Analyzing Complex Data Using Domain Constraints

Markus Mauder

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Markus Mauder
aus München

München, den 29.03.2017

Analyzing Complex Data Using Domain Constraints

Markus Mauder

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Markus Mauder
aus München

München, den 29.03.2017

Erstgutachter: Prof. Dr. Peer Kröger
Zweitgutachter: Prof. Dr. Bernhard Seeger
Tag der mündlichen Prüfung: 19.06.2017

Eidesstattliche Versicherung

Hiermit erkläre ich, Markus Mauder, gemäß § 8 Abs. 2 Pkt. 5 der Promotionsordnung vom 12.07.2011, an Eidesstatt, dass die vorliegende Dissertation von mir selbständig, ohne unerlaubte Beihilfe angefertigt ist.

München, 29.03.2017

Contents

List of Figures

List of Tables

Abstract

Data-driven research approaches are becoming increasingly popular in a growing number of scientific disciplines. While a data-driven research approach can yield superior results, generating the required data can be very costly. This frequently leads to small and complex data sets, in which it is impossible to rely on volume alone to compensate for all shortcomings of the data. To counter this problem, other reliable sources of information must be incorporated. In this work, domain knowledge, as a particularly reliable type of additional information, is used to inform data-driven analysis methods. This domain knowledge is represented as constraints on the possible solutions, which the presented methods can use to inform their analysis. It focusses on spatial constraints as a particularly common type of constraint, but the proposed techniques are general enough to be applied to other types of constraints.

In this thesis, new methods using domain constraints for data-driven science applications are discussed. These methods have applications in feature evaluation, route database repair, and Gaussian Mixture modeling of spatial data. The first application focuses on feature evaluation. The presented method receives two representations of the same data: one as the intended target and the other for investigation. It calculates a score indicating how much the two representations agree. A presented application uses this technique to compare a reference attribute set with different subsets to determine the importance and relevance of individual attributes.

A second technique analyzes route data for constraint compliance. The presented framework allows the user to specify constraints and possible actions to modify the data. The presented method then uses these inputs to generate a version of the data, which agrees with the constraints, while otherwise reducing the impact of the modifications as much as possible. Two extensions of this schema are presented: an extension to continuously valued costs, which are minimized, and an extension to constraints involving more than one moving object.

Another addressed application area is modeling of multivariate measurement data, which was measured at spatially distributed locations. The spatial information recorded with the data can be used as the basis for constraints. This thesis presents multiple approaches to building a model of this kind of data while complying with spatial constraints. The first approach is an interactive tool, which allows domain scientists to generate a model of the data, which complies with their knowledge about the data. The second is a Monte Carlo approach, which generates a large number of possible models, tests them for

compliance with the constraints, and returns the best one. The final two approaches are based on the EM algorithm and use different ways of incorporating the information into their models.

At the end of the thesis, two applications of the models, which have been generated in the previous chapter, are presented. The first is prediction of the origin of samples and the other is the visual representation of the extracted models on a map. These tools can be used by domain scientists to augment their tried and tested tools.

The developed techniques are applied to a real-world data set collected in the archaeo-biological research project FOR 1670 (*Transalpine mobility and cultural transfer*)[1] of the German Science Foundation. The data set contains isotope ratio measurements of samples, which were discovered at archaeological sites in the Alps region of central Europe. Using the presented data analysis methods, the data is analyzed to answer relevant domain questions. In a first application, the attributes of the measurements are analyzed for their relative importance and their ability to predict the spatial location of samples. Another presented application is the reconstruction of potential migration routes between the investigated sites. Then spatial models are built using the presented modeling approaches. Univariate outliers are determined and used to predict locations based on the generated models. These are cross-referenced with the recorded origins. Finally, maps of the isotope distribution in the investigated regions are presented.

The described methods and demonstrated analyses show that domain knowledge can be used to formulate constraints that inform the data analysis process to yield valid models from relatively small data sets and support domain scientists in their analyses.

[1] `http://www.en.for1670-transalpine.uni-muenchen.de/`

Zusammenfassung

Datengetriebene Forschungsansätze werden für eine wachsende Anzahl von wissenschaftlichen Disziplinen immer wichtiger. Obwohl ein datengetriebener Forschungsansatz bessere Ergebnisse erzielen kann, kann es sehr teuer sein die notwendigen Daten zu gewinnen. Dies hat häufig zur Folge, dass kleine und komplexe Datensätze entstehen, bei denen es nicht möglich ist sich auf die Menge der Datenpunkte zu verlassen um Probleme bei der Analyse auszugleichen.

Um diesem Problem zu begegnen müssen andere Informationsquellen verwendet werden. Fachwissen als eine besonders zuverlässige Quelle solcher Informationen kann herangezogen werden, um die datengetriebenen Analysemethoden zu unterstützen. Dieses Fachwissen wird ausgedrückt als Constraints (Nebenbedingungen) der möglichen Lösungen, die die vorgestellten Methoden benutzen können um ihre Analyse zu steuern. Der Fokus liegt dabei auf räumlichen Constraints als eine besonders häufige Art von Constraints, aber die vorgeschlagenen Methoden sind allgemein genug um auf andere Arte von Constraints angewendet zu werden.

Es werden neue Methoden diskutiert, die Fachwissen für datengetriebene wissenschaftliche Anwendungen verwenden. Diese Methoden haben Anwendungen auf Feature-Evaluation, die Reparatur von Bewegungsdatenbanken und auf Gaussian-Mixture-Modelle von räumlichen Daten. Die erste Anwendung betrifft Feature-Evaluation. Die vorgestellte Methode erhält zwei Repräsentationen der selben Daten: eine als Zielrepräsentation und eine zur Untersuchung. Sie berechnet einen Wert, der aussagt, wie einig sich die beiden Repräsentationen sind. Eine vorgestellte Anwendung benutzt diese Technik um eine Referenzmenge von Attributen mit verschiedenen Untermengen zu vergleichen, um die Wichtigkeit und Relevanz einzelner Attribute zu bestimmen.

Eine zweite Technik analysiert die Einhaltung von Constraints in Bewegungsdaten. Das präsentierte Framework erlaubt dem Benutzer Constraints zu definieren und mögliche Aktionen zur Veränderung der Daten anzuwenden. Die präsentierte Methode benutzt diese Eingaben dann um eine neue Variante der Daten zu erstellen, die die Constraints erfüllt ohne die Datenbank mehr als notwendig zu verändern. Zwei Erweiterungen dieser Grundidee werden vorgestellt: eine Erweiterung auf stetige Kostenfunktionen, die minimiert werden, und eine Erweiterung auf Bedingungen, die mehr als ein bewegliches Objekt betreffen.

Ein weiteres behandeltes Anwendungsgebiet ist die Modellierung von multivariaten Messungen, die an räumlich verteilten Orten gemessen wurden. Die räumliche Information, die zusammen mit diesen Daten erhoben wurde, kann als Grundlage genutzt werden um

Constraints zu formulieren. Mehrere Ansätze zum Erstellen von Modellen auf dieser Art von Daten werden vorgestellt, die räumliche Constraints einhalten. Der erste dieser Ansätze ist ein interaktives Werkzeug, das Fachwissenschaftlern dabei hilft, Modelle der Daten zu erstellen, die mit ihrem Wissen über die Daten übereinstimmen. Der zweite ist eine Monte-Carlo-Simulation, die eine große Menge möglicher Modelle erstellt, testet ob sie mit den Constraints übereinstimmen und das beste Modell zurückgeben. Zwei letzte Ansätze basieren auf dem EM-Algorithmus und benutzen verschiedene Arten diese Information in das Modell zu integrieren.

Am Ende werden zwei Anwendungen der gerade vorgestellten Modelle vorgestellt. Die erste ist die Vorhersage der Herkunft von Proben und die andere ist die grafische Darstellung der erstellten Modelle auf einer Karte. Diese Werkzeuge können von Fachwissenschaftlern benutzt werden um ihre bewährten Methoden zu unterstützen.

Die entwickelten Methoden werden auf einen realen Datensatz angewendet, der von dem archäo-biologischen Forschungsprojekt FOR 1670 (*Transalpine Mobilität und Kulturtransfer*[2]) der Deutschen Forschungsgemeinschaft erhoben worden ist. Der Datensatz enthält Messungen von Isotopenverhältnissen von Proben, die in archäologischen Fundstellen in den zentraleuropäischen Alpen gefunden wurden. Die präsentierten Datenanalyse-Methoden werden verwendet um diese Daten zu analysieren und relevante Forschungsfragen zu klären. In einer ersten Anwendung werden die Attribute der Messungen analysiert um ihre relative Wichtigkeit und ihre Fähigkeit zu bewerten, die räumliche Herkunft der Proben vorherzusagen. Eine weitere vorgestellte Anwendung ist die Wiederherstellung von möglichen Migrationsrouten zwischen den untersuchten Fundstellen. Danach werden räumliche Modelle der Daten unter Verwendung der vorgestellten Methoden erstellt. Univariate Outlier werden bestimmt und ihre möglich Herkunft basierend auf der erstellten Karte wird bestimmt. Die vorhergesagte Herkunft wird mit der tatsächlichen Fundstelle verglichen. Zuletzt werden Karten der Isotopenverteilung der untersuchten Region vorgestellt.

Die beschriebenen Methoden und vorgestellten Analysen zeigen, dass Fachwissen verwendet werden kann um Constraints zu formulieren, die den Datenanalyseprozess unterstützen, um gültige Modelle aus relativ kleinen Datensätzen zu erstellen und Fachwissenschaftler bei ihren Analysen zu unterstützen.

[2] `http://www.for1670-transalpine.uni-muenchen.de/`

Chapter 1

Introduction

Attribution

This chapter uses material from the following publications:

- M. Mauder, E. Ntoutsi, and P. Kröger. Influence of oxygen isotope ratio on classification. Technical report, FOR1670: Transalpine mobility and cultural transfer, 2014
- M. Mauder, E. Ntoutsi, P. Kröger, and G. Grupe. Data mining for isotopic mapping of bioarchaeological finds in a central European Alpine passage. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*, page 34. ACM, 2015
- M. Mauder, E. Ntoutsi, P. Kröger, C. Mayr, G. Grupe, A. Toncala, and S. Hölzl. Applying data mining methods for the analysis of stable isotope data in bioarchaeology. In *2016 IEEE 12th International Conference on eScience*, 2016

The archaeological database *transmo* (see Section 1.2.1.2) was designed in cooperation with Alexander Thoelke and Andrej Wallwitz.
See Section 1.3 for a detailed overview of incorporated publications.

The ever increasing capacity for recording, analyzing, and evaluating scientific measurements has led researchers from more and more subject areas to start collecting data in an effort to improve their processes and findings [71]. As a consequence, scientists in many areas now find themselves faced with relatively large amounts of data [34]. The availability and the need to analyze this data is confronting domain scientists with cultural, technological, and methodological challenges [10]. While data driven research offers many opportunities, it also asks for a different approach to research. With this approach comes

a requirement for data science and statistics skills in addition to the knowledge and skills required in any scientific field. Requiring an additional set of skills is a lot to ask and may not always be feasible.

There are various approaches to deal with this problem [73]. An easy solution is to keep the analysis simple. Easy to use tools, which have sprawled in recent years, allow domain experts with a limited background in statistics or informatics to perform some of their analysis work independently [60, 31, 35]. However, this ease of use often comes at the cost of reduced complexity of the resulting models (e.g. linear correlations only).

Another solution is to add an expert in data analysis to the research team. The role of the data scientist in this new scientific ecosystem is to translate between the requirements of the domain scientist and the available analysis technologies [16]. Some research projects have now begun working with data scientists (see Section 1.2.1), but for smaller research projects (and if the trend continuous increasingly for larger ones, too) there just are not enough data scientists available[1]. However, when data scientists are not familiar with the data analysis process on the domain experts' side, the translation of domain experts' knowledge into appropriate data models can be a slow and inefficient process.

If large amounts of data are available, more complex models can be trained with little user intervention. However, this approach is limited to this relatively rare case and confronts the user with a host of new problems (like scalability). Also, relying on the data to specify a complex model requires there to be only one valid interpretation of the data.

A more common scenario in areas that have only recently begun to consider data analysis as a tool, are small data sets that are generated after careful planning and analysis of relationships. In small data sets it is not practical to rely on the assumption that random errors are rare enough to be marginalized by the rest of the data.

The solution this thesis explores is to use domain knowledge to formulate desirable solution properties and use them to guide algorithms. Domain experts have rich knowledge of their area of expertise, which is reflected in relationships in the collected data. If these relationships can be formalized into properties of a valid model, this information can then be used to guide automatic data analysis.

This thesis introduces various ways, in which data science problems can be specified and solved in a way that ensures the solution complies with properties, which were specified by domain experts. We call these properties *domain constraints* (or *constraints* where the context is clear). These constraints can be specified by users and – if incorporated properly – can alleviate the need to develop new approaches and analyze the data in ways that require a data analysis expert. Chapter 2 introduces the concept of constraints in more detail.

Following this introduction, three data-driven science applications are discussed: feature evaluation (Chapter 3), trajectory analysis (Chapter 4), and spatial Gaussian Mixture Models (Chapter 5). For each of these applications, an approach to include domain constraints is introduced.

To demonstrate the introduced techniques, each is applied to an example task on a

[1] `http://visit.crowdflower.com/data-science-report.html` – retrieved 2017-3-21

data set from an interdisciplinary research project. The data is annotated with spatial information about the location where it was recorded. This data set is introduced in detail in the following Section.

While the presented techniques are general enough to be used with various other types of constraints, the presented evaluations use spatial data to formulate constraints and improve the built models. We chose spatially distributed data to illustrate the concepts, because spatial data and derived constraints are intuitive and help with a clear description.

In the following section we will look at data on which constraints can be formulated and see the example data set used as an illustration throughout this document.

1.1 Data

The algorithms presented here get data and constraints as input. The way that constraints are passed can differ between algorithms. For practical reasons constraints are usually given as additional information which is evaluated inside the algorithm to derive the information needed to adapt the analysis to the constraints. Constraints can be defined on the input data itself, but that limits the expressiveness of the constraints.

If additional data is used, it can either be categorical (allowing the algorithm to determine whether two points belong to the same group) or continuous (allowing the algorithm to use a distance function to differentiate better results from worse). Vector data can contain information that is not used as additional attributes. Some attributes are then interpreted as data and some as constraints. These sets can possibly overlap, although that is of little practical relevance.

Data sets with additional information are common in scientific data collection. There both measurements and information about the circumstances (such as time and place) of their recording are collected. These combined data sets are hard to analyze given non-specialized analyses because of different semantics underlying different attributes. However, algorithms that are adapted to analyzing a particular type of data may interpret some attributes differently in order to help e.g. build a model of the remaining attributes.

$$\left(\begin{array}{c} f_1 \\ f_2 \\ f_3 \\ s_1 \\ s_2 \end{array}\right) \quad \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \text{measurements} \\ \left.\begin{array}{l} \\ \end{array}\right\} \text{spatial origin} \end{array}$$

Figure 1.1: Typical schema of a data point with associated spatial information.

Spatial (and spatio-temporal) data points contain spatial (and possibly temporal) information and a payload of measurements (the *data*). See Figure 1.1 for a schematic depiction.

Additional information about data can also contain information about the relationship between measurements. Depending on the type of recorded information, the capabilities of the measurement equipment, and the design of the study different types of additional information are available. Measurements and additional information thus combine two or more notions of similarity that can be used by constraints. Constraints as input for analyses specify the expected interaction between measurements, interactions which can be used in the analysis to flag unexpected data or measure the fitness of a data point. With this information more complex models can be built and more realistic information extracted than from the data alone.

Many data mining tasks can profit from added information. Unsupervised data mining tasks like clustering can gain additional information about plausible clusterings from spatial distance, clustering models can be evaluated using spatial data as a baseline, sudden changes in additional information can indicate outliers, and so forth.

In the experiments below we will mostly concentrate on constraints derived from spatial information. The data set that is used to illustrate the presented algorithms contains additional data about the spatial origin of the data.

The following section introduces the spatial data set that will be used to evaluate the algorithms below.

1.2 FOR 1670 Isotope Data Set

The presented techniques are applied to a bio-archaeological data set generated by the interdisciplinary research group FOR 1670. The data set is used here for illustrative purposes only. The presented techniques can be applied to a wide range of problems. The FOR 1670 data set is typical for the type of data to which constraint techniques can be applied, because it is relatively small, but contains diverse measurements with complex interactions.

1.2.1 Research Project FOR 1670

The goal of research group FOR 1670 [26] is the construction of a large scale isotopic map of the reference region, the Inn-Eisack-Adige transect via the Brenner pass in the European Alps. This area covers a long distance from northern Italy (around Bolzano) to southern Germany (around Munich) and is based on a collection of data from approximately 30 sites along that route. The envisioned isotopic map will represent the common, local isotopic signatures (sometimes called *fingerprints*) characteristic for a given spatial region. The application of this map will help to differentiate between local finds and non-local finds, and for the definition of the place of origin of the latter in order to answer the aforementioned scientific questions regarding mobility, trade as well as cultural transfer. The reason behind this application is that knowledge of the spatial distribution of stable isotopes in the environment allows identifying outliers that represent primarily non-local individuals and predict places of origin of samples.

This map is to be used to answer question about transalpine mobility and cultural transfer in the past. For more details on the project see Grupe et al. [24]. A somewhat shorter overview is given by Grupe et al. [25]. The approach to this goal is to build an isotopic map of the studied region from biological samples. The area under study is the Inn-Eisack-Adige passage across the European Alps. This passage has been used at least since the Mesolithic, which makes it a suitable area to answer questions about mobility in the past.

1.2.1.1 Isotope Analysis

Isotopes of an element are atoms with different numbers of neutrons and thus mass, but otherwise identical properties. Measuring the ratio of different isotopes with each other has many applications, e.g., dating skeletal finds and archaeological sites, for reasoning about diet, climate, and migration patterns [3]. More on general principles and limitations of stable isotope analysis was given by Meier-Augenstein and Kemp [56].

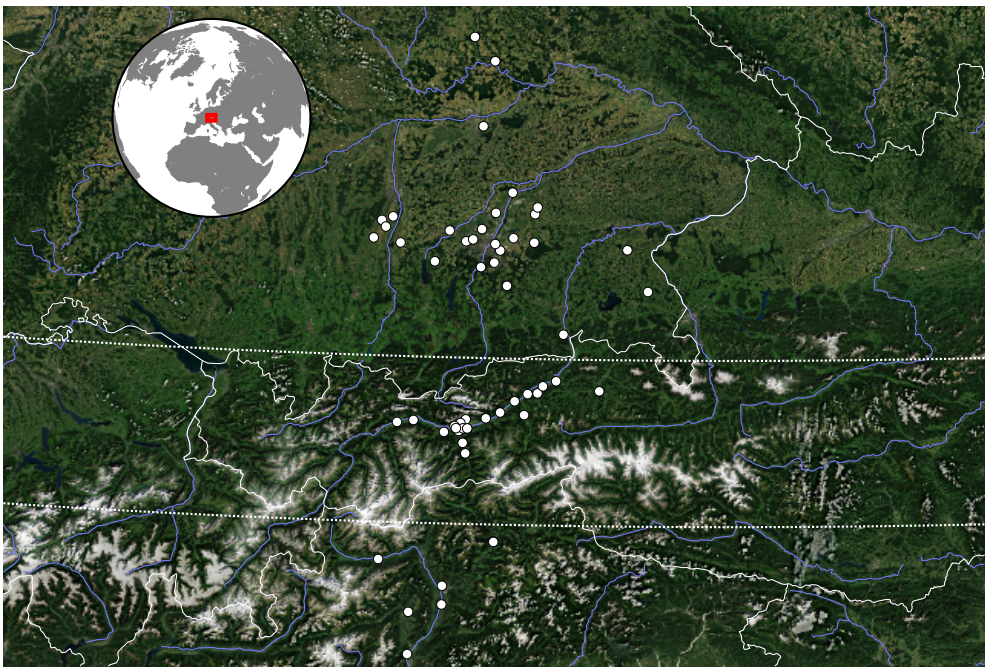


Figure 1.2: Sampling sites across the transalpine Inn-Eisack-Adige passage. Some locations not visible due to overlap. Data from the pictured locations is used in the evaluation in Section 3.5. The dotted lines represent the regional split in North, Center, and South classes used in the data set overview. Satellite imagery ©Earthstar Geographics — Esri, HERE, Garmin.

In bio-archaeology, isotopes are used to predict patterns that characterize the origin of geological and biological materials at a small spatial scale. Isotopic fractionation and mixing in an ecosystem generates compartments with characteristic isotopic signatures. Such

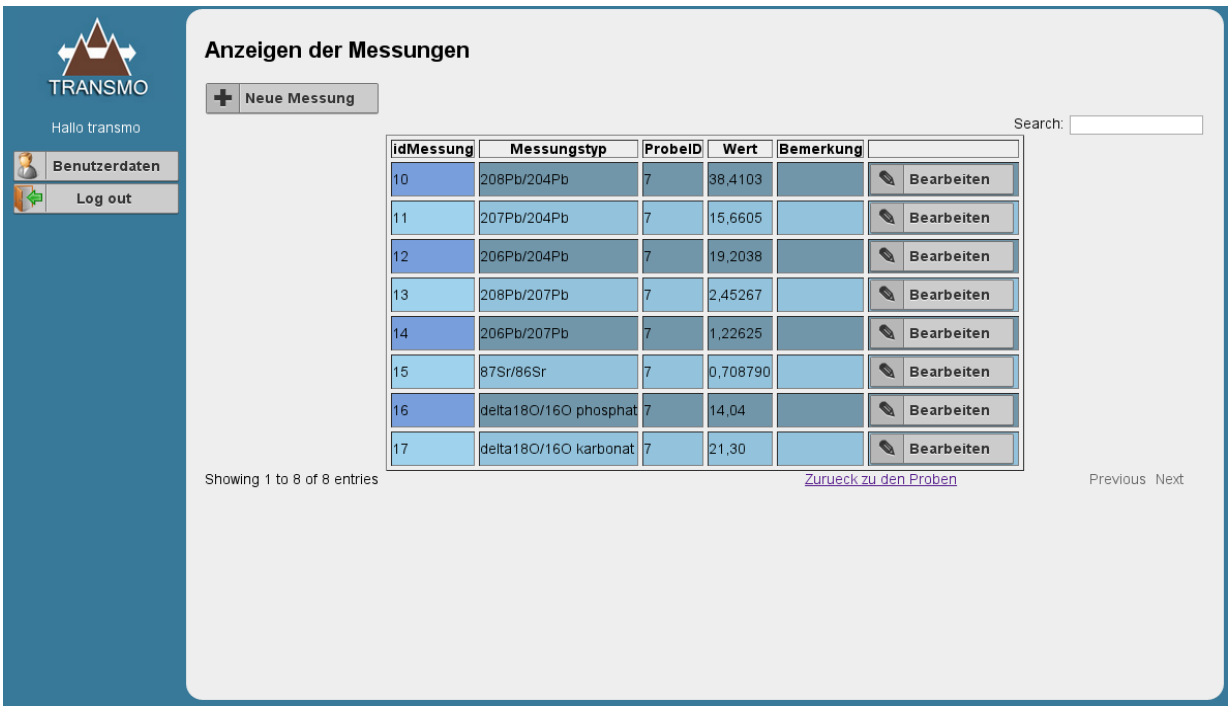


Figure 1.3: Editing a sample's measurements.

isotopic maps can be applied to predict the place of origin of archaeological finds in order to answer diverse archaeological questions concerning migration, trade, etc. Such isotopic maps are empirically generated by sampling the relevant environmental components and measuring their isotopic signatures. However, the vast majority of stable isotope studies in this field are small scale projects that lack the fundamental capabilities of prediction and modeling. FOR 1670 has investigated 60 archaeological sites in its study region (see Figure 1.2) to generate a data set based on which an isotopic map can be derived for this reference region.

This data is to be analyzed and turned into a spatial model (an *isotopic landscape* or *isoscape*) of the investigated region to answer questions about transfer of humans, goods, and culture through the passage. The term *isotopic landscape* describes *maps of isotopic variation produced by iteratively applying (predictive) models across regions of space using grid-based environmental data sets*, whereby one *common use of isoscapes is as a source of estimated isotopic values at unmonitored sites, which can be an important implementation for both local- and global-scale studies if the isoscape is based on a robust and well-studied model* [11]. The isotopic mapping of the transect aims at answering open archaeological questions related to transalpine mobility and culture transfer.

1.2.1.2 Database

Sample collection and measuring of isotope ratios was performed by team of domain experts from a range of fields and frequently shared with the data scientists for ongoing analysis.

To facilitate the sharing, a web-based database access was built. Figure 1.3 shows the measurements belonging to a single sample being edited. Figure 1.4 depicts a rough schema of the underlying database. The subset of the data that is the basis of this thesis is extracted from the *bone sample*, *sample*, *measurement*, and *excavation* tables. The resulting columns are *id* (from *sample*), *latitude*, *longitude*, *altitude* (from *excavation*), *species*, *bone* (from *bone sample*), and multiple isotope ratios (from *measurement*).

This data is described in more detail in the next section.

1.2.2 Attributes

377 samples were analyzed. Of these, most samples were from human remains (162), the second largest group was cow (87), closely followed by pig (80), and finally deer (48). From each investigated specimen, up to seven isotopes were measured: ^{18}O, ^{86}Sr, ^{87}Sr, ^{204}Pb, ^{206}Pb, ^{207}Pb, and ^{208}Pb. Due to technical particularities of isotope measuring, the strontium (Sr) and lead (Pb) isotopes were measured and recorded as fractions of isotopes of the same element, yielding the fractions $^{87}Sr/^{86}Sr$, $^{208}Pb/^{204}Pb$, $^{207}Pb/^{204}Pb$, $^{206}Pb/^{204}Pb$, $^{208}Pb/^{207}Pb$, and $^{206}Pb/^{207}Pb$. The oxygen isotope was normalized against ocean water isotope levels and recorded as $\delta^{18}O$. This yields a 7-dimensional feature vector for each recovered sample. In addition to these isotope measurements, each sample was annotated with a spatial description (latitude, longitude, altitude) based on the discovery area. Also, each sample was labeled with the species from which it was extracted (human or one of the three animal species pig, cattle, and red deer).

There were two types of samples in this study, which different slightly in the recorded attributes: bone samples and cremated samples. All animals and 16 human samples are available as bone samples. These samples can be measured in full and their recorded attributes correspond to the ones listed above. The remaining human samples were from cremated bodies. They are missing oxygen isotope measurements, because oxygen isotopes are not stable at the high temperatures that characterize cremation. The question whether oxygen isotopes are crucial for analysis is investigated in Section 3.2 and Section 3.5. Table 1.1 lists the recorded isotope ratios.

From an analysis perspective the data set, although small, is extremely interesting and challenging because there will not be a perfect fit to a (previously unknown) ground truth for several, already discussed, reasons. First, some of the samples might be diagenetically altered, considering that they were exposed to all kind of environmental conditions for such a long period of time. Second, the isotopic measurements are also subject to instrumental errors. Last but not least, the spatial coordinates of a sample denote the place of death rather than the origin of the corresponding animal. Given that the isotopic concentration within the body depends heavily on nutrition and air, the area where someone spends most of his life has a stronger impact than the place of death on his isotopic fingerprint. This might be especially problematic in case of animal migration or trade.

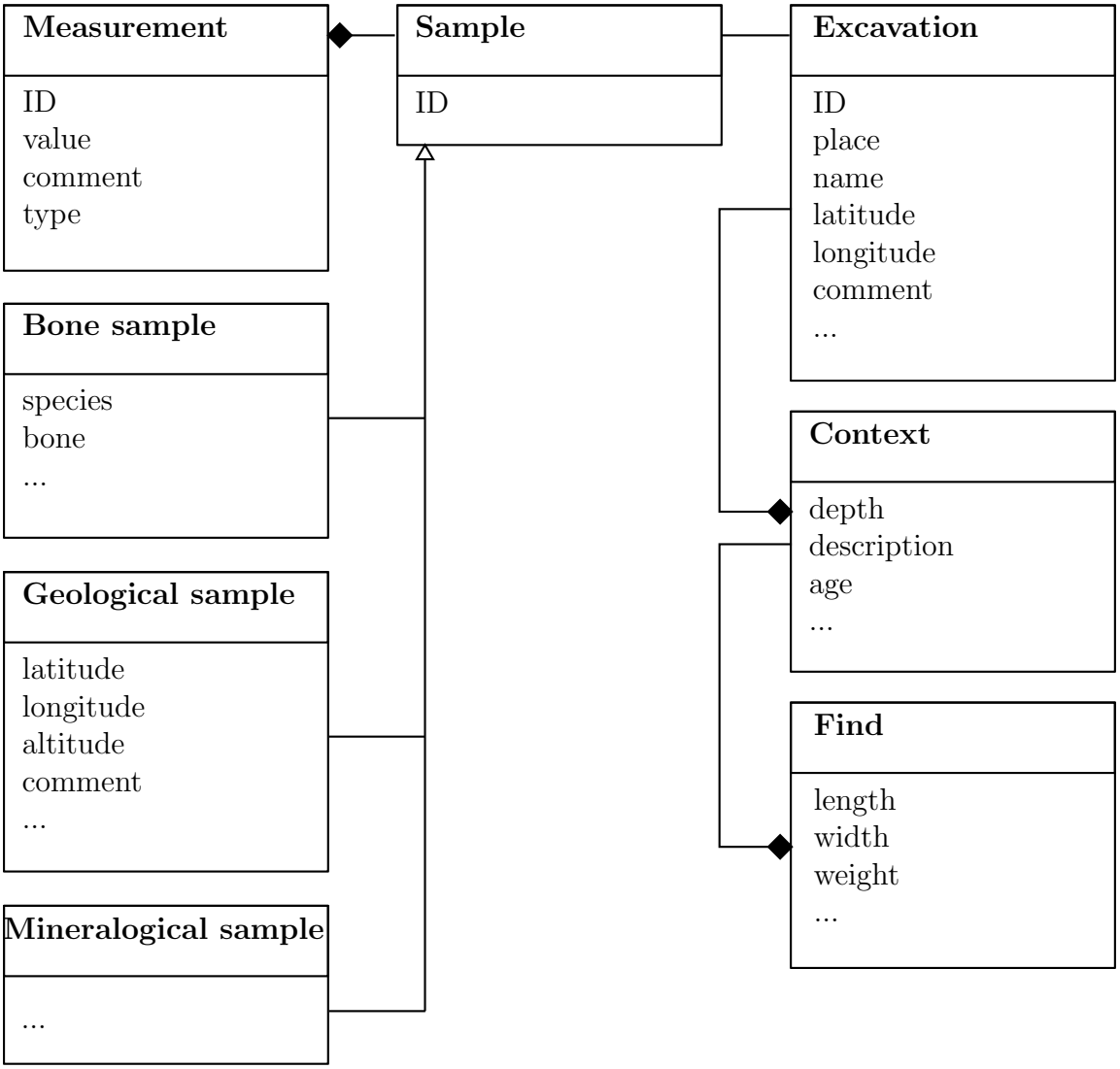


Figure 1.4: Schema of FOR 1670 database. Some relations and attributes omitted for clarity.

Attribute	Description
ffi^{18}O	Ratio of stable Oxygen isotopes ^{18}O and ^{16}O normalized against sea water standard per mil. Only available for uncremated finds.
^{87}Sr/^{86}Sr	Ratio of stable Strontium isotopes.
^{208}Pb/^{204}Pb	Ratio of stable Lead isotopes.
^{207}Pb/^{204}Pb	Ratio of stable Lead isotopes.
^{206}Pb/^{204}Pb	Ratio of stable Lead isotopes.
^{208}Pb/^{207}Pb	Ratio of stable Lead isotopes.
^{206}Pb/^{207}Pb	Ratio of stable Lead isotopes.
longitude (λ)	Geographic east-west coordinates in degrees.
latitude (ϕ)	Geographic north-south coordinates in degrees.
altitude	Geographic height above sea level in meters.
species	Human, pig, cow, or deer.

Table 1.1: Attributes available for each sample.

1.2.3 Isotope Distribution

The assumption underlying isotope fingerprint analysis is that there is a correlation between samples from the same spatial location. The data set used in this study contains multiple locations represented by more than one sample. To be a viable contribution to the identification of a sample's origin, the distribution of isotopes between locations must be distinct. As an example, consider Figure 1.5. It shows several locations' oxygen isotope distributions sorted by latitude. Colors indicate regions north, inside, and south of the Alps (according to the borders in Figure 1.2). The overlap between regions is an indication that the oxygen isotope is not a strong contributor to the spatial association of isotopes and illustrates the difficulty with reasoning about spatial origin in general.

The class distribution for different continuous attributes is shown in Figure 1.6. We can see that the attributes follow different distributions, but all classes cover similar value ranges. Each sample was annotated with *latitude*, *longitude*, and *altitude* attributes of the site at which it was discovered. The human data set contained one location (Latsch) where uncremated human remains were found. This allowed measuring oxygen isotopes for 16 human samples, which are also part of the data set.

Due to each species' diet, metabolism, and other factors, the distribution differs between species. In particular human fingerprints are expected to be different. These same influences vary over different locations and individuals.

An overview of the correlations between different attributes is shown in Figures 1.7 and 1.8. See also Table 1.5 for R^2 correlation measures between isotopes.

Of particular interest for spatial analysis is the correlation (or lack thereof) between data attributes and spatial attributes (see Table 1.2). Indeed, most attributes seem uncorrelated with spatial location. Longitude's correlation with any other attribute is clearly very low

		Latitude	Longitude
Human	$^{206}Pb/^{204}Pb$	0.18	0.03
	$^{206}Pb/^{207}Pb$	0.20	0.03
	$^{207}Pb/^{204}Pb$	0.01	0.00
	$^{208}Pb/^{204}Pb$	0.06	0.00
	$^{208}Pb/^{207}Pb$	0.10	0.00
	$^{87}Sr/^{86}Sr$	0.58	0.15
	^{18}O	0.00	0.00
Cow	$^{206}Pb/^{204}Pb$	0.14	0.05
	$^{206}Pb/^{207}Pb$	0.16	0.04
	$^{207}Pb/^{204}Pb$	0.03	0.05
	$^{208}Pb/^{204}Pb$	0.07	0.03
	$^{208}Pb/^{207}Pb$	0.11	0.00
	$^{87}Sr/^{86}Sr$	0.25	0.03
	^{18}O	0.02	0.04
Pig	$^{206}Pb/^{204}Pb$	0.15	0.01
	$^{206}Pb/^{207}Pb$	0.16	0.00
	$^{207}Pb/^{204}Pb$	0.02	0.01
	$^{208}Pb/^{204}Pb$	0.02	0.00
	$^{208}Pb/^{207}Pb$	0.06	0.01
	$^{87}Sr/^{86}Sr$	0.13	0.05
	^{18}O	0.12	0.00
Deer	$^{206}Pb/^{204}Pb$	0.11	0.00
	$^{206}Pb/^{207}Pb$	0.12	0.00
	$^{207}Pb/^{204}Pb$	0.04	0.00
	$^{208}Pb/^{204}Pb$	0.00	0.03
	$^{208}Pb/^{207}Pb$	0.05	0.02
	$^{87}Sr/^{86}Sr$	0.22	0.05
	^{18}O	0.07	0.02

Table 1.2: r^2 values of spatial and spatial variables in all data.

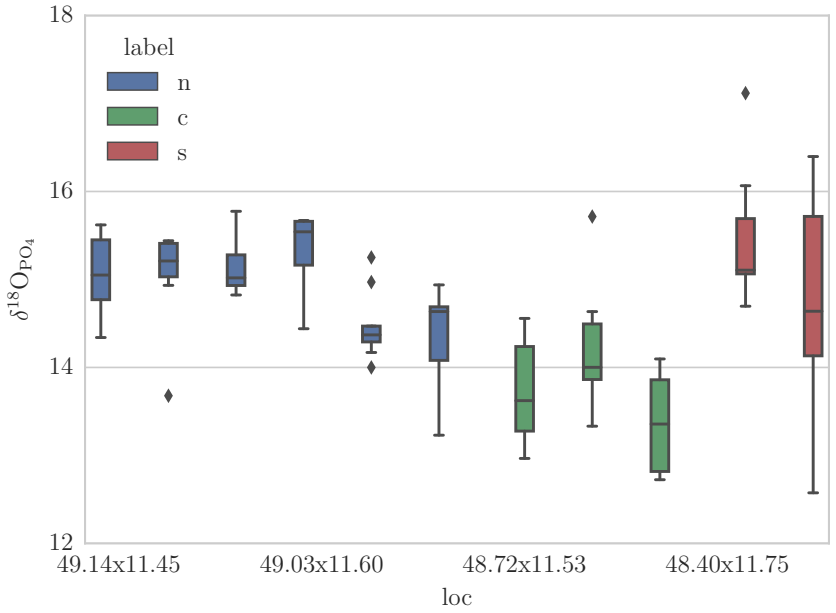


Figure 1.5: Oxygen isotope distribution by location.

with the exception of the human Strontium isotope, which has a still low, but respectable, 0.15 r^2 score. For latitude, only Strontium reaches r^2 values over 0.20 and then only in cows, deer, and humans (Sr vs latitude value in the remaining species (pigs) is 0.13, which is still higher than average). Lead ratios whose numerator is ^{206}Pb have consistent r^2 values over 0.10, while the remaining Lead isotope all come in consistently lower than 0.10. Ignoring scores under 0.10, humans always score higher than other species. Strontium even reaches a remarkable (for this data set) 0.58 score and ^{206}Pb/^{207}Pb has the highest Lead score (followed closely by ^{206}Pb/^{204}Pb at 0.18 also for the human data). Since humans for which δ^{18}O is available are all from the same site, no correlation is detected.

Correlations within lead isotope ratios can be very high. ^{206}Pb/^{204}Pb, ^{206}Pb/^{207}Pb is consistently 1.0. ^{207}Pb/^{204}Pb, ^{206}Pb/^{204}Pb consistently high $0.60 - 0.69$. ^{207}Pb/^{204}Pb, ^{206}Pb/^{207}Pb above 0.5. ^{208}Pb/^{204}Pb, ^{208}Pb/^{207}Pb up to 0.75. Correlations with strontium and oxygen are always low. Highest Strontium/Oxygen value is ^{87}Sr/^{86}Sr, ^{208}Pb/^{207}Pb (deer) at 0.26. See Table 1.5 and Figure 1.9 for details.

1.2.4 Outliers

According to Hawkins definition [33], "[a]n outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism". As the definition says, outliers are suspicious and may be of particular interest to domain scientists. For the outlier detection, we rely on the interquartile range

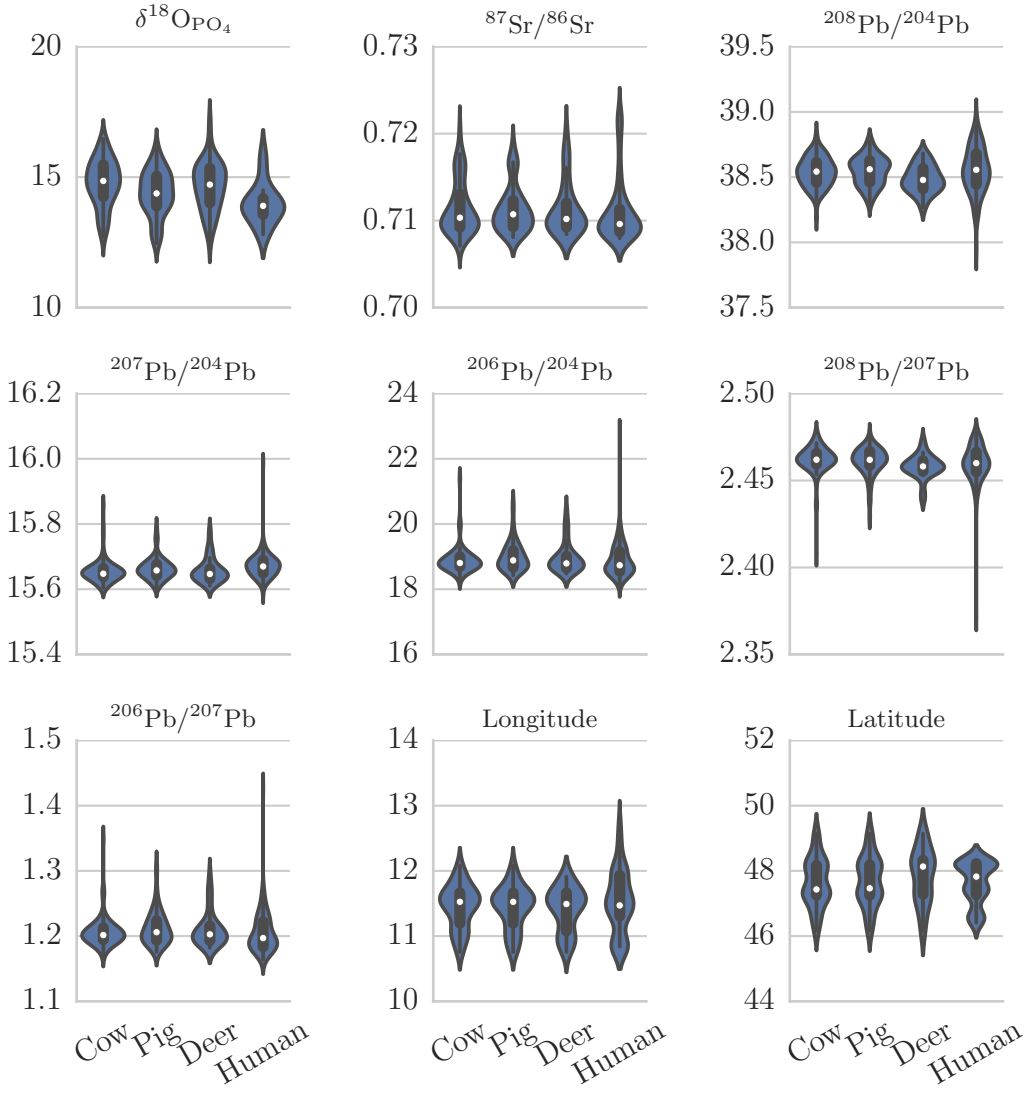


Figure 1.6: Class distribution for the different attributes.

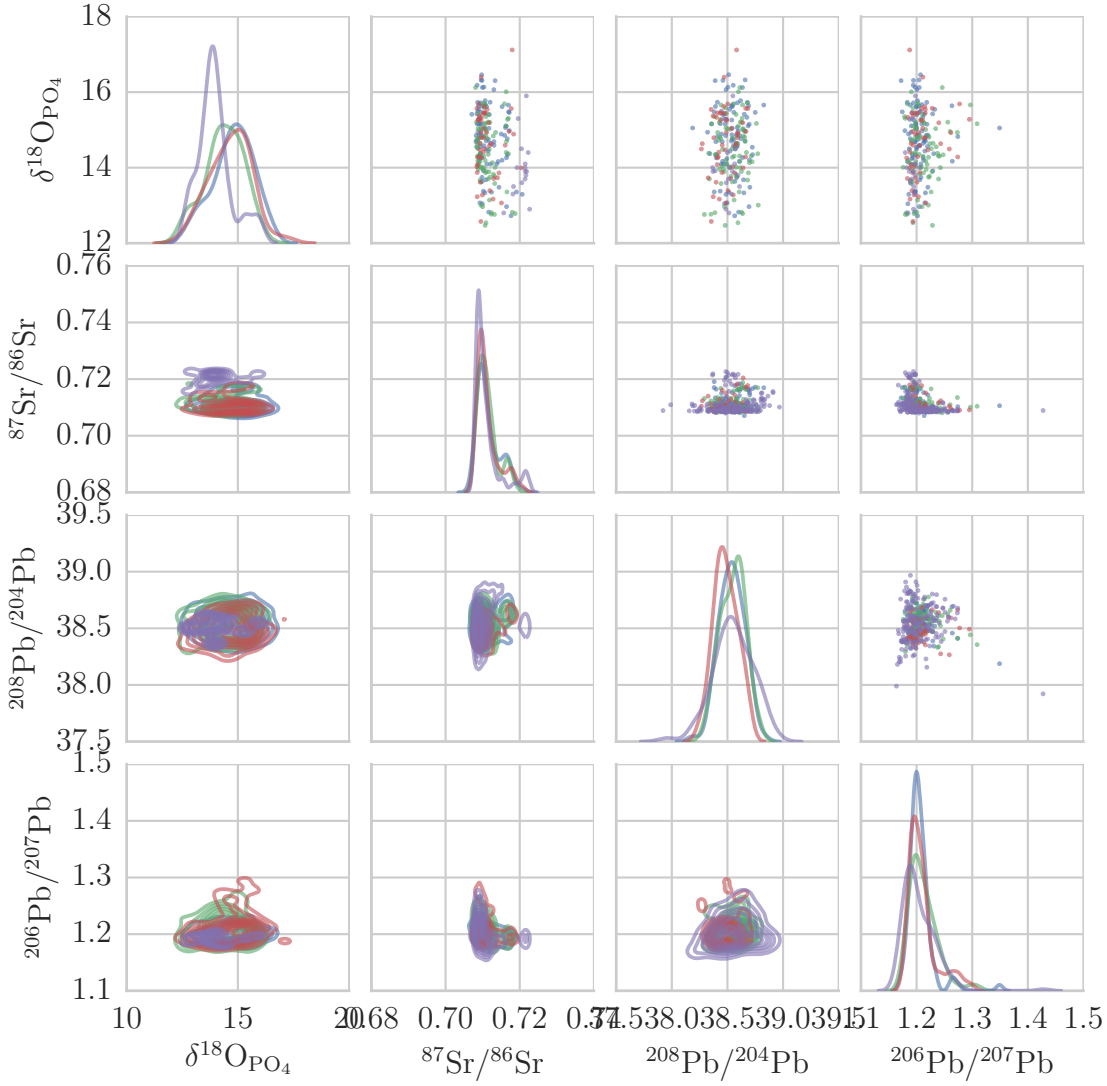


Figure 1.7: Correlations between isotope attributes.

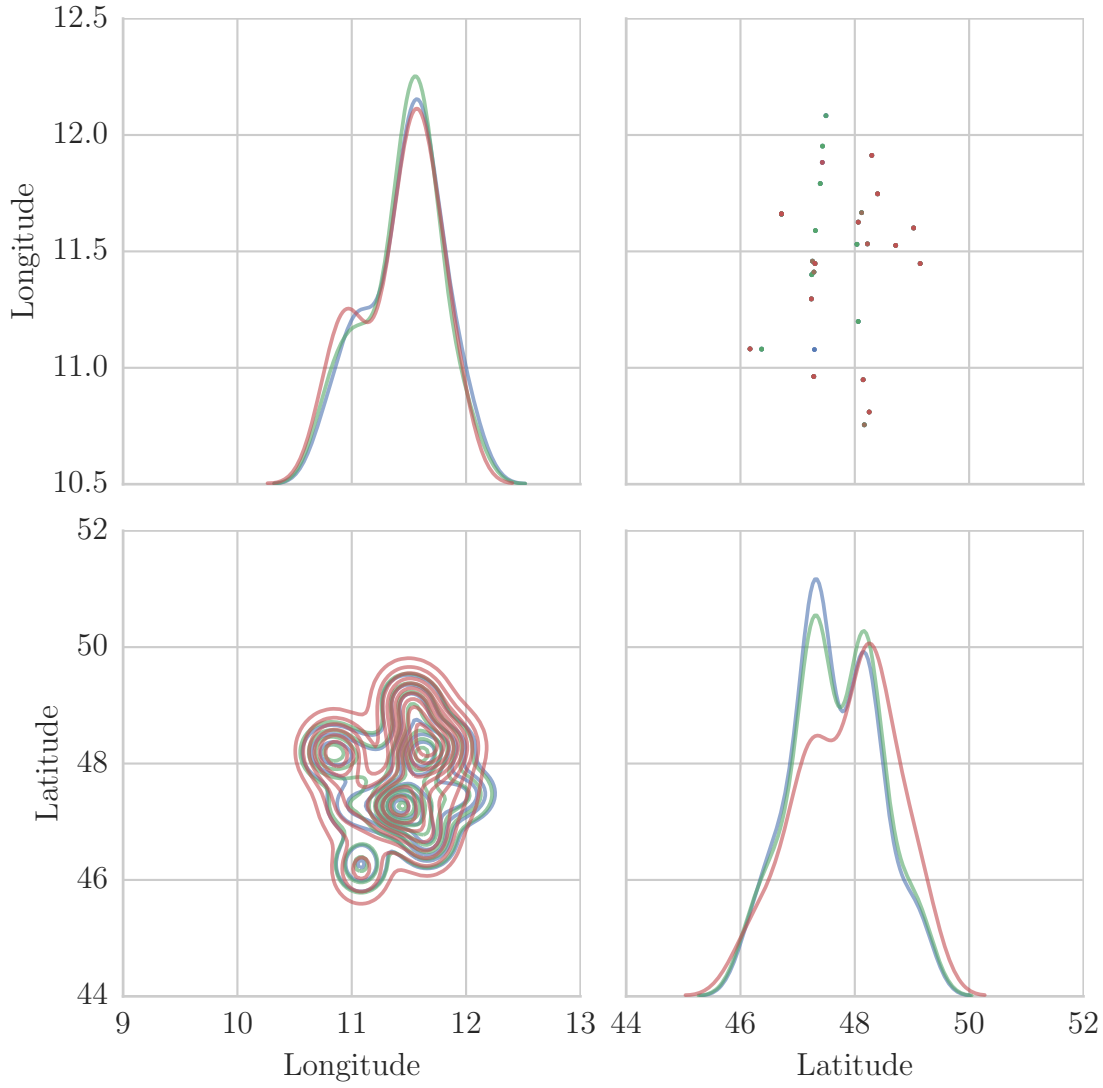


Figure 1.8: Correlations between spatial attributes.

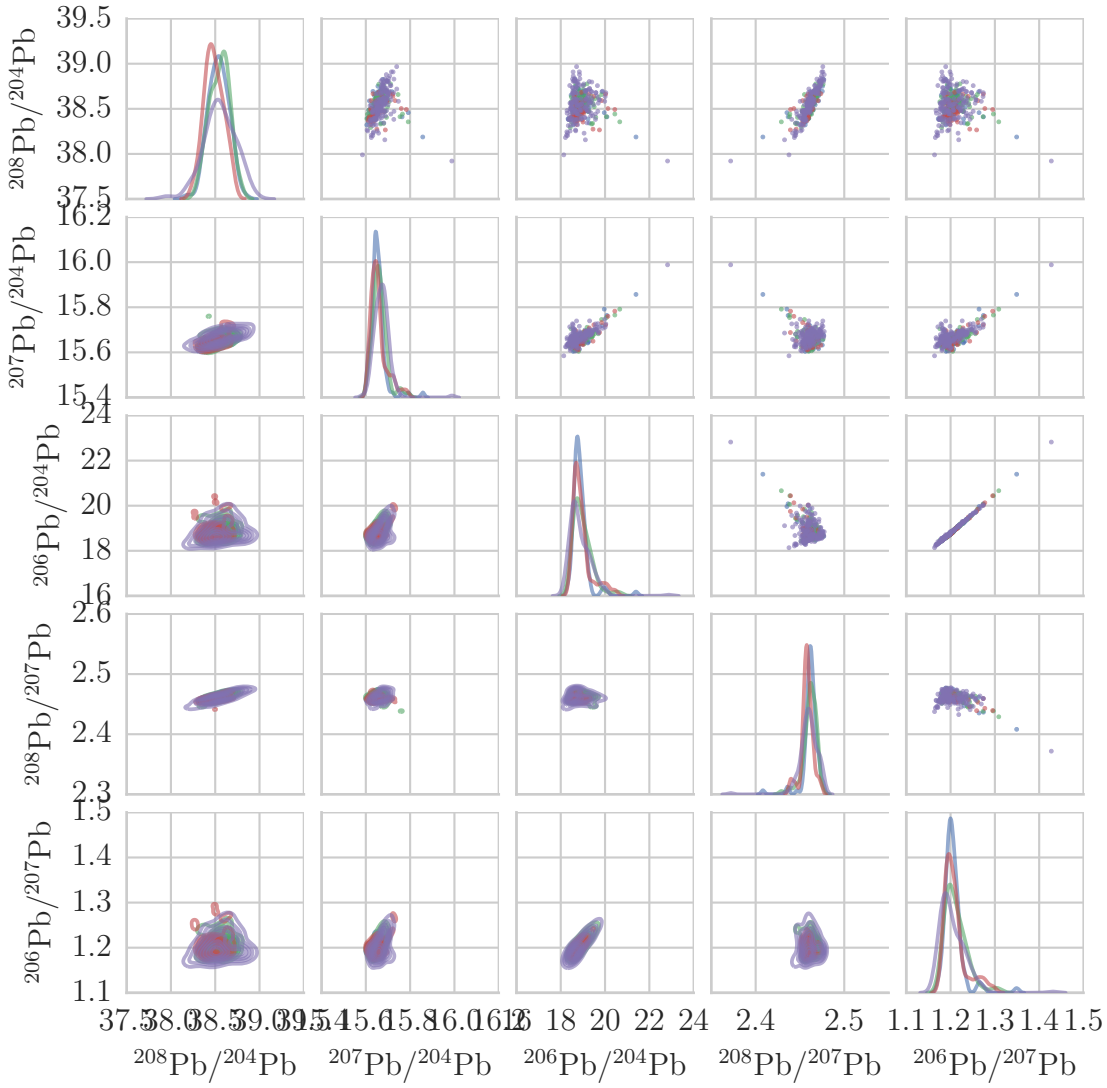


Figure 1.9: Correlations between lead isotope attributes.

test and consider as *extreme outliers* all those points that belong to the lower outer fence ($Q_1 - 3 \cdot IQ$) and the upper outer fence ($Q_3 + 3 \cdot IQ$). IQ is the interquartile range ($Q_3 - Q_1$), where Q_1 is the lower quartile (the 25th percentile) and Q_3 is the upper quartile (the 75th percentile). A visual explanation is given in Figure 1.10, where the extreme outliers area is pointed out in red.

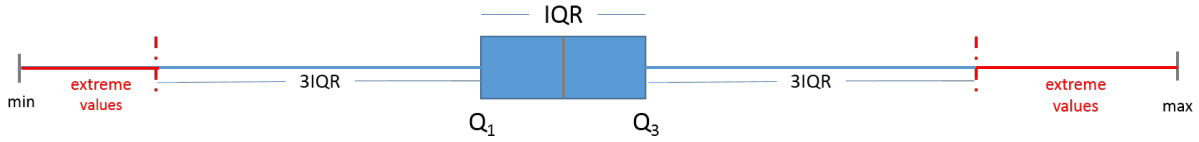


Figure 1.10: Extreme outliers test

Tables 1.6 and 1.7 show how much values differ from the mean of their site. Univariate outliers are identified by the distance to the corresponding region's mean (as a multiple of the region's standard deviation) regarding the investigated attribute. Traditionally, univariate outliers in isotope data were interpreted as non-local individuals. However, domain scientists soon realized that aside from mobility outliers can be caused by a number of factors. Possible causes include ranging or herding behavior and small-scale geological variability [78]. Even at the low outlier threshold of 2.0σ, no individual was flagged as an outlier by all attributes. This seems to indicate that a combined multivariate outlier score is required to get a good idea of each sample's outlierness. To calculate a multi-variate outlier score, each site's covariance is calculated and the distance score is calculated as multiples of it. The column *Mahalanobis* contains this multivariate outlier score. Figures 1.11 and 1.12 show the spatial locations of outliers. Some of the locations had to be joined for multivariate analysis to allow modeling of the covariance, which requires more tuples per locations than attributes to correlate.

