Multidimensional Clustering for Spatio-Temporal Data and its Application in Climate Research

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

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

genehmigte
Dissertation
von
Benjamin Ertl

Tag der mündlichen Prüfung:	03.12.2021
1. Referent:	Prof. Dr. Achim Streit
2. Referent:	Prof. Dr. Peter Braesicke

Erklärung zur selbständigen Anfertigung der Dissertationsschrift

Hiermit erkläre ich, dass ich die Dissertationsschrift mit dem Titel

Multidimensional Clustering for Spatio-Temporal Data and its Application in Climate Research

selbständig angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Regeln zur Sicherung guter wissenschaftlicher Praxis am Karlsruher Institut für Technologie (KIT) beachtet habe.

Ort, Datum Benjamin Ertl

To my family

Acknowledgements

I would like to thank Prof. Dr. Achim Streit for supervising this thesis and Prof. Dr. Peter Braesicke for co-supervising this thesis. Thank you both for your continuous support, advice, guidance and the opportunity to pursue the doctoral degree.

Also, this thesis would not have been possible without the support and advice of Dr. Jörg Meyer and Dr. Matthias Schneider, their encouragement, discussions, advice and proofreading of this thesis. Thank you so much!

I would also like to thank all friends, colleagues, and fellow doctoral researchers at IMK and SCC, especially Dr. Uğur Çayoğlu, for accepting to proofread this thesis and always been open for discussions within our research group.

Special thanks in no particular order to Marcus Strobl, Elnaz Azmi, Uros Stevanovic, Diana Gudu, Marcus Hardt, Mehmet Soysal, Eileen Kühn, Christopher Diekmann, Darko Dubravica, Marleen Braun, Amelie Röhling, Thomas Blumenstock, Frank Hase, Qiansi Tu, Farah Khosrawi, Jochen Groß, Carlos Alberti and everyone I forgot to mention here who have accompanied me over the years.

Last but not least, I would like to thank my family that has been growing since I started this thesis, for their unconditional support and encouragement throughout this endeavour.

Abstract

Through the increasing amount of high dimensional data that is available today through new technologies, higher processing power, bigger storage capacities, and data-driven research, the tasks of mining and analysing datasets have become more challenging. Especially advances in sensor and measurement technologies translate into an increasing amount of spatio-temporal data available for researchers across a wide variety of disciplines. Spatio-temporal data exhibit observations across space and time, for example, gathered by large sensor networks or satellites that provide remote sensing data or satellite imagery data. Finding clusters in such datasets can reveal interesting patterns and dependencies often caused by complex correlations. However, traditional full-space clustering algorithms suffer the curse of dimensionality. For example, points tend to become equidistant from one another as the dimensionality increases. Current subspace clustering and correlation clustering approaches overcome these issues but still face challenges when data points have complex relations or clusters in the dataset are not compact or clearly separable. Moreover, current methods lack the exploitation of available a priori knowledge that can improve the clustering results.

This thesis presents a novel data-driven approach of analysing clusters in spatio-temporal data that is based on the Gaussian mixture model in the area of unsupervised clustering, as well as a novel clustering algorithm based on the original DBSCAN algorithm, namely CoExDBSCAN, in the are of semi-supervised clustering. While the data-driven approach is best suited to extract cluster properties that can be analysed for changes over space and time, the semi-supervised clustering algorithm is able to align the outcome of the cluster analysis to a priori knowledge about the data.

Finally, this thesis presents the utilization of the proposed CoExDBSCAN algorithm for multivariate time-series. Especially recurring subsequences in streams of multiple measurements that can be organised as multivariate time-series, can be interpreted as recurring events or actions. These recurring events can be used to discover repeating patterns, understanding trends, detect anomalies and in general better interpret large and high-dimensional datasets. By utilizing the proposed CoExDBSCAN algorithm for such time-series data and constraining the cluster extension to the correlation

of time point values, clusters of segments with similar correlations can be identified. This novel semi-supervised approach for subsequence time-series clustering follows the pairwise semi-supervision approach and extends the concept to cluster-wide constraints. This approach has been demonstrated for semi-supervised time point clustering for multivariate time-series in general and as a semi-supervised approach for trajectory segmentation to identify different moisture processes in the atmosphere in particular.

The main field of application for the proposed clustering methods in this thesis is within the domain of climate research. The main interest in this domain has been to analyse pair distributions of water vapour (H_2O) and its isotopologue (HDO), which can be associated with atmospheric moisture processes. Identifying such processes is an important scientific task to infer the dynamics of cloud-circulation systems. Investigating the atmosphere from a cloud-circulation system perspective is essential to address the significant uncertainty of climate predictions. However, the application of the proposed clustering methods is not limited to a specific domain. The results of this thesis are beneficial to the analysis of spatio-temporal data in general.

Zusammenfassung

Durch die zunehmende Menge an hochdimensionalen Daten die heutzutage durch neue Technologien, höhere Rechenleistung, größere Speicherkapazitäten und datengetriebener Forschung verfügbar ist, sind die Aufgaben des Data-Minings und der Analyse von Datensätzen anspruchsvoller geworden. Vor allem Fortschritte in der Sensor- und Messtechnik führen zu einer steigenden Anzahl von raum-zeitlichen Daten die Forschern aus einer Vielzahl von Disziplinen zur Verfügung stehen. Raum-zeitliche Daten zeigen Beobachtungen über Raum und Zeit, zum Beispiel gesammelt von großen Sensornetzen oder Satelliten die Fernerkundungs- oder Satellitenbilddaten bereitstellen. Das Auffinden von Clustern in solchen Datensätzen kann interessante Muster und Abhängigkeiten aufdecken die oft durch komplexe Zusammenhänge verursacht sind. Existierende Clustering-Algorithmen die alle Attribute oder Attributkombinationen berücksichtigen leiden jedoch unter dem Fluch der Dimensionalität. Zum Beispiel neigen Punkte dazu äquidistant zueinander zu werden wenn die Dimensionalität zunimmt. Aktuelle Ansätze in den Bereichen der Unterraum-Clusteranalyse und Korrelations-Clusteranalyse überwinden diese Probleme, stehen aber immer noch vor Herausforderungen wenn Datenpunkte komplexe Beziehungen haben oder die Cluster im Datensatz nicht kompakt oder klar trennbar sind. Darüber hinaus fehlt es aktuellen Ansätzen an der Nutzung von verfügbarem a priori Wissen das die Clustering Ergebnisse verbessern kann.

Diese Dissertation präsentiert einen neuartigen datengetriebenen Ansatz zur Analyse von Clustern in raum-zeitlichen Daten der basierend auf dem Gaußschen Mischungsmodell im Bereich des unüberwachten Clusterings angesiedelt ist, sowie einen neuartigen Clustering-Algorithmus, CoExDBSCAN basierend auf dem ursprünglichen DBSCAN Algorithmus, der im Bereich des halbüberwachten Clusterings angesiedelt ist. Während der datengetriebene Ansatz am besten geeignet ist um Clustereigenschaften zu extrahieren die auf Änderungen über Raum und Zeit analysiert werden können, ist der halbüberwachte Clustering-Algorithmus in der Lage das Ergebnis der Clusteranalyse mit a priori Wissen über die Daten zu vereinbaren.

Abschließend präsentiert diese Arbeit die Verwendung des neu entwickelten CoExDBSCAN Algorithmus für multivariate Zeitfolgen. Besonders wiederkehrende Teilsequenzen in kontinuierlichen Mehrfachmessungen die als multivariate Zeitreihen organisiert werden können, können als wiederkehrende Ereignisse oder Aktionen interpretiert werden. Diese wiederkehrende Ereignisse können verwendet werden um wiederholende Muster zu entdecken, Trends zu verstehen, Anomalien zu erkennen und im Allgemeinen große und hochdimensionale Datensätze besser zu interpretieren. Durch Verwendung des vorgeschlagenen CoExDBSCAN Algorithmus für solche Zeitreihendaten und der Einschränkung der Cluster-Expansion anhand der Korrelation von Zeitpunktwerten können Cluster von Segmenten mit ähnlichen Korrelationen identifiziert werden. Dieser neuartige halbüberwachte Ansatz für das Clustering von Zeitreihen in Teilsequenzen folgt dem paarweisen halbüberwachten Ansatz und erweitert das Konzept auf clusterweite Einschränkungen. Diese Methode wird demonstriert anhand des halbüberwachten Zeitpunkt-Clustering für multivariate Zeitreihen im Allgemeinen und als halbüberwachter Ansatz zur Trajektoriensegmentierung für die Identifizierung verschiedener Feuchtigkeitsprozesse in der Atmosphäre im Besonderen.

Das Hauptanwendungsgebiet der in dieser Arbeit vorgeschlagenen Clustering-Methoden liegt im Bereich der Klimaforschung. Das Hauptinteresse in diesem Bereich ist die Analyse von Paarverteilungen von Wasserdampf (H_2O) und seinem Isotopolog (HDO), die mit atmosphärischen Feuchtigkeitsprozessen in Verbindung gebracht werden können. Die Identifizierung solcher Prozesse ist eine wichtige wissenschaftliche Aufgabe um die Dynamik der Wolkenzirkulation ableiten zu können. Die Atmosphäre aus der Perspektive eines Wolkenzirkulationssystems zu untersuchen ist wichtig um die erhebliche Unsicherheit von Klimavorhersagen zu verbessern. Die Anwendung der vorgeschlagenen Clustering-Methoden ist jedoch nicht auf eine bestimmte Domäne beschränkt. Die Ergebnisse dieser Arbeit verbessern vielmehr die Analyse von raum-zeitlichen Daten im Allgemeinen.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

4C Computing Correlation Connected Clusters.

ACC Cluster Accuracy.

ARI Adjusted Rand Index.

BIC Bayesian Information Criterion.

CASH Clustering in Arbitrary Subspaces based on the Hough transform.

COSMO COnsortium for Small-scale MOdelling.

CSR Complete Spatial Randomness.

CVQE Constrained Vector Quantization Error.

DTW Dynamic Time Warping.

ELKI Environment for deveLoping KDD-applications supported by Index-structures.

EM Expectation–maximization algorithm.

EUMETSAT European Organisation for the Exploitation of Meteorological Satellites.

GMM Gaussian Mixture Model.

HMRF Hidden Markov Random Field.

hPa Hectopascal.

IASI Infrared Atmospheric Sounding Interferometer.

IQR Interquartile range.

KDIR International Conference on Knowledge Discovery and Information Retrieval.

LAGRANTO Lagrangian Analysis Tool.

LIBRAS LÍngua BRAsileira de Sinais.

MIT Massachusetts Institute of Technology.

MUSICA MUlti-platform remote Sensing of Isotopologues for investigating the Cycle of Atmospheric water.

OPTICS Ordering Points To Identify the Clustering Structure.

ORCLUS Arbitrarily ORiented projected CLUSter generation.

PAM Partition Around Medoids.

ppmv parts per million by volume.

ROI Region of Interest.

SMOW Standard Mean Ocean Water.

TICC Toeplitz Inverse Covariance-based Clustering.

TRACLUS TRAjectory CLUStering.

UCI University of California, Irvine.

Glossary

active-semi-supervised-clustering Python module that provides active semi-supervised clustering algorithms for the scikit-learn machine learning library.

CK-means Constrained K-means algorithm.

CoExDBSCAN Density-based clustering with constrained expansion algorithm.

COSMO-iso Isotope-enabled limited-area COSMO model.

DBSCAN Density-based algorithm for discovering clusters in large spatial databases with noise.

HDBSCAN Density-based clustering algorithm based on hierarchical density estimate.

Java Java programming language.

k-means K-means clustering algorithm.

k-medoids K-medoids clustering algorithm.

PCK-means Pairwise constrained K-means algorithm.

Python Python programming language.

scikit-learn Machine learning library for the Python programming language.

statsmodels Python module that provides classes and functions for the estimation of statistical models.

tslearn Python module that provides a machine learning toolkit for time-series analysis.

Nomenclature

Mathematical Symbols

a, b, α, β Lowercase symbols are scalars

$\mathbf{x}, \mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\sigma}$ Bold lowercase symbols are vectors

$\mathbf{A}, \mathbf{B}$ Bold uppercase symbols are matrices

$\mathbf{x}^T, \mathbf{A}^T$ Transpose of a vector or matrix

$\mathbf{A}^{-1}$ Inverse of a matrix

$\mathbb{R}$ Real numbers

$\mathbb{R}^n$ n-dimensional vector space or real numbers

exp Exponential function

ln Natural logarithm

log Common logarithm with base 10

π Pi constant

$|.|$ Absolute value

$\|.\|$ Norm

max Maximum value

$perm$ Permutations

$det(\mathbf{A})$ Determinant of $\mathbf{A}$

$dist(p, q)$ Distance between points p and q

$\emptyset$ Empty set

$X \setminus x$ The set of elements in X without element x

$x \in X$ x is an element of set X

$\forall x$ Universal qualifier for all x

$\sum_{n=1}^{N} x_n$ Sum of $x_1 + ... + x_N$

$\mathbb{E}(X)$ Expectation of X

$\hat{x}$ Predicted value of x

$p(X)$ Probability of X

$p(X = x_i)$ Probability of X having value x_i

$p(X|Y)$ Conditional probability of X given Y

$X \sim p$ Random variable X is distributed according to p

$\mathcal{N}(\mu, \sigma^2, x)$ Gaussian distribution with mean μ and variance σ^2 for point x

$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{x})$ Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ for point $\mathbf{x}$

$SMOW$ Standard mean ocean water constant $3.1152 \cdot 10^{-4}$

Chemical Symbols

H_2O Chemical formula for water

HDO Chemical formula for semiheavy water, Deuterium hydrogen monoxide

Chapter 1

Introduction

Cluster analysis remains a research topic of great interest, and a considerable number of clustering algorithms have been developed and studied over time. Jain (2010) provides a well-received survey on clustering methods and summarizes major challenges and key issues in designing clustering algorithms. One of Jain's main conclusions is that:

> "Given the inherent difficulty of clustering, it makes more sense to develop semi-supervised clustering techniques in which the labelled data and (user specified) pair-wise constraints can be used to decide both (i) data representation and (ii) appropriate objective function for data clustering."

This conclusion is well demonstrated throughout this thesis, where the research focus evolves from unsupervised clustering to semi-supervised clustering. The overall goal and challenge of this thesis are to better understand the natural structure of spatio-temporal data that is reflected by meaningful clusters. In this context, meaningful clusters are partitions of data points that are similar to each other and can be related to known events or occurrences. In particular, this thesis is concerned with known occurrences that are correlated in multidimensional space.

The primary field of application for the proposed clustering methods is within the domain of climate research. The main interest in this domain has been to analyse pair distributions of water vapour (H_2O) and its isotopologue (HDO), which can be associated with atmospheric moisture processes. Identifying such processes is an important scientific task to infer the dynamics of cloud-circulation systems. Investigating the atmosphere from a cloud-circulation system perspective is essential to address the significant uncertainty of climate predictions (Bony et al., 2015).

Developing new methods for multidimensional clustering of spatio-temporal data poses several challenges. The main challenges addressed by this thesis can be described as following:

Challenge 1. *Learning the natural structure of spatio-temporal data reflected by meaningful clusters.*

Cluster analysis should aid researchers to learn the natural structure of data represented by meaningful clusters. Typically meaningful clusters are clusters of prominent groups in the data where points in the same cluster are more similar to each other than to points in different clusters.

Challenge 2. *Analysing existing clustering algorithms to cluster spatio-temporal data and to identify correlated structures in the dataset.*

Analysing existing clustering algorithms is the first step to find meaningful clusters in spatio-temporal data. For this purpose, a comprehensive review of available algorithms from literature has to be conducted and analysed in the context of spatio-temporal data. However, stipulating additional properties for the data points within the same cluster, i.e. defining terms for their similarity, such as identifying correlated structures, requires the inclusion of a priori knowledge into the clustering process.

Challenge 3. *Developing a clustering algorithm that can form partitions of data complying with a priori constraints in full value space or value subspaces.*

After identifying deficits in existing clustering algorithms, a novel clustering algorithm can be developed incorporating the findings of the analysis. In addition, this novel algorithm has to be capable of incorporating a priori knowledge in the form of constraints that restrict the formation of incomprehensible clusters.

Challenge 4. *Designing appropriate constraints and parameters for the developed clustering algorithm to identify correlated structures in spatio-temporal data.*

Suitable constraints have to be designed to fully utilize the a priori knowledge about the data, e.g. possible relations between variables. These constraints have to guide the clustering process to align the outcome of the clustering result with the expectations of the analyst.

Challenge 5. *Demonstrating the value of the developed clustering algorithm to perform cluster analysis of spatio-temporal data.*

The value of the developed clustering algorithm to perform cluster analysis of spatio-temporal data that incorporates a priori knowledge into the clustering process has to be demonstrated for scientific problems.

1.1 Scientific Contributions

This thesis addresses the presented challenges through the following achieved contributions in the respective research subtopics.

1.1.1 Unsupervised Cluster Analysis of Spatio-Temporal Data

The simultaneous observation of multiple variables or measurements at geo-referenced locations or areas over time enables the analysis of multivariate distributions of spatio-temporal data. In this context, cluster analysis is especially suited to discover and track multivariate distributions over time. The problem definition in the literature most closely related to these tasks is formulated as the analysis of moving spatio-temporal objects (Maciąg, 2017).

A **data-driven approach of tracking clusters in spatio-temporal data** has been presented at the International Conference on Knowledge Discovery and Information Retrieval (KDIR) (Ertl et al., 2019). This approach is based on the Gaussian Mixture Model (GMM) to extract cluster properties that can be analysed for changes over space and time. The work provides a methodology based on well-known algorithms and an interpretation of the algorithmic results in a spatio-temporal context.

In addition, the supervision of the master thesis by Weber (2019) has resulted in a comprehensive overview of clustering metrics for hyper-parameter optimization for spatio-temporal clustering with DBSCAN and HDBSCAN.

These activities can be summarised in the following contribution:

Contribution 1. *Cluster analysis of spatio-temporal data applying existing clustering algorithms to learn the intrinsic structure of the data.*

1.1.2 Semi-Supervised Cluster Analysis of Spatio-Temporal Data

Applying and analysing existing clustering algorithms has been the first step to find meaningful clusters in spatio-temporal data. While many advances in clustering algorithms have been made for spatio-temporal data (Maciąg, 2017; Wang et al., 2013), in general, the developed methods lack the exploitation of available a priori knowledge that might improve the clustering quality. Especially semi-supervised clustering algorithms, which incorporate a priori knowledge into the clustering process, can improve the quality of the results (Dinler and Tural, 2016).

After identifying the deficits with existing unsupervised clustering algorithms a **novel semi-supervised clustering algorithm** has been proposed that incorporates the findings of the initial analysis. The algorithm is based on the original DBSCAN clustering algorithm (Ester et al., 1996) and can be described as a **density-based clustering algorithm with constrained expansion**, namely **CoExDBSCAN**. The algorithm and the results have been published in Ertl et al. (2020) and presented at KDIR 2020. This can be summarised as following contribution.

Contribution 2. *Development of a novel semi-supervised clustering algorithm for spatio-temporal data.*

Especially recurring subsequences in streams of multiple measurements that can be organised as multivariate time-series, can be interpreted as recurring events or actions. These recurring events can be used to discover repeating patterns, to understand trends, to detect anomalies and in general to better interpret large and high-dimensional datasets (Hallac et al., 2017). By utilizing the proposed CoExDBSCAN algorithm for such time-series data and constraining the cluster extension to the correlation of time point values, clusters of segments with similar correlations can be identified. This novel semi-supervised approach for subsequence time-series clustering follows the pairwise semi-supervision approach from Aggarwal and Reddy (2013) and extends the concept to a **novel concept of cluster-wide constraints**. This approach has been demonstrated for **semi-supervised time point clustering for multivariate time-series** in general (Ertl et al., 2021a) at the Canadian Conference on Artificial Intelligence (CAI 2021) and as **a semi-supervised approach for trajectory segmentation to identify different moisture processes in the atmosphere** in particular (Ertl et al., 2021b) at the International Conference on Computational Science (ICCS 2021).

These activities can be summarised in the following three contributions:

Contribution 3. *Evaluation of constraints and parameters for the developed clustering algorithm to identify correlated structures in spatio-temporal data.*

Contribution 4. *Introduction of semi-supervised time point clustering for multivariate time-series.*

Contribution 5. *Introduction of a novel semi-supervised approach for trajectory segmentation to identify different moisture processes in the atmosphere.*

1.1.3 Operational Processing of Climate Research Data

Finally, the related work to the doctoral research presented in this thesis has provided datasets with accurate, long-term, global and high-resolution observations of tropospheric $\{H_2O, \delta D\}$ pairs from remote sensing data (Schneider et al., 2021b; Diekmann

et al., 2021a). This dataset is subject to a number of research studies in climate research (García et al., 2018; Borger et al., 2018; Diekmann et al., 2021b; Dahinden et al., 2021; Tu et al., 2021; Schneider et al., 2021a; Toride et al., 2021).

This activity can be summarised as the following contribution.

Contribution 6. *Generation of climate research data from observations for a five-year-long period for scientific studies.*

1.1.4 List of Publications

The following lists the publications to the contributions presented.

- Ertl, Meyer, Streit, and Schneider (2019).
  Application of Mixtures of Gaussians for Tracking Clusters in Spatio-Temporal Data. In Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1: KDIR, pages 45–54. INSTICC, SciTePress, 2019. ISBN 978-989-758-382-7. DOI:10.5220/0007949700450054.

- Ertl, Meyer, Schneider, and Streit (2020).
  CoExDBSCAN: Density-Based Clustering with Constrained Expansion. In Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1: KDIR, pages 104–115. INSTICC, SciTePress, 2020. ISBN 978-989-758-474-9.
  DOI: 10.5220/0010131201040115.

- Ertl, Meyer, Schneider, and Streit (2021a).
  Semi-Supervised Time Point Clustering for Multivariate Time Series. In Proceedings of the 34th Canadian Conference on Artificial Intelligence: CAI, 2021.
  DOI: 10.21428/594757db.9fa1eff5.

- Ertl, Schneider, Diekmann, Meyer, and Streit (2021b).
  A Semi-Supervised Approach for Trajectory Segmentation to Identify Different Moisture Processes in the Atmosphere. In Computational Science – ICCS 2021. Springer International Publishing, 2021.
  DOI: 10.1007/978-3-030-77961-0_23.

- Schneider, Ertl, Diekmann, Khosrawi, Weber, Hase, Höpfner, García, Sepúlveda, and Kinnison (2021b).
  Design and description of the MUSICA IASI full retrieval product. Earth System Science Data Discussions, pages 1-51. 2021.
  DOI: 10.5194/essd-2021-75

- Schneider, Ertl, Diekmann, Khosrawi, Röhling, Hase, Dubravica, García, Sepúlveda, Borsdorff, Landgraf, Lorente, Chen, Kivi, Laemmel, Ramonet, Crevoisier, Pernin, Steinbacher, Meinhardt, Deutscher, Griffith, Velazco, and Pollard (2021a).
  Synergetic use of IASI and TROPOMI space borne sensors for generating a tropospheric methane profile product. Atmospheric Measurement Techniques Discussions, 2021:1–37, 2021.
  DOI: 10.5194/amt-2021-31.

- Diekmann, Schneider, Ertl, Hase, García, Khosrawi, Sepúlveda, Knippertz, and Braesicke (2021a).
  The global and multi-annual MUSICA IASI $\{H_2O, \delta D\}$ pair dataset. Earth System Science Data, pages 5273-5292. 2021.
  DOI: 10.5194/essd-13-5273-2021.

- Diekmann, Schneider, Knippertz, de Vries, Pfahl, Aemisegger, Dahinden, Ertl, Khosrawi, Wernli, and et al. (2021b).
  A Lagrangian perspective on stable water isotopes during the West African Monsoon. Earth and Space Science Open Archive, pages 1-43. 2021
  DOI: 10.1002/essoar.10506628.1

- Dahinden, Aemisegger, Wernli, Schneider, Diekmann, Ertl, Knippertz, Werner, and Pfahl (2021).
  Disentangling different moisture transport pathways over the eastern subtropical North Atlantic using multi-platform isotope observations and high-resolution numerical modelling. Atmospheric Chemistry and Physics Discussions, pages 1-49. 2021
  DOI: 10.5194/acp-2021-269

- Tu, Hase, Blumenstock, Schneider, Schneider, Kivi, Heikkinen, Ertl, Diekmann, Khosrawi, Sommer, Borsdorff, and Raffalski (2021).
  Intercomparison of arctic XH2O observations from three ground-based Fourier transform infrared networks and application for satellite validation. Atmospheric Measurement Techniques, 14(3):1993–2011, 2021.
  DOI: 10.5194/amt-14-1993-2021.

- Toride, Yoshimura, Tada, Diekmann, Ertl, Khosrawi, and Schneider (2021). Potential of Mid-tropospheric Water Vapor Isotopes to Improve Large-Scale Circulation and Weather Predictability. Geophysical Research Letters, 48(5): e2020GL091698, 2021.
  DOI: 10.1029/2020GL091698.

1.2 Thesis Outline

This thesis is structured in seven chapters. Following the introduction and scientific contributions in this chapter, Chapter 2 explains the necessary concepts that serve as a foundation for this thesis. A basic introduction to cluster analysis is given with a detailed explanation of the Gaussian Mixture Model and the DBSCAN clustering algorithm, which are extensively studied throughout this thesis. Further, important metrics for the validation of clustering results are given, as well as an overview of the characteristics of spatio-temporal data and the climate research data subject to this thesis, the MUSICA IASI satellite-based remote sensing dataset, in particular.

Chapter 3 discusses related work in the area of unsupervised cluster analysis, semi-supervised cluster analysis and cluster analysis of climate data. In particular with focus on clustering moving objects (Section 3.1.1), time-series clustering (Section 3.1.2) and semi-supervised clustering with point-wise and pair-wise semi-supervision (Section 3.2.1, 3.2.2).

Chapter 4 presents the initial work on analysing the evolution of geo-referenced distributions over time by developing an unsupervised clustering method that allows for tracking moving, emerging and changing distributions over time.

Following the evolution from unsupervised clustering towards semi-supervised clustering, Chapter 5 details the design, implementation and verification of the CoExDBSCAN algorithm in the context of semi-supervised clustering of spatio-temporal data.

