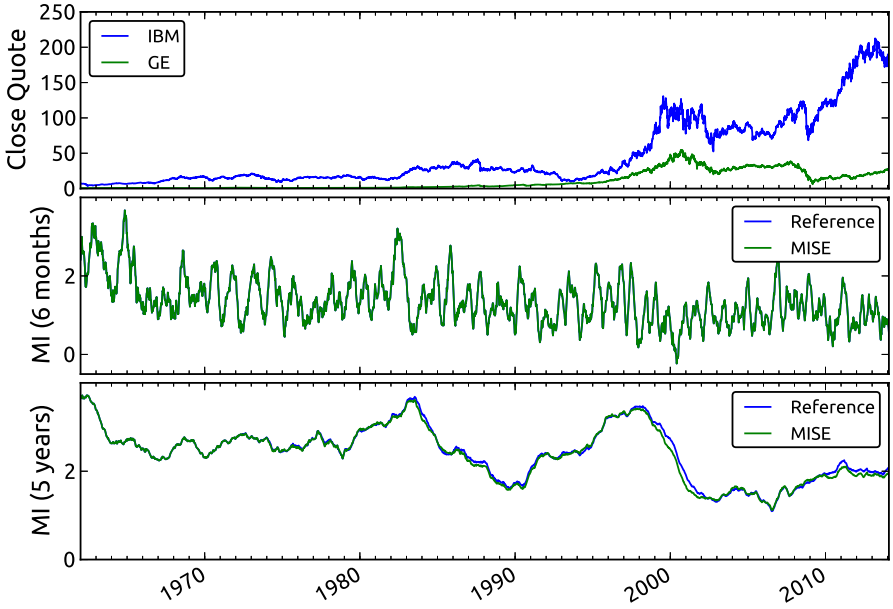


Figure 9.7: Stock exchange example

typical online setting (Section 9.6.1), how we can match the insert frequency to the stream frequency (Section 9.6.2), and focus on estimation quality in Section 9.6.3. We are not aware of any direct competitor that supports online mutual information queries. Therefore, we compare our approach to a static Kraskov estimation, which we allowed to use a (theoretically) infinite data reservoir. By using the same estimation principle based on an infinite reservoir as a ground truth, we can focus on the effects introduced by our finite summarization of the stream. This frees us from reevaluating the properties of Kraskov estimation in general, and we refer to existing studies [KBG+07, WWL09, KA13, KSG04] for details. The guidelines on choosing parameter k obtained in these studies directly apply in our case as well: The best trade-off between the statistical and the systematic estimation error is typically in the range of very low k values (e.g. $k \leq 4$). Since our evaluation scheme measures the relative estimation error we focus on the case that is most challenging: We use $k = 1$ in all our experiments, which maximizes the statistical error of Kraskov estimation and therefore maximizes the effect of using a finite reservoir in Mise.

A relative evaluation also allows us to run our experiments on a broad range of data streams, including a large number of real-world datasets. These data streams contain natural fluctuations of mutual information over time. Figure 9.7 shows an example of such dynamics found in our real world data: The top plot shows the raw time series themselves; in the example the quotes of the IBM and GE stocks. The middle plot shows mutual information measured using a 6 month sliding window. Compared to the bottom plot, which uses a 5 year sliding window, we can see that mutual information clearly shows different changes over these two different time scales. Such a "running mutual

information estimate" also gives the first impression of the potential of MISE: We can see that our estimation of MISE and the reference implementation give almost identical estimation results, i.e., estimation based on the finite summarization of MISE shows no significant difference to estimation from an infinite reservoir. However, as a result of the online processing in MISE, the total time to generate the graphs in Figure 9.7 were 3.8 minutes for MISE and 112.1 minutes for the reference implementation. In the following we will quantify these performance improvements systematically.

Experimental Setup. We have conducted all experiments on an Ubuntu 12.04 system running on an Intel® i3-550 processor with 8 GB RAM. We have implemented MISE in Scala 2.10 using Oracle JVM 7 as runtime environment.

9.6.1. Overall Performance

Our reference implementation of static mutual information estimation on a data stream works as follows: The insert operation simply appends a data sample to a theoretically infinite reservoir, while the query operation performs Kraskov estimation on the specified query window using the infinite reservoir. Since this reference approach provides no means of query precomputation, there is obviously a pronounced imbalance between the extremely cheap insert operation and the high complexity of the query. Thus, when comparing MISE to this reference implementation the crucial question is how the number of inserts compares to the number of queries. We express the ratio of queries-to-inserts by $QIR = {}^{\#\text{queries}}/_{\#\text{inserts}}$. Obviously, when there are no queries at all ($QIR = 0$), all query precomputations of MISE are futile and there is nothing to speed up. On the other hand, when there is a large number of queries compared to the number of inserts ($QIR \gg 1$), the benefits of MISE's precomputations can be made arbitrarily high. Therefore, we specifically analyze low QIR values to determine the point where the benefit of MISE begins.

To measure the speed-up we determine the ratio of the total runtimes for MISE and the reference implementation of calculating a "running mutual information estimate" (like shown in Figure 9.7). This running estimate is performed by inserting and querying the stream with a specific QIR ratio, e.g., for $QIR = 0.1$ we perform a query after every 10 inserts. Regarding the time offset of the queries we set $o = 0$. This means that the queries operate in the region of highest query anchor density, and thus, performance of MISE is worst. The data stream was sampled from a Gaussian distribution ($\rho = 0.1$) with a length of 100000. We started the measurement of the total runtime after the number of processed samples exceeded both the window size and the reservoir size in order to exclude warm-up artifacts.

The results of this experiment are shown in Figure 9.8. The main factors determining the speed-up are the query window size and the size of the reservoir used by MISE. The latter is determined either by α or S for the dynamic or fixed versions. Due to the more

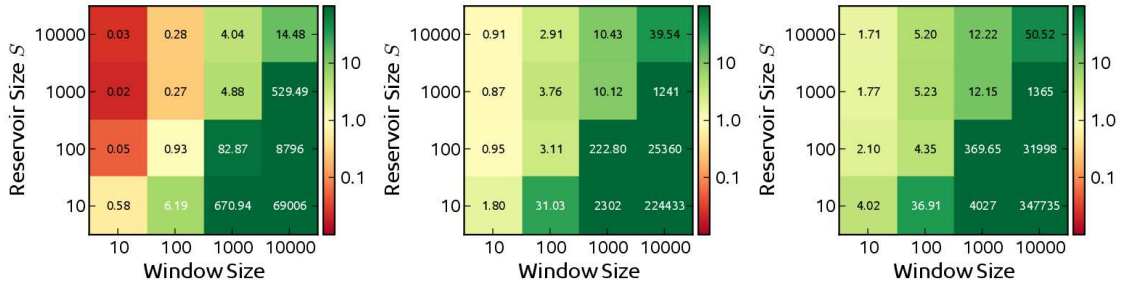


Figure 9.8.: Speed-up of accumulated runtime for query-to-insert-ratios of 0.01 (left), 0.1 (middle), 1.0 (right).

intuitive interpretation of the fixed reservoir size S we focus on this variant. Even though we focus on small QIR ratios, we can see that Mise can lead to drastic speed-ups. We visualize the speed-up threshold of 1.0 where usage of Mise starts to make sense by a yellow color; green and red indicate faster and slower runtimes for Mise respectively. It is interesting to see that a speed-up is even possible for the very low $QIR = 0.01$, where in fact 99% of the precomputations in the inserts were in vain. Overall we can conclude that there is a large potential for speed-ups as a result of our precomputations. Obviously this is especially pronounced for applications where both the window size and offset are free parameters for each query, and thus, having more queries than inserts is usual.

9.6.2. Scaling with Stream Frequency

The results of the previous experiment can also be interpreted as follows: Since Mise performs parts of the necessary query computations while processing the stream itself, it is possible to tune Mise such that its insert speed perfectly matches the frequency of the stream. This would mean that Mise performs just as much precomputations as possible to keep up with the stream and leads to maximization of the query quality for the given stream frequency (cf. Section 9.6.3). Thus the essential question becomes: How does the reservoir size of Mise influence the processing speed of the stream? We evaluate this question for both Mise versions, i.e., we analyze the insert speed in dependence of α and S for the dynamic and fixed reservoir versions respectively. Intuitively a higher S or α means higher estimation quality, but a slower insert processing. A user typically might want to set S or α to the largest possible value that still allows to process the given stream frequency.

Figure 9.9 shows a measurement of the insert times for different α or S values. It shows how the runtime of a single insert (y-axis) changes with the stream length (x-axis). We obtain the runtime of a single insert from the runtime of performing 5000 inserts in a batch. Again we sampled the data stream from a Gaussian distribution with $\rho = 0.1$. For the fixed Mise version we can see that the runtime of a single insert indeed becomes

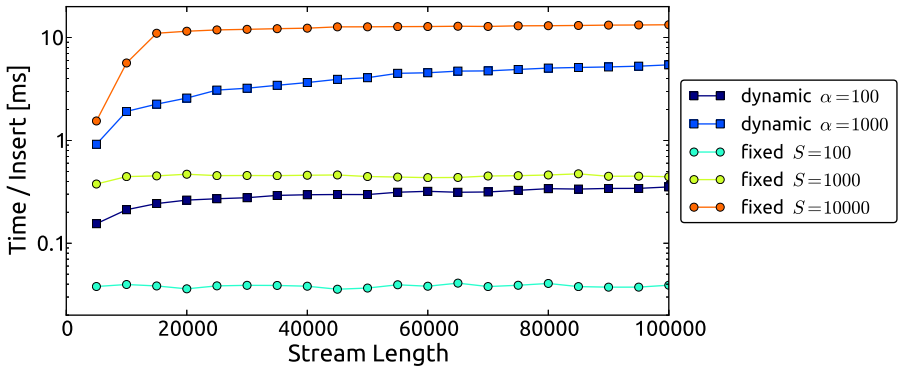


Figure 9.9.: Insert processing times

constant once the stream has reached a length corresponding to the fixed reservoir sizes S=100, 1000, 10000. For the dynamic version, the insert time slowly increases over time due to the logarithmic growth of the reservoir. We can see that for typical sizes of the reservoir the corresponding stream processing frequency is in the order of ~100 Hz (for S = 10000) up to ~20 kHz (for S = 100). Thus, despite performing query precomputation while processing the stream, it is possible to operate on very fast streams with sampling periods in the order of a millisecond. Furthermore, there is no dependence of insert performance on the stream length for the fixed Mise version. Thus, there is no degradation over time, which is a key property of efficient stream processing [BBD+02].

9.6.3. Quality

We now want to turn to the question how estimation from a limited reservoir affects the estimation quality. Therefore, we compare Mise to a variant of Kraskov estimation which also operates on a limited data reservoir. We implemented the limited data reservoir based on the two most prominent sampling approaches: Traditional reservoir sampling (RS), and sampling based on a sliding window (SW). We use static Kraskov estimation from an unlimited data reservoir as ground truth for the quality assessment. To facilitate the comparison we focus on the Mise variant with a fixed reservoir size. This allows us to use exactly the same reservoir size for RS, SW, and Mise. To pay attention to the challenge of a stream length being much larger than the reservoir size, we have used a reservoir size of S = 100 in the following experiment.

Data. We compiled a set of 26 data streams from various different sources. Our goal was to obtain a very large diversity of different streams, i.e., diversity in distributions and dynamics. Therefore the set contains various streams from different real world datasets plus a small number of synthetic streams. This includes streams of IMU sensors (various combinations of gyrometer, accelerometer, magnetometer streams), climate streams, smart meter streams, stock streams, and electrocardiogram measurements. All features of

the data streams are continuous variables, with a floating point precision between 4 and 10 decimal digits. Table 9.10 shows a summary of all data streams. Preliminary results on quality did not show a strong dependence on the individual data streams. Therefore, we present the aggregated quality over all streams in the following, and provide results on the individual datasets in Appendix A. This means that we calculate the quality measures discussed below for each query individually and aggregate by taking the average of these measures for all queries obtained from all data streams.

In order to generate a general result on the estimation quality it is necessary to analyze a broad range of different data distributions. In our opinion it is not satisfying to limit the analysis to certain synthetic distributions like a Gaussian or an exponential distribution: Sampling a data stream from a distribution with fixed parameters fails to evaluate the major challenge of a data stream, i.e., the dynamics or changes of a distribution over time. A possible solution would be to introduce some dynamics in the distribution parameters or to generate a mixture of several components and modify the set of components over time. For the sake of diversity, we will include a small number of such synthetic data streams in our stream collection. However, our main focus is on real world data streams, where the changes of distributions occur naturally.

To generate a real world data stream, we use databases from various different domains with temporal characteristic. Each pair of features in such a database can form one data stream. Some of the databases contain more than two features, and the total number of feature pairs, i.e possible data streams, can be very large. For instance in the case of stock exchange data, we extracted 3240 time series of daily stock quotes traded at NYSE. This results in more the 5 million possible data streams (neglecting the possibility that not all stock pairs were traded over the same period of time). Overall, we want to generate an aggregate of the quality which is not biased towards specific data stream types. Therefore we limit the total number of data streams extracted from each database to just five pairs. This also keeps the total number of data streams in a manageable order. In general we selected the five streams randomly from the set of all possible pairs. An exception to this rule was made for datasets with a varying co-occurrence of the individual time series. For instance in the case of stock data, many companies are created or closed over the total time period. Picking a random stock pair may a short or even vanishing overlap in time. Since we want to perform a large number of queries with various window sizes we are in general interesting in long data streams. Therefore, we sorted the pairs according to their length of co-occurrence and randomly selected five streams from the top 20.

In total our collection of data streams consists of 26 individual streams. We will briefly describe the different data streams of our collection in the following:

- Data Streams from an inertial measurement unit (IMU): We extracted IMU data streams from the PAMAP project [Pam], using the dataset PAMAP2. The data was collected as follows: A human subject performed different physical activities while being equipped with three Colibri wireless IMUs, with a sampling frequency

of 100 Hz. The three IMUs were attached to the wrist, the chest, and the ankle. The sensors record the typical features of an IMU measurement, i.e., acceleration, gyroscope, and magnetometer data. We picked five random combinations which cover variations of combining the same vs. different IMUs and axis-aligned vs. orthogonal vector components.

- Smart Meter: We extracted smart meter data streams from energy consumption measurings of 63 different KIT buildings. Each extracted time series corresponds to the smart meter reading of one particular building. The data covers a time span of 3 years using a measurement interval of one day. In some cases the smart meter device did show short periods of failure, which was reflected in missing values. We took care of the filtering of missing values and selected five random building pairs.
- Stock Data: We extracted the stock data streams from the web service Stooq [Sto]. The streams are based on the daily NYSE close quotes of a selection of stock pairs. We only selected pairs with a long stock market history in order to maximize the length of the resulting pairs (increasing the number of possible test queries).
- ECG Stream: This stream is based on BIDMC Congestive Heart Failure Database ECG and which uses a pair of ECG sensors for each patient.
- Climate Stream: The climate data stream is based on daily measurements of temperature vs air pressure in Karlsruhe provided by German Meteorological Service (Deutscher Wetterdienst).
- Static Gaussian: These are synthetic streams sampled from a Gaussian distribution with covariance matrices corresponding to different correlation coefficients *rho*.
- Dynamic Gaussian Mixtures: These synthetic data streams were created by sampling from a mixture of Gaussians. We varies the mixture of Gaussian itself by the following mechanism: At each time point we randomly delete an existing mixture component with a probability of p and/or create a new component also with probability p. We force a lower limit of at least 1 component and an upper limit of 10 components, i.e., we do not perform a deletion/addition of a component if it would violate these limits. The mean and elements of the covariance matrix for each component contains are randomly sampled (uniformly in [0,1]). In order to obtain streams of varying dynamic, we varied the change probability p from 1% to 10% in the following five streams. Dynamic Uniform Mixtures These synthetic data follow the mixture principle of the Gaussian mixture streams, but using uniform components instead. The boundaries of the uniform components are themselves sampled from a uniform distribution in [0,1].