The question of which points constitute a point's local region (distinct places of origin within which outlierness can be measured) is critical for the analysis. The modeling of individual sites (or spatially most close sites) are the most fine-grained regions. The model becomes more robust when locations are combined into regions of similar model.

Local outliers may fit with another location in the area covered by the data set (or outside it), maybe as a result of migration. This is one of the questions addressed in Chapter 5.

1.2.4.1 Example: individual outlier

As an example, consider *human 93*, an individual which may be foreign to the study region. For the exact numbers see Tables 1.3 and 1.4.

To allow local multivariate analysis, Hötting (5 samples) was joined with another site (211: Wilten, 2 samples). Based on the samples in these two sites, Human 93 has an outlier score of 2.27, which is quite high, but not extraordinary. On the other hand, its global outlier score (12.45) is very high indeed.

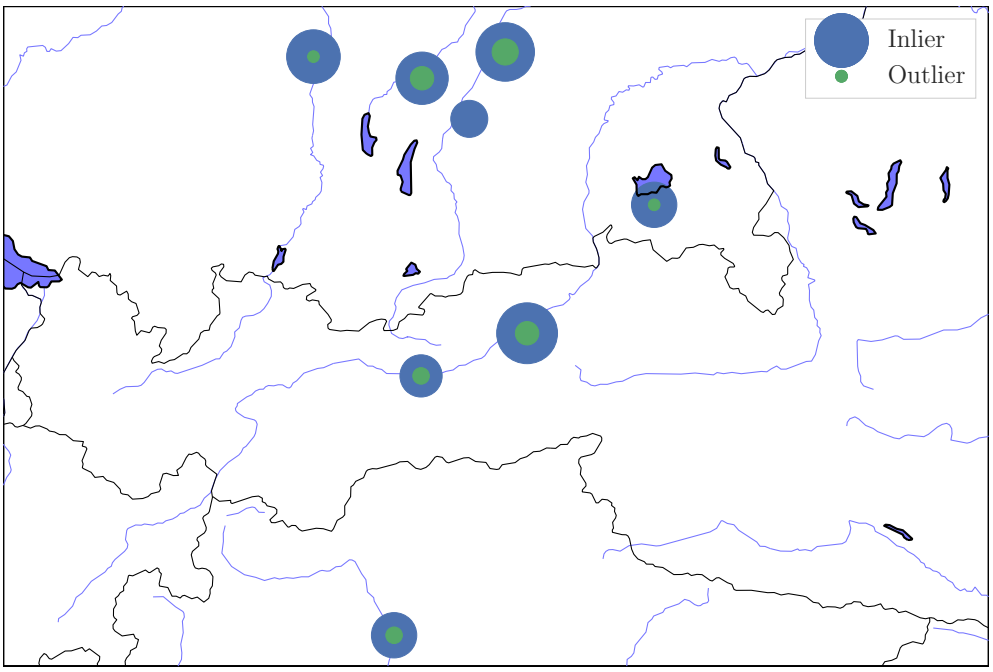


Figure 1.11: Multivariate outliers of data without $\delta^{18}O$ attribute at threshold 3.0. Some locations were joined to allow multivariate modeling. Some locations were merged to avoid overlap in presentation.

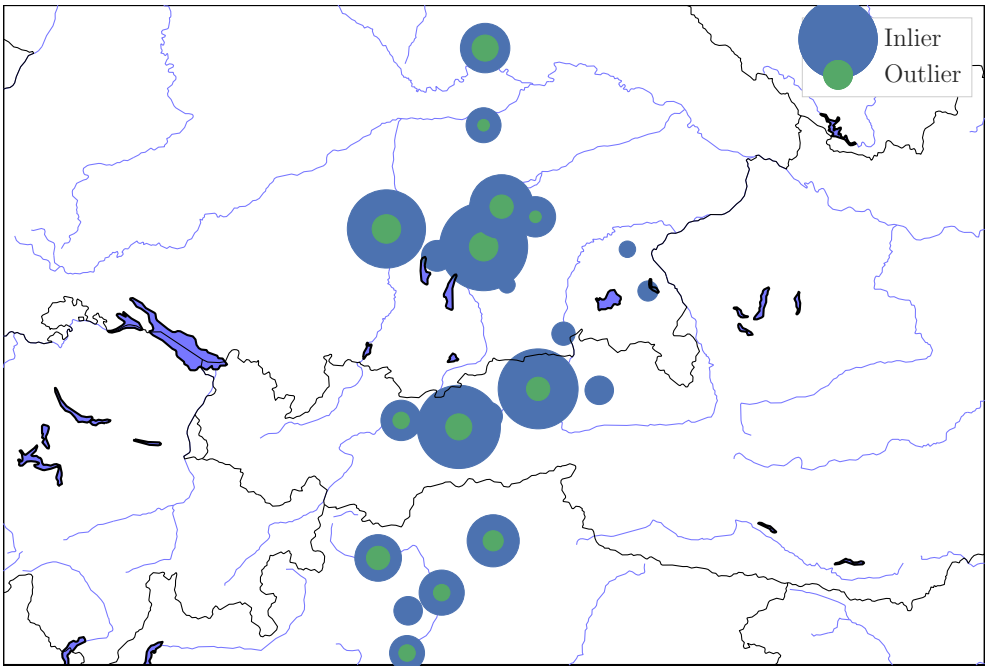


Figure 1.12: Outliers of all attributes of all data at threshold 2.0. Some locations were merged to avoid overlap in presentation.

	$^{87}Sr/^{86}Sr$	$^{208}Pb/^{204}Pb$	$^{207}Pb/^{204}Pb$	$^{206}Pb/^{204}Pb$	$^{208}Pb/^{207}Pb$	$^{206}Pb/^{207}Pb$
data	0.70889	37.9212	15.9882	22.8229	2.3718	1.4274
mean	0.711040	38.550294	15.671472	18.883906	2.459898	1.204906
std	0.003626	0.179491	0.038301	0.516933	0.011063	0.030544
outlier	0.592935	3.504885	8.269404	7.619929	7.963091	7.284347

Table 1.3: Global univariate outlier scores of Human 93.

	$^{87}Sr/^{86}Sr$	$^{208}Pb/^{204}Pb$	$^{207}Pb/^{204}Pb$	$^{206}Pb/^{204}Pb$	$^{208}Pb/^{207}Pb$	$^{206}Pb/^{207}Pb$
data	0.70889	37.9212	15.9882	22.8229	2.3718	1.4274
mean	0.710616	38.402929	15.706114	19.140071	2.445271	1.217943
std	0.003318	0.257892	0.125214	1.633861	0.033202	0.092993
outlier	0.520107	1.867944	2.252823	2.254065	2.212829	2.252385

Table 1.4: Local univariate outlier scores of Human 93 in Hötting region.

1.3 Overview and Attribution

This section gives an overview of the structure of this thesis as well as the previously published papers it includes.

This chapter (Chapter 1) introduced the idea of constraints and presented an overview over the used data and the research project, which generated it. It included some material from the publication *Influence of Oxygen Isotope Ratio on Classification* [50] by Markus Mauder, Eirini Ntoutsi, and Peer Kröger. This publication was a preliminary analysis of an early version of the FOR 1670 isotope data set generated in cooperation between the authors. For this thesis, most of its descriptive statistics were re-calculated on the final version of the data set and extended by further analyses by Markus Mauder. A second publication, from which some material was used, is *Applying Data Mining Methods for the Analysis of Stable Isotope Data in Bio-archaeology* [53] by Markus Mauder, Eirini Ntoutsi, Peer Kröger, Christoph Mayr, Gisela Grupe, Anita Toncala, and Stefan Hölzl. This publication introduces a new method for feature evaluation (see Chapter 3), which was not used in this chapter. Analyses and descriptions of the data by domain scientists Christoph Mayr, Gisela Grupe, Anita Toncala, and Stefan Hölzl were used to provide context of the data and analyses. These co-authors also provided the isotope data set used throughout this thesis. The publication *Data Mining for Isotopic Mapping of Bioarchaeological Finds in a Central European Alpine Passage* [51] by Markus Mauder, Eirini Ntoutsi, Peer Kröger, and Gisela Grupe sketched a first attempt to map the isotope distribution in the study area based on maximum likelihood assignment using the EM algorithm. The applied data model was conceived and generated by Markus Mauder, Eirini Ntoutsi, and Peer Kröger based on domain expertise by Gisela Grupe. The archaeological database *transmo* mentioned in see Section 1.2.1.2 was designed by Markus Mauder in cooperation with Alexander Thoelke and Andrej Wallwitz.

Chapter 2 introduces constraints in more detail including a taxonomy and possible approaches to specifying and incorporating constraints. It does not use any material from previous publications.

Chapter 3 addresses the problem of feature evaluation and how information about the structure of a good result can help identify interesting attributes. It is based on two of the same publications used in Chapter 1, but uses different parts of them. The majority of the used material was from the publication *Applying Data Mining Methods for the Analysis of Stable Isotope Data in Bioarchaeology* [53] by Markus Mauder, Eirini Ntoutsi, Peer Kröger, Christoph Mayr, Gisela Grupe, Anita Toncala, and Stefan Hölzl. The method proposed in this paper was developed by Markus Mauder, Eirini Ntoutsi and Peer Kröger, with input, analyses, and research help by the other authors, whose background is in the relevant domain sciences. The idea to compare clusterings based on different parameters and attributes was by Eirini Ntoutsi. An early version of the proposed method had previously been published as part of the other paper used in this chapter, *Influence of Oxygen Isotope Ratio on Classification* [50] by Markus Mauder, Eirini Ntoutsi, and Peer Kröger. The bulk of the execution of the two papers was shared between Eirini Ntoutsi and Markus Mauder.

Chapter 4 considers trajectory databases, specifically how impossible situations in a trajectory database can be specified, identified, and fixed. One application of this data is to reconstruct migration routes, the other considers the more complex case of interobject constraints and how to fix them generally. The framework and the second approach was previously described in *Minimal Spatio-Temporal Database Repairs* [21] by Tobias Emrich, Hans-Peter Kriegel, Markus Mauder, Matthias Renz, Goce Trajcevski, and Andreas Züfle[2] and a follow-up publication of the same title [54] by Markus Mauder, Markus Reisinger, Tobias Emrich, Andreas Züfle, Matthias Renz, Goce Trajcevski, and Roberto Tamassia. The research question and proposed methods in both papers were conceived by Markus Mauder, Tobias Emrich, and Andreas Züfle with valuable input from Matthias Renz and Goce Trajcevski. Experimental evaluation for the first paper was by Markus Mauder and for the second paper by Markus Mauder and Markus Reisinger.

Chapter 5 goes into the topic of spatial data modeling, particularly how Gaussian Mixture Models of the data can be generated while considering the data's spatial distribution. One possible approach that is being demonstrated is to cooperate with domain scientists interactively to generate a model that complies with their knowledge about the problem domain. The other approaches show different ways to incorporate constraints about the data distribution in the resulting model and discusses their particular strengths and weaknesses. Parts of the publications *Data Mining for Isotopic Mapping of Bioarchaeological Finds in a Central European Alpine Passage* [51] by Markus Mauder, Eirini Ntoutsi, Peer Kröger and Gisela Grupe. and *Applying Data Mining Methods for the Analysis of Stable Isotope Data in Bioarchaeology* [53] by Markus Mauder, Eirini Ntoutsi, Peer Kröger, Christoph Mayr, Gisela Grupe, Anita Toncala, and Stefan Hölzl were used to give the context for the proposed methods. Information about the study parameters in that publication were by

[2]Authors listed in alphabetical order.

the domain scientists Christoph Mayr, Gisela Grupe, Anita Toncala, and Stefan Hölzl. The interactive tool *GMMbuilder* was previously published in *GMMbuilder–User-Driven Discovery of Clustering Structure for Bioarchaeology* [49] by Markus Mauder, Yulia Bobkova, and Eirini Ntoutsi. The idea to use different models to find stable components is based on an approach first proposed by Eirini Ntoutsi [53]. The idea to combine these components into a full Gaussian Mixture Model was by Markus Mauder. The implementation of the tool was by Markus Mauder and Yulia Bobkova. The generalized constrained EM modeling method was designed by Markus Mauder with help from Peer Kröger, but not previously published. The idea, design, and implementation of this approach are by Markus Mauder. The approaches based on Monte Carlo and the constrained EM algorithm are previously unpublished methods by Markus Mauder.

Chapter 6 demonstrates ways to use the resulting models for research questions of domain scientists. One of the applications it considers is how to predict a possible better fitting origin of outlier data points, the other is showing the data distribution visually to make them accessible. *The Isotopic Fingerprint: New Methods of Data Mining and Similarity Search* [52] by Markus Mauder, Eirini Ntoutsi, Peer Kröger, and Hans-Peter Kriegel first introduced the map visualization presented in this chapter. *Data mining for isotopic mapping of bioarchaeological finds in a central European Alpine passage* by Markus Mauder, Eirini Ntoutsi, Peer Kröger and Gisela Grupe first described an early approach at origin prediction based on nearest-neighbor classification, but none of its material on the topic was used in this thesis. The presented visualization was conceived and implemented by Markus Mauder.

Chapter 7 presents some future research ideas and Chapter 8 concludes the text.

		$^{206}Pb/^{204}Pb$	$^{206}Pb/^{207}Pb$	$^{207}Pb/^{204}Pb$	$^{208}Pb/^{204}Pb$	$^{208}Pb/^{207}Pb$	$^{87}Sr/^{86}Sr$	^{18}O
Pig	$^{206}Pb/^{204}Pb$	1.00	1.00	0.66	0.00	0.34	0.10	0.03
	$^{206}Pb/^{207}Pb$	1.00	1.00	0.60	0.01	0.34	0.11	0.03
	$^{207}Pb/^{204}Pb$	0.66	0.60	1.00	0.04	0.23	0.01	0.01
	$^{208}Pb/^{204}Pb$	0.00	0.01	0.04	1.00	0.59	0.02	0.01
	$^{208}Pb/^{207}Pb$	0.34	0.34	0.23	0.59	1.00	0.04	0.00
	$^{87}Sr/^{86}Sr$	0.10	0.11	0.01	0.02	0.04	1.00	0.01
	^{18}O	0.03	0.03	0.01	0.01	0.00	0.01	1.00
Deer	$^{206}Pb/^{204}Pb$	1.00	1.00	0.69	0.01	0.37	0.07	0.05
	$^{206}Pb/^{207}Pb$	1.00	1.00	0.63	0.01	0.37	0.08	0.05
	$^{207}Pb/^{204}Pb$	0.69	0.63	1.00	0.17	0.21	0.03	0.06
	$^{208}Pb/^{204}Pb$	0.01	0.01	0.17	1.00	0.38	0.13	0.04
	$^{208}Pb/^{207}Pb$	0.37	0.37	0.21	0.38	1.00	0.26	0.00
	$^{87}Sr/^{86}Sr$	0.07	0.08	0.03	0.13	0.26	1.00	0.00
	^{18}O	0.05	0.05	0.06	0.04	0.00	0.00	1.00
Cow	$^{206}Pb/^{204}Pb$	1.00	1.00	0.68	0.02	0.44	0.09	0.00
	$^{206}Pb/^{207}Pb$	1.00	1.00	0.61	0.03	0.43	0.11	0.00
	$^{207}Pb/^{204}Pb$	0.68	0.61	1.00	0.01	0.33	0.01	0.00
	$^{208}Pb/^{204}Pb$	0.02	0.03	0.01	1.00	0.58	0.03	0.00
	$^{208}Pb/^{207}Pb$	0.44	0.43	0.33	0.58	1.00	0.04	0.00
	$^{87}Sr/^{86}Sr$	0.09	0.11	0.01	0.03	0.04	1.00	0.07
	^{18}O	0.00	0.00	0.00	0.00	0.00	0.07	1.00
Human	$^{206}Pb/^{204}Pb$	1.00	1.00	0.60	0.00	0.16	0.09	0.00
	$^{206}Pb/^{207}Pb$	1.00	1.00	0.54	0.00	0.16	0.09	0.00
	$^{207}Pb/^{204}Pb$	0.60	0.54	1.00	0.10	0.04	0.04	0.00
	$^{208}Pb/^{204}Pb$	0.00	0.00	0.10	1.00	0.75	0.01	0.00
	$^{208}Pb/^{207}Pb$	0.16	0.16	0.04	0.75	1.00	0.04	0.00
	$^{87}Sr/^{86}Sr$	0.09	0.09	0.04	0.01	0.04	1.00	0.11
	^{18}O	0.00	0.00	0.00	0.00	0.00	0.11	1.00

Table 1.5: r^2 values of features variables in all data.

	mahalanobis	^{18}O	$^{87}Sr/^{86}Sr$	$^{208}Pb/^{204}Pb$	$^{207}Pb/^{204}Pb$	$^{206}Pb/^{204}Pb$	$^{208}Pb/^{207}Pb$	$^{206}Pb/^{207}Pb$
b'1.115.14'	2.60	0.60	2.52	1.13	0.43	0.33	0.89	0.31
b'1.115.15'	2.65	2.52	0.11	1.04	0.21	0.29	1.50	0.31
b'1.106.2'	2.67	0.02	0.29	1.85	2.45	2.61	2.60	2.62
b'1.238.10'	2.97	1.40	2.77	1.11	0.81	1.54	0.85	1.49
b'1.237.6'	3.00	0.54	1.21	1.46	0.17	0.16	2.20	0.15
b'4.137.4'	3.01	0.22	0.35	1.99	1.93	1.70	1.52	1.64
b'4.137.11'	3.02	2.30	0.25	1.44	1.66	0.96	0.80	0.87
b'1.311.9'	3.03	1.62	0.79	0.21	0.83	0.77	0.29	0.66
b'1.237.11'	3.03	0.81	1.11	0.97	0.15	0.44	1.29	0.44
b'1.314.2'	3.04	0.41	0.59	0.61	1.11	1.01	0.17	1.14
156.0	3.05	0.03	0.49	2.13	2.04	1.62	1.71	1.46
b'1.148.4'	3.09	0.81	1.08	1.31	0.70	1.68	1.61	1.75
b'1.201.5'	3.10	0.81	0.67	0.66	0.95	0.33	0.18	0.27
b'1.148.5'	3.10	1.81	2.73	0.64	0.46	0.43	0.68	0.41
b'1.301.12'	3.11	0.97	1.73	1.13	1.74	0.92	0.55	1.15
b'1.229.12'	3.11	0.63	0.61	1.60	1.11	2.19	2.02	2.16
b'1.311.11'	3.11	0.03	0.43	0.07	1.66	0.75	0.98	1.09
b'1.310.7'	3.13	0.25	3.01	0.37	0.78	0.04	0.99	0.10
152.0	3.13	0.03	0.67	0.08	0.60	1.88	0.24	1.93
b'1.215.4'	3.16	1.02	1.56	2.01	1.21	1.77	1.31	1.81
b'1.237.1'	3.17	0.45	1.30	2.07	0.92	0.01	2.21	0.10
b'1.311.10'	3.20	1.84	0.23	1.05	0.85	1.39	1.88	1.30
b'1.311.12'	3.21	2.10	0.28	2.24	1.70	0.97	1.67	0.71
b'1.241.7'	3.23	0.76	1.14	0.51	0.72	1.75	0.15	1.73
159.0	3.27	1.48	2.59	0.03	1.03	1.13	0.71	1.07
b'4.137.6'	3.28	0.21	2.69	0.94	1.07	0.68	0.58	0.63
b'1.147.13'	3.29	0.48	0.85	0.36	0.19	0.64	0.44	0.73
b'1.203.3'	3.30	0.59	1.39	0.99	0.36	0.42	1.09	0.47
b'1.131.2'	3.33	0.77	0.13	0.82	1.01	0.98	0.07	0.95

Continued on next page

	mahalanobis	^{18}O	$^{87}Sr/^{86}Sr$	$^{208}Pb/^{204}Pb$	$^{207}Pb/^{204}Pb$	$^{206}Pb/^{204}Pb$	$^{208}Pb/^{207}Pb$	$^{206}Pb/^{207}Pb$
b'4.150.4'	3.33	0.74	2.30	1.56	0.39	0.37	1.01	0.46
b'1.301.2'	3.33	0.35	0.68	0.66	0.76	2.37	0.41	2.13
b'4.137.13'	3.37	1.35	0.24	0.60	0.46	0.63	1.77	0.73
b'4.137.12'	3.43	1.05	0.02	1.05	0.26	1.50	1.69	1.60
b'1.131.7'	3.51	1.07	0.52	0.50	3.15	2.31	2.22	2.12
b'4.116.1'	3.54	1.39	0.15	1.88	1.26	0.10	2.13	0.03
b'1.238.1'	3.60	0.19	0.62	1.12	2.49	1.05	0.71	1.39
b'1.237.7'	3.65	0.95	1.27	0.73	2.11	2.43	0.70	2.27
b'1.312.5'	3.69	1.00	2.36	1.43	1.09	0.57	1.22	0.40
b'1.236.6'	3.69	0.32	0.32	0.10	1.63	2.35	1.66	2.24
b'1.236.8'	3.81	0.18	0.44	0.99	1.85	0.40	0.16	0.63
b'1.312.6'	3.86	2.48	2.19	1.14	0.79	0.46	1.78	0.56
b'1.301.1'	3.88	2.27	0.80	0.26	0.03	0.38	0.33	0.37

Table 1.6: Outlier scores of data with $\delta^{18}O$ attribute per location with $\delta_{mahalanobis} > 3.0\sigma$ and $\delta_{univariate} > 2.5\sigma$. Locations with fewer samples than features (8) were joined to spatially close locations.

	mahalanobis	^{18}O	$^{87}Sr/^{86}Sr$	$^{208}Pb/^{204}Pb$	$^{207}Pb/^{204}Pb$	$^{206}Pb/^{204}Pb$	$^{208}Pb/^{207}Pb$	$^{206}Pb/^{207}Pb$
id								
90.0	2.89	nan	2.65	0.02	0.20	0.51	0.12	0.54
143.0	3.03	nan	0.49	1.12	0.91	0.58	0.81	0.78
141.0	3.03	nan	0.31	2.24	1.85	1.21	1.74	0.82
80.0	3.14	nan	0.60	2.31	0.55	1.31	2.65	1.27
57.0	3.15	nan	1.02	1.16	1.31	1.65	0.36	1.66
42.0	3.25	nan	1.25	2.13	2.40	1.27	1.59	1.10
126.0	3.27	nan	0.40	0.80	0.74	0.74	0.75	0.64
28.0	3.28	nan	2.52	0.43	1.01	0.50	0.08	0.43
102.0	3.32	nan	0.51	0.22	0.70	0.07	0.26	0.05
93.0	3.33	nan	0.82	2.45	3.25	3.29	3.18	3.28

Continued on next page

id	mahalanobis	^{18}O	$^{87}Sr/^{86}Sr$	$^{208}Pb/^{204}Pb$	$^{207}Pb/^{204}Pb$	$^{206}Pb/^{204}Pb$	$^{208}Pb/^{207}Pb$	$^{206}Pb/^{207}Pb$
21.0	3.38	nan	1.80	0.37	0.41	0.03	0.79	0.06
5.0	3.50	nan	0.88	0.05	0.07	0.68	0.04	0.73
115.0	3.51	nan	0.50	2.38	2.74	2.00	2.01	1.51
19.0	3.51	nan	0.70	1.00	1.39	0.78	2.32	0.70
77.0	3.62	nan	1.20	1.84	2.12	1.64	0.93	1.58
38.0	3.64	nan	1.96	0.51	0.22	0.52	0.66	0.55
9.0	3.66	nan	0.66	1.20	1.78	0.21	0.31	0.07
32.0	3.79	nan	0.69	0.39	0.31	0.48	0.28	0.57
130.0	3.82	nan	0.02	0.92	0.30	2.36	1.52	2.93
89.0	4.15	nan	3.95	1.07	0.35	0.79	1.37	0.82

Table 1.7: Outlier scores of data without $\delta^{18}O$ attribute per location with $\delta_{mahalanobis} > 3.0\sigma$ and $\delta_{univariate} > 2.5\sigma$. Locations with fewer samples than features (8) were joined to spatially close locations.

Chapter 2

Constraints

Attribution

This chapter does not use any material from previous publications.

Data modeling is among the most widely applied uses of data science and statistics techniques among researchers from a diverse set of scientific disciplines. Often those researchers limit themselves to a small set of analyses, which they are comfortable using. While the increasing adoption of these techniques by other fields is a welcome development, the necessity for data scientists to help drive the adoption of more appropriate models is not scalable. Often the information that researchers would like the results of computer-driven analyses to consider are fairly simple, yet out of reach of parameterizations of common algorithms.

Frequently information that is available about the data can easily be represented as properties that a good solution of the problem must satisfy. Measurements of data are e.g. commonly complemented by information about the measurements, such as geographical location and time of the measurement. This is particularly common during initial data collection, which is intended for building models. Traditional data mining algorithms discard this data, because it cannot be used as a model of the measurements in a straightforward manner. However, this data may hold information about the structure of the measurements that is not apparent in the measurements alone. Constrained algorithms try to find solutions for which constraints are satisfied.

In the following chapters, we will see various constrained algorithms. To express constraints the algorithms use properties associated with either a single or multiple points. They can be defined on the input data itself or on additional data associated with a point (or points). Combined with the current state of an algorithm (e.g. it current best guess at an appropriate output), these constraints are converted into costs, which are then consid-

ered to find a result that is more appropriate to the constraints.

2.1 Related Work

This section introduces other areas of computer science, which use the term constraints and compares their notions to the one applied in this thesis.

In semi-supervised learning [91], the term *constrained clustering* [86] refers to a class of semi-supervised clustering tasks. Constraints here refer to the fact that for some points in the training data, the solution is known. In practical terms of clustering, the label of some points is known and the majority of other points get assigned a label from the same labels (or a superset of the same labels) based on their respective similarity to the labeled points. Intuitively, this is one way to specify some domain knowledge about a solution and have the algorithm follow it. In terms of how constraints are interpreted in this thesis, labels can be considered additional information that is only available for some points. The constraint is evaluated by testing a solution of whether it violates the constraints $class(x^{(i)}) \neq class(x^{(j)}) \Leftrightarrow label(x^{(i)}) \neq label(x^{(j)})$ and $class(x^{(i)}) = class(x^{(j)}) \Leftrightarrow label(x^{(i)}) = label(x^{(j)})$. In this thesis we are not considering incomplete constraints, although this extension is a possibility (see Chapter 7). The problem presented in Chapter 3 is similar to constrained clustering, because it generates labels (albeit for every point), which it then considers to build a good solution.

Another use of the term *constraint* is in optimization theory [23, 63]. There – in the same sense as we use it here – constraints are attributes of a solution, which may not be violated, while a value is optimized. According to Jeavons et al. [37] an instance of a *constraint satisfaction problem* consists of:

- a finite set of variables, V ;
- a finite domain of values, D;
- a set of constraints $C_1; C_2; \ldots; C_q$.

And "[e]ach constraint C_i is a pair $(S_i; R_i)$, where S_i is a list of variables of length m_i, called the constraint scope, and R_i is an m_i-ary relation over D, called the constraint relation. (The tuples of R_i indicate the allowed combinations of simultaneous values for the variables in S_i.)"

The notion of constraints adopted in this thesis is similar, but specialized to a given task. This entails specifically that the type of constraint are inherent in the algorithm and only the data passed to it determines the actual constraints. Some of the presented problems can be expressed as optimization problems (and this was indeed one of the attempted solutions for the problem of Chapter 4), but this would arguably not solve the issue of making data analysis more accessible for domain scientists. The approaches presented in this thesis are specialized solutions to a narrower set of problems, which allows them being simpler to parameterize and yet produce acceptable solutions quickly.

2.2 Possible Constraints

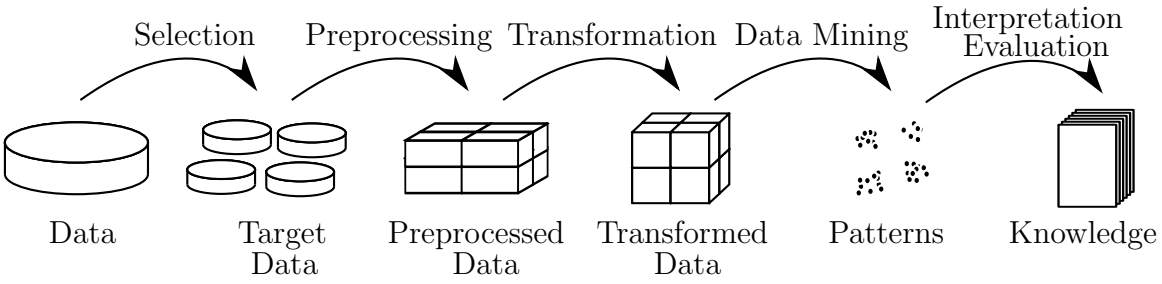


Figure 2.1: Process of Knowledge Discovery in Databases

Scientific questions are characterized by very complex interactions of many physical effects. For example, in the project described in Section 1.2.1 biologists have a deep understanding of metabolic peculiarities, feeding preferences, and habitat preferences, mineralogists know of decomposition artifacts in skeletal finds, and so on. Combining this diverse knowledge and compiling related questions into a set of data analysis tasks is no simple feat. The KDD process's many combinations of algorithms and parameters allows it to be adapted to the complexity of a wide range of scientific domains. This large number of combinations results in very complex analyses and requires a deep understanding of the effects each choice has on the solution. To translate these constraints into an analysis pipeline and suitable parameters for the employed algorithms commonly requires a data scientist [78]. The approach taken in this thesis is to build specialized data analysis approaches which allow domain scientists to specify their rich knowledge in terms of mathematical properties a solution must satisfy and have the automatic analysis take care of building an appropriate solution. Constraints can be used in all steps of the KDD process (see Figure 2.1). In this thesis we will look at a technique to be used in Selection (Chapter 3, Data Mining (Chapters 4 and 5), and Interpretation/Evaluation (Chapter 6).

2.2.1 Examples

Domain scientists may want diverse types of constraints satisfied. Examples of constraints in colloquial language are:

- A model should comply with predicate P.
- The spatial projection of a cluster's points are spatially close to each other.
- Two objects cannot be in the same place at the same time.
- The distance function between points should respect additional attributes a.
- Models should explain the distribution of attributes that are only available during training, but not incorporate them.

In the following we will look at some examples of constraints of increasing complexity.

2.2.1.1 Example: Constraints as parameters

Some solution properties are directly reflected by the parameters of an algorithm and thus easily enforced. A common example is the parameter (typically k) specifying the number of clusters in a clustering algorithm. A slightly more complex example is outlier detection where a commonly used approach is to determine the number of standard deviations that a point is from the model's mean. A value of 3σ is commonly used to identify outliers. This rule assumes that the data falls roughly in a normal distribution and that the 99.7% of points that are then expected to fall within a band of three standard deviations around the mean constitute close enough to all points to allow the reasoning that any points outside are indeed outliers. In a simple case like this it is trivial to modify an algorithm to comply with domain expertise. We expect domain experts to be able to make these simple adjustments without the assistance of a trained computer scientist, statistician, or data scientist. For a given problem an expert in the field may know from experience that the distribution the data is indeed likely to take is wider than a standard normal distribution and thus a value of e.g. 4σ might be more appropriate.

However, in realistic settings the relevant questions quickly increase in complexity until the influenced factors become more abstract and require deeper knowledge of the involved processes. Constraints allow domain experts to express knowledge about appropriate solutions in a way that is closer to their comfort zone.

2.2.1.2 Example: spatially coherent clustering

An example of an intuitive property defined over spatial information about measurement data is *spatial coherence*. (This constraint will be the discussed in detail in Chapter 5). Spatial coherence is the property that data which is spatially close has similar values. Spatial coherence is therefore a property that results of data analysis commonly should preserve. As an example consider the task of clustering data points, where points that get assigned to one cluster should also be spatially close. This agreement is plausible in the real-world where measurements usually progress smoothly with spatial distance. Very rarely are there edges where values suddenly change drastically or randomly. When divergence from this rule is detected this may indicate an interesting property of either a single data points or a sub-set of data points.

This uses additional attributes (the spatial information) to improve a solution. To predefine an optimal partitioning from the spatial information alone is not possible as it in turn does not necessarily reflect a good data model. However, specifying a possible measure of spatial coherence is comparatively easy, e.g.:

$$cost(C) := \sum_{c \in C} \frac{\sum_{x_i, x_j \in c} d_{\text{spatial}}(x_i, x_j)}{|c|^2},$$

where C is a clustering, x_i are points, and d_{spatial} is a spatial distance function. Minimizing *cost* (while generating a good solution to the problem) yields a spatially coherent clustering, i.e. one where the spatial distribution within the points belonging to a feature clusters is small.

A simple approach to generating a spatially coherent solution is to apply constraints to limit the possible solutions to spatially coherent clusterings. This type of constraint only validates results and is not constructive. The naive approach to make a spatially coherent clustering would require iterating over all potential clustering solutions to pick the spatially most coherent one. In Section 5.4.2 we will see an approach that uses a Monte Carlo simulation to make this approach computationally feasible. A preferred solution would be a heuristical approach using a cost function to zero in on a good solution directly. In Section 3.5.2.2 we will encounter spatial coherence as the goal of a feature selection step.

2.3 Types of Constraints

Constraints are manifested inside each algorithm from a set of constraint data, which is passed as an additional input, or based on the input data itself. The resulting constraints can be characterized by three attributes: They are either defined over a single data point (4.5, 5.4.3) or as a relation between data points (3.4, 4.7, 5.4.2, 5.4.4). They can be predicates (or Boolean constraints: this type of constraint is either satisfied or not satisfied, 3.4) or cost functions (one solution is better than another solution, 4.5, 4.7, 5.4.2, 5.4.3, 5.4.4). They can be local (point x violates constraint or $cost(x)$ is some value, 4.5, 4.7) or global (solution violates constraint or solution has some cost, 3.4, 5.4.2, 5.4.3, 5.4.4).

unary constraints The property is evaluated over each point individually.

n-ary constraints The property is evaluated over pairs (or larger sets) of points.

predicate constraints Assigns a Boolean compliant/non-compliant value.

cost constraints Assigns a continuous value as cost.

local constraints Assigns constraint satisfaction to individual elements from a data set.

global constraints Assigns constraint satisfaction to the global result.

Here are a few examples of combinations of these attributes:

Global predicate constraints Most simply (and least constructively) constraints can be applied at the end of an algorithm to verify that the result complies. This does not immediately over a solution for how to make it comply. Example: Section 5.4.2.

Global cost constraints This allows choosing a better solution. If other properties of the solution are known, the cost may be used to optimize the solution towards a better result. Example: Section 5.4.3 (single point), Section 5.4.4 (pairwise).

Local predicate pairwise constraints Earlier ways allow iterative refining of the solution to comply. Either predicate constraints, which give an indication of which parts of the model comply. Local predicates allow interpreting the number of violations as an indication of the global cost. Example: Section 4.7.

Local predicate single point Can be evaluated locally on each point. The number of local predicate violations can be interpreted as a global cost. Example: Section 3.4.

Local cost single point Can be evaluated locally on each point and may be optimized given an appropriate local cost function. Since only a single point is involved, a locally improved solution is likely globally better. Example: Section 4.5.

2.4 Satisfying Constraints

While constraints are generally not explicitly formalized, a solution that does not satisfy domain constraints will not be acceptable to a domain expert. To make a solution reflect a constraint, a data scientist can pass parameters to the algorithm or modify its input data. Properties of the result of the analysis can only be modified within the possibilities provided by exposed parameters. Algorithm parameters are by necessity close to the implementation of the algorithm and only indirectly reflect the change they effect. Thus to generate a constraint compliant solution requires an understanding of the way the algorithm works.

To overcome this limitation, existing algorithms can be modified to prefer solutions that comply with constraints, or new algorithms designed that support constraints natively. This work introduces some of these kinds of algorithms. The presented algorithms are all designed in a way that allows the user to pass additional data (*constraint data*) as input. The algorithm is then responsible to turning this data into constraints internally and consider these constraints during the analysis.

In the following chapter, a first example of this kind of algorithm will be introduced. The task we will focus on is feature evaluation based on information about desired structure in the training data.

Chapter 3

Application Specific Feature Evaluation

Attribution

This chapter uses material from the following publications:

- M. Mauder, E. Ntoutsi, P. Kröger, C. Mayr, G. Grupe, A. Toncala, and S. Hölzl. Applying data mining methods for the analysis of stable isotope data in bioarchaeology. In *2016 IEEE 12th International Conference on eScience*, 2016
- M. Mauder, E. Ntoutsi, and P. Kröger. Influence of oxygen isotope ratio on classification. Technical report, FOR1670: Transalpine mobility and cultural transfer, 2014

See Section 1.3 for a detailed overview of incorporated publications.

Feature evaluation is a task where data analysis and domain knowledge meet. In this chapter, we introduce a feature evaluation approach, which measures how relevant one data representation is relative to another one. In the context of feature evaluation, a representation can is a projection on a subset of the available attributes. Domain experts supply a representation of the data, which represents a desirable output (e.g., a full-dimensional representation, or one based on spatial data) and get an estimation of how similar to it another representation is. This allows domain experts to choose a data representation based on their domain knowledge.

The domain scientists' means to convey domain knowledge to the algorithm is by choosing a reference model to which the reference model should correspond as well as possible. The constraint data they can specify to this effect is a data set from which the reference

model is generated. The constraints that the algorithm generates from this are local predicate constraints indicating whether the investigated model's and the reference model's predictions agree for a given point. The proposed method outputs the constraint scores directly without using them to select a good result. Domain scientists are encouraged to try different approaches and choose a usable representation based on these scores. Automatically generating and choosing an investigated model is possible using this approach and would probably be considered a *feature selection* method.

The structure representation is based on a partitioning clustering, which can be based on a number of methods. To reflect the eventual data analysis task, the evaluation in this chapter is based Gaussian Mixture Models of the experimental data. To be usable as a partitioning clustering, their maximum likelihood cluster assignment is used to measure the relevance as well as the redundancy of each feature.

Domain experts can supply relevant subsets of features that reflect their knowledge about relationships inside the data. This knowledge is used to construct models that reflect it and use those models to evaluate how well different subsets of features reflect those same relationships. Or, conversely, how well other attributes can be substituted. This gives domain experts an indication of an attribute's relevance and redundancy.

In Section 3.5 we see an application of this technique to the bioarchaelogical data introduced in Section 1.2.2. The presented technique is applied to the task of provenance analysis, which uses well represented Gaussian Models as indicators for spatial origin of a sample. One aspect of the evaluation is if isotope data can be used to reflect the spatial distribution of the samples from which it was generated. The application of the presented data mining technique leads to new insights which were not found using standard bioarchaeological approaches.

3.1 Introduction

The task being investigated in this chapter is establishing the role of a feature in a data set given a GMM modeling. Assigning a single score to a feature (its "importance") is highly subjective, a problem that is not always acknowledged or addressed appropriately. In this chapter we consider the importance of a feature as a function of how well it is capable of expressing the structure of the data with respect to a set of constraints. These constraints can be specified as a set of features which are known to be relevant.

A common task in multivariate data analysis is assessing which features are relevant to the task at hand. The results of this kind of analysis can be used to limit the complexity of models and the number of measurements to record in the first place. In this chapter we introduce a feature evaluation technique that allows domain scientists to specify reference data, which has desirable properties, to understand a feature's role in the data distribution and to evaluate its importance for Gaussian Mixture Modeling.

Common analysis methods used by domain experts are limited to univariate and bivariate analyses only using visual inspection of histograms and scatter plots. However, some tasks require models over more than two features. For example, the task of provenance

analysis (predicting the origin of a sample) requires models over several isotopes to increase expressiveness and reliability. Choosing which features to generate (and which to possibly omit) is a non-trivial data analysis problem. This chapter describes a framework that was developed to solve this problem and support domain experts in making decisions about data generation.

This chapter is structured as follows: Section 3.2 describes a real research question from research project FOR 1670 for which this method was originally designed. Section 3.3 gives an overview of related literature. Then Section 3.4 describes our approach to constraint-driven feature evaluation In Section 3.5, we use isotope data to investigate which features should be measured in order to keep the costs for generating a reliable data source for an isotope map acceptable and address the question of oxygen relevance. Section 3.6 concludes the chapter with a summary of the presented approach.

3.2 Motivation: Relevance of Oxygen Isotopes for Spatial Distribution Modeling

Research group FOR 1670 aims at constructing a map of the study region to allow provenance analyses on new and suspect samples. One of the questions that need addressing before such a map can be built is which information must be collected to be able to build this map. Particularly – from a domain science point of view – not all isotopes are equally hard to work with. A particular problem for the study design is cremation as the principal (almost exclusive) burial custom [26]. Since light elements (including oxygen) are thermally unstable, oxygen data in cremated finds is unreliable. If it were established that oxygen is not necessary for a given task, the data sets could be combined.

Therefore, understanding the role of oxygen is important to communities which use isotopic fingerprinting as a tool. The effectiveness of oxygen and its contribution to models of isotope distribution is an ongoing discussion in the archaeological community. Also for further examples discussed in this thesis, the potential ability to omit oxygen is of practical relevance: About half of the collected samples are cremated human specimen. The process of cremation is characterized by high temperatures at which oxygen isotopes are not stable. This makes the oxygen isotope ratio unusable for cremated samples. In order to increase the sample size and thus the robustness of the resulting models, it would be convenient to discover that oxygen isotope ratios are redundant.

The following section presents a preliminary attempt to establish the importance of the oxygen isotope ratio to illustrate the idea underlying the approach presented in this chapter.

Is Oxygen a Good Indicator for Spatial Origin?

When trying to figure out the importance of a feature for a given task, it may help to plot the distribution of the feature relative to the target variable. Figure 3.1 shows the oxygen isotopes' distribution within the coarse latitudinal regions North, Center, and South as

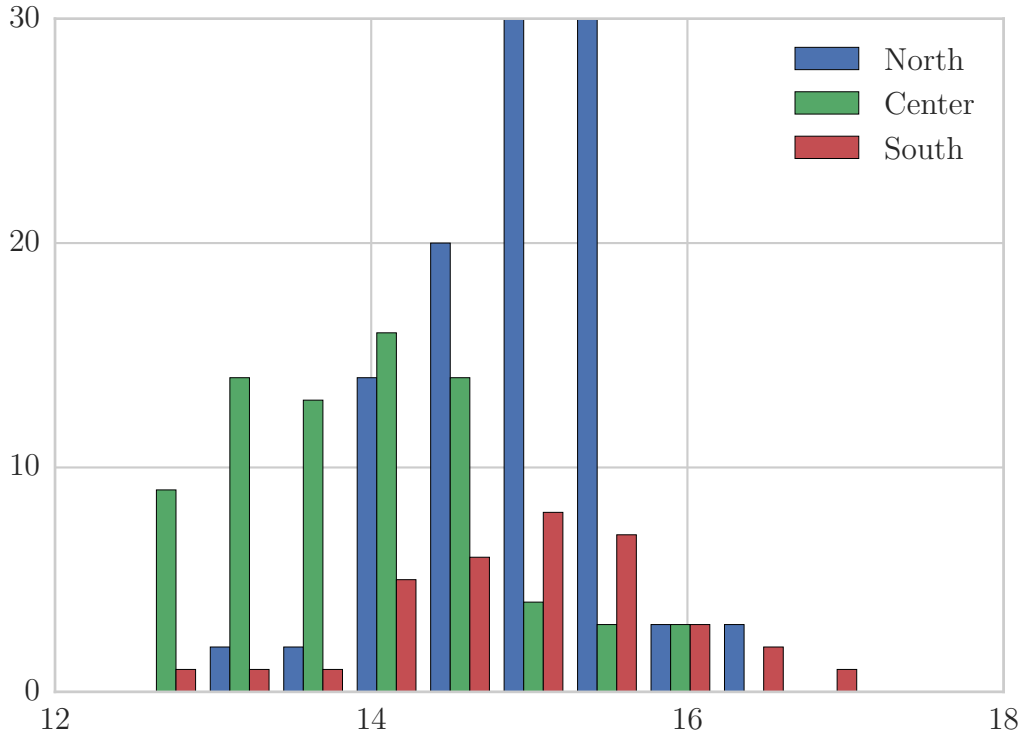


Figure 3.1: Distribution of oxygen isotope measurements by region. Although very large regions were picked, there is only a very weak correlation discernible.

shown in Figure 1.2. (A view of this data at a higher spatial resolution has already been shown in Figure 1.5.) The diagram shows clearly that all three regions' signatures overlap. North and Center have considerable overlap (about half of their respective populations) with one another, with North occupying generally larger values than Center. South's distribution is similar to that of North, but is considerably wider. Overall there seems to be no linear correlation between latitude (from which the label was derived) and $\delta^{18}O_{PO_4}$ratios that can be used to predict spatial origin. However, some value ranges do allow prediction of the displayed classes. For example, the value ranges below 14 are occupied mostly by data points from the Center group. Whether or not this information is useful in building in a model for provenance analysis is not immediately clear.

This question only becomes harder to answer as the granularity is reduced from only three groups to individual sites. To answer it in a quantifiable way is one of the objectives of this chapter.

Does Oxygen Contain Information Not Apparent from Other Isotope Ratios?

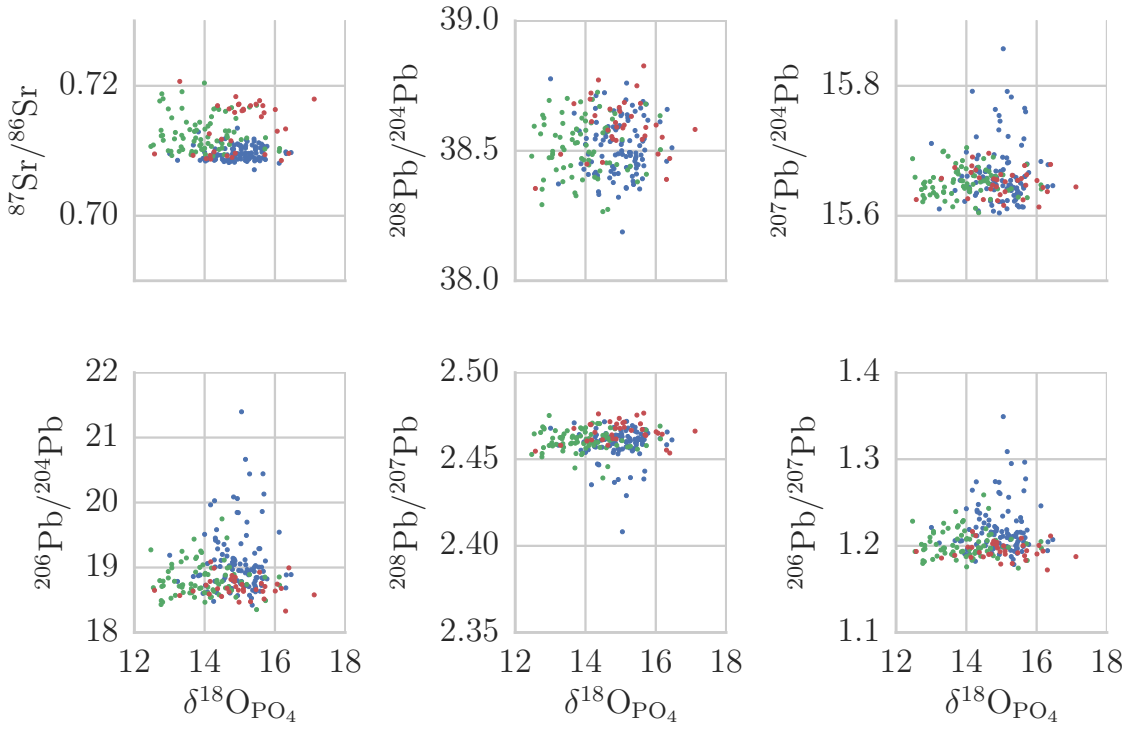


Figure 3.2: Correlation of oxygen with other isotopes. Colored by location: blue: north, green: center, red: south.

When deciding which features to use in an analysis, we may ask how indispensable that feature is. In a multivariate setting, two-dimensional projections are not always helpful to decide whether a feature (one constantly identical dimension of the plot) is correlated with a combination of other features. Figure 3.2 shows a plot of $\delta^{18}O_{PO_4}$ with all other available isotope ratios. No combination of $\delta^{18}O_{PO_4}$ with a single other isotope ratio shows an obvious correlation. However, this does not preclude the possibility that a combination of the features would show a correlation. And it does not give any indication whether a GMM of the data would find the same components with or without $\delta^{18}O_{PO_4}$. To decide whether $\delta^{18}O_{PO_4}$ has a unique contribution to the data is another question the technique presented in this chapter is trying to address.

3.3 Related Work

The task of assessing the importance of a feature for provenance analysis is reminiscent of feature selection and feature ranking. Feature selection generates a subset of the most suitable features for a given task, whereas feature ranking returns an ordering of features according to their importance for the task [30]. Most of the common approaches are supervised, meaning that they require class labels for accessing the quality of a feature or feature subspace [38]. Such information is not available for the discussed use case, therefore we have to rely on unsupervised feature selection approaches [15]. In particular, we follow a wrapper-based approach [40] where we use a learning algorithm (EM clustering in our case) for the evaluation of a feature or a subspace. FSSEM [18] is another well-known feature selection approach, which wraps the EM algorithm. Its extension FSSEM-k produces a feature ranking instead.

A big part of research on feature selection and feature ranking methods is focused on reducing the exponential search space of possible solutions. In our case, the feature space is low-dimensional but the domain scientists are interested especially in understanding i) the importance of each feature for the final model and ii) whether there are other features in the feature space that can replicate the "contribution" of that feature. The reason is that feature acquisition is an expensive process as domain experts have to follow lengthy and time consuming processes of cleaning the findings and measuring the isotope values. Moreover, in some cases it is not possible to measure all different isotopes for all available samples. This is the case for our project, where the oxygen isotope cannot be measured for cremated human findings. So it is extremely important for the domain experts to understand whether oxygen is a key feature for the analysis and also whether the remaining features can compensate for oxygen's contribution to the final model. Therefore, we follow a clustering-based feature evaluation approach, where we compare unsupervised learning results that convey aspects of the data structure (from a single feature point of view) with the data structure (as captured by the reference clustering).

3.4 Structure-based Feature Ranking

The question which features are useful for an analysis is generally termed "feature selection" or "feature evaluation". These tasks are subtly different in that feature selection aims at reducing the feature space, while feature evaluation suggests other ways of treating the results. The technique presented in this section falls in the feature evaluation category. Our motivating example is not to determine which features to eliminate (or conversely, which features to keep), but rather if the necessary removal of a feature poses a threat to the validity of the results. If it could be shown that basing the analysis on a complete feature set or a subset does not influence the output, then an analysis might be performed on the smaller set, making it more generally applicable and simpler. If, however, there is something to be gained from including more features, the question may now be whether the results are sufficiently similar to complete the analysis on different feature sets and use

them in further analyses as if they were equivalent.

An additional difference to most feature selection and feature evaluation methods is that we do not have a groundtruth against which the results of an analysis might be tested. Instead we have only the data set and a model which is to be applied. The presented method is (to the author's knowledge) unique in that it allows the user to specify a subset of features which have known properties (as reflected by the resulting Gaussian Mixture Model) and have another set of features ability to emulate these properties.

In this section, we present our approach to establish a score for measuring the influence of an attribute on a data set's structure in an unsupervised way.

The general challenge is to measure how much each feature is needed to separate the samples into these classes. However, since the ground truth is not known but its definition is rather part of the mining process, we need to employ unsupervised methods. This is in contrast to feature selection which typically relies on labeled data, for which a number of approaches exist (see Section 3.3).

In our approach, the quality of a feature, or feature subset, is assessed based on its contribution to one or more reference models. One application is to assess each attribute by measuring how stable (i.e., unchanged) the model is across feature space projections. We assume that a highly relevant projection will result in a model that resembles the reference model. So we can use a model as a reference against which to evaluate the other.

Our proposed feature evaluation framework consists of three steps:

1. data structure extraction (clustering)

2. data structure comparison (Adjusted Rand Index)

3. feature evaluation

Before explaining each of these steps, we introduce some notation: Let D be a data set in a feature space F. Let $F_0 \subset F$ be the feature set from which a model of the reference data structure is extracted by clustering; we refer to Θ^{F_0} as the *reference clustering* and to F_0 as the *reference feature space*. Let $F_v \subset F$ be a set of features to investigate w.r.t. their quality for the reference data structure, Θ^{F_0}. Note that F_v and F_0 are treated as being independent from each other even though they need not be disjoint.

3.4.1 Constrained Structure Extraction

The first step is creation of the labeling corresponding to the reference and investigated models. A cluster in the sense employed here is a set of samples, which when grouped minimize the samples' distance in isotope space, while maximizing the distance between clusters. As an "unsupervised" task, clustering will produce a model for the presented data regardless of the underlying real-world implications.

The presented method uses the EM algorithm to fit a *Gaussian Mixture Model* (GMM) to the input data. Many spatial data sets can be understood as having been generated by a mixture of a set of Gaussian distributions with a spatial center point characterized by

a vector of values, which dissipates and mixes with other values in a manner that follows a Gaussian distribution. A Gaussian Mixture Model (without a spatial component) can be approximated efficiently using the *Expectation Maximization* (EM) algorithm [17]. EM fits a number of multi-variate normal distributions over the given data set. Calculating the probability density of EM's Gaussian distributions, allows determining the probability of cluster membership for each data point given a model component. Choosing the model that fits a given point best, results in a *Maximum Likelihood* (ML) labeling. For points that were used to fit the Gaussian Mixture Model, this maximum likelihood labeling's probability is typically fairly high. The result of this maximum likelihood labeling is a set of partitions, $\Theta^F = \{\theta_1, \theta_2, \ldots, \theta_k\}$, where k is the number of model components in the underlying GMM.

3.4.2 Constrained Structure Comparison

Given two labelings, the position of a mismatch between the two is easily established (a local predicate constraint). The number of mismatches can be interpreted as a global constraint score. The labelings are based on two models, referred to as the *reference* and *investigated* model. The reference model typically serves as the target model. The resulting score is thus an indication of how similar the effects of both models are. Or, speaking from the point of view of the investigated model, how well the investigated model resembles the reference model.

Technically, to compare how well a labeling Θ^{F_v} (extracted from an investigated feature projection F_v) reflects the structure of a reference model Θ^{F_0}, we calculate the *Adjusted Rand Index* (ARI) [36] of the two clustering partitionings: ARI evaluates the agreement between two clusterings by counting pairs assigned to the same cluster under both clusterings and pairs assigned to different clusters versus the total number of pairs in the data set.

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left(\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right) / \binom{n}{2}}{\frac{1}{2}\left(\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right)} - \left(\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right) / \binom{n}{2}$$

ARI was proposed to reduce the influence of randomness on the traditional Rand Index (RI) [66] and has been proven to perform better when the number of clusters in the two clusterings is not the same [57, 84]. Like the rand index, ARI has a maximum value of 1 and takes the value 0 when the index equals its expected value. Negative values are possible, indicating an agreement that is less than one expected between two random clusterings. The ARI has a few desirable properties:

symmetric evaluates to the same score independent of which labeling is used as reference or investigated

positive for clusterings that are more ordered than random

invariant to label renaming the EM algorithm is non-deterministic. While the optimization of the model leads to a (possibly local) maximum of the likelihood function, the attained components can be permuted. This property makes this a non-issue.

If the model is not stable over two investigated feature subsets, the calculated difference may be high. This is a property of the investigated model, not the evaluation technique discussed here. We are not discussing how similar two data sets may be, but how similar their models are.

3.4.3 Evaluating Individual Features

Not all attributes are equally important for a given analysis task: A feature may be unnecessary to describe the result of a given analysis or the data reflected in the feature may be noise or encompassed by other attributes. We express the contribution of an attribute as two separate measurements: a measure of the influence of a given feature (its *structural relevance*) and one of the unique contribution (its *structural redundancy*) of the feature. By selecting a suitable comparison feature space we investigate the structural relevance as well as its structural redundancy.