In Chapter 6, this novel semi-supervised clustering method is demonstrated for semi-supervised time point clustering for multivariate time-series in general (Section 6.3.1) and as a semi-supervised approach for trajectory segmentation to identify different moisture processes in the atmosphere in particular (Section 6.3.2).

Chapter 7 presents the final conclusion and an outlook for future work.

Chapter 2

Background

In this chapter, the notion of cluster analysis in the context of spatio-temporal data and climate research data in particular is explained. These explanations are given to provide a common understanding of the terminology and notation used throughout this thesis. The first sections provide an overview about the properties of spatio-temporal data and climate research data in particular, which are the main types of datasets under consideration for this thesis. The subsequent sections provide a brief overview of the history of cluster analysis and introduce the basic methodology. Following this introduction, clustering algorithms are classified based on their a priori knowledge about the data. Two clustering algorithms extensively studied in this thesis are explained in detail, the Gaussian Mixture Model and DBSCAN, a density-based clustering algorithm for discovering clusters in large spatial databases with noise. The introduction to cluster analysis concludes with a summary of metrics that have been used to assess the quality of clustering algorithms throughout this thesis. The background presented in this chapter has been published in parts in Ertl et al. (2019, 2020, 2021a,b).

2.1 Spatio-Temporal Data

Spatio-temporal data contains information about where and when the data has been collected. Typical examples for spatio-temporal data are climate data or data about moving objects in space. Without the temporal dimension, the data is referred to as spatial data, for example, geographic data or any data that has an implicit or explicit association with a location in a given space. In the absence of the spatial dimension, the data is referred to as temporal data, for example, time-series where data points are indexed by the time of collection.

Kisilevich et al. (2010) provide a survey on spatio-temporal clustering and a possible classification of spatio-temporal data, see Figure 2.1.

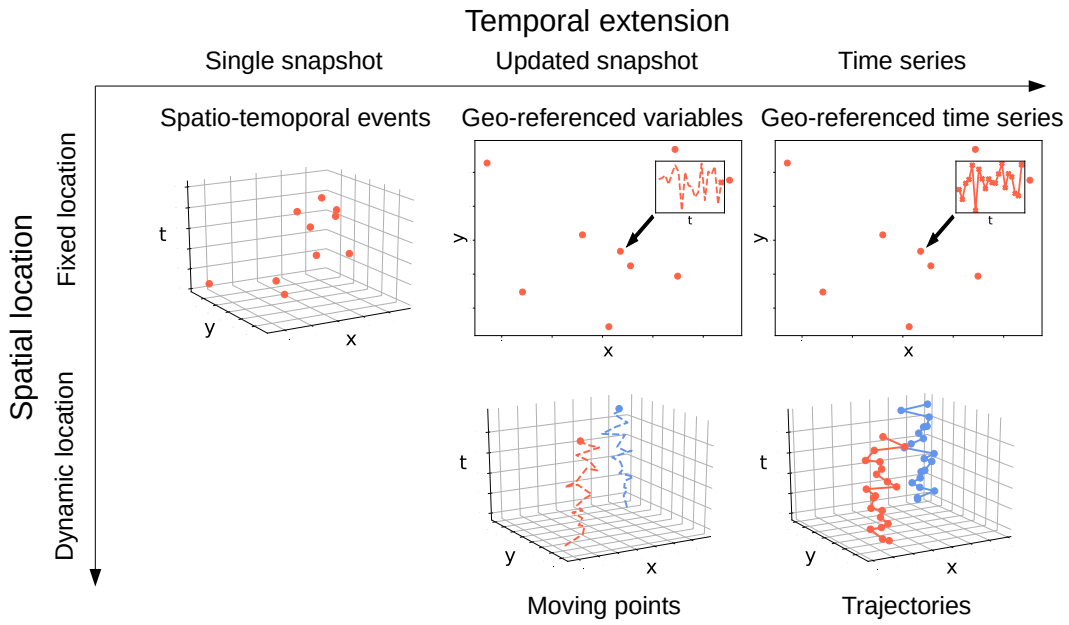


Figure 2.1: Classification of spatio-temporal data based on Kisilevich et al. (2010).

According to Kisilevich, spatio-temporal data can be distinguished based on the temporal extension, i.e. to which extent the temporal history of spatial points or objects is captured by the data, the spatial dimension, i.e. if the locations of the data points are fixed or change dynamically, and the spatial extent of the data points, i.e. a distinction between points and objects like lines, areas or volumes; the latter is not illustrated in Figure 2.1.

A single snapshot provides a static temporal view of the data, where points or objects do not evolve over time. A geo-referenced variable or moving points provide an updated temporal view of the data, i.e. the most recent values, thus points or objects evolving over time, but without any knowledge about the past history. If the past history is recorded as well, points or objects belong into the category of geo-referenced time-series or trajectories.

Depending on the analysis to conduct, spatio-temporal data can be transformed from one category to another by moving along the directions on the spatial location axis and the temporal extension axis indicated in Figure 2.1. From dynamic location to fixed location, for example, by interpolation, and from time-series to single snapshot, for instance, by analysing single time slices of the data.

The datasets mainly used throughout this thesis cover most of the spatio-temporal data classes by Kisilevich et al. (2010). For example the LÍngua BRAsileira de Sinais (LIBRAS) movement dataset (Dias et al., 2009) analysed in Section 6.3.1 and the atmospheric model data analysed in Section 6.3.2 can be classified as datasets of trajectories, while the MUSICA IASI dataset described in detail in the next section can be viewed as geo-referenced variables, geo-referenced time-series and moving points.

2.2 MUSICA IASI Data

Climate research data can be divided into measured data and simulated data. Measured data is recorded by means of scientific instruments, while simulated data is computed based on climate models. In the course of this thesis, datasets from both domains are under study to uncover challenges for existing cluster analysis methods and to propose solutions to these challenges.

The measured data primarily used throughout this thesis is the MUSICA IASI satellite-based remote sensing dataset (Borger et al., 2018; Schneider et al., 2021b; Diekmann et al., 2021a). This unique dataset has become only recently available through advances in satellite sensor technology and retrieval theory. The generation of this climate research data from observations for a five-year-long period as part of this thesis has been a major contribution to multiple scientific studies (García et al., 2018; Borger et al., 2018; Schneider et al., 2021a; Dahinden et al., 2021; Tu et al., 2021; Toride et al., 2021) and is subject of multiple ongoing scientific studies.

The MUSICA IASI dataset provides accurate, long-term, global and high-resolution observations of tropospheric $\{H_2O, \delta D\}$ pairs. H_2O corresponds to the water vapour concentration in the atmosphere and is measured in parts per million by volume (ppmv); δD corresponds to the standardised ratio value between heavy and light water, i.e. HDO and H_2O. The Infrared Atmospheric Sounding Interferometer (IASI) onboard the Metop-A, B and C polar-orbiting meteorological satellites measure approximately 90,000 spectra per orbit. Each satellite completes 14 orbits per day, which results in around 3.8 million spectra per day. Each spectrum has to be processed with the thermal nadir retrieval algorithm PROFFIT-nadir (Schneider and Hase, 2011; Wiegele et al., 2014). A comprehensive summary of the MUSICA IASI dataset is given by Schneider et al. (2021b).

The global $\{H_2O, \delta D\}$ pair distribution observed in the MUSICA IASI satellite-based remote sensing dataset can be linked to different moisture processes that occurred prior to the observation. For illustration purpose Figure 2.2 shows the global H_2O observations retrieved from the Infrared Atmospheric Sounding Interferometer (IASI)

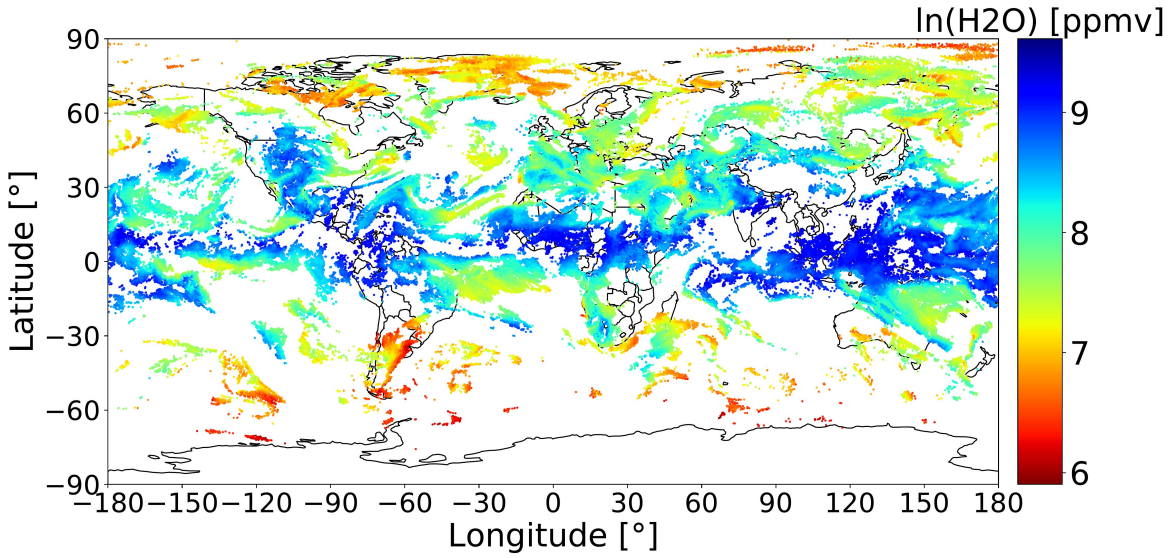


Figure 2.2: Example global MUSICA IASI H_2O data for morning satellite overpasses at 2016-06-08. H_2O values are in parts per million by volume (ppmv) in logarithmic scale. The depicted data are limited to cloud-free observations and have been filtered for best quality (retrievals with good sensitivity and low errors).

onboard the European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT) Metop-A and Metop-B satellites for morning overpasses at the 8$^{\text{th}}$ June 2016 for about five kilometers altitude. For this single day, 183,036 individual observations are available after filtering out cloudy and partly cloudy observations as well as observations with bad quality.

Figure 2.3 depicts the $\{H_2O, \delta D\}$ pair distribution starting from the same date at the 8$^{\text{th}}$ June 2016 until the 30$^{\text{th}}$ June 2016 for some area of interest All MUSICA IASI $\{H_2O, \delta D\}$ data are shown as grey dots and the contours are at 2.5%, 10% and 50% levels, meaning the percentage of data lying outside the indicated area.

Different water cycle processes affect the isotopic composition of atmospheric water differently. For example lighter isotopes evaporate preferentially while heavier isotopes condense preferentially. The red lines in Figure 2.3 illustrate the theoretical dependencies of δD as a function of H_2O. Noone (2012) differentiates between five processes that leave a distinct trace in the $\{H_2O, \delta D\}$ value space:

1. **Rayleigh pseudoadiabatic** process in which the liquid water that condenses is assumed to be removed as soon as it is formed by idealized instantaneous precipitation (red dotted line in Figure 2.3)

2. **Super-Rayleigh** remoistening associated with isotopic exchange as raindrops evaporate into a subsaturated layer (red solid line in Figure 2.3)

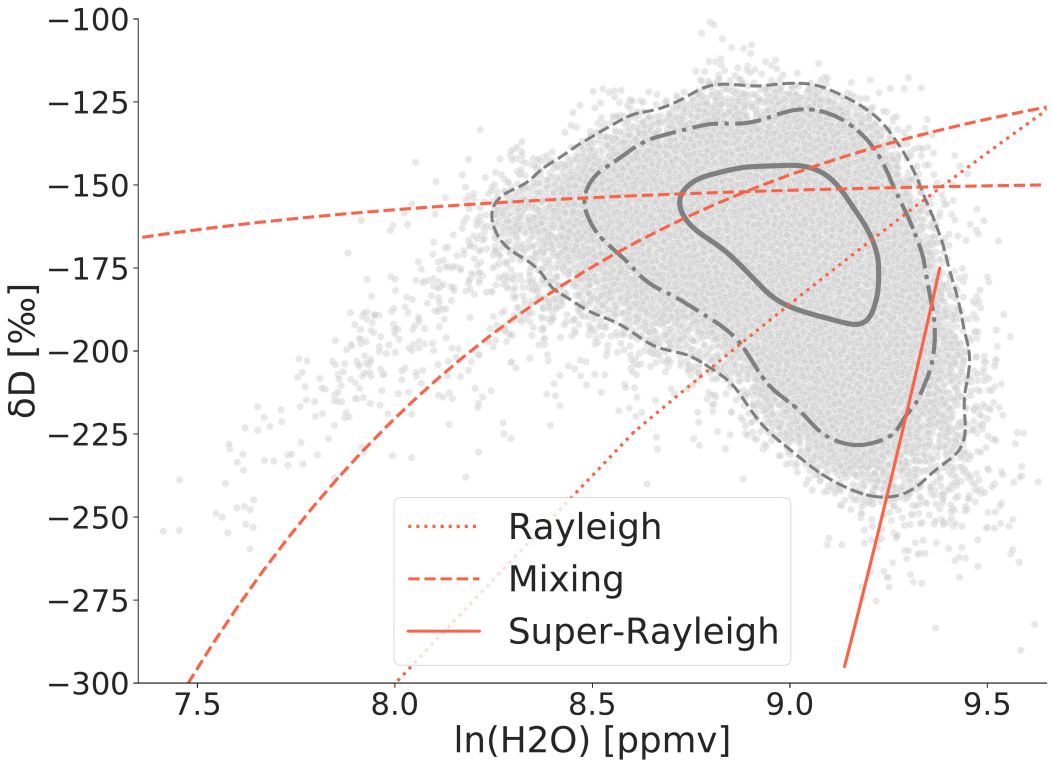


Figure 2.3: Example MUSICA IASI $\{H_2O, \delta D\}$ pair distribution. The 61,283 observations are for morning and evening satellite overpasses from 2016-06-08 to 2016-07-30 with H_2O in logarithmic scale in an area over West Africa; red lines indicate theoretical lines for different water cycle processes.

3. **Reversible moist adiabatic** process with a transition to a Rayleigh process when condensation is to ice and irreversible (not shown, would be a line with a weaker slope as the dotted line in Figure 2.3)

4. **Mixing** of two different mixing members having a specific $\{H_2O, \delta D\}$ characteristic (red dashed lines in Figure 2.3 show two examples)

5. **Terrestrial transpiration** mixing with land source (not shown in Figure 2.3; would be similar to the red dashed lines, but shifted to higher δD values)

Noone's work establishes a theoretical basis for using isotope ratio observations paired with the water vapour mixing ratio to identify different water sources, condensation processes, and transport pathways in the troposphere. Moreover, Noone et al. (2011) were able to derive slope and intercept of the linear relationship between H_2O and δD from measurements of the isotope ratio of water vapour at the Mauna Loa Observatory.

2.3 Atmospheric Model Data

The simulated data primarily used throughout this thesis is the high-resolution data from the regional isotope-enabled atmospheric model COSMO-iso (Pfahl et al., 2012) that models $\{H_2O, \delta D\}$ pairs along Lagrangian air parcel trajectories. These trajectories are determined with the Lagrangian Analysis Tool (LAGRANTO) (Sprenger and Wernli, 2015). The trajectories' calculation setup is oriented towards the overpass times and altitudes representative for the MUSICA IASI data. Analysing the model data helps to reveal the kind of moisture processes that can be observed in the MUSICA IASI $\{H_2O, \delta D\}$ pair data.

Figure 2.4 illustrates 11,853 Lagrangian air parcel trajectories with the arrival of all trajectories in an area above West Africa at pressure levels 575 and 625 Hectopascal (hPa), which corresponds to an altitude of about 5 km, i.e. the altitude of the data points shown in Figure 2.2 and Figure 2.3. The trajectories are calculated daily for local morning (9 am) and evening (9 pm) times during the period from June 8, 2016, to July 30, 2016, with 169 time steps each with a time delta of one hour; each trajectory comprises a time frame of 7 days. The coloured 1,479 trajectories (orange and green) have been labelled according to their similarity, using DBSCAN on a pre-computed distance matrix, as an illustrative example. For the pre-computed distance matrix, each trajectory has been converted to a $4 \cdot 169 = 676$ dimensional vector; the latitudinal (1) and longitudinal (2) difference for each time point to the arrival coordinates, the bearing (3) for each consecutive point and the scaled height difference (4) for each consecutive point, 169 points each.

Figure 2.5 shows the $\{H_2O, \delta D\}$ distributions of the model data illustrated in Figure 2.4 for all data points along the trajectories coloured according to their labelling. The contour levels are at 10% and 50%, meaning the percentage of data lying outside the indicated area. The $\{H_2O, \delta D\}$ distributions are visibly different for the two clusters of trajectories, indicating a different air composition depending on the origin of the trajectories (East or West). This elementary analysis demonstrates the information content in the $\{H_2O, \delta D\}$ pair data and motivates further investigations on $\{H_2O, \delta D\}$ clustering possibilities.

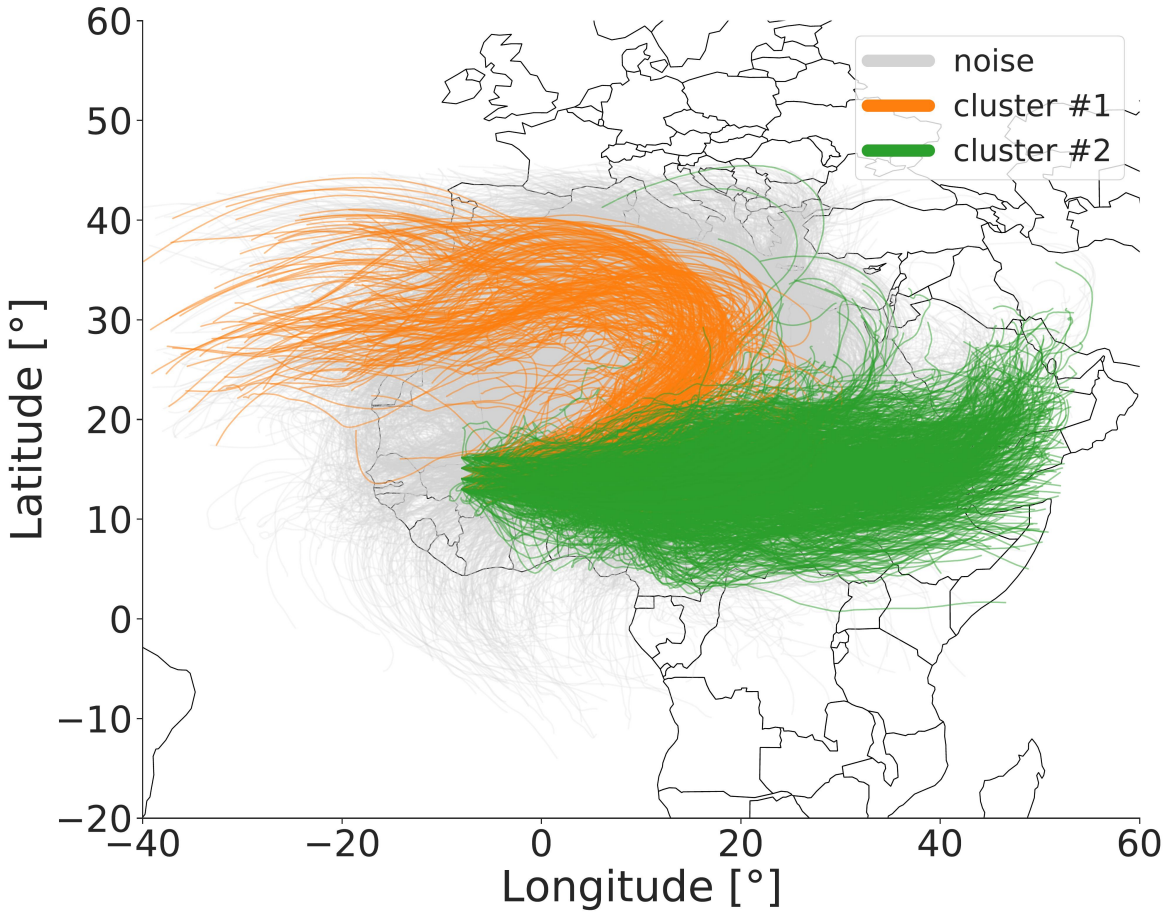


Figure 2.4: Illustration of 1,479 trajectories out of 11,853 that have been coloured according to their geographical closeness, bearing similarity and height difference along each individual trajectory.

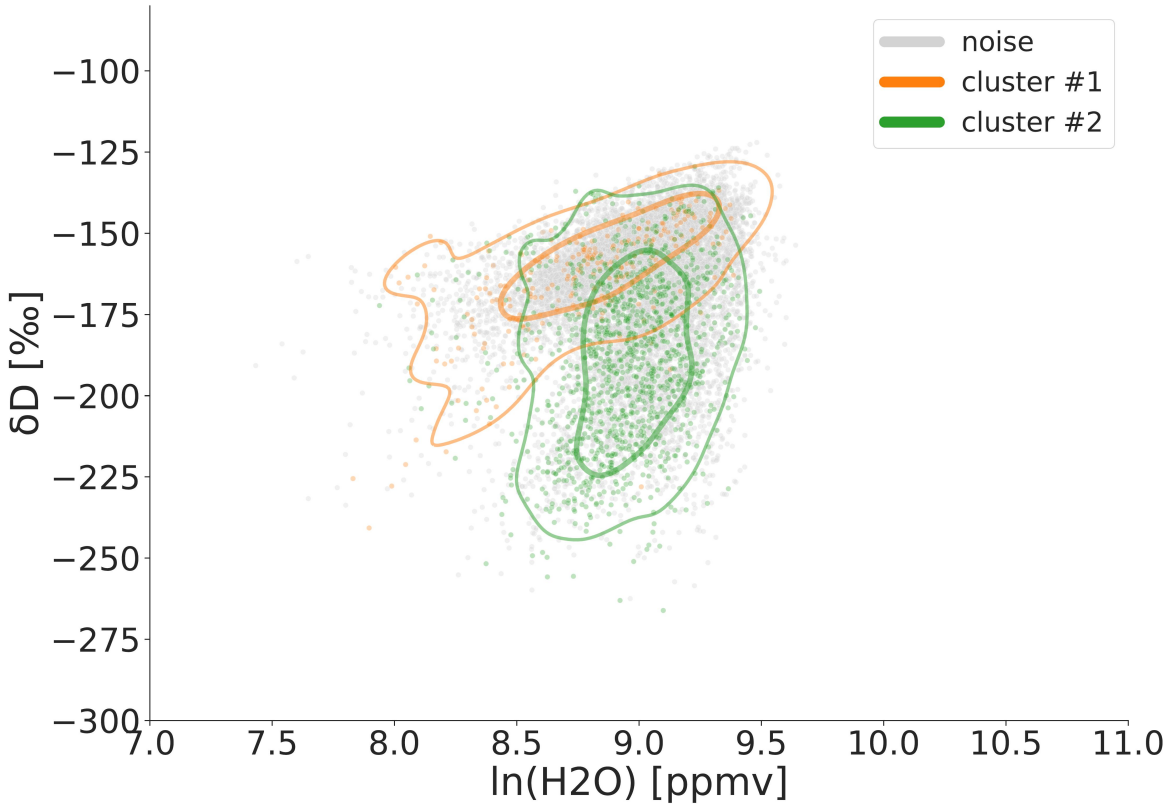


Figure 2.5: Example $\{H_2O, \delta D\}$ distributions of the model data illustrated in Figure 2.4 for all noise, cluster #1 and cluster #2 data points; contour levels are at 10% and 50%.

2.4 Cluster Analysis

Cluster analysis can be expressed as the task of finding a partition of a given set of points in a multidimensional space. The points within each partition are similar to one another. The similarity is typically expressed as the closeness of points, i.e. the pairwise distance of data points in a particular value space. Anthropologists have published the earliest concepts for cluster analysis in the early 20th century by Driver and Kroeber (1932), and were later picked up by psychologists Zubin (1938) and Tryon (1939), according to Blashfield and Aldenderfer (1988). In the mid 20th century, one of the most popular and simplest partitional algorithm k-means was independently discovered in different scientific fields by Steinhaus (1956), Lloyd (1982), Ball and Hall (1965) and MacQueen et al. (1967), as described by Jain (2010). In contrast to k-means, which requires the number of clusters to be specified as a parameter, the mean shift algorithm has been invented by Fukunaga and Hostetler (1975) as a non-parametric mode finding

procedure around the same time. Towards the end of the 20th century, Ester et al. (1996) published DBSCAN, a density-based clustering algorithm that introduced a new notion of clusters based on the density of point neighbourhoods. DBSCAN has been proven to be successful in many real-world applications and inspired many other algorithms (Schubert et al., 2017). HDBSCAN* by Campello et al. (2013) and OPTICS by Ankerst et al. (1999) are two examples of DBSCAN variants that are better known and implemented in multiple clustering toolkits and libraries, for example, scikit-learn (Pedregosa et al., 2011) or ELKI (Schubert and Zimek, 2019). Schubert et al. (2017) have shown that DBSCAN continues to be relevant even for high-dimensional data, although the parameters of the algorithm become hard to choose due to the loss of contrast in distances. DBSCAN is one of the main algorithms extensively studied in this thesis and is explained in more detail in Section 2.4.3.

2.4.1 Classification of Clustering Algorithms

A general distinction can be made between partitional and hierarchical clustering algorithms. Partitional or flat methods, e.g. the k-means algorithm and its variants, divide the data into several clusters in one go. Hierarchical methods represent clusters at different levels of granularity, built either top-down (i.e. divisive approach) by splitting bigger clusters into smaller clusters or bottom-up (i.e. agglomerative approach) by merging smaller clusters into bigger clusters.

Partitional and hierarchical clustering algorithms can further follow a parametric or non-parametric approach. Roberts (1997) provides a detailed view into parametric versus non-parametric clustering methods, where either the data density is estimated according to a parametric model or no parametric form of the data density is assumed. The parametric approach is also referred to as a probabilistic or generative model. Also, a combination of parametric and non-parametric clustering algorithms can be used to group the modelled data as a semi-parametric density (Zhang et al., 2019). With a probabilistic clustering model, data points can belong to several clusters with a certain probability, also referred to as soft- or fuzzy clustering, unlike hard clustering, where every data point belongs to exactly one cluster, for discriminative or distance-/similarity-based clustering approaches.