Queries. The queries we perform on our data streams range from a window size of 10 up to 1000. We perform these queries in appropriate steps that avoid an overlap of the query windows to ensure independent query results. We use three different Δ values (0, 1.0, 10.0) to determine the offset of the query window. Using $\Delta = 0$ means that we use an

Stream description	Length	# streams used
Pamap (IMU data)	198000	5
Climate data (temperature vs. air pressure)	21124	1
Smart Meter	17568	5
Stock time series	11122	5
Congestive Heart (two ECG measurements)	300000	1
Synth: Static Gaussian $\rho = 0$	∞	1
Synth: Static Gaussian $\rho = 0.95$	∞	1
Synth: Static random mixture of uniform distributions	∞	2
Synth: Dynamic random Gaussian mixture	∞	5

Table 9.10.: Set of data streams

offset $o = 0$ (i.e., we deliberately include the most favorable case for SW), while o follows the multiscale principle in the non-zero cases.

Quality Measure. A first question when performing a certain query on a system with a limited reservoir is whether the system actually has information available for the given window boundaries. Therefore, our first quality measure simply is the percentage of "successful" queries defined as: A query is successful if the query window contains at least a single element, allowing to compute a result. In case the system can answer a query, we are interested in how the limited reservoir influences the estimation in both bias and variance. Therefore, we use the two quality measures $(\hat{I} - I_{Ref})$ and $\sigma_{\hat{I}}$. Here $\hat{I}$ refers to estimation from the limited reservoir, while I_{Ref} is the ground truth obtained from the infinite reservoir; $\sigma_{\hat{I}}$ is the sample standard deviation.

Results. The results of our quality experiment over all data streams are shown in Figure 9.11. Regarding the success rate of queries we can clearly see the advantage of the MISE sampling: For $\Delta = 0$ and $\Delta = 1.0$, the success rate is 100%. The result shows that the success rate does not depend on the window size. It rather is constant for a given family of queries with a fixed Δ. RS in contrast never achieves a 100% success rate. By the nature of RS, we can clearly see the poor performance for small window sizes (e.g., low success rate, large bias). On the other hand for sliding window sampling, we obtain poor performance for large windows, visible for instance by the sudden drop of the success rate as soon as the offset is larger than the fixed window of size 100. A query with $\Delta = 10$ simply has always been too far into the past and could never be answered. In the second row of Figure 9.11, we can see that all approaches show a small negative bias, which is a general issue when estimating from very little data. We can see that MISE shows much better bias and variance (third row) compared to Kraskov estimation from limited reservoirs. This is caused by the more flexible placement of query anchors over time and the added information that is used as a result of the online processing. Furthermore the dependence on the query window size is much lower compared to RS or SW sampling. For SW sampling the bias and variance are obviously zero as long as the query window is fully covered by the sampling window. However, we can see that, even in the favorable case of a zero query offset ($\Delta = 0$), estimation quality quickly degrades as soon as the query

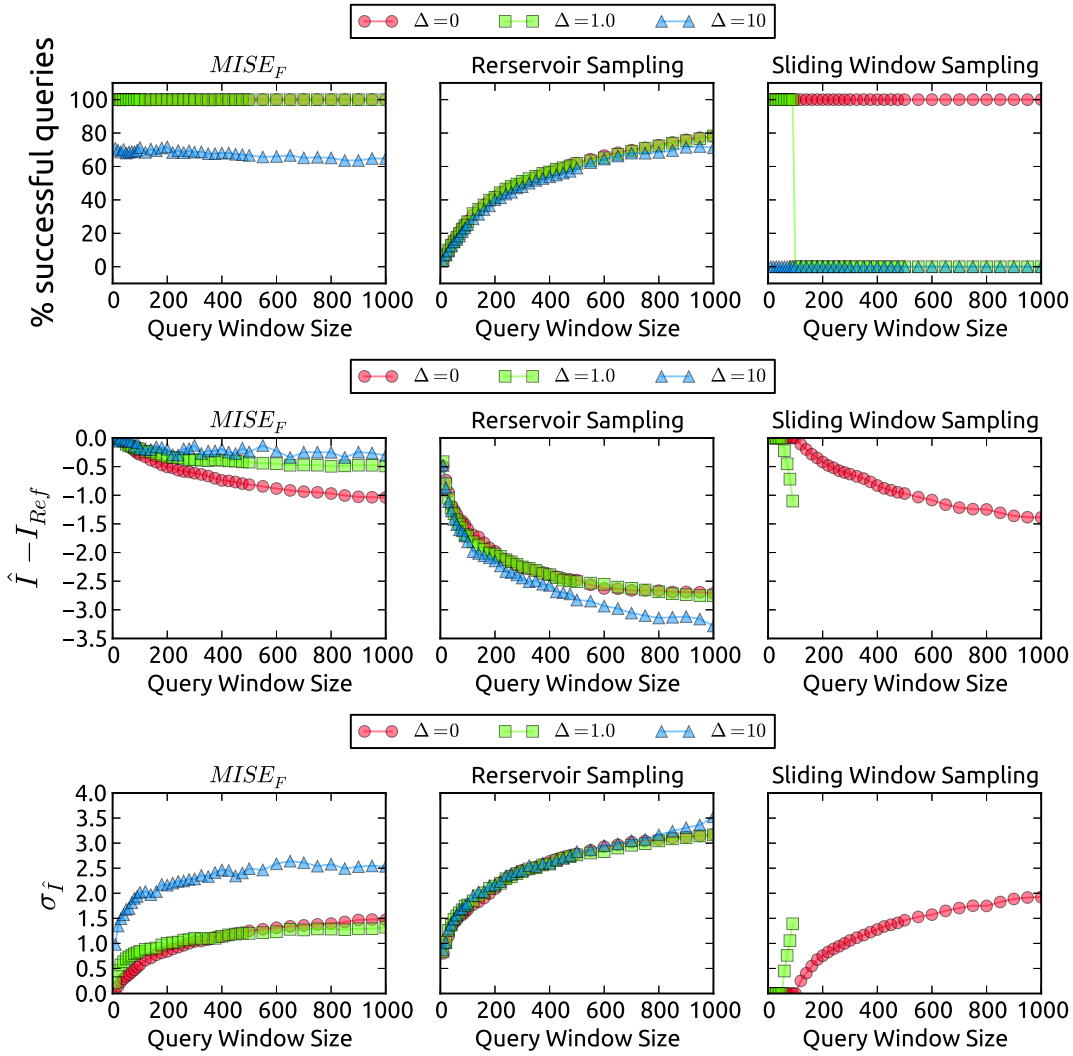


Figure 9.11.: Overall quality results on all data streams

window size exceeds the one of the sampling window. Overall we can conclude: RS and SW fail either for small or large window sizes respectively; MISE achieves the overall best results, and is most stable w.r.t. shifting a query into the past, as a result of featuring a multiscale sampling scheme.

In a final experiment we want to show that the estimation bias and variance of MISE can be reduced arbitrarily by increasing the reservoir size. For this experiment we use the dynamic version and vary α from 200 to 1000. The queries have $\Delta = 1.0$. See Figure 9.12 for the results. We can see that the estimation variance becomes almost independent of the window size in the range where query anchor saturation does no longer occur. The takeaway is that the reservoir size gives very fine control over the overall estimation quality. Combined with the results on insert speed this means that bias-free estimates with

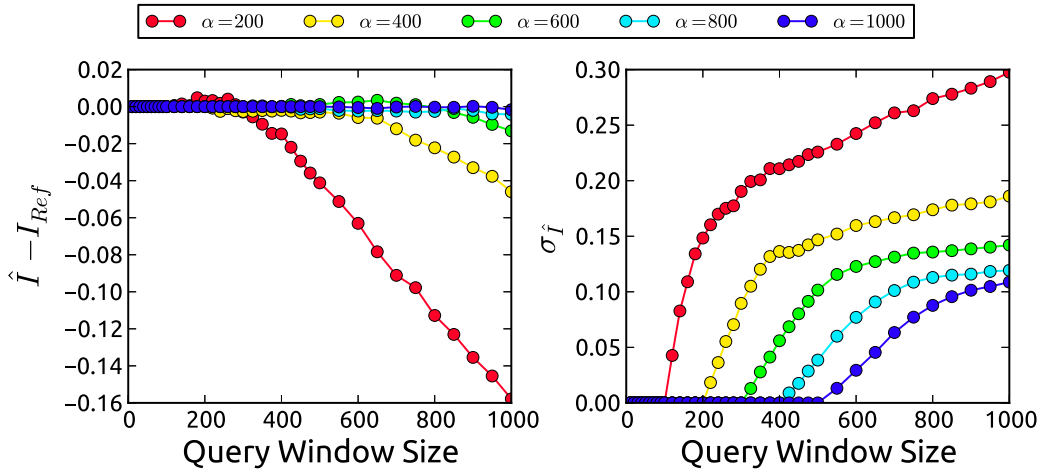


Figure 9.12.: Quality with different reservoir sizes

very low variance are possible without difficulties, while maintaining insert frequencies in the range of 1000 Hz.

9.6.4. Growth Rate of Marginal Points

We conclude our experiments with an empiric evaluation of the question of how the number of marginal points evolves over time. In the following experiment we determine the empirical growth rate for each of our data streams (cf. Table 9.10) individually. For each data stream, we randomly pick 1000 data points from the first half of the data stream (uniformly distributed) as test query anchor. This procedure ensures that our test query anchors have a random location within the data distributions. Next we process the data stream, i.e., we insert subsequent data points to our test query anchors. Finally we query all 1000 query anchors for the marginal counts in time forward direction with a varying query window size. Figure 9.13 shows the average marginal counts depending on the query window size, which corresponds to the number n of inserted data points. As a reference we have added the theoretical growth rate $\frac{\sqrt{\pi}}{2} n^{\frac{1}{2}}$ for the specific case of a uniform distribution analyzed in Section 9.5.

Overall, the result in Figure 9.13 reveal an interesting finding: For most data streams we observe a growth rate of approximately $n^{\sim\frac{1}{2}}$. This result makes sense considering that the overall spectrum of growth rates for each individual anchor ranges from constant to linear depending on its position in the data distribution (cf. Section 9.5). Apparently, the averaging of all the individual growth rates seems to yield a similar rate to the one obtained formally for the uniform distribution (Section 9.5). Another remarkable result is the absolute value of the marginal counts itself: A query anchor with an age of 20000 time units has to store less than 150 marginal points on average. Thus, the ratio of stored

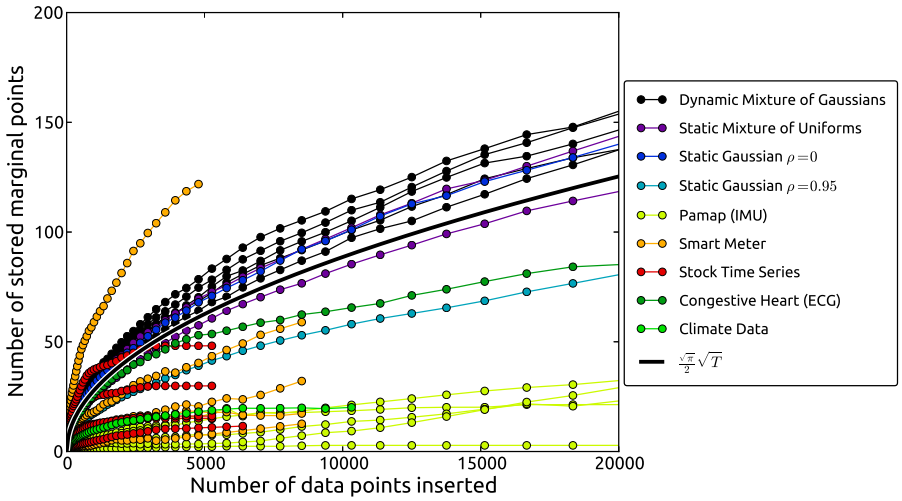


Figure 9.13.: Average marginal counts that have to be stored over time

marginal points vs inserted points is ~0.75% after 20000 time units, or ~0.09% after 1 million steps. This strong compression ratio shows that it is cheap to maintain "old" query anchors, which can explain the very good overall performance of MISE observed in the previous experiments.

9.7. Conclusions

In this chapter we have proposed a framework that allows a user or data mining algorithm to estimate mutual information on a data stream in arbitrary query windows. To our knowledge, it is the first such estimation technique that incorporates a summarization allowing online query precomputation. Furthermore, we have proposed a novel sampling scheme which provides a solution to the infinite nature of the stream while maintaining information equally over multiple time scales. Given these properties the experiments show that our approach clearly outperforms traditional approaches in the online context.

Regarding the detection of subspace outliers, the technique can be used to detect high contrast bivariate subspaces. Thus, it is a first step towards solving the challenges of subspace outliers on data streams as well. However, due to the high complexity of data streams, a complete solution of the problem requires future research. One important extension will be to move from bivariate subspaces to the general case of a multivariate analysis. Another direction for future research is to address the challenges of temporal contexts, for instance in a fully automatized detection of relevant temporal contexts similar to the detection of relevant attribute contexts.

Part IV.

Conclusions

10. Conclusions

In summary, this thesis creates a fusion of the topics outlier mining and attribute relationship analysis. In the following we want to briefly summarize our main contributions in these two domains. Overall, the techniques developed in this work have been well received by the research community, resulting in several publications which are directly based on ideas of this thesis. Therefore, we will discuss a few examples of follow-up work in Section 10.2. We conclude by giving an outlook of possible future research directions that may develop from the foundations presented here.

10.1. Summary

At the beginning of this thesis, we have summarized traditional outlier models (Chapter 2), followed by a discussion of the major open issue in outlier mining: Outliers hidden in subspaces (Chapter 3). Our analysis of such subspace outliers has revealed three essential properties: (1) Subspace outliers suffer from a *high-dimensional invisibility*. Therefore, traditional full-space methods fail to detect them reliably. (2) The problem of detecting such outliers in their respective deviating contexts is inherently a *multivariate* problem, i.e., subspace outliers can occur in subspaces of arbitrary dimensionality. Thus, a restriction to analyzing e.g. only pairwise attribute relations is not sufficient to solve the problem. (3) Detecting subspace outliers must be tackled from a *multi-view* perspective, i.e., all anomalies can have highly individual deviating contexts, which cannot be simultaneously captured by any global view on the data. In other problem domains, similar observations have lead to the development of subspace search techniques to address these challenges of high-dimensional data. However, all existing subspace search techniques (Chapter 4) focus on clustering only. This highlights the demand for novel subspace search techniques with the specific goal of outlier mining, as developed in this thesis.