Structural relevance how well a model built over a single attribute reflects the reference model.

Structural redundancy how well a model built over all other attributes in the investigated set of attributes reflects the reference model, i.e. how much is possible without the feature of interest.

To generate these scores, we extract a single feature $f \in F_v$. Let D_f be our original data set projected onto dimension f and let Θ^f be the model of $D_f : \Theta^f = \{\theta_1, \theta_2, \ldots, \theta_{k'}\}$, where k' is the number of components. We refer to Θ^f as the *univariate model*. Let $f_- = F_v \setminus f$ be the complementary feature space, that is, all dimensions in F_v except for the investigated feature f. Let D^{f_-} be the complementary data set, i.e., the data set projected onto the complementary feature space f_-. Applying EM on D^{f_-} generates a model $\Theta^{f_-} = \{\theta_1, \theta_2, \ldots, \theta_{k''}\}$ where k'' is the number of components. We refer to Θ^{f_-} as the *complementary model*.

To calculate the structural relevance of f, we compare the univariate model Θ^f derived from the specific feature f to the reference model Θ^{F_0}:

$$s_{relevance}(f, F_0) := ARI(\Theta^f, \Theta^{F_0})$$

To calculate the structural redundancy of f, we compare the complementary model Θ^{f_-} derived from the complementary feature space f_- to the reference model Θ^{F_0}:

$$s_{redundancy}(f, F_0) := ARI(\Theta^{f_-}, \Theta^{F_0})$$

The first comparison evaluates the structural relevance of f for Θ^{F_0}, whereas the second evaluates whether f's contribution can be reproduced by other features in the feature

space. In that sense, the first score derives the specific feature's structural relevance and the second score its structural redundancy due to the existence of other feature(s) in the feature space.

Due to the complimentary semantics underlying structural relevance and redundancy, they should not be combined into a single score. Instead, each feature f is characterized in terms of both structural relevance and structural redundancy. To help a domain expert glance the effect a feature may have on their analysis, we combine the two scores in one plot where the x-axis reflects the structural relevance score and the y-axis the structural redundancy. These plots will be explained in detail as part of the application described in the following section.

3.5 Application: oxygen's role in clustering

. In this section the introduced method is applied to assess the importance of the individual attributes in the FOR 1670 data set. First, we will focus on oxygen's role.

3.5.1 Manually comparing clusterings with and without oxygen

To illustrate how our method works, we first look at two GMMs and try to understand how they are related. The difference between the models is that one includes oxygen and the other does not. Then, we investigate how well the spatial distribution of the data is captured by those respective feature sets.

w/o oxygen	c_0^{-O}	c_1^{-O}	c_2^{-O}	c_3^{-O}	c_4^{-O}	c_5^{-O}	sum
w/ oxygen							
c_0	9	1	0	14	19	0	43
c_1	5	3	0	6	2	0	16
c_2	0	10	1	1	59	0	71
c_3	0	0	11	0	1	1	13
c_4	2	2	0	5	21	0	30
c_5	23	0	0	6	13	0	42
sum	39	16	12	32	115	1	215

Table 3.1: Confusion matrix of two clusterings on animals dataset using all attributes and all attributes without oxygen. The ARI of the depicted clusterings is 0.21.

Table 3.1 shows a confusion matrix of the maximum likelihood cluster labels derived from this task. The matrix's rows represent clusters in the oxygen-based clustering. The columns represent those from the attribute set without oxygen. We can observe that the cluster c_4^{-O} is much larger than any of the others (115 of 215). This is also apparent by inspecting which clusters in the attribute set without oxygen represent points that were originally part of a given cluster in the oxygen-based clustering: Cluster $l_{ox=0}$ of the oxygen-based clustering is split into c_0^{-O}, c_3^{-O}, and c_4^{-O}. c_1 into c_0^{-O}, c_1^{-O}, and c_3^{-O}. c_2 and

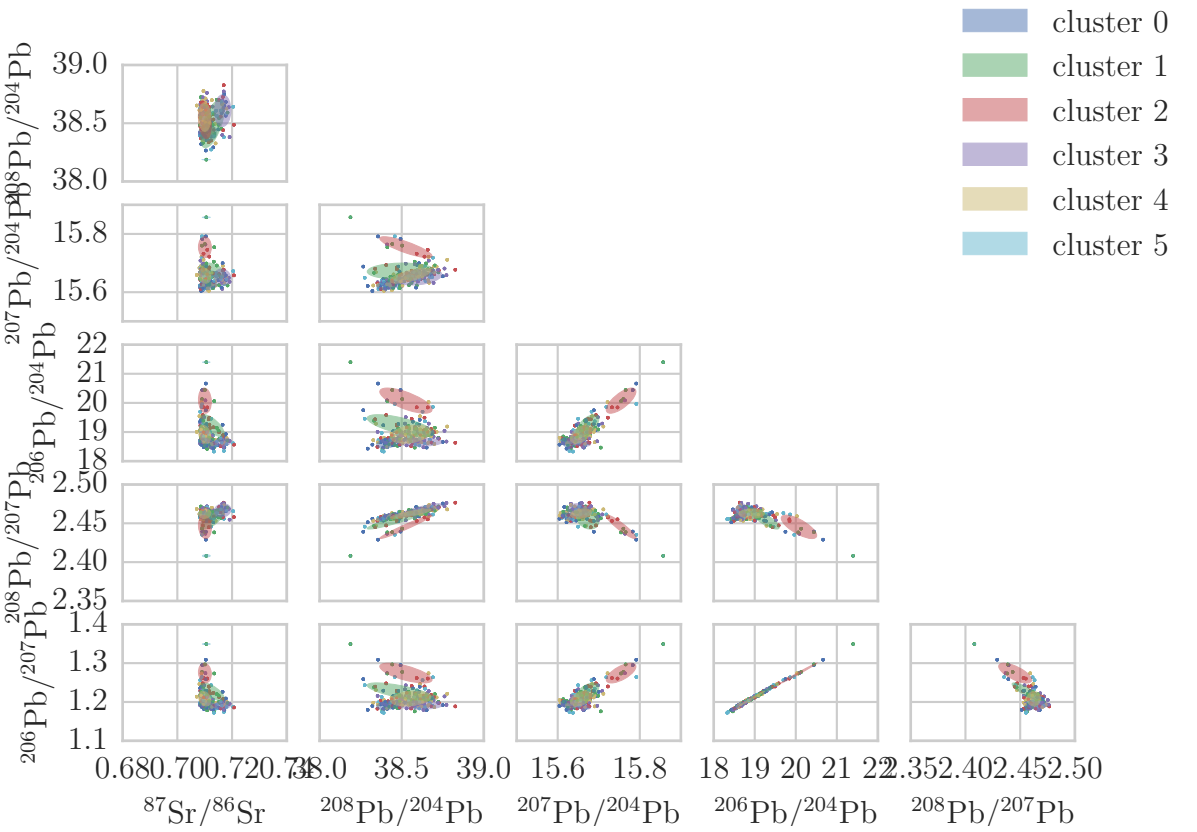


Figure 3.3: GMM distributions of feature attributes excluding oxygen in animals dataset.

c_4 became c_4^{-O}. c_3 became c_2^{-O}. c_5 became c_0^{-O} and c_4^{-O}. A possible explanation is that cluster c_4^{-O} is central in feature space and therefore not a good model. A look at the cluster models (a projection of which can be seen in Figure 3.3) supports that assessment. If we ignore Cluster 4 and look at Table 3.1 again, we can see that the remaining clusters (over the attribute set without oxygen) can still be recognized in the clustering including oxygen: c_0^{-O} becomes c_5, c_1^{-O} becomes c_2, c_2^{-O} becomes c_3, c_3^{-O} becomes c_0, and c_5^{-O}, which contains only one point, becomes c_3.

3.5.1.1 Comparing spatial distributions of models with and without oxygen

Looking at the spatial projection of the attribute set without oxygen (depicted in Figure 3.4), we can see that Cluster 4 (yellow) is indeed very prominent and spatially diverse. The cluster assignment probability for this cluster is nevertheless high (see Figure 3.5), indicating that it accounts for much of the overall data structure. However, the remaining clusters do show a clear spatial correlation. Cluster 3 (purple) extends from inside the Alps into the south, Cluster 1 (green) extends from the Alps into the north, and Cluster 2 (red) is located only in the North). Figure 3.6 shows the distribution of these other clusters more clearly by omitting the dominating cluster.

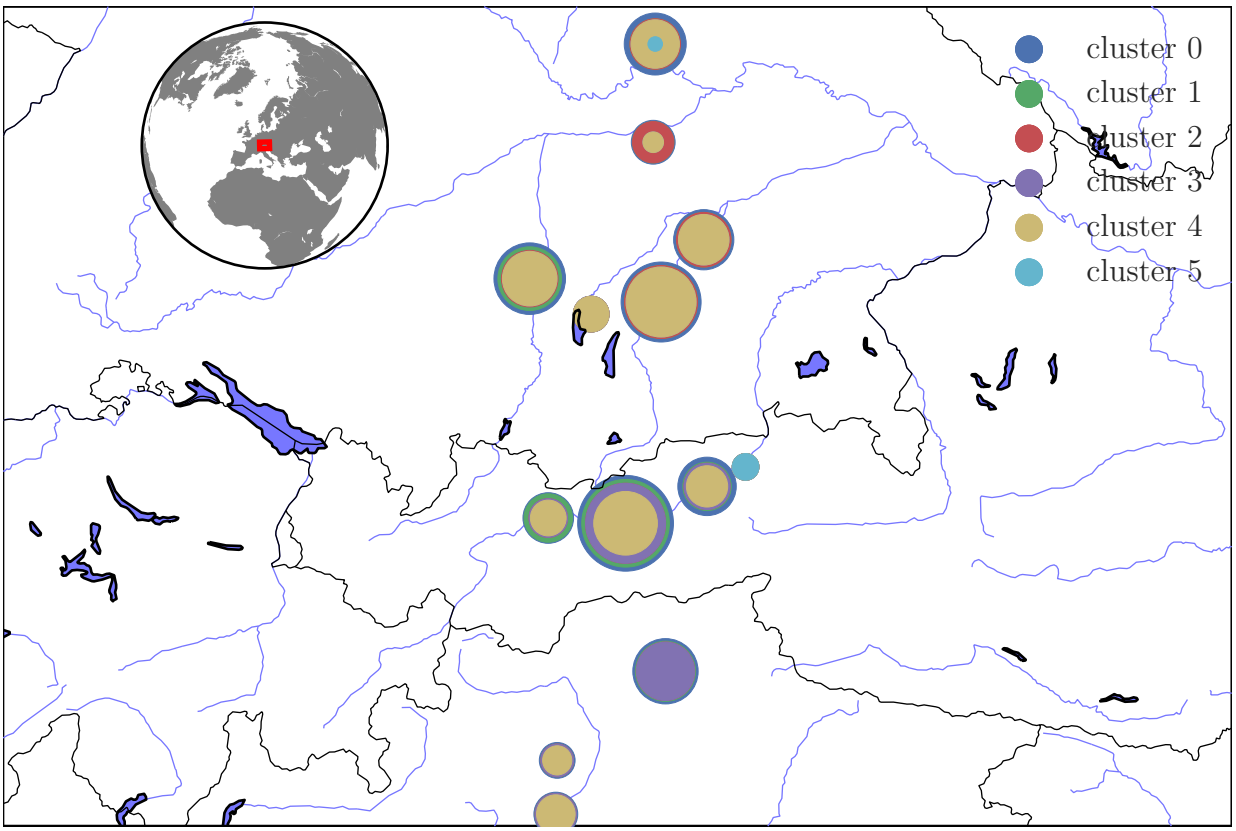


Figure 3.4: Spatial projection of maximum likelihood assignment to clustering of 6 components on animals dataset using all feature attributes except oxygen. Some locations were merged to avoid overlap in presentation.

The clustering of the data based on the features attribute subset, immediately shows a higher spatial correlation. For example, Cluster 2 (red) is located north of the Alps and Cluster 5 (cyan) mostly in the Inn valley (although some members of that cluster show up in many locations).

From a first visual inspection of the GMMs based on two feature subsets, our impression is that some labels remain unaffected (except for renaming), while others are not present in the other model. Particularly obvious is the emergence of a clear cluster in the center of the feature space, which absorbs many points.

3.5.2 Applying the Presented Method

The described technique explores the relevance and redundancy of individual attributes' contribution to a Gaussian Mixture Model in comparison to a reference clustering. Commonly evaluation of an analysis's result is based on comparing the result to a known reference result. Since there is no gold standard available, the presented technique uses domain knowledge to generate several *plausible* reference models that are estimations of

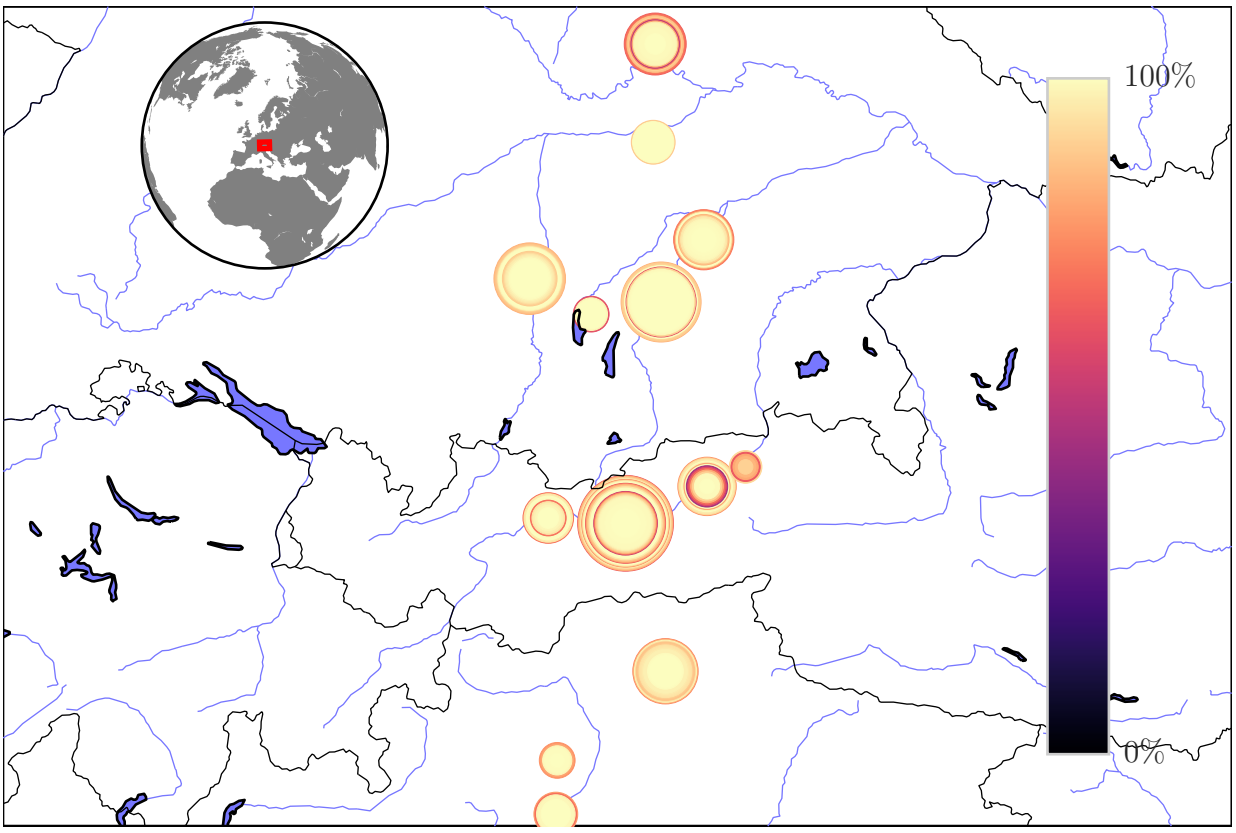


Figure 3.5: Spatial projection of maximum likelihood probabilities of clustering of 6 components on animals dataset using all feature attributes except oxygen. Some locations were merged to avoid overlap in presentation.

a ground truth. The choice of a reference model is informed by properties a final model should possess. Given a reference model, we explore the relevance and the redundancy of single features regarding a reference model and derive conclusions from these observations. by comparing a the result based on a projection on the feature under investigation (relevance) or all but the feature under investigation (redundancy).

We started this chapter with two questions:

1. Is oxygen a good indicator for spatial origin? (Section 3.2)
2. Does oxygen contain information not apparent from other isotope ratios? (Section 3.2)

The definitions of structural relevance and structural redundancy, which were introduced in Section 3.4.3, apply to these questions: The structural relevance of oxygen regarding a reference feature set encompassing spatial features gives an indication of oxygen's relationship to spatial origin. And the complimentary feature set's structural relevance regarding the full feature set (or the spatial subset) indicates whether oxygen is required or whether its information can be replicated by a set of other features.

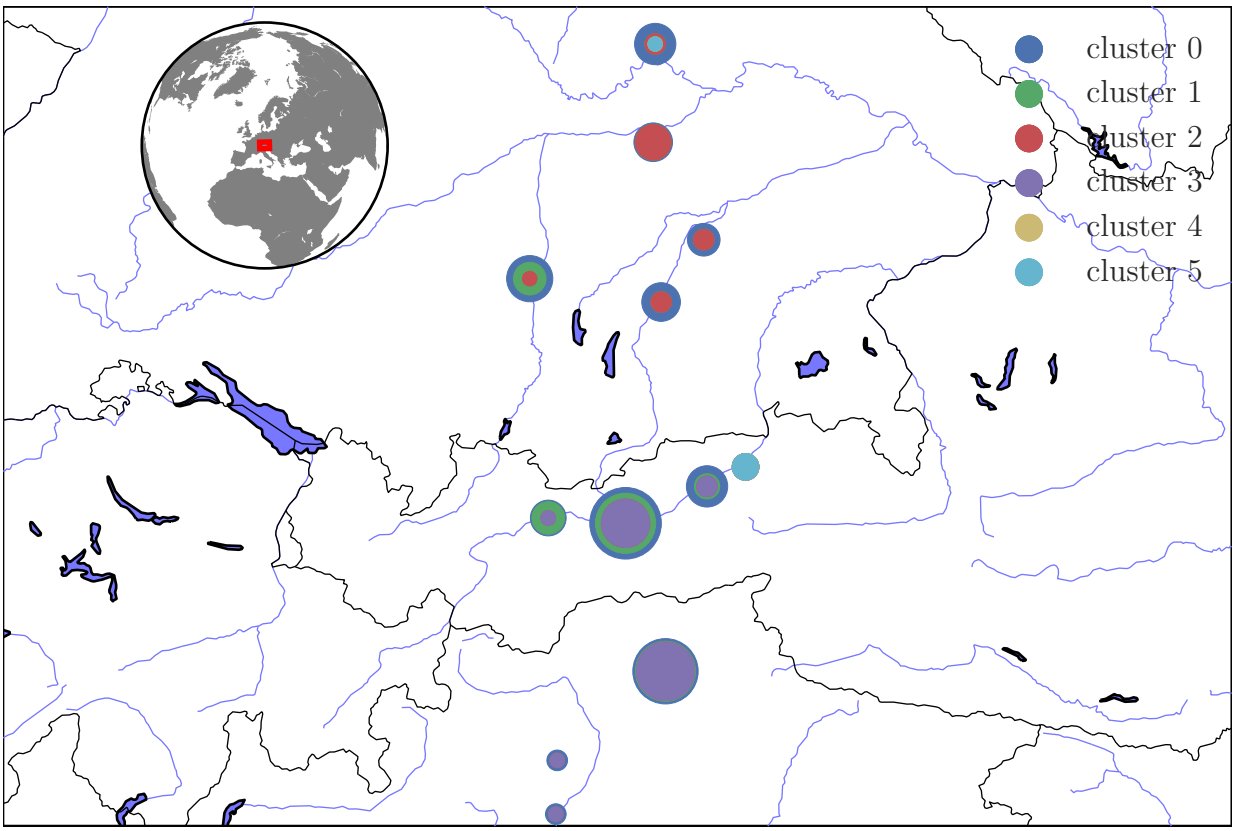


Figure 3.6: Spatial projection of maximum likelihood assignment to clustering of 6 components on animals dataset using all feature attributes except oxygen. Cluster 4 was omitted to show structure of remaining clusters more clearly. Some locations were merged to avoid overlap in presentation.

3.5.2.1 Reference Clusterings

The definition of the reference clustering is crucial for the presented feature evaluation technique. However, there are various choices for a possible reference clustering. We follow a mixture of a data driven approach enriched by domain expertise. Instead of using just one potential reference clustering, we investigated several possible definitions for the reference clustering based on the available features, in close collaboration with domain experts. The reference clusterings are generated using a data driven approach based on clustering, but rely on domain experts to specify the constraint data, i.e. decide which features to use for the reference clustering. Possible reference feature spaces range from containing all isotope and spatial features to containing only single domain features, i.e., isotopes, or spatial coordinates. n the following, the set I denotes all isotopic features, I^{-O} denotes all isotopic features except oxygen. In addition, we use S for all spatial attributes and S^{-lon} refers to the spatial attributes without longitude (see Table 3.2 for an overview). The investigated feature spaces are listed below. For a first set of experiments, the set of features under investigation is always the set of isotopic features, i.e. $F_v := I$.

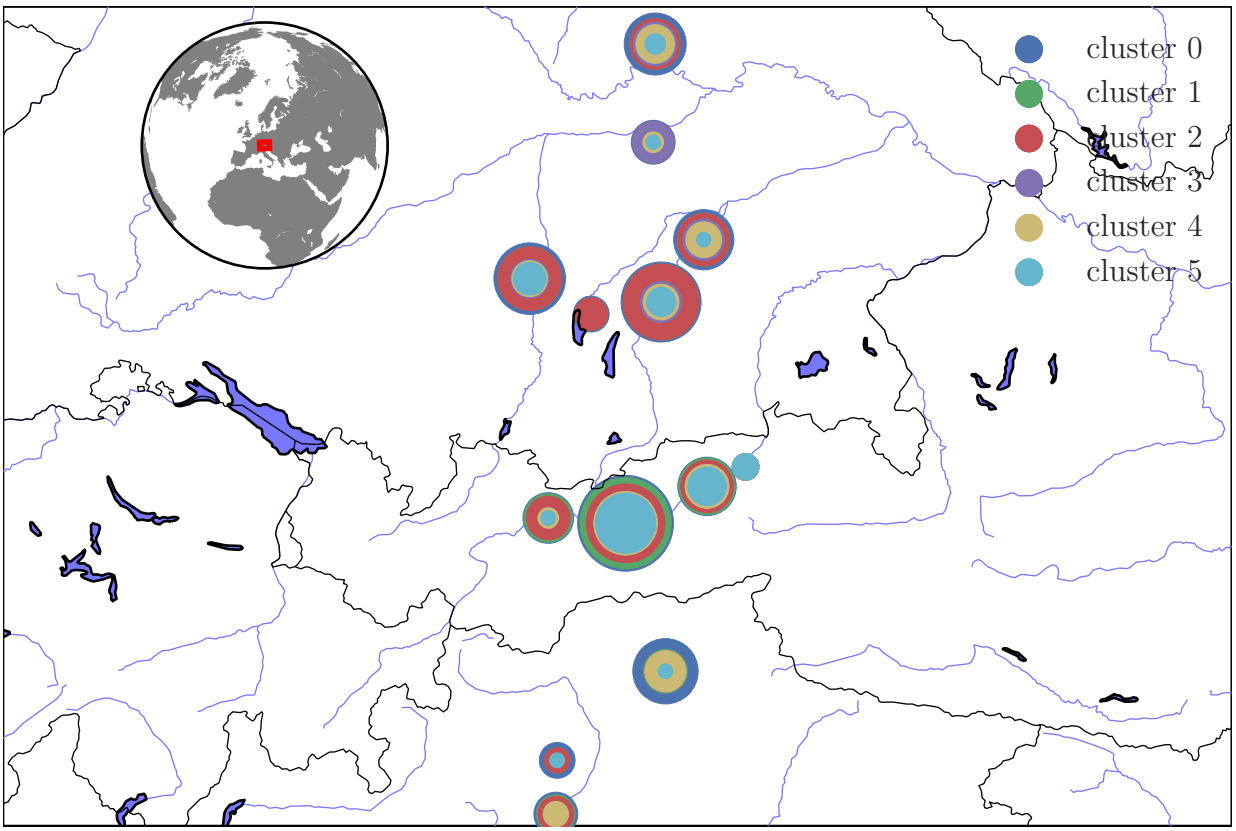


Figure 3.7: Spatial projection of maximum likelihood assignment to clustering of 6 components on animals dataset using all feature attributes. Some locations were merged to avoid overlap in presentation.

$F_0 = I \cup S$ **(Isotopes + Spatial)** The feature space consists of all available isotopic features and spatial features. This is the complete available information.

$F_0 = I \cup S^{-lon}$ **(Isotopes + (latitude, altitude))** From the spatial attributes only those that have been found to have an effect on the isotopes are retained, namely altitude and latitude. Since the spatial distribution of the data under inspection varies little in longitude, domain experts expect that longitude has only minor influence on the spatial compartments.

$F_0 = I$ **(Isotopes only)** The feature space consists only of the isotopic features without any spatial influence. This feature space is typically used for fingerprinting and predicting the spatial origin of unknown samples.

In a second analogously conducted series of experiments, oxygen was removed from the reference clusterings. This allows domain scientists to assume a different set of constraints to test the hypothesis how relevant oxygen is compared to other isotopes in this reference region and for this sample selection. The sample selection using a mix of three different

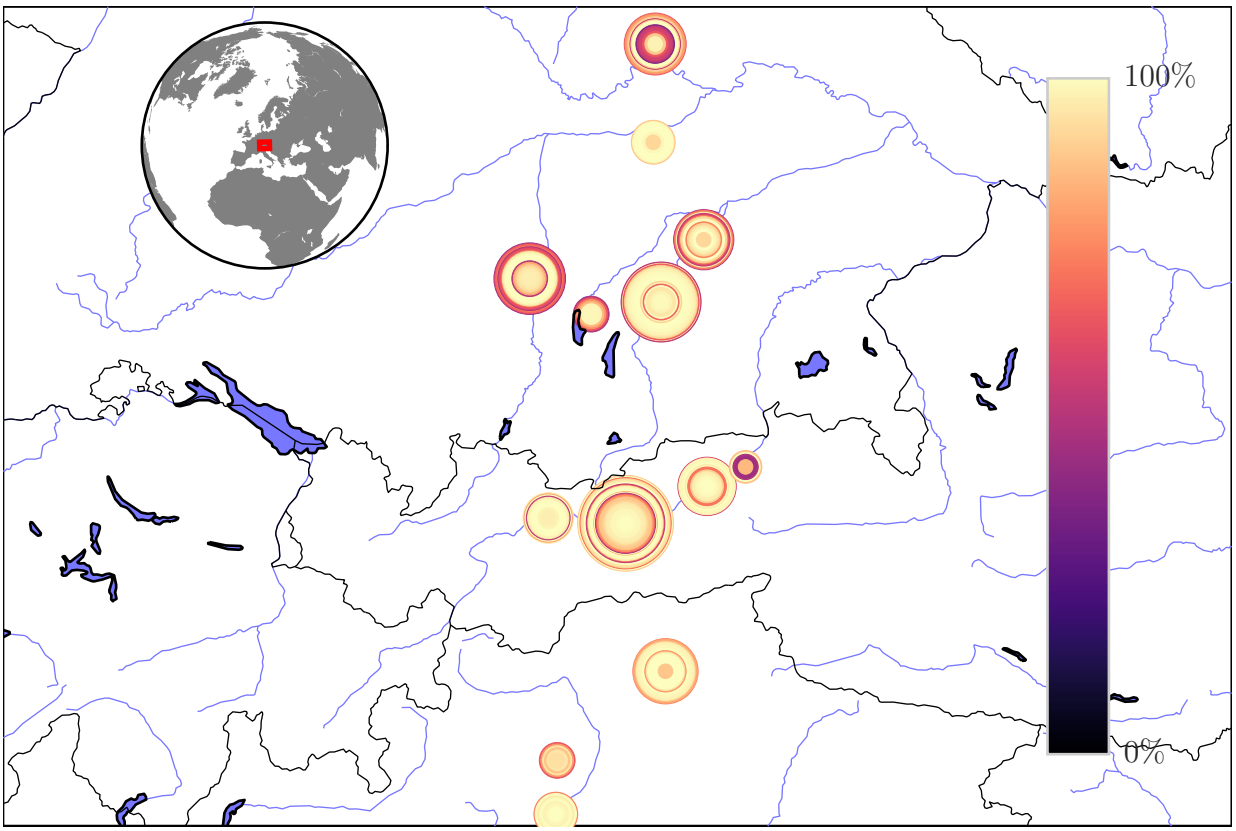


Figure 3.8: Spatial projection of maximum likelihood probabilities of clustering of 6 components on animals dataset using all feature attributes. Some locations were merged to avoid overlap in presentation.

species may have an impact on the model's $\delta^{18}O$-values according to domain experts. Analogously, the set of features under investigation is always the set of isotopic features without oxygen, i.e. $F_v = I^{-O}$. The resulting configurations are similar to the four alternatives listed above:

$F_0 = I^{-O}S$ **(Isotopes (except oxygen) + Spatial)** The feature space consists of all isotopes minus oxygen and all spatial features.

$F_0 = I^{-O}$ **(Isotopes only (except oxygen))** Only the isotope description, without the oxygen feature.

$F_0 = I^{-O} \cup S^{-lon}$ **(Isotopes (except oxygen) + (latitude + altitude))** Isotope description, without the oxygen feature and spatial coordinates except longitude.

A supplementary reference clustering is based only on spatial data:

$F_0 = S$ **(Spatial only)** The feature space consists only of spatial coordinates. Isotopic values do not play any role and findings from spatially close sites are considered

to be the same compartment. This ground truth scenario is complemented by a corresponding set of investigated features, i.e. $F_v = I$, and $F_v = I^{-O}$.

Reference clustering	Description (feature set)
I	all 7 isotopic features ($^{87}Sr/^{86}Sr$,$^{208}Pb/^{204}Pb$,$^{207}Pb/^{204}Pb$,$^{206}Pb/^{204}Pb$,$^{208}Pb/^{207}Pb$,$^{206}Pb/^{207}Pb$, $\delta^{18}O$)
I^{-O}	all isotopic features except oxygen ($^{87}Sr/^{86}Sr$,$^{208}Pb/^{204}Pb$,$^{207}Pb/^{204}Pb$,$^{206}Pb/^{204}Pb$,$^{208}Pb/^{207}Pb$,$^{206}Pb/^{207}Pb$)
S	all 3 spatial attributes ($altitude$,$latitude$,$longitude$)
S^{-lon}	all spatial features except longitude ($altitude$,$latitude$)

Table 3.2: Notations for the different subsets of features used to derive reference clusterings.

3.5.2.2 Experiments

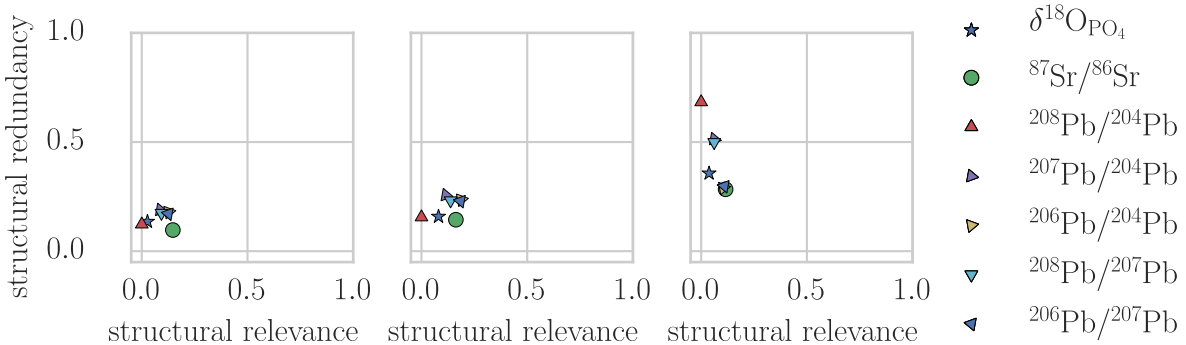
For each of the feature spaces described above, we apply EM to derive the reference clustering and we evaluate how each isotope attribute influences the corresponding reference clustering. The number of clusters was selected by cross-validation as implemented in the Weka data mining framework [31]. The investigated feature set F_v was the set of isotope ratios I or isotope ratios without $\delta^{18}O_{PO_4}$ I^{-O}. Which feature set was chosen depends on the chosen reference set. If F_0 contained I, $F_v = I$ was chosen. If F_0 contains only I^{-O}, $F_v = I^{-O}$ was chosen. A special case is $F_0 = S$, which does not contain any isotopes to compare with. In this case, both $F_v = I$ and $F_v = I^{-O}$ were used for completeness.

The results of reference clusterings containing isotopes including oxygen are presented in Figure 3.9, experiments with isotopes excluding oxygen are presented in Figure 3.10, and those with only spatial attributes are presented in Figure 3.11.

In the following we discuss the individual experiments' structural redundancy and structural relevance for all described reference feature sets and potential explanations for the observed values.

Each plot shows points corresponding to an isotope ratio. The structural relevance score depicts how well a model based on the corresponding isotope ratio alone represents the reference model. The structural redundancy score indicates how well the rest of the features can approximate the reference model.

Isotope Ratios: $F_v = I$



(a) $F_0 = I \cup S$, $F_v = I$ (b) $F_0 = I \cup S^{-lon}$, $F_v = I$ (c) $F_0 = I$, $F_v = I$

Figure 3.9: Structural relevance-vs-structural redundancy plots using reference clusterings with all isotope features.

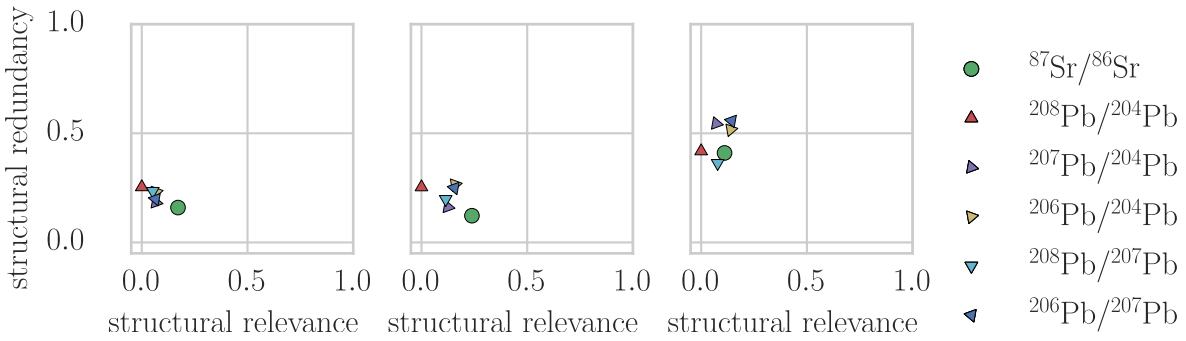
This set of experiments (see Figure 3.9) investigated the influence of the presence or absence of some spatial attributes relative to a reference attribute set of all isotope ratios. It is apparent that the redundancy scores are all low until all spatial information is removed from the clustering. This indicates that spatial information has a strong influence on the resulting model and that is not cleanly approximated by any of the investigated attribute subsets. Figure 3.9c is interesting in that the F_0 and F_v are identical. This is the base case that shows how well a single element performs in modeling based on I. Here we see that three lead ratios (mostly $^{208}Pb/^{204}Pb$) are least defining of the model. Oxygen is not redundant. $^{87}Sr/^{86}Sr$, $^{206}Pb/^{204}Pb$, and $^{206}Pb/^{207}Pb$ are least redundant in the model building.

Within what little difference between the investigated isotope ratios there is, $^{87}Sr/^{86}Sr$ is the most prominent ratio as it has the highest structural relevance score and the lowest structural redundancy score. Lead isotope ratios show a similar behavior, scoring average relevance and redundancy scores. An exception is $^{208}Pb/^{204}Pb$, which has a very low relevance. This result is mirrored in a separate experiment, in which building a univariate GMM over $^{208}Pb/^{204}Pb$ resulted in a model that consisted of a single Gaussian distribution. $\delta^{18}O_{PO_4}$ also has a very low relevance score.

The attributes that prove most relevant here are expected to be especially interesting for the generation of spatially coherent models.

Isotope Ratios Without Oxygen: $F_v = I^{-O}$

The following experiment also uses a feature reference attribute set, but it omits oxygen. Compared to $F_v = I$ the results are very similar. This indicates that oxygen (which has been removed for this experiment) has not had a huge influence on the model. A small,



(a) $F_0 = I^{-O} \cup S$, $F_v = I^{-O}$ (b) $F_0 = I^{-O} \cup S^{-lon}$, $F_v = I^{-O}$ (c) $F_0 = I^{-O}$, $F_v = I^{-O}$

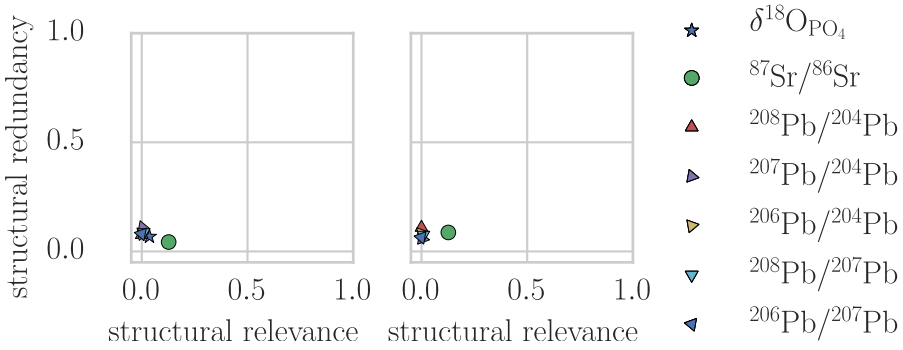
Figure 3.10: Structural relevance-vs-structural redundancy plots using reference clusterings with all isotopes except oxygen.

but noticeable change is that strontium is more relevant. Figure 3.10c again shows the individual ratios' performance versus the same reference attribute set. All isotope ratios' redundancy is higher than in the previous experiment, indicating that oxygen does influence the model somewhat when it is present. Variance between isotope ratios is small, but $^{87}Sr/^{86}Sr$ is again the most relevant attribute, whereas – if anything – the relevance of lead decreases. Removal of all spatial information (in addition to the removed $\delta^{18}O_{PO_4}$) increases the redundancy of all isotope ratios, but decreases their relevance. This is an interesting result in that the relevance of a single isotope ratio actually **decreases** when attributes are removed from the reference data set. This may be an indication that $^{87}Sr/^{86}Sr$ reflects some of the spatial structure of the data.

Spatial Attributes: $F_v = S$

The final experiment uses spatial attributes as the reference model's attribute set. This scenario tests how well the isotope's structure lines up with the spatial structure. We expect very little alignment as the spatial structure will be dominated by the density of sample sites, which the isotope values reflect indirectly at best. The lead isotopes have very low redundancy and relevance scores, indicating that they neither reflect the spatial structure, nor does their complimentary feature space do so. Strontium seems to represent the entire reflected structure, like we suspected in the discussion of the previous experiment. The results are compatible with previous experiments by confirming that the isotope's structure does not spontaneously reflect spatial structure. The very low redundancy of all isotope ratios is a result of the inability of any attribute set to reflect the spatial structure.

If oxygen is omitted from the investigated feature set, the situation changes only marginally. This indicates that oxygen had little influence on the structure of the isotope space, consistent with the analysis above.



(a) $F_0 = S, F_v = I$ (b) $F_0 = S, F_v = I^{-O}$

Figure 3.11: Structural relevance-vs-structural redundancy plot using reference clustering on spatial data. The investigated clusterings are based on all isotopes (a) and all isotopes except oxygen (b).

3.5.3 Discussion

In this section, we look at the presented experiments and try to extract some information about the role of the isotope ratios. Since domain experts recognize the need for multivariate analysis, we expected that no single feature can capture the entire structure of the data. This would result in low structural relevance values for all features. Indeed, this is what we see consistently in the analyses above.

Redundancy is expected to be be relatively high for lead isotope ratio, because there are five different lead isotopes which might "replicate" the effect of each other. Mathematically, they are fractions of one another, so no single lead isotope ratio should be required to inform the model. At first it seems that this hypothesis was wrong: lead isotope ratios' redundancy were not particularly high. However, this is easily explained by the low relevance of all feature sets. The complimentary feature space (containing many lead isotope ratios) was not able explain much of the structure of the data at all. Thus the redundancy scores cannot reach high values at all.

It is to be expected that the choice of reference clustering influences the ranking of different isotopes with respect to their structural relevance and structural redundancy. There are some properties which are apparent across different configurations. When spatial information is included in the reference clustering, non-spatial isotopes' scores are much less distinct than otherwise. This indicates that spatial features have a strong influence on the Gaussian Mixture Model. The low scores achieved by all isotopes against a reference clustering consisting only of spatial features illustrates that there is no trivial correspondence between the two domains, isotope and spatial. Domain knowledge suggests a connection, but it is not pronounced enough to be automatically reflected by the isotope feature set. Therefore the combination of both domains to extract a spatially coherent isotope map is also not trivial and will require more complex models.

With respect to *structural redundancy*, where there is a distinction to be made, $^{87}Sr/^{86}Sr$ generally has low redundancy, implying that the information in strontium is not replicated

by some other isotope or combination of isotopes in the data set. The lead isotopes display relatively (see above) high redundancy as expected since we have five different lead isotopes in our data set. All lead based isotope ratios behave fairly uniformly. This is expected, because the lead isotopes used in this study are measured relatively against the same two baseline isotopes ^{204}Pb and ^{207}Pb and (while they were measured separately) can mathematically be expressed as fractions of one another. Two lead isotope ratios that behave particularly similarly are $^{206}\text{Pb}/^{204}\text{Pb}$ and $^{206}\text{Pb}/^{207}\text{Pb}$. The other pair of lead isotopes that share a numerator do not show quite the same level of similarity.

Overall low relevance scores indicate that no isotope alone reflects the full structure of the data. This supports the emerging trend to use multivariate analyses in domain sciences.

Regarding the domain scientists' questions posed at the beginning of this chapter, we can observe a few things:

A multivariate isotopic fingerprint is needed instead of a univariate analysis relying on oxygen only. Our analysis showed that despite its popularity, oxygen does not provide exceptional structure to the data set (average structural relevance), nor is it unique in the role it plays (no exceptionally low structural redundancy values). Thus, at least in this reference region, provenance studies based solely on oxygen is bound to fail. On the other hand, the implication from our results is that the envisioned isotopic map can benefit strongly from a multi-isotopic fingerprint that includes strontium and lead isotopes as well.

Oxygen does not seem to be a particularly good indicator for spatial origin. Oxygen's relevance with $F_0 = S$ was very low. It does not seem to represent more information than the other isotope ratios.

It could not be shown that oxygen holds any information that is not also represented in other isotope ratios. Omission of oxygen in the isotopic fingerprint does not considerably decrease the quality of the fingerprinting. Oxygen did not show a particularly low redundancy. Its redundancy scores were always comparable with other isotopes, reaching values of up to 35%. This indicates that oxygen does not play an exceptional role in the data model and that other isotopes can provide much the same information as oxygen. Its low relevance score indicates that oxygen does not dominate the structure (i.e., other isotopes are needed). Since all isotope ratios showed very low scores, this result is not particularly strong and further research should be preformed.

The fact that oxygen seems not very relevant to provenance analysis in the reference region, opens up several opportunities. Although the inclusion of oxygen does not seem to diminish the clustering results, its omission also has little negative impact. This suggests that merging data sets may be worth the loss of $\delta^{18}\text{O}_{PO_4}$ information. So far, the isotopic map was designed to rely on animal bones only. Including human remains would be generally beneficial but available human samples are typically cremated, making oxygen values unavailable. The low relevance of oxygen opens up the possibility to explore this cremated material on a larger scale.

It should be pointed out that while the data mining methods presented here are generic in the sense that they can be applied to virtually any data from any reference region, the

concrete results of the case study (e.g. the relevancy of single features) do only hold for this particular reference region. However, due to the generality of the methods, it is easy to integrate more data in the future or even open the focus of this study to other parts of the Alps like Switzerland and France in the west or the other parts of Austria in the east.

3.6 Conclusion

This chapter presented a technique to judge the relevance of data relative to that of some reference data. Domain scientists can include their domain knowledge by defining which representation includes information that they would like a good representation to reproduce.

We applied this technique to the task of feature evaluation. In absence of other criteria, the inclusion of more features gives a more true representation of the data structure. By comparing subsets of the feature space with the complete feature space, we can gain an idea of how relevant or redundant a feature or set of features is. A particularly interesting variation of this theme is to compare the feature data with other information about the data, e.g. their origin, to see which attributes are usable in the modeling of spatial origin. The technique's purpose is to inform decisions about features, such as whether to record a variable in the first place, as well as guide further investigations into the role of a feature. After analysis, domain scientists are presented with two scores for each isotope: the structural relevance, which indicates to what degree the data's structure is represented in a given feature, and the structural redundancy, which indicates how much of the data structure is lost without the feature.

By splitting the result into two independent scores (structural relevance and structural redundancy) domain scientists can grasp two important orthogonal properties of the data that could otherwise not be discerned from univariate and bivariate visualizations. A variable that is structurally relevant, but redundant, may still be less important than one that is structurally less relevant, but cannot be replaced by a combination of different isotopes, or the other way around. In low-dimensional data sets individual variables are expected to be generally more relevant than in higher-dimensional ones. However, no single variable is indispensable if multi-variate analysis is employed. Indeed if the analysis could be based on only a single variable, multi-variate analysis would not be necessary for the application at hand.

The use of the ARI makes this model applicable to any data modeling approach that can be transformed into a partitioning clustering without much loss of information. This suggests the possibility of using a more spatially aware modeler to find feature models that reflect spatial distribution better. This would result in higher relevance scores for reference data sets that contain spatial information and may give better insights for domain scientists whose data does not accurately reflect the spatial structure.

To illustrate our technique in a practical context, we applied it to the FOR 1670 data set. In an application context these measurements inform further investigations of the

role of features in domain models. In the presented case study, domain scientists were presented with scatter plots of the structural relevance and structural redundancy scores of each isotope system in an archaeological data set. We described some insights which were gained from this analysis.

Feature evaluation is an early step in the KDD process. In particular, the goal of FOR 1670 is the creation of a model explaining the distribution of samples in the investigated region and its application to mobility and cultural transfer in the past. This analysis was important for the project since it supports (though it does not prove) the possibility to combine the project's two data sets. Some of the discussed results are also interesting to the domain scientists and have prompted further research.

A later chapter will describe techniques to generate a map of the spatial distribution of data. The presented analysis indicated that oxygen is not crucial, making it possible to use the human data set. To avoid issues caused by mixing species, we rely on the largest single group: humans.

Before we see this application, however, the following chapter will introduce constraints in the analysis of trajectory data.

Chapter 4

Improving Route Data

Attribution

This chapter uses material from the following publications:

- T. Emrich, H.-P. Kriegel, M. Mauder, M. Renz, G. Trajcevski, and A. Züfle. Minimal spatio-temporal database repairs. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 492–495. ACM, 2013
- M. Mauder, M. Reisinger, T. Emrich, A. Züfle, M. Renz, G. Trajcevski, and R. Tamassia. Minimal spatio-temporal database repairs. In *Advances in Spatial and Temporal Databases*, pages 255–273. Springer International Publishing, 2015

See Section 1.3 for a detailed overview of incorporated publications.

4.1 Introduction

This chapter addresses constraints in the context of spatial object databases. Initially constraints are introduced on routes, which are sequences of spatial locations. Each location in a route can be evaluated using a local unary predicate constraints. The aggregation of constraint violations indicates whether the database is constraint compliant.

This general framework is extended in two ways. First the constraint at each location in a route is extended to be continuously valued, indicating how much this particular location agrees with domain knowledge. Second inter-object constraints are introduced, binary predicate constraints indicating whether two objects together violate a constraint. Since the problem of finding an optimal database repair on inter-object constraints is

NP-hard, we propose a number of heuristics to repair a spatio-temporal database, which are organized into three general categories of solution, including time-distortion, space-distortion, and hybrid approaches.

For both extension, repair rules are defined, which can be applied to reduce the number (or cost) of constraint violations. Keeping the changes to the data set minimal is desirable in order to keep the data as close to original state as possible and thus increase the likelihood of producing a state that is actually closer to the true model than the input. The focus is not on removing the inherent uncertainty of the input data, which can be one of the reasons for inconsistencies. Rather, we aim at reducing the number of known problems to arrive at a solution that is more likely according to domain knowledge (because it does comply with known constraints). As a simple example, the interpolation of GPS signals may lead to the consequence of having a trajectory of a given car going through a lake. Fixing this problem by having the trajectory going around the lake may still yield the wrong trajectory, as the true trajectory may look different. This approach cannot alleviate the root-causes for errors in spatial databases, but it can yield interpretations that are more likely than naive acceptance of the data at face value. Clearly, a method generating constraint conformity should aim at minimizing the distortion between the original database and the repaired database. To achieve constraint compliance with a minimal set of changes, a way to describe constraints on routes is defined, a method to measure the difference of a modified data set to the original is defined, and means to minimize the distance while reducing the number (or degree) of constraint violations is defined.

The rest of this chapter is organized as follows. Section 4.2 introduces the task of reconstructing plausible migrations routes, which motivates the presented approach. Section 4.3 presents a review of related work. Section 4.4 introduces definitions and a general framework for reasoning over spatial information using predicate constraints. first a general definition of semantic constraints of a trajectory database and possible means to modify the data set to minimize violations of these constraints are given. In order to minimize the changes to the data set, the presented approaches are geared towards minimizing the number of violations. To measure the magnitude of the change of the data set, a measure of dissimilarity between the initial database and its repaired state is defined. Also, to minimize dissimilarity several simple rules of space- and time-distortion that shift inconsistent observations in space and time to remove inconsistencies are introduced. Section 4.5 extends this section to continuously valued cost functions and uses this extension in Section 4.6 to solve the migration route task. Section 4.7 extends the original definitions to the domain of spatio-temporal databases and particularly Moving Object Database (MOD) inconsistencies. The definition of constraints is extended to spatio-temporal constraints. To evaluate their ability to solve more complex settings, a large data set of trajectories is presented, constraints to formalize collision events are defined, and possible repairs are generated. The performance of the introduced approaches is measured on this data set. Finally, Section 4.9 concludes the chapter.

4.2 Motivation: Routes of Transalpine Mobility and Cultural Transfer

In any project discussing migration and trade of the past in a mountainous area, which is hard to traverse, needs to consider possible migration routes. The data set introduced in Section 1.2.2 consists of finds from 30 sites in the Alps region where human remains were found. These individuals were buried, so we assume that these locations represent settlements. However, there is no information about the routes, which connected them. While there may be more immediate ways for a domain scientist to estimate prehistoric migration routes, data analysis methods can be helpful in determining possible routes based on constraints regarding the preferences of people in historical contexts. To reconstruct the movements of the people whose remains were found and analyzed, we must incorporate knowledge about the environment that is not part of the original data set. The only information we have about these individuals' whereabouts are the places that they were buried and subsequently found. If there are multiple locations which share individuals with similar isotopic signatures, we can hypothesize that there was migration between these locations. We do not – however – know at which positions between these locations those same individuals may have lived or traveled.

Given sufficiently large data sets and sufficiently powerful models, it could become possible to extract other places of by extrapolating the point that most closely corresponds to the measured isotope values. However, as we have seen in previous chapters, the correlation between spatial position and isotope values is very indirect and burdened by multiple sources of error. This makes the granularity of spatial prediction very low and thus unusable for predicting of trade and migration routes.

What other information might we use to reconstruct routes of travel given available data? Previous work has shown that the investigated passage was been used for a very long time. What is characteristic about it is that it forms the lowest path across the Alps mountain range in this region. We may infer that peoples in this area were well aware of the geography and were able to pick routes which required the least energy to traverse. This leads us to an approach of using knowledge of the geography of a region to reconstruct possible migration routes. Given a simplified route (e.g., a line of flight connection between nearest investigation sites) we may refine the route to better resemble the actual trade routes using constraints extracted from the area's geography.

4.3 Related Work

We give an overview over the literature on several different topics related to the problems addressed in this paper. However although each body of work has yielded interesting and relevant results, none has addressed the specific problems tackled by this work, nor has provided a readily applicable "tool-chain."

4.3.1 Relational Database Repairs

Traditional database approaches *repair* [4, 8, 87] the identified inconsistencies by removing objects or by changing attribute values. Such approaches however, can not be applied directly to spatio-temporal data with inter-object constraints. Arbitrarily changing a (location, time) pair is likely to yield new inconsistencies, as the changed trajectory may reach an unreachable state, or may give an individual object unrealistically high velocity in the repaired version of the database. The main challenge in spatio-temporal data is to incorporate repair rules to span a space of semantically meaningful repairs.

4.3.2 Probabilistic Spatio-Temporal Database Repairs

The approach by Parisi and Grant[61] aims at repairing probabilistic spatio-temporal databases as defined Parker et al. [62]. In this setting, each mobile object is assigned a set of spatial regions and a probability interval defining the likelihood to be within this region. In an interpretation of such a database, the probability of a region must be within its interval and the probabilities of all regions of an object must sum up to one. Such a database is inconsistent if no interpretation exists. The approach of Parisi and Grant shows how to minimally change probability intervals in order to obtain an interpretation. The problem setting in this work can not be extended to trajectory databases.

4.3.3 Interpolation Models

A large body of research has addressed the problem of estimating the position of a spatio-temporal object between discrete observations. This important work is able to repair inconsistencies that violate the constraint that an object must be at exactly one position at any time during its lifespan. The most common approach to handle this issue is to assume linear interpolation for modeling the motion of spatio-temporal objects [64, 69, 75], but other approaches have been proposed. Tao et al. [76] introduced a framework that allows the future motion of objects to be described in a more complex manner than linear interpolation. An interesting interpretation of the problem of *dead reckoning* is that it can be viewed as a special case of the problem defined in this paper: Inconsistencies are given by objects not having a (location, time) in the time intervals defined by their discrete observations, and a repair is required to fix these inconsistencies by incurring a minimal deviation from the expected position of a moving objects. Many of the consistency-violations in trajectory databases are a consequence of the interpolation model, as any applications requires some form of interpolation between discrete measurements. The unique challenge addressed in this paper is to go a step further and repair inconsistencies incurred by an imperfect interpolation model.

4.3.4 Space-Time Approximations and Uncertainty

A different approach to address uncertainty in spatio-temporal data represents each object by a set of trajectories, so-called possible worlds, rather than by a single trajectory. Semantically, each spatio-temporal object is guaranteed to equal one of its possible worlds. Consequently, following this approach, a database is defined by a large set of possible database worlds defined by the cross product of the possible worlds of all database objects. The prevalent approach is to bound all possible trajectories of an object by a simple geometric structure in time and space, such as sheared cylinders [80, 81, 82], diamonds [59] and so called beads [41, 79] for the case of two spatial dimensions (the third dimension is time). Queries on these models include range queries [65, 82, 81] and kNN queries [80, 14]. The main problem of all these approaches is that no probability information is given for any object approximation. Thus, it is not possible to assess the probability of an object to satisfy some query predicate. In particular, this probability can be zero due to the conservative nature of these approximation models. Simple assumptions to estimate this probability, such as a uniform distribution over the conservative approximation, are often impractical: In practice, a vehicle having a fairly constant velocity between two (location, time) pairs is more likely than the same vehicle going at maximum speed, passing its destination, then performing a U-turn to race back the opposite direction in order to barely reach the second (location, time) pair in time.