Another distinction can be made between full-space clustering algorithms and subspace clustering algorithms, depending on the dimensionality of the input data to the algorithms. In general, full-space clustering algorithms have problems with high dimensional data because of the curse of dimensionality. The curse of dimensionality refers to a number of problems with high dimensionality, e.g. multivariate density estimation (Scott, 2015; Beyer et al., 1999). As the number of dimensions in a dataset

increases, the distance measures become more and more meaningless. In very high dimensions, the data points are spread out until they are almost equidistant from each other (Parsons et al., 2004). Therefore, in addition to full-space clustering algorithms, subspace clustering algorithms have been proposed. Subspace clustering algorithms aim to find clusters in multiple, possibly overlapping subspaces (Parsons et al., 2004).

Another differentiation can be made between traditional fully unsupervised clustering algorithms and semi-supervised clustering algorithms. While unsupervised clustering algorithms make no assumptions about the data subject to the clustering process, semi-supervised clustering algorithms incorporate domain knowledge to some extent, e.g. in the form of labels for a portion of the data, constraints on data points to be in the same cluster (i.e. must-link) or in different clusters (i.e. cannot-link), or otherwise formulated constraints. The main goal of semi-supervised clustering methods is to align better the data partitioning with the domain knowledge. More details on semi-supervised clustering methods are given in the next chapter, Chapter 3.

2.4.2 Gaussian Mixture Model

A frequently mentioned first publication that proposed a mixture of Gaussian models to explain and partition data dates back to the statistician Karl Pearson (1894). Since then, mixture models have been the subject of extensive research with finite mixtures of distributions providing a sound mathematical basis for statistical modelling of a wide variety of random phenomena (McLachlan et al., 2019). Finite mixture models are formed by linear combinations of basic distributions that are superimposed. By using a sufficient number of Gaussian distributions with adjusted means and covariances as well as adjusted contributions to the linear combination, any continuous density can be approximated to arbitrary accuracy with few exceptions (Bishop, 2006). For the purpose of cluster analysis, this means that data is going to be partitioned into a number of possible multi-variate Gaussian distributions with the mean and variance parameters optimized with regards to the comprising data points. The Gaussian mixture model for clustering data belongs into the category of unsupervised, partitional and parametric model-based clustering algorithms, while non-parametric modifications have been proposed as well, for example by Mallapragada et al. (2010).

Ankur Moitra provides a coherent introduction to the Gaussian mixture model in his book "Algorithmic Aspects of Machine Learning" accompanying the identically named MIT lecture (Moitra, 2015) that is adopted and complemented with the description of the Gaussian Mixture Model by Bishop (2006) in the following summary of definitions and equations. A list of mathematical symbols and expressions used in the following is given in the nomenclature.

The probability density function of a one-dimensional Gaussian distribution with mean μ and variance σ^2 for a data point x is:

$$\mathcal{N}(\mu, \sigma^2, x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{2.1}$$

For a multidimensional Gaussian distribution in $\mathbb{R}^n$ with covariance matrix $\mathbf{\Sigma}^{n\times n}$ the probability density function is given by:

$$\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma}, \mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n det(\mathbf{\Sigma})}} exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \tag{2.2}$$

A mixture model of two one-dimensional Gaussians, denoted as p_1 and p_2, and probability (i.e. weight) w_1 to choose from either distribution can be formulated without loss of generality as:

$$p(x) = w_1 \underbrace{\mathcal{N}(\mu_1, \sigma_1^2, x)}_{p_1(x)} + (1-w_1) \underbrace{\mathcal{N}(\mu_2, \sigma_2^2, x)}_{p_2(x)} \tag{2.3}$$

The multidimensional model can be formulated accordingly with the total number of mixture components K:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \mathbf{w}_k \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{\Sigma}_k, \mathbf{x}) \tag{2.4}$$

with each Gaussian density component $\mathcal{N}(\boldsymbol{\mu}_k, \mathbf{\Sigma}_k, \mathbf{x})$ having its own mean $\boldsymbol{\mu}_k$ and covariance $\mathbf{\Sigma}_k$ and the mixing coefficients or weight parameters $\mathbf{w}_k$ satisfying the condition:

$$\sum_{k=1}^{K} \mathbf{w}_k = 1 \tag{2.5}$$

The maximum likelihood can be estimated by the Expectation–maximization algorithm (EM) to set the values for the Gaussian Mixture Model (GMM) parameters $\boldsymbol{\mu}_k$, $\mathbf{\Sigma}_k$ and $\mathbf{w}_k$. The EM algorithm finds the local optimum for parameters in the likelihood for models with latent variables from the given data (Dempster et al., 1977). The logarithm of the likelihood function for Equation 2.4 summing over N number of observations is given by:

$$\ln p(\mathbf{X}|\mathbf{w}, \boldsymbol{\mu}, \mathbf{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \mathbf{w}_k \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{\Sigma}_k, \mathbf{x_n}) \right\} \tag{2.6}$$

After initializing the parameters $\boldsymbol{\mu}_k$, $\mathbf{\Sigma}_k$ and $\mathbf{w}_k$ and evaluating the initial value of the likelihood (see Equation 2.6) the EM algorithm iterates in two steps until convergence for maximizing the likelihood.

1. **Expectation step**: For all data points the posterior probabilities for each *responsibility* are computed, i.e. the probability that the i-th data point $\mathbf{x}_i$ belongs to the k-th component of the mixture. Where z_k is the k-th element of the binary random variable $\mathbf{z}$ with $\sum_{j=1}^{K} z_j = 1$ and $z_j \in \{0, 1\}$, specified in terms of the mixing coefficients $\mathbf{w}_k$ such that $p(z_k = 1) = \mathbf{w}_k$.

$$p(z_k = 1|\mathbf{x}_i) = \frac{p(z_k = 1)p(\mathbf{x}_i|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}_i|z_j = 1)} = \frac{\mathbf{w}_k \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \mathbf{x}_i)}{\sum_{j=1}^{K} \mathbf{w}_j \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \mathbf{x}_i)} \tag{2.7}$$

2. **Maximization step**: The parameters $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and $\mathbf{w}_k$ are re-estimated using the computed responsibilities. If the convergence criterion is not satisfied the algorithm returns to the expectation step.

$$\begin{aligned} \boldsymbol{\mu}_k &= \frac{\sum_i \left\{p(z_k = 1|\mathbf{x}_i)\mathbf{x}_i\right\}}{\sum_i p(z_k = 1|\mathbf{x}_i)} \\ \boldsymbol{\Sigma}_k &= \frac{\sum_i \left\{p(z_k = 1|\mathbf{x}_i)(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T\right\}}{\sum_i p(z_k = 1|\mathbf{x}_i)} \\ \mathbf{w}_k &= \frac{\sum_i p(z_k = 1|\mathbf{x}_i)}{N} \end{aligned} \tag{2.8}$$

After convergence of the EM algorithm, each k-th Gaussian distribution represents a cluster, and the posterior probability for each data point $\mathbf{x}_i$ can be computed (see Equation 2.7). Each data point is associated with the cluster/mixture component with the highest probability at the end.

Estimating the number of mixture components k can be done in an efficient way by evaluating the Bayesian information criterion (BIC) (Schwarz, 1978) and penalizing the likelihood by the number of clusters (Scott Shaobing Chen and Gopalakrishnan, 1998). This procedure is explained in more detail in Section 2.4.4 together with other metrics for clustering algorithms.

2.4.3 DBSCAN

Ester et al. (1996) presented the DBSCAN algorithm at the International Conference on Knowledge Discovery and Data Mining (KDD) in 1996 as a density-based algorithm for discovering clusters in large spatial databases with noise. Ester introduced six essential definitions that are recapitulated in the following.

Definition 1. *ϵ-neighbourhood of a point*

Let D be a set (database) of points. The ϵ-neighbourhood of a point p, denoted by $N_\epsilon(p)$, is defined by

$$N_\epsilon(p) = \{q \in D | dist(p, q) \leq \epsilon\}$$

The shape of any ϵ-neighbourhood is determined by the distance function $dist(p, q)$. The choice of a distance function has a significant impact on the clustering results. Any metric can be used, while popular choices are the Euclidean distance or the Mahalanobis distance (Xu and Tian, 2015). In particular, any distance matrix constructed to reflect pairwise relationships can be utilized.

Definition 2. *Directly density-reachable*
A point p is directly density-reachable from a point q wrt. ϵ and minPts if

1. $p \in N_\epsilon(q)$ *and*
2. $|N_\epsilon(q)| \geq minPts$ *(core point condition).*

According to Definition 2 only core points, i.e. points that have at least $minPts$ in their ϵ-neighbourhood, can have directly density-reachable relations to other points within their ϵ-neighbourhood. To allow the algorithm to explore the full data space, the directly density-reachable definition is extended as follows:

Definition 3. *Density-reachable*
A point p is density-reachable from a point q wrt. ϵ and minPts if there is a chain of points $(p_1, ..., p_n, p_1 = q, p_n = p)$ such that p_{i+1} is directly density-reachable from p_i.

Definition 3 covers the reachability from core points ($|N_\epsilon(q)| \geq minPts$) to border points, points with less than $minPts$ in their ϵ-neighbourhood that are not noise (see Definition 6), but not from border points to border points, which is defined as following.

Definition 4. *Density-connected*
A point p is density-connected to a point q wrt. ϵ and minPts if there is a point o such that both, p and q are density-reachable from o wrt. ϵ and minPts.

With Definition 1 to 4 a density-based cluster can be defined as following.

Definition 5. *Cluster*
A cluster C wrt. ϵ and minPts is a non-empty subset of a set of points, D, satisfying the following conditions:

1. $\forall p, q$*: if* $p \in C$ *and q is density-reachable from p wrt. ϵ and minPts, then* $q \in C$*. (Maximality)*
2. $\forall p, q \in C$*: p is density-connected to q wrt. ϵ and minPts. (Connectivity)*

All points that do not belong to any cluster following Definition 5 are classified as noise.

Definition 6. *Noise*
Let $C_1, ..., C_k$ be the clusters of a set of points D wrt. parameters ϵ_i and $minPts_i$, $i = 1, ..., k$. Then the noise can be defined as the set of points in D not belonging to any cluster C_i, i.e. noise $= \{p \in D | \forall i : p \notin C_i\}$.

Equipped with these six definitions, a density-based algorithm has to find all clusters with respect to the parameters ϵ and $minPts$ for a given set of data points D. The DBSCAN algorithm achieves this goal as outlined in the pseudocode representation given in Algorithm 1, that has been adopted from Schubert et al. (2017), and can be described as following (all Line references refer to Algorithm 1).

DBSCAN starts with an arbitrary point from the set of all points and iterates over all points (outer loop, Line 1). The algorithm moves to the next point if the current point has already been labelled (Line 2). If not, the ϵ-neighbourhood for the given point is located (Line 3). If the ϵ-neighbourhood contains less than $minPts$ points (Line 4), the point is labelled as noise (Line 5) and the algorithm moves to the next point. If not, the point is considered a core point, and the algorithm starts to form a cluster with the next available label (Line 7-8). Proceeding from the core point, all neighbouring points are added to a set of initially empty seeds (Line 9).

For each point in this set (inner loop, Line 10), if the point has been previously labelled as noise, the point is labelled according to the current cluster label (Line 11). If the point has already been labelled with a cluster label (Line 12), the algorithm moves to the next point in the set (inner loop). If not, the ϵ-neighbourhood for the given point is located, and the point is labelled with the current cluster label (Line 13-14). If the ϵ-neighbourhood contains less than $minPts$ (Line 15), the algorithm moves to the next point in the set (inner loop). If not, each point from the ϵ-neighbourhood of the current seed point is added to the set of seed points (Line 16-17). The algorithm moves to the next point in the set (inner loop). If the set of seed points is empty, the algorithm moves to the next point in the dataset (outer loop).

Algorithm 1: Pseudocode of the DBSCAN Algorithm

```
input  : dataset D
input  : radius ε
input  : density threshold minPts
input  : distance function dist
output : point labels label initially undefined
foreach point p in dataset D do
    if label(p) ≠ undefined then continue;
    Neighbours N ← RangeQuery(D, dist, p, ε);
    if |N| < minPts then
        label(p) ← Noise;
        continue;
    c ← next cluster label;
    label(p) ← c;
    Seed set S ← N \ {p};
    foreach q in S do
        if label(q) = Noise then label(q) ← c;
        if label(q) ≠ undefined then continue;
        Neighbours N ← RangeQuery(D, dist, q, ε);
        label(q) ← c;
        if |N| < minPts then continue;
        foreach s in N do
            S ← S ∪ s;
```

2.4.4 Metrics

Clustering validation metrics can be categorized into external clustering validation and internal clustering validation (Aggarwal and Reddy, 2013, Chapter 23, Xiong et al.). External validation metrics use external information, for example, the true labels of the data, to validate the clustering result. Internal validation metrics use measurements with respect to the discovered clusters, for example, the within- and between-cluster distance. If a priori knowledge is available about the dataset subject to cluster analysis, internal cluster measurements and cluster properties can provide a sound validation of the clustering results.

The Rand index has been introduced by Rand (1971a) as a measure of similarity between clusterings and is an internal metric. The Rand index measures the similarity

between two data clusterings by counting equal elements in subsets created by the two partitions of the data, the true partition (true labels) and the computed partition (predicted labels) (Rand, 1971b). For this purpose, if $P = \{P_1, ..., P_s\}$ is the set of true partitions and $C = \{C_1, ..., C_t\}$ the set of computed partitions and for the incidence matrix $N \times N$ for N data points $O_1, ..., O_N$, two indicators can be defined, $P_{ij} = 1$ if O_i and O_j belong to the same true partition in P, $P_{ij} = 0$ otherwise, and $C_{ij} = 1$ if O_i and O_j belong to the same computed partition in C, $C_{ij} = 0$ otherwise. With these two indicators, P_{ij} and C_{ij}, four categories can be defined: 1) TT: $C_{ij} = 1$ and $P_{ij} = 1$, 2) FF: $C_{ij} = 0$ and $P_{ij} = 0$, 3) TF: $C_{ij} = 1$ and $P_{ij} = 0$ and 4) FT: $C_{ij} = 0$ and $P_{ij} = 1$. Category 1) and 2) indicate agreement while category 3) and 4) indicate disagreement. With these definitions the Rand index can be formulated as following:

Definition 7. *Rand index*

$$Rand = \frac{|Agreement|}{|Agreement| + |Disagreement|} = \frac{|TT| + |FF|}{|TT| + |TF| + |FT| + |FF|}$$

Since the expected value of the Rand index, $\mathbb{E}(Rand)$, of two random partitions does not take a constant value, Hubert and Arabie (1985) introduced an adjustment for chance to the Rand index. The adjusted Rand index is thus ensured to have a value close to zero for random labelling independently of the number of clusters and samples and exactly one when the clusterings are identical, up to a permutation (Pedregosa et al., 2011). With Definition 7 the adjusted Rand index can be formulated as follows:

Definition 8. *Adjusted Rand index*

$$ARI = \frac{Rand - \mathbb{E}(Rand)}{max(Rand) - \mathbb{E}(Rand)}$$

Another metric, the clustering accuracy, finds the best match between true labels and predicted labels. The greater the clustering accuracy, the better the clustering performance (Role et al., 2019). The clustering accuracy can be defined as follows:

Definition 9. *Clustering accuracy (ACC)*
Given a number of clusters K and the set of all permutations P in $[1; K]$, the clustering accuracy between true labels y and predicted labels $\hat{y}$ for n data points is

$$ACC(y, \hat{y}) = \max_{perm \in P} \frac{1}{n} \sum_{i=0}^{n-1} 1(perm(\hat{y}_i) = y_i)$$

The set of all permutations can be efficiently computed using the Hungarian algorithm (Papadimitriou and Steiglitz, 1998).

The Bayesian Information Criterion (BIC) is a model selection criterion in statistics proposed by Schwarz (1978). This criterion is especially helpful in determining the number of clusters for clustering algorithms that require such a parameter beforehand. As a rough guide, the higher the complexity of the data, the more clusters are needed to represent the data. Scott Shaobing Chen and Gopalakrishnan (1998) proposed to choose the number of clusters by optimizing the BIC, where each clustering is evaluated by its BIC value and the clustering with the highest BIC value is chosen. Scott Shaobing Chen and Gopalakrishnan (1998) showed that the Bayesian Information Criterion for a clustering $C_k = \{c_i : i = 1, ..., k\}$ with k clusters of a dataset $\{x_i \in \mathbb{R}^d : i = 1, ..., N\}$, where each cluster c_i can be modelled as a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \mathbf{x_i})$ and has therefore $d + \frac{1}{2}d(d+1)$ parameters, can be formulated as:

Definition 10. *Bayesian Information Criterion (BIC)*

$$BIC(C_k) = \sum_{i=1}^{k}\{-\frac{1}{2}n_i log|\boldsymbol{\Sigma}_i|\} - Nk(d + \frac{1}{2}d(d+1))$$

The Silhouette Coefficient (Rousseeuw, 1987) is another internal metric that allows a visual representation and interpretation of a clustering result. Rousseeuw (1987) proposed the Silhouette Coefficient as a new graphical display for partitioning and clustering methods. Each cluster is represented by a silhouette, a plot of all Silhouette Coefficients ranked in decreasing order for all objects in the respective cluster. The Silhouette Coefficient for each data point is computed using the average dissimilarity of a data point p to all other data points within the same cluster $a(p)$ and the minimum average dissimilarity of the same data point to all other data points in all other clusters $b(p) = \min_{C \neq A} d(p, C)$. The Silhouette Coefficient of a data point p is therefore defined as:

Definition 11. *Silhouette Coefficient*

$$s(p) = \begin{cases} 1 - \frac{a(p)}{b(p)} & \text{if } a(p) < b(p) \\ 0 & \text{if } a(p) = b(p) \\ \frac{b(p)}{a(p)} - 1 & \text{if } a(p) > b(p) \end{cases}$$

The authors noted that this metric is particularly useful if the dissimilarities can be measured on a ratio scale as in the case of the Euclidean distance to identify compact and clearly separated clusters (Rousseeuw, 1987).

In this thesis, the Bayesian Information Criterion (BIC) (Definition 10) is primarily utilized for model selection and the Silhouette Coefficient (Definition 11) for parameter evaluation, see Chapter 4. For example, comparing Gaussian mixture models with different parameters and selecting the model with the highest BIC score.

The evaluation of clustering algorithms in Chapter 5 and Chapter 6 is mainly based on the Adjusted Rand Index (ARI) and the Cluster Accuracy (ACC), see Definition 8 and Definition 9.

A more comprehensive overview of clustering metrics for hyper-parameter optimization for spatio-temporal clustering, specifically with DBSCAN and HDBSCAN, is given in the master thesis by Weber (2019), supervised within the context of this doctoral research.

Chapter 3

Related Work

Following the introduction of the necessary concepts in Chapter 2, this chapter presents related work directly connected to the research of this thesis. The presented overview is by no means exhaustive but focuses on key scientific work that has been built on. The related work presented in this chapter has been published in parts in Ertl et al. (2019, 2020, 2021a,b).

3.1 Unsupervised Clustering of Spatio-Temporal Data

Unsupervised learning methods such as cluster analysis are instrumental in the analysis of large amounts of data since it allows domain experts to consider groups of objects rather than individual objects and to focus on a higher-level representation of the data (Wang et al., 2013). Unsupervised clustering of spatio-temporal data is an active research area analysing spatial and temporal data at a higher level of abstraction by grouping data points according to their similarity into meaningful clusters. Depending on the spatio-temporal datatype and the analysis to conduct, as detailed in Section 2.1, a variety of clustering algorithms have been proposed providing solutions for different aspects of spatio-temporal data clustering. Most relevant literature for the work of this thesis address the following problem definitions.

3.1.1 Clustering Moving Objects

The problem definition of analysing moving spatio-temporal objects can be compared to the presented approach of analysing the evolution of geo-referenced distributions over time in Chapter 4. Maciąg (2017) published a comprehensive survey on data mining methods for clustering complex spatio-temporal objects, noting that current

approaches often leverage variations of well-known methods and algorithms that have been modified to operate on spatio-temporal data. The clustering methods summarised by Maciąg (2017) range from clustering spatio-temporal events, polygons (geographical areas), trajectories to clustering moving objects. In this context, Gaussian mixture models have been proposed, for example, for image matching (Greenspan et al., 2001) or to identify dynamic clusters in spatio-temporal data (Paci and Finazzi, 2018).

Greenspan et al. (2001) proposed a transition of the image pixels to coherent regions in feature space via Gaussian mixtures to apply a probabilistic measure of similarity between the Gaussian mixtures. The number of mixture components is chosen according to the minimum description length (MDL) principle (Rissanen, 1978; Grünwald, 2007) that chooses the simpler model, i.e. the model with the least number of free parameters with regards to the number of mixture components, over more complex models. Similar images are identified by the Kullback–Leibler divergence (Kullback and Leibler, 1951) of their mixing components. Greenspan et al. (2001) evaluated their methodology theoretically and experimentally in computer vision and discussed their approach for audio segmentation. However, the authors noted that a Gaussian model is a suitable representation for homogeneous regions with an ellipsoid-like shape but a Gaussian distribution poorly represents non-convex regions.

Paci and Finazzi (2018) proposed a dynamic model-based clustering method for spatio-temporal data with finite mixtures of Gaussians introducing spatio-temporally varying mixing weights to accommodate space-time dependence. These adjusted mixing coefficients allow assigning similar cluster membership probabilities to data points at nearby locations and at consecutive points in time. Additionally, Paci and Finazzi (2018) deploy a state-space model to describe the temporal evolution of the locations belonging to each cluster. The authors showed that their approach is very flexible and allows clusters identification with geo-referenced time-series even affected by missing data. The model further allows prediction of the cluster membership probability of data points and any observed location and any time, including future predictions. However, the authors also noted that their approach needs to be extended from the univariate setting to a multivariate setting.

Jin et al. (2005) presented a clustering system based on the Gaussian mixture model with independent attributes within clusters. The main concept is to modify the Expectation–maximization algorithm (EM) to generate a mixture from the summary statistics of a set of subclusters of the given dataset (Huidong Jin et al., 2005). The authors conducted a series of experiments on synthetic and real-life datasets and showed that their approach can run faster and can generate much more accurate clustering results than the random sampling EM algorithm.

Kalnis et al. (2005) provided a formal definition of moving clusters and presented three algorithms based on the DBSCAN clustering algorithm to identify moving clusters over a period of time. Moving clusters will be identified over consecutive time slices if the ratio of their intersect density to their joint density is greater or equal to a specified threshold. The author's experimental results demonstrated that their methods are suitable for large datasets with varying object distribution and agility. The different proposed algorithms can either be used to achieve the exact or approximate identification of moving clusters.

3.1.2 Time-Series Clustering

Time-series clustering is a well-established and active research field across different application domains, for example, in industry, biology, energy, medicine, finance or climate. Multiple surveys provide a clear and structured overview of past and current research in time-series clustering and its subdomains whole time-series clustering, subsequence time-series clustering and time point clustering (Warren Liao, 2005; Aghabozorgi et al., 2015; Zolhavarieh et al., 2014).

Trajectories can be described as sets of measurements that are measured as a function of an independent variable, typically time, where each individual trajectory measures a possible multidimensional response variable (Gaffney and Smyth, 1999), see Section 2.1. Following the given notion of trajectory data, there is no distinction to time-series data in general, however, the data records per individual trajectory can frequently be too short to be amenable to conventional time-series modelling techniques, which requires specialized approaches (Gaffney and Smyth, 1999).

The most relevant related work for this thesis falls into the category of time point clustering. Zolhavarieh et al. (2014) described time point clustering in their review of subsequence time-series clustering as

> "[...] the clustering of time points on the basis of a combination of their temporal proximity and the similarity of their corresponding values. This approach is similar to time-series segmentation. However, time point clustering is different from segmentation in the sense that all points do not need to be assigned to the cluster; that is, some of [the] points are considered noise."

Following this distinction, without the differentiation on noise points, i.e. points that do not belong to any cluster, algorithms and methods developed for subsequence clustering are inter-comparable to those developed for time point clustering in terms of extracting similar segments from individual time-series. Zolhavarieh et al. (2014)

provided a comprehensive overview of methods for subsequence time-series clustering, especially within the context of the discussion if any method can produce meaningful results at all or if all methods for subsequence time-series clustering are actually meaningless (Keogh and Lin, 2005).

For the purpose of subsequence time-series clustering, time-series have to be segmented and clustered in a way that the temporal proximity of time points is taken into account and multiple segments can belong to the same cluster. Since the traditional Euclidean distance metric as a similarity measure for clustering algorithms is not taking the order of the data points into account, a similarity measure called Dynamic Time Warping (DTW) has been proposed by Sakoe and Chiba (1978) and has been improved over time, for example by averaging a set of sequence to be used with similarity-based methods like k-means (Petitjean et al., 2011).

Besides distance-based algorithms, model-based clustering algorithms for time-series have been proposed as well. Hallac et al. (2017) used graph representations for time series subsequences from Markov random fields (MRF) to group similar sequences into clusters, called Toeplitz inverse covariance-based clustering (TICC). TICC simultaneously segments and clusters the data based on its correlation and has been demonstrated to be able to find structural similarities in real-world data (Hallac et al., 2017).

Also in the field of unsupervised learning and deep unsupervised learning, model-based clustering algorithms for time-series data are subject to recent and continuous research. Zhang et al. (2019) proposed a method for unsupervised salient subsequence learning (USSL) to extract salient subsequence features from time-series called shapelets.

A number of methods can tackle the task of subsequence time-series clustering for univariate time series (Warren Liao, 2005), but fall short to discover interpretable clusters for multivariate time series (Ienco and Interdonato, 2020). Specifically, methods merely based on distance metrics such as euclidean distance or dynamic time warping (Berndt and Clifford, 1994) can not capture structural similarities based on correlations across time. For static data, there has been a growing interest in semi-supervised clustering methods, for example, constrained clustering, where additional a priori information or domain knowledge is incorporated into the clustering process, to capture complex relations between features better (Pourrajabi et al., 2014; Basu et al., 2008; Dinler and Tural, 2016).