In Chapter 5, we have presented HiCS, the first subspace search approach for outlier mining. Based on a formalization of the properties of outliers hidden in subspaces, we have derived a simple necessary condition for the existence of subspace outliers: The deviation from statistical independence of the attributes of a subspace. By introducing the so-called subspace contrast, we have provided a novel subspace quality measure, which allows for a highly efficient and robust evaluation of the relevance of subspaces. Technically, the subspace contrast is based on the empirical (i.e., sample-based) comparison of marginal

and conditional distributions. A thorough analysis and experimental evaluation has demonstrated that such an approach has many advantages. In particular, the distribution comparison can be performed by any statistical test which checks whether two-samples are derived from the same underlying population. This allows to instantiate the subspace contrast by tests like the Kolmogorov-Smirnov or Welch-t-test, which are both computationally efficient and statistically robust. Furthermore, we have proposed a novel slicing scheme, which allows to evaluate subspaces of arbitrary dimensionality, as required by the multivariate property of subspace outliers. As a combination of the distribution comparison and the slicing scheme, our contrast measure satisfies a property which we call subspace-equitability. This refers to the fact that the contrast values of two subspaces are immediately comparable irrespective of their dimensionality. These properties allow to construct a highly efficient subspace search framework based on subspace contrast. Overall, we have demonstrated this novel approach to subspace search leads to significant improvements regarding outlier detection: In both synthetic and real-word data sets, HiCS clearly outperforms both traditional outlier models and subspace search methods designed for clustering. Therefore, our approach fills an important gap in the research community.

While our primary goal in developing HiCS was to solve the challenges of subspace outliers, it is possible to consider its properties not only from the perspective of outlier mining but also from the perspective of analyzing attribute relationships. This change of perspective is the topic of Chapter 6, where we have compared the properties of our contrast measure against other popular correlation measures. This study has revealed that subspace contrast has very interesting properties which are not provided by existing correlation measures. In particular, we have shown that the subspace contrast is sensitive to the degree of multiplicity in functional relationships. This means the subspace contrast can differentiate between stronger one-to-one and weaker one-to-many relations by reflecting the different multiplicities in the contrast value. Other non-linear correlation measures cannot show such a sensitivity, since it is in conflict with their specific properties. Thus, the proposal of our subspace contrast offers a novel characteristic for analyzing attribute associations. Furthermore, our experiments have shown that its computation is very efficient and scales very well to large data sets. For the case where one wants to evaluate the correlation of all variable pairs of a data set, it clearly outperforms all other non-linear correlation measures.

In Chapter 7 we have provided an approach showing how to utilize the information of high contrast subspaces for manual outlier assessment. The idea follows the general scheme that we have discussed in the introduction: In this thesis, the aspect of knowledge discovery is two-fold, i.e., our concern is providing knowledge regarding both outliers and attribute relationships. In Chapter 7 we have illustrated the synergies of combining the two types of information. On the one hand, knowledge of high contrast subspaces – implying attribute relationships – provides knowledge regarding outliers, i.e., it helps to understand the deviating characteristic of an outlier. Thus, it enables to describe outliers, which we have formalized as so-called outlier rules. The other way around,

the knowledge of subspace outliers provides insight about attribute relationships, since it implies existence of some kind of structure in the deviating context. Therefore, our evaluation framework allows a manual assessment of both outliers and deviating contexts by means of various visualization techniques.

After these supplementary studies of high contrast subspaces, we come back to the topic of subspace search in Chapter 8 by presenting RefOut, our second major subspace search approach. In contrast to HiCS, the idea behind RefOut is to adapt the subspace search specifically to a given outlier model. This idea allows to formalize the notion of subspace outliers more precisely. We have proposed an algorithmic solution, which exploits tiny fluctuations of outlier scores in order to extract an approximation of the deviating context for the given outlier model. To this end, the algorithm first gathers the joint information of outliers and attribute relationships in random subspace projections. Based on these results, the algorithm refines the set of relevant subspaces, focusing on the most promising outlier/subspace candidates. Overall, this results in a refined set of subspaces specifically adapted to the underlying outlier model. Thus, this novel approach allows a much more specific detection of subspace outliers. To demonstrate the full power of such a model-specific subspace search, we have proposed a modified evaluation method for subspace outlier mining based on a model-specific ground truth obtained by brute-force search. In a thorough evaluation we have shown that RefOut is highly adaptive to the outlier model. This means that RefOut can work very well with any outlier model, while for all other subspace techniques the result depends on whether the outlier model fits to the subspace search objective. Furthermore, the specific search for deviating contexts facilitates outlier description mining, since the deviating contexts are obtained for each outlier individually by design.

For the second part of the thesis we have turned from static databases to the case of dynamic data in the form of data streams. In the beginning of Chapter 9, we have discussed how the dynamic nature of data streams leads to a fundamental change of the general problem statement. Due to the dynamics, any attribute relationship – and accordingly the deviating contexts of subspace outliers – may change over time themselves. Therefore, it is necessary to not only consider outliers in the context of attributes, but also in their corresponding temporal contexts. Finding the dependencies over arbitrary temporal contexts has not been addressed in the literature before, not even for the simplest case of bivariate subspaces. Therefore, we have focused on the most basic case of finding pairwise attribute relationships on data streams. Specifically, we have aimed at enabling quantification of attribute relationships by means of mutual information estimation. Our Mise approach presented in Chapter 9 is the first technique allowing to quantify attribute relationships over arbitrary time contexts. This is achieved by developing a novel data structure called query anchor, which is the foundation of the Mise framework. Furthermore, Mise provides a unique property by maintaining a time-scale invariant estimation quality. This means that the evaluation of temporal contexts does not have a quality bias for either short time scales (high-frequency effects) or long time scales (low-frequency effects), but achieves an equal treatment of time scales in general. We

have completed Chapter 9 by a thorough evaluation of MISE, which has shown excellent results regarding both estimation quality and efficiency.

10.2. Impact

Overall, the techniques presented in this thesis have been well received by the research community. This is especially the case for HICS, which was our first publication in the direction of subspace search for outlier mining. Since its publication in 2012 it has served as inspiration for other researchers, which have either adopted algorithmic ideas or the general topic of subspace search for outlier mining. In the following we want to discuss interesting follow-up publications and their relation to our work.

An immediate follow-up technique to HICS has been published in [NMV+13]. In this work, the basic idea to search for subspaces is exactly the same as ours: The technique searches for subspaces deviating from the case of independent dimensions. Similar to HICS, the subspace contrast is defined as the deviation between the joint probability and the product of marginal probabilities. The key difference in the subspace contrast is that it uses cumulative entropy to compare between the joint distribution and the dimension that is singled out. The resulting subspace contrast does possess different properties, e.g., it is no longer normalizable to $[0, 1]$, but will take arbitrarily large values. Furthermore, the deviation values depend on the domain of each dimensions, i.e., an attribute with a larger numerical range can produce larger deviation results than dimensions with a narrow domain. Unfortunately, there is no clear motivation for this decision and no discussion how it compares to using a standard statistical test like in HICS. Another difference is the solution to single out a dimension for the deviation comparison. While HICS averages over multiple deviations in a Monte Carlo approach, [NMV+13] uses an interesting modification: It generates a single permutation of all attributes in a greedy algorithm. For small data sets this is advantageous in terms of the run time, because it avoids the repeated assessment of HICS. On the other hand, this increases the complexity w.r.t. the dimensionality, since the greedy algorithm has a quadratic complexity. The processing of subspaces is the same Apriori-like processing as in HICS, which is referred to a beam-search in [NMV+13]. Overall, [NMV+13] shows many similarities to our work, suggesting several algorithmic modifications trading off technical details.

Another work showing a clear influence by HICS is [NMB13], since the aim of this work is also to perform a subspace search depending on correlations. However, it addresses one of the major challenges in subspace search: The processing scheme of subspace candidates. Both HICS and [NMV+13] rely on an Apriori-like processing, i.e., they process subspaces from low-dimensional to high-dimensional in a stepwise fashion. For very high-dimensional subspace structures this requires to evaluate a large number of intermediate subspace candidates in order to reach these structures. Thus, it is challenging to detect structures of high dimensionality. The idea behind [NMB13] is to simplify the detection

scheme by limiting the search to subspaces which possess a visible pairwise dependence. This allows to transform the problem of subspace search to the problem of clique mining. As a result, this allows to get rid of a level-wise processing by searching for maximal cliques immediately. Overall, this is an interesting idea to speed-up the processing of high-dimensional subspaces, but it is limited to the case of pairwise dependencies.

The idea of HiCS has also been applied in completely different domains. For instance, Iglesias et al. have studied the problem of finding congruent subspaces in attributed graphs [ISML+13]. For many graph mining algorithms, it is an important property that a graph satisfies the homophily condition, i.e., the condition that a similarity in attribute values is reflected by a structural similarity w.r.t. the graph structure. Congruent subspaces consist of a set of attributes which possess this property. Therefore, detecting congruent subspaces is an important preprocessing step for many other graph mining algorithms. In order to find congruent subspaces, the algorithm proposed in [ISML+13] applies a slicing approach similar to HiCS: The algorithm uses the slices in order to constrain the attributes of a subspace to random intervals. This results in a conditioned sample like in HiCS. In order to measure the congruence of a subspace, a statistical test is used to check whether the conditional sample has either a random edge distribution or whether it shows homophily. By repeating several Monte Carlo iterations over different slices, the overall subspace congruence can be obtained similarly to the subspace contrast in HiCS. Overall, [ISML+13] has showed that the detection of congruent subspaces leads to significant improvements of existing algorithms on attributed graphs. It is very interesting to see an adoption of the ideas behind HiCS in such a different research field.

While one of our motivations for HiCS was to bring subspace search from clustering to outlier mining, our work in return has had influence on the clustering field: In [Hö14], a subspace search technique for clustering has been proposed, which follows the ideas behind HiCS. The technique aims at finding small clusters which are embedded in noise. Similar to HiCS the technique defines the interestingness of a subspace depending on the deviation from independence. The technique also adapts the idea of subspace slicing, i.e., to define slices w.r.t. the rank orders of the dimensions, also avoiding to operate on the original attribute domains. The key difference is how to extract a deviation value from the conditional sample corresponding to the slicing: Instead of analyzing the distribution of the conditional sample as we do in HiCS, the deviation is only determined by the size of the sample. Basically, this defines a subspace contrast based on the deviation of our Equation 5.7, which specifies the expected sample size under the independence assumption. Regarding the processing of subspaces, [Hö14] proposes a depth-first approach instead of an Apriori-processing. Overall, the technique in [Hö14] produces very good results in an experimental study on clustering. Thus, it is interesting to see that the notion of subspace contrast in a broader sense is also capable of detecting cluster structures.

10.3. Future Research Directions

An immediate direction for future research is in the area of data streams. In this thesis, we have provided the very first steps for subspace search on data streams by considering pairwise combinations of attributes. We have shown that the complexity of subspace search is significantly increased due to the temporal characteristic, and a complete coverage of all implications of dynamic data is beyond the scope of this thesis. Regarding future research, our approach can be used as a foundation for more complex subspace search approaches. One of the first challenges will be to extend the subspace analysis from bivariate to the general case of multivariate subspaces. Typically, the formation of higher dimensional subspaces is based on low-dimensional subspaces. However, such an incremental processing from lower to higher dimensions is difficult on data stream, since it could mean that the evaluation of high-dimensional structures is temporally delayed due to this dependence in the computation. Other challenges arise regarding temporal contexts. One possibility would be to automatize the detection of temporal contexts, which would result in a simultaneous search of subspaces and temporal contexts. On the technical side, keeping track of temporal contexts will always raise the question of how to deal with infinite stream lengths. To this end, future subspace search techniques will require approximation approaches similar to our query anchor data structure. Overall, subspace search on data stream offers a plethora of open issues to be addressed by the research community.

On the other hand, even with static data there are still many interesting directions for future research, for instance regarding the general relation of subspace search and correlation measures. We have seen in Chapter 6 that even for bivariate correlation measures it is challenging to formalize and compare their properties. We have discovered that our notion of subspace contrast has the properties subspace-equitability and it can detect differences in the multiplicity of a dependence. On the other hand, the maximal information coefficient provides e.g. noise-equitability (approximately). This raises the question how such properties relate to each other formally. For instance it may be possible to prove that such properties are mutually exclusive. Previous attempts to analyze such properties formally have turned out to be challenging [KA13, MMM14]. The comparison of correlation measures becomes even more challenging in the case of multivariate correlation. One of the aims of future research could be to formalize the properties of all multivariate correlation measures as well as subspace relevance measures like our subspace contrast. This would allow to understand the differences of these notions more clearly, leading to a better understanding of their individual use cases.

Even for traditional outlier mining there is a potential emerging research field related to our work, which has also been discussed by a recent positional paper [Agg13b]. In this work, the author picks up the idea behind RefOut of using multiple outlier models by suggesting to investigate outlier ensemble models in general. Thus, the suggestion is to not only perform outlier analysis w.r.t. one arbitrary but fixed outlier model like in

RefOut, but to use multiple models simultaneously. A key question for future research is whether such ensemble methods can be equally successful for unsupervised learning as they are in the supervised domain. In the supervised case, forming an ensemble is straightforward since e.g. the weights of weak classifiers can be inferred directly from the objective function. In the unsupervised domain, the construction of ensemble models is more challenging. Nevertheless, they have the potential to provide more knowledge compared to a single model. For instance, an algorithm could exploit the fact that certain outlier models either agree or disagree on the anomaly of certain objects, maybe allowing to infer the degree of anomaly w.r.t. a model without actually applying it. In other words, an ensemble method could become a meta learning algorithm, since it will learn the properties of the underlying models. Another strength of ensembles could be to combine faster approximate models with slower but more precise models, and apply a dynamic switching between them with the goal of optimizing both run time and quality. Hence, the multi-model idea behind RefOut could play an important role for the development of future outlier ensemble models.

Finally, we also see a potential for future research regarding manual outlier assessment as discussed in Chapter 7. In our opinion, it will be necessary to bridge the gap between algorithms and users in the future. In this spirit, the output of algorithms should allow for a much more immediate interaction with users. Ideally, this not only applies for mining outliers, but also for mining attribute relationships: In both cases, a user might typically already know certain types of anomalies or relationships in the data. From a user perspective, it is not helpful if the algorithm output is cluttered with such existing knowledge. What is of interest for a user is typically only the things that they do not know yet. Furthermore, allowing a user to directly interact with an algorithm has a mutual benefit: It not only allows a user to obtain more specific and compact results, it also allows the algorithm to operate more efficiently by pruning unnecessary information. Therefore, we think that both outlier mining and attribute relationship analysis will benefit in the long term by shifting from the fully unsupervised domain towards interactive semi-supervised techniques, allowing to incorporate user objectives more directly.

In summary, our work has produced many novel ideas, models, and evaluation results for both outlier mining and attribute relationship analysis. We have seen that the ideas presented in this thesis have already inspired others to apply them in different fields as well. Therefore, we hope that our contributions can provide a basis for future research.