4.3.5 Uncertain Spatio-Temporal Databases

Emrich et al. [20] model the motion of a spatio-temporal object by a stochastic process, such that each possible world is indeed associated with a probability. Constraints such as "Object x must not be in state s at time t" can be incorporated into this model by adapting the corresponding probabilities. More complex constraints, such as inter-object constraints that prohibit objects from being at in same state at the same time, can not be incorporated into such models as easily.

4.3.6 Linear Temporal Logic

Constraints on trajectories can be formulated using temporal logic. For instance, using Propositional Linear Temporal Logic (LTL) [19], a trajectory $T = s_1, s_2, ..., s_{|T|}$ can be described using the *eventually* operator $\diamond$ by $\diamond s_1 \diamond s_2 ... \diamond s_{|T|}$. Semantically, this LTL formulation induces a trajectory where eventually state s_1 must be visited after any number of intermediate states, then s_2 must eventually be visited after possible more intermediate states and so on. Further constraints can be formulated, e.g. to constrain the database such that no two objects may be at the same location at the same time, by applying the always operator $\Box$ to express the the rule $\forall T_1, T_2 \in \mathcal{D}, t \in T : \Box T_1(t) \neq T_2(t)$. Logical solvers for LTL [67] can efficiently find an interpretation for each trajectory such that all constraints are satisfied, if any such interpretation exists. While LTL allows formulating any semantic constraint, its main problem is that, being a logic rather than a function, it does not

allow finding an optimal solution. Thus, LTL allows checking if there exists a model that satisfies all given constraints. In most applications, the problem of finding such a model is trivia. For example, the solution of using a *serial schedule*, which avoids any inconsistency between objects by simple removing any temporal overlap between trajectories, does always work. However, while the solution based on serially scheduling each trajectory is valid, it is prohibitively expensive, since "repaired" trajectories may be extensively distorted in time. The solutions accepted by the presented approach minimize the changes to the database performed by the repair.

4.4 Approach: Finding Constraint Compliant Routes

This section introduces the concepts used to specify constraints and repair their violations in route databases. Additionally, we propose deliberately simple implementations of the above definitions and combine them to give an algorithm to remove inconsistencies from a route database $\mathcal{D}$.

Before discussing our algorithmic solutions for route database repairs in Section 4.4.4, we specify the following components:

1. route constraints and techniques for their detection
2. repair rules
3. a dissimilarity function to measure the quality of a database repair

A general solutions for the problem of fixing inconsistencies in route data suggests the following outline:

1. finding constraint violations
   we concentrate here on the inter-object *collision* constraint.
2. degrees of freedom in database manipulation
   spatial and temporal. absolute and relative.
3. applying database manipulations to constraint violations
   results in sets of rules of the form *on each constraint violation* $p \in P$*, apply repairs* $\forall_{r \in R} : r(p)$.
4. assessing the effect of a database manipulation
   assigns a score to each rule's outcome.
5. choosing a series of manipulations to generate a repaired database

This general schema will be extended to two types of route databases below.

4.4.1 Constraints

The violation of a constraint in a database $\mathcal{D}$ indicates that $\mathcal{D}$ contains wrong data. In route databases these errors may e.g., have been caused by inaccuracy of measurements, or (if moving objects were observed) faulty dead reckoning. Since the cause for the inconsistency is unknown, the only viable approach is to modify the data in order to mitigate the consequences of this lack of information.

Definition 1 (Constraint satisfaction)**.** *Let $\mathcal{C}$ be a set of Boolean constraints. A database $\mathcal{D}$ is said to satisfy $\mathcal{C}$, noted as $\mathcal{D} \vDash \mathcal{C}$, if all constraints are satisfied in $\mathcal{D}$. If $\mathcal{D} \nvDash \mathcal{C}$, then $\mathcal{D}$ is said to be inconsistent.*

Loosely speaking, a constraint can be thought of as a required property of the routes in $\mathcal{D}$. A constraint $c \in \mathcal{C}$ pertains to an individual object. In Section 4.5 we will encounter continuous constraints (not Boolean). And in Section 4.7 we will encounter constraints involving multiple routes simultaneously.

4.4.2 Repair Rule

Given some constraints, a few modifications of the data to possibly reduce constraint violations are required. For this purpose, there may be a number of route database repair rules. These rules define the set of possible repairs $\overline{\mathcal{T}^{\mathcal{R}}}$ for a given route $T \in \mathcal{D}$.

A space distorting repair allows a route to avoid inconsistencies by replacing transitions by any spatial detour leading to the same state. A detour between two to states s_i and s_j is a path starting at s_i and leading to s_j. A route repair that uses detour-based repairs only is defined as follows:

Definition 2 (Route Database Repair Rule)**.** *Let $\overline{\mathcal{D}}$ denote the set of all possible route databases. A route database repair rule $R : \overline{\mathcal{D}} \mapsto \overline{\mathcal{D}}^*$ is a function, which maps a route database $\mathcal{D}$ to a set of possible repairs.*

Definition 3 (Detour-based route repair)**.** *Let $T = [s_1, ..., s_{|T|}] \in \mathcal{D}$ denote a route. A* detour-based *repair of T is a route $T^R \in \overline{\mathcal{T}}^{dtr} := [s_1, D(s_1, s_2), ..., D(s_{|T|-1}, s_{|T|})]$. The notation $D(s_i, s_j)$ corresponds to a detour between state s_i and state s_j. The set $\overline{\mathcal{T}}^{dtr}$ denotes the infinite set of detour-based repairs of T.*

A simple detour repair rule is the spatial shift rule:

Definition 4 (spatial shift rule)**.** *This repair rule returns s vertices, which are distributed on a circle with radius δ around the original vertex v.*

$$v'_i = (x(v) + \delta \cdot sin(i/s \cdot 2\pi), y(v) + \delta \cdot \cos(i/s \cdot 2\pi)) \ , \tag{4.1}$$

where $i \in \mathbb{N}$ and $0 < i \leq s$.

4.4.3 Database Repair

The defined constraints and rules are combined to generate a data set that does not contain any constraint violations using a *database repair*. A heuristic solution will generally generate a number of possible solutions, one of which will be chosen as the best solution after a finite amount of processing time. For this purpose, a quality-measurement function $dist(\mathcal{D}, \mathcal{D}^R)$ for repairs is required upon which a ranking can be based.

Definition 5 (Database repair)**.** *Let $\mathcal{D}$ be a route database inconsistent with respect to a set of constraints $\mathcal{C}$ and let $\mathcal{R}$ be a set of repair rules. Let $\mathcal{D}^R \in \mathcal{R}^*(\mathcal{D})$ be a route database derived by iteratively applying repair rules $R \in \mathcal{R}$ to $\mathcal{D}$. If $\mathcal{D}^R \models \mathcal{C}$ holds, then the route database $\mathcal{D}^R$ is called a* database repair *of $\mathcal{D}$.*

In many cases, such as the aforementioned exemplary repair rule that allows to discard routes, one trivial way of obtaining a database repair $\mathcal{D}^R$ which satisfies all given constraints $c \in C$ is, for example, the empty database $\mathcal{D}^R = \{\}$. Given the lack of any actual route, it trivially satisfies many constraints. Hence, strictly speaking, the challenge is not only to find just any database repair, but to find a database repair having the *minimal difference* from the initial database $\mathcal{D}$.

Definition 6 (Dissimilarity function)**.** *Let $\mathcal{D}$ be a route database inconsistent with respect to a set of constraints C. Then $dist(\mathcal{D}, \mathcal{D}^R)$ is a dissimilarity function between databases, if it tracks the number of changes, penalizes unfair distribution of changes, and promotes semantically plausible changes.*

This notion of dissimilarity can then be used to define what makes a repair minimal:

Definition 7 (Minimal database repair)**.** *Let $\mathcal{D}$ be a route database inconsistent with respect to a set of constraints C. Let $dist(\mathcal{D}, \mathcal{D}^R)$ be a dissimilarity function between databases. A minimal repair $\mathcal{D}^R_{min}$ is defined as*

$$\mathcal{D}^R_{min} = argMin_{\mathcal{D}^R \in \overline{\mathcal{D}^{\mathcal{R}}}, \mathcal{D}^R \models C} dist(\mathcal{D}, \mathcal{D}^R),$$

where $\overline{\mathcal{D}^{\mathcal{R}}}$ represents the set of all possible repairs of $\mathcal{D}$.

Generic database dissimilarity functions The goal of this section is to design a generic database dissimilarity function for route databases. As the cause of a constraint violation is unknown, the only sensible approach is to limit the changes to the database as much as possible. Accordingly, the quality of a repair is assessed by the magnitude of its effect on $\mathcal{D}$.

To measure the total dissimilarity between $\mathcal{D}$ and $\mathcal{D}^R$, we can simply aggregate the dissimilarity of individual objects:

$$dist(\mathcal{D}, \mathcal{D}^R) = \sum_{T \in \mathcal{D}} dist(T, T^R),$$

where $dist(T, T^R)$ is a dissimilarity function defined on objects in the data set. If the compared objects are routes, a dissimilarity function might be the average Euclidean-distance or edit distance.

In addition, changes in $\mathcal{D}^R$ should be divided fairly among routes, in order to avoid starvation of single routes in the repaired database. Such fairness can be enforced as follows

$$dist(\mathcal{D}, \mathcal{D}^R) = \sum_{T \in \mathcal{D}} g(dist(T, T^R)),$$

where $g(x)$ is a function that monotonically increases in $\mathbb{R}^+$, such as the square function, to take into account the distances of individual routes.

4.4.4 Route Database Repairs

The components outlined above can be combined to create an algorithm to generate a database repair. As finding a minimal database repair can be NP-hard (see Section 4.7.3.1), any resulting algorithm should employ heuristics to find a good (but not necessarily optimal) repair.

In our description of these algorithms we use the following functions:

- $c : \mathcal{D} \to \mathcal{V}$ returns the set of vertices that are part of any conflict in $\mathcal{D}$.
- $R_v : \mathcal{D} \to \mathcal{D}$ is the repair function R, but limited to manipulations of the conflicting vertex v.

Instead, in the next section, we will propose approximate algorithms, which return a database repair $\mathcal{D}^R$ which may not be minimal in terms of distortion of the original database $\mathcal{D}$, or which may fail to satisfy some constraints.

4.4.4.1 Random

The simplest approach does not try to choose a good repair function at all. Instead it applies a random instance of a set of possible repair functions to a random conflicting vertex in the database. See Algorithm 1 for a detailed description.

Algorithm 1: Random($\mathcal{D}$, $\mathcal{R}$)

```
1: while c(D) ≠ ∅ do
2:     V ← c(D)
3:     v ← rnd(V)
4:     R ← rnd(R)
5:     D ← R_v(D)
6: end while
```

Applying a random repair function does not necessarily reduce the number of conflicts. As a consequence, the algorithm might not converge on a solution. A positive aspect of

this algorithm is that it does not need to evaluate several possible repairs and can instead pick one immediately.

4.4.4.2 Greedy

The more sophisticated *Greedy* algorithm uses the number of remaining constraints after applying each function to make a better choice. The Random algorithm's weak spot is its unguided choice of repair function. The Greedy algorithm considers only the local improvement of each repair. The repair yielding the lowest number of remaining constraint violations is picked and applied to $\mathcal{D}$. See Algorithm 2 for details.

Algorithm 2: Greedy algorithm

```
1: while c(D) ≠ ∅ do
2:     V ← c(D)
3:     v ← V[0]
4:     R_opt ← argmin_{R∈ℛ} ‖R(D)‖
5:     D ← R_opt(D)
6: end while
```

Given suitable rules, the Greedy algorithm can be used to find a repair quickly, but this result is unlikely to be minimal (or close to minimal). Compared to the Random algorithm, Greedy's locally optimal repairs yield a much faster convergence on a (possibly local) optimum. Inconsistencies are repaired with fewer iterations. However, the increase of complexity leads to an increase in run time. As there is no further information that could be used for deciding between multiple minima, the decision needs to be done randomly.

To find a repair that is closer to the minimal database repair, an algorithm must avoid the local minimum Greedy is prone to converge on. The following algorithm addresses this problem by combining random and greedy elements.

4.4.4.3 Simulated Annealing

The deterministic nature of greedy algorithms makes them prone to local minima. To increase the likelihood of finding a global minimum, we now describe an algorithm based on simulated annealing. See Algorithm 3 for a detailed description.

Simulated annealing [39] (SA) describes a class of algorithms that use heuristics to approximation solutions to global optimization problems. The common characteristic of these algorithms is that they become less likely to accept a bad refinement as time passes to zero in on a solution in a fixed amount of time.

By consolidating the Random and Greedy algorithms we counter the overly deterministic nature of greedy algorithms by introducing some randomness in a directed way. The goal is to find a tradeoff between longer running times and quality of the result. The result might not get the best solution in the shortest time, but an acceptable one in less time. This algorithm avoids local minima by initially choosing random repairs, then trying

Algorithm 3: SA($\mathcal{D}$, $\mathcal{R}$)

```
δ = 1
while c(D) ≠ ∅ do
    if random(]0; 1]) < δ then
        D ← Random(D, R)
    else
        D ← Greedy(D, R)
    end if
    δ ← δ − Δ_δ
end while
```

to improve on the best random result using more and more greedy approaches. In each iteration, this algorithm first decides to either perform a Random repair or a Greedy repair with increasing bias toward greediness. In the first iteration, the probability δ of performing a Greedy repair is zero. In each subsequent iteration, this probability increases by a parameter $\Delta_\delta \in [0, 1]$.

Restricting the set of possible repairs to detour-based repairs only, yields the following variant of a minimal database repair.

Definition 8 (Detour-Based Minimal Database Repair)**.** *Let $\mathcal{D}$ be a route database inconsistent with respect to a constraint C. Let $dist(\mathcal{D}, \mathcal{D}^R)$ be a dissimilarity function between databases. A minimal repair $\mathcal{D}^{dtr}_{min}$ is defined as*

$$\mathcal{D}^{dtr}_{min} = argMin_{\mathcal{D}^{dtr}=\{T_1 \in \overline{\mathcal{T}}^{dtr}_1, \ldots, T_N \in \overline{\mathcal{T}}^{dtr}_N\}, \mathcal{D}^{dtr} \models C} dist(\mathcal{D}, \mathcal{D}^{dtr}).$$

Clearly, the quality of a repair T^R of a route T depends on the quality of a chosen detour. In particular, a detour must exist, e.g. given an underlying road network, and should have similar "cost" in time and space. The assessment of the quality of a detour, has to be performed by the dissimilarity function $dist(\mathcal{D}, \mathcal{D}^R)$.

In the following section we will extend these general notions of route repair to apply them to a first problem: reconstruction of plausible migration routes between sites from the FOR 1670 data set.

4.5 Extension: Continuous Cost Constraints

Previously, we have considered route constraints predicates, i.e. any constraint violation is bad and all are equally bad. Repairs have focused on removing these constraint violations altogether. However, there are cases where one state is less desirable than another one and a better state can be found by minimizing cost constraints. In the following we will see how route constraints can be extended to allow for continuous cost functions.

4.5.1 Route Cost Constraints

The first obvious place that needs to be modified to support continuous cost is the definition of a constraint itself. Previously, a logical statement that specifies when a constraint is violated was enough. With the extension to cost-based constraints, a cost function that returns a indicator of the quality of a solution is required.

Definition 9 (Cost of constraint violations)**.** *In cases with continuous costs, constraints are given by functions, which return a scalar cost for a vertex:*

$$cost : \mathcal{V} \rightarrow \mathbb{R}^{+}$$

In place of predicate constraints (which must invariably all be fixed), the total cost of a database can now be optimized.

Definition 10 (Cost of a Database)**.** *Let $\mathcal{C}$ be a set of continuous cost functions. The total cost of a database is the sum of all cost functions applied to all states. The cost of a data set gives a sortable scalar value representing the amount of damage in a data set. It is simply the sum of all constraint violation costs:*

$$cost(\mathcal{D}) = \sum_{v_{i,j} \in \mathcal{D}} cost(v_{i,j})$$

where $v_{i,j}$ is a vertex on route i.

Contrary to previously, the continuous constraints formalization does not expect to reach a flawless state (i.e. satisfaction of constraints). Instead, it is concerned with finding a configuration corresponding to minimum cost within the constraints. A consequence of this new view is that cost functions are not required to reach zero. Since there are very few properties required of suitable cost functions (and thus few can be assumed to optimize efficiently), guaranteeing a minimum is not possible in acceptable time. And a minimum that was found is not guaranteed to be global.

4.5.2 Repair Rules

The same database repair rules as previously introduced can be applied. Definition 2 still applies here.

4.5.3 Database Dissimilarity Function

A *Database Repair* is not applicable, because we cannot hope to reach an ideal state. Instead we define a database modification:

Definition 11 (Database Modification)**.** *Let $\mathcal{D}$ be a route database, $\mathcal{C}$ a set of constraint cost functions, and $\mathcal{R}$ a set of repair rules. Let $\mathcal{D}^R \in \mathcal{R}^*(\mathcal{D})$ be a route database derived by iteratively applying repair rules $R \in \mathcal{R}$ to $\mathcal{D}$. We call $\mathcal{D}^R$ a Database Modification.*

To be able to define an optimal database modification we need a notion of minimality. For this purpose we can use the same *Dissimilarity Function* (see Definition 6) as before.

Definition 12 (Optimal Database Modification)**.** *Let $\mathcal{D}$ be a route database and C a set of continuous constraints. Let $dist(\mathcal{D}, \mathcal{D}^R)$ be a dissimilarity function between databases. A optimal modification $\mathcal{D}^R_{min}$ is defined as*

$$\mathcal{D}^R_{min} = \underset{\mathcal{D}^R \in \overline{\mathcal{D}^\mathcal{R}}}{\operatorname{argmin}} dist(\mathcal{D}, \mathcal{D}^R) + \alpha \cdot \big(cost(\mathcal{D}^R) - cost(\mathcal{D})\big) \ ,$$

where $\overline{\mathcal{D}^\mathcal{R}}$ represents the set of all possible repairs of $\mathcal{D}$. α is a parameter to weight the changes in the database against the severity of the constraint violation cost.

This extension does not need to modify the applicable repair rules and by extension it incorporates existing route database dissimilarity functions. The same concrete examples from the previous section can be used.

4.5.4 Route Database Repairs on Continuous Cost Functions

As in Section 4.4 the newly introduced components can be integrated into the existing algorithm to find an optimal database modification. As before the specified algorithms use the function $c : \mathcal{D} \to \mathcal{V}$ to get the set of vertices that are part of a conflict in $\mathcal{D}$ and a repair function $R_v : \mathcal{D} \to \mathcal{D}$, which can modifies a conflicting vertex v.

4.5.4.1 Greedy

Contrary to before, the Greedy algorithm now has some additional information to go by. It can now pick the modification that resulted in the lowest remaining cost, or use single violations' costs to prefer a violation with the highest cost.

Algorithm 4: Greedy($\mathcal{D}$, $\mathcal{R}$)

```
1: repeat
2:    c ← cost(D)
3:    for all V ∈ D do
4:       r ← argmin_{R∈R} cost(V)
5:       if cost(r(V)) < cost(V) then
6:          V ← r(V)
7:       end if
8:    end for
9: until c ≤ cost(D)
```

4.6 Application: Reconstructing Routes Between Archaeological Sites in Alps Region

This application is concerned with finding potential routes between known spatial points. This application considers the slope of a given path and constructs routes that prefer low slopes while minimizing deviation from the shortest path.

4.6.1 Repair Strategy

The objective of this approach is to find plausible routes between known settlements in the Alps region. To implement this scenario the components discussed in the previous section need to be specified to fit this application.

4.6.1.1 Cost Function

What determines if a route is "plausible" is expressed by the following constraint:

$$cost(v_{i,j}) = \frac{z(v_{i,j-1}) - z(v)}{d(v_{i,j-1}, v)} \tag{4.2}$$

where $d(v_1, v_2) = \sqrt{(x(v_1) - x(v_2))^2 + (y(v_1) - y(v_2))^2}$ is the Euclidean distance between the coordinates of the given vertices and $z(v)$ is the altitude at coordinate $(x(v), y(v))$. This can be interpreted as the inclination of the path leading up to vertex v, i.e. minimizing this value for all vertices finds a path that is easier to walk than the input. For the entire database, all segment costs are summed:

$$cost(\mathcal{D}) = \sum_{v_{i,j} \in \mathcal{D}} cost(v_{i,j}) \tag{4.3}$$

4.6.1.2 Repair Rule

The possible repairs are to shift the coordinate of each point by a fixed distance using the spatial shift rule introduced above.

4.6.1.3 Database Distance

The databases are compared by their absolute route length. This causes routes that are detours to be considered worse than direct connections (with which the algorithm starts). A modification (which must (at least initially) be a detour) is only accepted if the cost induced by reduced inclination outweighs the required detour's costs.

$$dist(\mathcal{D}, \mathcal{D}^R) = \left| len(\mathcal{D}) - len(\mathcal{D}^R \right|), \tag{4.4}$$

where $len(\mathcal{D}) = \sum_{v_{i,j}, v_{i,j+1}} d(v_{i,j}, v_{i,j+1})$ is the total length of all routes in $\mathcal{D}$.

4.6.1.4 Combined Cost Function

Applying the outlined choices to Definition 12 gives the following total cost function:

$$cost(\mathcal{D}, \mathcal{D}^R) = dist(\mathcal{D}, \mathcal{D}^R) + \alpha \cdot \left(cost(\mathcal{D}^R) - cost(\mathcal{D})\right) , \tag{4.5}$$

where the parameter α specifies the tradeoff between distance and inclination.

This parameter α is picked empirically, because the author was unable to find any applicable information in the relevant literature. We conjecture that for steep slopes relatively far detours were acceptable as some immobile members of the group would not have been able to pass these steep slopes.

4.6.2 Experimental Evaluation

To restore a possible migration route across the Alps, the following experiment identified ten settlements that were on a plausible short route across the Alps (see Figure 4.2a). These sites were connected to form the shortest possible route and then subjected to the repair algorithm. The evaluation will focus on improving this path using the Greedy algorithm.

To determine the cost of a route, the individual legs are broken up into linear pieces at 0.02 degree intervals. Each leg's slope is calculated according to Formula 4.2. The total cost of a route is the average squared slope over these pieces to penalize higher slopes disproportionately. The chosen weight parameter was $\alpha = 1000$.

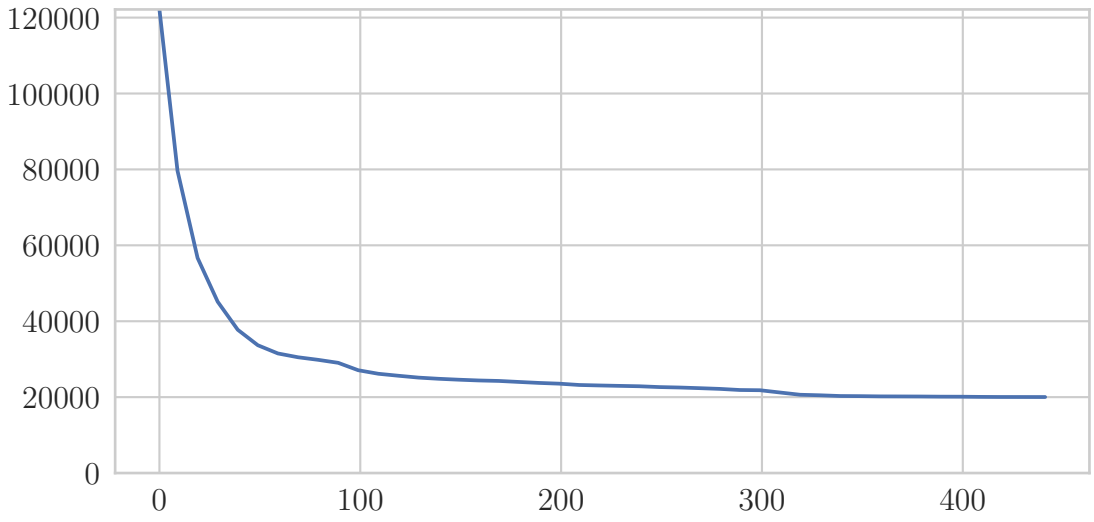
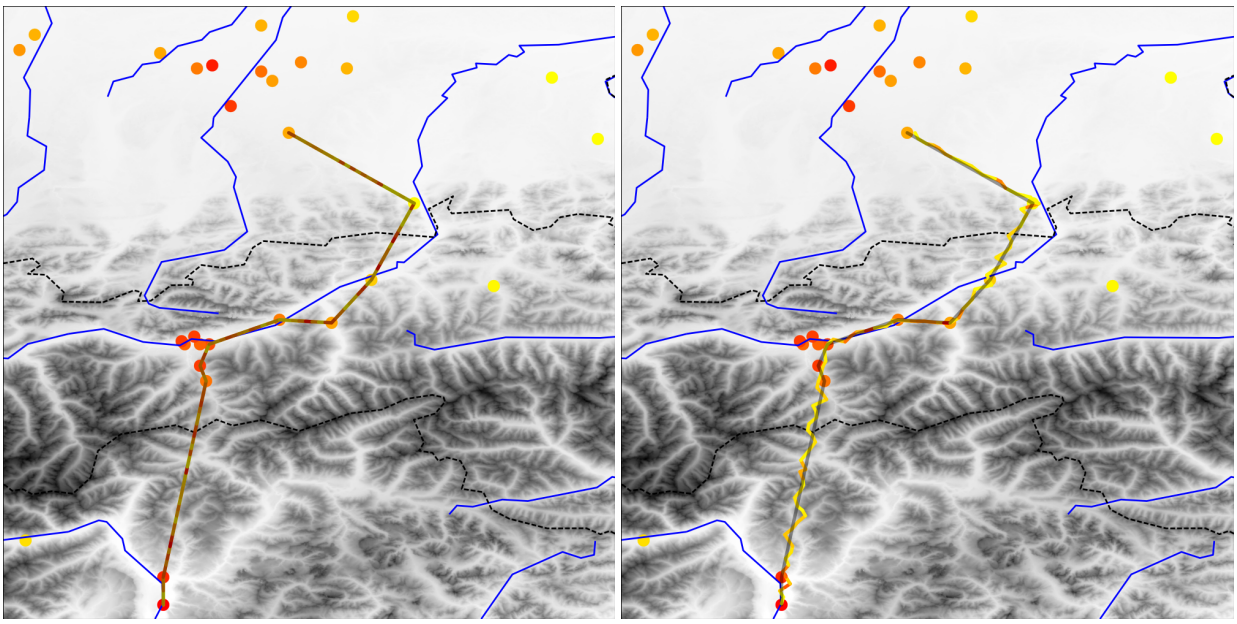


Figure 4.1: Change of costs with iterations.

The chosen δ is 0.02 degrees. The algorithm converges after 442 iterations, but has a strong outlier in a comparatively flat region of the map. Figure 4.1 shows the progression of the remaining costs after n iterations. Figure 4.2b shows the route after convergence.



(a) Initial state. The cost of the initial route using quadratic slope penalizer is 122150.

(b) Repaired route after 442 iterations. The remaining cost of the route after repair is 20026.

Figure 4.2: Costs of the presented routes.

Since there are no inter-object constraints in this application, the sequential local optimization of individual routes yields the same result as a global optimization. In the next section we will see an extension of the constraint concept from routes to inter-object constraints.

4.7 Extension: Spatio-Temporal Inter-Object Constraints

In the previous sections we have looked at small scale static location data. However, there is a large and growing corpus of more complex spatial data that is subject to constraints. At the end of 2014, there were nearly 7 billion mobile subscriptions worldwide [58]. Along with miniaturization of computing and sensing devices and GPS and RFID technologies, this has lead to a proliferation of location data, generating extremely large volumes of location-in-time data: petabytes of location-based (i.e., spatio-temporal) data are generated every day [47]. The management of (*location*, *time*) information about mobile entities is essential for a variety of application domains, ranging from navigation and efficient traffic management to emergency/disaster rescue management, and environmental monitoring. Essentially, every application requiring some form of Location Based Services (LBS) [70] needs efficient techniques for storage, retrieval and query processing of spatio-temporal data—topics studied in the field of Moving Objects Databases (MOD) [29].

Physical factors, such as the imprecision of sensing devices and communication links, often cause the location data to be inaccurate and noisy. In addition to this problem—even

with perfect sampling accuracy—the data intended to capture a continuous motion can be represented only at discrete time-instances. Moreover, data records can be obsolete as users may update their location infrequently, e.g., due to bad connectivity or to preserve battery life. Thus, one has to cater to the uncertainty as a natural factor when considering the representation of spatio-temporal data [13]. A complementary observation is that data sources may be various heterogeneous devices: roadside-sensors, weather stations, satellite imagery, (mobile) weather radar, crowd sourced observations, ground and aerial LIDAR—to name but a few. Having multiple sources may not only cause type-mismatch issue, but also generate conflicting location information about the same object and cause problems in reconciling the data [90]. Complementary to uncertainty, the above contexts may cause other types of semantic inconsistencies that have not been addressed so far. Namely, a user posing a continuous k-Nearest Neighbor (k-NN) query, may be presented with an answer containing two (or more) vehicles that "have collided." This is a simple example of violating the basic semantic constraint that two objects cannot be in the same place at the same time. Such a violation may be due to imprecise location-samples. Also, it often arises from the use of interpolation (linear, Bezier, etc.) in-between observed samples [22].

This section presents novel types of constraints in the context of large MODs. In contrast with the introduction to this chapter, the data now also contains time information (i.e. spatio-temporal data), which allows us to look at changes in the data, and in particular how objects interact. A spatio-temporal database $\mathcal{D}^{ST}$ stores triples (*oid*, *location*, *time*), where $oid \in \{o_1, ..., o_N\}$ is a unique object identifier, $location \in \mathcal{S}$ is a spatial position in space and $time \in \mathcal{T}$ is a point in time. Semantically, each such triple corresponds to the location of object o_i at some time. In $\mathcal{D}$, an object can be described by a function $tr_{o_i} : \mathcal{T} \to \mathcal{S}$ that maps each point in time to a location in space $\mathcal{S}$; this function is called *trajectory*. The corresponding trajectory database is denoted as $\mathcal{D} = \{tr_{o_1}, ..., tr_{o_N}\}$. This approach assumes a discrete and finite space of possible states $\mathcal{S} = \{s_1, ..., s_{|\mathcal{S}|}\}$ and a discrete and finite space of points of time $\mathcal{T} = \{0, ..., s_{|\mathcal{T}|-1}\}$. This assumption, common in MOD literature, is mandatory to allow a finite representation of arbitrary trajectories. Hence, we have inconsistencies in trajectory data that are consequences of the model based on discrete approximation of continuous phenomena (motion).

In addition, assuming that the location of an object o_i is known for any point in time is unrealistic as the location of object o_i can only be determined at discrete time-instants. The frequency of location-samplings is also limited by physical constraints, such as the availability of a GPS signal. Between discrete observations, the position of a moving object has to be estimated via some type of interpolation (common linear approximation, Bezier curves, polynomial splines, etc.). These estimations are based on incomplete information, and thus, may be imprecise.

In this section, the goal is to alleviate this problem by repairing constraint violations using the presented approach.

4.7.1 Constraints

The addition of time allows for more complex constraints than we have previously considered. An example of an *Object Constraint* is the constraint "An object must not enter a specified area R on Sunday between 2am and 5am." This constraint can be formally expressed as

$$\forall(tr_o \in \mathcal{D}), \forall(t \in [\text{Sunday 2am}, \text{Sunday 5am}]) : tr_o(t) \notin R.$$

In contrast, an *Inter-object constraint* may be defined between trajectories, such as "two objects must not be in the same place at the same time" which can be expressed as

$$\forall(tr_{o_i}, tr_{o_j}, i \neq j), \forall t : tr_{o_i}(t) \neq tr_{o_j}(t).$$

In practice, constraints involving more than one object lead to hard optimization problems, as a single repair of one trajectory may have a large number of consequences on the constraints involving other objects. Section 4.7.3.1 will show that such constraints lead to NP-hard optimization problems. Since we are considering the general case, we will be considering such hard inter-object constraints in our experimental evaluation in Section 4.8.

Definition 13 (Spatio-Temporal Constraint)**.**

The focus of this section is on the constraints pertaining to (co)locations of objects like, e.g., *two objects must be within certain distance from each other* or *two objects can not be at the same location at the same time*. In this context, $\mathcal{D}$ is considered to be *inconsistent* with respect to a constraint c, if c is violated by (some trajectories in) $\mathcal{D}$. The predicate $c(\mathcal{D})$ yields true if and only if $\mathcal{D}$ satisfies c. Generally speaking we have the following:

Definition 14 (Inconsistency)**.** *Let $\mathcal{D}$ denote a trajectory database and let C be a spatio-temporal constraint, then $\mathcal{D}$ is denoted as* inconsistent *with respect to C, if constraint C is violated by (some trajectories in) $\mathcal{D}$. The predicate $C(\mathcal{D})$ is defined such that $C(\mathcal{D})$ yields true if and only if $\mathcal{D}$ satisfies C.*

4.7.2 Repair Rules

With the addition of time to the data, new repair rules can be defined. The rules presented here are in addition to the space distortion repair defined in Section 4.4, which of course still apply. A time distorting repair allows a trajectory to avoid inconsistencies by increasing or decreasing its velocity when traversing its sequence of states (i.e., arriving earlier or later in a given location). A waiting-based database repair allows a trajectory to avoid inconsistency by repeating, thus semantically waiting in, any state.

A possible instantiation of this model is defined as follows

Definition 15 (Waiting-based trajectory repair)**.** *Let $T = [s_1, ..., s_{|T|}], T \in \mathcal{D}$ be a trajectory. A* waiting-based *repair of T is a trajectory $T^R \in \overline{T}^{wait} := [s_1^+, ...s_{|T|}^+]$. Here, the*

notation s_i^+ corresponds to a sequence of $k \in \mathbb{N}$ repeats of state s_i. The set $\overline{\mathcal{T}}^{wait}$ denotes the infinite set of possible wait-based repairs of T. The waiting-based repair rules R is thus defined as

$$(\mathcal{D}, \mathcal{D}') \in R \Leftrightarrow \mathcal{D} = [T_1, ..., T_N] \wedge \mathcal{D}' = [(T_1|T_1^R), ..., (T_N|T_N^R)],$$

where the | operator denotes an alternative.

By constraining the set of possible repairs to waiting-based repairs only, the problem of finding a minimal database repair (c.f. Definition 7) can be stated in a more specific form as follows.

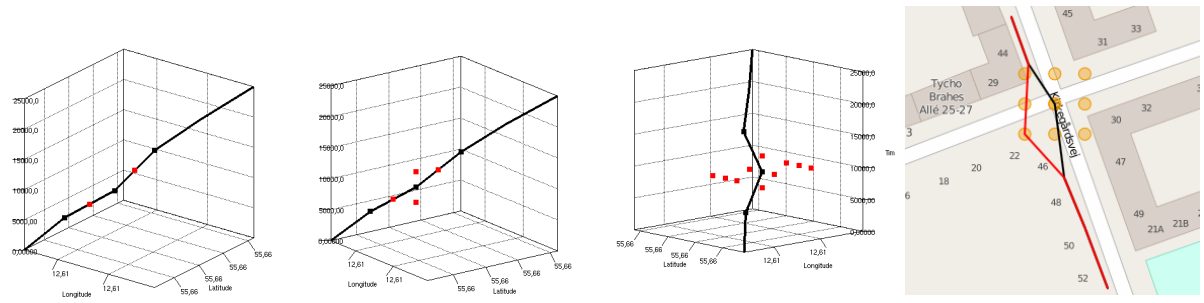
Definition 16 (Waiting-based minimal database repair)**.** *Let $\mathcal{D}$ be a trajectory database inconsistent with respect to a semantic constraint C and let $dist(\mathcal{D}, \mathcal{D}^R)$ be a dissimilarity function between databases. A minimal repair $\mathcal{D}_{min}^{wait}$ is defined as:*

$$\mathcal{D}_{min}^{wait} = argMin_{\mathcal{D}^{wait}=\{T_1 \in \overline{\mathcal{T}}_1^{wait}, ..., T_N \in \overline{\mathcal{T}}_N^{wait}\}, \mathcal{D}^{wait} \models C} dist(\mathcal{D}, \mathcal{D}^{wait}).$$

To complete our tool set for trajectory database repair, we extend the spatial repairs given in Section 4.4.2 by time-distortion repair rules and combined space-distorting and time-distorting rules:

- *Spatial domain:* Manipulating the spatial position of a trajectory vertex has also impact on the speed of the movement.
- *Time domain:* The manipulation of a vertex v back in time implies that the movement from the previous vertex to v is slowed down and the movement from v to its subsequent vertex is sped up. The manipulation of v forward in time has the opposite effect. Note that the time manipulation of a vertex is constrained by its predecessor and its successor. Manipulating the time of v beyond the times of its predecessor or its successor yields anomalous movement in the spatial domain.
- *Time and spatial domains*: Obviously, the spatial and temporal manipulation can be combined. A special case of spatio-temporal manipulation is the manipulation of v along the spatio-temporal path to its predecessor or its successor.

To identify the vertex to be repaired to remedy a constraint violation, we always consider the vertices closest to the violation point on both involved trajectories. This reduces the difference between the input database and the generated output database, which is a requirement of finding a minimal database repair. For these reasons, our experimental evaluation limits potential manipulations of a given database $\mathcal{D}$ to manipulation of existing vertices. Each vertex can be manipulated in either the spatial or the time domain. They can be moved by constant or relative values. Repairs only affect existing vertices. $v(p, T)$ maps a collision to its spatially closest vertex on trajectory T. $v_p(p, T)$ is either $v(p, T)$'s previous vertex or (if none exists) a linear interpolation backwards in time. $v_f(p, T)$ is either $v(p, T)$'s following vertex or (if none exists) a linear interpolation forwards in time.



(a) Resample Rule (b) Temporal Rule (c) Spatio-Temporal Rule (d) Spatio-Temporal Rule

Figure 4.3: Repair Rules

Space-Distortion Repairs change the location (x, y) of vertices. This does not change the sequence of vertices, because their order is given by their temporal component. We assume the sequence of vertices v_p, v, v_f. If we manipulate x and y of v, the speed between v_p, v and/or v, v_f also changes.

A *Time-Distortion Repair* manipulates only the time component of a vertex. A special case *Relative Time-Distortion Repair* shifts time relatively to total difference. The spatial position of the object is not changed. No anomalies can be generated by relative repairs. However, on the downside, the algorithm will not terminate, if v_f also violates the same constraint. The algorithm can not solve the case where a constraint violation occurs on a trajectory consists of multiple sequenced vertices with the same t.

To generate repair rules, we combine these vertex manipulations. Throughout this section, the input to a rule is the repair triple v_p, v, v_f, where v is the vertex to repair, v_p is the predecessor of v, and v_f is the successor of v in the trajectory. Furthermore, a vertex v is characterized by the triple $(v.t, v.x, v.y)$ representing the time, the x position and the y position of v, respectively. Based on these observations, we define the following three rules:

The Resample Rule uses relative time repair.

Definition 17 (Resample Rule)**.** *Given repair triple v_p, v, v_f, the Resample Rule returns two vertices v'_1 and v'_2, where*

$$v'_1 = \frac{v_p + v}{2}$$
$$v'_2 = \frac{v_f + v}{2}$$

The two vertices returned by the Resample Rule are located in time and space half the way forward and backward around vertex v. The idea is that a vertex between two other vertices can be moved by a fraction of the total distance between the other vertices. This reflects the measurement precision (as indicated by the distance between the measurements) and returns two candidates that differ by less than the sampling rate.

The Temporal Rule adds temporal repairs.

Definition 18 (Temporal Rule)**.** *Given repair triple v_p, v, v_f and time distortion Δt, the Temporal Rule returns the two vertices returned by the Resample Rule plus the two vertices v'_3 and v'_4, where*

$$v'_3 = (v.t - \Delta t, v.x, v.y)$$
$$v'_4 = (v.t + \Delta t, v.x, v.y)$$

if $v.t - \Delta t \geq v_p.t$ and $v.t + \Delta t \leq v_f.t$

This rule shifts the vertex either forward or backward on the trajectory, or changes the trajectory by speeding a segment up (reaching the half way mark on the way from v to v_f at the same time that v was originally reached) or slowing a segment down (reaching the half way mark on the way from v_p to v at the same time that v was originally reached).

A final rule (the Spatio-Temporal Rule) adds eight absolute spatial distortions. Besides the Time-Distortion Repairs, eight location distortions are provided. This is a result of the number of combinations that can be made adding and subtracting Δd in latitude and longitude.

Definition 19 (Spatio-Temporal Rule)**.** *Given repair triple v_p, v, v_f, time distortion Δt, and space distortion Δs, the Spatio-Temporal Rule returns the following ten vertices:*

$$v'_3 = (v.t - \Delta t, v.x, v.y)$$
$$v'_4 = (v.t + \Delta t, v.x, v.y)$$
$$v'_5 = (v.t, v.x - \Delta s, v.y)$$
$$v'_6 = (v.t, v.x + \Delta s, v.y)$$
$$v'_7 = (v.t, v.x, v.y - \Delta s)$$
$$v'_8 = (v.t, v.x, v.y + \Delta s)$$
$$v'_9 = (v.t, v.x - \Delta s, v.y - \Delta s)$$
$$v'_{10} = (v.t, v.x + \Delta s, v.y + \Delta s)$$
$$v'_{11} = (v.t, v.x - \Delta s, v.y + \Delta s)$$
$$v'_{12} = (v.t, v.x + \Delta s, v.y - \Delta s)$$

if $v.t - \Delta t \geq v_p.t$ and $v.t + \Delta t \leq v_f.t$

Figure 4.3 gives an overview of these three rules Figure 4.3a shows the shift on the segment and Figure 4.3b shows the additional time shift. Figures 4.3c and 4.3d show all ten options. The effectiveness of these rules will be evaluated later on. We expect the Spatio-Temporal Rule to perform best, as it offers most alternatives and incorporates absolute spatial shifts. The Resample Rule and Temporal Rule use relative shifts which brings the discussed disadvantages.

4.7.3 Dissimilarity Functions

In addition to measuring the distance between a database $\mathcal{D}$ and its repair $\mathcal{D}^R$, dissimilarity functions can be used for increased fairness. We introduce the concepts of *intra-object fairness* and *inter-object fairness*, which can be utilized to obtain a semantically good database repair $\mathcal{D}^R$.

Besides $dist(\mathcal{D}, \mathcal{D}^R)$ other measures can be used to assess spatio-temporal repair algorithms, e.g., number of repairs, and run time. The spatial dissimilarity function introduced before is one possible approach to assess spatio-temporal databases. However, specialized dissimilarity functions should be preferred to avoid unforeseen loopholes. The Weighted Euclidean Distance function is a possible extension, which adjusts the influence of the time attribute relative to the spatial dimensions.

A spatio-temporal dissimilarity function, should force a good solution to evenly divide changes of a trajectory over time. For example, changing a trajectory by adding two short waits at two distinct times may be considered a less severe change than adding a long wait once. This can be achieved by adding an exponent to the differences:

$$dist_{weighted}(tr, tr^R) = \sum_{i \in [1,|tr|]} \left(w_y(v_{i_y} - v^R_{i_y})^2 + w_x(v_{i_x} - v^R_{i_x})^2 + w_t(v_{i_t} - v^R_{i_t})^2 \right)^{\frac{1}{2}}.$$

Finally, as a different approach to avoid unfair distribution of changes, the third function is based on the Maximum Distance:

$$dist_{max}(tr, tr^R) = \sum_{i \in [1,|tr|]} max\{(v_{i_y} - v^R_{i_y}), (v_{i_x} - v^R_{i_x}), (v_{i_t} - v^R_{i_t})\}.$$

4.7.3.1 Complexity Analysis

The consideration of inter-object constraints much increases the complexity of the search space. Since the cost function is designed to minimize the changes to the data set, its optimal state is the initial state. This precludes any optimization based approaches to improve run-time. In this section we give a formal proof that the presented problem is indeed hard.

The goal of this work is to efficiently compute, for a given trajectory database $\mathcal{D}$ and a set of semantic constraints C, a minimal repair $\mathcal{D}^R_{min}$ of $\mathcal{D}$. This problem falls into the class of constraint satisfaction problems and we show here that it is NP-hard. For this purpose, we show that the simpler problem of finding *any* repair is already NP-complete.

Lemma 1. *Given a trajectory database $\mathcal{D}$, a set of inter-object constraints C and a set of repair actions A, the problem of deciding whether there exists a repair $\mathcal{D}^R$ which is derived from $\mathcal{D}$ using rules in A, such that $\mathcal{D}^R \models C$ is NP-complete.*

Proof. Let $\mathcal{D}$ be a database of arbitrary trajectories, and let A be repair actions such that for each trajectory $T_i \in \mathcal{D}$ there exists exactly one possible repair. For each $T_i \in \mathcal{D}$, let p_i denote the unrepaired trajectory T_i, and let $\hat{p}_i$ denote the repaired trajectory which is

derived by applying the only possible repair in A to T_i. Furthermore, let C be a set of inter-object constraints such that each constraint $c_{s,t} \in C$ requires that at least one object must be in state s at time t. Let $c_{s,t}(\mathcal{D}, A) \subseteq \bigcap_{1 \leq i \leq N} \{p_i, \hat{p}_i\}$ denote the set of all possible trajectories that satisfy constraint $c_{s,t}$, i.e., all possible trajectories that are located in state s at time t. Since each constraint $s_{s,t}$ requires at least one trajectory to be in state s at time t, the constraint $s_{s,t}$ can be rewritten as the disjunction of all trajectories satisfying this constraint:

$$c_{s,t} = \bigvee_{p \in c_{s,t}(\mathcal{D},A)} p\,.$$

This Boolean formula returns true iff the constraint $c_{s,t}$ is satisfied. For all constraints to be satisfied, the conjunction of all these disjunctions yields the following Boolean formula:

$$\bigwedge_{c_{s,t} \in C} \bigvee_{p \in c_{s,t}(\mathcal{D},A)} p\,.$$

This formula returns true, iff a given database repair $\mathcal{D}^R \in \{p_1, \hat{p}_1\} \times \{p_N, \hat{p}_N\}$ satisfies all constraints in C. Consequently, the problem of finding a valid repair of $\mathcal{D}$ is equivalent to the satisfiability problem of the above Boolean formula. This satisfiability problem, known as *k-SAT*, is known to be NP-complete. □

Due to the hard nature of the problem, we do not attempt to give an exact algorithm to find an optimal database repair. In the evaluation presented in the following section, we instead use heuristic solutions that yield a database repair with sufficiently low dissimilarity to the initial database using the approximate algorithms introduced in Section 4.4.4.

4.8 Application: Finding Object Collisions

There are several alternatives for spatio-temporal constraints. In this section, we consider the following very general constraint: "Two objects must not be within a threshold of ε meters of each other at any time." This constraint is formally expressed as

$$\forall(tr_{o_i}, tr_{o_j}, i \neq j), \forall t : dist(tr_{o_i}(t), tr_{o_j}(t)) > \varepsilon\,.$$

This constraint is able to ensure that objects with a spatial extent of ε never occupy the same space at the same time, or that objects do not get too close to each other.

4.8.1 Data set

To apply the techniques introduced in the previous section, we gathered a large data set of moving object data. The spatio-temporal data set that we are using consists of workout GPS data, i.e., running and hiking GPS-traces obtained from Endomondo[1]. For each GPS trace of a workout, a trajectory is stored in $\mathcal{D}$ using linear interpolation, which is the main

[1] `https://www.endomondo.com`

source of inconsistencies. The service is most popular in Scandinavia, so most workouts are located in cities there. The data set we used was from the area of Copenhagen, which has 652854 vertices. In a data cleaning step, we modified the retrieved data in the following ways:

1. outlier GPS signals yielding a speed of more than 50 kilometers per hour and
2. reset all starting times to zero, to create collisions for this evaluation.

4.8.2 Repair Strategies

In this evaluation, we use four straightforward algorithms as a baseline. These four algorithms randomly pick a conflicting GPS-signal p that is adjacent to a conflicting trajectory segment. The p is distorted by

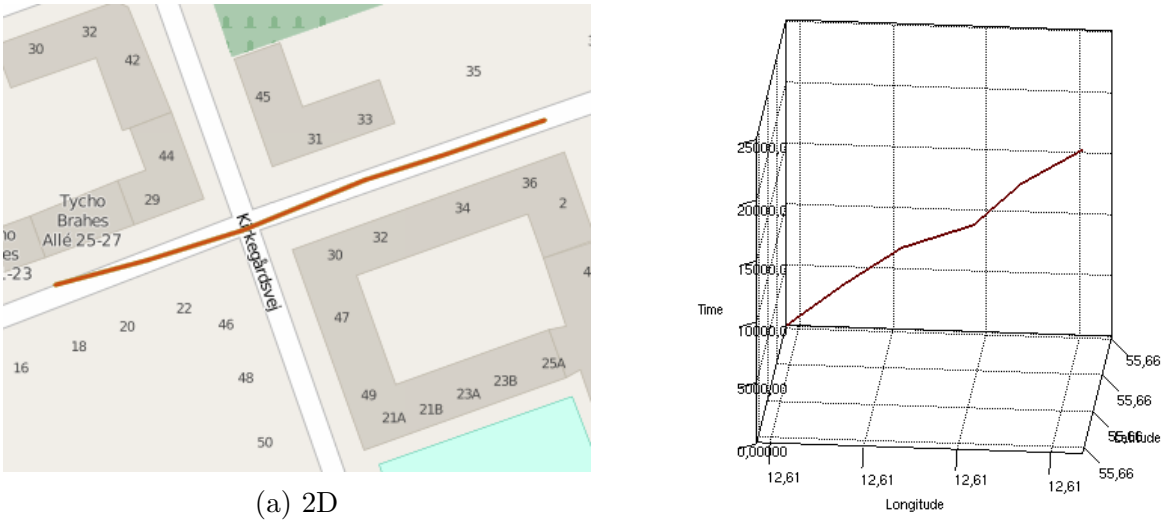
1. moving its time-stamp by a fixed time towards the next GPS-signal time stamp (*Absolute Time-Distortion*), or
2. by moving its time-stamp half-way to the time of the next GPS-signal (*Relative Time-Distortion*), or
3. moving its location a fixed distance towards the location of the next GPS-signal (*Absolute Location-Distortion*), or
4. by moving its location half-way to the location of the next GPS-signal (*Relative Location Distortion*).

Parameters The Time-Distortion Repairs can be influenced by setting the absolute time shift Δt to different values. The Location-Distortion Repair's absolute time shift value is controlled by the parameter Δd. The relative distortion repairs always use one half of the segment's extent in time or space.

4.8.3 Examples

This section illustrates the presented repair rules by applying them on simple example violations involving only a few trajectories. These situations were manually created to demonstrate the behavior of the algorithms in extreme cases.

Identical Trajectories A particularly problematic situation is duplication of the same trajectory in the data set. No single modification can solve the entire set of constraint violations at a time and there are elegant and less elegant solutions. Figure 4.4 shows three identical trajectories on a map and including the time axis. The effect of various configuration of repair rules can be seen in Figure 4.5. This test case is an example of the perceived smoothness of a repair (see Section 4.8.5.5). The dissimilarity functions suggest



(a) 2D

(b) 3D

Figure 4.4: Example: Identical trajectories.

that the Spatio-Temporal Rules solved the inconsistency more smoothly than using the Temporal Rules or the Resample Rules. The baseline time-distortion and space-distortion repairs are also able to repair these situations. Figure 4.5d shows a solution based on the absolute time-distortion repair.

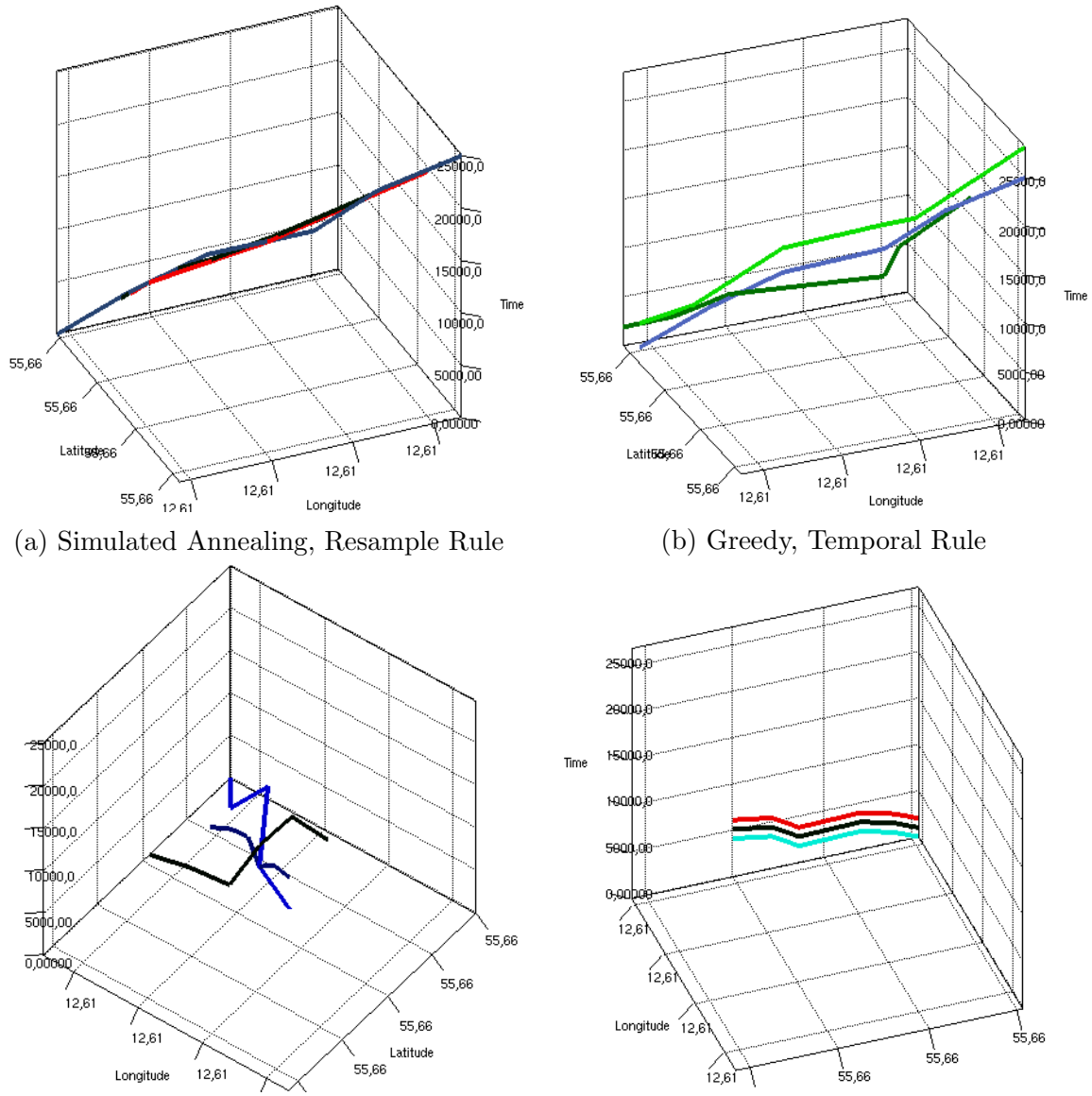
Constraint violations at trajectory endings Trajectory endings are problematic locations for constraint violations, because relative repair strategies cannot change the endings of a trajectory. Figures 4.6a and 4.6b show examples of this type of constraint violation.