3.2 Semi-Supervised Clustering

While many advances in clustering algorithms have been made for spatio-temporal data (Maciąg, 2017; Wang et al., 2013) many proposed methods lack the exploitation of available a priori knowledge that might improve the output quality (Ertl et al., 2019). Especially semi-supervised learning clustering algorithms, which incorporate a priori knowledge into the clustering process, can improve the quality of the results (Dinler and Tural, 2016). According to Aggarwal and Reddy (2013) much of early work in semi-supervised clustering focused on extending feature-based clustering methods with regards to two types of semi-supervision, pointwise semi-supervision and pairwise semi-supervision.

3.2.1 Pointwise Semi-Supervision

Pointwise semi-supervision describes the process where the availability of cluster labels for a small number of points can guide the clustering method (Aggarwal and Reddy, 2013; Basu et al., 2002). Basu et al. (2002), for example, used partially labelled data to generate seed clusters for the initialization of the k-means algorithm, namely Seeded-KMeans. The initial centres for clusters are the mean values of the seed points. In the subsequent steps in the algorithm, the seed labels are not further taken into consideration, i.e. are disregarded.

Different from the Seeded-KMeans algorithm, the Constrained-KMeans algorithm uses the partially labelled data to initialize the cluster centres and also uses the seed labels in the subsequent k-means algorithm, i.e. the seed labels are kept unchanged and only the labels of the non-seed data are re-estimated (Basu et al., 2002). Apparently, cluster labels for pointwise semi-supervision can be converted to pairwise constraints in the form of "must-link" and "cannot-link" constraints, detailed in the next section. Data points with the same label define a pairwise "must-link" constraint, while data points with different labels define a pairwise "cannot-link" constraint.

3.2.2 Pairwise Semi-Supervision

Pairwise semi-supervision describes the process where "must-link" and "cannot-link" constraints between some pairs of points are available to guide the clustering method (Aggarwal and Reddy, 2013). "Must-link" constraints specify that two instances have to be in the same cluster, where "cannot-link" constraints specify that two instances cannot be in the same cluster (Wagstaff and Cardie, 2000).

For example, the COP-KMeans algorithm (Wagstaff et al., 2001) follows the k-means algorithm but assures that a point will only be assigned to its nearest cluster if none of

its constraints is violated; otherwise the algorithm aborts. Wagstaff et al. (2001) showed significant improvements in accuracy with random constraints on different datasets. They suggested optimising the order-sensitive assignment of instances to clusters due to the given constraints and relax constraints, moving from hard to soft constraints.

Another algorithm that generalizes the k-means algorithm to handle constraints is the Constrained Vector Quantization Error (CVQE) algorithm (Davidson and Ravi, 2005) and its linear-time variant LCVQE (Pelleg and Baras, 2007). The authors of the CVQE algorithm introduce a new differentiable error function, the constrained vector quantization error. The algorithm minimizes this CVQE in the first step and updates the cluster centroids accordingly in a second step. According to the authors, this algorithm results in faster convergence and the satisfaction of a vast majority of constraints.

Basu et al. (2004) further proposed a cost function for pairwise constrained clustering equivalent to the configuration energy of a Hidden Markov Random Field (HMRF) with a well-defined potential function and noise model. Basu et al. (2004) showed that the pairwise constrained clustering problem could be solved by finding the HMRF configuration with the highest posterior probability, i.e. minimizing its energy, and proposed the PCK-means algorithm for solving this problem.

3.3 Cluster Analysis of Climate Data

While there are many examples that use cluster analysis in the domain of climate research, this section focuses on the most relevant methods that have been demonstrated to be valuable in this area.

One of the first comprehensive methods for cluster analysis of climate data has been introduced by Gaffney and Smyth (1999), who proposed a probabilistic mixture regression model applying the Expectation–maximization algorithm (EM) to cluster trajectories and demonstrated their approach analysing extratropical cyclones (Gaffney et al., 2007). Gaffney et al. (2007) applied curve-based mixture models to perform probabilistic clustering of wintertime North Atlantic extratropical cyclone trajectories in latitude-longitude space. The authors were able to identify three groups of tracks oriented south-to-north, southwest-to-northeast and west-to-east predominantly. They were also able to associate common attributes for each group in their simulation and reanalysis datasets and associate the cyclone-track clusters in the reanalysis dataset with well-defined anomalies in sub-weekly storm track variance and well-known low-frequency teleconnection patterns.

However, since clustering whole trajectories can overlook common behaviour in partial segments of the trajectories, Lee et al. (2007) proposed a partition-and-group framework and a trajectory clustering algorithm called TRACLUS, which they demonstrated among others in the field of climate research for hurricanes landfall forecasts. TRACLUS partitions a trajectory into a set of line segments at characteristic points and groups similar line segments in a dense region into a cluster. Lee et al. (2007) were able to identify seven representative trajectories in their hurricane track dataset, which the authors correlated with known hurricane movement patterns.

Another approach by Birant and Kut (2007) used a spatio-temporal extension to DBSCAN and demonstrated the applicability of the algorithm by clustering regions with similar seawater characteristics from station and satellite data. Birant and Kut (2007) extensions to the DBSCAN algorithm introduce two additional parameters, an ϵ distance parameter for the spatial attributes and a threshold parameter $\Delta\epsilon$ to keep the cluster values close to the cluster attributes mean value. The authors could discover clusters of data points with similar sea surface temperature characteristics and significant wave height values for their task to discover the regions that have similar seawater characteristics.

Steinhaeuser et al. (2011) provide a comprehensive comparative study on different clustering algorithms and different regression algorithms for their predictability on climate indices, i.e. time-series that summarise variability at local or regional scales with relation to other events. The authors compared clustering methods based on their ability to predict climate variability and demonstrated that the network-based indices have significantly more predictive power. The clustering algorithms under investigation are network communities based on the method by Pons and Latapy (2005) to compute communities in large networks using random walks, the k-means algorithm, the Partition Around Medoids (PAM) algorithm, the k-medoids algorithm by Kaufman and Rousseeuw (2009), the spectral clustering implementation by Ng et al. (2001) and the Gaussian Mixture Model (GMM) as an Expectation–maximization algorithm (EM) implementation. The empirical comparison of clustering methods by Steinhaeuser et al. (2011) shows that specifically community detection in climate networks stands out among competing methods as the superior approach across a diverse range of test cases. According to Steinhaeuser et al. (2011), reinforcing the notion that networks can capture the complex relationships within the global climate system effectively.

Chapter 4

Analysing the Evolution of Geo-Referenced Distributions over Time

This chapter mainly addresses challenges one and two defined in Chapter 1, namely learning the natural structure of spatio-temporal data reflected by meaningful clusters by analysing existing clustering algorithms to cluster spatio-temporal data and to identify correlated structures in the dataset. The presented work establishes a data-driven approach of tracking clusters in spatio-temporal data based on the Gaussian Mixture Model (GMM) to extract cluster properties that can be analysed for changes over space and time. The methodology is based on well-known algorithms, and an interpretation of the algorithmic results is given in a spatio-temporal context. The presented method and results in this chapter are part of the initial work during the course of this thesis and have been published in parts in Ertl et al. (2019).

4.1 Motivation

The simultaneous observation of multiple variables or measurements at geo-referenced locations or areas over time enables the analysis of multivariate distributions of spatio-temporal data. In particular remote sensing data, i.e. data provided by remote sensors, exhibit these properties. For example, a single measured infrared emission spectrum at a given location and time can be further processed to estimate the content of different trace gases and water vapour as well as the temperature in the vertical column of the atmosphere over this particular location and at the given time. By looking at multiple multivariate observations, scientists can analyse multivariate distributions that can evolve over space and time.

In this context, cluster analysis is especially suited to discover and track multivariate distributions over time. The problem definition in the literature most closely related to these tasks is formulated as the analysis of moving spatio-temporal objects, see Section 3.1. The following presents a data-driven approach to tracking clusters in spatio-temporal data. This approach is based on the Gaussian Mixture Model (GMM), see Section 2.4.2, to extract cluster properties that can be analysed for changes over space and time. A methodology based on well-known algorithms is provided and an interpretation of the algorithmic results in a spatio-temporal context is presented.

4.2 Data-Driven Approach

The proposed data-driven approach to track clusters and their changing properties over space and time in spatio-temporal data based on GMM can be summarised in four steps as follows.

1. **Splitting the data in spatial regions of interest**
   This first step needs careful consideration if splitting the data significantly affects the mixture components by including or excluding specific observations ("edge-effects"). Also, splitting the data into spatial regions of interest (ROI) might not always be applicable, for example, if all the clusters are tracked within one spatial area. However, if the nature of the dataset allows the statistical evaluation and mitigation of edge-effects, for example, by assuming or proving complete spatial randomness (Diggle, 2014), and if the analysis is conducted over multiple spatial areas, this step significantly increases the scalability of the model. Each mixture model for any ROI can be computed in parallel, which decreases the overall runtime, see Section 4.3.2. Selection criteria for the size of the ROI typically depend on domain knowledge or the premise of the analysis.

2. **Modelling the observed data in each ROI with mixtures of Gaussians**
   A range of Gaussian mixture component models with a different maximum number of mixture components is fitted to the data for each spatial region. The model with the lowest Bayesian Information Criterion (BIC) score determines the number of clusters for the most suitable mixture model, see Section 2.4.4. An alternative approach to infer the number of clusters of the most suited mixture model is the variational Bayes approach (Attias, 1999). This approach, however, requires additional fine-tuning of hyperparameters and introduces additional computational overhead.

3. **Extracting cluster parameters for each Gaussian mixture component**
   Following a data summarisation approach (Nassar et al., 2004), the ellipsoid properties of the multivariate Gaussian distribution, the centre and principal axes given by the mean and length of the eigenvectors of the covariance matrix, as well as the polar angle of the major axis are extracted. These properties most prominently describe the underlying distribution. However, different cluster properties can be selected according to the nature of the data and the expected behaviour of spatio-temporal changes of the clusters.

4. **Comparison of cluster parameters for spatio-temporal changes**
   This last step can be performed by a variety of suitable methods, which demonstrates the flexibility of the proposed model. For the model evaluation and results in the following experimental studies and evaluation, the DBSCAN clustering algorithm (see Section 2.4.3) is used on the extracted cluster properties to find similar clusters and account for spatio-temporal changes. By comparing the extracted cluster parameters between clusters in different spatial regions over time, similar clusters can be identified, and further occurrences can be reviewed to investigate any underlying motion.

4.3 Evaluation

This section details the experimental studies to evaluate the proposed model on a synthetic and a real dataset. First, the setup for conducting the experimental studies with three main objectives is given. The main objectives are (1) to track moving clusters, (2) to track emerging clusters and (3) to track changing clusters over space and time. Second, the results for each objective are presented together with runtime measurements for sequential and parallel computations.

4.3.1 Setup

Synthetic Data

The synthetic dataset is generated of similar size and structure as the real-world dataset but with already known distributions in feature space. The spatial extent is given in latitude and longitude for global coverage, and the temporal extent comprises six consecutive days equal to the real-world dataset. The spatial points are generated from a *N-conditioned Complete Spatial Randomness (CSR) process* (Diggle, 2014) to mimic the remote sensing data generation process of the real-world dataset. A N-conditioned CSR process can be defined as following, see Rey and Anselin (2007).

Remark. N-conditioned Complete Spatial Randomness (CSR) process
Given the total number of events N occurring within an area A, the locations of the N events represent an independent random sample of N locations where each location is equally likely to be chosen as an event.

The two-dimensional feature patterns are well separated isotropic Gaussian blobs that change their scale over time and are assigned to k-means clustered spatial regions. The spatial partitioning in the data generation process into k-means clustered spatial regions is different from the spatial partitioning in the analysis, a regular lattice, to evaluate possible edge-effects better. The number of k-means clusters is equal to the number of spatial grid cells so that clusters and grid cells have similar spatial extent, but some data points at the cluster edges will be assigned to different grid cells during the analysis. The k-means algorithm has been selected since it can form clusters of equal variance with an extent of an individual cluster similar to the extent of an individual Region of Interest (ROI).

Figure 4.1 illustrates the spatial extent of the synthetic dataset for one time snapshot, i.e. one day, with colours indicating the k-means clustered spatial regions and the grid lines indicating the spatial partitioning for the cluster analysis. For each k-means clustered spatial region, three isotropic Gaussian blobs are generated by randomly choosing their Gaussian distribution's mean and standard deviation out of five predefined mean and standard deviation values plus some random noise, summarised in Table 4.1. The number of total distributions is based on the occupancy of the feature space, centre and edges (one + four); the number of selected distributions based on empirical evaluation. Each blob of data points is scaled according to the time snapshot of the data points, i.e. for the first snapshot, the scale factor is one, and for the last snapshot, the scale factor is one point six, simulating an arbitrary 10% rate of expansion per day.

All generated data points following the described process are illustrated in Figure 4.2, five isotropic Gaussian blobs with 960,000 points in total, 160,000 points for each time snapshot.

Table 4.1: Value range, noise and scale for the synthetic dataset.

	Value range	Noise	Scale
μ	[(-5,-5),(-5,5),(0,0),(5,5),(5,-5)]	uniform [0,1) * 0.1	
σ	[0.25,1.0,0.5,1.0,0.25]	uniform [0,1) * 0.01	[1.0, 1.1, 1.2, 1.3, 1.4, 1.5]

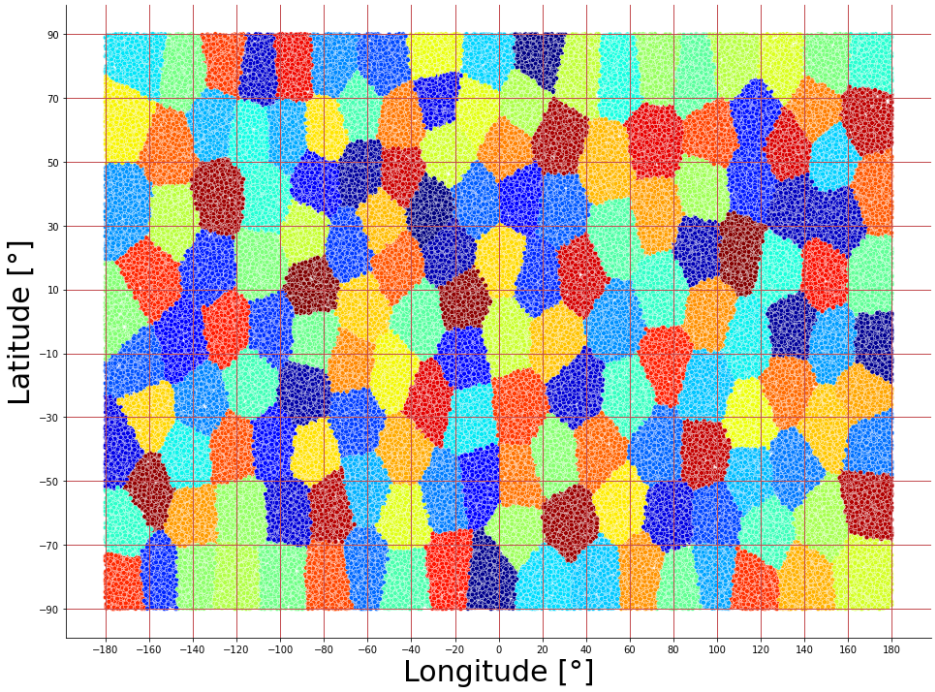


Figure 4.1: Synthetic data for a one-time snapshot with k-means clustered spatial regions and a $20° \times 20°$ grid overlay in accordance with the ROI of the real-world dataset, indicating the spatial partitioning for the cluster analysis.

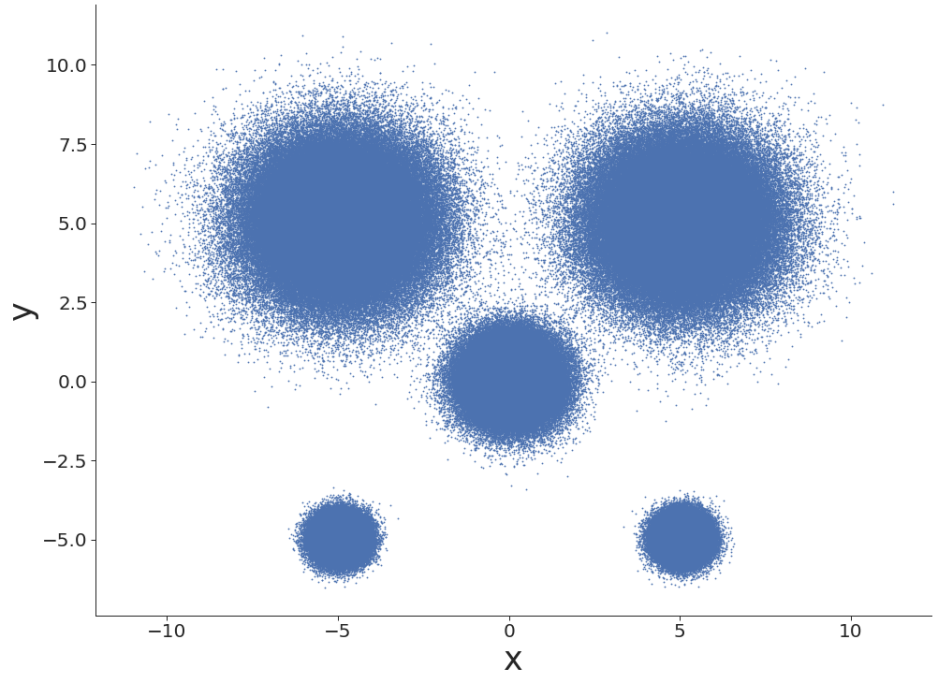


Figure 4.2: Synthetic data in value space; five isotropic Gaussian blobs in total. Each k-means clustered spatial region (see Figure 4.1) is associated with three random blobs scaled according to the temporal order of the snapshot.

Real-World Data

The real-world dataset contains time snapshots from the MUSICA IASI satellite-based remote sensing dataset as described in Section 2.2. The dataset for this experimental study contains global observations for six consecutive days from June 8, 2016, to July 30, 2016, at around five kilometer height, that have been processed according to the MUSICA IASI retrieval process.

Figure 4.3 shows an example snapshot of one day of global observations with 383,591 individual observations after filtering out cloudy and partly cloudy observations as well as observations with bad quality. Each daily dataset is partitioned into $20° \times 20°$ longitude and latitude Region of Interest and clustered in the $\{H_2O, \delta D\}$ value space.

Figure 4.4 shows an example $\{H_2O, \delta D\}$ pair distribution for a single $20° \times 20°$ grid cell as indicated by the red grid lines in Figure 4.3.

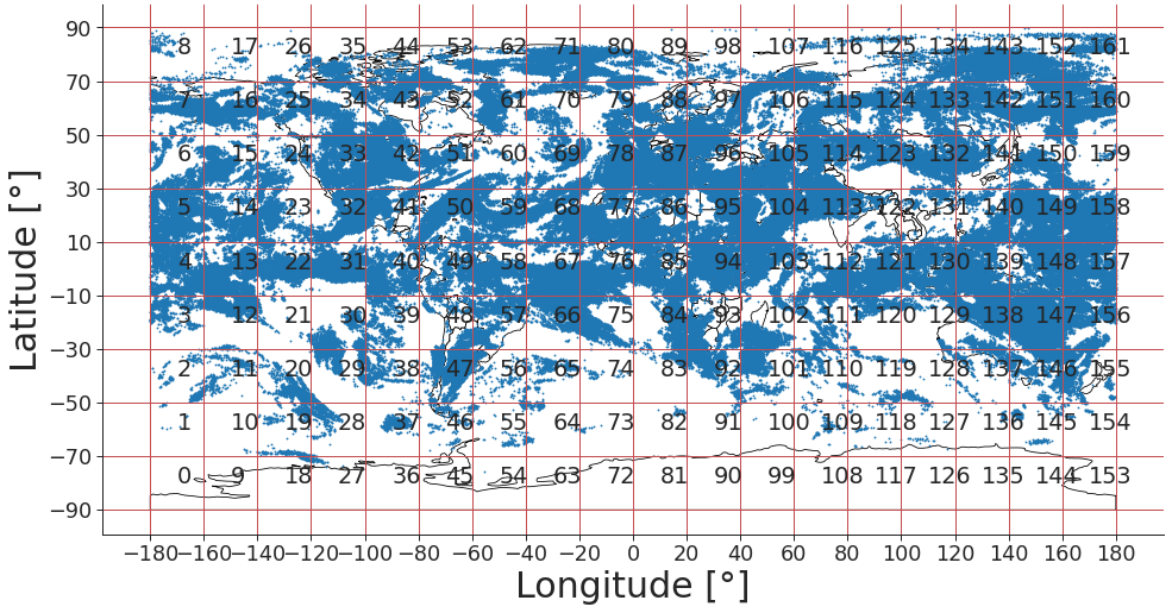


Figure 4.3: Real-world data from the MUSICA IASI dataset of 383,591 geo-referenced observations at {Longitude,Latitude} for a one-time snapshot/day (2016-06-08) with a $20° \times 20°$ grid overlay indicating the spatial partitioning for the cluster analysis.

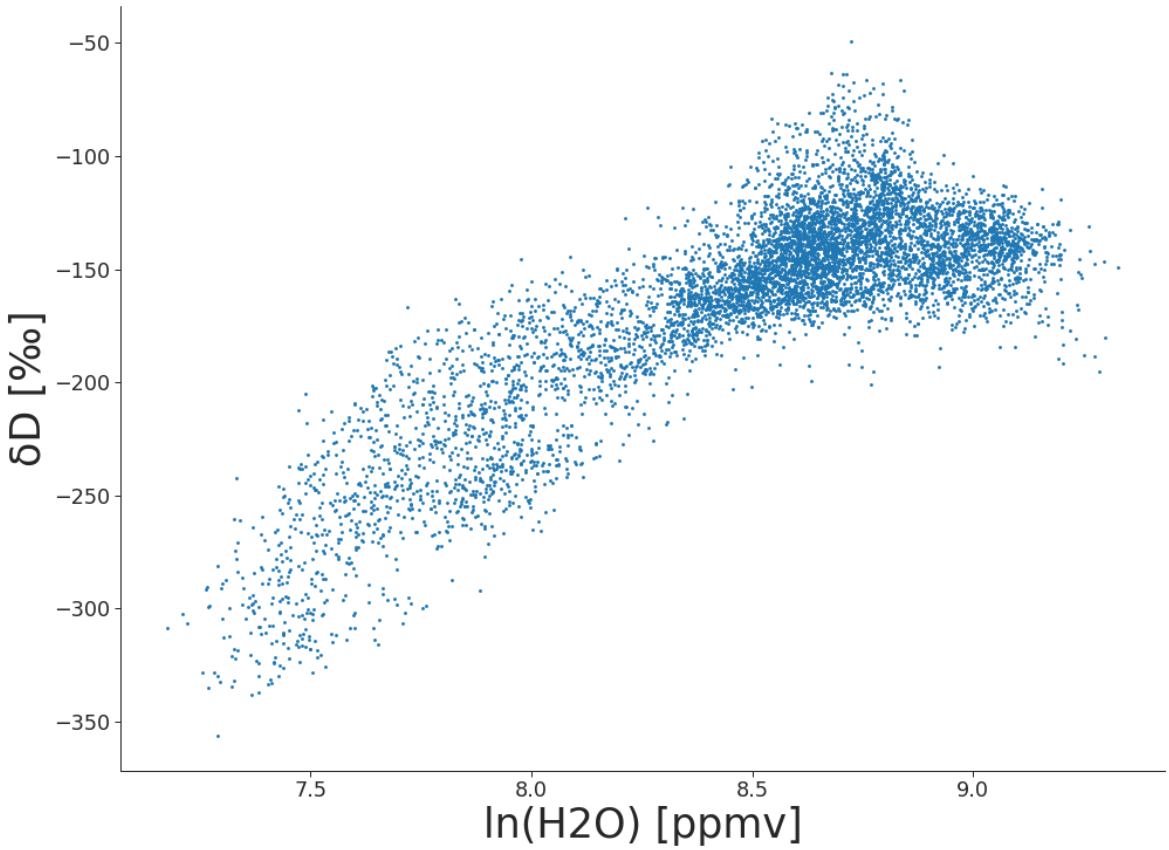


Figure 4.4: Real-world data in value space ($\{H_2O, \delta D\}$ pair distribution) for a random single 20° × 20° grid cell (#77 from Figure 4.3) with 6,779 data points.

Application

The proposed method described in Section 4.2 is applied according to the presented setup for each individual step for the synthetic and real-world dataset as follows.

1. **Splitting the data in spatial regions of interest**
   Synthetic and real data is split into geospatial regions on a regular grid with a grid size of 20 x 20 degrees for longitude 180 degrees West to 180 degrees East and latitude 90 degrees South to 90 degrees North. The specific size of the ROI is postulated by the domain expert. This step results in 162 ROIs, which is imposed on the real and synthetic dataset.

2. **Modelling the observed data in each ROI with mixtures of Gaussians**
   For each spatial region, multiple Gaussian mixture models are fitted to the data with at least two observations per cluster and a maximum number of ten components. Each model is evaluated according to its BIC score, and the model with the lowest BIC score is selected as the best model. The maximum number

of ten components has been selected out of multiple experimental runs, where the trade-off between searching for models with a higher number of components and lower BIC scores and the number of observations per cluster has been determined.

3. **Extracting cluster parameters for each Gaussian mixture component**
   For each Gaussian mixture component identified in Step 2, the ellipsoid properties of the multivariate Gaussian distribution have been extracted, namely the centre and principal axes given by the mean and eigenvectors of the covariance matrix as well as the angle of the major axis. More specifically, the major and minor axis length and the axis angle in degrees for the bivariate contour where 95% of the probability falls has been extracted. These ellipsoid properties have been proven to be most descriptive of the underlying distributions.

4. **Comparison of cluster parameters for spatio-temporal changes**
   The comparison is conducted using the DBSCAN algorithm with a maximum distance between two samples ϵ and the number of minimum points $minPts$ in each ϵ-neighbourhood. Each ϵ-neighbourhood is defined by the specified radius ϵ and the number of minimum points ($minPts$) within ϵ from a point under consideration so that this point can be identified as a core point. Details on the DBSCAN algorithm have been introduced in Section 2.4.3. The DBSCAN parameters have been assessed and evaluated on the data through empirical analysis.