A. Mise Quality Results per Data Stream

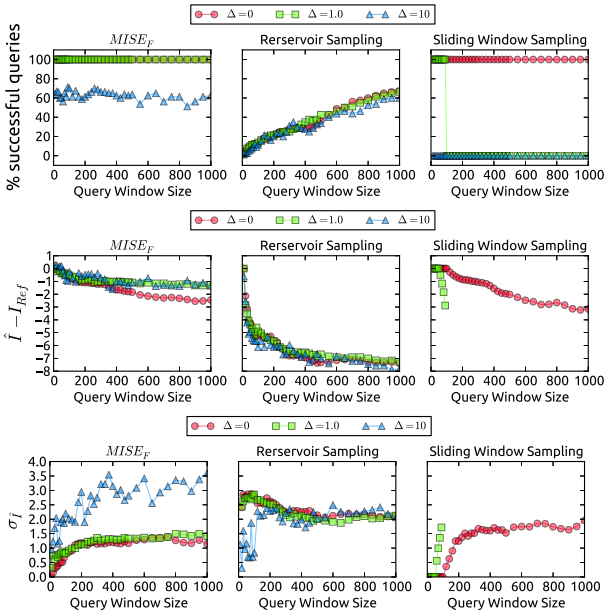


Figure A.1.: Result on IMU Stream 1 (PAMAP)

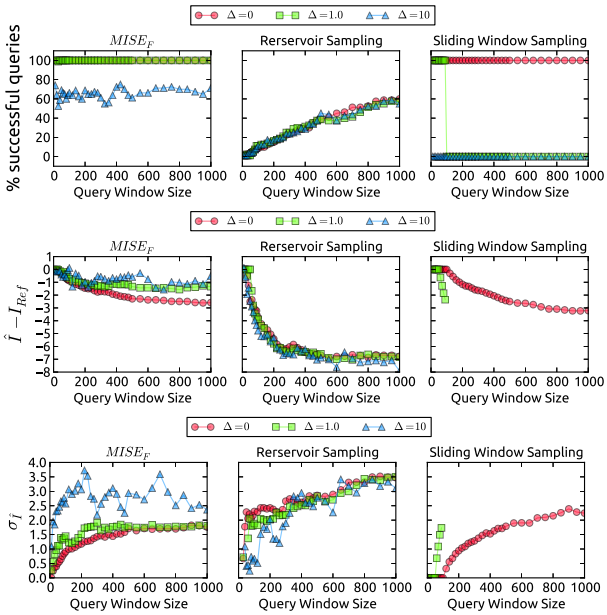


Figure A.2.: Result on IMU Stream 2 (PAMAP)

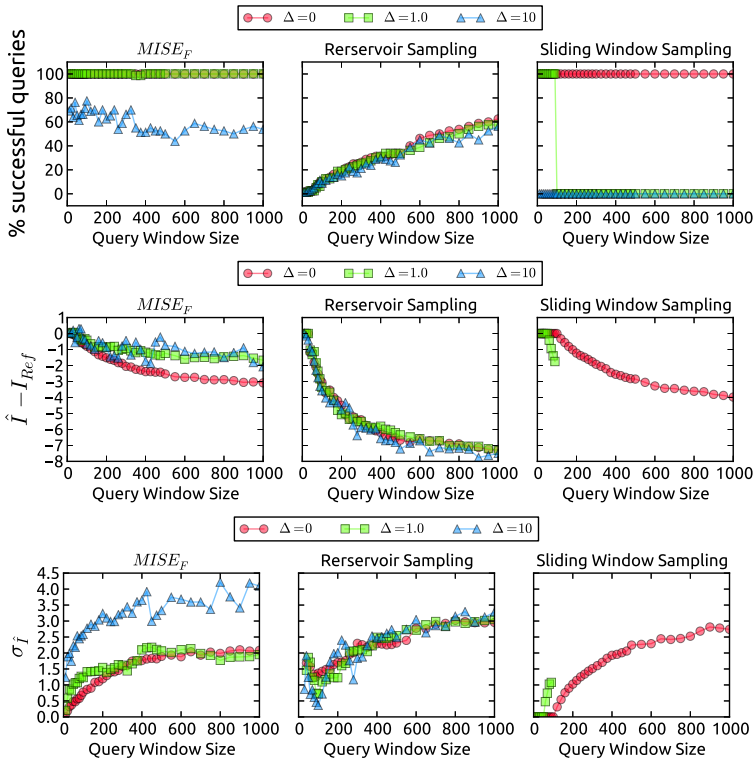


Figure A.3.: Result on IMU Stream 3 (PAMAP)

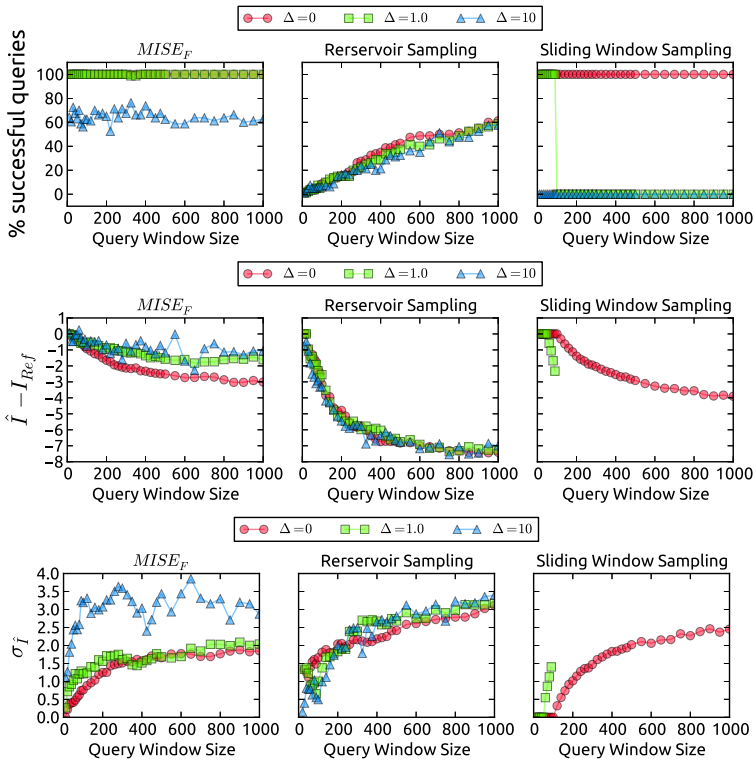


Figure A.4.: Result on IMU Stream 4 (PAMAP)

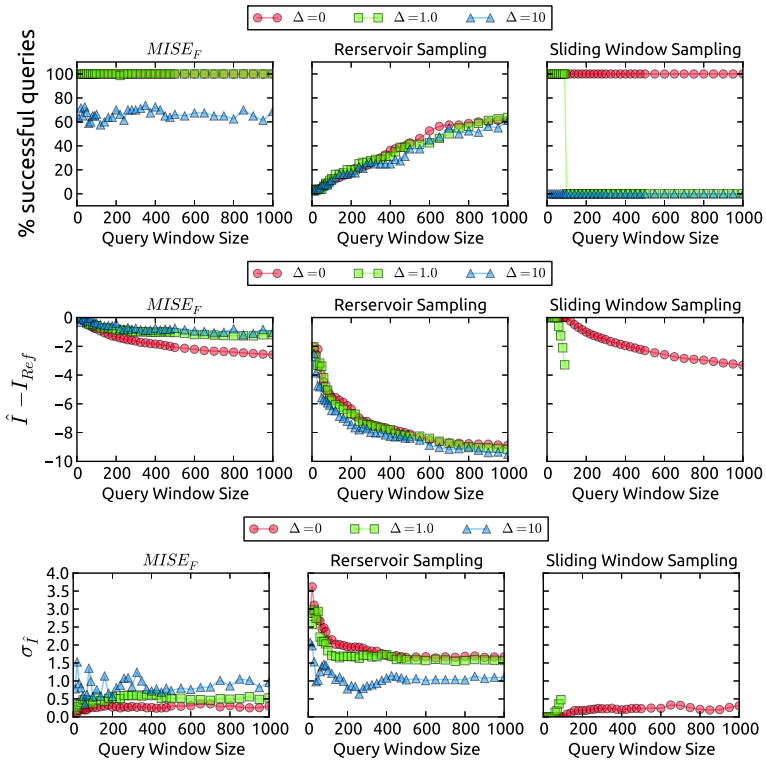


Figure A.5.: Result on IMU Stream 5 (PAMAP)