Violations at consecutive vertices A further challenge are consecutive collisions involving unmoving vertices. These inconsistencies cannot be solved by Time-Distortion Repairs, while the Space-Distortion Repairs solve them easily. Time-Distortion Repairs do not terminate in this scenario[2]. An example solution can be seen in Figure 4.7.

4.8.4 Implementation

This section discusses aspects of the implementation that are relevant to the interpretation of the results.

[2]Unless otherwise stated, non-termination was assumed after 1000 applied fixes had not yielded a result.



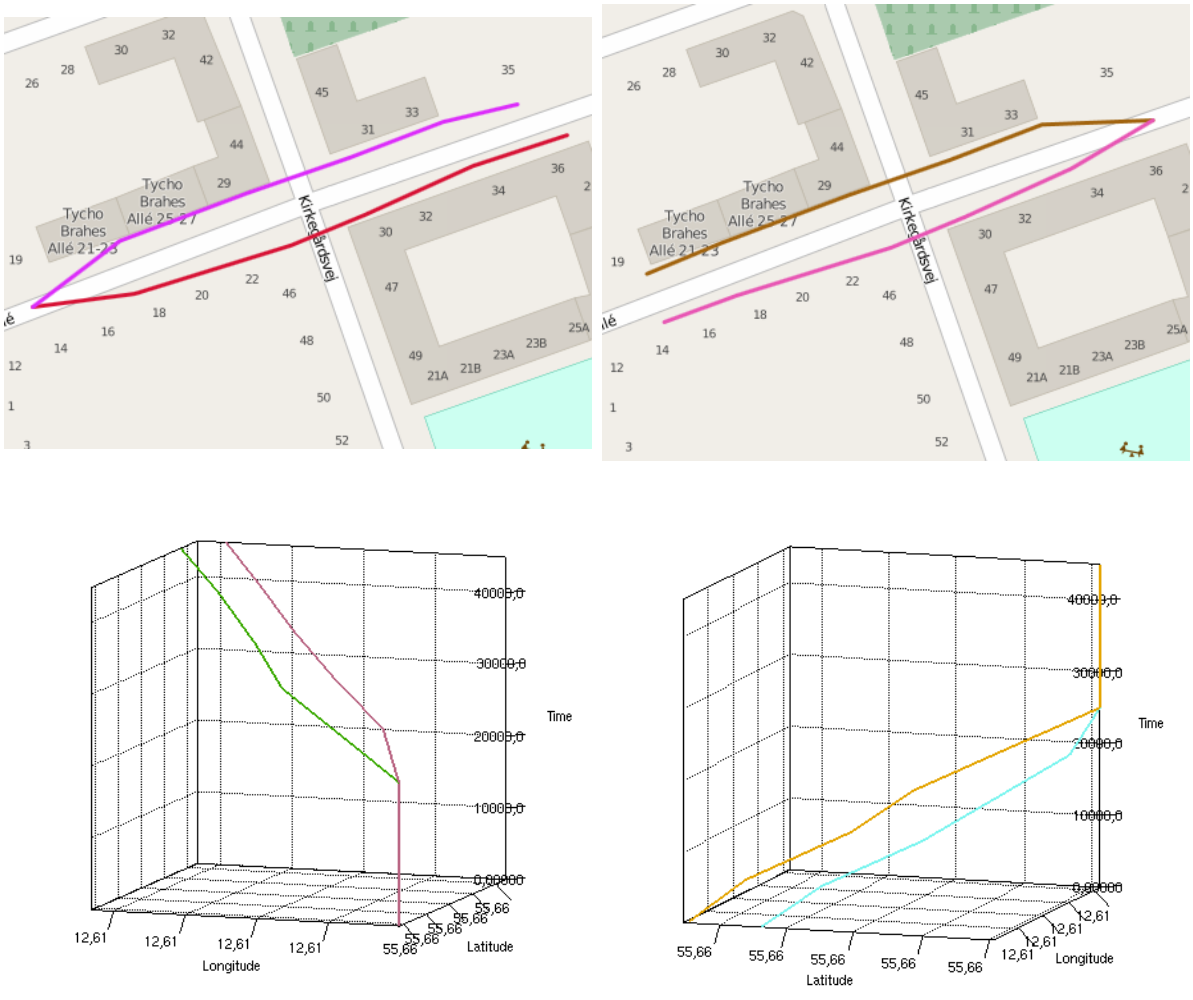
(a) Simulated Annealing, Resample Rule

(b) Greedy, Temporal Rule

(c) Greedy, Spatio-Temporal Rule

(d) Absolute Time Distortion Repair

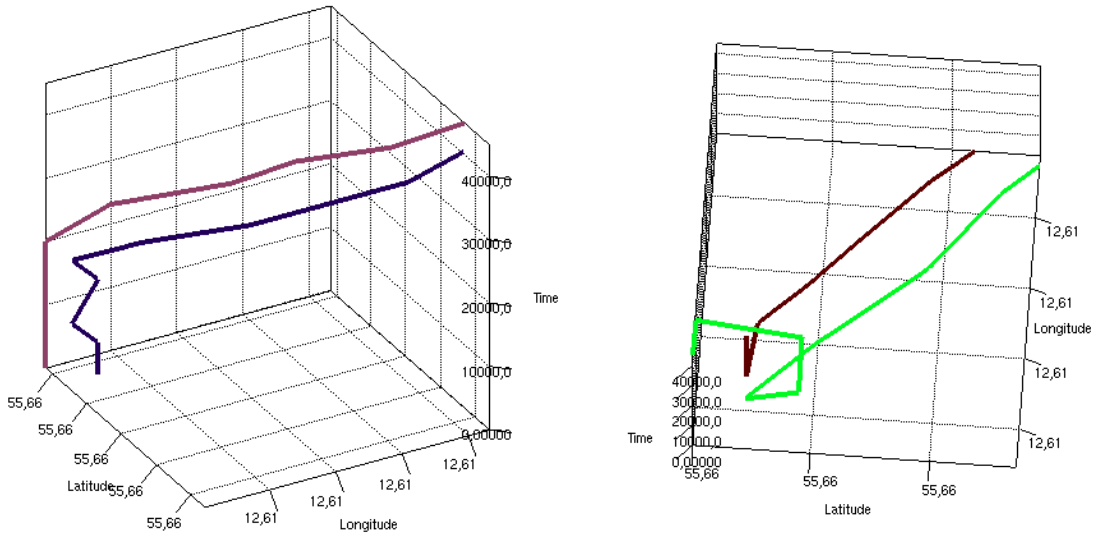
Figure 4.5: Effected repairs of identical trajectories with Closest-Point Distance and $\epsilon = 10$



(a) Identical start.

(b) Identical end.

Figure 4.6: Test cases with identical endings.



(a) Collision at trajectory start (Greedy, Resample Rule)

(b) Collision at trajectory end, (Greedy, Spatio-Temporal Rule)

Figure 4.7: Resolution of constraint violation at trajectory start or end. Closest-Point Distance and $\epsilon = 10$.

4.8.4.1 Collision Detection

The implementation uses an R^*-tree (from the ELKI-framework [1]) as an index structure to speed up collision detection. The constraint is that two objects must not be closer than ϵ to each other. ϵ can be changed in order to alter the number of detected collisions. Unless otherwise specified, the default value is $\epsilon = 3$.

To detect collisions, we use a spatio-temporal R^*-tree to index the set $\mathcal{S}$ of all trajectory segments defined by two successive GPS signals of the same object, using time as a third dimension. Each trajectory segment s is minimally bounded by a rectangle $\square(s)$ and added to the tree. Thus, each leaf of the R^*-tree is a single rectangle pointing to the exact representation of the approximated trajectory segment. To find all initial collisions, we perform a similarity self-join [12] using ε as the similarity threshold for the spatial dimensions (not time). The result is a set of intersection pairs (s, c) where s and c are segments of two different trajectories. The result set needs to be filtered to return only trajectories that were within ε of one another at the same point in time.

Once the initial collisions have been found, future collisions caused by database repairs can be found very efficiently, by querying only for segments that were changed by a repair.

4.8.5 Evaluation

The experimental evaluation presented in this section was conducted using a desktop computer on an Intel i7-870 CPU at 2.93 GHz and 8GB of RAM.

4.8.5.1 Collision Detection

Figure 4.8a shows the total time required to find all initial collisions, which requires a large number of intersection queries. The number of collisions is also influenced by the minimal object distance ϵ, and Figure 4.8a illustrates the effect on the Endomondo data set.

Surprisingly, the time required to find collisions seems independent of ϵ. This is attributed to the fact that even for a large ϵ, the number of collision candidates that have to be evaluated is too small to significantly impact the run-time. Thus, the vast majority of time is lost in the collision candidate generation step.

Figure 4.8b shows the time required to repair the found collisions. In each iteration of each algorithm, three steps are required:

1. Repairing a collision,
2. updating the index with the new distorted trajectory, and
3. finding new collisions involving the distorted trajectory.

The times required for these three steps are shown in Figure 4.8b. We note that despite the use of an efficient index structure, the time needed to repair two colliding trajectories lasts only a fraction of the time needed to find the collision and update the trajectory.

4.8.5.2 Algorithm

In this section we take a closer look at the three algorithms Random, Greedy, and Simulated Annealing and how they work with the three different repair strategies. The results are shown in Figure 4.9.

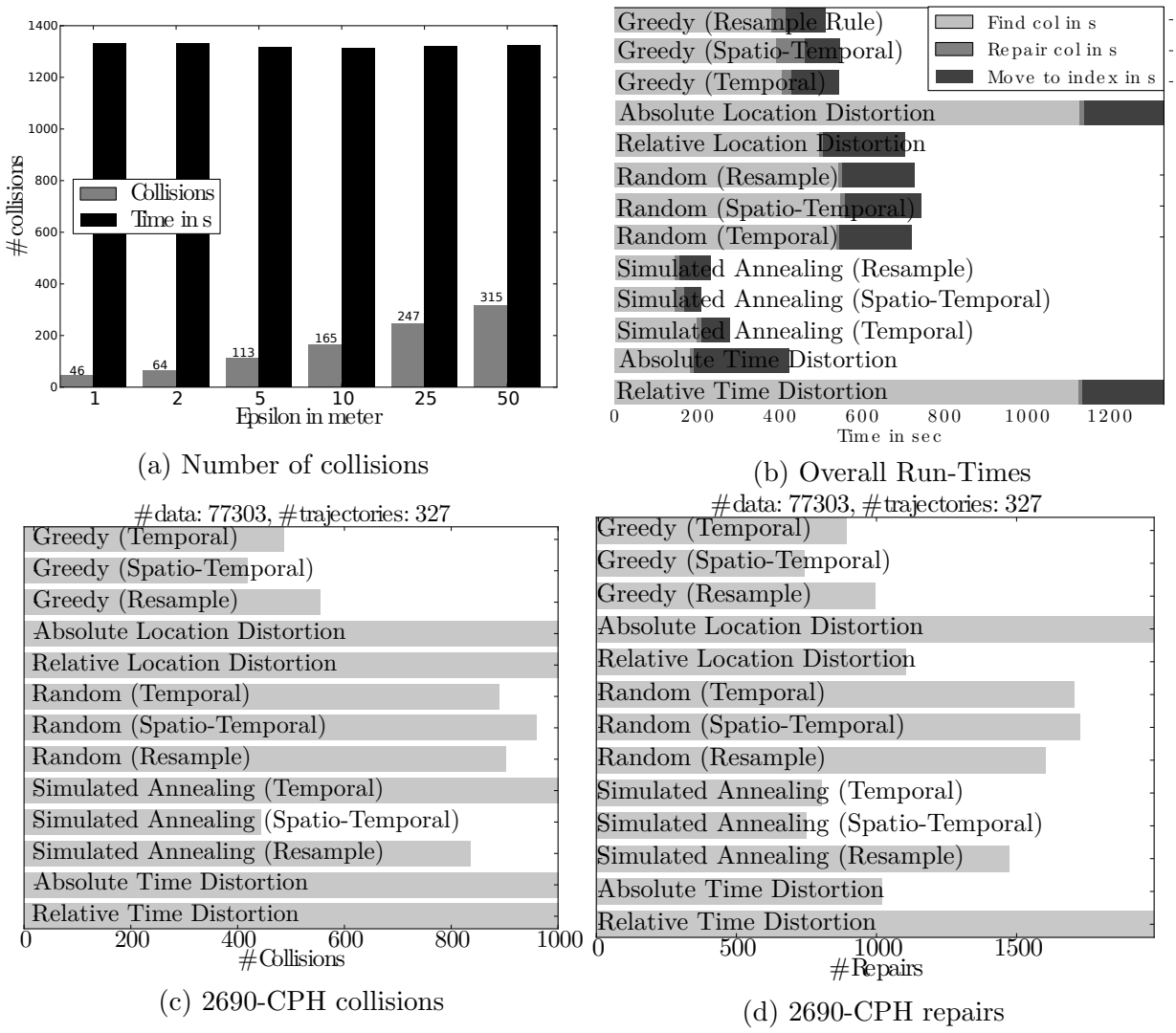
As expected, the number of collisions created (as a result of fixing others) by applying the spatio-temporal repair rules was smaller than that of either the Time-Shift Rules or the Resample Rules.

The time needed by Simulated Annealing is much smaller that that needed by other algorithms, caused mostly by the time needed for the search step. The performance of the search step can be seen in Figure 4.9c, which uses data generated over several different algorithms to avoid bias. Random always decides fastest (by not considering any alternatives), followed by Simulated Annealing, and finally Greedy.

The Resample Rules have the highest rate of termination. While these rules are relative (for which there are known termination problems), none were observed in any of the experiments.

4.8.5.3 Repair Strategy

Time-Distortion Repair This set of experiments investigates the influence of the parameter Δt, which affects how large the changes applied by time distortion repair rules were. In this set of experiments, some data sets could not be repaired with any of the algorithms under the usual limit of 1000 applied fixes. In these cases the number of maximal



(a) Number of collisions

(b) Overall Run-Times

(c) 2690-CPH collisions

(d) 2690-CPH repairs

Figure 4.8: Run time Experiments

iterations was raised. For example, in the experiments presented in Figure 4.10a, some experiments were stopped after 1000 iterations for $\Delta t = 10$ and $\Delta t = 2000$.

For this data set $\Delta t = 10$ is too small to repair the collisions and $\Delta t = 2000$ is too big, causing other collisions near the detected one. Values 100 and 1000 performed much better.

Location-Distortion Repair The Location-Distortion Repairs effect of the absolute location change Δd, using the Closest-Point Distance is shown in Figure 4.11.

Δd should be larger than 2ϵ to ensure that the worst case of (pieces of) trajectories being identical can be repaired. Only the parameters $\Delta d = 5$ and $\epsilon = 2$ allowed the algorithm to remove all inconsistencies from the data set.

Algorithm	Time to repair	# Repairs	t/#rep
Greedy (Temporal)	16.294	342	0.047643
Greedy (Spatio-Temporal)	51.181	330	0.155094
Greedy (Resample)	10.522	429	0.024527
Absolute location distortion	0.198*	341	0.000581
Relative location distortion	20.92*	1341	0.015600
Random (Temporal)	0.557	545	0.001022
Random (Spatio-Temporal)	0.503	519	0.000969
Random (Resample)	0.898	684	0.001313
Simulated Annealing (Temporal)	16.046	343	0.046781
Simulated Annealing (Spatio-Temporal)	47.673	332	0.143593
Simulated Annealing (Resample)	11.708	464	0.025233
Absolute time distortion	18.02*	5506	0.003273
Relative time distortion	17.823*	1340	0.013301

Table 4.1: Run time of all algorithms

4.8.5.4 Run time

Run times of all tested configurations are shown in Table 4.1. In Table 4.1, asterisks indicate that in at least one case, the repair algorithm did not terminate. Non-terminating cases are ignored for the computation of run-times in this experiments. Purely time distorting heuristics and purely location distorting heuristics are able to repair a database quickly. However, due to the simple rules that these approaches follow, they are unable to handle some of the more complicated cases which may occur in trajectory databases. For relative repairs two trajectory segments, which completely fall into each other's ϵ-range, cannot be repaired. For absolute repairs, non-moving trajectories are shifted, but the likelihood of reaching a state where all signals are collision free becomes minimal. When omitting the cases where these approaches do not terminate, the fastest repair is achieved by absolute location distortion heuristics. Furthermore, we can see that among the algorithms, Random performs best. This is expected, because Random does not need to evaluate expensive trials. Greedy and Simulated Annealing require approximately the same time as each other to apply their repairs, but require significantly more time than the Random approach. The run time of Simulated Annealing and Greedy increases sub-linearly in the number of repair rules. Since Random does not evaluate more than one rule, it is not affected by the magnitude of choices. Greedy requires a smaller number of total repair iterations to fix the database, resulting in a lower total run time.

4.8.5.5 Magnitude of changes

Section 4.7.3 introduced three dissimilarity functions. The results of the experiments are shown in Figure 4.12a. The Euclidean and Maximum dissimilarity functions almost always return the same values, as in most cases, individual trajectories are modified at only one

segment. The influence of the time domain for the weighted Euclidean Distance was set to 0.5 to compensate for its larger range of values.

The results of this evaluation are hard to interpret, because not all combinations were able to repair the hardest scenarios. Particularly the purely time or location distorting approaches tend not to terminate. Considering only terminating cases, time distorting and location distorting heuristics yield very good results.

The resulting dissimilarity of many approaches decreases significantly as the number of possible repair rules increases. In particular, the Spatio-Temporal Rules (the approach that allows repairing collisions by distorting space in one of eight directions or by distorting time in one of two directions) achieves an extremely small dissimilarity. Comparing the three heuristics to choose a repair rule, the random heuristic performs worst, because it contains a large number of needless distortions. The Greedy heuristic and the Simulated Annealing heuristic show comparable results. In fact, the Simulated Annealing approach yields smaller dissimilarity in some cases. This is possible, as the Greedy approach only selects the locally best next repair rule, which does not necessarily contribute to the globally best repair. In contrast, Simulated Annealing initially uses the Random strategy often and can quickly remove the majority of collisions. This seems to be a good compromise.

To summarize, the baseline repair rules using only spatial distortion and using only time distortion are not able to repair complex inconsistencies. Nevertheless, these approaches are easily implemented and have low run-times, such that these approaches might find applications in cases where a few remaining inconsistencies can be tolerated. The Random heuristic is able to achieve the fastest run-time, but incurs a repair-error that may not be tolerable in practice. The Greedy approach has the worst run-time, which is attributed to the fact that all possible repair rules are tested for each iteration. The Simulated Annealing approach appears to be a good compromise, achieving a final dissimilarity comparable to that of the Greedy approach, while being much faster. Furthermore, we saw a trade-off between run-time and dissimilarity relative to the number of repair rules: a larger number of repair rules leads to a (sub-linear) increase in run-times but also much smaller dissimilarity. Choosing the right repair rules is highly domain specific, depending on the types of inconsistencies that are to be repaired, and depending on the acceptable run time.

4.9 Conclusion

In this chapter applying constraints to route problems was discussed. In an initial step we saw how predicate constraints can be defined on route data. To suit application scenarios we extended the general framework to two specialized domains.

4.9.1 Continuous Cost Constraints

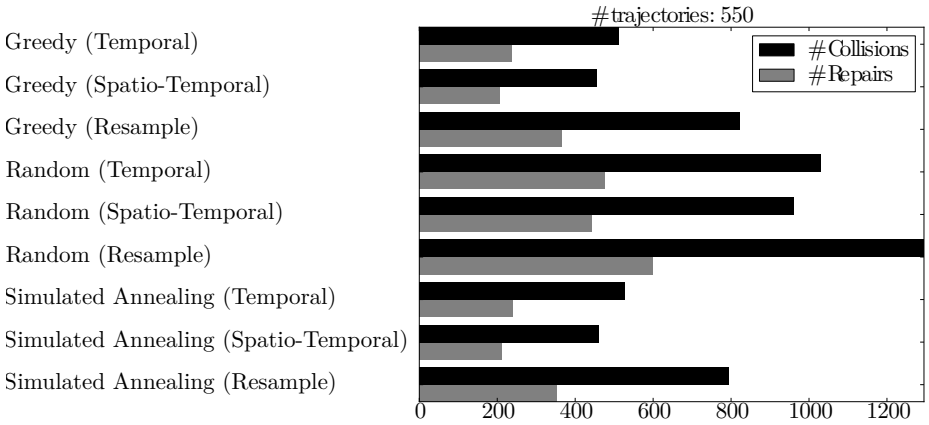
In an extension of the general problem, route constraints were extended to support general cost functions instead of predicate constraints. Applying this extended approach to the

sites of the FOR 1670 data set yielded routes connecting these sites and mapping a route for crossing the Alps in prehistoric times.

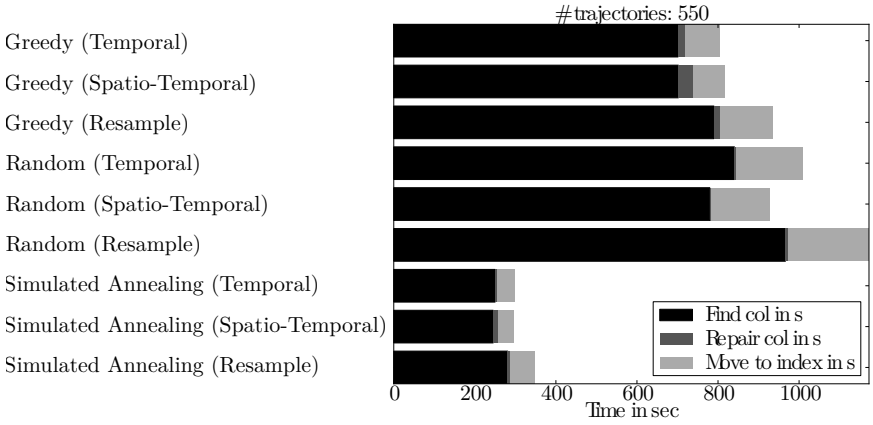
4.9.2 Spatio-Temporal Inter-Object Constraints

A further extension concerned a spatio-temporal databases and inter-object constraints. A type of problem involving these extensions are moving objects. Applying the introduced technique (and its extension) to this type of data allows addressing a category of problems that has been largely neglected in moving object literature: repairing inconsistencies in trajectory databases. This is an important problem since moving object databases are inherently uncertain for a number of reasons and, in addition, attempt to capture continuous phenomena via discrete values. We saw that the problem of repairing inter-object constraints is NP-hard. To solve it anyway, the presented approaches used heuristics to find good (rather than optimal) repairs. For this purpose, we presented a number of initial solutions, including a time-distortion algorithm, a space-distortion algorithm, as well as a set of generic algorithms that apply pre-defined repair rules, including a random algorithm, a greedy algorithm and a simulated annealing algorithm.

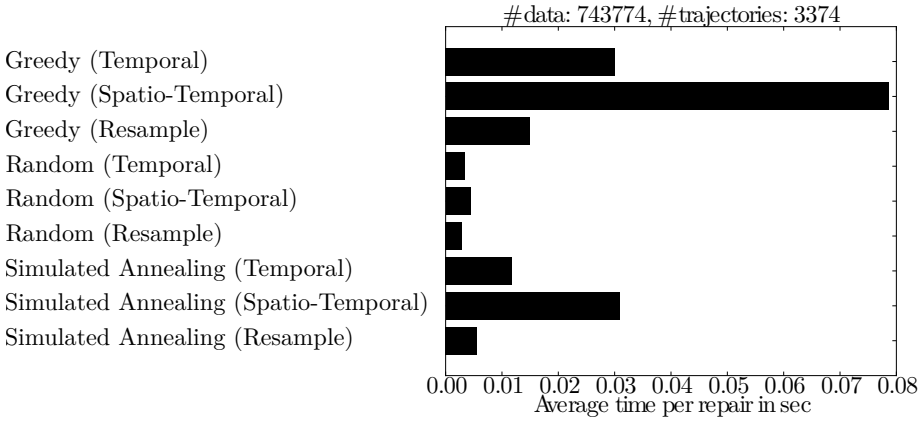
In the experimental evaluation of the presented approaches, we saw an application to object collisions, which are well-defined inter-object constraints. To generate a data set containing collisions, we used a real-world data set of workout trajectories and started them all simultaneously resulting in a large number of collisions. The results of the evaluation show that overly simple approaches fail to find any repair at all. In contrast, the proposed repair-rule based solutions are able to find a good repairs in acceptable time.



(a) 2646-CPH: Collisions and repairs

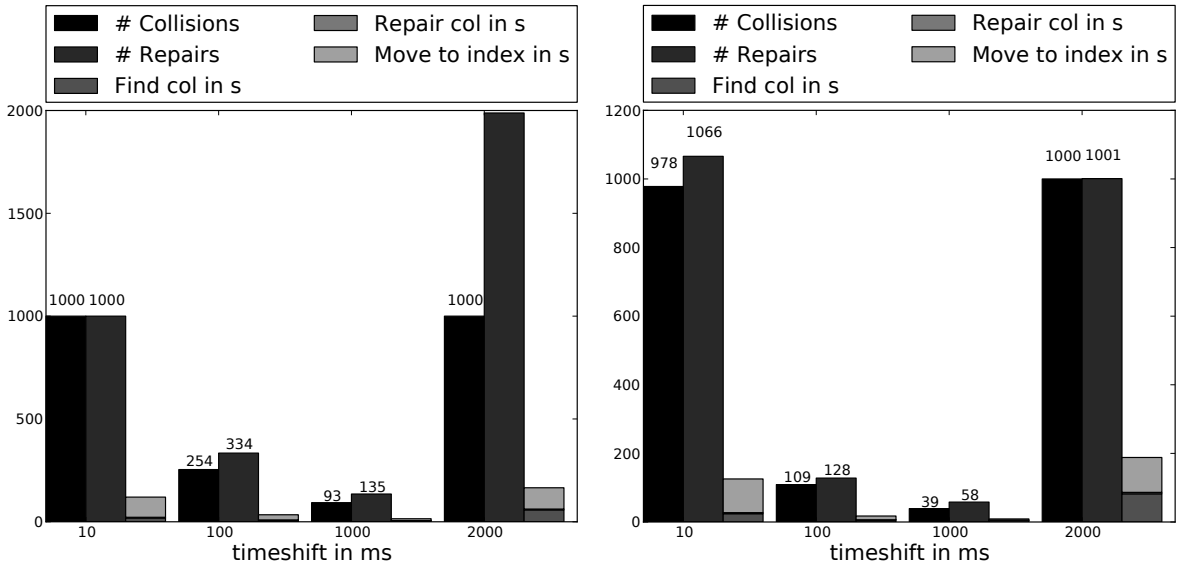


(b) 2646-CPH: Time



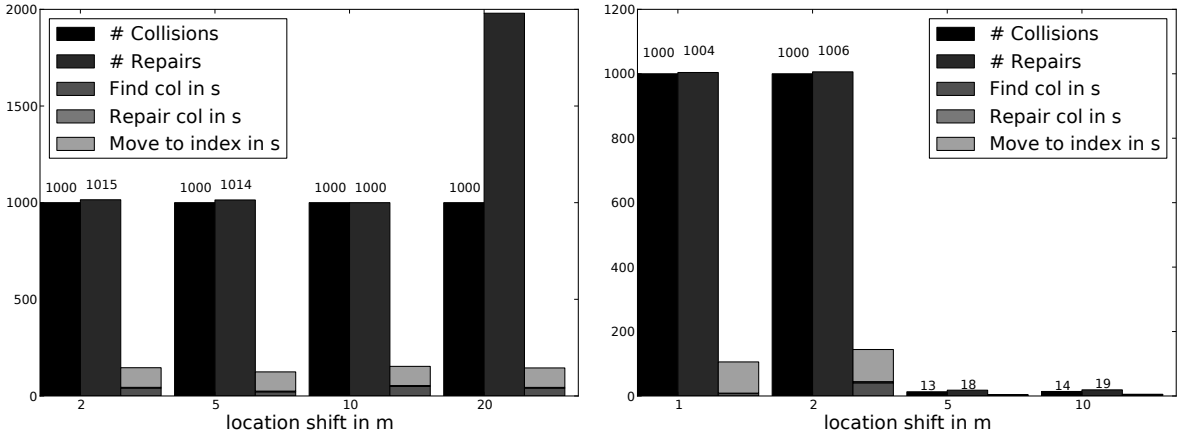
(c) Time to repair for multiple data sets

Figure 4.9: Output of Repair Constraints



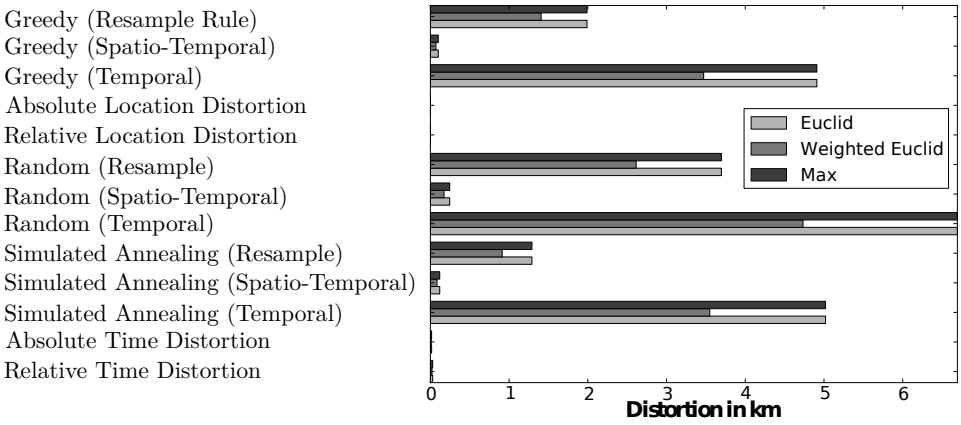
(a) $\epsilon = 10$ (b) $\epsilon = 2$

Figure 4.10: Using different Δt to repair same data set

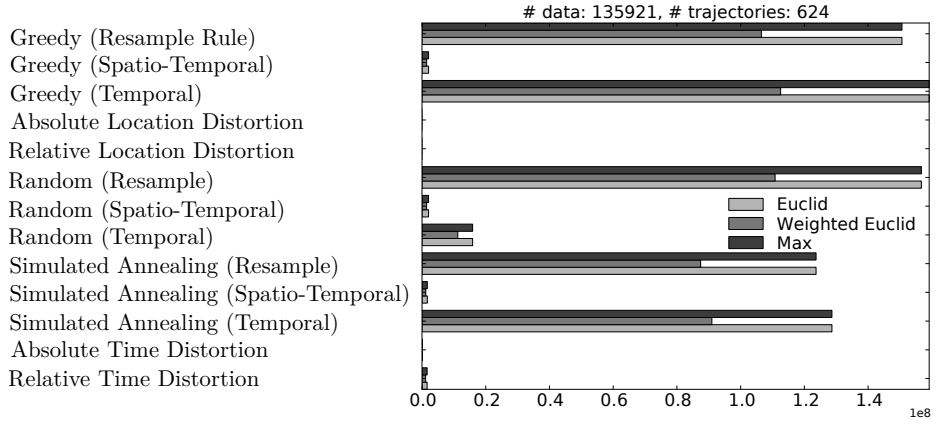


(a) $\epsilon = 10$ (b) $\epsilon = 2$

Figure 4.11: Using different Δt to repair same data set



(a) 2640-CPH



(b) 2692-CPH

Figure 4.12: Dissimilarity functions and quality of repair

Chapter 5

Spatially-Constrained Gaussian Mixture Models

Attribution

This chapter uses material from the following publications:

- M. Mauder, E. Ntoutsi, P. Kröger, and G. Grupe. Data mining for isotopic mapping of bioarchaeological finds in a central European Alpine passage. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*, page 34. ACM, 2015
- M. Mauder, E. Ntoutsi, P. Kröger, C. Mayr, G. Grupe, A. Toncala, and S. Hölzl. Applying data mining methods for the analysis of stable isotope data in bioarchaeology. In *2016 IEEE 12th International Conference on eScience*, 2016
- M. Mauder, Y. Bobkova, and E. Ntoutsi. GMMbuilder – user-driven discovery of clustering structure for bioarchaeology. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 8–11. Springer, 2016

See Section 1.3 for a detailed overview of incorporated publications.

In this chapter we concentrate on a particularly frequent and easy to follow type of constraint: the desire to find models of spatially localized regions. The considered data are spatially distributed measurements, which are represented by a set of feature vectors and for which no explicit groundtruth is available. Instead, an expected characteristic of the data is that there exists a connection between the resulting components and the spatial domain. This is plausible, because the signal we observe in the feature space is produced by

spatially located phenomena. According to Tobler's first law of geography [77], spatially close observations are more similar in their features than spatially distant observations. This implies that a spatially coherent model has a higher probability of being correct than one whose spatial projection is not correlated with the feature model. Where this information about spatial origin of a measurement is available, this information can be used to improve the model by preferring models that yield spatially coherent model.

The constraint data passed to all the presented methods are the spatial locations of the analyzed samples. Each methods uses this data as a different constraint to suit their particular approach. They all have in common that their output is a Gaussian Mixture Model, which represents the structure in the data in a natural way suiting many real data sets. The constraint data is used to inform the modeling process such that spatially close samples are more likely to have a high membership probability for the same component as one another. The constraint being addressed with these approaches is spatial coherence, i.e. the property that data points grouped in the model should come from a spatially similar location as well. Another way to imagine this property is as a spatial projection of a model where only points that have a high likelihood to belong to a given component are shown. We would expect the points belonging to one component to be in spatial proximity with each other. Spatially coherent models use information from a different domain to reveal something about the structure of a data set that is lost in a purely data based model.

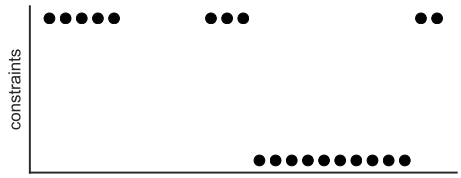
An optimistic approach is to assume that this connection will emerge from a purely measurement based model. If we are less optimistic, this effect can be achieved if modeling is guided by the spatial information to ascertain a coherent spatial projection, while at the same time being exclusively based on the measurements.

It is important to reiterate that while the goal of the discussed models is to be spatially coherent, the models do not represent any spatial information. The spatial information is only used to guide the modeling of the measurement data to a result that preserves the spatial structure, but does so based exclusively on the data. In other words, the model will be based only on feature data that can be extracted without knowledge of the spatial information, but the spatial information is supplied to the classifier in order to find a feature model that is closer to the latent model. Practically speaking, the resulting model can be applied without knowledge of the spatial domain, but reflects the spatial structure in the feature information.

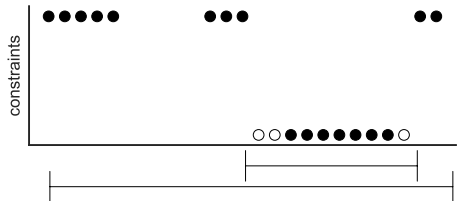
Spatial coherence is not automatically a usable constraint in any spatial data. On the one hand the assumption that spatially close points have similar measurements is not a necessity. On the other hand, insufficiently representative sampling can yield data that is beyond useful analysis. If the sampling is so far apart that the data has changed too much, measurements from even a very clean underlying distribution can result in very distinct values that cannot be reasonably modeled to something resembling the latent model. Since we have no information about the development of features between regions, whether spatial coherence holds cannot be tested within the proposed framework. We must trust domain scientists to choose this approach only if it is reasonable to assume that the data set is spatially coherent.

If a constraint compliant model has been found, the model represents both the data and

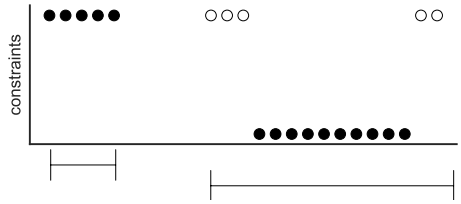
the constraints well. Instead of picking the most likely model according to the measurements alone, the algorithm can prefer "worse" solutions that tease apart some of the finer details of spatial structure, which may not be visible from the data alone. The data may suggest a different structure, but a similarly good model may exist that reflects another (unknown) structure better. This can be understood as a way to avoid overfitting of the data to the feature domain. An optimization based modeling approach may encounter a local minimum (corresponding to a locally ideal model in feature space), but miss a possibly better model nearby. By including additional information about the underlying structure, the resulting model can be nudged towards the better global result. Particularly in applications where data sets are small, being able to use the comparatively cheap (meta-) information of spatial origin to improve the finer points of the model is very desirable.



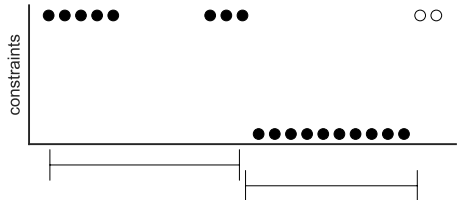
(a) Example data set. This data set illustrates a few common problems of spatial data sets. Left is a sampling problem: a region that is clearly part of the "upper" cluster is only partially sampled. Right is a data set problem: the "upper" feature component is not continuous. Although it is sampled continuously, its spatial distribution is discontinuous.



(b) Model based on spatial classification. This model cannot consider the feature distribution to pick a good feature model. One cluster spans the full data set to capture the outliers top right. This is incorrect and we prefer a model that misses the outliers. The inclusion of the outliers increases the classification accuracy to 85% (34:6), but might be considered overfitted.



(c) Feature based model and full dimensional model. This model includes points that are from a different location in the model. This decreases its accuracy to 75% (30:10).



(d) Spatially coherent feature model. While not all points are associated with their spatial solution, the missed points are clearly distinct in their feature value. Including them with the model would decrease the model quality. At 90%, the accuracy is not perfect, but the model misclassifies only likely outliers.

Figure 5.1: Example data set and models based on different paradigms.

5.1 Properties of Spatially Distributed Samples

A common property of spatial data collection is preferential sampling, i.e. the sampling is driven by motivations other than what samples are most useful for the following data mining task. For example, the data set introduced in Section 1.2.2 was sampled in archaeologically interesting locations, driven by the availability of samples, not in a more systematic (e.g. grid-like) fashion that would be beneficial to building a map. However, modeling a sample of points based on spatial distribution will identify dense spatial regions, which reflect the sampling process, not the latent distribution of features in the sampled region.

Trying to reflect spatial distribution through models over the data can be problematic if several components have very similar models, but different spatial locations. Modeling in the data domain will identify them as a single region. Inversely, two similar, but distinct, components in the same location will be explained as different sets, but as the same set by a spatial model.

These considerations are not exclusive to spatial constraints, but apply to other kinds of constraints on the distribution as well. It is desirable to build models of data that comply with a notion of similarity that is not captured by the model. Some common properties of spatial sampling and spatial data make constraints particularly appropriate for spatial modeling problems.

Figure 5.1 shows a schematic data set which illustrates these shortcomings. For presentation purposes, the example consists only of a single feature dimension (to base the model on) and an additional dimension representing the two spatial locations of the samples. Figure 5.1a shows the data without a model. Two clusters are present, clearly distinguishable by their y-coordinate (e.g. their spatial location). In addition there is a set of points that is part of the same cluster in constraint space, but closer to the other cluster in the data dimension. The shortcomings of a non-constrained approach to generating feature models are illustrated in Figures 5.1b and 5.1c. They show two output models (over the data domain), which were generated by minimizing the internal distance of samples based on different subsets of input data. Given only the data (x-axis), subsets of the clusters share the same range and cannot be differentiated. Given only the constraint data (y-axis), the resulting model will subsume too large ranges of the feature dimension. However, a model can be built to separate the spatial clusters in the data domain, because the constraint information allows separation of the groups that can be translated into data ranges. By considering the constraint information, we can derive a single model operating only on the data to separate the data according to the constraints. Figure 5.1d shows a possible solution represented only by a model of the data, but (applied to the training data) preserving the constraint information as much as possible.

The objective of the presented approaches is to build a model describing the data while representing the spatial distribution as well as possible. The model is a GMM over the data, ensuring that all properties of the data being determined by the approaches are based on properties that the resulting model can actually represent.

The rest of this chapter is structured as follows: Section 5.2 presents the example

application based on the data set introduced in Section 1.2.2. In Section 5.3 an overview of the diverse range of related work is given. Section 5.4 introduces four algorithms, which tackle the problem in different ways. Section 5.4.1 presents an interactive tool, which allows domain experts to generate Gaussian Mixture Models that comply with their domain knowledge without having to formalize it as constraints. Section 5.4.2 shows a simple, yet effective method to find constrained solutions using the example of spatial coherence and GMMs. Section 5.4.3 introduces a more scalable approach in the shape of a modified EM algorithm used to build constrained Gaussian Mixture Models through an efficient optimization-based approach. Section 5.4.4 builds upon Section 5.4.3 to sketch an idea for a related approach, which places fewer restrictions on the type of constraints that it can employ. After the algorithms have been introduced, Section 5.5 presents experimental evaluations including the real data set that was introduced in Section 1.2.2. Section 5.6 concludes the chapter.

5.2 Motivation: Predicting Places of Origin Based on Features

Chapter 1 described the objective of research group FOR 1670 as "the construction of a large scale isotopic map of the reference region, the Inn-Eisack-Adige transect via the Brenner pass in the European Alps". In this chapter we will attempt to build such a map. The desired maps are (according to two of the project's primary investigators [26]) "spatially and temporally defined stable isotopic patterns in geological and ecological settings [and] indispensable tracers for the monitoring of the flow of matter through geo/ecological systems." The intended application of this map is to either predict the origin of a new sample or to describe the distribution of the data so that domain scientists can use this information to generate new hypotheses. Both of these tasks require a model of the data and spatial information connected to it. The difficulty with modeling the spatial distribution of the data is that the sampling is highly selective. There are only few sites where samples are available and many samples share the same location.

The underlying data are stable isotope measurements of samples from archaeological sites in the Alps region. *Stable isotopes* are indispensable markers for the monitoring of the flow of matter through biogeochemical systems. Isotopes are atoms of the same element that have the same number of protons and electrons, but differ in the number of neutrons. Isotopes are generated, e.g., by the decay of parent isotopes, or by reactions with subatomic particles in the environment. For example, the three stable isotopes of oxygen are ^{16}O, ^{17}O, and ^{18}O. All of these have 8 protons and 8 electrons, but range from 8 (^{16}O) to 10 neutrons (^{18}O). An isotope is called "stable" if it does not spontaneously decay into another isotope. Oxygen atoms with fewer (e.g. ^{15}O) or more (e.g. ^{19}O) neutrons are unstable and will eventually decay into other stable isotopes. Differences in the number of neutrons results in different atomic masses and lead to differences in molecular bond strength and vibration energies. This, and the different thermodynamic reactivity of light and heavy

isotopes leads to isotopic fractionation (i.e., uneven partitioning of isotopes between source and product). Isotopic fractionation and mixing in an ecosystem generate compartments with characteristic isotopic signatures. For example, evaporation and condensation in the course of hydrological processes lead to predictable distributions of oxygen isotopes in the atmosphere and in precipitation. Isotopic labels, which are shared by certain ecological components such as soil, water, plants, microbia, and animals, have been successfully used for the generation of *isotopic maps* or *isoscapes* for the investigation of landscape ecology. Such isotopic maps representing the common, local isotopic signatures (or fingerprints), can later be applied to distinguish local and non-local finds: a local outlier, i.e., a sample found at location l that has an isotopic fingerprint different to the local fingerprint of l according to the map, is interpreted as non-local. If the isotopic fingerprint of the non-local sample matches the isotopic fingerprint of another location o, it is likely that o is its place of origin. Both the knowledge of outliers and their potential places of origin is very valuable for answering research questions in biology. Another example of a successful application of isotopic fingerprints is predicting the place of origin of ivory samples, potentially classifying this sample as illegally harvested [92, 83, 85].

Isotopic maps are empirically generated by sampling the relevant environmental components and by measuring their isotopic signatures. Since the geological processes on Earth differ considerably within surprisingly small regions, the surface of the planet can be divided into many small catchment areas with a distinctive, characteristic geological isotope fingerprint. These samples are used to define catchment areas featuring a homogeneous, characteristic isotopic fingerprint. In bioarchaeology, such samples are human and animal remains found in archaeological sites. However, due to intricate biological and chemical processes, these samples do not directly reflect the geological characteristics. Examples of such processes include metabolic differences between species and individuals (some of the inter-species differences can be reduced by applying empirically determined formulas), aging, integration over various environmental conditions, weathering of bones, metabolization, etc. Thus the geological characteristics of a region are only one of a few factors contributing to the measured isotope ratios. Since we cannot know the details of the metabolism that crucially influence the isotopic composition of an organism, the only realistic way of modeling the distribution of isotopes in animals found in a region is by building a model based on the measurements associated with them.

The resulting description of a region can be used for *provenance analysis*, i.e. the task of determining a sample's origin based on its measurements. The prediction of the origin of an individual relies on the connection between its isotope measurements and the environment where it was located in its lifetime. Isotope values in the organism of a given animal are the products of the metabolism of the animal. As such, the values are based on the food digested during the organism's life which in turn depends on geological processes of the environment. The relationship between the isotopic ratios of the surface of a catchment area (geological isotopic fingerprint) and within an organism is not a trivial relationship because the metabolism of organisms typically has a significant effect on the ratios of isotopes. However, this relationship enables researchers to draw conclusions about the place of origin of individuals like humans, animals or even plants from isotopic fingerprints

obtained from organic remains. Despite the fact that the resulting values may not be applicable as a model to a single sample, which is still subject to individual variability as outlined above, the aggregated values from the model building represent a much more reliable model. As a consequence, while the local isotopic fingerprints provided by isotopic maps will never be as reliable as the term fingerprint may suggest, they can be applied subject to a careful probabilistic interpretation.

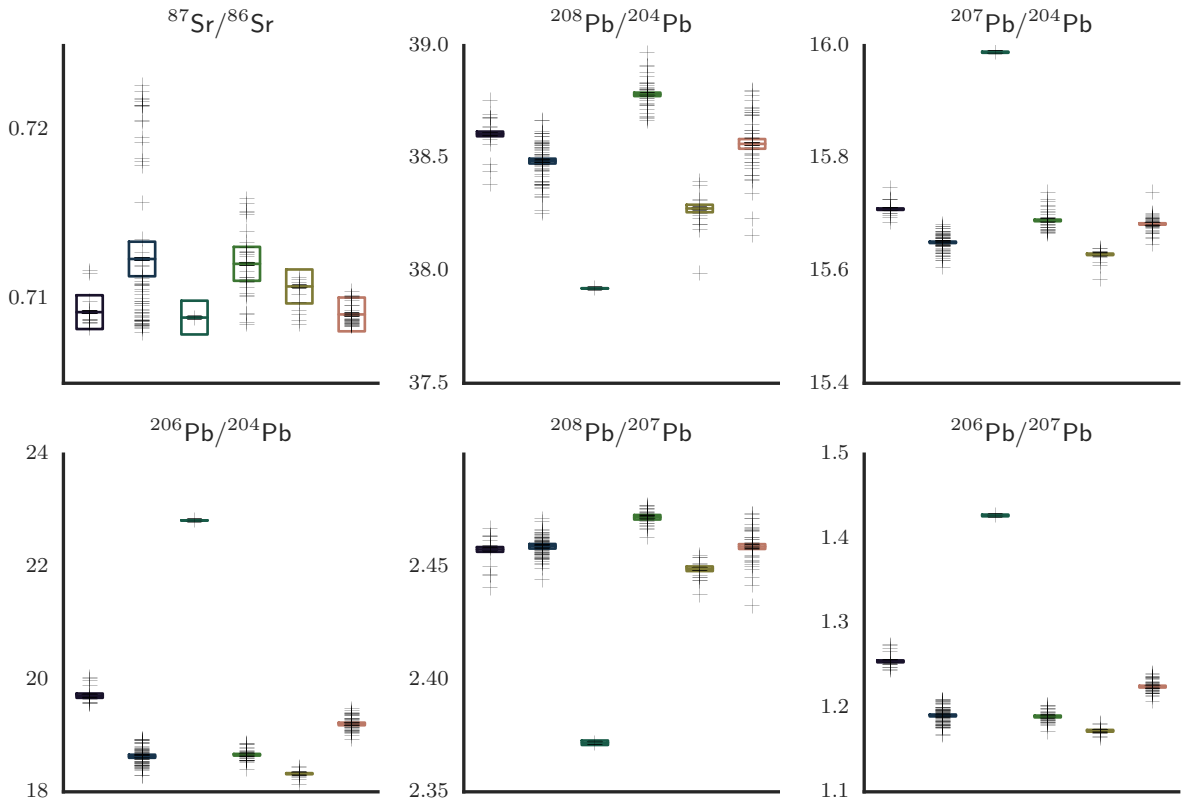
Historically, stable isotopes in bio-archaeological finds were measured and simply compared to the known spatial distribution of the isotopic system under study such as $^{87}Sr/^{86}Sr$ in geological maps, or the climate and habitat dependent distribution of C_3- and C_4-plants which is reflected in the ^{13}C-values of the consumers' tissues. For that purpose, the isotopic measures of the samples were typically plotted (annotated with spatial information) and grouped visually. Obviously, this procedure limits the application to 1D or 2D data, i.e. at most two isotopic systems can be evaluated simultaneously. Outliers, detectable by conservative statistics (e.g. [27]), were readily interpreted as immigrant individuals. Very often, this was simply done by measuring one specific isotopic system, e.g. $\delta^{18}O$ from phosphate in bones, and then manually determining local models and outliers in the univariate plots of the resulting values. In most applications, however, this model is too simple because the mechanisms that researchers are studying can only be captured when considering several isotopic systems [55]. Rather, techniques for multi-dimensional data analysis are necessary. This makes the standard visual analysis approaches not applicable any more. Growing insights into small-scale variabilities in isotopically characterized ecogeographical compartments gave rise to more fruitful discussions on mobility versus migration and trade in the past

In data mining terms, isotopic mapping aims at finding spatially coherent components representing individuals with homogeneous isotopic features. The spatial extent of such a component represents a catchment area and the features of the individuals represent the characteristic fingerprint of this area. Based on this characteristic fingerprint, samples can be classified as either local (if its isotopic finger print matches the fingerprint of the catchment area where it has been found/obtained) or non-local.

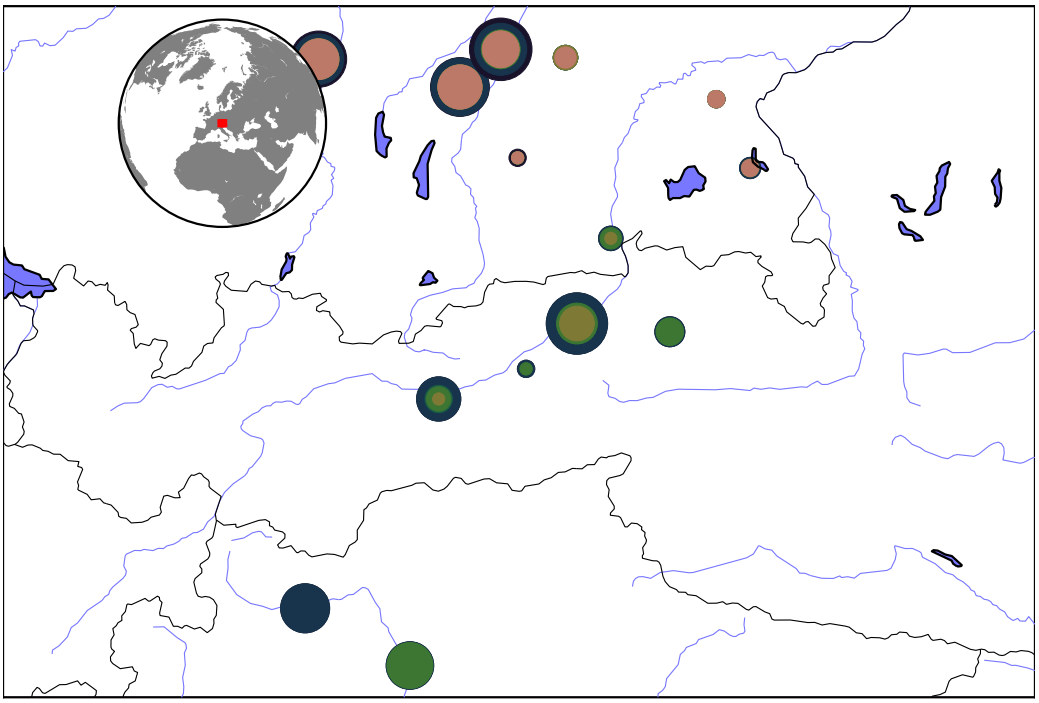
5.2.1 Provenance in the Alps Region

In this chapter, techniques will be introduced that can be used to build isotopic maps. Modeling and understanding spatial isotopic variation is complicated by the small number of available samples, potential mobility of the investigated samples, sample preservation quality, uncertainty of measurements, and so forth.

The task is to employ a data-driven approach to define the catchment areas and derive a characteristic isotopic fingerprint from these areas that can be used as predictive model. This enables archaeologists to evaluate the probabilities that a given finding originates from each of the catchment areas. In particular, it gives the domain expert a probability that a given sample is local or has moved to the place where it has been found. The objective of these maps is first and foremost to explain the data distribution reliably. Only



(a) Distribution of components by attribute.



(b) EM result map.

Figure 5.2: EM result. Best of 500 runs according to spatial silhouette score.

based on a good description of the distribution is it even possible to reason about spatial distribution. For this purpose we are building models of the data distribution and employ spatial information only to guide the modeling such that the resulting models are also spatially coherent.

Figure 5.2 shows an example of a possible description. The depicted model is the result of a EM-based Gaussian Mixture Model built from the human samples in the FOR 1670 data set. Figure 5.2a shows the model's mean and variance per attribute and component. Figure 5.2b shows the spatial projection of the maximum likelihood assigned label. Section 5.5 explains the details of how the model and map were generated.

5.3 Related Work: Spatial GMMs

This chapter addresses constraints in spatial modeling. Constraints have been mentioned several times in this text, most prominently in Section 2.1. In this section we will discuss existing methods for spatial modeling, the constraint "spatial coherence" in the context of partitioning clustering, and existing adaptations of the Expectation-Maximization algorithm.

5.3.1 Spatial Modeling

There is a range of statistical methods for modeling spatial data. The most prominent way to reason over spatial models is through *kriging* [48], predicting values at unobserved locations from known observations. A commonly used structure for this purpose are *Gaussian Markov Random Fields* (GMRF), which are Gaussian distributions which only depend on their immediate neighbors. Fitting Markov Random Fields and Gaussian Markov Random Fields is commonly achieved using *Markov Chain Monte Carlo* (MCMC) methods. A popular new technique to build these models are *Integrated Nested Laplace Approximations* (INLA) introduced by Rue et al. [68].

Contrary to the method described here, these methods do not explain the data distribution, but concentrate on predicting the most plausible value. In this chapter we aim to explain the observed values by fitting the component distributions of a *Gaussian Mixture Model* (GMM). GMMs do have the property of fitting a soundly defined distribution, but in their original form they ignore any spatial data.

5.3.2 Gaussian Random Field Mixture Models

The presented techniques extract Gaussian Mixture Models, which comply with spatial constraints, but do not model the spatial distribution explicitly.

Explicitly modeled spatial distribution of Gaussian Data is modeled by *Gaussian Random Fields* [74, 46], multi-dimensional Gaussian processes. They are particularly well suited to noise reduction, but are not particularly capable of dealing with discontinuities.

GMMs with an explicit spatial distribution could be described as multivariate *Gaussian Random Field Mixture Models* (GRFMM). Bolin et al. [9] proposed a mathematically founded theory of GRFMMs in 2014. They lay out the mathematical properties of GRFMMs and propose a method of fitting GRFMMs using a stochastic gradient algorithm.