Tracking Moving Clusters

Tracking moving clusters can be achieved by looking at all GMM clusters that DBSCAN has assigned to the same group with differences in time and space. Such differences have to be defined a priori to identify moving clusters, for example clusters that appear in neighbouring grid cells with a defined time delay. For the evaluation of this experiment a time lag of one day has been chosen, looking at all neighbouring grid cells (North, Northeast, East, Southeast, South, Southwest, West and Northwest). However, the proposed method allows to analyse similar clusters with different time lags and paths across the imposed lattice as well.

Tracking Emerging Clusters

Emerging and disappearing clusters can be identified by comparing clusters of the same group in the same spatial region over time. This can be achieved by observing cluster occurrences over consecutive time snapshots, where at some snapshots, the clusters are present, and at other snapshots, they are not present. For example, by observing

cluster occurrences over three consecutive days, where a cluster has been identified on the first and third day, but not on the second day.

Tracking Changing Clusters

The proposed approach also allows comparing statistics of cluster properties within the same group and across cluster groups. By looking at the mean, standard deviation, minimum, maximum and percentiles of the identified clusters within the same group and across groups, these statistical measures can provide valuable insight into the variability of cluster properties and possible inference between clusters and associated events.

4.3.2 Results

Synthetic Data

The results after applying the proposed method to the synthetic dataset range from a number of four to 260 clusters with a mean Silhouette Coefficient over all data points between -0.3 and 0.4 respectively, depending on the ϵ and $minPts$ parameters, evaluated for a ϵ range of $[0.1, 1.0]$ with a step size of 0.1 and a $minPts$ range of $[2, 15]$ with a step size of 1; range and step size have been chosen to cover the main variability of the number of clusters and the Silhouette Coefficient. It can be observed that results with a high number of clusters also show the highest number of noise count and occur at the lower end of the ϵ parameter range. Figure 4.5 and Figure 4.6 illustrate the parameter dependencies.

Results with the highest mean Silhouette Coefficient have five distinct groups that represent all of the generated Gaussian distributions following the generation process as described with the parameters summarised in Table 4.1.

The five groups are in accordance with the generation process and can be visualised by plotting the mean ellipsoids defined by the extracted cluster properties, the bi-variate distributions' mean, the major and minor axis length and the axis angle of the 95% contour, illustrated in Figure 4.7.

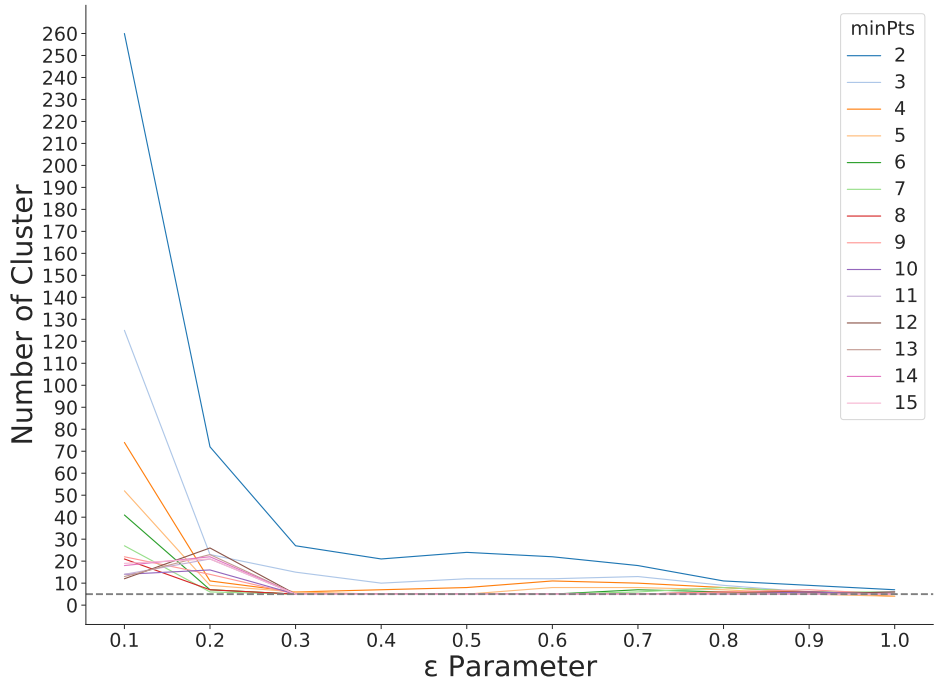


Figure 4.5: Dependencies between ϵ and the number of clusters with colours indicating the *minPts* parameter. The grey dotted line is at five, the number of true cluster.

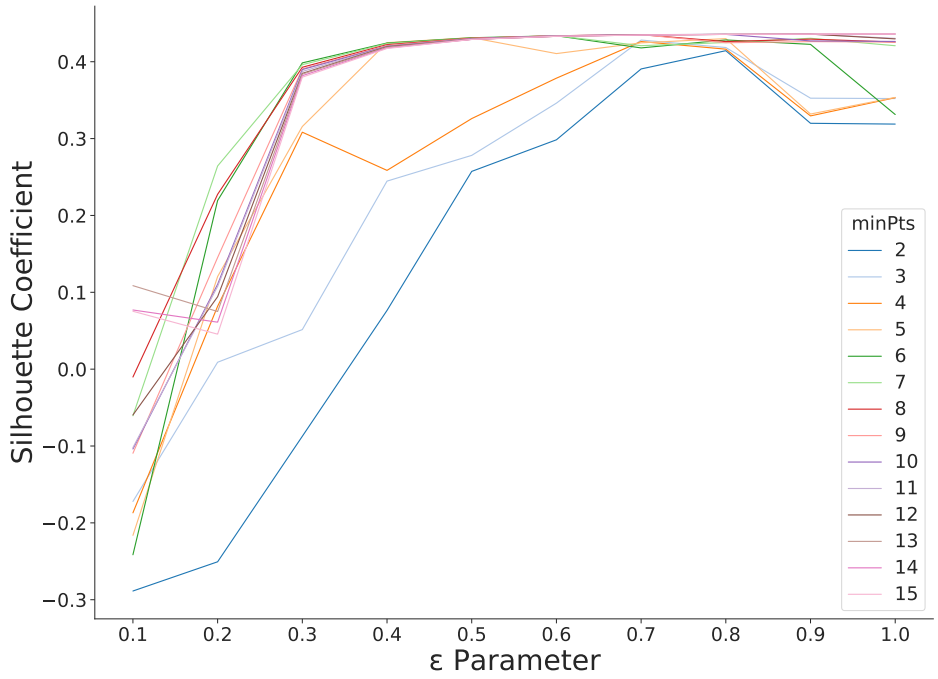


Figure 4.6: Dependencies between ϵ and the Silhouette Coefficient with colours indicating the *minPts* parameter.

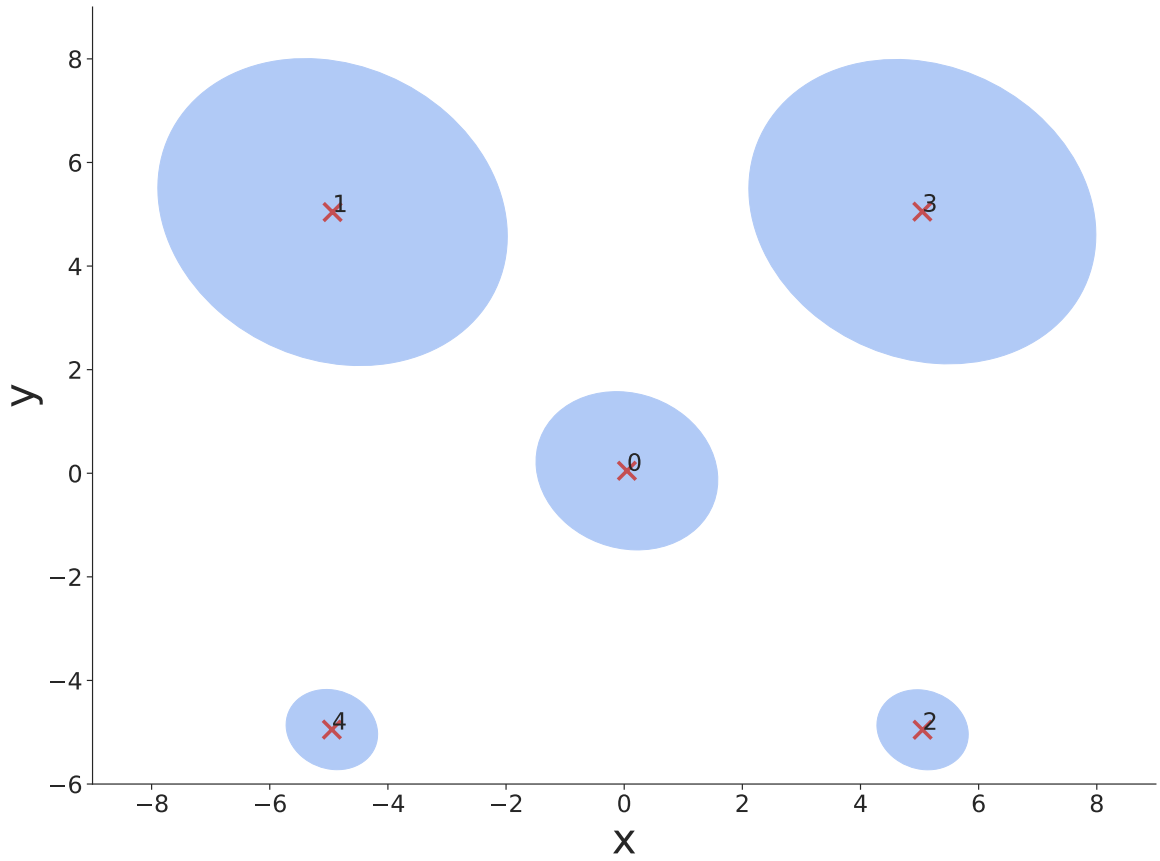


Figure 4.7: Mean ellipsoids of the DBSCAN cluster groups for the synthetic dataset with $\epsilon = 1.0$ and $minPts = 15$.

Real-World Data

Applying the proposed method to the MUSICA IASI dataset yields results ranging from zero to 337 cluster with a mean Silhouette Coefficient over all data points between -1 and 0.6 respectively, depending on the ϵ and $minPts$ parameters, evaluated for a ϵ range of $[0.1, 1.0]$ with a step size of 0.1 and a $minPts$ range of $[2, 15]$ with a step size of 1; range and step size have been chosen to cover the main variability of the number of clusters and the mean Silhouette Coefficient. It can be observed that results with $\epsilon = 0.3$ have the highest number of clusters depending on the $minPts$ parameter, see Figure 4.8. While the mean Silhouette Coefficient continuously increases with an increase of the ϵ parameter, see Figure 4.9, the number of clusters decreases and plateaus around one cluster. Figure 4.8 and Figure 4.9 show that the choice of DBSCAN parameters has to be a tradeoff between the number of clusters and the coefficient of the mean intra-cluster distance and the mean nearest-cluster distance, as measured by the Silhouette Coefficient.

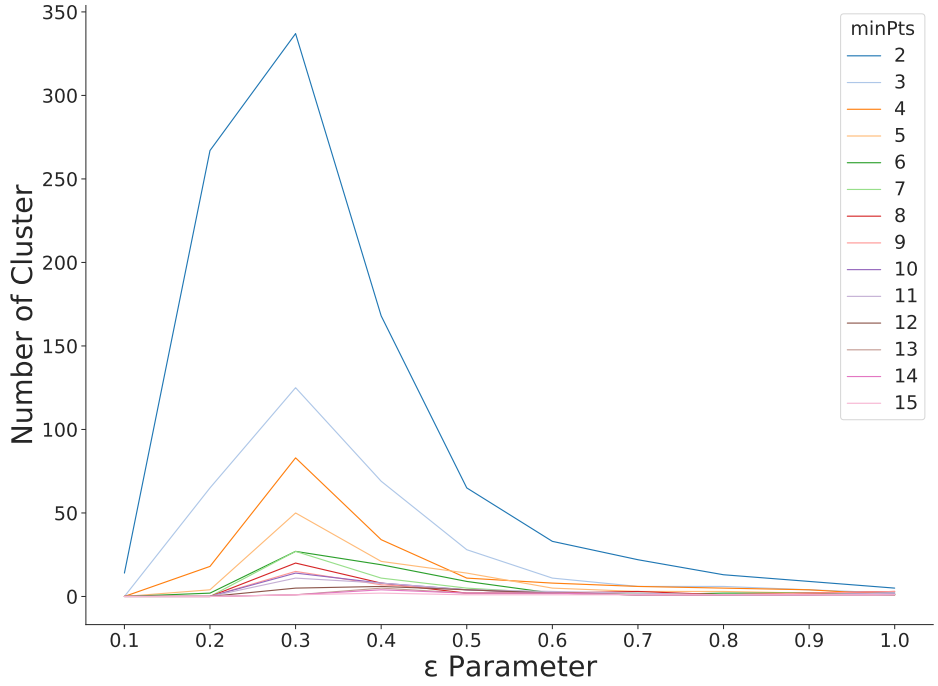


Figure 4.8: Dependencies between ϵ and the number of clusters with colours indicating the *minPts* parameter.

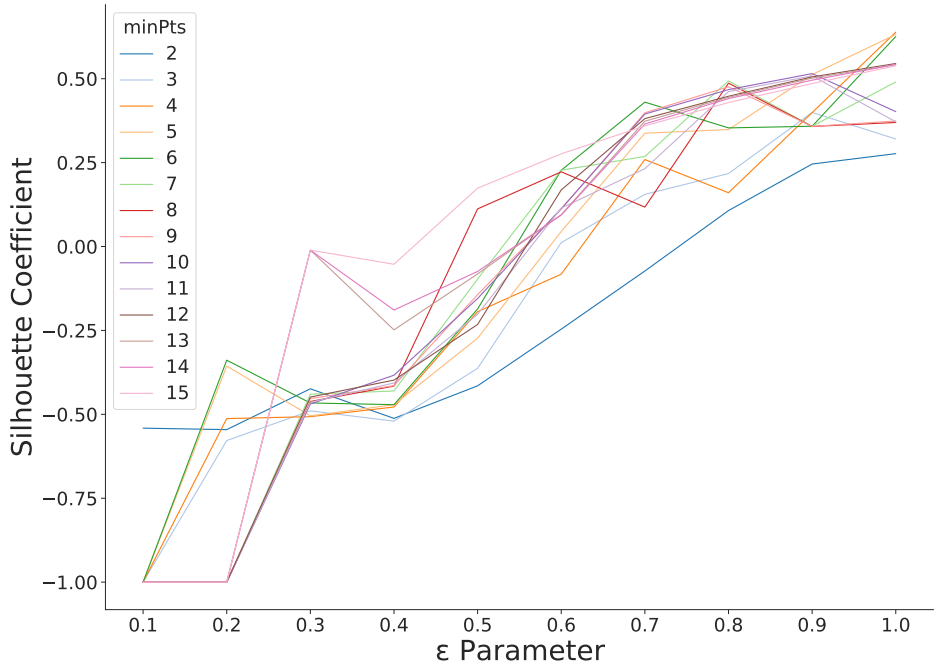


Figure 4.9: Dependencies between ϵ and the Silhouette Coefficient with colours indicating the *minPts* parameter.

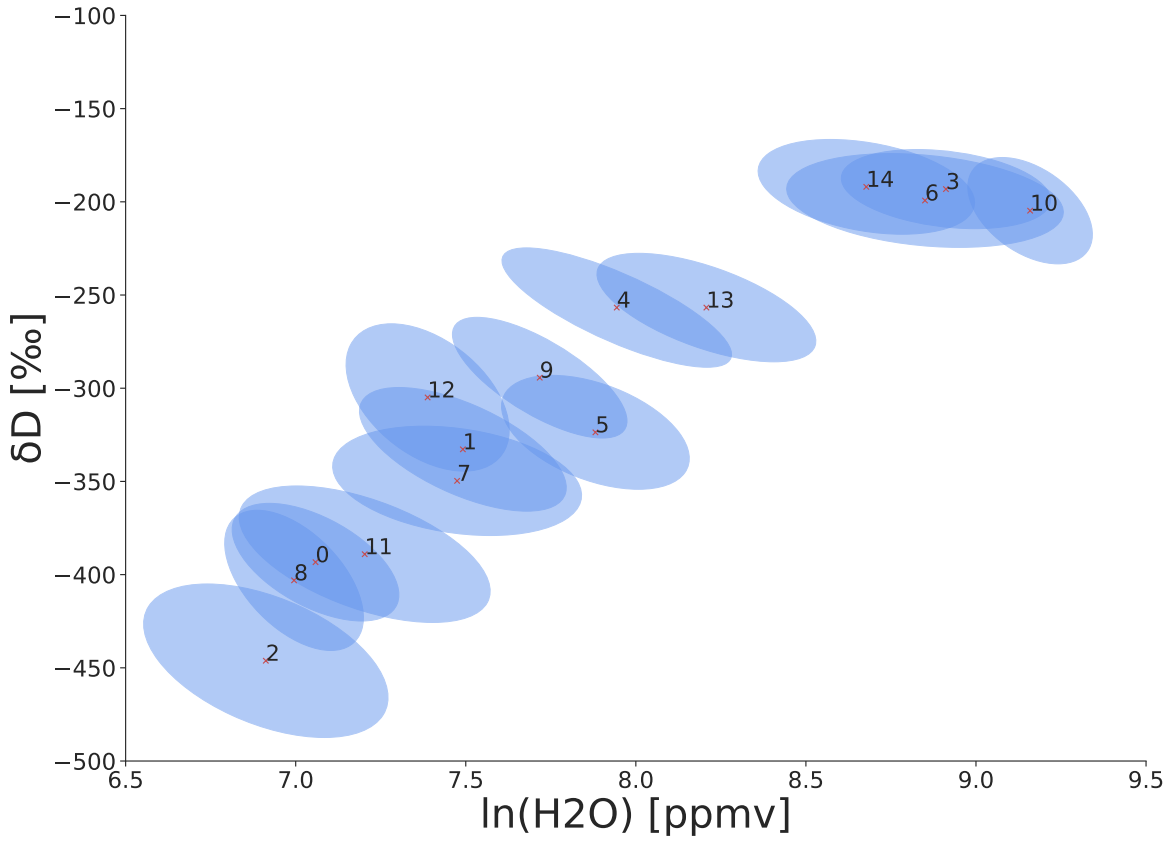


Figure 4.10: Mean ellipsoids of the DBSCAN cluster groups for the real-world dataset with $\epsilon = 0.3$ and $minPts = 9$.

In the case of the real-world dataset, the results with the highest Silhouette Coefficient do not necessarily produce fitting cluster representations for the analysis to conduct, as it is the case for the synthetic dataset. Rather than according to the Silhouette Coefficient, the DBSCAN parameters have to be chosen according to the desired granularity of cluster representation for the data analysis in this case. For the following analysis, a granularity of 15 representative clusters has been chosen that allows to observe moving, changing and emerging clusters as described in the experimental setup.

The 15 clusters can be visualised by plotting the mean ellipsoids defined by the extracted cluster properties, the bi-variate distributions' mean, the major and minor axis length and the axis angle of the 95% contour, illustrated in Figure 4.10.

Runtime Measurements

As stated in Section 4.2 each mixture model for any ROI can be computed in parallel, which decreases the overall runtime significantly. To demonstrate the scalability of the model, tests have been run on both the real-world and synthetic data with an increasing number of CPU cores and an increasing number of maximum mixture components

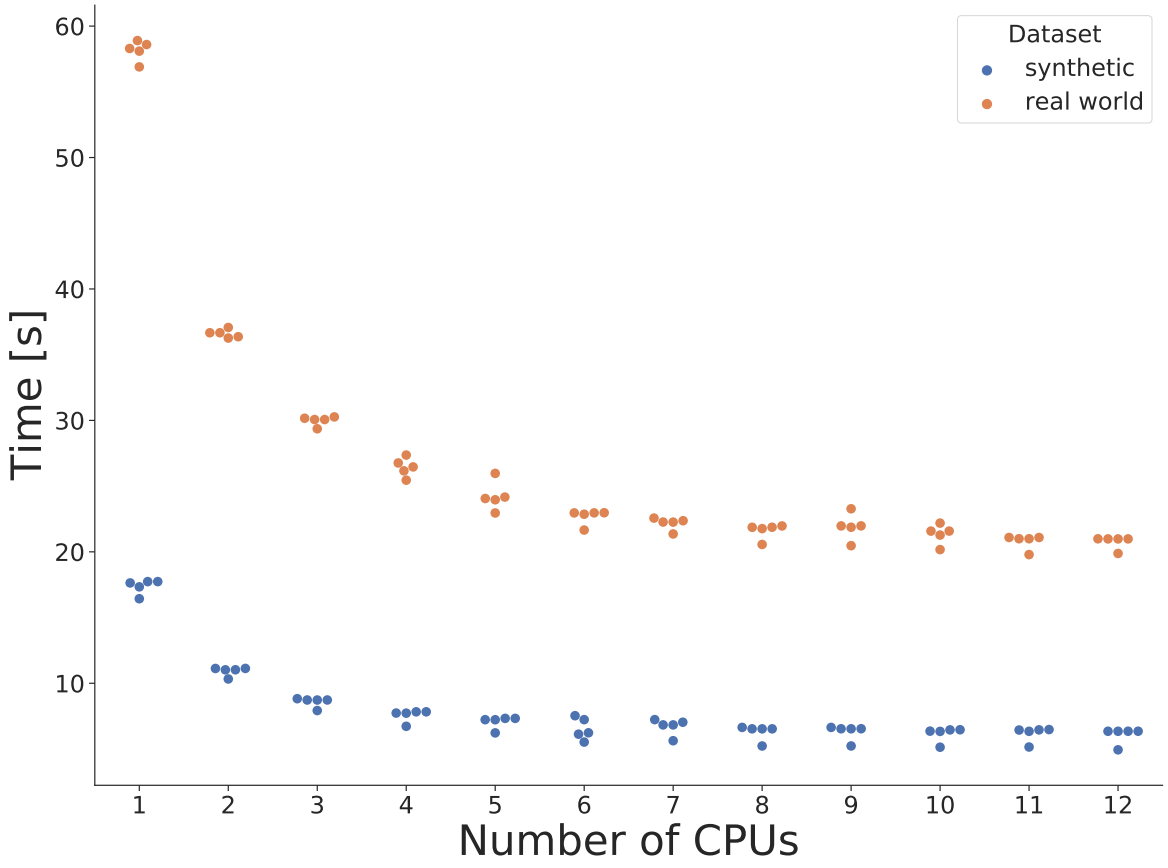
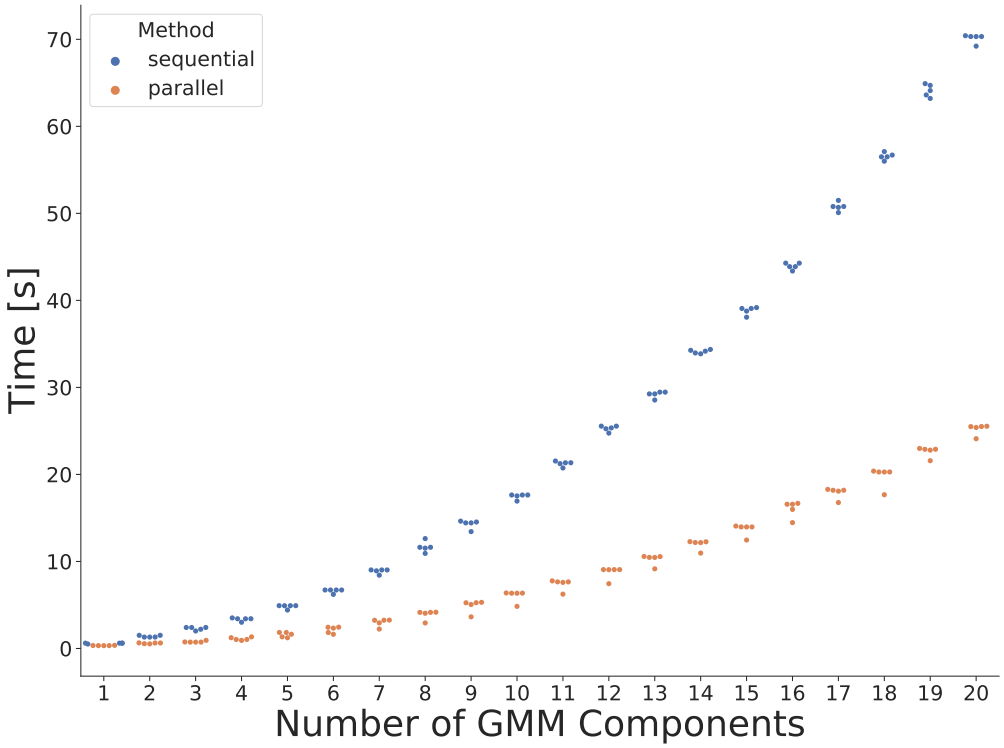


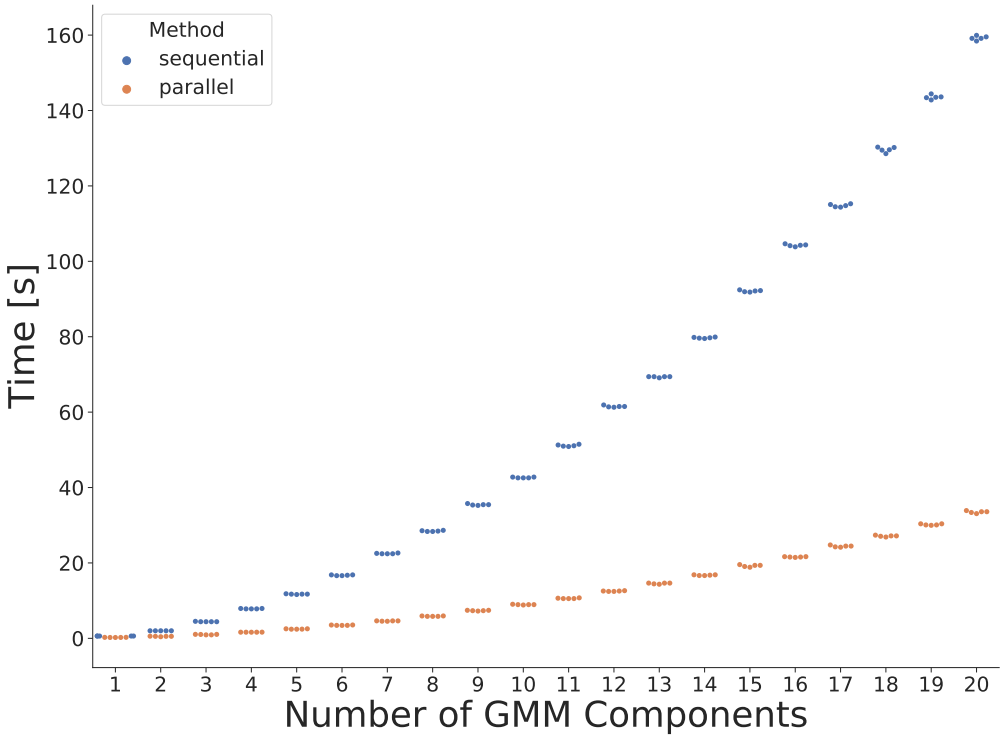
Figure 4.11: Runtime per CPUs for the synthetic and real-world dataset.

sequentially and in parallel. The measurements have been taken on a computer with six Intel© Core™ i7-9750H CPU at 2.60GHz (12 cores in total) and 64 GB main memory.