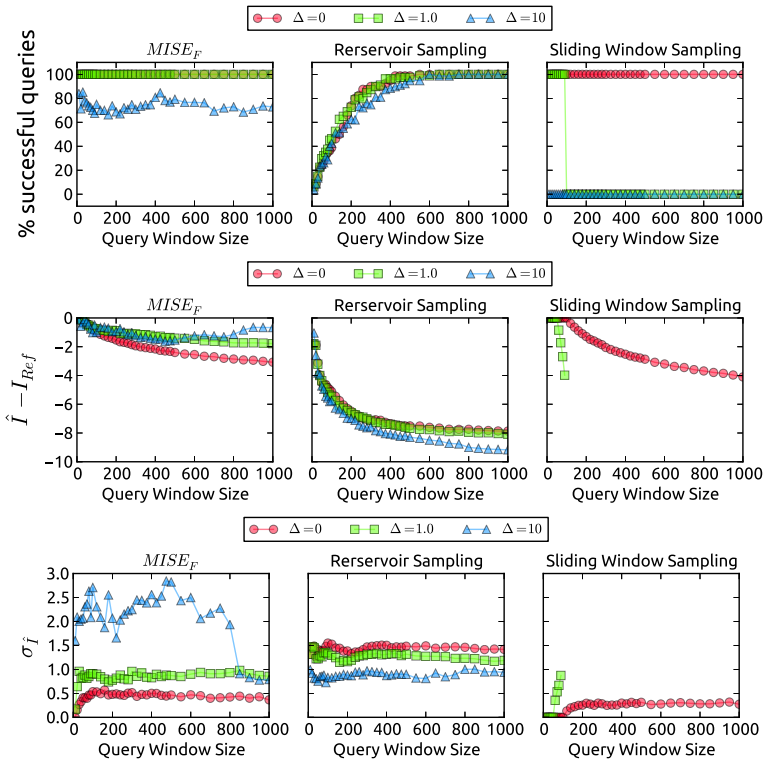


Figure A.6.: Result on Smart Meter Stream 1

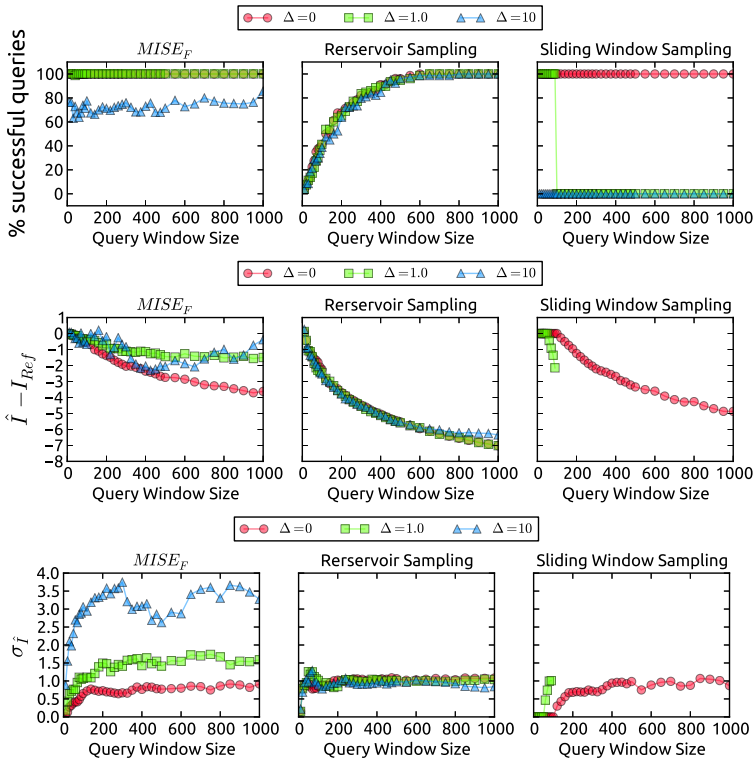


Figure A.7.: Result on Smart Meter Stream 2

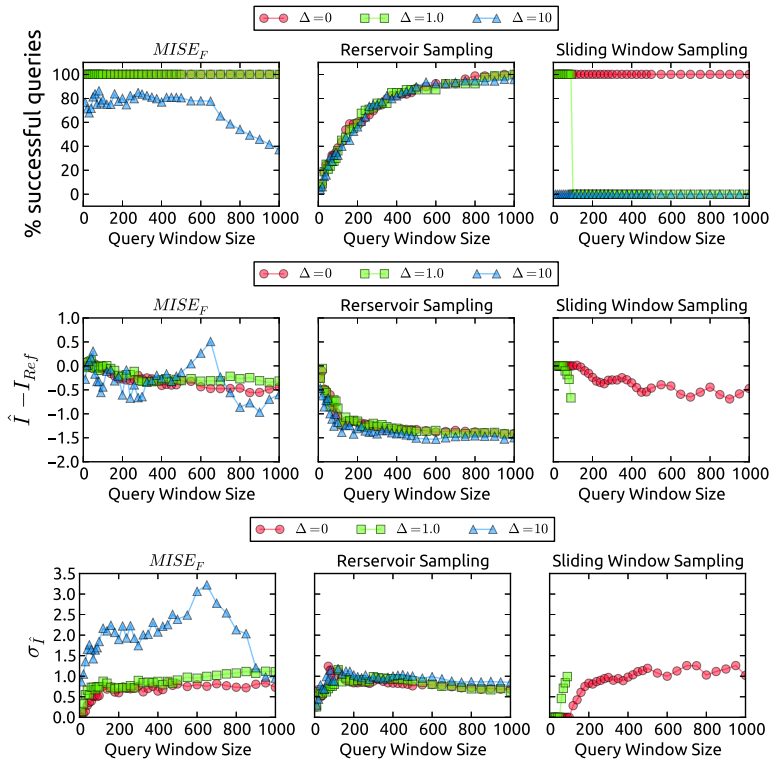


Figure A.8.: Result on Smart Meter Stream 3

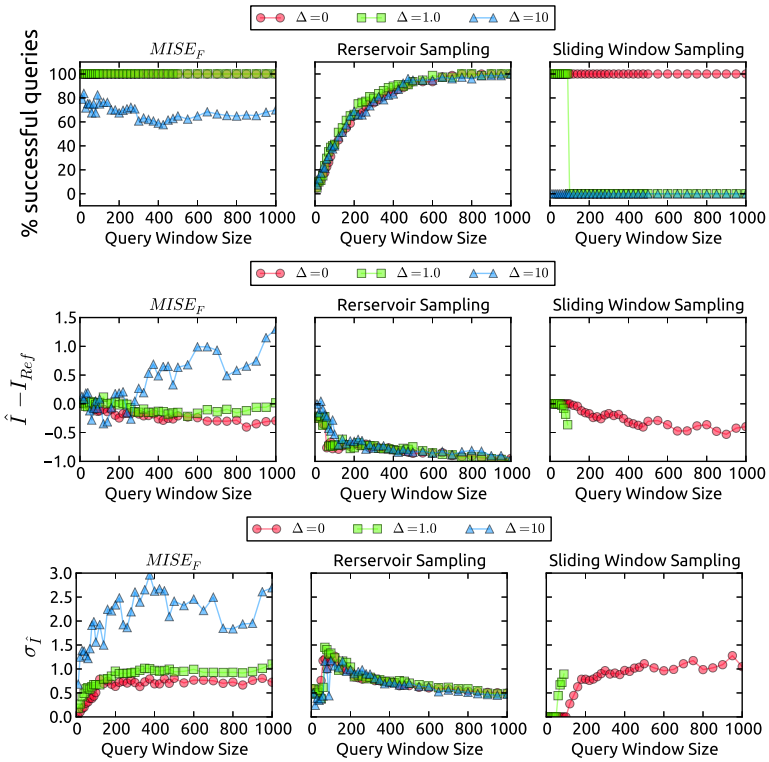


Figure A.9.: Result on Smart Meter Stream 4

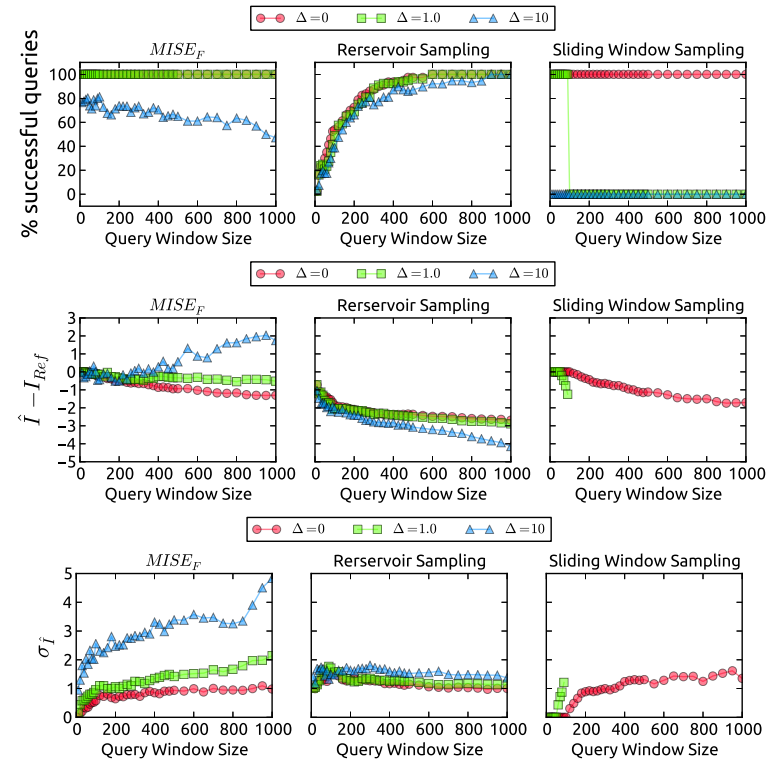


Figure A.10.: Result on Smart Meter Stream 5

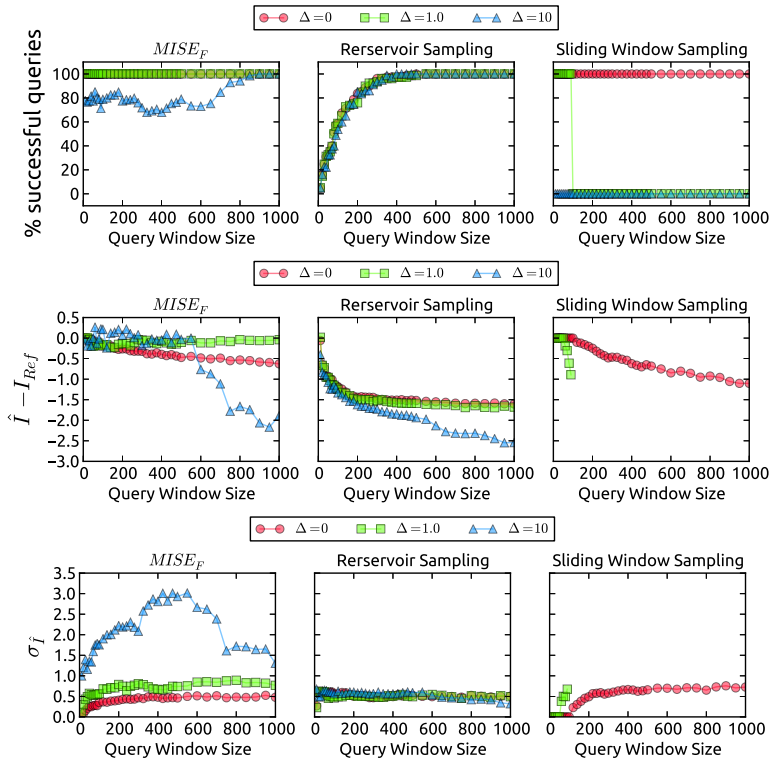


Figure A.11.: Result on Stock Data Stream (IBM + GE)

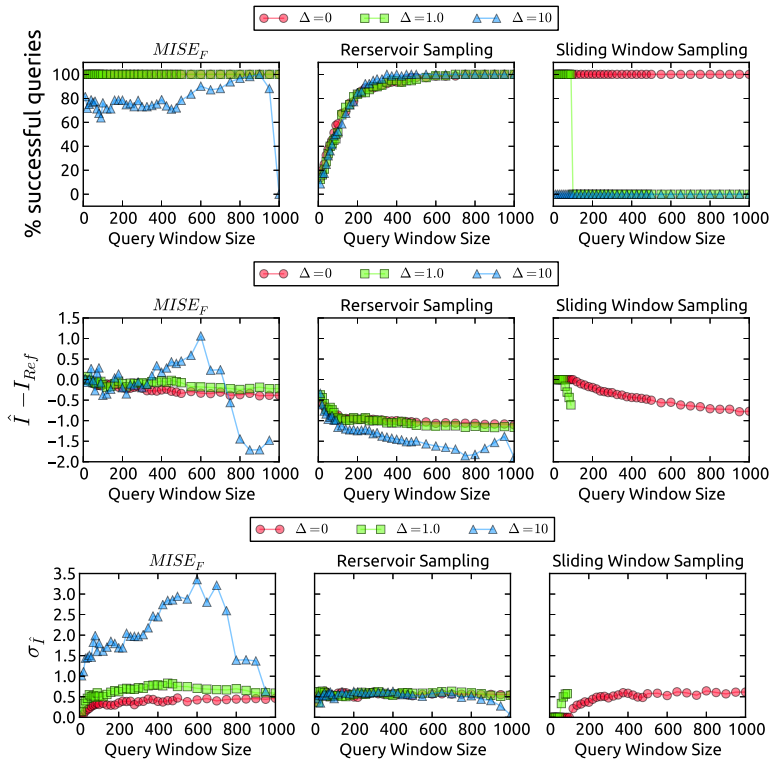


Figure A.12.: Result on Stock Data Stream (PG + IP)

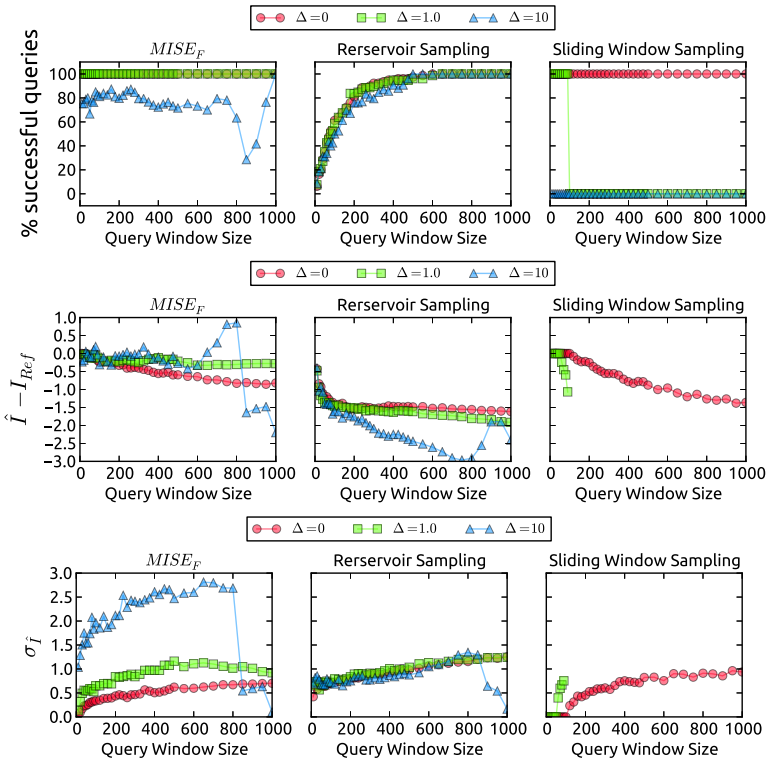


Figure A.13.: Result on Stock Data Stream (PG + KO)

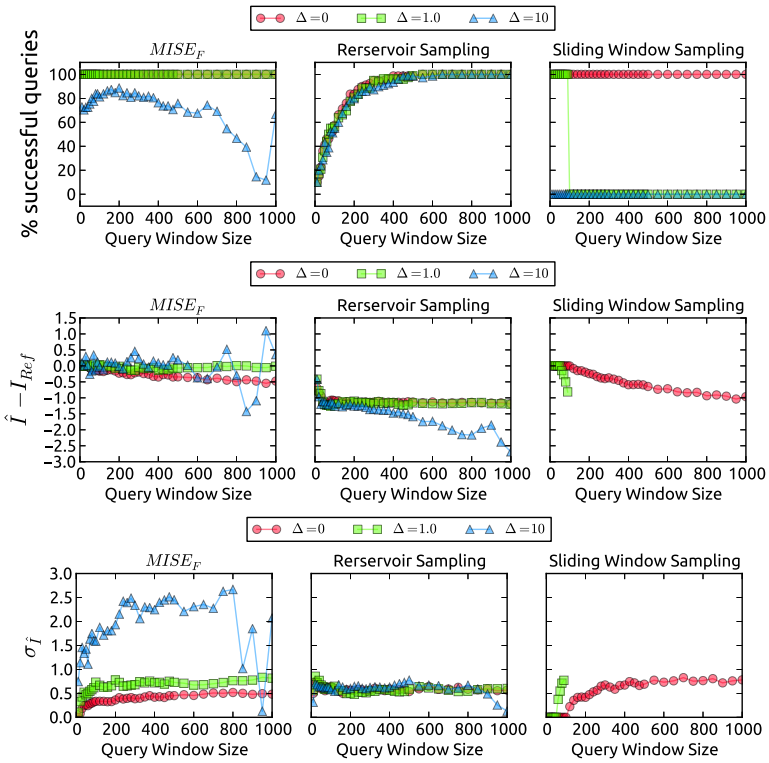


Figure A.14.: Result on Stock Data Stream (PG + MMM)

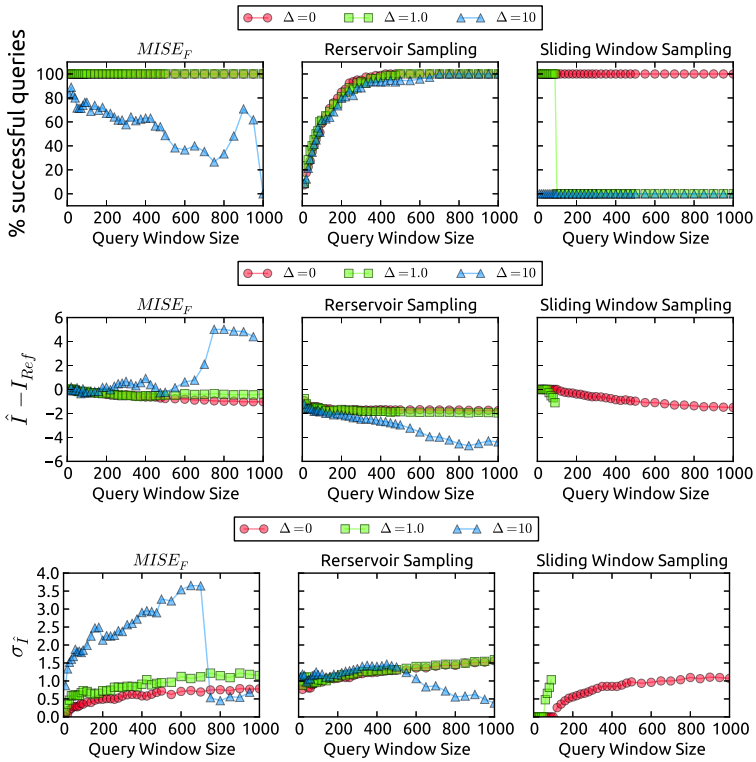


Figure A.15.: Result on Stock Data Stream (PG + MRK)

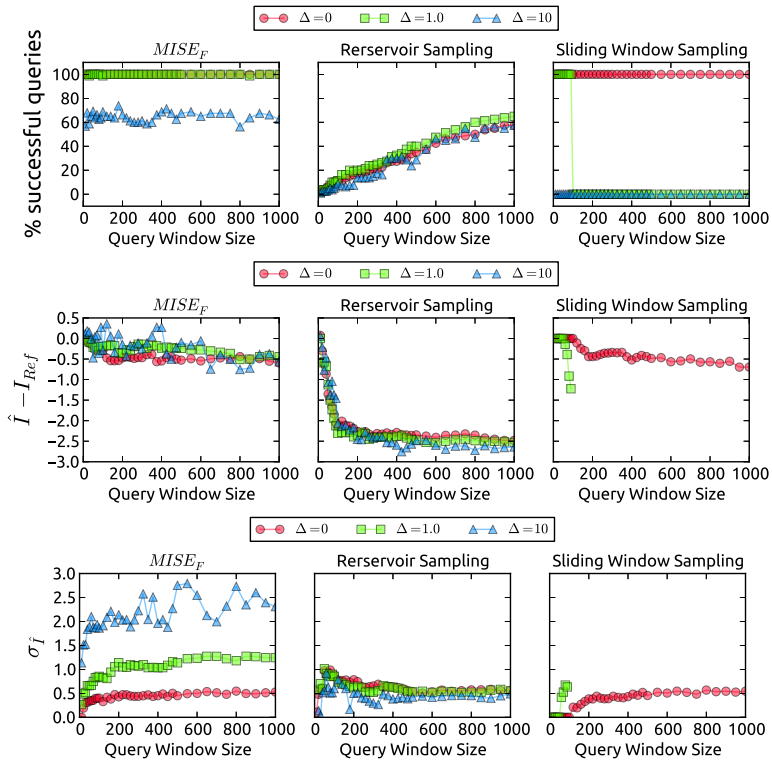


Figure A.16.: Result on ECG Stream (Congestive Heart)