5.3.3 Applications of the EM Algorithm

The EM algorithm, which two of the algorithms introduced in this chapter are based on, is a very flexible paradigm, which has been used in many domains to estimate many different statistical models. Two famous examples are the Baum-Welch algorithm for the estimation of Hidden Markov Models and the Inside-Outside algorithm for the estimation of probabilistic context free grammars for natural language processing.

The Baum-Welch algorithm [5] estimates the maximum likelihood of hidden Markov models (HMM) using the EM paradigm [7]. To estimate the probability of an observation after a given time, the Baum-Welch algorithm fits a hidden Markov model from a series of observations. The model consists of the initial probability of starting in each state, the probability of transitioning from any given state to any state (time independent because of the Markovian property), and the probability of making any given observation if the model is in any given state. The probability of being in a given state at a given time based on the current parameters and the probability of transitioning from a given state to any given state after a given time can be calculated from the current model state. These two estimations constitute the expectation step. The probabilities for all possible combinations are sufficient to update the model (maximization step).

The *inside-outside algorithm* [6] uses the EM paradigm to estimate *probabilistic context free grammars* (PCFG) in natural language processing [44]. The expectation step of the inside-outside algorithm estimates how well a PCFG explains a set of input sentences. The parameters of the PCFG are then re-estimated based on these probabilities.

The regularized EM algorithm by Li et al. [45] is similar to the presented approach in that it modifies the GMM probability function in EM to achieve further goals. Its goal is to reduce the uncertainty of missing data by introducing a regularizer that prefers a model that is in agreement with the data. Similarly to the approach presented here, they compare models to make the algorithm choose a higher quality solution according to a different model. However, compared to our approach the authors use a very different model that is based on the estimation of the hidden data, not additional constraint data.

5.3.4 Spatial Coherence

Spatial coherence is a common concern in image segmentation where a set of pixels is partitioned according to their color information. In this task it is acceptable to subsume spatial outliers into a surrounding cluster if the spatial coherence is thus retained. Of particular interest in this area is the exact spatial position of cluster borders. Additionally, the spatial connectivity is a much more rigid condition than in our proposed approach. Many methods from image segmentation approach the problem as an optimization problem

and use e.g., Graph-Cuts, to find an approximate solution. In *Spatially Coherent Clustering Using Graph Cuts* [89], Zabih and Kolmogorov propose a partitioning cluster algorithm, which minimizes an energy function of a set of clusters and a labeling of pixels in an image, penalizing both spatially incoherent clusters and clusters which fit the data badly without picking a particular clustering method. Its major differences to our proposed algorithm are that the clustering is partitioning and that the spatial coherence notion of pixels in images is comparatively limited.

Another class of algorithms applies clustering to determine areas associated with characteristic measurements. One such example is *Regionalization of forest pattern metrics for the continental United States using contiguity constrained clustering and partitioning* [43]. In this paper Kupfer et al. generate a map of forest patterns using a hierarchical clustering method (*REDCAP* [28]), which optimizes a homogeneity measure. This approach generates a partitioning clustering, in which the spatial information is used to "fix" an incorrect assignment. This is in strong contrast to our model-based clustering approach in which the cluster assignment is governed exclusively by the model (including allowing single points to deviate from their spatial surroundings), not by a one time assignment of a label.

5.4 Approaches

The task of building Gaussian Mixture Models that are spatially coherent is a special case of more general class of constraints. The original inspiration for this work was the interdisciplinary research project described in Section 1.2.1 where spatial coherence was particularly important. In this section, we will see four different approaches to building Gaussian Mixture Models over data while respecting constraints as much as possible. The first approach (see Section 5.4.1) is an interactive tool, which allows domain experts to build models from sets of reliably grouped points while incorporating any and all constraints that they see in the manifestation of the continuously updated model. Following that, Section 5.4.2 introduces a Monte Carlo approach to the problem. Many constraint compliant models are produced and evaluated, picking the best model in the process. The presented approach uses heuristics to reduce the number of models to evaluate (limiting the type of constraints to spatial coherence in the process). To find a solution efficiently, Section 5.4.3 shows an approach based on the famous Expectation Maximization algorithm. This approach finds solutions quickly by optimizing a random initial model. To encourage spatial coherence in the resulting model, it modifies the terms used in the expectation and maximization steps to optimize towards a spatially coherent and data fitting solution based on an auxiliary GMM over constraint data. This approach is efficient, but limits the usable constraint information to point data. Section 5.4.4 introduces an approach which requires only pairwise distances for each sample point. Since the resulting equations are not analytically solvable, this approach uses a heuristic to perform the maximization step.

5.4.1 Interactive Gaussian Mixture Model Building GMMbuilder

Interdisciplinary research has the potential to speed up scientific discovery and improve many areas of knowledge. However, varying backgrounds of participating researchers also make effective communication harder. This section presents GMMbuilder, a tool that allows domain scientists to build Gaussian Mixture Models (GMM) that adhere to domain specific constraints. This tool can improve cooperation between domain scientists and data scientists on the task of data modeling or even allow domain scientists to build sensible models on their own. GMMbuilder works with univariate or multivariate, continuous, or discrete data. If the data contains attributes that map to spatial coordinates, the tool supports spatial projections of the data and analysis results. Domain experts can use the tool to generate a set of input models, extract stable object communities across these models, and use these communities to interactively design a final model that explains the data but also considers constraints that are implicit in prior beliefs and expectations of the domain experts.

The model is built by identifying strong object communities in the data and incorporating the models of these communities into the final clustering model. To derive strongly connected components in the data it relies on unsupervised learning. In particular, it generates multiple clusterings from the data and finds object formations that are stable across many clusterings.

The intuition behind using communities is that similar objects that get grouped together consistently across the different clusterings have a higher chance to also be grouped in the resulting GMM. Community members have shown a strong tendency to form groups with each other over a range of clusterings and are more likely to represent a cluster in any final model-based clustering. Their robustness towards splitting suggests that the underlying model of each component GMM represented them in some consistent way. Building a Gaussian model over these points should reproduce these models without any interference from the rest of the data set.

The domain expert has a very active role in the process: from the selection of the clusterings from which the communities will be extracted to the selection of the communities that will form the basis for the final clustering model. Figure 5.3 depicts the GMMbuilder architecture, consisting of several modules that will be presented hereafter. As the figure shows, the role of the domain expert is vital.

Input The data mining step in the KDD process produces different results depending on the input data and the model parameters and algorithm parameters. A different setup will usually result in different GMMs and selecting the best setup is not straightforward. The selection of the input data D for the GMM, the selection of different subsets of data points or attributes, and varying input parameters for EM, e.g., k, the number of components. In a first step, the user selects all inputs and parameters that could plausibly produce a good model.

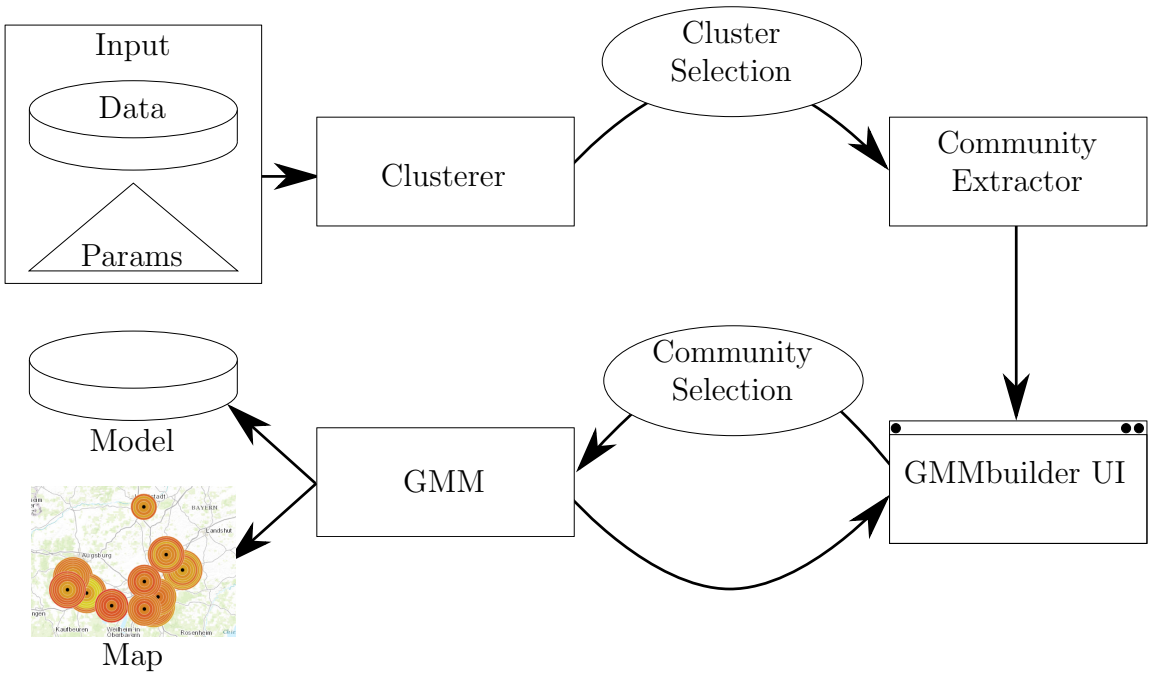


Figure 5.3: An overview of *GMMbuilder*. Oval shapes depict user interaction.

Clusterer The Clusterer module derives a clustering over a given data set D. A GMM is used as the basis for the clustering as a GMM is also the intended output of the tool. To extract the GMM, the tool applies the Expectation Maximization (EM) algorithm [17], which fits k multi-variate normal distributions over the given data set subject to the different parameters (including k) chosen in the previous step.

Cluster Selection By running the EM algorithm on different combinations of the mentioned factors different GMMs are generated. The tool relies on the domain experts to decide which of the generated models are acceptable. The decision is based on the user's expertise supported by a range of information about the generated models. The tool provides a detailed clustering description, in terms of the spatial projection of the cluster members and the distribution of the isotope values in each cluster. The result of this step is a set of user-accepted clusterings $\mathcal{C}$.

Community Extractor By examining the different clusterings, objects that are frequently assigned to the same cluster together (*communities*) are identified. They are selected by combining results of various clusterings and selecting points that are frequently assigned to the same clusters together. These communities can be examined and – if they seem promising – selected for the final model. More formally, a stable community c consists of a set of points $p \in D$ that are clustered into the same cluster across multiple clusterings $C_i \in \mathcal{C}$:

$$c(C_1, C_2, \ldots, C_n) = \{p \mid p \in C_{1,i} \wedge p \in C_{2,j} \wedge \ldots \wedge p \in C_{n,m}\} \ ,$$

where $C_{i,j}$ is the set of points in the jth cluster in clustering C_i. Stronger communities span more clusterings compared to weaker ones and indicate high similarity between their objects. The extracted communities are used as the basis of a Gaussian Model that becomes part of the final GMM.

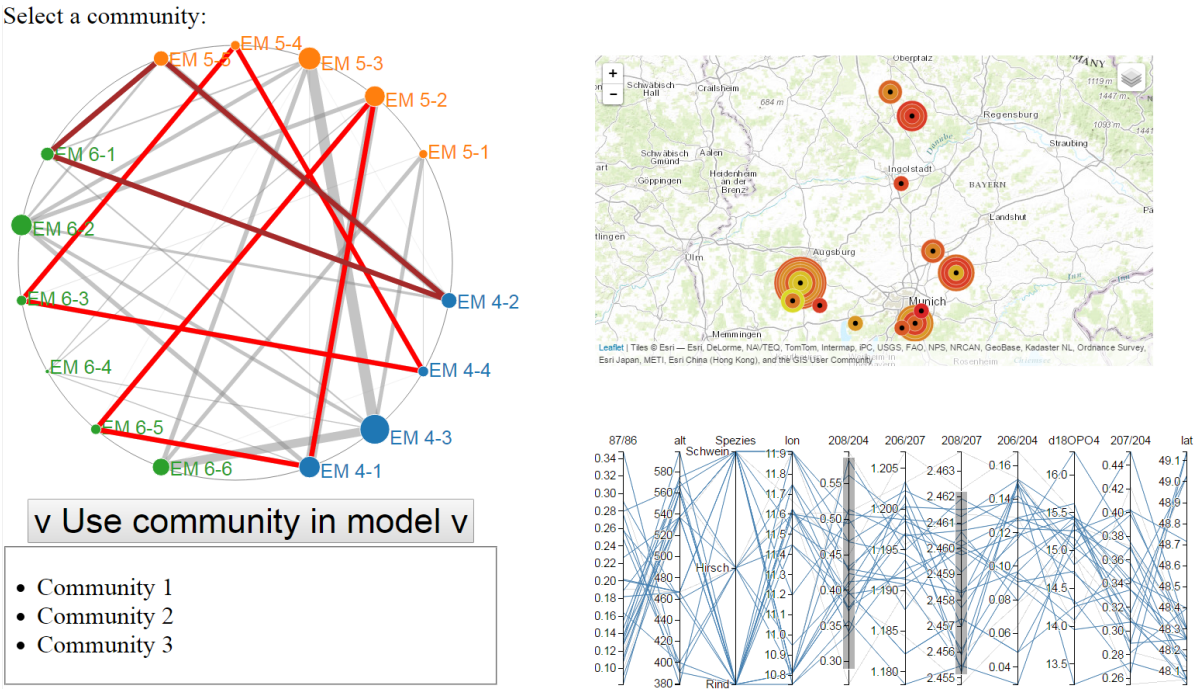


Figure 5.4: GMMbuilder: Interactive GMM building - inspecting one of the communities found in all three clusterings (orange, green and blue).

GMMbuilder UI To support the user in choosing appropriate communities, GMMbuilder presents a visual representation of communities and their participating clusterings (Figure 5.4). Each cluster is represented as a node on a circular projection of clusterings. Each clustering is represented by adjacent nodes. Communities are represented by edges between cluster nodes. Each node can be selected to show which communities it participates in. The edges representing communities can be selected to choose a community to investigate more closely. The circular projection of clustering information visually depicts how cluster members change between clusterings and allows comparing more than two clusterings. This allows the user to spot interesting communities quickly and get an idea of their robustness.

Community Selection When the user picks a community, a description of the data is being displayed: parallel coordinates of the attribute values in the community give an indication of what is characteristic of the community. If the used data has a spatial component, a spatial visualization of cluster members allows the expert to establish whether a cluster is spatially constrained and if any outliers are suspect.

Based on these visualizations, the user can pick communities to use in building a new GMM. If a community extracted in the previous step does not agree with domain experts' prior beliefs and expectations, they will reject it. The tool relies again on the domain expert to decide which of the detected stable communities should inform the final model generation. This decision should be informed by the domain experts knowledge, i.e. express the implicit constraints they know apply in this setting.

GMM When the expert selects a community c to evaluate, a Gaussian model of its objects is extracted and displayed. When a community's model is added to the GMM, the model's effect is re-evaluated to assess the data fit. The new GMM is used to re-evaluate the membership probability of each data point in the data set D and a new clustering is created based on c's model. The user can directly inspect the results and decide whether it is a good or bad model for the final clustering. After adding the component based on the current community to the model and converting it to be represented as a Gaussian model, its members are re-evaluated in light of the other components of the GMM. The probability density of each point and the maximum likelihood cluster assignment of the final model are shown. Trivially, the first community to be added will initially have the highest density for all points in the data set. Further communities may gain or loose members, depending on other present models. After the expert has chosen to add the community to the mixture model, it too will be considered when future communities' membership are being evaluated. The user can then select another community c' to evaluate.

This is an iterative process. When the user adds communities, they can directly inspect the effect on the final clustering. The process stops when the user is satisfied with the results. The output of this step is a set of user-accepted communities from which Gaussian models have been extracted.

The resulting model contains no explicit knowledge of the constraints that were considered in its building. It requires some experimentation to generate a model that fits the constraints sufficiently to be acceptable to the domain scientist. Once this has been achieved, the resulting model is a Gaussian mixture model of the data that represents the domain scientist's constraints as much as possible.

This approach relies heavily on user input. That makes it very flexible, allowing a user to express any kind of constraints, even if the constraints are nothing more than a conviction that the result does not "feel right". While this allows for quick and satisfying results, the resulting models are less rigorous and require heavy human intervention. In the following section, we will see a first attempt at a fully automatic approach to constrained data modeling.

5.4.2 Monte Carlo

This section will describe a relatively simple approach to build a model that is both spatially coherent and based only on the input data. The constraints being applied is that the points represented by each component can at most have a given diameter. This constraint is a

global predicate based on the constraint data. The constraint data is the spatial coordinates associated with the input data.

The approach we use to find the best model, which is compliant with a constraint, is to list all possible configurations $\mathcal{P}$ that comply with the constraint and pick the one that has otherwise the best performance according to some score s:

$$P^* = \underset{P \in \mathcal{P}}{\operatorname{argmax}}\, s(\mathcal{X}|P)$$

However, due to the large search space $\mathcal{P}$ an exhaustive search is almost never feasible. Instead we draw samples from $\mathcal{P}$ and find the maximum over these samples in acceptable time.

In order to further increase the chance of finding a good fit, it is important to only evaluate models that comply with the constraint. In the special case of spatial coherence and GMMs, a possible way to randomly generate a spatially coherent partitioning of the data $\mathcal{X}$ is based on a pre-computed reachability graph $\mathcal{R}$. Reachability is a necessary property of spatial coherence that can be pre-computed and reduces the search space. The reachability graph is defined as

$$\mathcal{R} = \{(p, q) \mid d(p, q) < \epsilon\} \ ,$$

where $d(p, q)$ is the spatial distance between data points $p, q \in \mathcal{X}$. A reachability relation $s \rightarrow_Q q$ over $\mathcal{R}$ such that

$$s \rightarrow_Q q \Leftrightarrow \exists q_1, \ldots, q_n \in Q : (s, q_1) \in \mathcal{R} \wedge (q_1, q_2) \in \mathcal{R} \wedge \ldots \wedge (q_n, p) \in \mathcal{R}$$

can be used to generate a spatially coherent partitioning. We define the set $\mathcal{S}$ of all spatially-coherent subsets $S \subset \mathcal{X}$:

$$\mathcal{S} = \{S \subset \mathcal{X} \mid \forall p, q \in S : p \rightarrow_S q \wedge d(p, q) < \alpha\epsilon\}$$

as the set of all reachably connected subsets below a maximum spatial diameter $\alpha\epsilon$. Based on this set, the set $\mathcal{P}$ of all spatially-coherent partitionings P into k components is

$$\mathcal{P} \subseteq \mathcal{S} : \forall P \in \mathcal{P} : |P| = k \wedge \bigcup P = \mathcal{X}$$

This yields a labeling for all points in the data set. This labeling's feature model can be evaluated using any score $s(\mathcal{L}, \mathcal{X})$ of the features given the labeling. See Section 5.5.4 for an application of this technique to the archaeological data set introduced in Section 1.2.

In the following section we will look at an alternative solution to the spatial coherence problem that uses optimization inside a modified Expectation Maximization Algorithm to find a (locally) optimal solution efficiently.

5.4.3 Constrained EM Algorithm

In this section we are looking at a direct approach to determine an optimal solution to building a constrained GMM. Based on the EM algorithm we are designing an algorithm that incorporates constraints into its optimization. The general idea is to measure if the constraint data about a good model are in line with the model's current state. The resulting constraints are how well the spatial data fits the model built over the data. This constraint is a global costs based on each points' constraint data.

Problem Specification Input of the spatial coherence algorithm are data points and associated constraint attributes in a distinct subspace. Result of the algorithm is a GMM over the data domain, which when applied to the training data yields clusters that reflect the same structure as the associated constraints. The proposed algorithm extends the EM clusterer. Like EM-GMM it outputs a set of Gaussian models over the feature data, whose mixture gives an output that explains the observed data. It behaves like any non-constrained GMM over the feature data. Any information about the relationship between data points is used to drive the modeling process, but not incorporated into and not necessary for the application of the resulting model. Its goal is to use constraints during the training phase to guide the clusterer towards a compliant model.

5.4.3.1 The Expectation-Maximization-Algorithm

Our proposed algorithm is based on the *Expectation-Maximization algorithm* by Dempster et al. [17]. For a high-level description of EM as a probabilistic clustering algorithm, see Han and Kamber [32]. EM proceeds by optimizing a set of parameters Θ that represent a model of the entire data $\mathcal{X} \subset \mathbb{R}^m$ until convergence is reached. Θ describes a (finite) Gaussian Mixture Model, made up of k components θ_j. Initially, the model is a guess based (typically) on some of the data $\mathcal{X} \subset \mathbb{R}^m$. A probability function $P(x^{(i)} \in c_j)$ gives the likelihood of point $x^{(i)}$ belonging to component distribution $\theta_j \in \Theta$.

The model is generated through iterative refinement of an initial estimate. The optimization procedure refines the model Θ by repeatedly

1. calculating the expected probability $P(x^{(i)} \in c_j)$ for each $x^{(i)} \in \mathcal{X}$ and $c_j \in C$ (*Expectation*) and

2. refining the model based on these probabilites, such that the likelihood of the component distributions are maximized by the new model $\tilde{\Theta}$ (*Maximization*).

This process is repeated until no (or a sufficiently small) improvement in the global likelihood is reached in one step. The process reaches some optimum fast, but the resulting model may not represent a global optimum.

In the following we give a more technical description of the EM algorithm to support our derivation of the modified terms below. The input data $\mathcal{X}$ is considered incomplete data, with the membership probabilities ϕ being the hidden data. EM's approach to estimate Θ

is to iteratively refine a prior estimate. The fitness of the current estimate Θ is determined by a likelihood function L (or more specifically its logarithm ℓ, which has an optimum at the same Θ). Instead of optimizing ℓ directly (which is numerically not possible), an auxiliary function $Q(\Theta, \phi)$, which is a lower bound to ℓ, is defined. When Θ are the model parameters and $\phi_j^{(i)}$ is the assigment of each point $x^{(i)}$ to component j, $Q(\Theta, \phi)$ is defined by

$$\begin{aligned} Q(\phi, \theta) &= \sum_i \sum_j \phi_j^{(i)} \log \frac{p(x^{(i)}|\theta_j)}{\phi_j^{(i)}} \\ &\leq \sum_i \log \sum_j \phi_j^{(i)} \frac{p(x^{(i)}|\theta_j)}{\phi_j^{(i)}} \\ &= \ell(\phi, \theta) \end{aligned} \tag{5.1}$$

The inequality in Equation 5.1 is Jensen's inequality. Due to the monotonicity of the logarithm, the maximum of $Q(\phi, \theta)$ is at equality with $\ell(\phi, \theta)$, i.e. maximizing $Q(\phi, \Theta)$ for ϕ gives $\ell(\phi)$. In the opposite direction, maximizing $Q(\phi, \Theta)$ for Θ gives a set of parameters maximizing ℓ. Additionally, when $Q(\phi, \Theta)$ is maximized over the parameters Θ, its value grows, i.e. $Q(\phi, \tilde{\Theta}) \geq Q(\phi, \Theta)$.

The EM algorithm uses these properties of the auxiliary function to estimate the current likelihood ϕ and use this estimate to improve the current parameter estimate Θ. The first step is to estimate the best assigment to each cluster given the current model (Expectation). Given a set of parameters θ, EM finds a local optimum for the parameters θ using the best fit assigment ϕ. Jensen's inequality becomes equal, when the term inside the function is a constant, i.e. the bound is tight ($\ell(\phi, \theta) = Q(\phi, \theta)$) at exactly the point where

$$\phi_j^{(i)} \frac{p(x^{(i)} \mid \theta_j)}{\phi_j^{(i)}} = const$$

Since we know that ϕ is a distribution, we know that

$$\sum_j \phi_j^{(i)} = 1\,.$$

To ensure this property, we must normalize against the cummulated ps and get

$$\phi_j^{(i)} = \frac{p(x^{(i)} \mid \theta_j)}{\sum_j p(x^{(i)} \mid \theta_j)} \tag{5.2}$$

because log is a strictly convex function ($f''(log(x)) = x^{-2} = 1/x^2 > 0$) and thus $E(f(x)) = f(E(x)) \leftrightarrow X = E(x)$.

In turn, to estimate a new set of parameters $\tilde{\theta}$ based on these assigments (Maximization), the optimum with regard to the parameters θ can be calculated. The auxiliary function Q is partially derived with respect to each parameter, which when set zero yields their value at the optimum.

EM GMM To calculate the model parameters Θ requires specifying the probability function p, so we will now look at the classic application of the EM algorithm: Gaussian Mixture Models. The GMM-EM algorithm specializes the EM paradigm to the case of Gaussian Mixture Models. The parameters θ of a GMM are the mean μ and covariance matrix Σ.

Given these parameters, the probability density of a single component in a GMM is

$$p(x^{(i)}|\mu_j, \Sigma_j) = \frac{1}{(2\pi)^{n/2} \left|\Sigma_j\right|^{1/2}} \exp\left(-\frac{1}{2}(x^{(i)} - \mu_j)^T \Sigma_j^{-1}(x^{(i)} - \mu_j)\right). \tag{5.3}$$

Since maximization is specified using the full term p, the maximization term takes the same form as previously introduced. This probability can be directly used to calculate the new expectation $\tilde{\phi}$ given a model $\theta_j = (\mu_j, \Sigma_j)$:

$$\tilde{\phi}_j^{(i)} = \frac{p_j(x^{(i)} \mid \mu_j, \Sigma_j)}{\sum_{j=1}^{k} p_j(x^{(i)}|\mu_j, \Sigma_j)}$$

to make the log-likelihood function equal to the expected likelihood. To get the values of μ and Σ at the current optimum, Q is maximized with regard to them. This yields new estimates

$$\tilde{\mu}_k = \frac{\sum_{i=1}^{n} \phi_k^{(i)} x^{(i)}}{\sum_{i=1}^{n} \phi_k^{(i)}}$$

$$\tilde{\Sigma}_k = \frac{\sum_{i=1}^{n} \phi_k^{(i)} (x^{(i)} - \tilde{\mu}_k)(x^{(i)} - \tilde{\mu}_k)^T}{\sum_{i=1}^{n} \phi_k^{(i)}}.$$

Since this property holds for any model θ, this approach can be applied iteratively.

5.4.3.2 Constrained EM

The modification of the EM algorithm presented in this section is intended to incorporate domain constraints in the modeling process. This will be accomplished by letting the probability function incorporate additional information to estimate the contribution of an additional set of constraints. The goals of the presented approach are two-fold:

1. Build a model that can be expressed in the data domain $\mathbb{R}^m$.
2. Build a model that expresses the training set $\mathcal{X}$ in a manner consistent with a set of constraints $\mathcal{Y} \in \mathbb{C}$.

In order to be able to reach Goal 1, the original probability function p (cf. Equation 5.3) should emerge for suitable constraint values. For Goal 2, the probability function should be augmented to make solutions more likely that comply with the constraints $\mathcal{Y}$. To implement these design criteria, a new probability function will be constructed that uses a second

probability density function over $\mathbb{C}$ and combines it with the existing probabilty function to use their joint probability as the new $\hat{p}$. The joint probability $p(x^{(i)}|\Theta) \cdot \tilde{p}(y^{(i)}|\Theta^{\mathbb{C}})$ (where $y^{(i)}$ is the constraint space entity correponding to $x^{(i)}$) of two probability functions can be understood as a similarity of the points under the model Θ and $\Theta^{\mathbb{C}}$. A high agreement between the model domain and the constraint domain is characterized by the two probability functions p and $\tilde{p}$ agreeing.

Given a distribution ϕ the maximization can be computed on the constraint domain $\mathbb{C}$, yielding a model $\Theta^{\mathbb{C}}$ over that domain. The probability $p(y^{(i)} \mid \Theta^{\mathbb{C}})$ over that model can be computed in the same way as $p(x^{(i)} \mid \Theta)$ was. This will yield a additional probability to $p(x^{(i)} \mid \Theta)$. Making these probabilities correspond is the approach used in this section.

This approach assumes that the constraint data can be modeled by an EM paradigm as well. If the type of constraints being considered is metric data, another GMM is a possible choice. In the following, it is assumed that $\mathcal{Y}$ is modeled as a GMM. In Section 5.4.4 we discuss possible extensions to alleviate this limitation.

The joint probability over the two models Θ and $\Theta^{\mathbb{C}}$ is

$$\hat{p}(x^{(i)}, y^{(i)}|\Theta, \Theta^{\mathbb{C}}) = p(x^{(i)}|\Theta) \cdot p(y^{(i)}|\Theta^{\mathbb{C}})^{\alpha} .$$

The parameter α allows the influence of the constraint domain to be balanced with the data's. Using the joint probability over two models, results in the following likelihood function:

$$L(\Theta, \Theta^{\mathbb{C}}|\mathcal{X}, \mathcal{Y}) = \sum_i \log \sum_j \phi_j^{(i)} \frac{\hat{p}(x^{(i)}, y^{(i)}|\Theta)}{\phi_j^{(i)}} .$$

Expectation As in the previous description of EM based algorithms, the optimization is performed by re-estimating the contribution of a point to a given component and then adjusting the model accordingly. The best model given the data (including hidden data) derives analogously to EM-GMM from the auxiliary function

$$Q(\Theta, \Theta^{\mathbb{C}}) = \sum_i^N \sum_j^K \phi_j^{(i)} \log \frac{\hat{p}(x^{(i)}, y^{(i)}|\Theta, \Theta^{\mathbb{C}})}{\phi_j^{(i)}} . \tag{5.4a}$$

From Jensen's inequality we know that Q reaches a maximum at equality with ℓ (which is its upper bound), i.e. when

$$\sum_i^N \sum_j^K \phi_j^{(i)} \log \frac{\hat{p}(x^{(i)}, y^{(i)}|\Theta, \Theta^{\mathbb{C}})}{\phi_j^{(i)}} = \sum_i^N \log \sum_j^K \phi_j^{(i)} \frac{\hat{p}(x^{(i)}, y^{(i)}|\Theta, \Theta^{\mathbb{C}})}{\phi_j^{(i)}} . \tag{5.4b}$$

For a strictly convex function (such as log) this holds true when the term inside the expectation ($\frac{\hat{p}(x^{(i)}, y^{(i)}|\Theta, \Theta^{\mathbb{C}})}{\phi_j^{(i)}}$) is constant. Additionally we know that $\phi^{(i)}$ is a distribution, i.e. $\sum_j \phi^{(i)} = 1$. These conditions are satisfied by

$$\phi_l^{(i)} = \frac{\hat{p}(x^{(i)}, y^{(i)}|\theta_l, \theta_l^{\mathbb{C}})}{\sum_j^K \hat{p}(x^{(i)}, y^{(i)}|\theta_j, \theta_j^{\mathbb{C}})} . \tag{5.4c}$$

Maximization Given the optimized distribution ϕ it is possible to optimize the remaining parameters Θ and $\Theta^{\mathbb{C}}$. Therefore we calculate the derivative of Q with respect to the parameter in question and set the resulting equation zero.

$$\begin{aligned}
&\sum\sum \phi_j^{(i)} \log \hat{p}(x^{(i)}, y^{(i)}|\Theta, \Theta^{\mathbb{C}}) - \log \phi_j^{(i)} \frac{\partial}{\partial \Theta} \\
&= \sum\sum \phi_j^{(i)} \log \hat{p}(x^{(i)}, y^{(i)}|\Theta, \Theta^{\mathbb{C}}) \frac{\partial}{\partial \Theta} \\
&= \sum_j \sum_i \phi_l^{(i)} \log p(x^{(i)}|\Theta) + \alpha \sum_j \sum_i \phi_l^{(i)} \log p(y^{(i)}|\Theta^{\mathbb{C}}) \frac{\partial}{\partial \Theta} \\
&= 0
\end{aligned}$$

For $\mu \in \Theta$ (we can get $\tilde{\mu} \in \Theta^{\mathbb{C}}$ analogously using y for x) this becomes

$$\sum_i \sum_j \phi_j^{(i)} \log \frac{1}{(2\pi)^{n/2} \left|\Sigma_j\right|^{1/2}} \exp\left(-\frac{1}{2}(x^{(i)} - \mu_j)^T \Sigma_j^{-1} (x^{(i)} - \mu_j)\right) \frac{\partial}{\partial \mu_l} = 0$$

Splitting the logarithm and applying the derivative gets rid of the normalization.

$$\sum_i \sum_j \phi_j^{(i)} \left(-\frac{1}{2}(x^{(i)} - \mu_j)^T \Sigma_j^{-1} (x^{(i)} - \mu_j)\right) \frac{\partial}{\partial \mu_l} = 0$$

After specializing to μ_l and simplifying, we arrive at

$$\sum_{i=1}^{m} \phi_l^{(i)} \left(\Sigma_l^{-1} x^{(i)} - \Sigma_l^{-1} \mu_l\right) = 0$$

and thus

$$\mu_l = \frac{\sum_{i=1}^m \phi_l^{(i)} x^{(i)}}{\sum_{i=1}^m \phi_l^{(i)}} .$$

The derivation of Σ as well as $\tilde{\mu}$ and $\tilde{\Sigma}$ are analogous.

Convergence The convergence property of EM is not affected by the use of a different probability function. The likelihood (as defined by the probability $\hat{p}$) is reached by optimizing ϕ. The parameters of the likelihood function are then maximized (which also makes them monotonically grow). This means that the lower bound of the next iteration is higher (or equal) than the likelihood (upper bound) of the previous iteration. So the likelihood keeps growing until convergence with the (local) optimum value of ℓ.

5.4.3.3 Spatial EM

Above, we made the assumption that the constraint data can be modeled by a GMM as well. This necessary limitation of the presented technique makes it very easy to extend to spatial constraints. The introduction to this chapter introduced spatial coherence constraints. Like the distribution model itself, the spatial data follows a Gaussian Mixture Model. The model

is a description of the data in terms of a set of means $\mu \in \mathbb{C}^m$ and a set of corresponding covariance matrices $\Sigma \in \mathbb{C}^{m \times m}$, which follows the Gaussian distribution around a mean μ following the covariances specified by Σ. Recall that for our GMM, the parameter $\Theta^{\mathbb{C}}$ subsumed the parameters $\mu^{\mathbb{C}}$ (the spatial mean of each component distribution) and $\Sigma^{\mathbb{C}}$ (their covariance matrices).

Given a set of responsibilities $\phi_j^{(i)}$ of a point $x^{(i)}$ and a component given by θ_j under the current model, the maximization equation for $\Theta^{\mathbb{C}}$ can be used to estimate a spatial distribution model corresponding to the current state of Θ. The spatial information $\mathcal{Y}$ can then be interpreted as constraints, whose probability $p(\mathcal{Y} \mid \Theta^{\mathbb{C}})$ under the spatial model gives an indication of the general applicability of the model. Section 5.5.5 presents the application of this method to an example data set and the real world isotope data set.

5.4.4 Distance-Based Constrained EM

The constrained GMM approach presented so far relies on the distribution of the constraint data following a Gaussian Mixture Model. This assumption is analogue to assuming that the spatial sampling reflects the data distribution, which is commonly not the case. To alleviate this problem, a constrained version of EM will be shown here, which relies only on pairwise distances. But we will – unfortunately – also see the mathematical roadblocks this approach faces. Nevertheless possible heuristics for an approximate solution are given and the problems (and possible solutions) are discussed.

Contrary to above the constraints now are pairwise distances that apply locally. This has the advantage of comparing to the constraint data directly, not a model of it, which is influenced by many constraint data points. Most types of constraints can be translated into pairwise distances between points with straight-forward semantics: if two points are close to one another and share similar model probabilities, them being generated by the model is more likely.

We are addressing the problem of building a model of a multivariate random variable $\mathcal{X} \subset \mathbb{R}^m$. The idea behind this *generalized constrained EM algorithm* is to place less restrictions on the type of constraints that can be incorporating by accepting any similarity matrix $\mathcal{W} \in \mathbb{R}^{n \times n}$ as constraints. Specifically, the design criteria for a new probability function are:

- The resulting probability $\hat{p}$ should be the same as the one corresponding to the underlying model p, if constraints are uniform (no apparent structure).
- Weight the incorporated probabilities p and $\mathcal{W}$ to measure influence of joint probability (not any single probability disproportionately).

The input of the presented algorithm is a set of samples $\mathcal{X} \subset \mathbb{R}^m$ and a set of associated similarity values $\mathcal{W}$ for each pair from X. The objective is to generate a model that assigns a membership probability to x_i and x_j that reflects the similarity $\mathcal{W}$. The type of constraints we are considering here are given as pairwise similarities $\mathcal{W}$ over some domain. It is interesting to note that the likelihood of a random agreement of the constraint probability

with the model probability is much lower than the probability of a low agreement. It is therefore not as crucial that two points disagreeing is a negative influence as two agreeing should be a positive one.

The first component to specify is the constraint similarity matrix $\mathcal{W}$, which indicates how similar points $x^{(i)}$ and $x^{(j)}$ are according to domain constraints. $\mathcal{W}$ is more generic than the constraint probability function used in $\hat{p}$: Since $\mathcal{W}$ does not need to know about the responsibilities ϕ, it does not need to have distribution properties itself. Symmetry is a necessary property, but if it is not inherent, the resulting probability term will average the two directions into a single ("symmetrical") value.

The second component is the joint similarity function. Each point's membership to any component can be calculated using the usual probability function $p(x|\theta_j)$. The joint probability of two points $x^{(i)}$ and $x^{(j)}$ being generated by the same component θ_j is given by

$$p(x^{(i)}, x^{(j)}|\theta_k) = p(x^{(i)}|\theta_k) \cdot p(x^{(j)}|\theta_k)\,.$$

For each point, the similarity to each other point is investigated to determine whether the two assessments agree.
Combining the constraint similarity $\mathcal{W}$ and the joint probability p gives the combined model/constraint similarity for two points $x^{(i)}$ and $x^{(j)}$:

$$\hat{p}_k(x^{(i)}, x^{(j)}) = \mathcal{W}_{ij} \cdot p(x^{(i)}, x^{(j)}|\theta_k)$$

To get the model fit of a point $x^{(i)}$, we integrate over all points and get

$$\sum_j \hat{p}_k(x^{(i)}, x^{(j)})$$

In order to normalize out the original probabilities (and get only the effect of joining them), we normalize by their cumulative marginal probabilities.

$$\hat{p}_k(x^{(i)}) = \frac{\sum_j \mathcal{W}_{ij} \cdot p(x^{(i)}, x^{(j)}|\theta_k)}{\sum \mathcal{W}_{ij} \cdot \sum p(x^{(i)}, x^{(j)}|\theta_k)}$$

This results in large values for $\hat{p}$ when the investigated points fit the model with similar probabilities and are similar according to the constraints $\mathcal{W}$, too. This function is a measure of the x's fit of the constraints given by $\mathcal{W}$.

To be able to build the original model (required to evaluate p and the desired descriptive output), we combine the original probability function p, add the new function $\hat{p}$, and use a parameter α to allow the user to regulate the influence of each.

$$p^{\mathbb{C}}(x^{(i)}|\Theta) = p(x^{(i)}|\Theta) + \alpha\hat{p}(x^{(i)}|\Theta)\,.$$

Expectation As we have seen in Equation 5.2, calculating the expectation of the model is straight forward. The likelihood reaches a maximum given the current model θ at

$$\phi_j^{(i)} = \frac{\hat{p}(x^{(i)} \mid \theta_j)}{\sum_j \hat{p}(x^{(i)} \mid \theta_j)}\,.$$

Maximization The problem with maximizing this equation is that it is not possible to extract the PDF over the data from the term to maximize it. So we cannot find the parameter values, which make Q maximal.

When we start with the auxiliary function

$$Q(\theta, \phi) = \sum_i \sum_j \phi_j^{(i)} \log \frac{p^{\mathbb{C}}(x^{(i)}|\theta_j)}{\phi_j^{(i)}}$$

and expand the term as before to be try and calculate the partial derivative with respect to the model parameters, we get

$$Q(\theta, \phi) = \sum_i \sum_j \phi_j^{(i)} \left(\log p^{\mathbb{C}}(x^{(i)}|\theta_j) - \log \phi_j^{(i)} \right)$$

$$Q(\theta, \phi) = \sum_i \sum_j \phi_j^{(i)} \log \left(p(x^{(i)}|\Theta) + \alpha \cdot \mathcal{W}_{ij} p(x^{(i)}|\theta_k) p(x^{(j)}|\theta_k) \right) - \sum_i \sum_j \phi_j^{(i)} \log \phi_j^{(i)} .$$

The equation cannot be further derived, because the logarithm cannot carry over the addition. As a result it is impossible to calculate the logarithm of $p(x^{(j)})$ and to maximize that PDF over the data.

This method cannot be considered mathematically sound until a proper optimizable formalization is found. For the moment, we will use the work-around to omit the constraint term from auxiliary function before deriving the Maximization. This is equivalent to optimizing the original probability $p(x^{(i)}|\hat{\Theta}, \hat{\phi})$ over the new expectations instead of $\hat{p}$:

$$\sum_i \sum_j \phi_j^{(i)} \log p(x^{(i)}|\theta_j) \frac{\partial}{\partial \mu_l}$$

The resulting optimization function will therefore be the familiar ones:

$$\mu_k = \frac{\sum_{i=1}^n \phi_k^{(i)} x^{(i)}}{\sum_{i=1}^n \phi_k^{(i)}}$$

and

$$\Sigma_k = \frac{\sum_{i=1}^n \phi_k^{(i)} (x^{(i)} - \mu_k)(x^{(i)} - \mu_k)^T}{\sum_{i=1}^n \phi_k^{(i)}} .$$

5.4.4.1 Spatial generalized constrained EM

To apply the generalized constrained EM algorithm (despite its mathematically deficiencies), all we need is a similarity matrix $\mathcal{W}$. Since we assume a geo-spatial setting, we will use the Euclidean distance as the basis for the underlying similarity function:

$$d(x^{(i)}, x^{(j)}) = \sqrt{\sum_k (x_k^{(i)} - x_k^{(j)})^2}$$

To turn the pairwise distances into a similarity matrix, we need to make it so that the maximum similarity is the highest value.

$$\mathcal{W}_{ij} = 1 - \frac{d(x^{(i)}, x^{(j)})}{max_{ij}(d(x^{(i)}, x^{(j)}))}$$

In the following section we will apply the presented techniques to a real-world data set of spatially distributed measurements that should be spatially coherent.

5.5 Application

In the previous section we saw several approaches to building Gaussian Mixture Models that adhere to (either explicit or implicit) domain constraints. In this section these approaches will be applied to the data introduced in Section 1.2. The goal will be to find plausible descriptions of the isotope distributions measured in this data set that can be used by domain scientists to explain and draw conclusions from the these and further finds in the area.

As the discussed constraint, we will use *spatial coherence* (see the introduction to Chapter 5. Spatial coherence is the property that the additional information of spatial origin of a sample (while not part of the model), is indirectly represented in the model. Spatial coherence can be measured by applying the model to the spatial data and measure how spatially similar the points in a component are. The following section describes scores for measuring model fit. One measure (the *Silhouette score*) can be applied to measure the spatial coherence of a model.

5.5.1 Evaluation

Below each of the described methods will be applied to the real-world data set presented in Section 1.2.2. For each of them, the resulting model will be presented in two ways:

boxplots showing the distribution of each attribute in each component.
For each of the d features (isotopic ratios), the distributions of its values per component are depicted as box plots. Although only a single-attribute view of the components is provided by this figure, we can see that there is some variation in the values across the different components for all isotopes.

spatial projections showing the spatial projection of the model on a map.
Each sample is represented as a circle around the site where it was found. Colors indicate the components they were assigned to with the highest probability.

All presented algorithms are non-deterministic. In order to allow for a fair comparison, each (with the exception of the interactive GMMbuilder) was run 500 times and the best result was used. These results are also what was used in the comparative evaluation (Section 5.6.1).

The evaluation was based on the subset of isotope ratios that have been extracted from human samples. This was done to reduce the influence of effects other than spatial distribution from the data set, which contains many different influences anyway. Different species differ in their metabolism and behavior, resulting in different isotope ratios in their bones. The evaluation is based on human samples, because humans represent the largest group in the data set.

To compare the results, performance measures were calculated for each chosen result. The *silhouette coefficient* evaluates how close each point in one cluster is to points in the neighboring clusters. Its values lie in the [-1,+1] range with +1 indicating points that are very distant from neighboring clusters, 0 indicating points that are not distinctly in one cluster or another and -1 indicating points that are probably assigned to the wrong cluster. The Bayesian Information criterion *BIC* [72] and the Akaike information criterion *AIC* [2] are measures that assess the relative quality of models. In their case, smaller scores are better. The silhouette score relies on labels to assess model quality, which makes it not ideally suited to the probabilistic predictions of a GMM. However, it has an advantage over BIC and AIC in that it can be applied to a different domain than the one the model was trained on, i.e. it allows the application of the score to the constraint domain and compare the performance of the models over this domain as well. We will use this property below to examine the spatial coherence of a solution.

In order to allow a comparison of the models, they should all be based on the same parameters as much as possible.

5.5.2 Model Parameters

The following approaches build models of the data distribution while considering constraints. The first model (described in Section 5.5.3) is interactive and affords the user the maximum amount of freedom to choose a good model. The rest, however, all perform automatic model extraction. To be able to build a mixture model of the data, these approaches need to know the number of components k to fit to the data. This parameter is notoriously difficult to pick. In order to be able to compare the resulting models, they should share one choice of k. To be independent of any one implementation, we use the unmodified EM-GMM algorithm, initialize it with different values of k, and generate 100 models with each setting. The setting that achieved the best mean score is then picked.

Figure 5.5 shows the performance measures introduced in the previous section. On the presented experiments, AIC shows an optimum at $k = 6$, while the silhouette score and BIC prefer $k = 3$. However, $k = 3$ is not plausible given the task, so where a parameter k is required, the experiments will be performed at $k = 6$.

5.5.3 Interactive Gaussian Mixture Model Building: GMMbuilder

Section 5.4.1 introduced the tool *GMMbuilder*, which allows domain scientists to interactively build Gaussian Mixture Models that adhere to domain specific constraints. The user is presented with a number of "stable" subsets of points (*communities*), which are derived

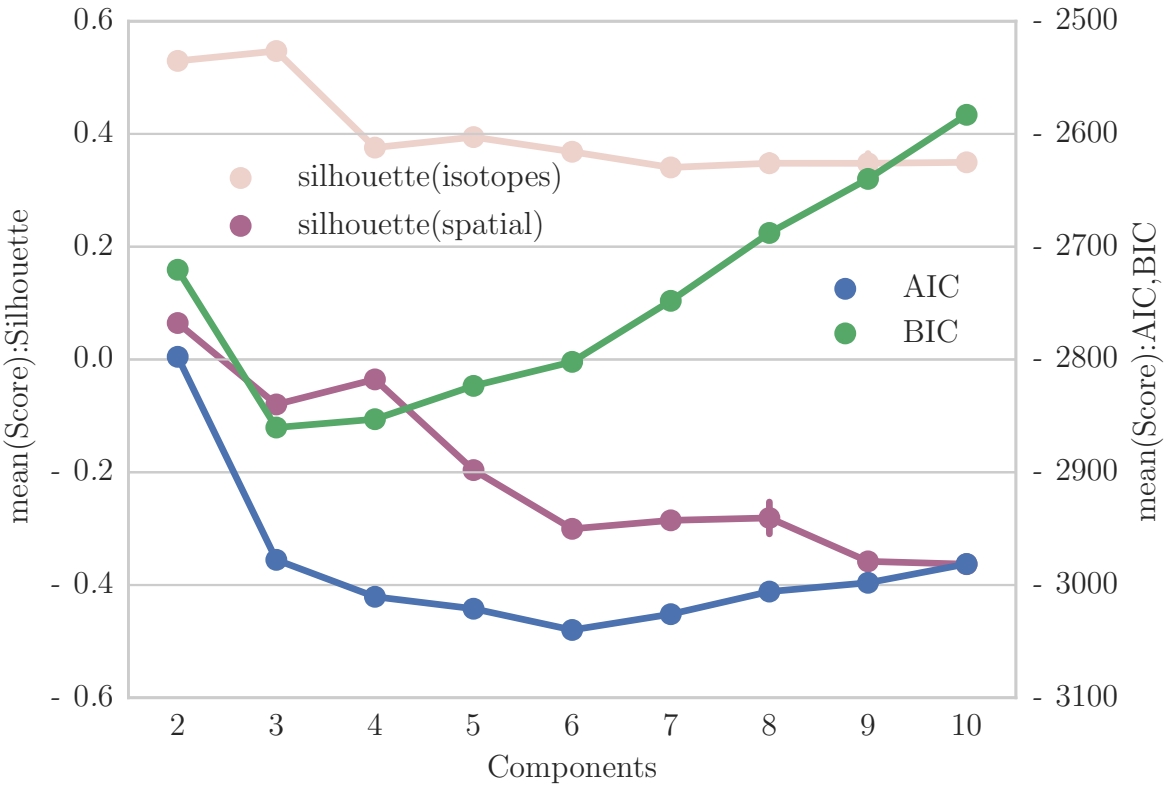
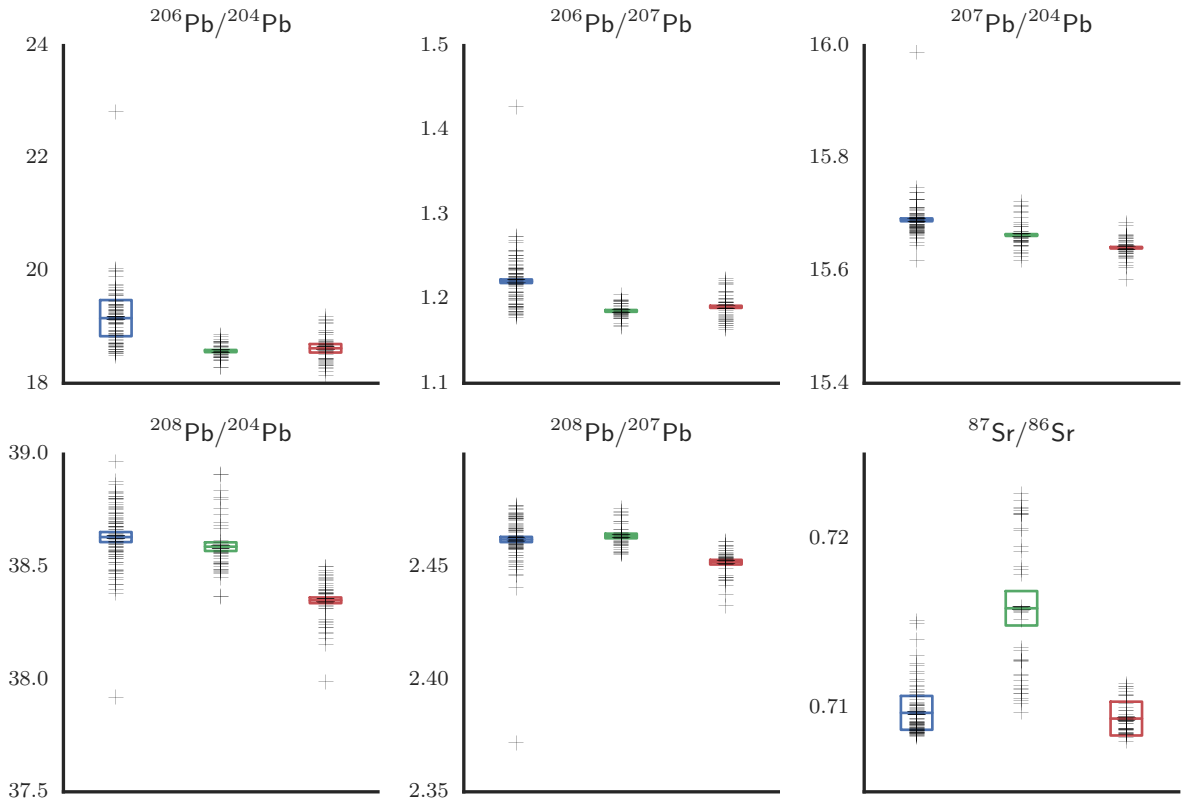


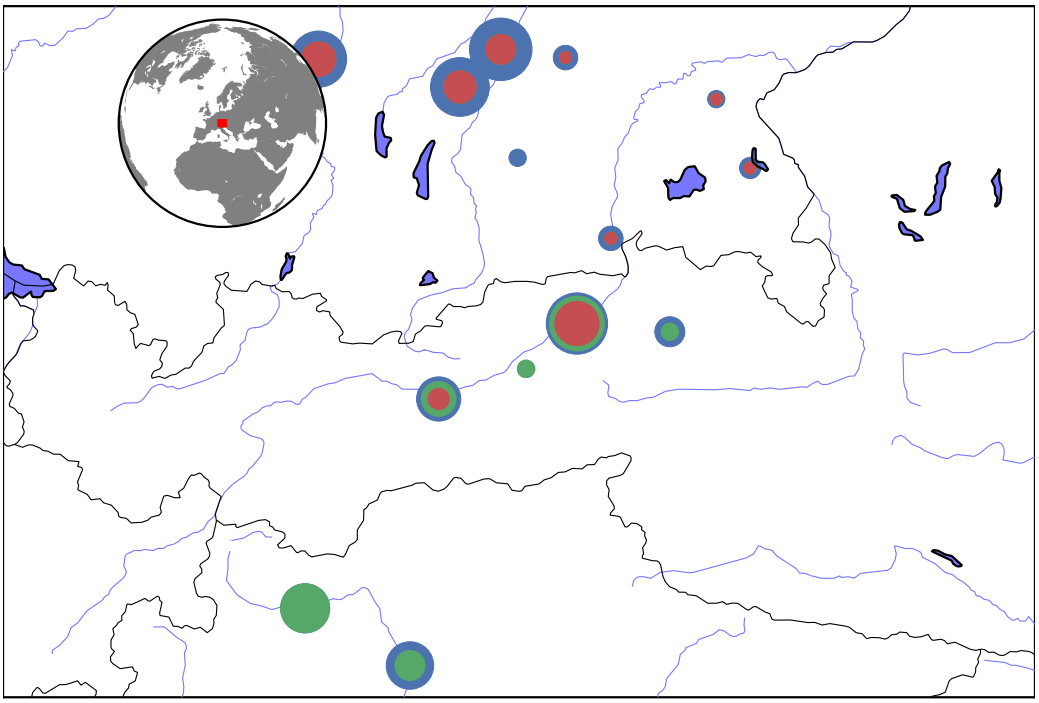
Figure 5.5: Model evaluation scores for different component numbers of the human data set.

from multiple clusterings and consist of points that were part of the same cluster in several clusterings. The tool's purpose is to help the user generate a model that complies with an intuitive notion of constraints. To capture the notion of a specific constraint is an intuitive task for a domain expert, while representing constraints mathematically and build or train an algorithm to incorporate them in a model is a complicated task. In addition, some constraints may be intuitive to an expert but hard or impossible for them to represent mathematically.

In this section, we applied GMMbuilder to the multivariate, continuous data with spatial components that was introduced in Section 1.2. Since the data contains attributes that map to spatial coordinates, GMMbuilder presents a maps-based view of the data and analysis results to assist the user in estimating spatial constraint fit. The goal was to generate a model that both explains the data distribution and is spatially coherent. Note that the resulting model relies solely on the isotope characteristics of the data, i.e. spatial coordinates are not used for clustering. However, the spatial coordinates will be indirectly incorporated into the model driven by the users' decisions to include certain communities over others.



(a) Distribution of components by attribute.



(b) GMMbuilder result map.

Figure 5.6: GMMbuilder result map. Converted from an interactively generated model. Contrary to the other examples, here k=3 seemed appropriate to the user.