For the sequential execution of fitting the GMM models to one spatial region after another, one CPU core was busy for up to around one minute for the real dataset and around 20 seconds for the synthetic dataset. Running the GMM fitting for 12 spatial regions in parallel by distributing the work as separate processes on all 12 cores reduced the runtime significantly by at least a factor of three for the real-world dataset and synthetic dataset, providing the same overall results. Figure 4.11 shows the decreasing runtime with an increasing number of CPUs. The number of Gaussian mixture components has been set to ten for all runs. It is clearly visible that the maximum gain has been achieved with four cores while adding more cores decreases the runtime more slowly towards the minimum time required to analyse one single spatial cell. By looking at the runtime measurements, the conclusion can be made that the proposed approach scales well with the number of spatial regions computed in parallel.



(a) Runtime per number of GMM components for the synthetic dataset.



(b) Runtime per number of GMM components for the real-world dataset.

Figure 4.12: Runtime measurements for sequential and parallel runs applying the proposed method to the synthetic (a) and real-world (b) dataset.

Figure 4.12a and Figure 4.12b illustrates the increasing amount of computational time necessary with the increasing number of maximum mixture components, starting from one up to 20 components. While an increasing number of maximum mixture components requires an almost quadratic increasing amount of time, the overall computational time can be significantly decreased by running the GMM fits for each spatial region in parallel.

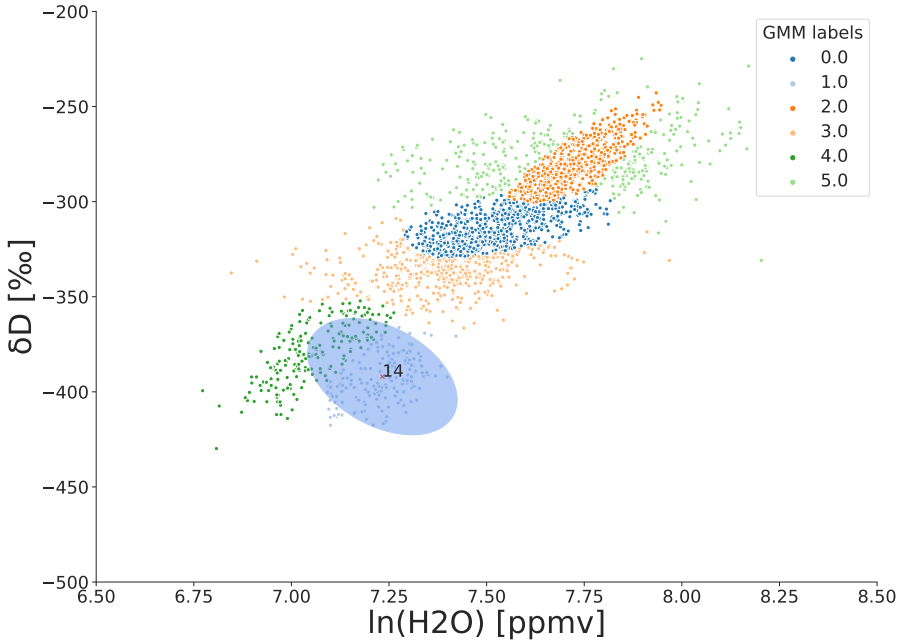
These results highlight the scalability of the approach, which becomes more and more critical with the increasing amount of data that needs to be analysed. In fact, even if splitting the data in spatial Region of Interest (ROI) might not be applicable, analysing sub-samples can be done in parallel as well. By uniform random sampling or biased sampling, the operation of general data mining tasks like clustering can be significantly speed-up (Kollios et al., 2003).

Tracking Moving Clusters

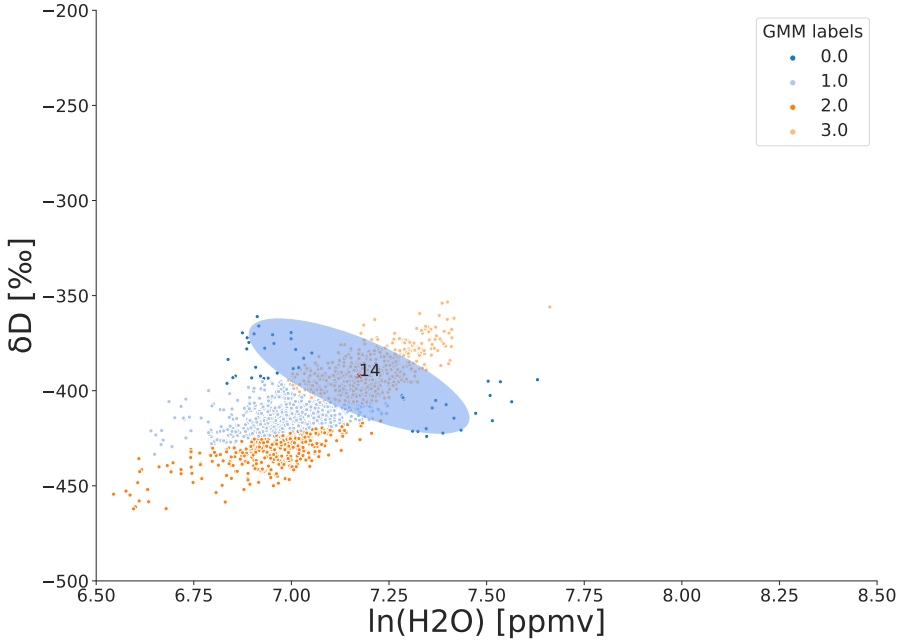
If two GMM clusters are similar according to the result of the DBSCAN clustering and one is present in grid cell A at day one and present in grid cell B, a neighbour grid cell to A, at day two, the assumption can be made that the cluster or cluster generating process has moved from cell A to cell B. The results for the real dataset show multiple moving clusters according to the above definition; specifically, 218 occurrences can be identified that comply with the above definition of moving clusters with the granularity of cluster representation illustrated in Figure 4.10.

Figure 4.13a and Figure 4.13b give one example of two neighbouring grid cells with observations from 2016-06-11 and 2016-06-12, at longitude 40 degrees East to 60 degrees East, latitude 50 degrees North to 70 degrees North and latitude 70 degrees North to 90 degrees North respectively. Both cells contain a specific cluster that DBSCAN has assigned to the same group and which is highlighted with the corresponding ellipses. The presented result has been selected as a representative example for detecting moving clusters between spatial regions. The clusters identified in Figure 4.13a and Figure 4.13b show close similarities; their total absolute deviation of mean, angle and major/minor axis length is below 1.7.

However, the proposed method allows also tracking clusters with varying cluster properties, which might not be immediately apparent. The proposed model is not limited to the definition of moving clusters used in this example. As outlined in 4.3.1, the definition of a moving cluster has been generalised to one spatial neighbour and one time lag, in this case, one day. However, each step in the model can be adjusted to the premise of the analysis.



(a) Observations at grid cell 40°E to 60°E and 50°N to 70°N with colours according to the best GMM fit. The blue ellipse indicates a specific DBSCAN cluster group.



(b) Observations at grid cell 40°E to 60°E and 70°N to 90°N with colours according to the best GMM fit. The blue ellipse indicates a specific DBSCAN cluster group.

Figure 4.13: Observations for two neighbouring grid cells at consecutive days. The blue ellipses represent the same cluster group identified by DBSCAN, indicating the movement of the distribution or the generating process.

For example, imposing different spatial structures and temporal slices (Step 1), varying the number of mixture components (Step 2), applying and adjusting different clustering algorithms for clustering the mixture components properties (Step 3) and analysing the clusters of cluster properties according to the definition of the motion (Step 4), including more complex search patterns such as trajectories across multiple spatial regions with varying timestamps.

Tracking Emerging Clusters

Following the predefinition of emerging and disappearing clusters in Section 4.3.1, 32 occurrences can be identified with the granularity of cluster representation illustrated in Figure 4.10. One example of an emerging cluster is given in Figure 4.14 for the grid cell at 100 degrees West to 80 degrees West and 50 degrees South to 30 degrees South. The observations of three consecutive days are plotted, from top to bottom, with observations belonging to the same cluster group indicated by the blue ellipses in the top and bottom graph. In the middle graph, the observations are not represented by the two distributions in the areas indicated by the red ellipses. This behaviour allows implicating that the cluster generating process is disappearing on day two and reemerging on day three.

Detecting emerging, disappearing and reappearing clusters with varying cluster properties in the real-world climatology data can be a strong indicator for emerging, disappearing and reappearing climatology events. In the presented case, the emerging and disappearing cluster in the $\{H_2O, \delta D\}$ feature space can be associated with atmospheric water transport due to air mass mixing with air masses of distinctive isotopologue fingerprints.

Tracking Changing Clusters

The proposed method allows tracking the evolution of cluster properties within the same cluster group and between different cluster groups. Figure 4.15 illustrates this approach by showing the rate of expansion per day that has been put into place for the synthetic dataset by showing the mean values of the major and minor axis length that is scaled according to the time snapshot of the data.

Trends in the cluster features can be identified by looking at the temporal changes of cluster properties of any cluster group. Converging trends for two different cluster groups can indicate the convergence of the initial separated cluster generating processes over a more extended period of time, while steady time-series of cluster properties can indicate cluster generating processes that stay separated even over a longer period of time.

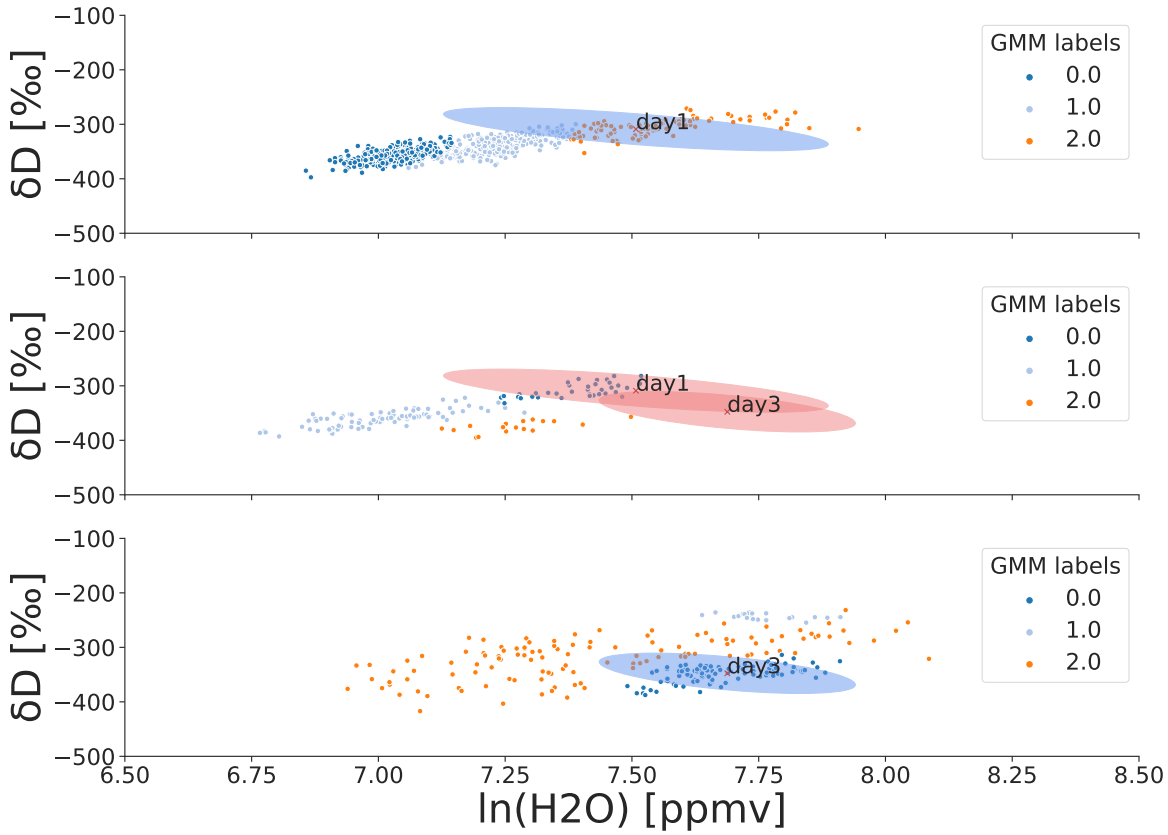


Figure 4.14: Observations for one grid cell at 100°W to 80°West and 50°S to 30°S for three consecutive days (from top to bottom). The blue ellipses indicate the GMM components from the same cluster group for day one and day three; the red ellipses indicate the lack of these components at day two.

Figure 4.16 and Figure 4.17 illustrate the temporal trends over six consecutive days for the mean values of two different cluster groups each in features space ($\{H_2O, \delta D\}$) for the real-world dataset. While Figure 4.16 shows steady time-series of mean values for two different cluster groups, Figure 4.17 shows intersecting and possibly converging time series of mean values for two different cluster groups, which can give rise to further in-depth analysis of both cluster groups to explore possible interactions between them.

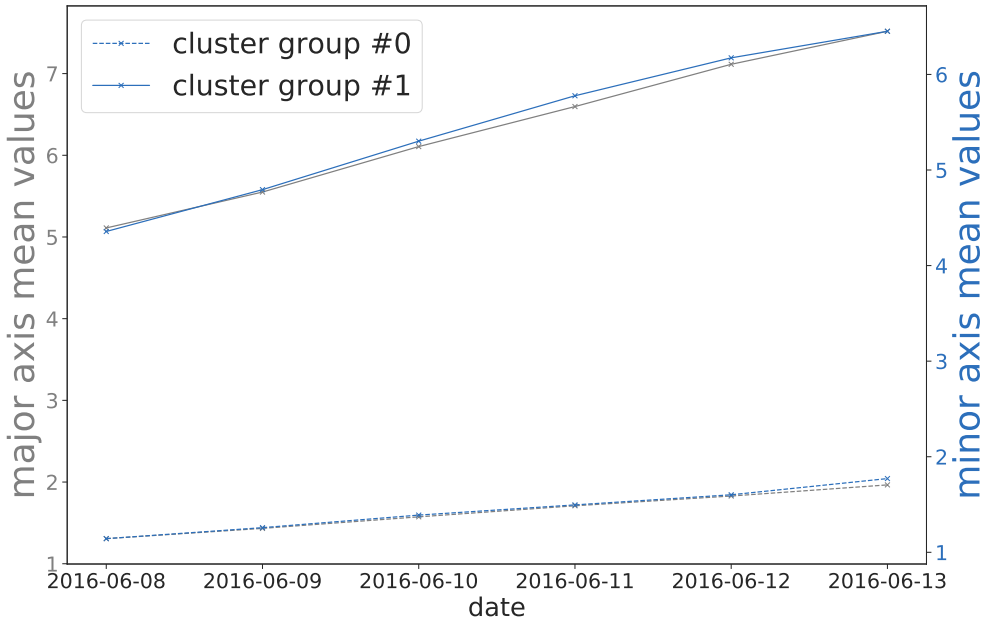


Figure 4.15: Illustration of the rate of expansion per day that has been put into place for the synthetic dataset; major ellipse axis mean value in grey (left y-axis) and minor ellipse axis mean value in blue (right y-axis) for two cluster groups (straight and dotted line).

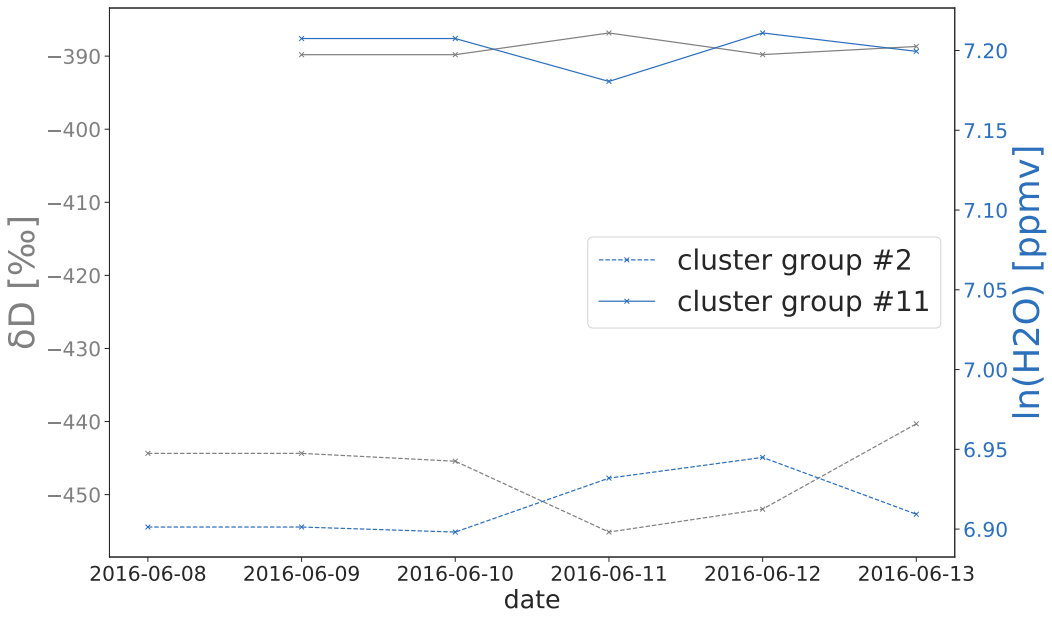


Figure 4.16: Steady time-series of mean values for two different cluster groups; δD mean value in grey (left y-axis) and $\ln(H_2O)$ mean value in blue (right y-axis) for two cluster groups (straight and dotted line).

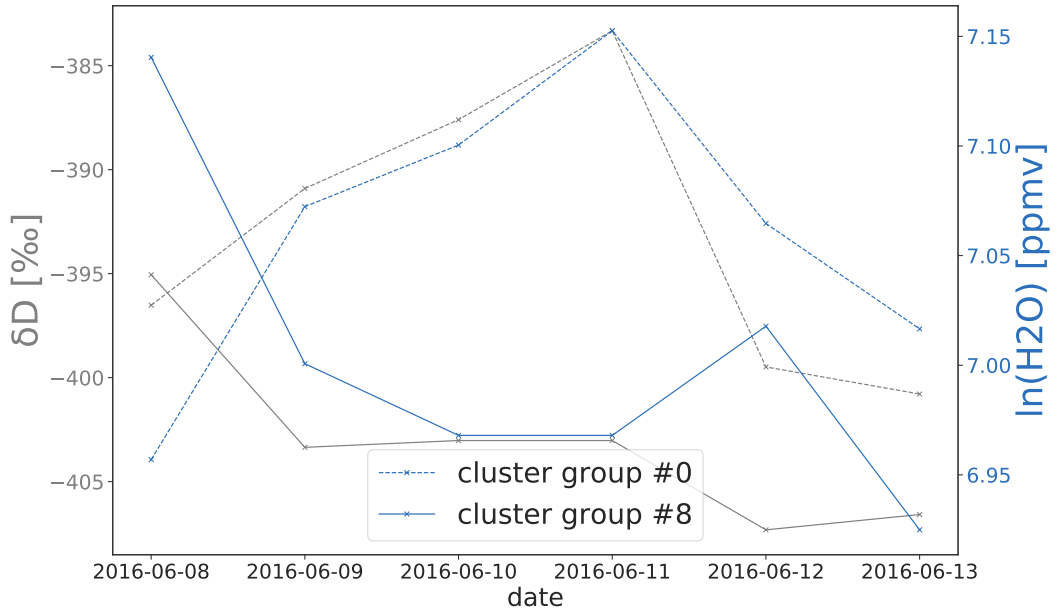


Figure 4.17: Intersecting time-series of mean values for two different cluster groups; δD mean value in grey (left y-axis) and $\ln(H_2O)$ mean value in blue (right y-axis) for two cluster groups (straight and dotted line).

4.4 Summary

This chapter presents a scalable approach for tracking clusters in spatio-temporal data. The presented method models the observed data in each spatial region with a mixture of Gaussians and compares extracted cluster properties over time. By dividing the initial dataset into spatial regions or applying sampling techniques such as random uniform sampling or biased sampling, each subset can be processed independently and in parallel, which significantly improves the overall runtime of the proposed model. As verified on synthetic data with known data generating processes and applied to real-world climatology data, the proposed model can reliably detect cluster changes over space and time, indicating moving clusters, appearing and disappearing events, and the evolution of clusters over time. Thus this approach enables the analysis of the evolution of geo-referenced distributions over time.

The initial step of the proposed model, dividing the data into spatial regions of interest, allows the parallel computation of Gaussian mixture models per area, which considerably increases the scalability of the approach. Even if dividing the data into spatial regions before applying the Gaussian mixture model might not always be applicable, for example, if splitting the data has a significant effect on the mixture components by including or excluding certain observations. In these cases, an analysis of

the point generating process can help to determine if the edge-effects can be statistically evaluated and mitigated (Diggle, 2014). To further improve the scalability of the model, each Gaussian mixture model could theoretically be run in parallel. Instead of sequentially evaluating the BIC score, the model with the best BIC score could be selected from parallel computations.

The results of the experimental evaluation in Section 4.3.2 show that the approach can detect changes in cluster properties over space and time successfully. The examples illustrated in Section 4.3.2 identify clusters moving between neighbouring spatial regions with similar cluster properties according to the applied DBSCAN algorithm on all extracted cluster properties. Although the proposed approach is not restricted to any method for comparing the Gaussian mixture components properties, DBSCAN shows good results and has the advantages that the number of clusters has not to be specified as a model parameter and the ϵ and $minPts$ parameters can be fine-tuned according to the cluster properties.

While identifying spatio-temporal changes successfully, one drawback of the proposed approach is that modelling the data with mixtures of Gaussians with the number of components depending on the model BIC score is not necessarily describing the underlying process that generated the data accurately. Therefore the interpretation of the results additionally relies on domain knowledge associating cluster and cluster changes to processes. The utilization of a more specific model that can also capture the underlying data generating processes more accurately is presented in the next chapter.

Chapter 5

CoExDBSCAN: Semi-Supervised Clustering for Spatio-Temporal Data

This chapter introduces a novel semi-supervised clustering algorithm with regard to spatio-temporal data; however, it is not limited to any particular set of data. The algorithm is based on the original DBSCAN clustering algorithm and can be described as a density-based clustering algorithm with **co**nstrained **ex**pansion, namely **CoEx**DBSCAN. The algorithm and the results presented in this chapter have been published in parts in Ertl et al. (2020).

5.1 Motivation

The experimental studies and evaluation of unsupervised clustering methods for spatio-temporal data detailed in the previous chapter, Chapter 4, show that in order to identify correlated structures in spatio-temporal datasets, there is the need for a clustering algorithm that can form partitions of data complying with a priori constraints in full value space or value subspaces. To overcome the issues of conventional full-space clustering methods, there has been a growing interest in subspace clustering and correlation clustering methods (Achtert et al., 2008). For this purpose, different subspace and correlation clustering algorithms have been proposed, which extend traditional clustering algorithms to detect correlations in subsets of features (Agrawal et al., 1998; Aggarwal and Yu, 2000). While some correlation algorithms are able to find arbitrarily oriented subspace clusters, for example, the Clustering in Arbitrary Subspaces based on the Hough transform (CASH) algorithm (Achtert et al., 2008), or can identify local subgroups of data objects sharing a uniform but arbitrarily complex correlation, for example the Computing Correlation Connected Clusters (4C) algorithm (Böhm et al., 2004), theses algorithms still face challenges, for instance, with

overlapping clusters or uncorrelated complex relations between features. For this reason, there has been a growing interest in semi-supervised clustering methods, for example, constrained clustering, where additional a priori information or domain knowledge is incorporated into the clustering process to better capture complex relations between features (Pourrajabi et al., 2014; Basu et al., 2008; Dinler and Tural, 2016). This chapter introduces a novel semi-supervised clustering algorithm combining different techniques from subspace, correlation and constraint clustering.

5.2 CoExDBSCAN Algorithm

The proposed algorithm, CoExDBSCAN, modifies the original DBSCAN algorithm (1) to allow a subspace to be used for the density-based definition of neighbourhoods and (2) to allow the restriction of the cluster expansion according to a priori constraints in the same or a different subspace of the dataset. For this purpose, the DBSCAN algorithm, that has been introduced in detail in Chapter 2, Section 2.4.3, needs to be adopted for both proposed extensions. First, the ϵ-neighbourhood definition of a point, see Definition 1, has to be reformulated.