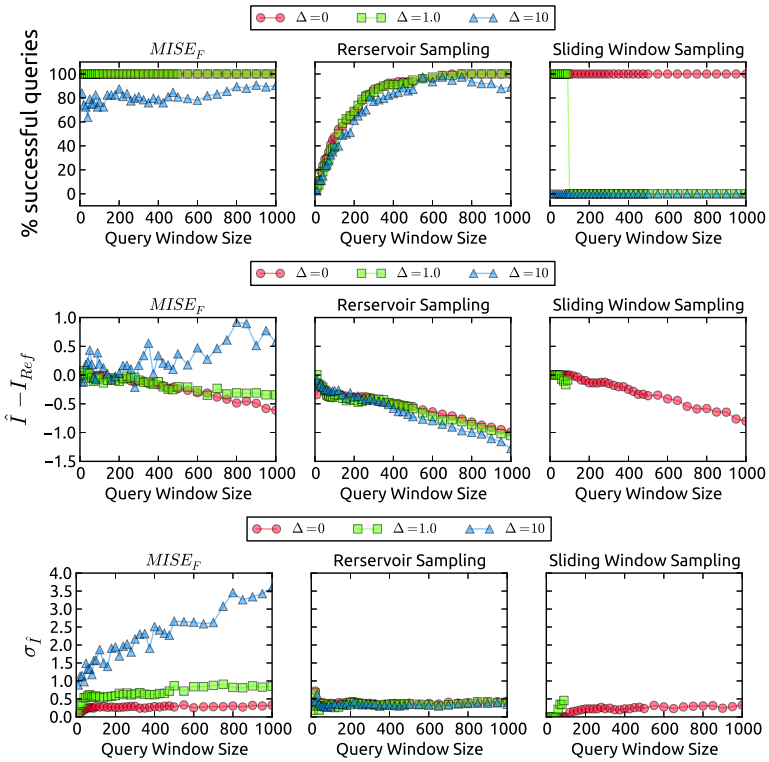


Figure A.17.: Result on Climate Stream

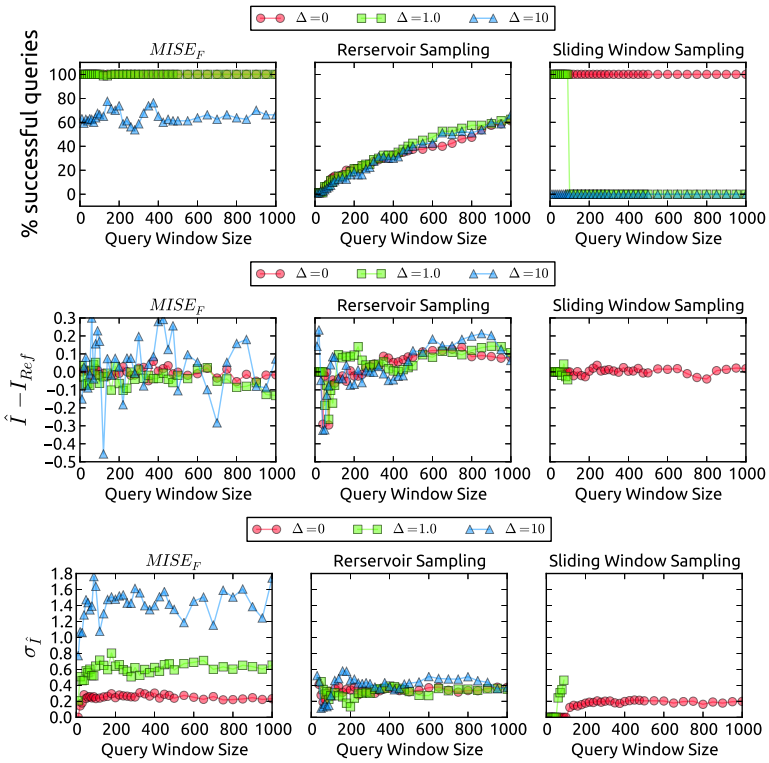


Figure A.18.: Result on Static Gaussian Stream (rho = 0)

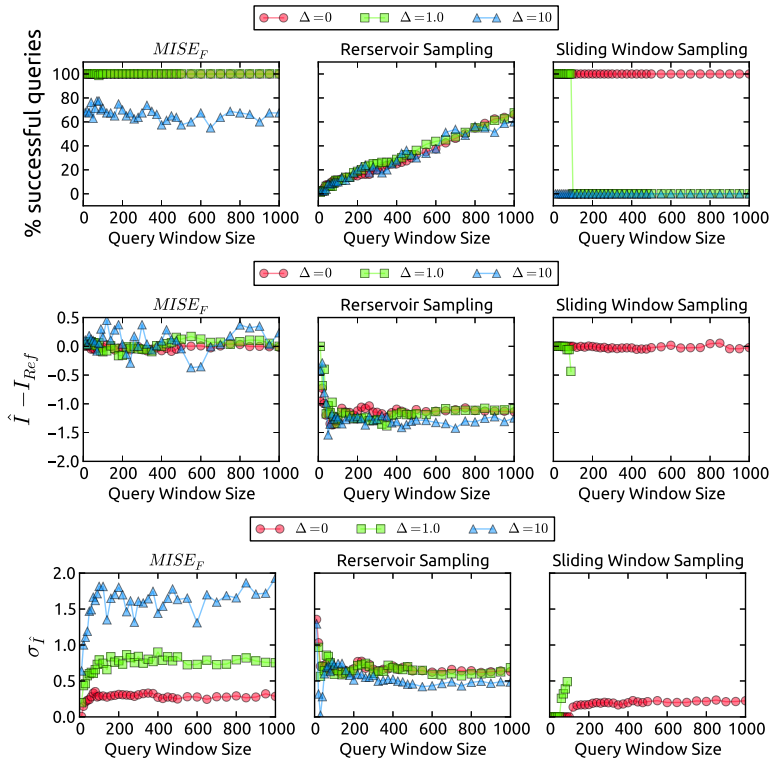


Figure A.19.: Result on Static Gaussian Stream (rho = 0.95)