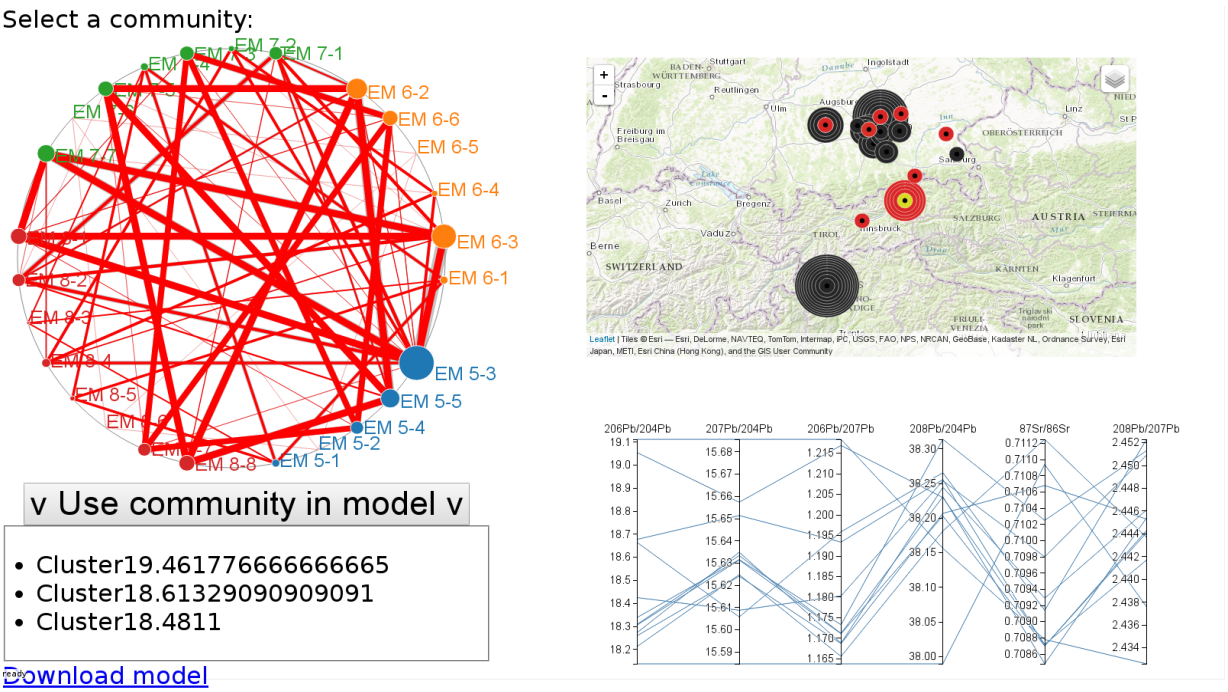


Figure 5.7: GMMbuilder being used to generate the model described in this section.

In our example scenario we generate different clusterings by varying the number of clusters for the EM algorithm from 4 to 8. Figure 5.7 shows a screenshot of the final model in GMMbuilder. Since EM produces a probabilistic model, the community extraction is based on the maximum likelihood assignment. This is acceptable as a previous evaluation [51] showed the assignments of the model to strongly favor one component in most cases. The users can inspect the individual clusterings, with the help of the clustering statistics and visualization window, and select those that they think are a good basis for community extraction. This would typically be that they characterize some regions well already. The tool then extracts communities from these clusterings and presents them in the community widget. The user can interactively choose any component and examine it in the map view and parallel coordinate view. This is important, because communities are based on several clusterings, which are all performed on the data domain, disregarding any constraint data. Thus, a community might consist of objects which are close in the isotopic space, but their spatial coordinates are far apart. Since domain experts are interested in an isotopic model that is also spatially coherent, the aforementioned community is not a good "seed" for the GMMbuilder and should not be picked. The user can then choose which of these communities should be part of the final model. User decisions are reflected in the final model so the user can directly inspect the effect of their decisions and proceed accordingly by removing or adding certain components.

To apply GMMbuilder to a real data set requires a domain scientist to manually choose communities and combine them into a GMM that satisfies all constraints they may have identified. In composing the model evaluated here, the author checked many communi-

ties for spatial coherence. It became clear fairly soon that some regions were repeatedly represented by different communities. These communities frequently consisted of similar subsets of the same points. Three communities were clearly limited to a small set of closely located sites. Three communities – each representing one site particularly closely – were picked and their models added to the mixture. This resulted in a set of three components, each representing one region north, inside, and south of the Alps. This model was exported and subjected to the aforementioned evaluation.

Figure 5.6 shows the resulting model. The three components each represent a part of the map more strongly than each other does. They presumably correspond to the regions from which the underlying communities were extracted. However, they are not as clearly separated from one another as would be desirable. Despite the spatially limited extent of the selected communities, the resulting GMM does not reflect this property. The assumption that points that remain together over several clusterings and have the same spatial origin are not representative for points from other locations may not be generally valid. They probably depend on the right set of complementing components to subsume these other points. The author was not able to construct a model that represented only a small region despite the strongly connected set of points that corresponded to these regions that were the basis for each component. It may be possible to refine these communities and get better results with more practice. However, as shown here, the results are not more impressive than those of an automatically generated GMM, which disregards spatial information.

5.5.4 Monte Carlo

This section describes the application of the method introduced in Section 5.4.2.

A common property of a spatially coherent solution (see the introduction to Chapter 5) is that the points in a component are within a certain distance of another point in the component. The evaluated approach expressed this in terms of a reachability graph that was extracted from the spatial distribution of the samples. To apply this method, we extract the reachability graph by representing each site as a node and connecting those nodes whose associated sites are within an Euclidean distance of $\epsilon = 0.5$ degrees of one another. [1] The choice of ϵ is based on the requirement to connect all points transitively while keeping the degree of the graph's nodes small to increase the likelihood that a random sample from the graph is spatially coherent. Figure 5.8 shows the resulting reachability graph.

Based on this reachability graph, k spatially connected components with a maximum diameter of 3ϵ are repeatedly generated. The choice of $\alpha = 3$ is based on a desire to

[1] Treating coordinates as euclidean vectors is of course not entirely accurate on a sphere. Using euclidean distances on latitude and longitude pairs results in a distorted representation of the great-circle distance. Due to the location of the points in the data set the experiments are based on, this causes longitudinal differences to have approximately twice the contribution to the distance that latitudinal distance has. However, since the maximum latitudinal distance between two points is less than ϵ degrees, this has little practical effect.

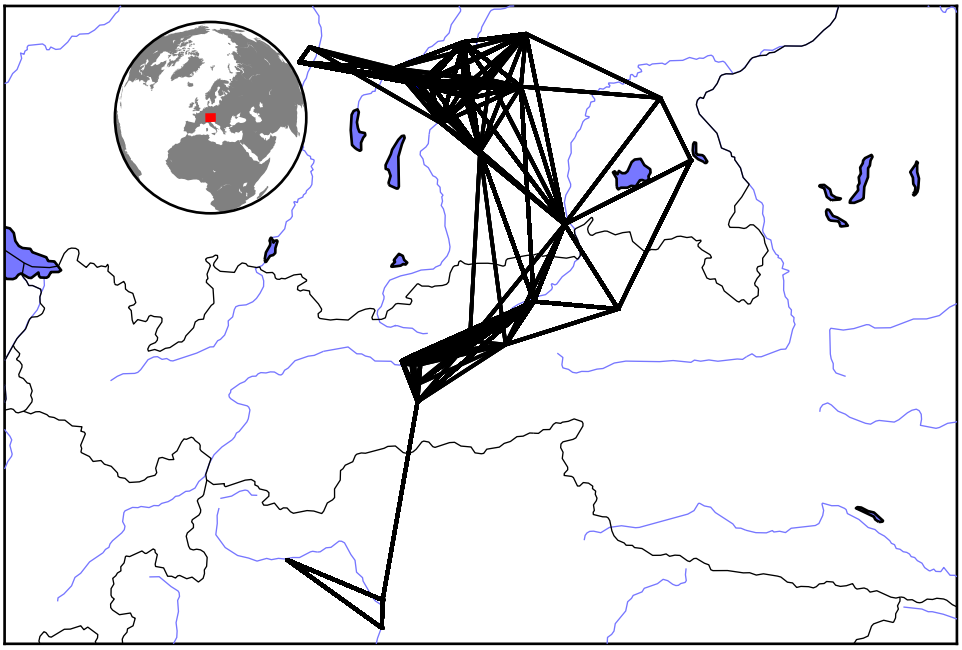


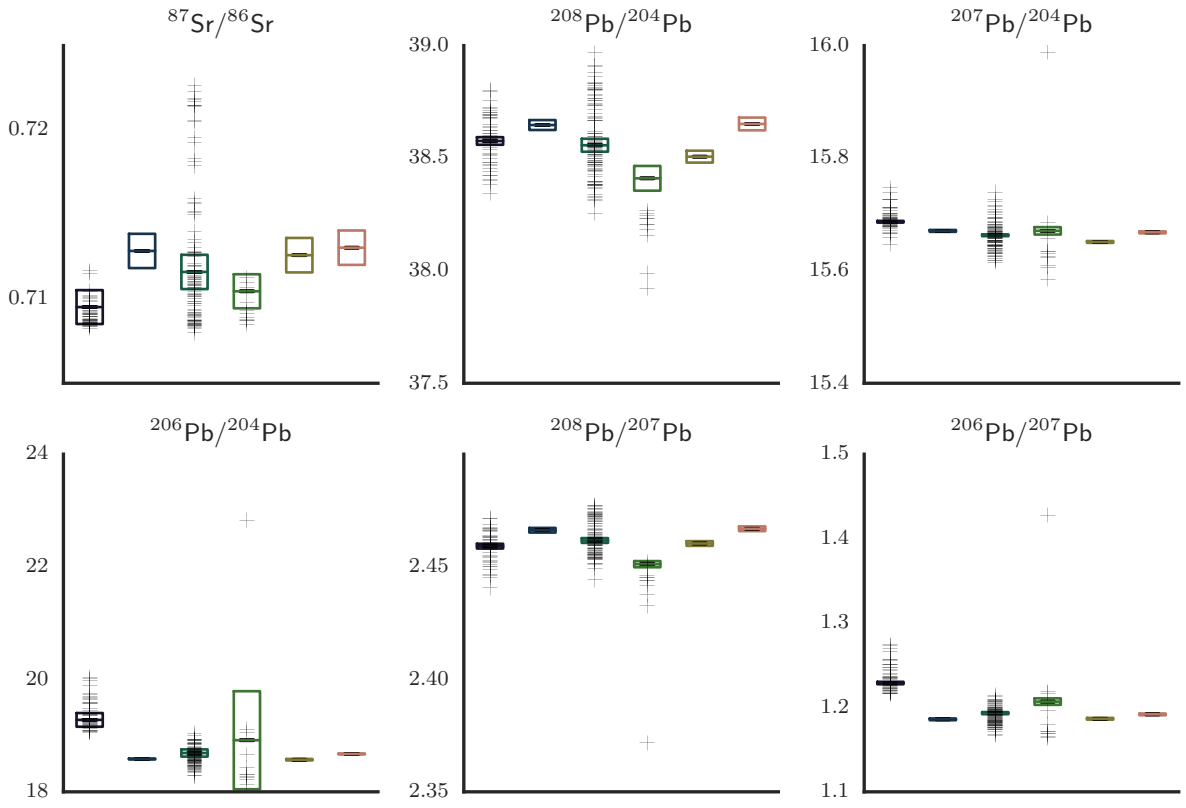
Figure 5.8: Plot of the reachability graph used in the evaluation.

reduce overlap between the extracted components (for high spatial coherence) and being able to reach as many members of a plausible feature model as possible (for a good model fit). Each set of components is converted into a Gaussian Mixture Model by calculating each component's mean and covariance. The resulting GMM is evaluated through the previously described model evaluation (see Section 5.5.1). This is repeated for $n = 10000$ runs and the best scoring model according to the silhouette score of the data domain is retained.

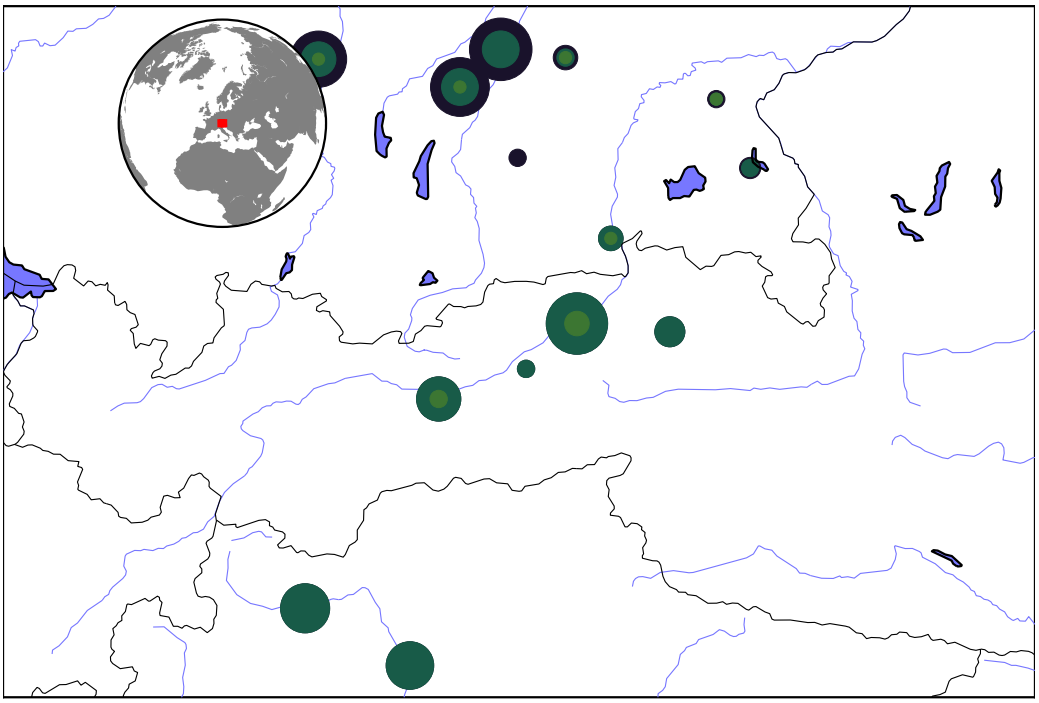
Figure 5.9 shows the model with the best silhouette score in k experiments. Although the Monte Carlo based point assignment consisted of $k = 6$ components, the resulting maximum likelihood assignment based on the GMM generated from this assignment contains only three classes that points were assigned to. The silhouette score of this assignment was remarkably high at 0.49, while still achieving a positive spatial silhouette score of 0.02.

5.5.5 Constrained EM Algorithm

This section evaluates the algorithm introduced in Section 5.4.3. To illustrate how the presented algorithm works, we perform two evaluations: The first is on a synthetic data set intended to illustrate how the algorithm proceeds to find constraint compliant models.



(a) Distribution of components by attribute.

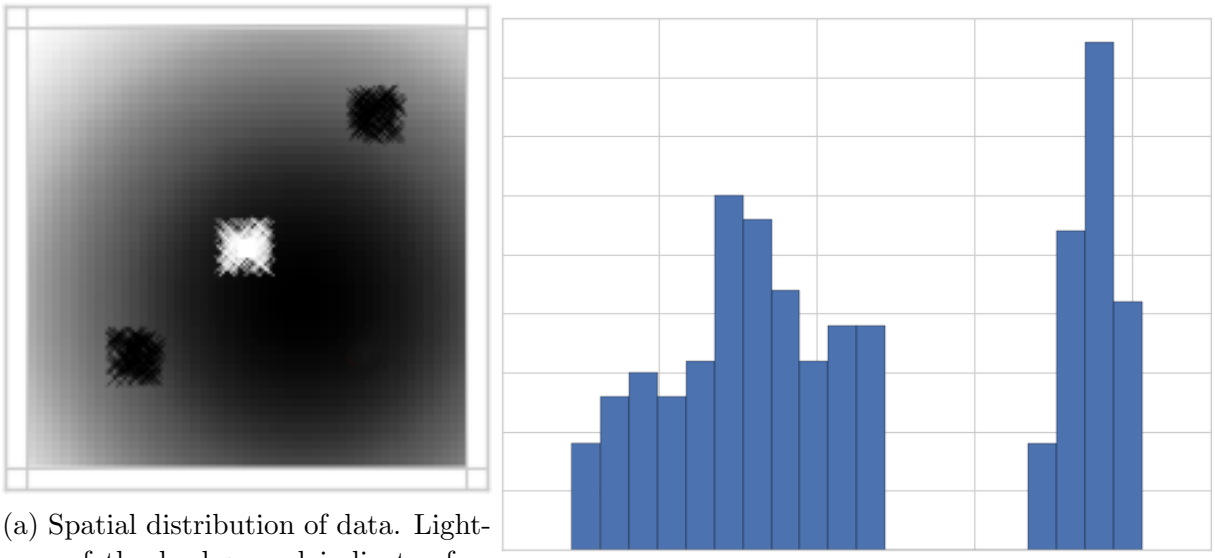


(b) Monte Carlo result map.

Figure 5.9: Monte Carlo result. Best of 10,000 runs according to spatial silhouette score.

We show a plot of the calculated likelihoods and how they are being incorporated into the Expectation step to reach a model state that complies with the constraints. To illustrate that our technique is applicable to real-world problems, we present an application to same real-world data as the other approaches.

The behavior of our algorithm is as follows: In an early phase, when the model has not yet diversified (resulting in a low likelihood for all components), components with overlapping or similar distributions will likely be characterized by similar means. This results in probabilities close to the a priori probability. Regarding the constraint data, this will result in a fairly low cumulative probability of points in these components. This will be similar for each of these components and thus not influence the model much. When the likelihood of the model increases, some points will not fit with similarly modeled points and receive a lower constraint probability. This will contribute to a stronger influence of spatial disagreement. Shrinking constraint likelihood shifts the model towards accepting the subset of points more whose constraint likelihood is higher. This bias will make the groups differentiate such that one is more likely to accept points that have a different spatial origin as another. The model will stabilize on a state whose two kinds of likelihood are both relatively high. This will allow the model to explain new data in terms of the measurement distribution, while increasing the likelihood that the description actually matches a value that is in line with domain knowledge.



(a) Spatial distribution of data. Lightness of the background indicates feature values.

(b) Resulting data distribution.

Figure 5.10: Synthetic data set illustrating the resulting mode of operation.

Synthetic data Figure 5.10 shows a synthetic data set that will be used to illustrate the inner workings of the presented algorithm. Figure 5.10a shows the spatial distribution of the data. Figure 5.10b shows the resulting value distribution. Clearly, there are

only two obvious distributions in the data that can be estimated. Considering the spatial information, it becomes clear that the left distribution is actually made up of two distributions that are spatially disjoint. The data set was generated from a pre-defined Gaussian Mixture Model. To reduce outside influences, the model from which the data is sampled is very simple and consists only of a single Gaussian distribution. The sample regions are chosen such that the resulting values suggest a distinct distribution and that two components have similar, though not identical, feature values. The constraints are derived from a two-dimensional space, which corresponds with the measurement locations via a covariance matrix. This can easily be pictured as a spatial distribution, which is sampled in three distinct locations. The sample locations are distributed randomly over a square region to simulate some degree of noise.

Figure 5.11 shows how the optimization of the data functions. The colors indicate one component each. Initially, the chosen starting configuration puts the component depicted in green firmly in the distinct distribution. The red and blue components have a similarly high probability for each component. The lightness of the colors in the figure code for the spatial position. We can see that the spatial position appearing darker is penalized for the blue component due to its current spatial projection.

As the model develops, the spatial penalty generates a divergence between the green and red component, resulting in their models diverging as well. They differentiate according to their spatial position (dark red component gravitates up and so does the light green). After convergence (Figure 5.11) the spatial model has differentiated so much that the spatial probability is close to 1 or 0 for all points, i.e. the spatial structure has been reproduced very well. The reason for the at first sight extremely well fitting spatial model is the clear separation of points in the spatial domain. As soon as one component has begun to prefer the value ranges corresponding to one location, that component's spatial model will have a very low variance, resulting in the observed extreme spatial probabilities. The resulting model components still overlap (necessary to model very close distributions), but the lighter colored points from one distribution dominate in the blue component, whereas the darker points from the other distribution dominate in the other.

Case Study: archaeo-biological measurement data To illustrate the real-world applications of the presented approach, it was applied to a real-world data set (see Section 1.2.2). Let us briefly reiterate the important aspects of this data set: The data set is a multivariate set of isotope measurements from an archaeological research project covering a route from Italy through the central European alps, into Germany. The samples are extracted from the remains of 162 humans found at 30 locations in the investigated area. The sampling was driven by the availability of samples at known archaeological sites along the route. Their spatial coverage is therefore very low. The target model uses isotope measurements to establish a likely place of origin for the specimen under investigation. We use the isotope measurements as data and spatial information as constraints.

Figure 5.12 shows the results of applying scEM on the presented data set for different values of α. The result indicate that the training and the test sets have identical

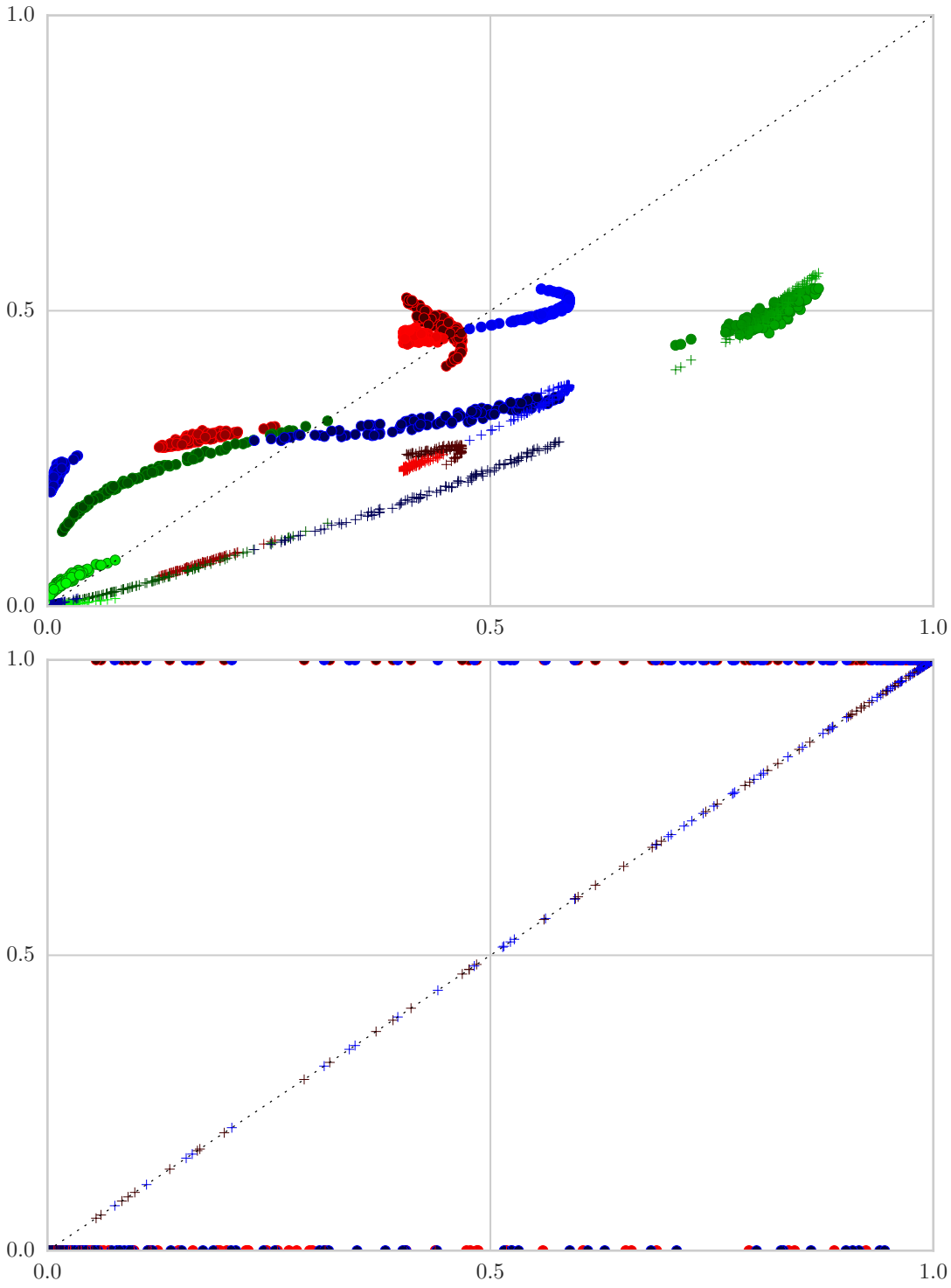


Figure 5.11: Probability of each point's membership in each component before and after convergence. x-axis represents probability of component membership, while y-axis represents the spatial constraint membership probability. Lightness indicates the point's spatial position (so the lightness (position) and color (component) with the highest probability (x-axis score) of any point should correspond). In a traditional EM x-axis and y-axis values would always be identical (dashed line).

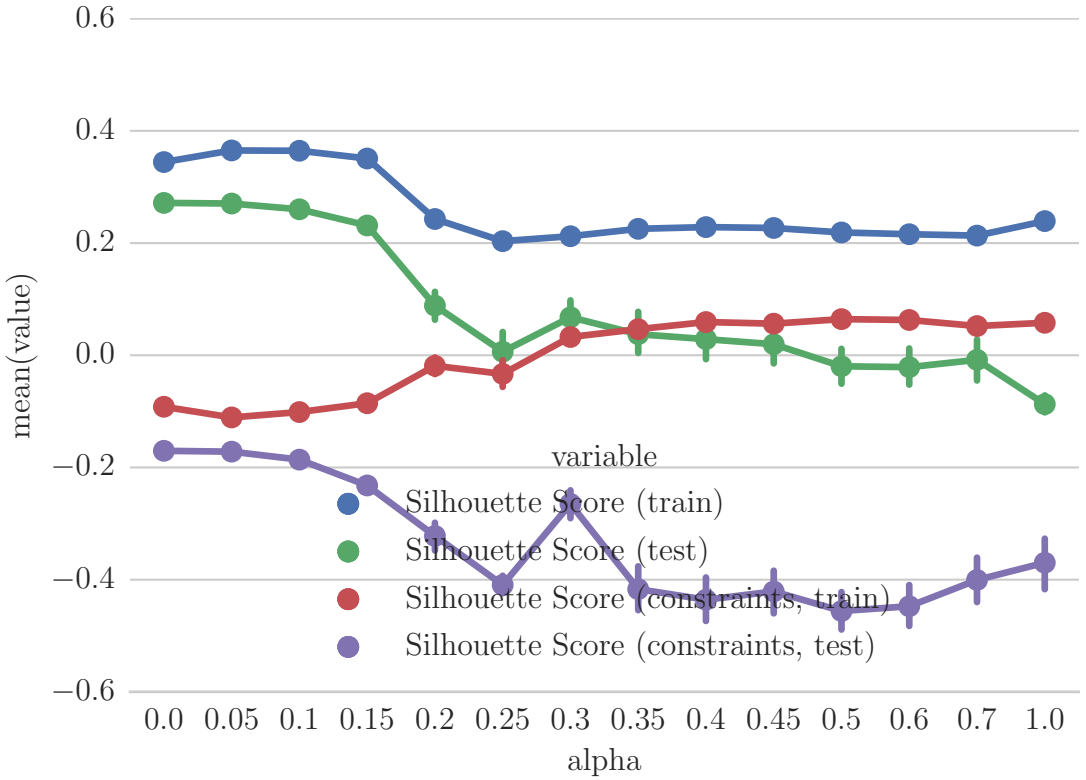


Figure 5.12: Influence of the parameter weighing data and spatial coherence.

probabilities around $\alpha = 0.35$. This value will be used in the rest of this evaluation.

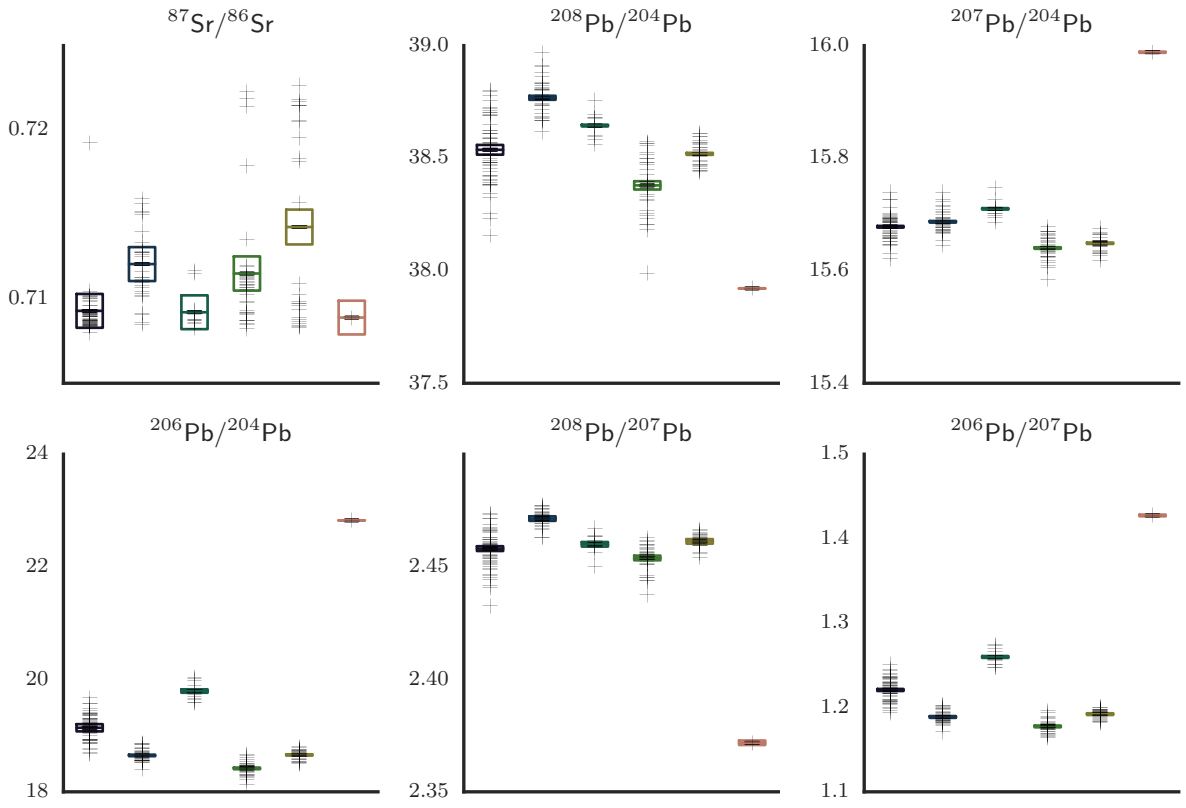
The resulting model's properties are shown in Figure 5.13. It has a silhouette coefficient over the data of 0.29, which is a worse fit than vanilla EM, but with a much improved spatial silhouette score of -0.16 vs EM's -0.29.

In the next section we will attempt to use the generalized EM algorithm to analyze the same data sets.

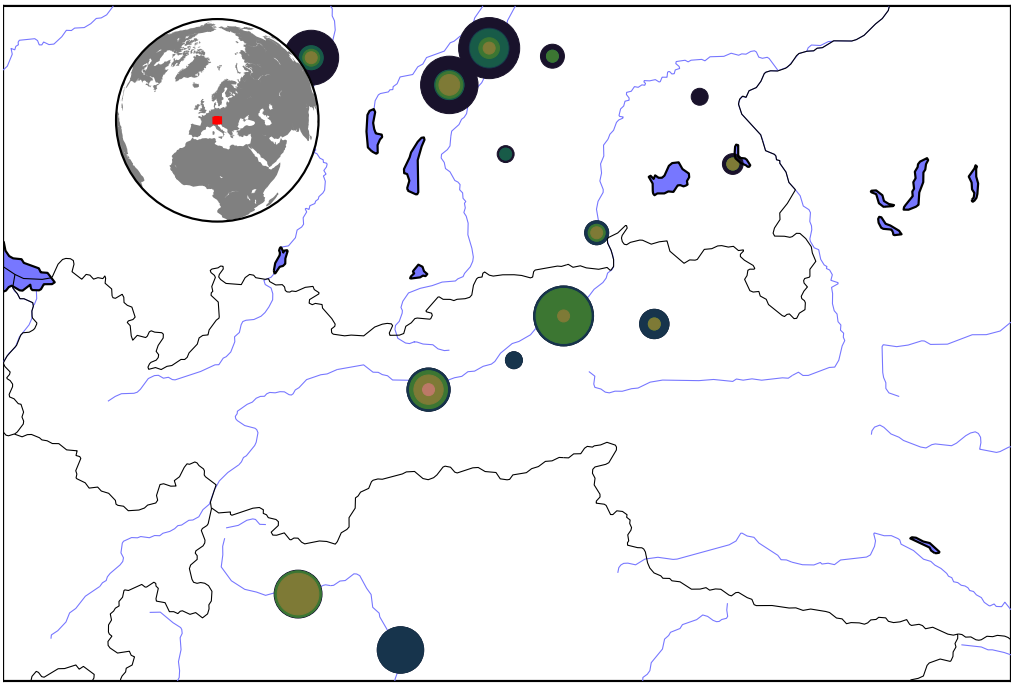
5.5.5.1 Preliminary Results: Distance Based Constraints

The preliminary generalization of EM to pairwise distances was implemented with the mentioned heuristical Maximization equation. The pairwise similarity matrix W was created according to the method described in Section 5.4.4.1.

Synthetic data Analogously to constrained EM approach, Figure 5.14 shows the optimization of the involved probabilities. The top figure shows the initial configuration after seeding the algorithm with a random point as its mean. One of the spatial clusters has been hit by exactly one mean value (red) that results in a high probability of the points in this cluster. The green and blue components' means have both hit the wider cluster and



(a) Distribution of components by attribute.



(b) Constrained EM result map.

Figure 5.13: Constrained EM result. Best of 500 runs according to spatial silhouette score.

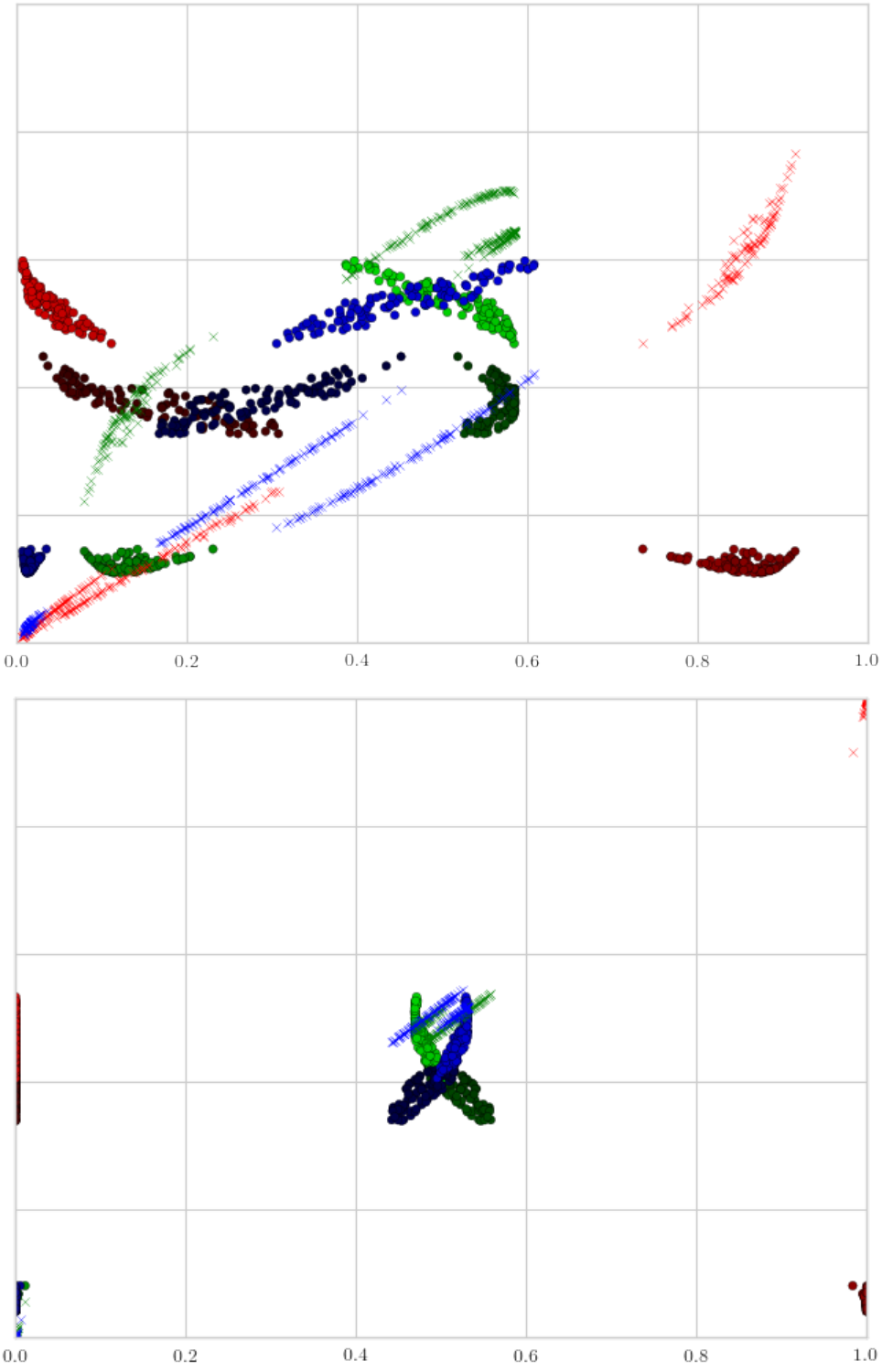


Figure 5.14: x-axis represents the data based model's estimate of probability p per point (x value) and component (color). y-axis represents the total constrained probability $p^{\mathcal{C}}$ (pluses) and the spatial model fit term $\hat{p}$ (dots). Lightness of color represents spatial position. In a spatially coherent model, components (color) with highest probability (x value) should correspond with spatial position (lightness).

the resulting models are in a less clear defined state. The dark green and dark blue points are from one spatial location, while the lighter green and blue points are from another. As we have seen in Figure 5.10 these distributions are not separate in feature space and can only be differentiated by their spatial location to build a model that represents the on average slightly different values of the two component distributions. After convergence the model of the cluster that was already well represented (red) has been differentiated strongly. The two other components have resulted in quite similar probabilities (informed by the necessity to model very close distributions), but the lighter colored points from one distribution dominate in the blue component, whereas the darker points from the other distribution dominate in the other.

Case Study: archaeo-biological measurement data This algorithm has also been applied to the same archaeo-biological data as the other approaches. The resulting model is presented in Figure 5.15. Despite its heuristical Maximization step (which is identical to the original EM-GMM algorithm's), the achieved results are quite impressive. The spatial coherence is visible in the map projection and the resulting scores are also impressive. The approach reaches a silhouette score of 0.38, which is higher than the constrained EM score. Also, at -0.05 its spatial silhouette score is still below zero, but also higher than constrained EM's.

5.6 Conclusion

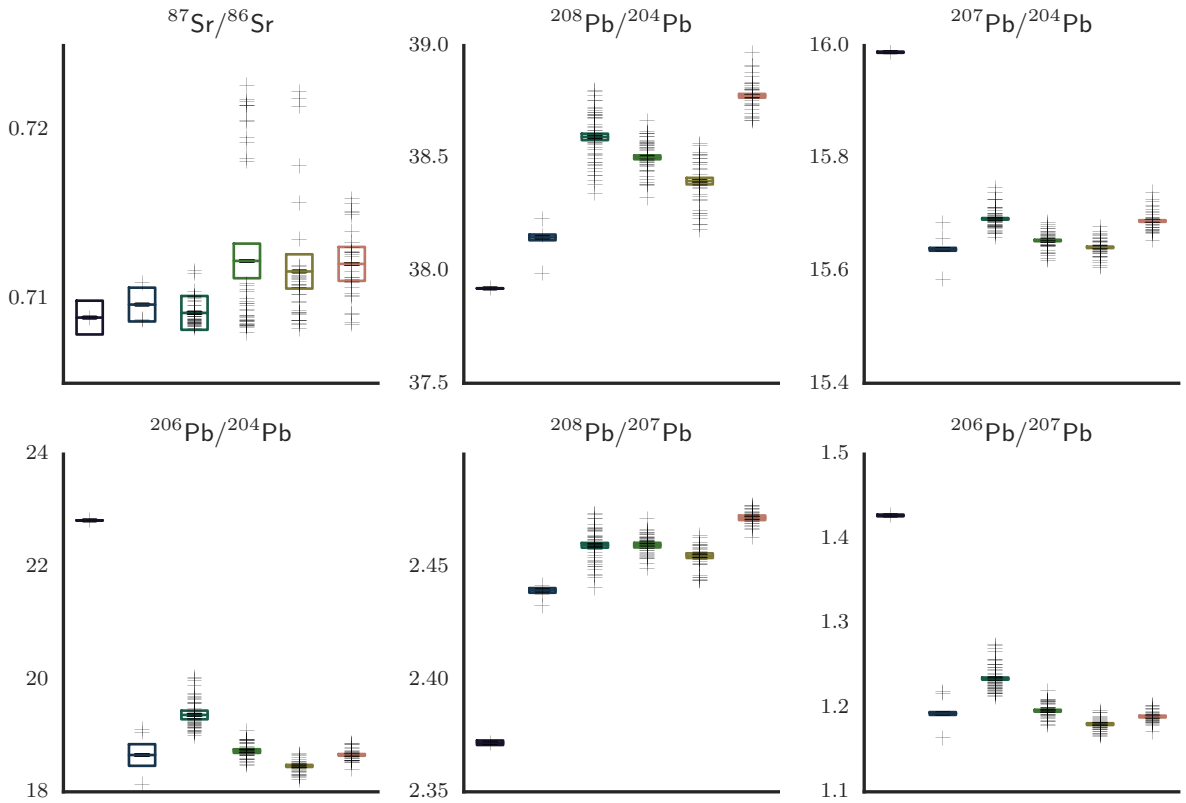
We introduced the problem of building a spatially coherent, model-based feature-space models. A solution to this problem consists of a model which is defined purely in the data domain, but has been optimized to agree with a notion of spatial coherence in the training set.

We proposed several different approaches to finding constraint compliant spatial models of the data.

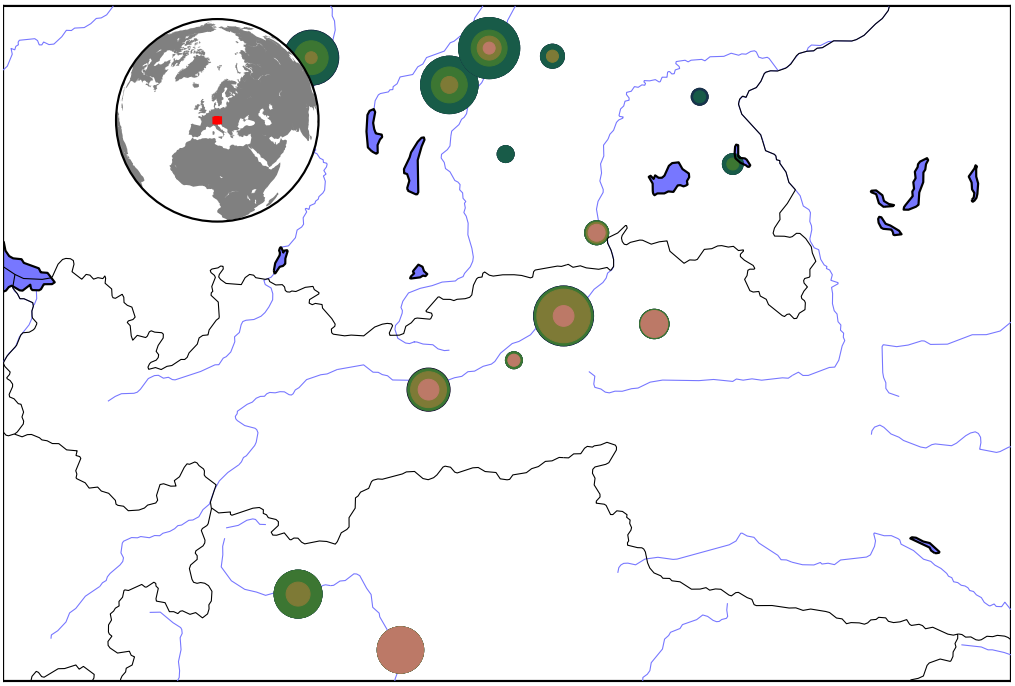
GMMbuilder An interactive tool allowing domain experts to view especially robust subsets of points and their projection into the spatial domain. By picking particularly promising regions as defined by (possibly implicit) domain knowledge, the expert can guide the algorithm towards a model that complies with any constraint.

Monte Carlo An approach using a generalization of spatial coherence (the reachability graph between sites) to define possible subsets of points to define components. By evaluating these components, this approach can pick a model that is not only spatially coherent (as asserted by the input set), but also well-expressed in the data domain. This approach needs to randomly explore a very large search space, so no guarantees can be made about its performance. A small data set is beneficial for this algorithm.

Constrained EM This solution suggests modified E- and M-step equations for the EM algorithm. These equations keep track of the distribution of the data in both the



(a) Distribution of components by attribute.



(b) Generalized constrained EM result map.

Figure 5.15: Generalized constrained EM result. Best of 500 runs according to spatial silhouette score.

data and the spatial domain and weigh the goodness of fit of each according to a user-defined parameter.

Generalized Constrained EM Like the previously described constrained EM algorithm, this algorithm modifies the involved equations to incorporate constraints in the solution. These equations allow it to consider the spatial distance of each point in conjunction with its assignment probability to give a score of the agreement of the spatial dimensions with the description of the data set's structure driven by the feature dimensions.

In Section 5.5, we applied these approaches to the problem of isotopic mapping of the transalpine Inn-Eisack-Adige passage across the German-Austrian-Italian Alps. This passage, which has been used since prehistoric times, is of great archaeological interest. Our finding over real isotope measurements from human remains discovered in the Alps, suggest that our methods offer good models of the area under investigation and offer a useful tools for domain experts for understanding the isotopic fingerprint of the area.

5.6.1 Resulting Models

In this chapter we saw several approaches to solving constrained Gaussian Mixture Modeling tasks. Here we will see which approach yielded the best model. The results are shown

	EM	constrained EM	generalized cEM	GMMbuilder	Monte Carlo
AIC	-3039.74	-2975.13	-2999.74	17742.62	-2690.72
BIC	-2801.99	-2737.39	-2761.99	17869.21	-2452.98
Sil(data)	0.37	0.29	0.38	0.09	0.49
Sil(spatial)	-0.29	-0.16	-0.05	0.02	0.02

Table 5.2: Performance measures of the presented spatial models.

in Table 5.2.

The spatial projection of these models all show clusters primarily represented in the north and south of the area. In most cases, the southern cluster extends into the north, mixing with an additional component in the center. Although the models consisted of six components, the resulting models' maximum likelihood assignment sometimes only assigned points to one of three components. Interestingly, when the parameter $k = 6$ was first chosen (see Section 5.5.2), the choice to make $k = 3$ was discussed, but ultimately discarded for making the model implausibly simple.

The scores each model achieved were surprisingly diverse. The smallest AIC was achieved by EM (the baseline), which is unsurprising. Of the new approaches, the generalized constrained EM algorithm performed best according to this measure. Similarly, the smallest BIC (after EM) also was achieved by the generalized constrained EM algorithm. The best silhouette score over the data was achieved by the Monte Carlo approach

(followed in some distance by the generalized constrained EM algorithm again). Finally, only Monte Carlo and the manually generated model from GMMbuilder achieved positive spatial silhouette scores at a meager 0.02.

Overall, despite its theoretical shortcomings, most scores would suggest the generalized constrained EM algorithm as the best approach.

In the following chapter we will look at methods to interpret the results of the previous chapters. Present them to domain experts and validate whether the results are plausible.

In the following chapter we will see how these models can be applied to help domain scientists answer relevant questions.

Chapter 6

Applications of Spatially-Constrained Gaussian Mixture Models

Attribution

This chapter uses material from the following publication:

- M. Mauder, E. Ntoutsi, P. Kröger, and H.-P. Kriegel. The isotopic fingerprint: new methods of data mining and similarity search. In G. Grupe, A. Grigat, and G. C. McGlynn, editors, *Across the Alps in Prehistory: Isotopic Mapping of the Brenner Passage by Bioarchaeology*, pages 105–125. Springer, 2017

See Section 1.3 for a detailed overview of incorporated publications.

The previous chapter described data and constraints as feature models. These feature models (as we saw in Section 5.4.3) can be translated into a different subspace using the maximization step of EM on this data projection. By using this connection, the descriptive models can be turned into predictive models over data from that subspace. In the common case of spatial data (which was e.g. used as constraints in Section 5.4.3.3) this can be used to predict feature values for a given spatial locations. By predicting values for a dense grid of data, this can be used to generate a map of the results (see Section 6.1). Inversely, using the feature values of a sample of unknown origin can be used to predict a location or region (see Section 6.2).

6.1 Making Maps

Statistical data modeling based on Gaussian models is a common and powerful technique that is used in many domain sciences. The purpose of these models is to allow domain experts to reason about their data. However, so far the models the presented approaches generate only live inside a computer in an abstract state as a statistical structure. In order to make the model accessible for domain scientists they should be presented in a more relatable way. A common tool for presenting spatially distributed data are maps, which have been in use for many millennia and are ever more present today.

Given only a GMM over some data does not allow for a map-based presentation of course. Instead we again require the connection to the spatial origin of the data. To express it in the familiar way, the input is both the data and the constraint that the data shall be presented as relating to the auxiliary spatial information. The resulting presentations can be used by domain scientists to evaluate theories and understand spatial distributions of complex data sets.

To construct this map, it is helpful to be able to derive a new model from an existing one by applying its probabilities onto measurements from another domain. Given the data upon which the feature model is based and associated spatial dimension, we can calculate the membership probability of each point and use this information combined with the spatial information to derive a corresponding model in the spatial domain. This spatial model can then be used to determine the spatial extent of each data model.

Of course, to make the map intuitive requires not only the presentation of each model's spatial projection separately, but also to combine them in an intuitive way. The obvious choice for distinguishing the components visually is color. The simplest way is to pick equidistant colors from some rainbow for each component, but this does not convey any information other than that the components are distinct. By choosing colors whose presumed (dis-)similarity is related to the represented models' similarity the user of the map can get an intuitive feeling for how similar two regions are. In combination, the spatial projection of the data and colors corresponding to the semantic values contained in the model visualize the semantic connection between the models and their influence on the spatial projection.

In a first step, we generate a spatial projection of the model.

6.1.1 Spatial Distribution

Given a model Θ, data X, and corresponding spatial data Y, we can determine a spatial model $\Theta^{\mathbb{C}}$. With a Gaussian Mixture Model $\Theta = (\mu_k, \Sigma_k)$, the probability function p can be used to determine the probability of component membership for each component c_k. These probabilities can then be applied to Y to determine the spatial projection of the model.

$$\mu_k^{\mathbb{C}} = \frac{\sum_{i=1}^{n} p_k(x^(i)) \cdot x_{\mathbb{C}}^{(i)}}{\sum_{i=1}^{n} p_k(x^{(i)})} \tag{6.1}$$

The covariance is then estimated over the data Y from this newly updated $\mu^{\mathbb{C}}$.

To express the probability of a spatial location to belong to a given component, we can calculate the spatial probability density and normalize to sum to 1. This relative probability of a point to belong to a component can be applied to the data model to calculate an expected value.

$$E(x^{\mathbb{C}}) = \sum_{j=0}^{k} p^{\mathbb{C}}(x^{\mathbb{C}}) \cdot \mu_k$$

This yields a data vector with the highest probability for the given location according to the models Θ and $\Theta^{\mathbb{C}}$.

6.1.2 Color Model

To plot a map, this data vector must be transformed into a visible representation. As with the spatial projection, we are again using the model to transform it into a different representation for a different purpose.

We require an intuitive representation of the (possibly multidimensional) underlying feature model. We are facing the issue that the model is made up of k components consisting of m mean values and an $m \times m$ covariance matrix each. This is a lot to represent in a color space that is typically represented by no more than four values. To address the goal of having the color represent human perception of difference between components, we pick a color model that is well suited. The LUV color model is designed to preserve perceptual differences proportional to the Euclidean distance between the color vectors. Given that LUV is described by three values, we need to represent each model as a three-dimensional input.

The model is expressed in a data space that is m dimensional. If $m < 3$ it can be trivially extended by e.g. duplicating attributes. If $m > 3$ the entire data space is reduced to three dimensions using principal component analysis. Normalization of the (possibly transformed) component means yields a set of three-dimensional vectors that can interpreted as LUV vectors like this:

$$v(x) = \frac{\mathrm{PCA}_3(x \mid d) - \min(\mathrm{PCA}_3(d \mid d))}{\max(\mathrm{PCA}_3(d \mid d)) - \min(\mathrm{PCA}_3(d \mid d))}$$
$$col(x) = \mathrm{RGB}_{\mathrm{LUV}}(v_0(x) \cdot 100, v_1(x) \cdot 200 - 100, v_2(x) \cdot 200 - 100)$$

where d is the training data, $\mathrm{PCA}_3(x \mid d)$ transforms x according to d's three principal components, and $\mathrm{RGB}_{\mathrm{LUV}}$ is a function that transforms LUV color space to RGB for displaying. The scaling of the values of v are chosen to approximate the LUV color space's limits, but due to LUV's design being modeled after human perception, it is possible for this formula to yield invalid (but approximately correct) RGB values. This problem can be addressed by normalizing the resulting values to valid RGB feature ranges in a final step. This weakens the perceptual distance between the points, but is limited to the small range that was not expressible as LUV values.

Given the presented color model, it becomes possible to express the expected feature values in the region of interest.

6.1.3 Projection

Given a GMM over the spatial data and colors corresponding to a feature value, all that is left to do is to combine the one with the other (i.e. pick places to which to apply the color). To build a visual map that covers a given area, a naive approach is to break the area into small enough pieces to calculate one representative area for it. These pieces may be pixels inherent in some output medium. To determine which color to use for each piece, various levels of simplification of the model can be used.

6.1.3.1 Maximum Likelihood Projection

In Section 6.1.1 we saw how to calculate a representative value for any given spatial coordinate. However, the simplest way to represents each component is as its most likely value. This value (which is of course the model's mean) is a single value, is represented by a single color. The regions that these colors are applied to are based on which component is most likely for a given coordinate. [1] We find that in practice domain practitioners find the Maximum Likelihood projection to be most intuitive.

Figures 6.1 through 6.5 show the maps that result from applying this projection to the models built in Sections 5.5.3 through 5.5.5.1. As suggested by its previous evaluation, the map based on the GMMbuilder model (Figure 6.1) is unhelpful. Two similar components surround one distinctly different one, which makes for a very unhelpful map. The EM based result (Figure 6.2) identifies distinct regions, with two notably distinct spatially small clusters near Innsbruck. The Monte Carlo map (Figure 6.3) is surprisingly unhelpful, with one component covering most of the surveyed area. The constrained EM result (Figure 6.4) is quite well compartmentalized, with three distinct areas in north-south direction and a second, larger, area in the south. It too shows the distinct cluster near Innsbruck. And finally, the generalized constrained EM result (Figure 6.5) is reminiscent of the constrained EM result with one additional area covering the east of the area in a distinct color.

6.1.3.2 Continuous Projection

A more involved projection of the model onto a map uses the transferred model to predict and visualize the exact value that is most likely at the current position. Contrary to the Maximum Likelihood projection, it acknowledges that values vary with space and expresses them as colors. The value at each position on the map is calculated using $E(x^{\mathbb{C}})$ and individually translated into a color using $col(E(x^{\mathbb{C}}))$. To achieve this, it gets the probability for each component in the spatial domain and then weights the corresponding components in the feature domain to arrive at a predicted feature value. This feature value is then translated into a color. Get the actual feature value at this location and represent that. i.e. stay in the feature representation as long as possible.