Definition 12. *CoExDBSCAN ϵ-neighbourhood of a point*
Let D be a set (database) of points. The CoExDBSCAN ϵ-neighbourhood of a point p, denoted by $CoExN_\epsilon(p)$, is defined by

$$CoExN_\epsilon(p) = \{q \in D | dist(p_\varsigma, q_\varsigma) \leq \epsilon \wedge constraints(p_\rho, q_\rho)\}$$

where p_ς, q_ς are the subspace representations of point p and q of the user-defined spatial subspace ς, p_ρ, q_ρ are the subspace representations of point p and q of the user-defined constraint subspace ρ and the constraints function evaluates true for each constraint Γ_i in a user-defined set of constraints $\Gamma = \{\Gamma_1, \Gamma_2, ..., \Gamma_m\}$.

The pseudocode of the CoExDBSCAN algorithm is given in Algorithm 2; differences to the original DBSCAN algorithm, see Algorithm 1, in the pseudocode representation by Schubert et al. (2017) are coloured in red and marked by a star (*). In addition to the original input parameters ϵ and $minPts$, CoExDBSCAN requires the specification of the spatial subspace ($sDim$) and the constraint subspace ($cDim$), as well as the declaration of the user-defined constraints ($cFunc$).

Algorithm 2: Pseudocode of the CoExDBSCAN Algorithm

input : dataset D
input : radius ϵ
input : density threshold $minPts$
input : distance function $dist$
input : spatial dimensions $sDim$ *
input : user-defined constraints $cFunc$ *
input : constraint dimensions $cDim$ *
output : point labels $label$ initially $undefined$

```
1  foreach point p in dataset D do
2      if label(p) ≠ undefined then continue;
3      Neighbours N ← RangeQuery(D, dist, sdim, p, ε) *;
4      if |N| < minPts then
5          label(p) ← Noise;
6          continue;
7      c ← next cluster label;
8      label(p) ← c;
9      Seed set S ← N \ {p};
10     foreach q in S do
11         if label(q) = Noise then label(q) ← c;
12         if label(q) ≠ undefined then continue;
13         if PointConstraint (cFunc, cDim, q) is false then continue *;
14         Neighbours N ← RangeQuery(D, dist, sdim, q, ε) *;
15         label(q) ← c;
16         if |N| < minPts then continue;
17         foreach s in N do
18             S ← S ∪ s;
```

The `RangeQuery` function from the DBSCAN algorithm returns all points in the dataset that are within a certain distance to a given point, i.e. the ϵ-neighbourhood (Definition 1). For the CoExDBSCAN algorithm, the `RangeQuery` function has been modified to allow the specification of a subspace for the distance function `dist`, so that the distance function is only evaluated in the given subspace.

The *PointConstraint* function has been newly introduced for the CoExDBSCAN algorithm and evaluates for a given point the given set of user-defined constraints in a specified subspace. While there are no general restrictions for the user-defined constraints, to constrain the expansion of clusters, the constraints should be formulated based on the cluster points currently under consideration. This means that the algorithm should be enabled to decide if a new point should be included in the current set of points to form a cluster or not.

With these modifications, the algorithm works similar to the DBSCAN algorithm as following. CoExDBSCAN starts with an arbitrary point from the set of all points and iterates over all points (outer loop, Line 1). If the current point has already been labelled (Line 2), the algorithm moves on to the next point. If not, the CoExDBSCAN ϵ-neighbourhood for the given point is located (Line 3). This initial ϵ-neighbourhood corresponds to the DBSCAN ϵ-neighbourhood (Definition 1) with the difference that the distance function is only evaluated in the given subspace. The additional restriction on the user-defined constraints (Definition 12) is evaluated in the inner loop starting at Line 10. If the ϵ-neighbourhood contains less than $minPts$ points (Line 4), the point is labelled as noise (Line 5), and the algorithm moves on to the next point. If not, the point is considered a core point and starts to form a cluster, e.g. is labelled with the next available label (Line 7-8). Proceeding from the core point, all neighbouring points are added to a set of initially empty seeds (Line 9). For each point in the set (inner loop, Line 10), if the point has been previously labelled as noise, the point is labelled according to the current cluster label (Line 11). If the point has already been labelled with a cluster label (Line 12) the algorithm continues with the next point in the seed set (inner loop). If not, the user-defined constraints are evaluated for the given point in the seed set and the specified subspace (Line 13). If the evaluation fails, the algorithm continues with the next point in the seed set (inner loop). If the evaluation holds true, the CoExDBSCAN ϵ-neighbourhood for the given point is located (Line 14), and the point is labelled with the current cluster label (Line 15). If the ϵ-neighbourhood contains less than $minPts$ (Line 16), the algorithm continues with the next point in the seed set (inner loop). If not, each point from the ϵ-neighbourhood of the current seed point is added to the set of seed points (Line 17-18). The algorithm continues with the next point in the seed set (inner loop). If the seed points are empty, the algorithm continues with the next point in the dataset (outer loop).

CoExDBSCAN Example

To provide an intuitively accessible example, a simplified constraint formulation can be used that sets a threshold for a specific feature dimension. Figure 5.1 illustrates the restriction for the cluster expansion step of the original DBSCAN algorithm; Figure 5.1a and Figure 5.1b both depict the state of the respective algorithm after the first time a set of seed points has been located.

Starting from the blue-marked point, the original DBSCAN algorithm creates the first cluster with all neighbouring points within the ϵ-neighbourhood of the starting point, see red points in Figure 5.1a. Each point within the ϵ-neighbourhood is also considered a seed point, and the algorithm tries to expand the cluster proceeding from each seed point. A simplified constraint on the cluster expansion can be formulated as following:

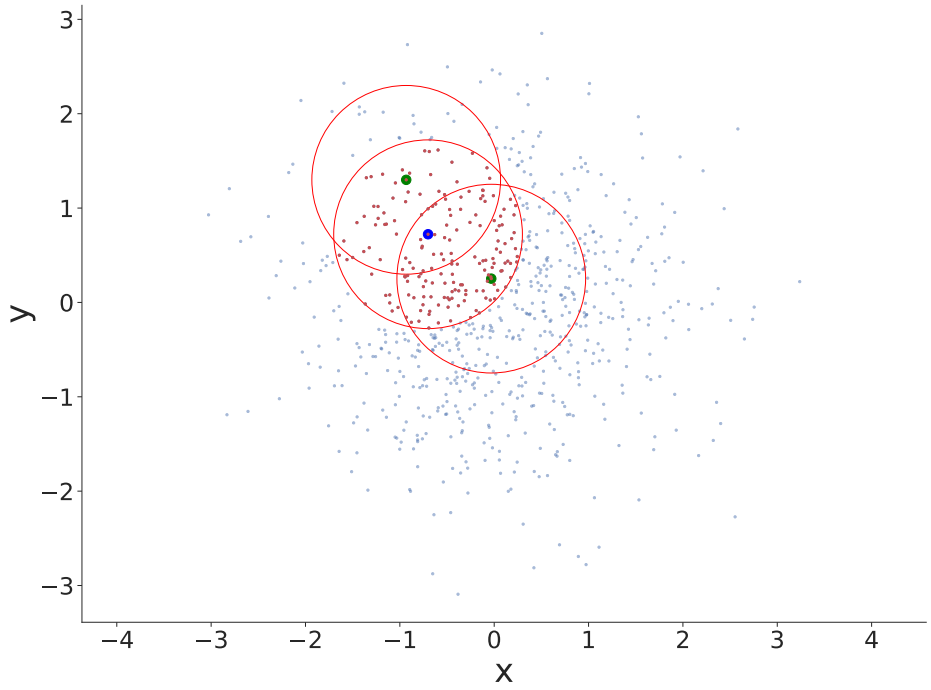
Definition 13. *A Point p_i belongs to a cluster $C = p_1, ..., p_n$, according to Definition 5, with n number of cluster points iff*

$$p_i(cDim) > \zeta \tag{5.1}$$

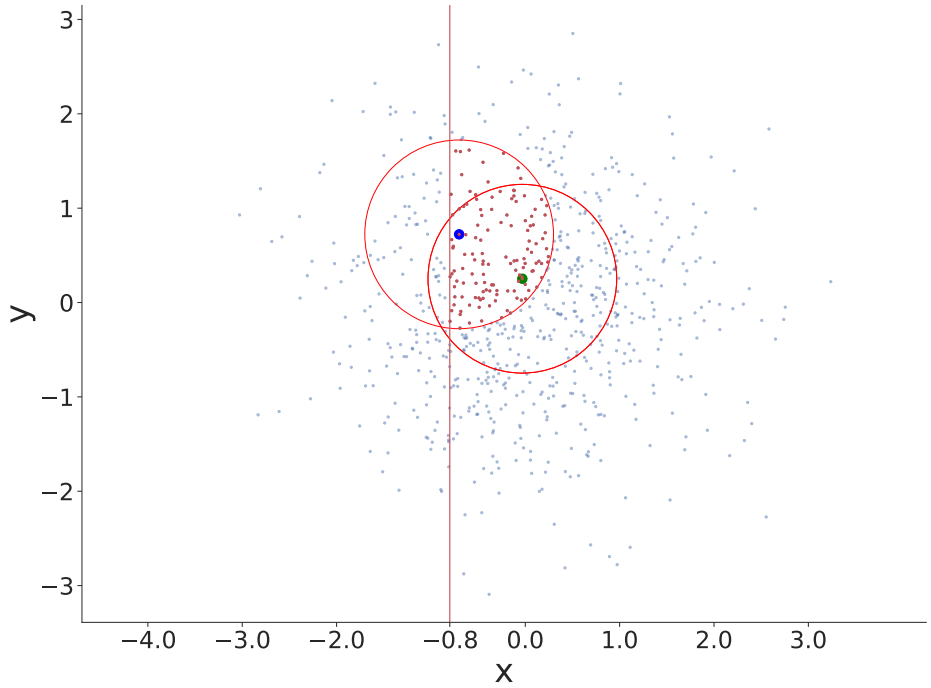
where $cDim$ is the constraint dimensions and ζ a specified threshold.

The threshold ζ in the illustrated example is arbitrarily set to -0.8 for demonstration purpose, and the x feature of the dataset is chosen as the constraint dimension $cDim$. The first notable difference for the CoExDBSCAN algorithm compared to the DBSCAN algorithm is the reduced number of cluster points (red points). Since the constraint is also enforced at the initial creation of the cluster, the initial cluster contains fewer points than compared to the initial cluster of the DBSCAN algorithm. With the initially reduced cluster points there are consequently less neighbouring points, i.e. less points in the set of seeds.

With the reduced set of seed points, the cluster expansion step can only operate on the reduced set, which only allows the cluster to grow according to the formulated constraint.



(a) State of the DBSCAN algorithm before expanding the first cluster.



(b) State of the CoExDBSCAN algorithm before expanding the first cluster.

Figure 5.1: Simplified example to demonstrate the constraint on the cluster expansion step of the CoExDBSCAN algorithm compared to the original DBSCAN algorithm. Initial cluster points are marked in red, with the initial core point (blue marking) and two random neighbours (green marking); red circles indicate the ϵ-neighbourhood of the marked points.

5.3 Evaluation

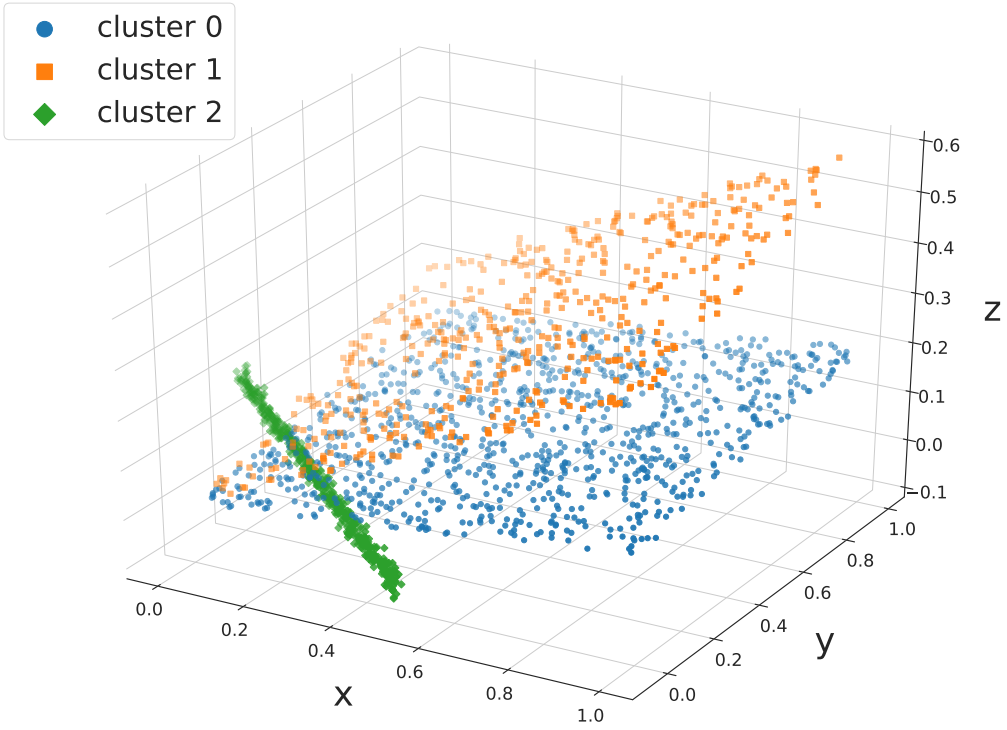
This section details the experimental studies to evaluate the proposed model on synthetic and real-world datasets. First, the setup for conducting the experimental studies is given in the following section, with the main objective to detect correlations, especially linear correlations, in subsets of features. Second, the results of comparing the CoExDBSCAN algorithm to the original DBSCAN algorithm as well as state-of-the-art algorithms for semi-supervised clustering and correlation clustering are presented together with the runtime analysis.

5.3.1 Setup

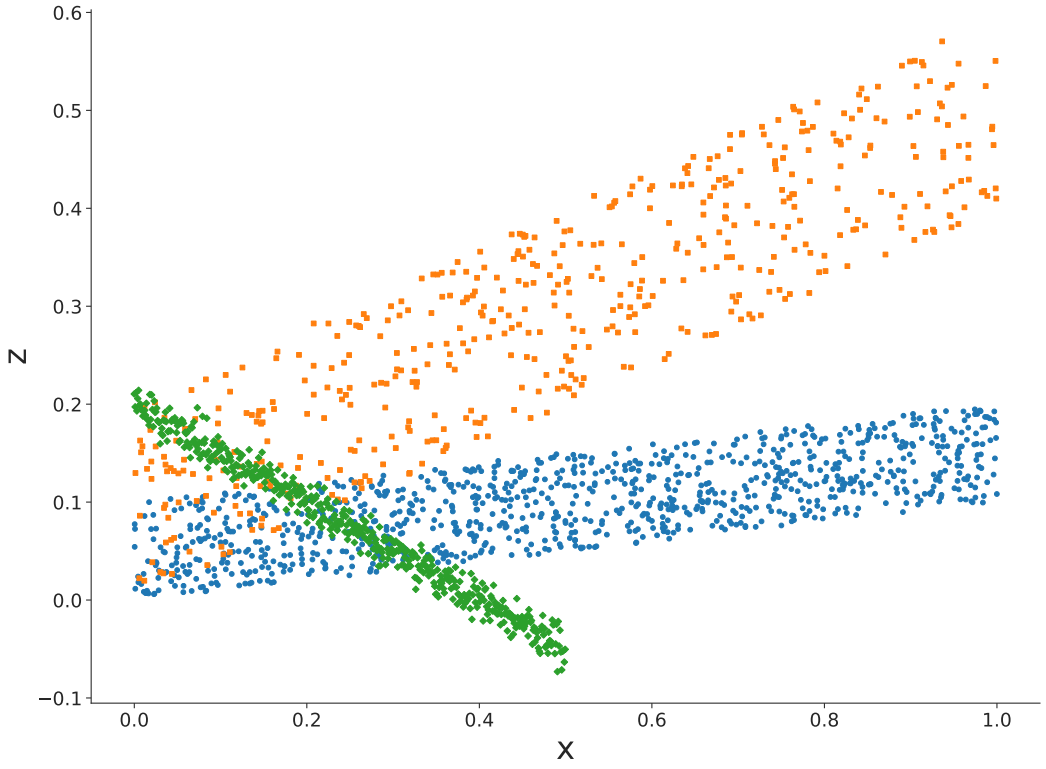
Synthetic Data

The correctness of the CoExDBSCAN algorithm is verified on a generated synthetic datasets. The dataset has been generated since the generation of a particular dataset allows full control over the properties of the clusters. This dataset has been generated to be especially challenging for density-based clustering methods as well as subspace and correlation clustering methods but remains intuitively accessible and has a clear visualization.

The synthetic dataset contains three true clusters with overlapping areas of all three clusters and in total 2,000 points with a different number of points per cluster to generate inhomogeneous clusters. Table 5.1 lists the generation methods and interval ranges of the variables together with their linear dependencies. Figure 5.2 illustrates the synthetic dataset in three-dimensional space (Figure 5.2a) and two-dimensional projection (Figure 5.2b). The values for the x and y variables of the cluster with label zero, blue colour in Figure 5.2, are generated by sampling the random uniform distribution in the half-open interval $[0, 1)$; the values for the z variable are computed using the linear equation $0.1x + 0.1y$. For the cluster with label one, orange colour in Figure 5.2, the values for the x and y variables are generated by sampling the random uniform distribution in the half-open interval $[0, 1)$; the values for the z variable are computed using the linear equation $0.4x + 0.2y$. The green coloured cluster with label two in Figure 5.2 is generated by evenly spaced x values in the closed interval $[0, 1]$ and the values for the y and z variables following the linear equation $-0.5x + 0.2 + \xi$, where ξ is some random variation with $\xi \sim \mathcal{N}(0, 0.01)$.



(a) Synthetic data with true labels.



(b) Synthetic data {x,z} projection.

Figure 5.2: Synthetic dataset for the evaluation of the CoExDBSCAN algorithm.

Table 5.1: Value range and dependencies for the first synthetic dataset.

Cluster	Points	x	y	z
0	1,000	uniform [0,1)	uniform [0,1)	$0.1x + 0.1y$
1	500	uniform [0,1)	uniform [0,1)	$0.4x + 0.2y$
2	500	evenly [0,1]	$-0.5x + 0.2 + \xi$	$-0.5x + 0.2 + \xi$

All parameters for generating the synthetic data, e.g. the number of clusters, points per cluster and linear equation parameters, have been chosen from empirical knowledge to provide a challenging dataset that remains intuitively accessible and has a clear visualisation but poses a challenge to the evaluated clustering algorithms because 1) the clusters have different densities, with a very dense, line-shaped cluster and two looser, plane-shaped clusters and 2) all clusters are overlapping with some proportions. The plane-shaped clusters have an overlapping edge, which is crossed by parts of the line-shaped cluster. In addition, the dataset has an inhomogeneous distribution of points for the three clusters.

Real-World Data

The real-world dataset remains the same as in the previous chapter, see Chapter 4 (Section 4.3.1), and contains time snapshots from the MUSICA IASI satellite-based remote sensing dataset as described in Section 2.2. Different to the dataset from Chapter 4, the evaluation of the CoExDBSCAN algorithm focuses on a particular area of interest which is subject of further analysis in the next chapter as well.

Figure 5.3 illustrates the first time snapshot at 2016-06-08 with the area of interest, 7.5 degrees North to 15 degrees North and 8 degrees East to 8 degrees West, highlighted by the red rectangle. Domain experts have proposed this particular area of interest to be aligned with multiple moisture transport processes studies. In the following, the data from the period 2016-06-08 to 2016-06-14 of the area of interest are analysed.

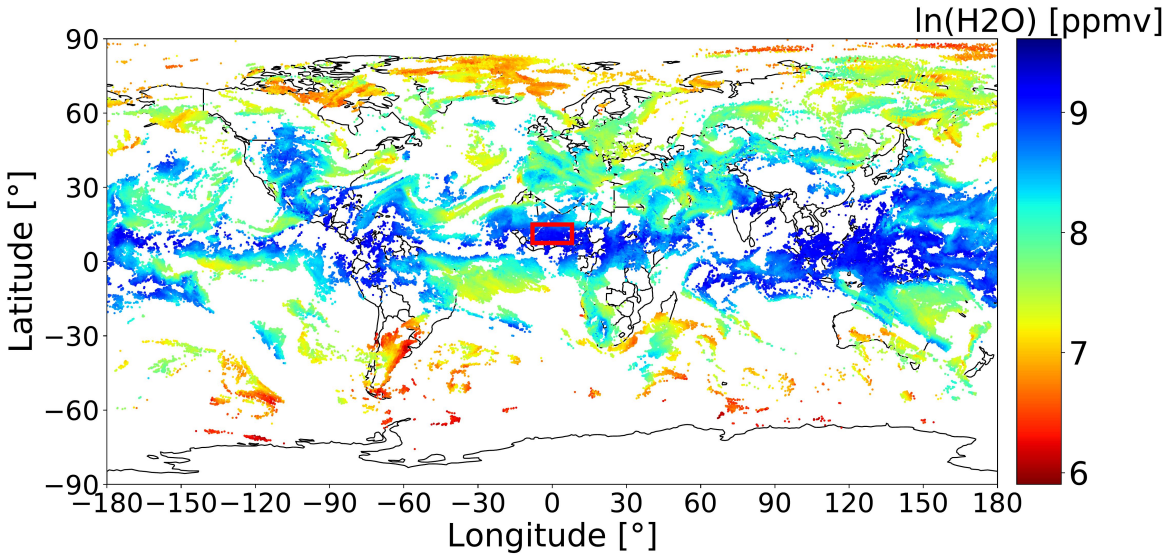


Figure 5.3: MUSICA IASI H_2O data for morning satellite overpasses at 2016-06-08 with the area of interest highlighted by the red rectangle. H_2O values are in parts per million by volume (ppmv) in logarithmic scale. The depicted data are limited to cloud-free observations and have been filtered for best quality (retrievals with good sensitivity and low errors).

Application

To evaluate linear correlations in the synthetic dataset and real-world dataset, the following constraint can be formulated to restrict the expansion of clusters to a defined margin around the linear regression of existing cluster points.

Definition 14. *A Point p_i belongs to a cluster $C = p_1, ..., p_n$, according to Definition 5, with n number of cluster points iff*

$$(Y_{C \cup p_i} - \hat{Y}_{C \cup p_i})^2 < \delta \cdot \frac{1}{n} \sum_{C \setminus p_i} (Y_{C \setminus p_i} - \hat{Y}_{C \setminus p_i})^2 \tag{5.2}$$

where Y and $\hat{Y}$ are the dependent variable and fitted value of the linear regression respectively, fitted to the user-defined subspace of features.

In other words, a point p_i is added to an existing cluster C only if the squared residuals of the linear regression of all cluster points including p_i is smaller than the mean squared residuals of all cluster points without p_i times a certain threshold δ. The linear regression is fitted to the user-defined constraint subspace, see Definition 12, and the threshold parameter δ provided by the user as well. Experiments have shown that this constraint is a suitable constraint that expresses the a priori knowledge of

correlated structures in the dataset generic and allows the algorithm to expand clusters on arbitrarily correlated structures and changing correlations up to a certain degree.

The CoExDBSCAN algorithm outlined in Algorithm 2 has been implemented in Python utilizing the scikit-learn (Pedregosa et al., 2011) machine learning package implemented in Python as well. For the constraint formulated in Definition 14 the linear regression is estimated by the ordinary least squares implementation by the statsmodels Python module (Seabold and Perktold, 2010).

5.3.2 Results

Runtime Analysis

The main computational cost of the original DBSCAN algorithm is due to the computation of distance queries for each point in the dataset. Ester et al. (1996) argue that such distance queries can be supported efficiently by spatial access methods such as R^*-trees (Beckmann et al., 1990). R^*-trees have a height of $O(\log n)$ for a dataset of n points in the worst case, and a query with a small query region has to traverse only a limited number of paths (Ester et al., 1996). Ester et al. further argue that since the ϵ-neighbourhood queries are expected to be in a small region compared to the size of the whole data space, the average runtime complexity of a single region query is $O(\log n)$. DBSCAN has to execute one query for each of n points in the dataset, which results in an overall average runtime complexity of $O(n \log n)$.

Introducing a set of constraints $\Gamma = \{\Gamma_1, \Gamma_2, ..., \Gamma_m\}$ (Definition 12) to the expansion step of the DBSCAN algorithm adds the complexity of $O(n \cdot max(\Gamma))$ to check the set of constraints for each point. The complexity of constraints can vary greatly, for example from hash table searches with average time complexity $O(1)$ (Cormen, 2009) to linear regression complexity $O(w^2 n + w^3)$ for n number of observations and w number of weights (Mohri et al., 2012). The runtime complexity of CoExDBSCAN depends on the user-defined constraints and is on average the runtime complexity of DBSCAN plus the maximum complexity of the user-defined constraints, $O(n \cdot max(\Gamma) + n \cdot \log n)$.

Synthetic Data

Since the true labels of the synthetic dataset are known, the CoExDBSCAN algorithm can be evaluated according to the Adjusted Rand Index (ARI) and Cluster Accuracy (ACC) metrics as detailed in Chapter 2, Section 2.4.4.

The evaluation on the synthetic dataset compares the CoExDBSCAN algorithm to the original DBSCAN algorithm as a baseline approach, the CK-means algorithm (Pelleg and Baras, 2007) and the PCK-means algorithm (Basu et al., 2004) as a comparison to existing semi-supervised clustering algorithms and the CASH algorithm by Achtert et al. (2008) as a state-of-the-art comparison.

The constrained k-means algorithm (CK-means) and the pairwise constrained k-means algorithm (PCK-means) are semi-supervised variants of the K-means algorithm, see Chapter 3, Section 3.2.1 and Section 3.2.2. The implementations of both algorithms are provided by the active-semi-supervised-clustering Python package.

The Clustering in Arbitrary Subspaces based on the Hough transform (CASH) algorithm has been chosen, because according to the authors Achtert et al. (2008) CASH significantly outperforms other correlation clustering algorithms, such as the Arbitrarily ORiented projected CLUSter generation (ORCLUS) algorithm or the Computing Correlation Connected Clusters (4C) algorithm, on datasets with highly overlapping clusters in terms of robustness and effectiveness. The implementation is provided by the Environment for deveLoping KDD-applications supported by Index-structures (ELKI) data mining software (Schubert and Zimek, 2019) written in Java.