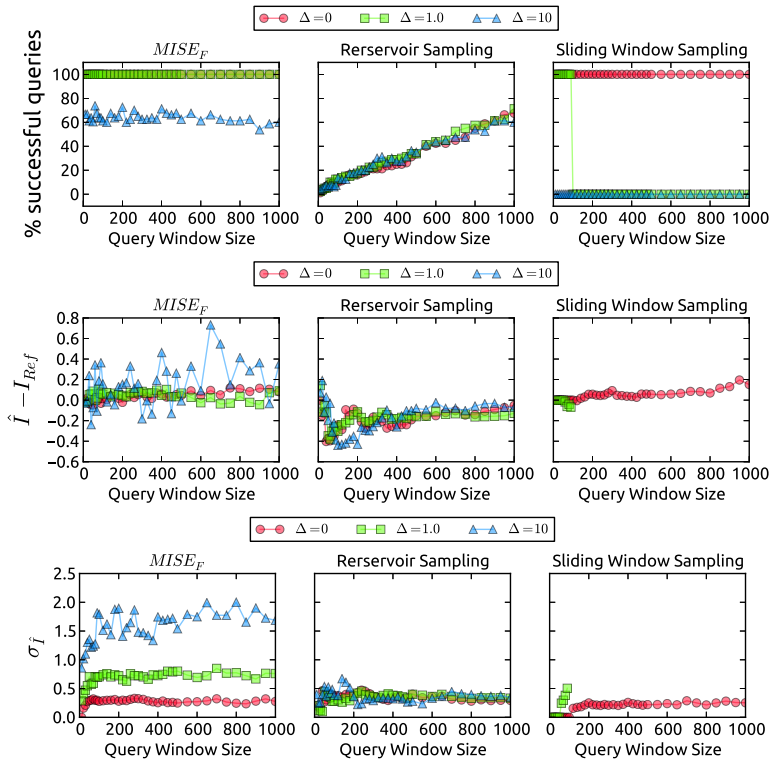


Figure A.20.: Result on Synthetic Gaussian Mixture Stream 1

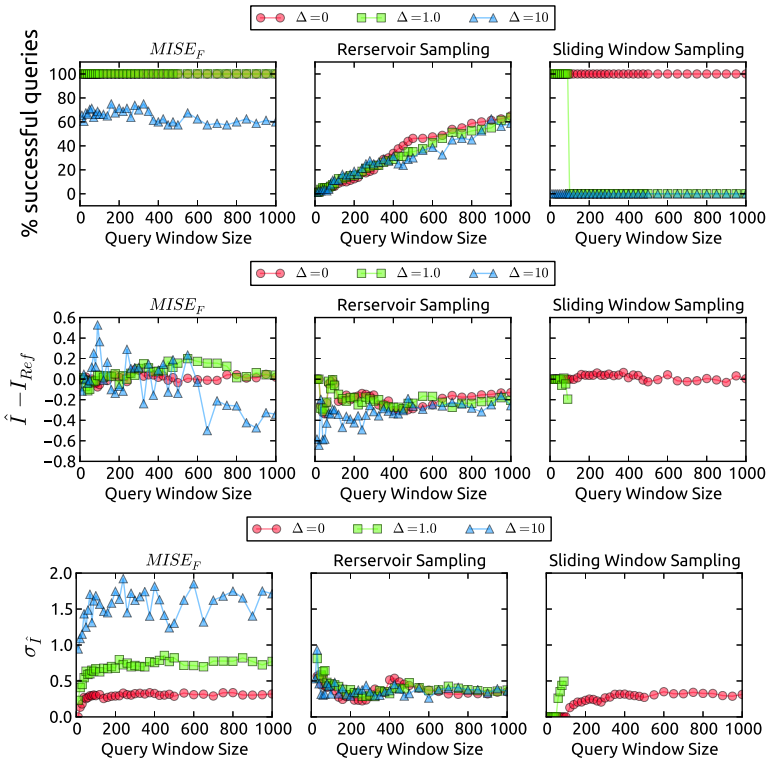


Figure A.21.: Result on Synthetic Gaussian Mixture Stream 2

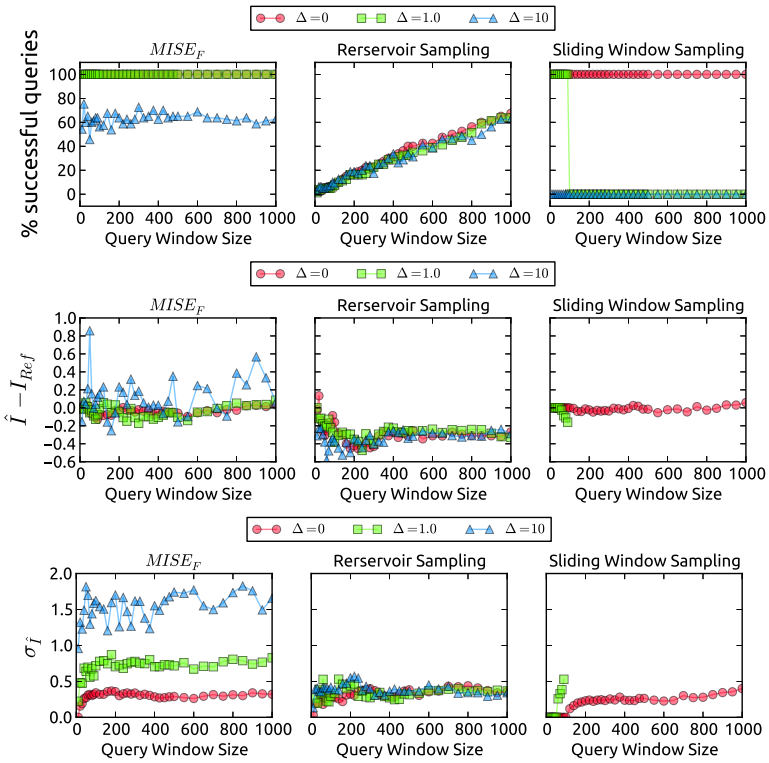


Figure A.22.: Result on Synthetic Gaussian Mixture Stream 3

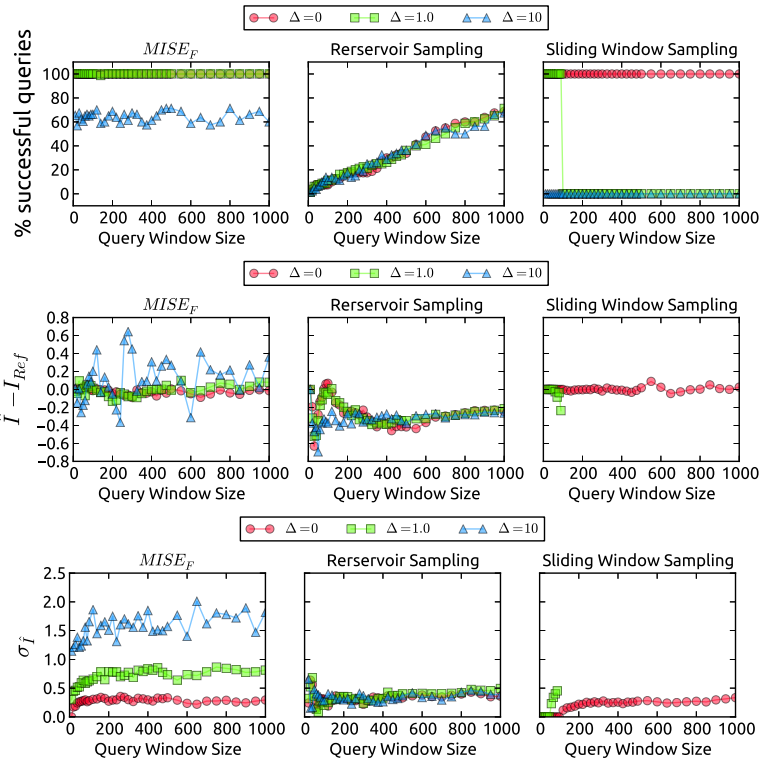


Figure A.23.: Result on Synthetic Gaussian Mixture Stream 4

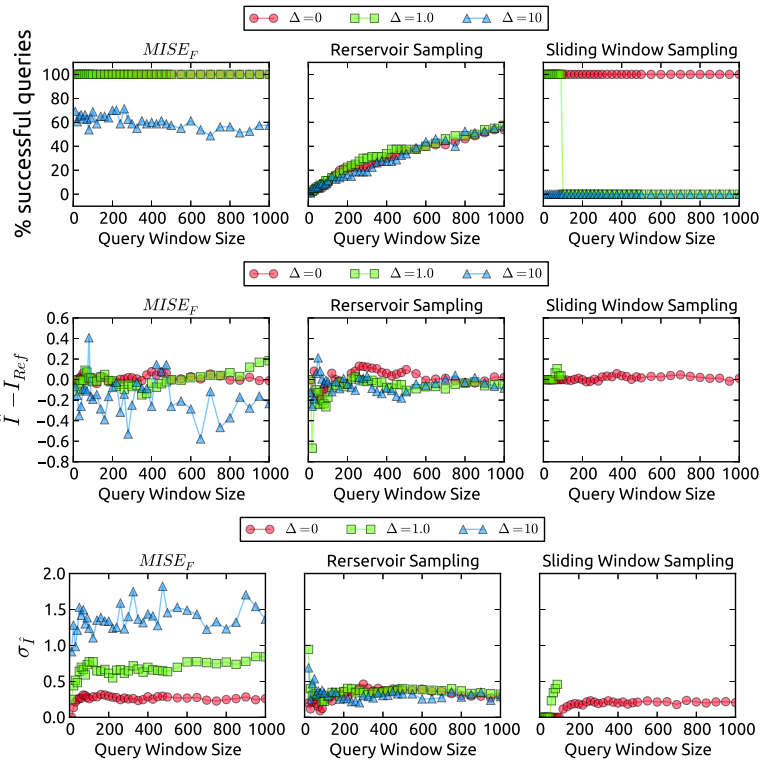


Figure A.24.: Result on Synthetic Gaussian Mixture Stream 5

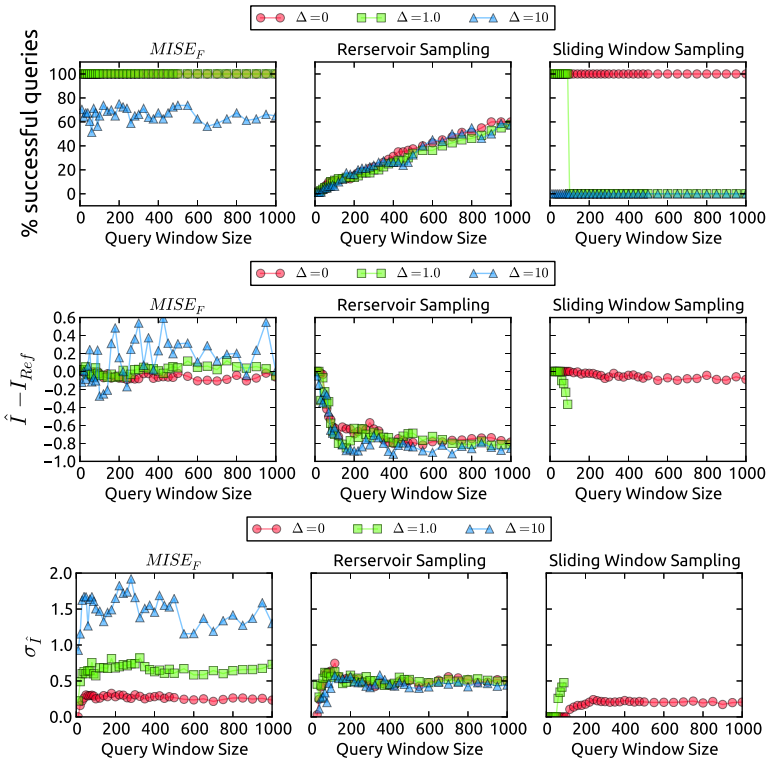


Figure A.25.: Result on Synthetic Uniform Mixture Stream 1

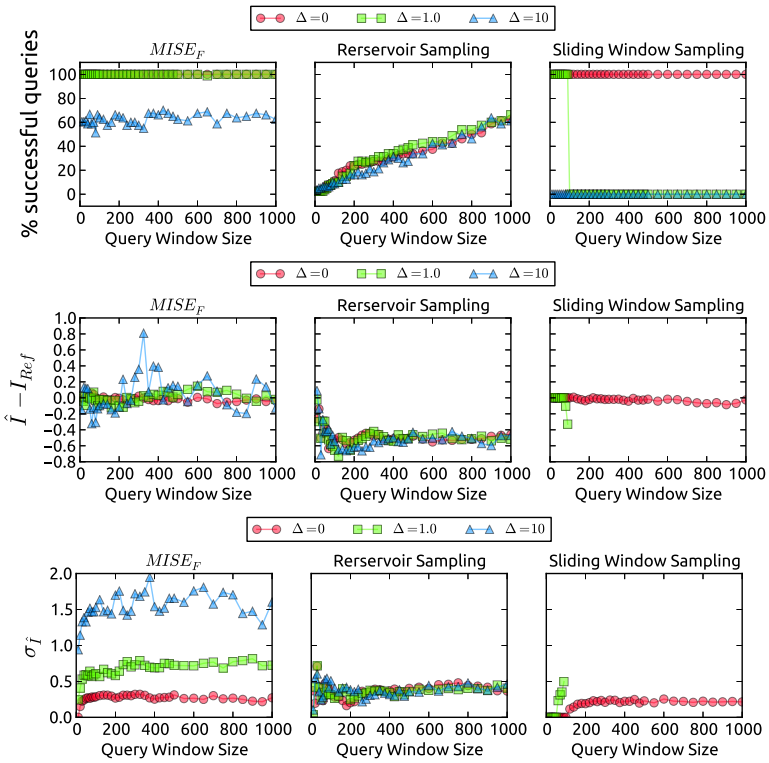


Figure A.26.: Result on Synthetic Uniform Mixture Stream 2

List of Tables and Figures

Bibliography

[AFP09] Angiulli, Fabrizio ; Fassetti, Fabio ; Palopoli, Luigi: Detecting Outlying Properties of Exceptional Objects. In: ***ACM Transactions on Database Systems*** 34 (2009), Nr. 1

[AFV+13] Albanese, Davide ; Filosi, Michele ; Visintainer, Roberto ; Riccadonna, Samantha ; Jurman, Giuseppe ; Furlanello, Cesare: minerva and minepy: a C engine for the MINE suite and its R, Python and MATLAB wrappers. In: ***Bioinformatics*** 29 (2013), Nr. 3

[Agg06] Aggarwal, Charu C.: On Biased Reservoir Sampling in the Presence of Stream Evolution. In: ***Proceedings of the International Conference on Very Large Data Bases (VLDB)***, 2006

[Agg13a] Aggarwal, Charu C.: ***Outlier Analysis***. 2013

[Agg13b] Aggarwal, Charu C.: Outlier Ensembles: Position Paper. In: ***ACM SIGKDD Explorations Newsletter*** 14 (2013), Nr. 2

[AGGR98] Agrawal, Rakesh ; Gehrke, Johannes ; Gunopulos, Dimitrios ; Raghavan, Prabhakar: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In: ***Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)***, 1998 (SIGMOD '98)

[AKMS07] Assent, Ira ; Krieger, Ralph ; Müller, Emmanuel ; Seidl, Thomas: DUSC: Dimensionality Unbiased Subspace Clustering. In: ***Proceedings of the IEEE International Conference on Data Engineering (ICDE)***, 2007

[AKMS08] Assent, Ira ; Krieger, Ralph ; Muller, Emmanuel ; Seidl, Thomas: INSCY: Indexing subspace clusters with in-process-removal of redundancy. In: ***Proceedings of the IEEE International Conference on Data Mining (ICDM)***, 2008

[AKR+10] Achtert, Elke ; Kriegel, Hans-Peter ; Reichert, Lisa ; Schubert, Erich ; Wojdanowski, Remigius ; Zimek, Arthur: Visual evaluation of outlier detection models. In: ***Database Systems for Advanced Applications***, 2010

[AM04] Arasu, Arvind ; Manku, Gurmeet S.: Approximate Counts and Quantiles over Sliding Windows. In: ***Proceedings of the ACM Symposium on Principles of Database Systems (PODS)***, 2004

[AS94] Agrawal, Rakesh ; Srikant, Ramakrishnan: Fast Algorithms for Mining Association Rules in Large Databases. In: ***Proceedings of the International Conference on Very Large Data Bases (VLDB)***, 1994 (VLDB '94)

[AWY+99] Aggarwal, Charu C. ; Wolf, Joel L. ; Yu, Philip S. ; Procopiuc, Cecilia ; Park, Jong S.: Fast Algorithms for Projected Clustering. In: ***Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)***, 1999 (SIGMOD '99)

[AY01] Aggarwal, Charu C. ; Yu, Philip S.: Outlier Detection for High Dimensional Data. In: ***Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)***, 2001 (SIGMOD '01)

[Aze08] Azevedo, Ana Isabel Rojão L.: KDD, SEMMA and CRISP-DM: a parallel overview. (2008)

[BBD+02] Babcock, Brian ; Babu, Shivnath ; Datar, Mayur ; Motwani, Rajeev ; Widom, Jennifer: Models and Issues in Data Stream Systems. In: ***Proceedings of the ACM Symposium on Principles of Database Systems (PODS)***, 2002

[BG06] Bhuvanagiri, Lakshminath ; Ganguly, Sumit: Estimating Entropy over Data Streams. In: ***Algorithms – ESA***. 2006

[BGRS99] Beyer, Kevin S. ; Goldstein, Jonathan ; Ramakrishnan, Raghu ; Shaft, Uri: When Is "Nearest Neighbor" Meaningful? In: ***Proceedings of the 7th International Conference on Database Theory***, 1999 (ICDT '99)

[BKNS00] Breunig, Markus ; Kriegel, Hans-Peter ; Ng, Raymond T. ; Sander, Jörg: LOF: Identifying Density-Based Local Outliers. In: ***Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)***, 2000

[BN03] Belkin, Mikhail ; Niyogi, Partha: Laplacian Eigenmaps for dimensionality reduction and data representation. In: ***Neural Computation*** 15 (2003), Nr. 6

[BOPY07] Böhm, C. ; Ooi, Beng C. ; Plant, C. ; Yan, Ying: Efficiently Processing Continuous k-NN Queries on Data Streams. In: ***Proceedings of the IEEE International Conference on Data Engineering (ICDE)***, 2007

[BPR+04] Baumgartner, C. ; Plant, C. ; Railing, K. ; Kriegel, H.-P. ; Kroger, P.: Subspace selection for clustering high-dimensional data. In: ***Proceedings of the IEEE International Conference on Data Mining (ICDM)***, 2004

[BPV03] Bengio, Yoshua ; Paiement, Jean-Francois ; Vincent, Pascal: Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. In: ***Advances in Neural Information Processing Systems (NIPS)***, 2003

[BS03] Bay, Stephen D. ; Schwabacher, Mark: Mining Distance-based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule. In: ***Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)***, 2003 (KDD '03)

[CBK07] Chandola, Varun ; Banerjee, Arindam ; Kumar, Vipin: Outlier detection: A survey. In: ***ACM Computing Surveys*** (2007)

[CBK09] Chandola, Varun ; Banerjee, Arindam ; Kumar, Vipin: Anomaly Detection: A Survey. In: ***ACM Computing Surveys*** 41 (2009), Nr. 3

[CCM07] Chakrabarti, Amit ; Cormode, Graham ; McGregor, Andrew: A Near-optimal Algorithm for Computing the Entropy of a Stream. In: ***Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)***, 2007

[CFZ99] Cheng, Chun-Hung ; Fu, Ada W. ; Zhang, Yi: Entropy-based Subspace Clustering for Mining Numerical Data. In: ***Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)***, 1999 (KDD '99)

[CGHJ12] Cormode, Graham ; Garofalakis, Minos ; Haas, Peter J. ; Jermaine, Chris: Synopses for Massive Data: Samples, Histograms, Wavelets, Sketches. In: ***Found. Trends Databases*** 4 (2012), Nr. 1-3

[CKT08] Cormode, G. ; Korn, F. ; Tirthapura, S.: Exponentially Decayed Aggregates on Data Streams. In: ***Proceedings of the IEEE International Conference on Data Engineering (ICDE)***, 2008

[CSSX09] Cormode, G. ; Shkapenyuk, V. ; Srivastava, D. ; Xu, Bojian: Forward Decay: A Practical Time Decay Model for Streaming Systems. In: ***Proceedings of the IEEE International Conference on Data Engineering (ICDE)***, 2009

[CTX07] Cormode, Graham ; Tirthapura, Srikanta ; Xu, Bojian: Time-decaying Sketches for Sensor Data Aggregation. In: ***Proceedings of the ACM Symposium on Principles of Distributed Computing (SODC)***, 2007

[DGIM02] Datar, Mayur ; Gionis, Aristides ; Indyk, Piotr ; Motwani, Rajeev: Maintaining Stream Statistics over Sliding Windows. In: ***Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)***, 2002

[DH07] Dijck, Gert V. ; Hulle, Marc M. V.: Speeding Up Feature Subset Selection Through Mutual Information Relevance Filtering. In: ***Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)***. 2007

[DSSK04] Daub, Carsten O. ; Steuer, Ralf ; Selbig, Joachim ; Kloska, Sebastian: Estimating mutual information using B-spline functions - an improved similarity measure for analysing gene expression data. In: ***BMC Bioinformatics*** 5 (2004)

[DV99] Darbellay, G.A. ; Vajda, I.: Estimation of the information by an adaptive partitioning of the observation space. In: ***IEEE Transactions on Information Theory*** 45 (1999), Nr. 4

[EKSX96] Ester, Martin ; Kriegel, Hans-Peter ; Sander, Jörg ; Xu, Xiaowei: A density-based algorithm for discovering clusters in large spatial databases with noise. In: ***Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*** Bd. 96, 1996

[ES06] Efraimidis, Pavlos S. ; Spirakis, Paul G.: Weighted Random Sampling with a Reservoir. In: ***Inf. Process. Lett.*** 97 (2006), Nr. 5

[FA10] Frank, A. ; Asuncion, A.: ***UCI Machine Learning Repository***. 2010 http://archive.ics.uci.edu/ml

[FL95] Faloutsos, Christos ; Lin, King-Ip: FastMap: A Fast Algorithm for Indexing, Data-mining and Visualization of Traditional and Multimedia Datasets. In: ***Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*** 24 (1995), Nr. 2

[FMW08] Filzmoser, Peter ; Maronna, Ricardo ; Werner, Mark: Outlier identification in high dimensions. In: ***Computational Statistics & Data Analysis*** 52 (2008), Nr. 3

[FPSS96] Fayyad, Usama ; Piatetsky-Shapiro, Gregory ; Smyth, Padhraic: From data mining to knowledge discovery in databases. In: ***AI magazine*** 17 (1996), Nr. 3

[GK01] Greenwald, Michael ; Khanna, Sanjeev: Space-Efficient Online Computation of Quantile Summaries. In: ***Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)***, 2001

[GNC99] Goil, Sanjay ; Nagesh, Harsha ; Choudhary, Alok: MAFIA: Efficient and scalable subspace clustering for very large data sets. In: ***Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)***, 1999

[GP06] Gama, João ; Pinto, Carlos: Discretization from Data Streams: Applications to Histograms and Data Mining. In: ***Proceedings of the ACM Symposium on Applied Computing***, 2006

[GPO08] Ghoting, Amol ; Parthasarathy, Srinivasan ; Otey, Matthew E.: Fast Mining of Distance-based Outliers in High-dimensional Datasets. In: ***Data Min. Knowl. Discov.*** 16 (2008), Nr. 3

[HA04] Hodge, Victoria J. ; Austin, Jim: A Survey of Outlier Detection Methodologies. In: ***Artificial Intelligence Review*** 22 (2004), Nr. 2

[Haw80] Hawkins, Douglas M.: ***Identification of outliers***. Bd. 11. 1980

[HQYY12] Huang, Hao ; Qin, Hong ; Yoo, Shinjae ; Yu, Dantong: Local anomaly descriptor: a robust unsupervised algorithm for anomaly detection based on diffusion space. In: ***Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)***, 2012 (CIKM '12)

[HS10] Hachiya, Hirotaka ; Sugiyama, Masashi: Feature Selection for Reinforcement Learning: Evaluating Implicit State-reward Dependency via Conditional Mutual Information. In: ***Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)***, 2010

[Hö14] Höppner, Frank: A Subspace Filter Supporting the Discovery of Small Clusters in Very Noisy Datasets. In: ***Proceedings of the 26th International Conference on Scientific and Statistical Database Management***, 2014 (SSDBM '14)

[ISML+13] Iglesias Sánchez, Patricia ; Müller, Emmanuel ; Laforet, Fabian ; Keller, Fabian ; Böhm, Klemens: Statistical Selection of Congruent Subspaces for Mining Attributed Graphs. In: ***Proceedings of the IEEE International Conference on Data Mining (ICDM)***, 2013

[ISS03] Iwerks, Glenn S. ; Samet, Hanan ; Smith, Ken: Continuous K-nearest Neighbor Queries for Continuously Moving Points with Updates. In: ***Proceedings of the International Conference on Very Large Data Bases (VLDB)***, 2003

[Jol86] Joliffe, I.: ***Principal Component Analysis***. 1986

[JYX13] Ji, Bo ; Ye, Yang-Dong ; Xiao, Yu: Mutual information evaluation: A way to predict the performance of feature weighting on clustering. In: ***Intelligent Data Analysis*** 17 (2013), Nr. 6

[KA13] Kinney, Justin B. ; Atwal, Gurinder S.: Equitability, mutual information, and the maximal information coefficient. 2013 (1301.7745). – arXiv e-print

[KBG+07] Khan, Shiraj ; Bandyopadhyay, Sharba ; Ganguly, Auroop R. ; Saigal, Sunil ; Erickson, David J. III ; Protopopescu, Vladimir ; Ostrouchov, George: Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data. In: ***Physical Review E*** 76 (2007), Nr. 2