[1] In theory an even simpler model would use Voronoi cells around the spatial mean. This approach does not need any of the model transferring of Section 6.1.1. However, it ignores the covariance matrix altogether and is as such equivalent to applying the maximum likelihood approach to a suitably constrained spatial GMM. In this sense the Voronoi approach is a special case of the maximum likelihood one.

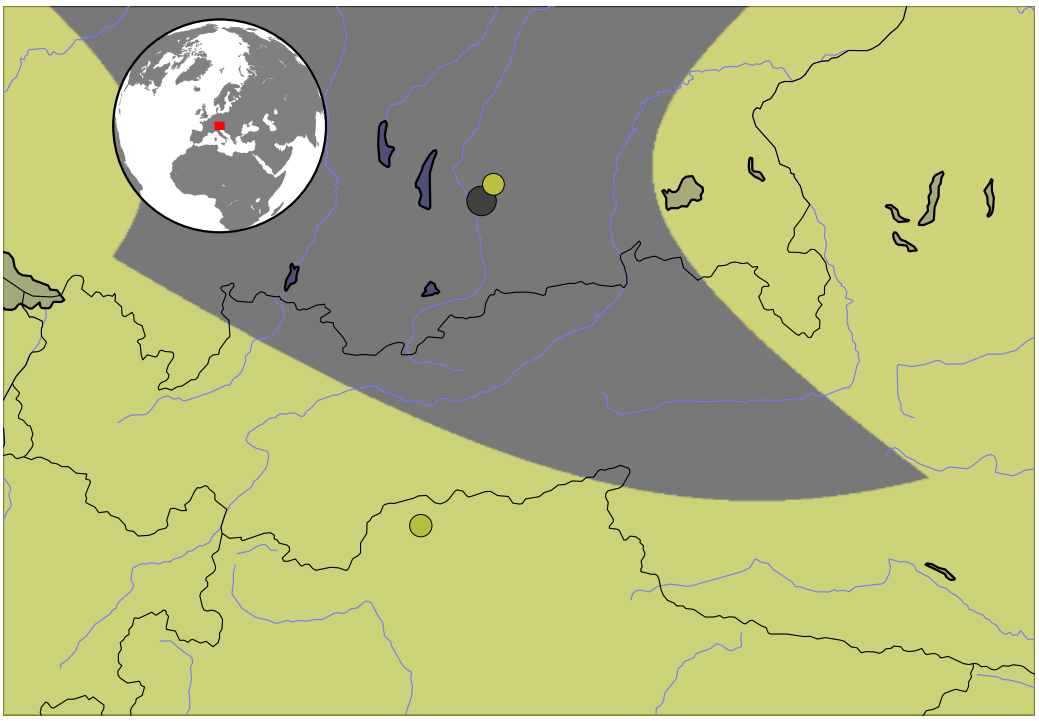


Figure 6.1: GMMbuilder result map.

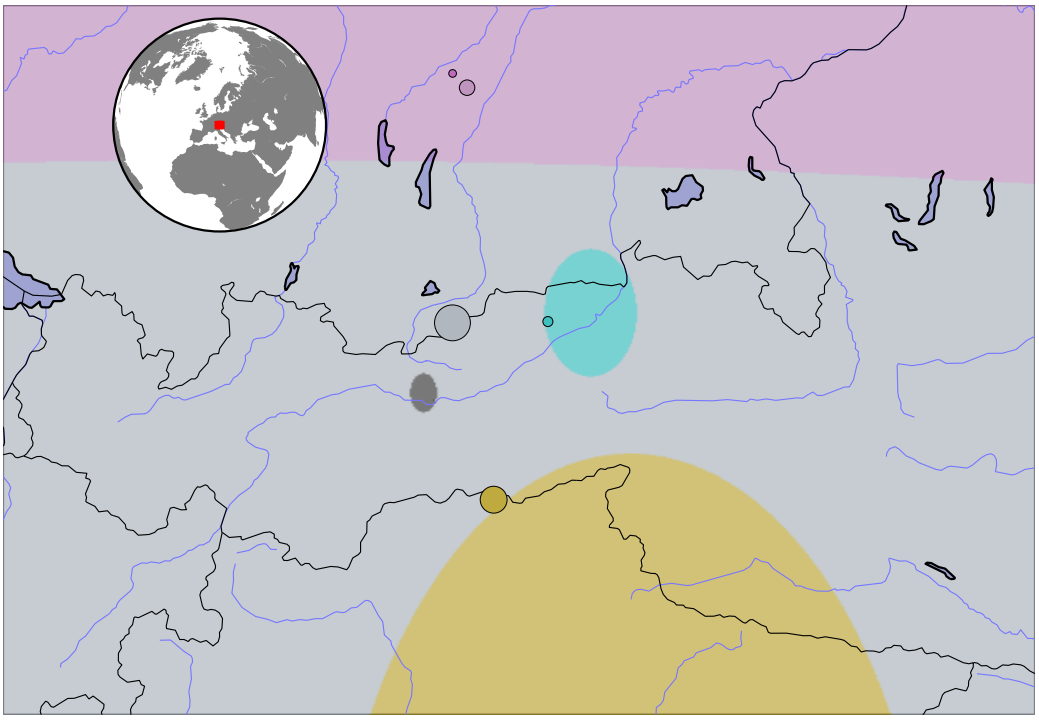


Figure 6.2: EM result map.

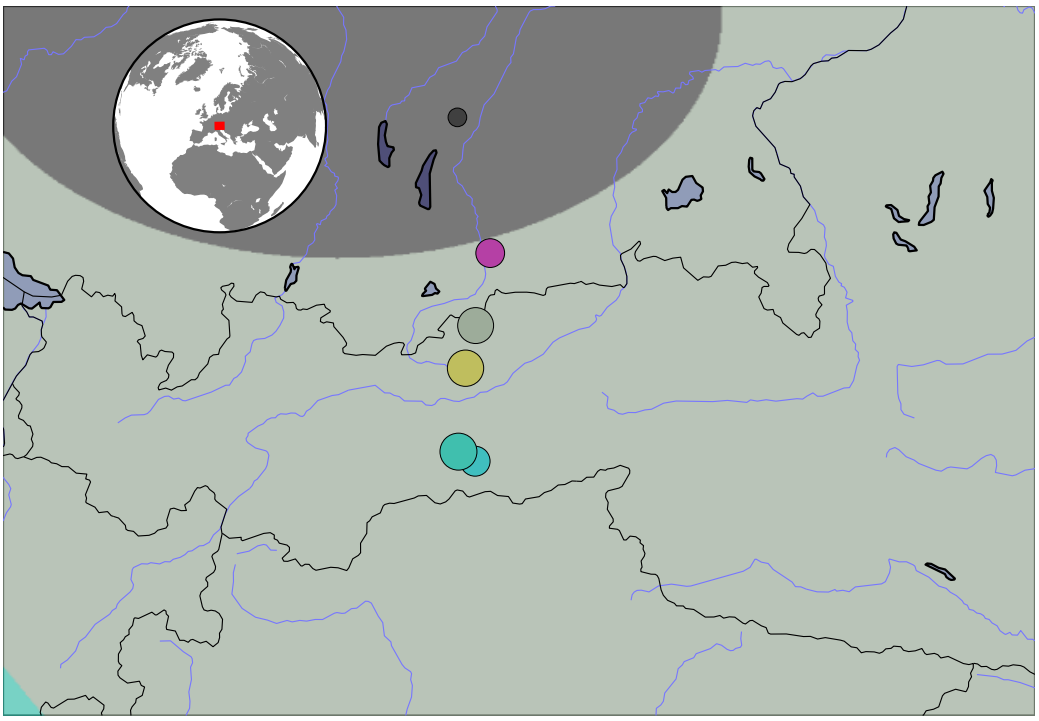


Figure 6.3: Monte Carlo result map.

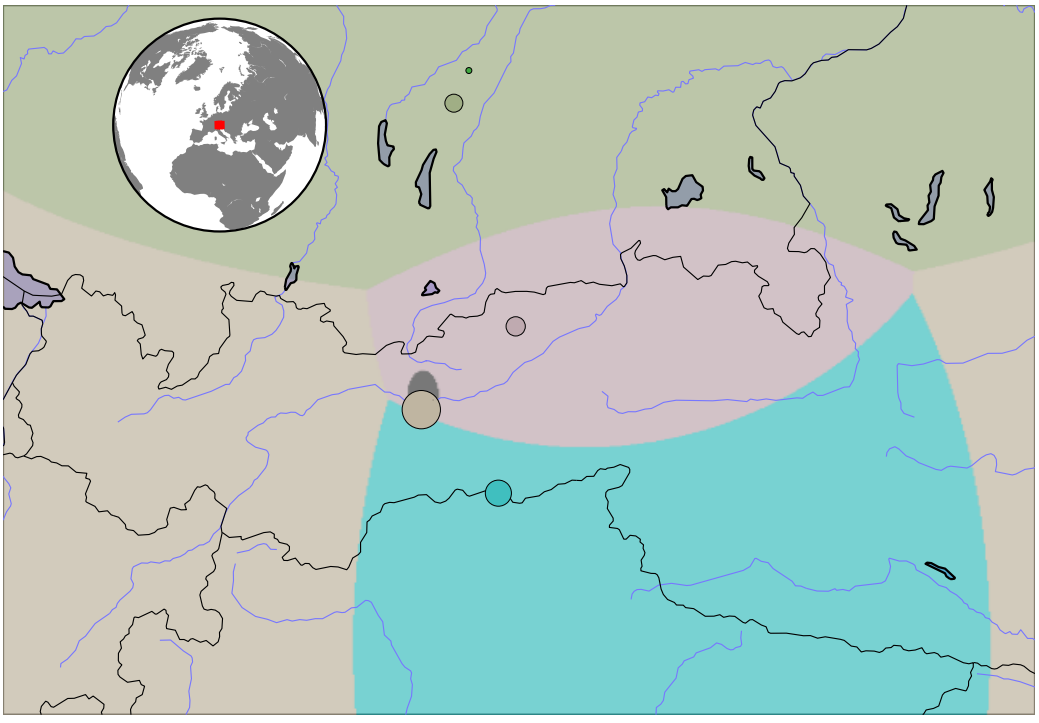


Figure 6.4: Constrained EM result map.

The Gaussian Mixture model over the feature space allows predicting the value generated by the mixture of the feature means. This allows assigning any given point an appropriate feature value and corresponding color.

Figure 6.6 shows the continuous projections of the generalized constrained EM algorithm, corresponding to Figure 6.5.

6.2 Outlier Origin Prediction

In Section 1.2.4 20 points with particularly high outlier scores were identified. Before the intricacies of isotope distribution and analysis were fully appreciated, these outliers would have readily been assumed to be due to migration or trade. Having built a model of the data in Chapter 5, we can now ask what plausible or even probable locations for these points might be. As the outliers are modeled over the full data, it is probable that there will be no location that corresponds to them exactly. But they may fit much better in one region than all the others. Or their relatively low fit in all components may suggest that they are indeed from an entirely different area.

6.2.1 Approach

To identify a likely spatial origin of a sample based on a spatially constrained GMM (like those introduced in Sections 5.4.3 and 5.4.4) the measurements of the point are again transformed into the spatial domain via their assignment probability to each component. The expected value of a point is given by

$$E(x) = \sum_{j=0}^{k} p_j(x)\mu_j$$

since p is a distribution. This is of course rather boring, because this is exactly the inverse of how the probability was calculated in the first place. However, by applying this formula not to the "normal" μ, but to the spatial model's $\mu^{\mathbb{C}}$, a spatial expectation can be calculated.

$$E^{\mathbb{C}}(x) = \sum_{j=0}^{k} p_j(x)\mu_j^{\mathbb{C}}$$

This allows us to calculate the expected spatial position of any given feature point and is in a sense the opposite application of the technique presented in Section 6.1.

6.2.2 Prediction

In this section we calculate the predicted spatial position of some data points from the human data set (see Section 1.2.2) and compare them with the spatial position of their recorded sample site. The first evaluation is based on the constrained GMM from Section 5.5.5. Figure 6.7 shows the sites where outliers were found (empty circle) and their

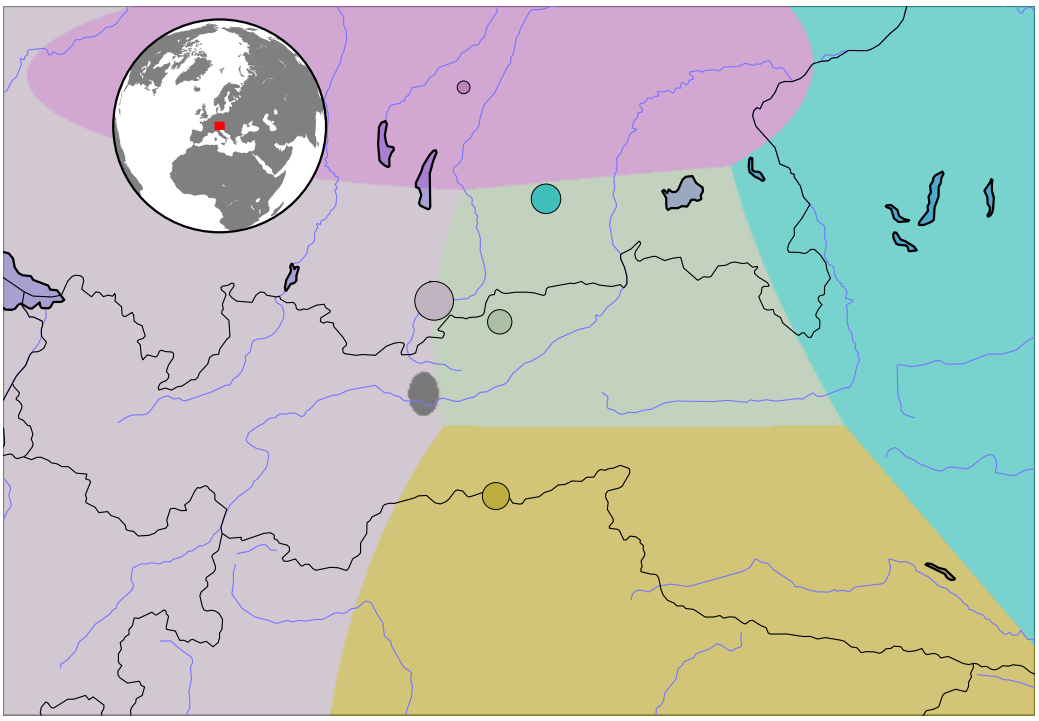


Figure 6.5: Generalized constrained EM result map.

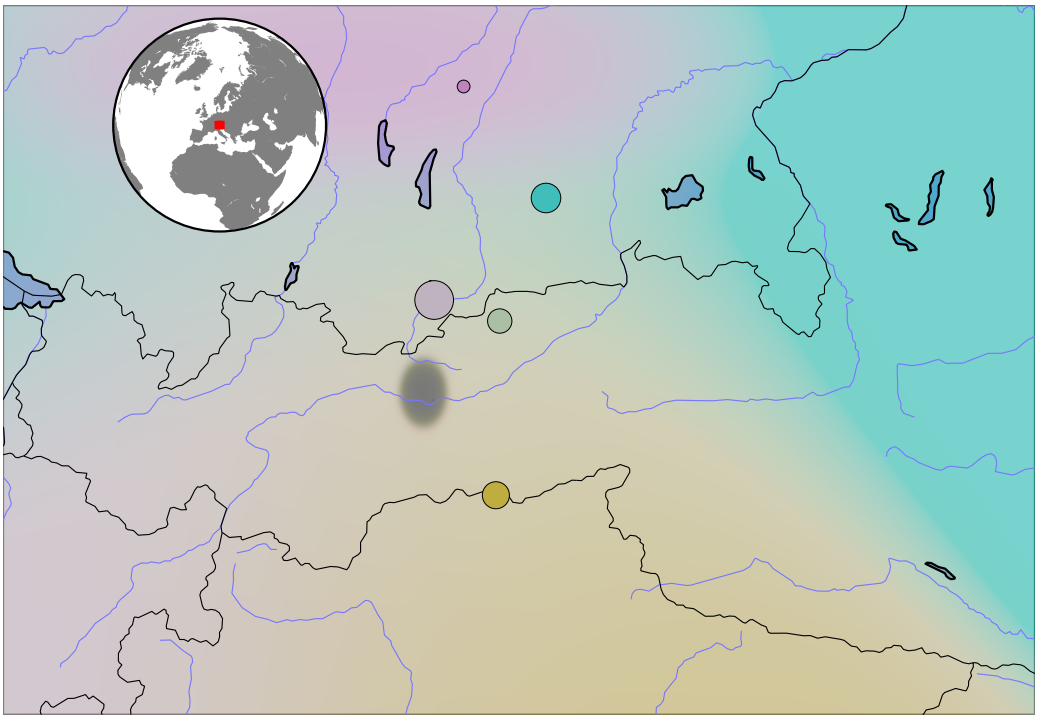


Figure 6.6: Generalized constrained EM result map (continuous projection).

predicted "real" place of origin (filled). Interestingly, most predicted locations are somewhat close to their sample sites. However, there are a few points that get predicted at a different location. These would be interesting samples for domain scientists to have another look at. There seems to be a stronger tendency for very southern points to be predicted at a position clearly north of their sample site. This tendency becomes more pronounced when looking at the full data's predicted locations. Figure 6.8 shows all sites and their samples' predicted "real" place of origin (filled). There is an obvious tendency to predict longitudinally mean values (which can be seen from the apparent line of samples stretching almost the entire length of the sample region). This is not necessarily a contradiction with domain knowledge, as it is expected that most variance in the data is latitudinal variation. However, the relative sparsity of predicted locations in the south may indicate a similar effect in latitudinal direction. While the model is still clearly predicting different locations, the sample density seems to have had an impact on the model.

When we instead look at the model based on the generalized constrained GMM from Section 5.5.5.1 Figure 6.9 results. Interestingly, many points are clearly assigned to a component, but not necessarily one that is close to their sample site. Some outliers get predominantly assigned to a single component and their predicted origin ends up very close to that component's spatial mean. However, there are a few points that are not assigned to any one component and end up somewhere in between sites. These points apparently do not correspond to any particular location. If the model is correct, these points may indicate foreign individuals. Figure 6.10 shows all sites and their samples' predicted "real" place of origin (filled). Obviously, many points' predicted origin does not fall neatly around the sites that they were found at, but the model is much more diverse than in the previous model. Some points are predicted away from the spatial centers, yet fall fairly close to each other. It would be interesting to defer these sites to domain scientists to find out whether there is a plausible explanation for their presence in the model.

6.3 Conclusion

In this chapter we saw some ways how the models generated in Chapter 5 can be applied by domain experts to identify interesting locations and samples. These techniques make a connection between the spatial and the feature domains that are only possible because of the explicit connection between the data and constraint spaces introduced by these techniques.

Interestingly, these applications of the models also reveal some characteristics of them. It appears that the constrained EM algorithm introduced in Section 5.4.3 tends to produce models whose spatial projection tends towards mean values. This behavior can be explained when the spatial influence is low. Then the fitted spatial model would be built over points that are from a wide range of spatial origins and their spatial mean would therefore be in the center of the distribution. However, there is some variation in latitudinal direction, indicating that indeed the model is expressing spatial coherence.

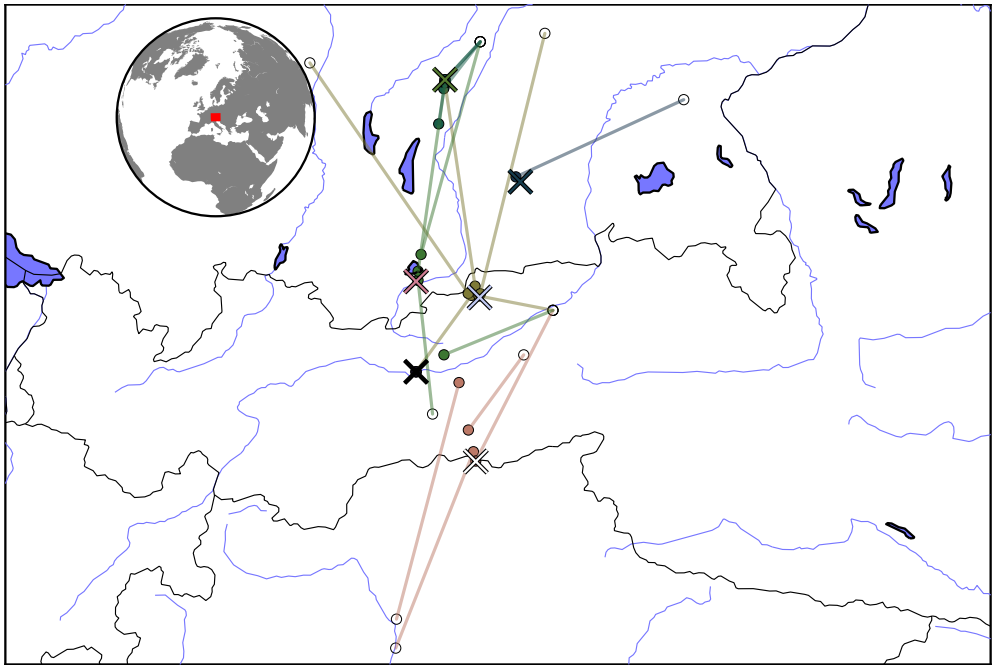


Figure 6.7: Predicted places of origin vs found location for global outliers points in the data set using the model based on the best performing constrained EM algorithm.

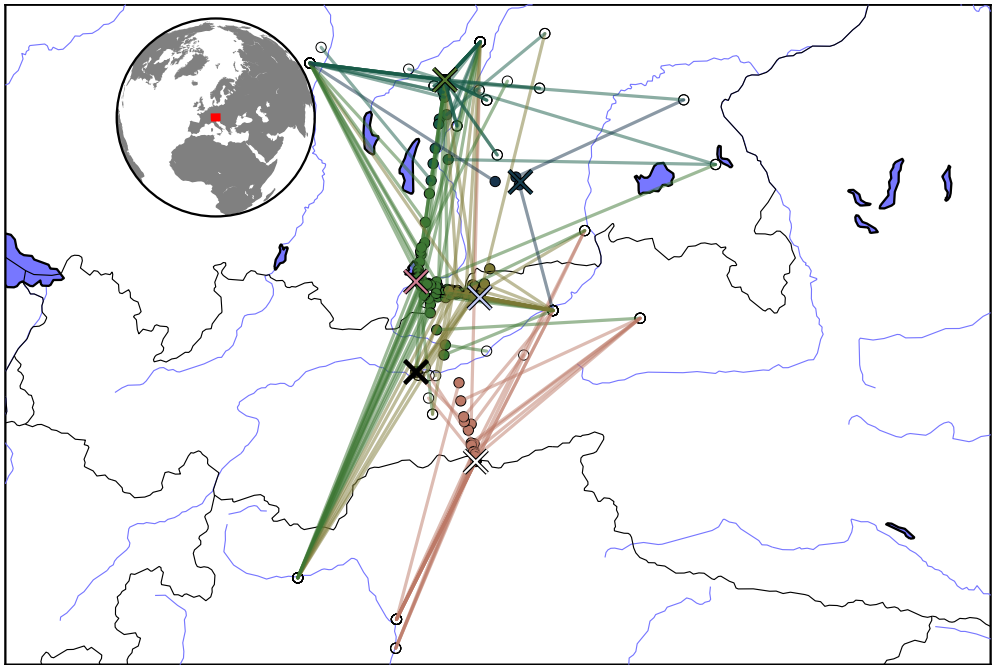


Figure 6.8: Predicted places of origin vs found location for all points in the data set using the model based on the best performing constrained EM algorithm.

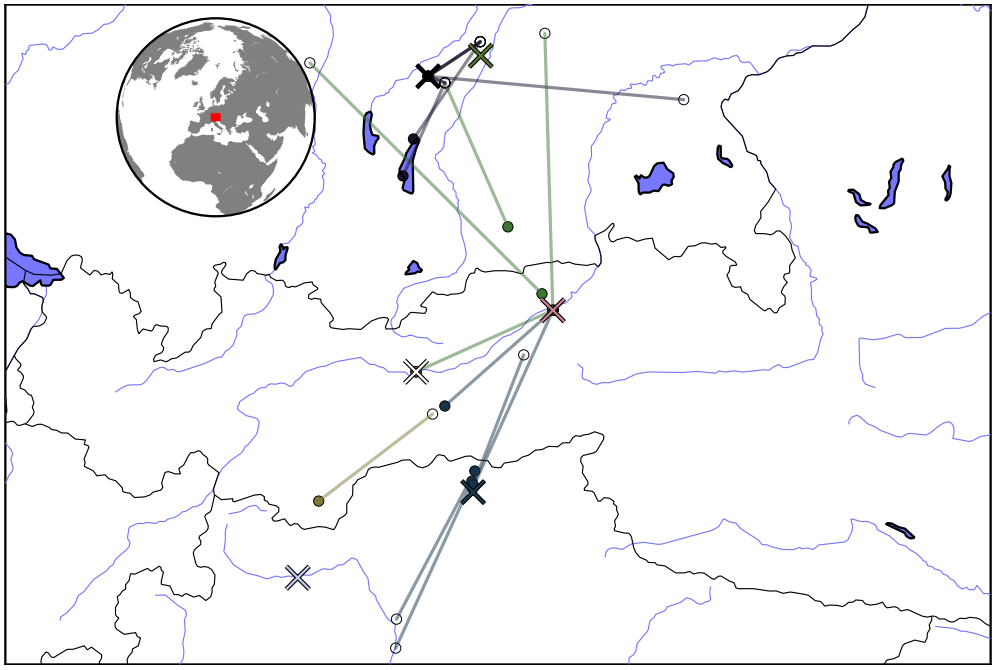


Figure 6.9: Predicted places of origin vs found location for global outliers using the model based on the best performing generalized constrained EM algorithm.

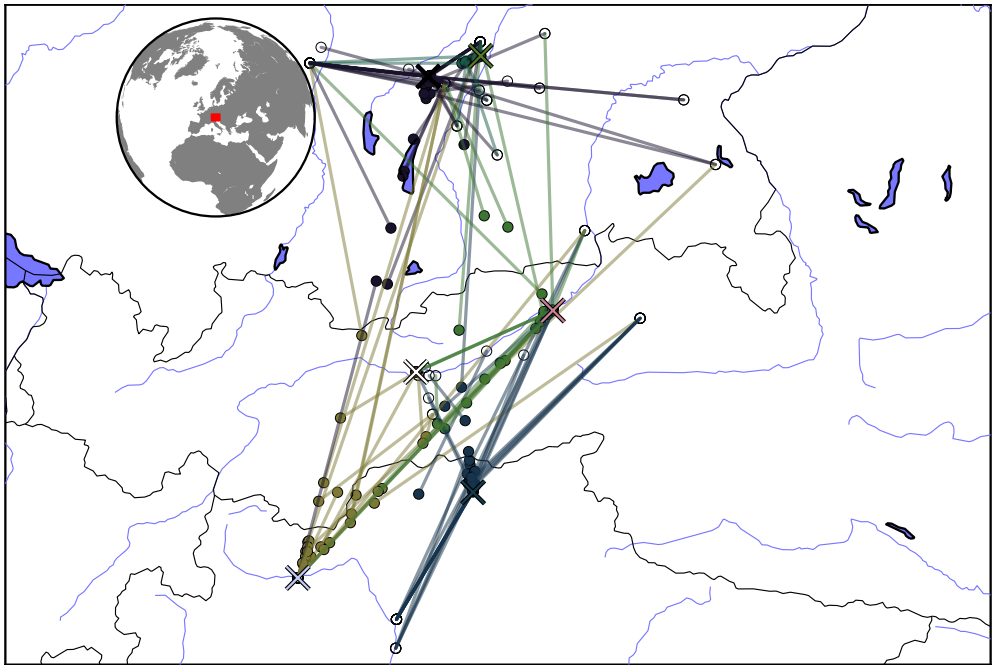


Figure 6.10: Predicted places of origin vs found location for all points in the data set using the model based on the best performing generalized constrained EM algorithm.

The generalized constrained EM algorithm (despite its mathematical deficiencies) generates more spatially diverse models and more realistically looking spatial predictions. This is a reassuring result, which encourages the further development of this technique.

Chapter 7

Outlook

Attribution

This chapter does not use any material from previous publications.

This thesis discussed using domain constraints to improve data analysis methods. This serves two purposes:

1. help data scientists to achieve desirable model properties and
2. make it easier for domain scientists to analyze their data.

The work presented here only scratches the surface of these goals. This section will discuss ways to widen the applicability of the notion of constraints and the presented methods in particular, as well as give some further ideas on how to improve the experience of working with them.

7.1 Powerful Constraints

An important step towards making the methods more universally applicable is a stronger notion of constraints. This would allow the application of the constraint concept to a wider range of problems.

The current implementation of constraints calculates the constraints internally, exposing the constraint mechanism via the ability to pass constraint data. Unfortunately, this limits the power of constraints to the notion implemented in each algorithm. To alleviate this limitation, the predicate or cost function to use could be supplied as a parameter instead (or in addition to) the constraint data. A case where this has been done is the interobject trajectory constraint approach presented in Chapter 4. Instead of supplying

constraint data, this framework expects as one of its building blocks a constraint function, which can be evaluated as a binary local predicate. However, as the evaluation of this approach has shown, an understanding of the implications of this constraint were required: in order to implement this efficiently via the mechanisms afforded by an index structure required the understanding that intersection of segment bounding boxes is a necessary condition for object proximity. While the rest of the application stayed true to the framework concept (made possible by the generality of the used algorithms), this illustrates the power of specialized approaches to make a problem solvable.

The notion of constraints and in particular the one used in the presented approaches, should be made more powerful and applicable to a wider range of problems. In order to achieve this, the challenge will be to establish which properties are required and which can be handled generally to produce both versatile and efficient solutions.

Another way to increase the power of constraints is to implement more general constraint types in the algorithms and allow the user to supply more than one set of constraint data.

In the following each method's potential for generalization will be discussed.

Feature evaluation (Chapter 3) The presented feature evaluation technique is a fairly simple approach to constrained analysis. The current implementation generates a reference clustering and compares it to clusterings on the investigated data. This technique can be broadened to any analysis based on the comparison of models or model effects. An obvious way to make the presented feature evaluation technique more versatile is to allow labels for each point to be specified instead of a reference data set. This would, for example, allow applying this technique to model selection without requiring a stronger notion of constraints. However, if the notion of constraints could be broadened, it could also compare models directly, e.g. through the Kullback-Leibler divergence [42].

Route databases (Chapter 4) The route database analysis has been presented as a framework with a notion of constraints that is reminiscent of optimization theory. This allows it to be very general. The two presented extensions (to continuous cost constraints and binary local predicates) use this framework to apply it to a wider range of problems. This design can be seen as a template for future algorithms. However, generality has some caveats, which have already been discussed above.

Spatial modeling: Monte Carlo (Section 5.4.2) A Monte Carlo approach is another example of a very general solution. As in the route database application, it uses some shortcuts to increase the chance of finding viable solutions. A challenge for future developments is to generalize these shortcuts in order to make them applicable to a wider range of problems.

Spatial modeling: constrained EM (Section 5.4.3) The constrained EM algorithm is mostly limited by the requirement to represent the constraint data as a Gaussian Mixture

model. This could be replaced by any global unary cost constraint, which allows itself to be optimized through an EM paradigm. To alleviate this limitation was the motivation behind the generalized constrained EM algorithm discussed next.

Spatial modeling: generalized constrained EM (Section 5.4.4) The generalized constrained EM algorithm supports the most general type of constraint supported through constraint data. This makes it very powerful, but did not allow optimizing the model in the most efficient way. The experimental evaluation seems promising, but in order to serve as a template for future algorithms this problem should be solved or at the very least be more thoroughly investigated. An even more general constraint would be to base the similarity used as constraints not on a static matrix, but have it be computed depending on the current model. This will make the model yet harder to optimize if at all possible.

When multiple approaches have been defined on a more general notion of constraints, a survey of the solutions might lead to generally applicable insights into constraint analysis, which could be consolidated in more advanced, more general approaches.

7.2 Specifying Constraints

Usability is one of the goals of constrained algorithms. The more powerful notion of constraints discussed above, will make the algorithms harder to use. Taken to its extreme, constraints can be any mathematically defined property. Even if it were possible to support any type of constraint, they would become hard to generate, defeating the purpose of constraints as a helpful tool.

A future improvement of the concept of constraints is to find ways to generate constraints more easily. One of the difficulties of defining constraints is to figure out what the desired properties even are. GMMbuilder (Section 5.4.1) addresses this problem by letting the user interactively experiment with different manifestations of a model. However, it does not attempt to understand the user's motivation. Instead it happily applies the resulting model as is.

A different approach at interactively developing constraints would be for the user to flag "wrong" data points. The analysis of this feedback should be reflected in a change to the constraints. A reevaluation of the model based on the new constraints could give the user feedback whether their constraints have been correctly interpreted. This kind of tool would probably need to be implemented (or at least adapted) for different kinds of tasks. To make this tool usable, the changes should be presented in such a way that it is transparent to the user what changed and why, which is a challenge for a complex model.

As discussed earlier, a generic way to express constraints is as binary cost functions. An approach to generate constraints through user interaction is to represent the perceived similarity between points and output a similarity matrix. This similarity matrix could serve as input to a fairly wide range of constrained analyses. If required, a more dynamic analysis could be supported by a similarity function generated through an application of similarity learning [88].

If successful, this approach might also be extended to output constraints to be used as inputs of a general optimization-based solver. Since the requirement to be usable limits the tool to a particular task, constraints, which specify the task, can be included in the output.

The current approach of supplying constraints as additional data solves this problem by making it very clear how the supplied data will be used. For sufficiently powerful algorithms, this may be an appropriate means to specify constraints. For example, labels are a very simple, yet very powerful, type of constraint data. The implied equality constraint is also simple to understand. This is the constraint formulation used in constrained clustering [86]. A weakness of the current implementations is that every point needs constraint data. This may not be a necessity for all approaches. For example, one of the design criteria of the EM variants presented in Sections 5.4.3 and 5.4.4 was that the original algorithm could be used for appropriate constraint data. If no constraint data is supplied for a point, the input could be interpreted as the kind of data that would make the constraint neutral at this point.

Chapter 8

Summary and Discussion

This thesis introduced the concept of constraints for data analysis by domain experts. Constraints are properties that the output of an analysis must satisfy to be considered appropriate by the user. This concept is related to constraint satisfaction problems from optimization theory, but takes a more hands-on approach in an attempt to ease working with constrained data for domain scientists without requiring a firm grasp of optimization theory.

Constraints are here specified as additional attributes passed to the analysis. The conversion of this data into solution properties lies with the specific analysis method. For the purpose of incorporating constraints into analysis approaches, constraints are defined as functions over the supplied constraint data. The data can either be the input data itself or additional data passed to the algorithm. Passing additional data is generally more useful, because analyzing the input data again has little potential to reveal more information. The constraint functions can be defined on single points or connect multiple points together.

The developed techniques were applied to a real-world data set of isotope measurements collected in an archaeological research project FOR 1670 of the German Science Foundation, which investigates archaeological sites in the Alps region of central Europe. The results of this application were studied to yield insights into the subject matter of the research project.

The thesis introduced several methods using constraints to improve their results. In a first method constraints are manifested from different attribute subsets in a clustering setting. This method uses a set of attributes to generate a clustering of input data and compare it to other clusterings based on a different subset of attributes. The constraint data are different representations of the same data. The approach investigates whether the structure extracted from the investigated representation is similar to one based on the reference representation. The structure of the data is represented by class labels (which are extracted from a different representation of the data) and compared it with a labeling extracted using the reference representation. The extracted labels are local unary predicate constraints, i.e. they occur at a specific data point, which either matches the reference or does not. This approach directly outputs the constraint score to allow the user to reason about a representation's significance versus another. It can be applied as a feature selection

algorithm by returning the representation with the best constraint score to optimize the result using constraints. In order to allow re-labeling the implementation considers pairs of points and evaluates whether they are in the same or in different clusters in both the reference and investigated labeling.

The second method considers route and trajectory databases. Routes are sequences of points in space, trajectories map points in time to locations. Constraint data for these data types can e.g. by information about each location. A first approach considers local unary continuous constraints, i.e. each location is given a single constraint score, which can be optimized. By attempting to shift each location by a fixed amount, the resulting route is optimized to be in a lower state of entropy. Another applications considers co-location of objects in the same place at the same time. This corresponds to a much more costly binary local predicate constraints, specifically the interaction between trajectories in the database. For many applications only one object can be in the same place at the same time and this constellation in a data set represents an error. In order to minimize the number of constraint violations, trajectories are modified to remove the local constraint violation. Due to the nature of trajectories, this can cause other violations. Various approaches are applied to the data, which approximate a global optimal solution in order to reduce the number of issues.

The final area of application is spatial modeling of data. Here we saw how the same input data can be interpreted as very different constraints. The constraint data in this approach is spatial information, which is used to build a more appropriate data model. In one formulation (Section 5.4.2 the constraints are global, n-ary (all points), predicate constraints. This is reflected in a reachability graph over the whole data (n-ary), which is evaluated for spatial coherence (global predicate) In a second formulation (Section 5.4.3 the constraints are global (a change in the model affects the valuation of all points), but unary, costs (not predicates). A model of the spatial distribution is evaluated for each point's fit (unary continuous), but a change in the model changes that valuation for all points (global). In another formulation the constraints are considered global binary cost constraints. There each pair of points' distance with one another is taken into account (binary cost), but the final cost outcome is determined by the model state, which changes the weights for each pair (global). The output of each of these approaches was a model over the input data which corresponds to a spatially coherent solution.

In Chapter 6 the same constraint data was used to apply the previously generated models. The used constraint was that the feature distribution follows a spatial distribution. The spatial information associated with the input data was used to build a spatial distribution corresponding to the feature distribution. The supplied data's predicted probabilities were calculated based on the supplied model and the resulting weights were used to fit a model over the spatial domain. This spatial model was then used to

1. predict the origin of a feature point and

2. predict the feature value at a given location.

The first task was then used to predict the spatial origin of a sample if the origin is either unknown or suspect. The second approach was applied to outlier points recorded in Chapter 1 to see whether they are predicted to be from a different location.

This work addressed the application of constraint data to various data analysis tasks. Constraints can be used to generate models, which are more appropriate to an application than the ones that would be generated by an unmodified algorithm. They can also help to build valid models on small data sets and to help domain experts build complex models that would otherwise require a data analysis expert. The application of the presented methods was demonstrated on a real-world data set of archaeo-biological spatial measurements.

Acknowledgments

This work was supported by the German Research Foundation within the interdisciplinary DFG Research Group FOR 1670 “Transalpine mobility and cultural transfer”. Details at: http://www.en.for1670-transalpine.uni-muenchen.de

Bibliography

[1] E. Achtert, H.-P. Kriegel, E. Schubert, and A. Zimek. Interactive data mining with 3D-parallel-coordinate-trees. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 1009–1012. ACM, 2013.

[2] H. Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[3] S. H. Ambrose and J. Krigbaum. Bone chemistry and bioarchaeology. *Journal of Anthropological Archaeology*, 22(3):193–199, 2003. ISSN: 0278-4165. DOI: http://dx.doi.org/10.1016/S0278-4165(03)00033-3. URL: http://www.sciencedirect.com/science/article/pii/S0278416503000333. Bone Chemistry and Bioarchaeology.

[4] M. Arenas, L. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '99, pages 68–79, Philadelphia, Pennsylvania, USA, 1999.

[5] L. R. Bahl, F. Jelinek, and R. L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, (2):179–190, 1983.

[6] J. K. Baker. Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 65(S1):S132–S132, 1979.

[7] J. A. Bilmes et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models, 1998.

[8] P. Bohannon, W. Fan, M. Flaster, and R. Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Baltimore, ML*, pages 143–154, 2005.

[9] D. Bolin, J. Wallin, and F. Lindgren. *Multivariate latent Gaussian random field mixture models*. Department of Mathematical Sciences, Division of Mathematical Statistics, Chalmers University of Technology and University of Gothenburg, 2014.

[10] C. L. Borgman. *Scholarship in the digital age: Information, infrastructure, and the Internet*. MIT press, 2010.

[11] G. J. Bowen. Isoscapes: spatial pattern in isotopic biogeochemistry. *Annual Review of Earth and Planetary Sciences*, 38:161–187, 2010.

[12] T. Brinkhoff, H. Kriegel, and B. Seeger. Efficient processing of spatial joins using r-trees. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993.* Pages 237–246, 1993. DOI: 10.1145/170035.170075. URL: http://doi.acm.org/10.1145/170035.170075.

[13] R. Cheng, T. Emrich, H. Kriegel, N. Mamoulis, M. Renz, G. Trajcevski, and A. Züfle. Managing uncertainty in spatial and spatio-temporal data. In *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*, pages 1302–1305, 2014.

[14] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Querying imprecise data in moving object environments. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1112–1127, 2004.

[15] M. Dash, H. Liu, and J. Yao. Dimensionality reduction of unsupervised data. In *Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on*, pages 532–539. IEEE, 1997.

[16] T. H. Davenport and D. Patil. Data scientist. *Harvard business review*, 90(5):70–76, 2012.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*:1–38, 1977.

[18] J. G. Dy and C. E. Brodley. Feature selection for unsupervised learning. *Journal of machine learning research*, 5(Aug):845–889, 2004.

[19] E. Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, 1990.

[20] T. Emrich, H.-P. Kriegel, N. Mamoulis, M. Renz, and A. Züfle. Indexing uncertain spatio-temporal data. In *Proceedings of the 21st ACM Conference on Information and Knowledge Management (CIKM), Maui, HI*, pages 395–404, 2012.

[21] T. Emrich, H.-P. Kriegel, M. Mauder, M. Renz, G. Trajcevski, and A. Züfle. Minimal spatio-temporal database repairs. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 492–495. ACM, 2013.

[22] T. Gindele, S. Brechtel, and R. Dillmann. Learning driver behavior models from traffic observations for decision making and planning. *IEEE Intelligent Transportation Systems Magazine*, 7(1):69–79, 2015. DOI: 10.1109/MITS.2014.2357038. URL: http://dx.doi.org/10.1109/MITS.2014.2357038.

[23] G. Gottlob, N. Leone, and F. Scarcello. A comparison of structural CSP decomposition methods. *Artificial Intelligence*, 124(2):243–282, 2000.

[24] G. Grupe, A. Grigat, and G. C. McGlynn, editors. *Across the Alps in Prehistory. Isotopic Mapping of the Brenner Passage in Bioarchaeology*. Springer International Publishing, Mar. 2017, to appear.

[25] G. Grupe, M. Grünewald, M. Gschwind, S. Hölzl, B. Kocsis, P. Kröger, A. Lang, M. Mauder, C. Mayr, G. C. McGlynn, C. Metzner-Nebelsick, E. Ntoutsi, J. Peters, M. Renz, S. Reuß, W. W. Schmahl, F. Söllner, C. S. Sommer, B. Steidl, A. Toncala, Trixl, and D. Wycisk. Networking in bioarchaeology: the example of the DFG research group FOR 1670 "Transalpine mobility and culture transfer." In G. Grupe, G. McGlynn, and J. Peters, editors, *Bioarchaeology beyond Osteology. Documenta Archaeobiologiae 12*, pages 13–51. Verlag Marie Leidorf, 2015.

[26] G. Grupe and G. C. McGlynn. *Isotopic landscapes in bioarchaeology*. Springer, 2016.

[27] G. Grupe, T. D. Price, P. Schröter, F. Söllner, C. M. Johnson, and B. L. Beard. Mobility of bell beaker people revealed by strontium isotope ratios of tooth and bone: a study of southern bavarian skeletal remains. *Applied Geochemistry*, 12(4):517–525, 1997.

[28] D. Guo. Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP). *International Journal of Geographical Information Science*, 22(7):801–823, 2008.

[29] R. H. Güting and M. Schneider. *Moving Objects Databases*. Morgan Kaufmann, 2005.

[30] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3:1157–1182, Mar. 2003.

[31] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009. ISSN: 1931-0145. DOI: 10.1145/1656274.1656278. URL: http://doi.acm.org/10.1145/1656274.1656278.

[32] J. Han, M. Kamber, and J. Pei. *Data mining, southeast asia edition: Concepts and techniques*. Morgan kaufmann, 2006.

[33] D. Hawkins. *Identification of Outliers*, volume 1. Chapman and Hall, 1980.

[34] T. Hey, S. Tansley, K. M. Tolle, et al. *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA, 2009.

[35] M. Hofmann and R. Klinkenberg. *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman & Hall/CRC, 2013. ISBN: 1482205491.

[36] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[37] P. Jeavons, D. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM (JACM)*, 44(4):527–548, 1997.

[38] F. Keinosuke. Introduction to statistical pattern recognition. *Academic Press Inc*, 1990.

[39] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, et al. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[40] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[41] B. Kuijpers and W. Othman. Trajectory databases: data models, uncertainty and complete query languages. In *JCSS*, pages 538–560, 2010.

[42] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*:79–86, 1951.

[43] J. A. Kupfer, P. Gao, and D. Guo. Regionalization of forest pattern metrics for the continental united states using contiguity constrained clustering and partitioning. *Ecological Informatics*, 9:11–18, 2012.

[44] K. Lari and S. J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer speech & language*, 4(1):35–56, 1990.

[45] H. Li, K. Zhang, and T. Jiang. The regularized EM algorithm. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 807. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[46] F. Lindgren, H. Rue, and J. Lindström. An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.

[47] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers. Big data: the next frontier for innovation, competition, and productivity. *McKinsey & Company Report*, May 2009.

[48] G. Matheron. Krigeage d'un panneau rectangulaire par sa périphérie. *Note géostatistique*, 28, 1960.

[49] M. Mauder, Y. Bobkova, and E. Ntoutsi. GMMbuilder – user-driven discovery of clustering structure for bioarchaeology. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 8–11. Springer, 2016.

[50] M. Mauder, E. Ntoutsi, and P. Kröger. Influence of oxygen isotope ratio on classification. Technical report, FOR1670: Transalpine mobility and cultural transfer, 2014.

[51] M. Mauder, E. Ntoutsi, P. Kröger, and G. Grupe. Data mining for isotopic mapping of bioarchaeological finds in a central European Alpine passage. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*, page 34. ACM, 2015.

[52] M. Mauder, E. Ntoutsi, P. Kröger, and H.-P. Kriegel. The isotopic fingerprint: new methods of data mining and similarity search. In G. Grupe, A. Grigat, and G. C. McGlynn, editors, *Across the Alps in Prehistory: Isotopic Mapping of the Brenner Passage by Bioarchaeology*, pages 105–125. Springer, 2017.

[53] M. Mauder, E. Ntoutsi, P. Kröger, C. Mayr, G. Grupe, A. Toncala, and S. Hölzl. Applying data mining methods for the analysis of stable isotope data in bioarchaeology. In *2016 IEEE 12th International Conference on eScience*, 2016.

[54] M. Mauder, M. Reisinger, T. Emrich, A. Züfle, M. Renz, G. Trajcevski, and R. Tamassia. Minimal spatio-temporal database repairs. In *Advances in Spatial and Temporal Databases*, pages 255–273. Springer International Publishing, 2015.

[55] W. Meier-Augenstein and H. Kemp. Stable isotope analysis: general principles and limitations. *Wiley Encyclopedia of Forensic Science*, 2012.

[56] W. Meier-Augenstein and H. F. Kemp. *Stable isotope analysis: general principles and limitations*. In *Wiley Encyclopedia of Forensic Science*. John Wiley and Sons, Ltd, 2009. ISBN: 9780470061589. DOI: 10.1002/9780470061589.fsa1041. URL: http://dx.doi.org/10.1002/9780470061589.fsa1041.

[57] G. W. Milligan and M. C. Cooper. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research*, 21(4):441–458, 1986.

[58] Mobile subscribers 2014. Source: ITU World Telecommunication/ICT Indicators database. (http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2014-e.pdf).

[59] H. Mokhtar and J. Su. Universal trajectory queries for moving object databases. In *Mobile Data Management (MDM)*, pages 133–144, 2004.

[60] D. Murray. *Tableau Your Data!: Fast and Easy Visual Analysis with Tableau Software*. Wiley Publishing, 1st edition, 2013. ISBN: 1118612043.

[61] F. Parisi and J. Grant. Repairs and consistent answers for inconsistent probabilistic spatio-temporal databases. In *Scalable Uncertainty Management*, pages 265–279. Springer, 2014.

[62] A. Parker, V. Subrahmanian, and J. Grant. A logical formulation of probabilistic spatial databases. *Knowledge and Data Engineering, IEEE Transactions on*, 19(11):1541–1556, 2007.

[63] J. Pearson and P. G. Jeavons. A survey of tractable constraint satisfaction problems. Technical report, Technical Report CSD-TR-97-15, Royal Holloway, University of London, 1997.

[64] D. Pfoser, C. S. Jensen, and Y. Theodoridis. Novel approaches to the indexing of moving object trajectories. In *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB), Cairo, Egypt*, pages 396–406, 2000.

[65] D. Pfoser and C. S. Jensen. Capturing the uncertainty of moving-object representations. In *Proceedings of the 6th International Symposium on Large Spatial Databases (SSD), Hong-Kong*, pages 111–132, 1999.

[66] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[67] K. Rozier and M. Vardi. LTL satisfiability checking. In *Automated Technology for Verification and Analysis*, 2011.

[68] H. Rue, S. Martino, and N. Chopin. Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, 71(2):319–392, 2009.

[69] S. Saltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez. Indexing the positions of continuously moving objects. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Dallas, TX*, pages 331–342, 2000.

[70] J. Schiller and A. Voisard. *Location-Based Services*. The Morgan Kaufmann Series in Data Management Systems, 2004.

[71] C. Schöch. Big? Smart? Clean? Messy? Data in the humanities. *Journal of Digital Humanities*, 2(3):2–13, 2013.

[72] G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[73] B. Seeger, B. König-Ries, and T. Härder. Editorial. *Datenbank-Spektrum*, 16(3):191–194, 2016. ISSN: 1610-1995. DOI: 10.1007/s13222-016-0234-5. URL: http://dx.doi.org/10.1007/s13222-016-0234-5.

[74] M. L. Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012.

[75] Y. Tao, D. Papadias, and J. Sun. The TPR*Tree: an optimized spatio-temporal access method for predictive queries. In *Proceedings of the 29th International Conference on Very Large Data Bases (VLDB), Berlin, Germany*, pages 790–801, 2003.

[76] Y. Tao, C. Faloutsos, D. Papadias, and B. Liu. Prediction and indexing of moving objects with unknown motion patterns. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Paris, France*, pages 611–622, 2004.

[77] W. R. Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*:234–240, 1970.

[78] A. Toncala, F. Söllner, C. Mayr, S. Hölzl, K. Heck, D. Wycisk, and G. Grupe. Isotopic map of the Inn-Eisack-Adige-Brenner passage and its application to prehistoric human cremations. In G. Grupe, A. Grigat, and G. C. McGlynn, editors, to appear. Springer International Publishing, 2017.

[79] G. Trajcevski, A. N. Choudhary, O. Wolfson, L. Ye, and G. Li. Uncertain range queries for necklaces. In *11th International Conference on Mobile Data Management (MDM 2010), Kansas City, Missouri*, pages 199–208, 2010.

[80] G. Trajcevski, R. Tamassia, H. Ding, P. Scheuermann, and I. F. Cruz. Continuous probabilistic nearest-neighbor queries for uncertain trajectories. In *Proceedings of the 12th International Conference on Extending Database Technology (EDBT), Saint-Petersburg, Russia*, pages 874–885, 2009.

[81] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain. Managing uncertainty in moving objects databases. *ACM Transactions on Database Systems (TODS)*, 29(3):463–507, 2004.

[82] G. Trajcevski, O. Wolfson, F. Zhang, and S. Chamberlain. The geometry of uncertainty in moving objects databases. In *Proceedings of the 8th International Conference on Extending Database Technology (EDBT), Prague, Czech Republic*, pages 233–250, 2002.

[83] N. Van der Merwe, J. Lee-Thorp, J. Thackeray, A. Hall-Martin, F. Kruger, H. Coetzee, R. Bell, and M. Lindeque. Source-area determination of elephant ivory by isotopic analysis. *Nature*, 346(6286):744–746, 1990.

[84] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1073–1080. ACM, 2009.

[85] J. Vogel, B. Eglington, and J. Auret. Isotope fingerprints in elephant bone and ivory. *Nature*, 346(6286):747–749, 1990.

[86] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al. Constrained k-means clustering with background knowledge. In *International Conference on Machine Learning*, volume 1, pages 577–584, 2001.

[87] J. Wijsen. Database repairing using updates. *ACM Transactions on Database Systems (TODS)*, 30(3), 2005.

[88] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, volume 15 of number 505-512, page 12, 2002.

[89] R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–437. IEEE, 2004.

[90] B. Zhang and G. Trajcevski. The tale of (fusing) two uncertainties. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Dallas/Fort Worth, TX, USA, November 4-7, 2014*, pages 521–524, 2014.

[91] X. Zhu. Semi-supervised learning literature survey, 2005.

[92] S. Ziegler and D. Jacob. Development of a spatial reference database for ivory. *TRAFFIC bulletin*, 23(1), Dec. 2010.

Publications Incorporated in this Thesis

[21] T. Emrich, H.-P. Kriegel, M. Mauder, M. Renz, G. Trajcevski, and A. Züfle. Minimal spatio-temporal database repairs. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 492–495. ACM, 2013.

[25] G. Grupe, M. Grünewald, M. Gschwind, S. Hölzl, B. Kocsis, P. Kröger, A. Lang, M. Mauder, C. Mayr, G. C. McGlynn, C. Metzner-Nebelsick, E. Ntoutsi, J. Peters, M. Renz, S. Reuß, W. W. Schmahl, F. Söllner, C. S. Sommer, B. Steidl, A. Toncala, Trixl, and D. Wycisk. Networking in bioarchaeology: the example of the DFG research group FOR 1670 "Transalpine mobility and culture transfer." In G. Grupe, G. McGlynn, and J. Peters, editors, *Bioarchaeology beyond Osteology. Documenta Archaeobiologiae 12*, pages 13–51. Verlag Marie Leidorf, 2015.

[49] M. Mauder, Y. Bobkova, and E. Ntoutsi. GMMbuilder – user-driven discovery of clustering structure for bioarchaeology. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 8–11. Springer, 2016.

[50] M. Mauder, E. Ntoutsi, and P. Kröger. Influence of oxygen isotope ratio on classification. Technical report, FOR1670: Transalpine mobility and cultural transfer, 2014.

[51] M. Mauder, E. Ntoutsi, P. Kröger, and G. Grupe. Data mining for isotopic mapping of bioarchaeological finds in a central European Alpine passage. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*, page 34. ACM, 2015.

[52] M. Mauder, E. Ntoutsi, P. Kröger, and H.-P. Kriegel. The isotopic fingerprint: new methods of data mining and similarity search. In G. Grupe, A. Grigat, and G. C. McGlynn, editors, *Across the Alps in Prehistory: Isotopic Mapping of the Brenner Passage by Bioarchaeology*, pages 105–125. Springer, 2017.

[53] M. Mauder, E. Ntoutsi, P. Kröger, C. Mayr, G. Grupe, A. Toncala, and S. Hölzl. Applying data mining methods for the analysis of stable isotope data in bioarchaeology. In *2016 IEEE 12th International Conference on eScience*, 2016.

[54] M. Mauder, M. Reisinger, T. Emrich, A. Züfle, M. Renz, G. Trajcevski, and R. Tamassia. Minimal spatio-temporal database repairs. In *Advances in Spatial and Temporal Databases*, pages 255–273. Springer International Publishing, 2015.