[KKK04] Kailing, Karin ; Kriegel, Hans-Peter ; Kröger, Peer: Density-connected subspace clustering for high-dimensional data. In: ***Proceedings of the SIAM International Conference on Data Mining (SDM)*** Bd. 4, 2004

[KKKW03] Kailing, Karin ; Kriegel, Hans-Peter ; Kröger, Peer ; Wanka, Stefanie: Ranking Interesting Subspaces for Clustering High Dimensional Data. In: ***Knowledge Discovery in Databases: PKDD 2003***. 2003 (Lecture Notes in Computer Science 2838)

[KKSZ09] Kriegel, Hans-Peter ; Kröger, Peer ; Schubert, Erich ; Zimek, Arthur: Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data. In: ***Advances in Knowledge Discovery and Data Mining***. 2009 (5476)

[KKSZ11] Kriegel, Hans-Peter ; Kröger, Peer ; Schubert, Erich ; Zimek, Arthur: Interpreting and Unifying Outlier Scores. In: ***Proceedings of the SIAM International Conference on Data Mining (SDM)***, 2011

[KKSZ12] Kriegel, H. ; Kroger, P. ; Schubert, E. ; Zimek, A.: Outlier Detection in Arbitrarily Oriented Subspaces. In: ***Proceedings of the IEEE International Conference on Data Mining (ICDM)***, 2012

[KKZ09] Kriegel, Hans-Peter ; Kröger, Peer ; Zimek, Arthur: Clustering High-dimensional Data: A Survey on Subspace Clustering, Pattern-based Clustering, and Correlation Clustering. In: ***ACM Transactions on Knowledge Discovery from Data (TKDD)*** 3 (2009), Nr. 1

[KL87] Kozachenko, L. F. ; Leonenko, Nikolai N.: Sample estimate of the entropy of a random vector. In: ***Problemy Peredachi Informatsii*** 23 (1987), Nr. 2

[KMB12] Keller, Fabian ; Müller, Emmanuel ; Böhm, Klemens: HiCS: High Contrast Subspaces for Density-Based Outlier Ranking. In: ***Proceedings of the IEEE International Conference on Data Engineering (ICDE)***, 2012

[KMB15] Keller, Fabian ; Müller, Emmanuel ; Böhm, Klemens: Estimating Mutual Information on Data Streams. In: ***Proceedings of the 27th International Conference on Scientific and Statistical Database Management***, 2015 (SSDBM '15)

[KMWB13] Keller, Fabian ; Müller, Emmanuel ; Wixler, Andreas ; Böhm, Klemens: Flexible and adaptive subspace search for outlier analysis. In: ***Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)***, 2013 (CIKM '13)

[KN98] Knorr, Edwin M. ; Ng, Raymond T.: Algorithms for Mining Distance-Based Outliers in Large Datasets. In: ***Proceedings of the 24rd International Conference on Very Large Data Bases***, 1998 (VLDB '98)

[KN99] Knorr, Edwin M. ; Ng, Raymond T.: Finding Intensional Knowledge of Distance-Based Outliers. In: ***Proceedings of the 25th International Conference on Very Large Data Bases***, 1999 (VLDB '99)

[Kri09] Krippendorff, Klaus: Information of interactions in complex systems. In: ***International Journal of General Systems*** 38 (2009), Nr. 6

[KSG04] Kraskov, Alexander ; Stögbauer, Harald ; Grassberger, Peter: Estimating mutual information. In: ***Physical Review E*** 69 (2004), Nr. 6

[KShZ08] Kriegel, Hans-Peter ; S hubert, Matthias ; Zimek, Arthur: Angle-based Outlier Detection in High-dimensional Data. In: ***Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining***, 2008 (KDD '08)

[LB08] Loekito, Elsa ; Bailey, James: Mining Influential Attributes That Capture Class and Group Contrast Behaviour. In: ***Proceedings of the 17th ACM Conference on Information and Knowledge Management***, 2008 (CIKM '08)

[LK05] Lazarevic, Aleksandar ; Kumar, Vipin: Feature Bagging for Outlier Detection. In: ***Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining***, 2005 (KDD '05)

[LLR83] Leadbetter, M. R. ; Lindgren, Georg ; Rootzén, Holger: Extremes and related properties of random sequences and processes. (1983)

[LLZG10] Li, Hua ; Lv, Gui-Wen ; Zhang, Su-Juan ; Guo, Zhi-Fang: Using mutual information for fuzzy decision tree generation. In: ***Proceedings of the International Conference on Machine Learning and Computing (ICMLC)*** Bd. 1, 2010

[LSO+06] Lall, Ashwin ; Sekar, Vyas ; Ogihara, Mitsunori ; Xu, Jun ; Zhang, Hui: Data Streaming Algorithms for Estimating Entropy of Network Traffic. In: ***ACM SIGMETRICS***, 2006

[LTZ08] Liu, F.T. ; Ting, Kai M. ; Zhou, Zhi-Hua: Isolation Forest. In: ***Eighth IEEE International Conference on Data Mining, 2008. ICDM '08***, 2008

[Lux07] Luxburg, Ulrike: A tutorial on spectral clustering. In: ***Statistics and Computing*** 17 (2007), Nr. 4

[MAG+09] Müller, Emmanuel ; Assent, Ira ; Günnemann, Stephan ; Krieger, Ralph ; Seidl, Thomas: Relevant Subspace Clustering: Mining the Most Interesting Non-redundant Concepts in High Dimensional Data. In: ***IEEE International Conference on Data Mining (ICDM)***, 2009

[MAIS+12] Müller, Emmanuel ; Assent, Ira ; Iglesias Sánchez, Patricia ; Mülle, Yvonne ; Böhm, Klemens: Outlier Ranking via Subspace Analysis in Multiple Views of the Data. In: ***Proceedings of the IEEE International Conference on Data Mining (ICDM)***, 2012 (ICDM '12)

[MASS08] Müller, Emmanuel ; Assent, Ira ; Steinhausen, Uwe ; Seidl, Thomas: OutRank: ranking outliers in high dimensional data. In: ***IEEE 24th International Conference on Data Engineering Workshop, 2008. ICDEW 2008***, 2008

[MKBB12] Müller, Emmanuel ; Keller, Fabian ; Blanc, Sebastian ; Böhm, Klemens: OutRules: A Framework for Outlier Descriptions in Multiple Context Spaces. In: ***Machine Learning and Knowledge Discovery in Databases***. 2012 (7524)

[MMM14] Murrell, Ben ; Murrell, Daniel ; Murrell, Hugh: R2-equitability is satisfiable. In: ***Proceedings of the National Academy of Sciences*** 111 (2014), Nr. 21

[MMPP07] Mouratidis, Kyriakos ; Mouratidis, K. ; Papadias, D. ; Papadias, Dimitris: Continuous Nearest Neighbor Queries over Sliding Windows. In: ***IEEE Transactions on Knowledge and Data Engineering (TKDE)*** 19 (2007), Nr. 6

[MPH05] Mouratidis, Kyriakos ; Papadias, Dimitris ; Hadjieleftheriou, Marios: Conceptual Partitioning: An Efficient Method for Continuous Nearest Neighbor Monitoring. In: ***Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)***, 2005

[MS08] Moise, Gabriela ; Sander, Jörg: Finding Non-redundant, Statistically Significant Regions in High Dimensional Data: A Novel Approach to Projected and Subspace Clustering. In: ***Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)***, 2008 (KDD '08)

[MSE06] Moise, Gabriela ; Sander, Jörg ; Ester, Martin: P3C: A robust projected clustering algorithm. In: ***Proceedings of the IEEE International Conference on Data Mining (ICDM)***, 2006

[MSG+10] Müller, E. ; Schiffer, M. ; Gerwert, P. ; Hannen, M. ; Jansen, T. ; Seidl, T.: SOREX: Subspace Outlier Ranking Exploration Toolkit. In: ***Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)***, 2010

[MSS10] Müller, Emmanuel ; Schiffer, Matthias ; Seidl, Thomas: Adaptive Outlierness for Subspace Outlier Ranking. In: ***Proceedings of the 19th ACM International Conference on Information and Knowledge Management***, 2010 (CIKM '10)

[MSS11] Müller, Emmanuel ; Schiffer, Matthias ; Seidl, Thomas: Statistical selection of relevant subspace projections for outlier ranking. In: ***IEEE 27th International Conference on Data Engineering (ICDE)***, 2011

[NDJ10] Niu, Donglin ; Dy, Jennifer G. ; Jordan, Michael I.: Multiple Non-Redundant Spectral Clustering Views. In: ***Proceedings of the International Conference on Machine Learning (ICML)***, 2010

[NMB13] Nguyen, Hoang V. ; Müller, Emmanuel ; Böhm, Klemens: 4S: Scalable subspace search scheme overcoming traditional Apriori processing. In: ***IEEE International Conference on Big Data***, 2013

[NMV+13] Nguyen, Hoang V. ; Müller, Emmanuel ; Vreeken, Jilles ; Keller, Fabian ; Böhm, Klemens: CMI: An information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In: ***Proceedings of the SIAM International Conference on Data Mining (SDM)***, 2013

[NOMI10] Nguyen, Minh Q. ; Omiecinski, Edward ; Mark, Leo ; Irani, Danesh: A Fast Randomized Method for Local Density-Based Outlier Detection in High Dimensional Data. In: ***Data Warehousing and Knowledge Discovery***. 2010 (6263)

[Pam] Pamap: ***PAMAP - Physical Activity Monitoring for Aging People***. http://www.pamap.org/

[Pan03] Paninski, Liam: Estimation of Entropy and Mutual Information. In: ***Neural Computation*** 15 (2003), Nr. 6

[PJAM02] Procopiuc, Cecilia M. ; Jones, Michael ; Agarwal, Pankaj K. ; Murali, T. M.: A Monte Carlo Algorithm for Fast Projective Clustering. In: ***Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)***, 2002 (SIGMOD '02)

[PKGF03] Papadimitriou, S. ; Kitagawa, H. ; Gibbons, P.B. ; Faloutsos, C.: LOCI: fast outlier detection using the local correlation integral. In: ***19th International Conference on Data Engineering, 2003. Proceedings***, 2003

[PP12] Pham, Ninh ; Pagh, Rasmus: A Near-linear Time Approximation Algorithm for Angle-based Outlier Detection in High-dimensional Data. In: ***Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining***, 2012 (KDD '12)

[PSMP07] Panzeri, Stefano ; Senatore, Riccardo ; Montemurro, Marcelo A. ; Petersen, Rasmus S.: Correcting for the Sampling Bias Problem in Spike Train Information Measures. In: ***Journal of Neurophysiology*** 98 (2007), Nr. 3

[Qiu12] Qiu, Zhi-Wei: Mutivariable mutual information based feature selection for electricity price forecasting. In: ***Proceedings of the International Conference on Machine Learning and Computing (ICMLC)*** Bd. 1, 2012

[RL87] Rousseeuw, P.J. ; Leroy, A.M.: ***Robust Regression and Outlier Detection***. 1987

[RRF+11] Reshef, David N. ; Reshef, Yakir A. ; Finucane, Hilary K. ; Grossman, Sharon R. ; McVean, Gilean ; Turnbaugh, Peter J. ; Lander, Eric S. ; Mitzenmacher, Michael ; Sabeti, Pardis C.: Detecting Novel Associations in Large Data Sets. In: ***Science*** 334 (2011), Nr. 6062

[Sat46] Satterthwaite, F. E.: An approximate distribution of estimates of variance components. In: ***Biometrics Bulletin*** 2 (1946), Nr. 6

[SC08] Singer, Amit ; Coifman, Ronald R.: Non-linear independent component analysis with diffusion maps. In: ***Applied and Computational Harmonic Analysis*** 25 (2008), Nr. 2

[Sch04] Schürmann, Thomas: Bias Analysis in Entropy Estimation. 2004 (cond-mat/0403192). – arXiv e-print

[She00] Shearer, Colin: The CRISP-DM model: the new blueprint for data mining. In: ***Journal of data warehousing*** 5 (2000), Nr. 4

[SN96] Strang, Gilbert ; Nguyen, Truong: ***Wavelets and filter banks***. 1996

[Spe87] Spearman, C.: The proof and measurement of association between two things. In: ***American J. of Psych.*** 15 (1987), Nr. 1

[SS12] Suzuki, Taiji ; Sugiyama, Masashi: Sufficient Dimension Reduction via Squared-Loss Mutual Information Estimation. In: ***Neural Computation*** 25 (2012), Nr. 3

[Ste70] Stephens, M.: Use of the Kolmogorov-Smirnov, Cramer-von Mises and related statistics without extensive tables. In: ***J. of the Royal Stat. Society*** (1970)

[Sto] Stooq: ***Stooq***. http://stooq.com/

[SV11] Smets, Koen ; Vreeken, Jilles: The Odd One Out: Identifying and Characterising Anomalies. In: ***Proceedings of the SIAM International Conference on Data Mining (SDM)*** Bd. 108, 2011

[SWS+00] Schölkopf, Bernhard ; Williamson, Robert C. ; Smola, Alex J. ; Shawe-Taylor, John ; Platt, John C.: Support Vector Method for Novelty Detection. In: ***Advances in Neural Information Processing Systems 12***. 2000

[SZ04] Sequeira, K. ; Zaki, M.: SCHISM: a new approach for interesting subspace mining. In: ***Proceedings of the IEEE International Conference on Data Mining (ICDM)***, 2004

[TSL00] Tenenbaum, Joshua B. ; Silva, Vin d. ; Langford, John C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. In: ***Science*** 290 (2000), Nr. 5500

[VCH10] Vries, T. de ; Chawla, S. ; Houle, M.E.: Finding Local Anomalies in Very High Dimensional Space. In: ***2010 IEEE 10th International Conference on Data Mining (ICDM)***, 2010

[WPT11] Wang, Ye ; Parthasarathy, S. ; Tatikonda, S.: Locality Sensitive Outlier Detection: A ranking driven approach. In: ***2011 IEEE 27th International Conference on Data Engineering (ICDE)***, 2011

[Wro97] Wrobel, Stefan: An Algorithm for Multi-relational Discovery of Subgroups. In: ***Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery***, 1997 (PKDD '97)

[WWL09] Walters-Williams, Janett ; Li, Yan: Estimation of Mutual Information: A Survey. In: ***Rough Sets and Knowledge Technology***. 2009 (5589)

[XMA05] Xiong, Xiaopeng ; Mokbel, M.F. ; Aref, W.G.: SEA-CNN: scalable processing of continuous k-nearest neighbor queries in spatio-temporal databases. In: ***Proceedings of the IEEE International Conference on Data Engineering (ICDE)***, 2005

[XSTK04] Xiong, Hui ; Shekhar, Shashi ; Tan, Pang-Ning ; Kumar, Vipin: Exploiting a Support-based Upper Bound of Pearson's Correlation Coefficient for Efficiently Identifying Strongly Correlated Pairs. In: ***Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)***, 2004

[YPK05] Yu, X. ; Pu, K.Q. ; Koudas, N.: Monitoring k-Nearest Neighbor Queries over Moving Objects. In: ***Proceedings of the IEEE International Conference on Data Engineering (ICDE)***, 2005

[ZX08] Zhou, Wenjun ; Xiong, Hui: Volatile Correlation Computation: A Checkpoint View. In: ***Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)***, 2008

[ZX11] Zhou, Wenjun ; Xiong, Hui: Checkpoint evolution for volatile correlation computing. In: ***Machine Learning*** 83 (2011), Nr. 1