The CoExDBSCAN and DBSCAN algorithm are evaluated for an ϵ range of $[0.01, 0.20]$ with a step size of 0.01 and a $minPts$ range of $[10, 70]$ with a step size of 10; range and step size have been chosen to cover the main variability of the number of clusters and the cluster accuracy. The δ threshold parameter for the constraint formulated in Definition 14 is evaluated for three values $[3, 4, 5]$ that yield the best accuracy score on a given set of parameters. As spatial subspace and constraint subspace for the CoExDBSCAN algorithm, see Definition 12, only the x and z features have been selected, see Figure 5.2, based on empirical analysis. For the DBSCAN, CK-means, PCK-means and CASH algorithm the full value space has been provided resulting in higher accuracy scores in general.

The results after applying the proposed CoExDBSCAN algorithm to the synthetic dataset range from zero to 26 clusters. The ARI score is between 0 and 0.60 and the Cluster Accuracy (ACC) ranges from 0.32 to 0.82, depending on the ϵ and $minPts$ parameters of the original DBSCAN algorithm and the δ threshold parameter. Figure 5.4 illustrates the parameter dependencies to the Cluster Accuracy. The parameter set with the highest accuracy ($\sim 82\%$) corresponds to $\epsilon = 0.03$, $minPts = 20$ and $\delta = 5$, indicated by the first red dashed line in Figure 5.4. The plot of the parameters' dependencies and the ARI is given in Appendix A.1.

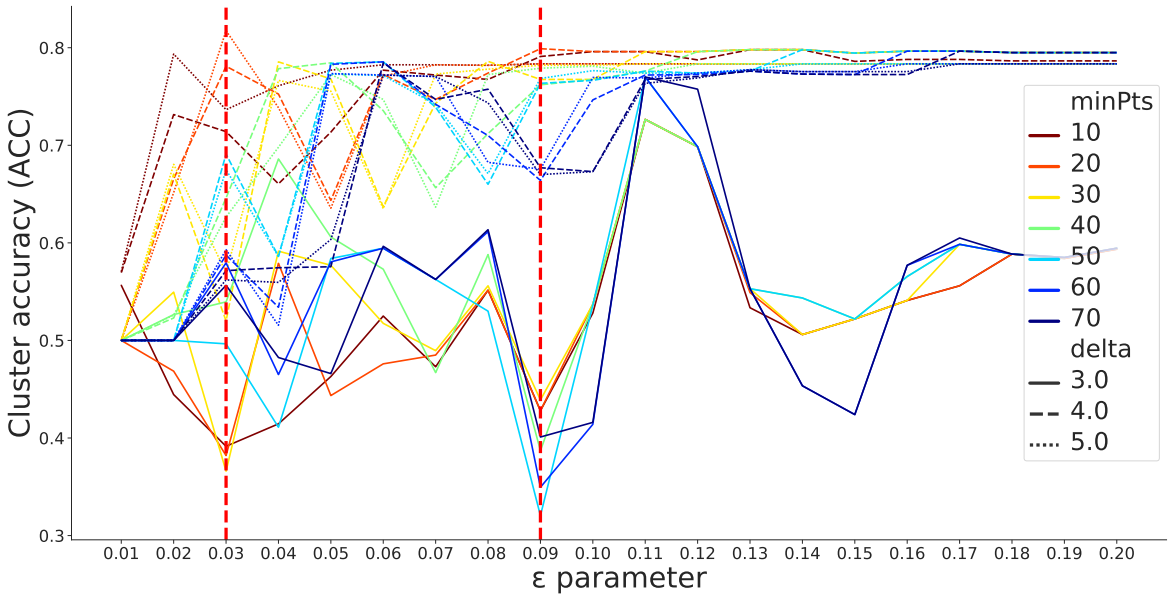


Figure 5.4: Dependencies between ϵ, $minPts$ and δ parameters and the Cluster Accuracy (ACC). The two red dashed lines indicate the set of parameters with the highest accuracies, 0.82 and 0.80.

However, the parameter set with the second highest accuracy ($\sim 80\%$), with $\epsilon = 0.09$, $minPts = 20$ and $\delta = 4$ indicated by the second red dashed line in Figure 5.4, yields a better result in terms of the number of noise points, see Table 5.2, which can be validated visually as well, see Figure 5.5 and Figure 5.6.

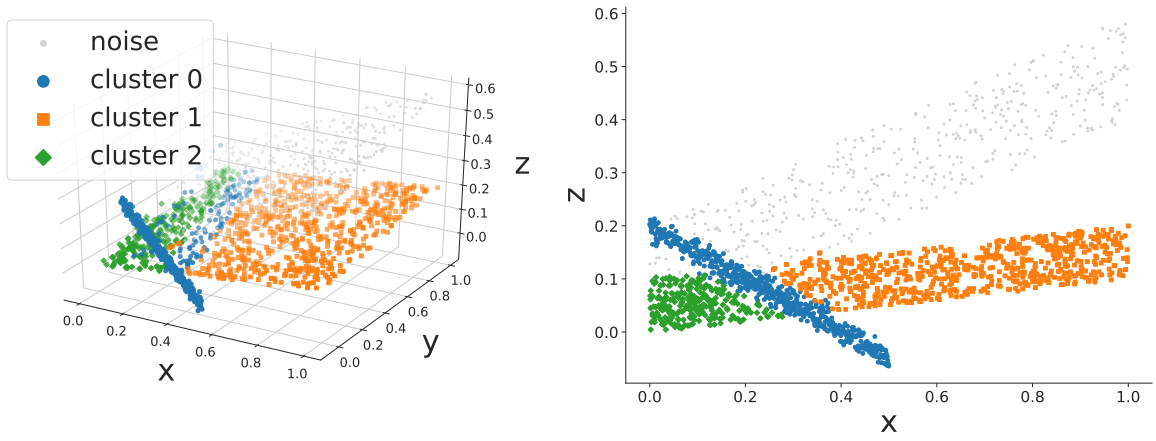


Figure 5.5: CoExDBSCAN clustering result with the highest accuracy ($\sim 82\%$), $\epsilon = 0.09$, $minPts = 20$ and $\delta = 4$, indicated by the second red dashed line in Figure 5.4.

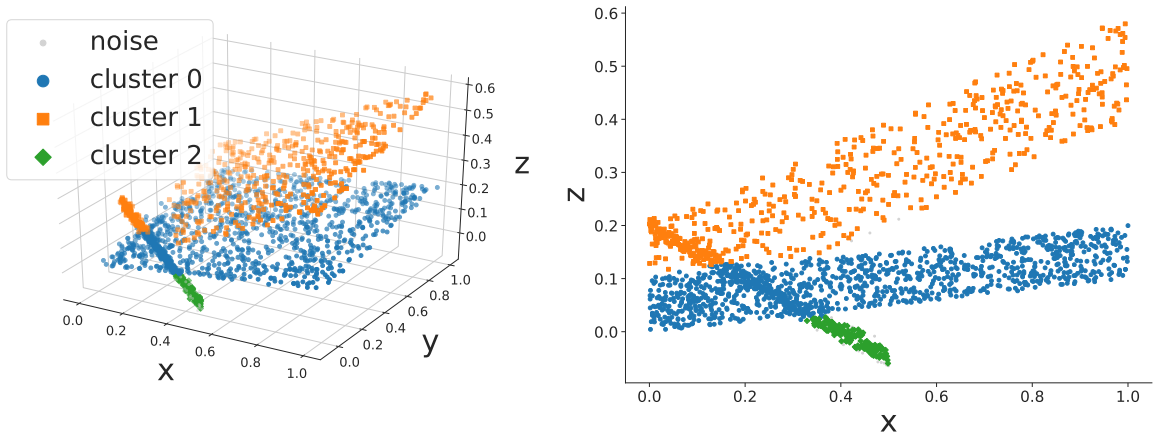


Figure 5.6: CoExDBSCAN clustering result with the second highest accuracy ($\sim 80\%$), $\epsilon = 0.03$, $minPts = 20$ and $\delta = 5$, indicated by the first red dashed line in Figure 5.4.

Applying the DBSCAN algorithm to the synthetic dataset for the same parameter range results in clusterings that range from zero to 24 clusters. The ARI score is between 0 and 0.52 and the ACC ranges from 0.50 to 0.74, depending on the ϵ and $minPts$ parameters. The plots of the parameters' dependencies and both metrics are given in Appendix A.1. The clustering result with the highest accuracy produces only one cluster; see Figure 5.7.

The clustering result with the highest accuracy ($\sim 70\%$) and the correct number of clusters does not correctly represent the true structure of the clusters as well, as illustrated in Figure 5.8. An example for a clustering result that better captures the true structure of the data is given in Figure 5.9. More details on the results are given in Table 5.2.

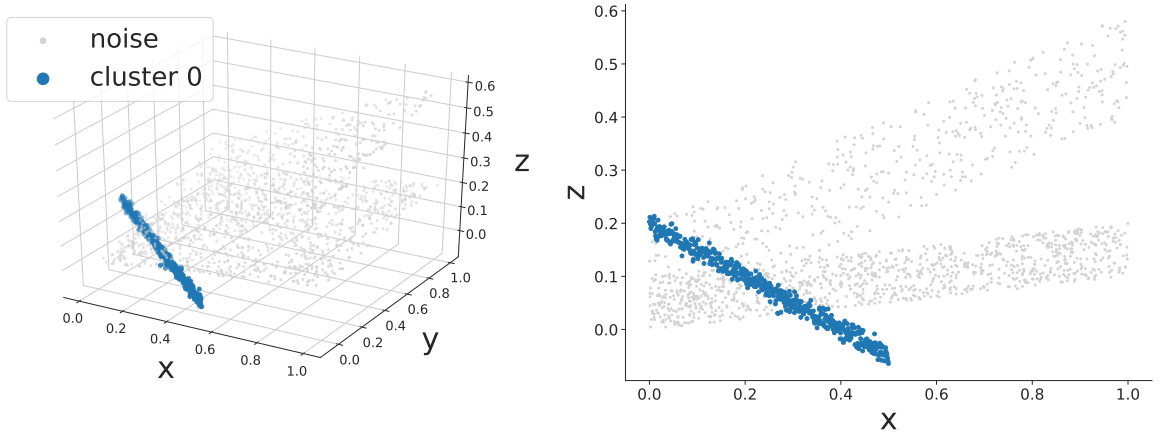


Figure 5.7: DBSCAN clustering result with the highest accuracy ($\sim 74\%$), $\epsilon = 0.03$ and $minPts = 30$.

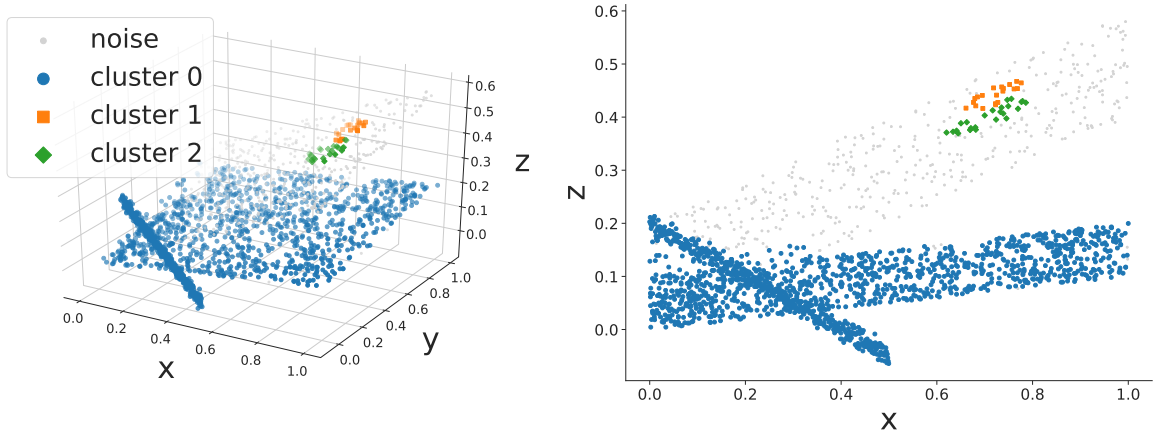


Figure 5.8: DBSCAN clustering result with the highest accuracy ($\sim 70\%$) and correct number of clusters, $\epsilon = 0.09$ and $minPts = 20$.

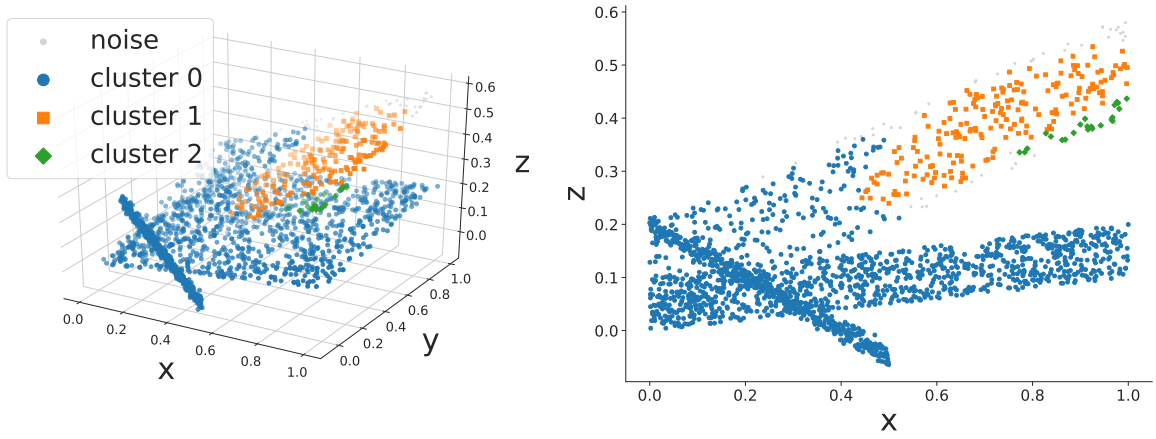


Figure 5.9: Example DBSCAN clustering result with a better visually verifiable representation of the true clusters; cluster accuracy $\sim 60\%$, $\epsilon = 0.14$ and $minPts = 30$.

While the CK-means algorithm can be applied to the synthetic data with partially labelled data, the true labels have to be transformed into pairwise constraints for the PCK-means algorithm, see Chapter 3 (Section 3.2.2). For the must-link constraints, all pairwise combinations for each point within each cluster are generated, and for the cannot-link constraints, all pairwise combinations between cluster points have to be generated.

Figure 5.10 shows the increase of accuracy and the ARI score for the CK-means algorithm depending on the fraction of true labels provided as partially labelled a priori information to the algorithm. The fraction of labels has been randomly sampled from

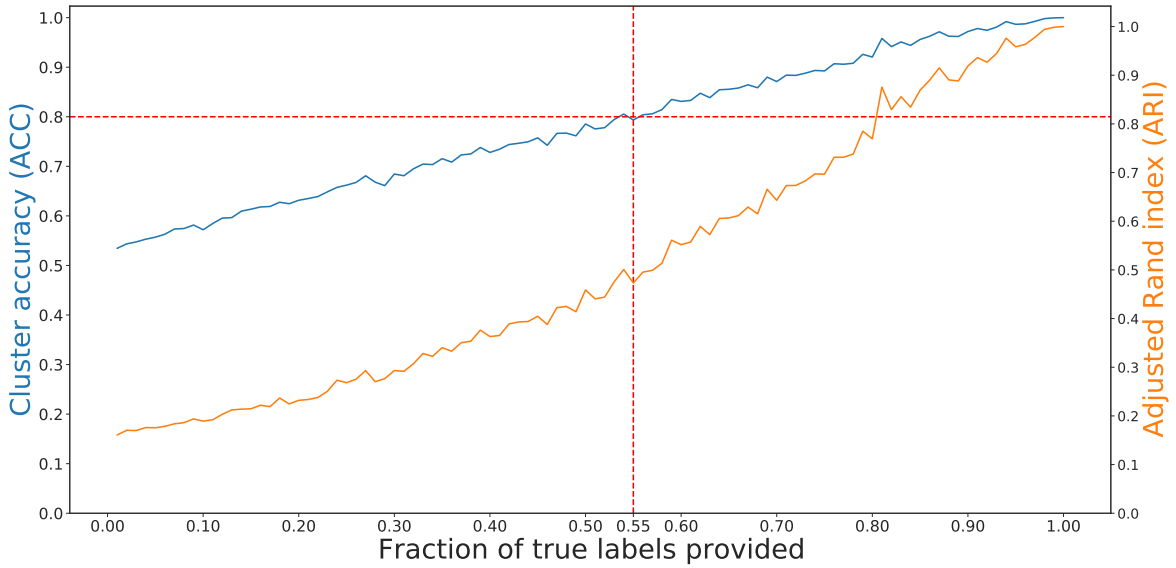


Figure 5.10: Cluster Accuracy and Adjusted Rand Index for the constraint k-means clustering algorithm depending on the fraction of true labels provided; red dashed lines indicate the same accuracy ($\sim 80\%$) as for the CoExDBSCAN algorithm by $\sim 55\%$ of true labels provided.

all true labels without replacement. It can be observed that providing around 55% of the true labels results in the same accuracy of the CK-means clustering algorithm as the accuracy of the CoExDBSCAN algorithm.

Figure 5.11 visualizes an example clustering result of the synthetic data applying the CK-means algorithm with 55% of the true labels randomly sampled provided to the algorithm. With around half of the true labels provided in addition to the true number of clusters $k = 3$, the clusters identified by the CK-means algorithm show overlapping areas with a different inherent correlation structures. This can be observed for example for cluster 0 (blue colour) and cluster 1 (orange colour) in the $x - z$ projection for x values greater 0.6. The parameter values, Cluster Accuracy (ACC) and Adjusted Rand Index (ARI) are summarised in Table 5.2.

Figure 5.12 shows the increase of accuracy and the ARI score for the PCK-means algorithm depending on the number of pairwise constraints provided. The pairwise constraints have been sampled from the transformed true labels with the same proportion of must-link and cannot-link constraints.

It can be observed in Figure 5.12 that providing around $2,200$ constraints, $1,100$ must-link and $1,100$ cannot-link constraints respectively, results in the same accuracy of the PCK-means clustering algorithm as the accuracy of the CoExDBSCAN algorithm.

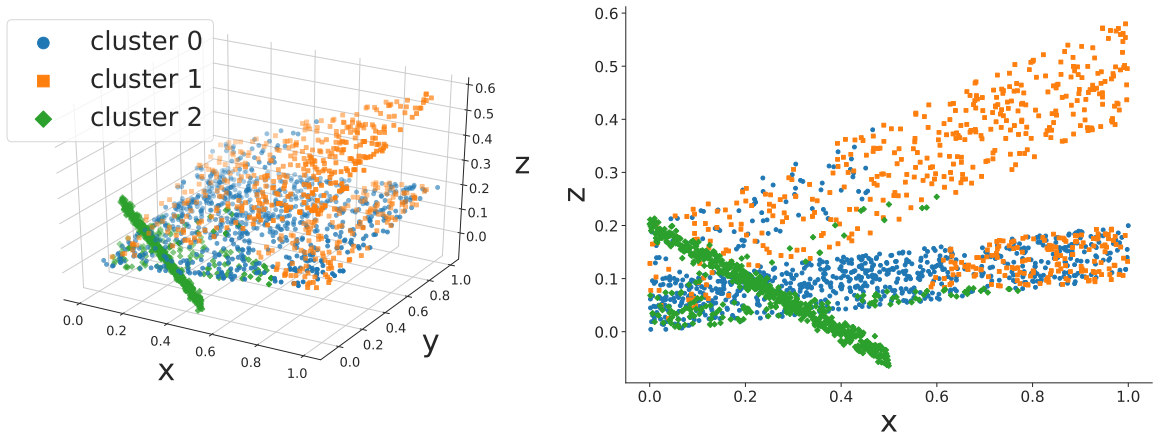


Figure 5.11: Example constraint k-means clustering result; cluster accuracy $\sim 80\%$, number of true clusters given $k = 3$ and 55% of true labels provided.

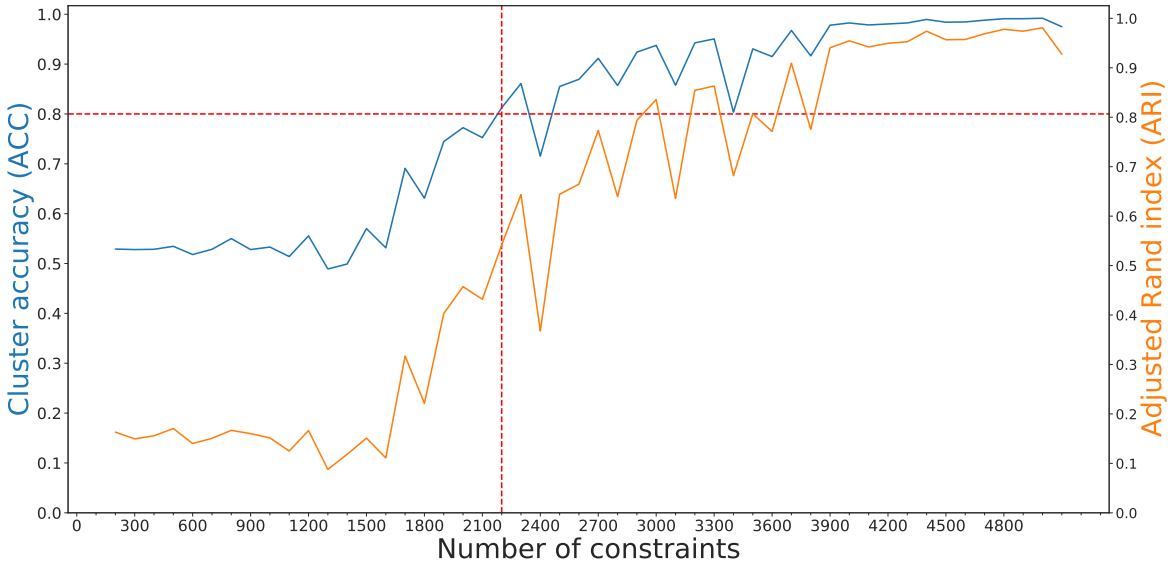


Figure 5.12: Cluster Accuracy and Adjusted Rand Index for the pairwise constraint k-means clustering algorithm depending on the number of constraints provided; red dashed lines indicate the same accuracy ($\sim 80\%$) as for the CoExDBSCAN algorithm by $2,200$ constraints provided.

Figure 5.13 visualizes an example clustering result of the synthetic data applying the PCK-means algorithm with $2,200$ constraints based on the true labels randomly sampled provided to the algorithm. With $2,200$ constraints, $1,100$ must-link and $1,100$ cannot-link constraints respectively, provided in addition to the true number of clusters $k = 3$, the PCK-means algorithm shows a qualitative better result as compared to the CK-means clustering algorithm, where the overlap of identified clusters is visibly smaller.

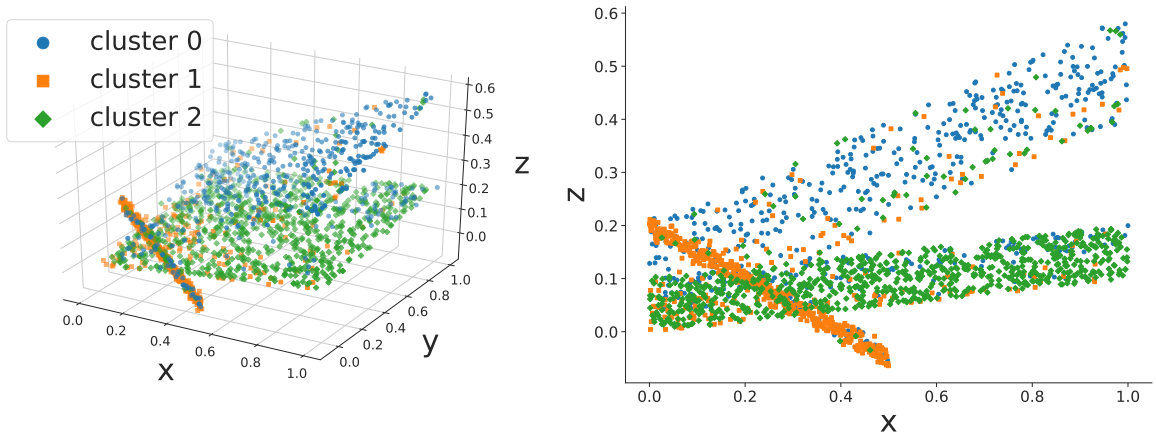


Figure 5.13: Example pairwise constraint k-means clustering result; cluster accuracy $\sim 78\%$, number of true clusters given $k = 3$ and $2,200$ constraints based on the true labels provided.

The parameters for the CASH algorithm as a state-of-the-art comparison that need to be defined are first the minimum number of sinusoidal curves that need to intersect a hypercuboid in the parameter space such that this hypercuboid is regarded as a dense area (Achtert et al., 2008). The transformation from data space to parameter space is done following an approach based on the Hough transformation (Duda and Hart, 1972). According to Achtert et al. (2008) this parameter represents the minimum number of points in a cluster. Second, the maximum number of splits along a search path in parameter space has to be given that controls the maximally allowed deviation from the hyperplane of the cluster in terms of orientation and jitter. Third, the jitter parameter has to be defined that allows a certain degree of deviation from exact intersections of sinusoidal curves due to the discretization of the parameter space into grid cells.

Figure 5.14 illustrates the dependencies of the three CASH parameters to the cluster accuracy (y-axis) for the synthetic dataset, the minimum number of sinusoidal curves/minimum number of points ($minPts$), the maximum number of splits ($maxSplit$) indicated by colours and the $jitter$ indicated by the line style. The parameter space has been evaluated for the minimum number of curves in the interval $[1, 1000]$ with a step size of one, the maximum number of splits for 5, 10 and 20 and the jitter values 0.1, 0.15 and 0.2. The range for each parameter has been chosen to cover the main variability in cluster accuracy.

The parameter space exploration shows that CASH reaches its highest cluster accuracy ($\sim 77\%$) for $minPts = 460$, $maxSplit = 20$ and $jitter = 0.2$, see Figure 5.14. Figure 5.15 visualizes an example clustering result of the synthetic data applying the CASH algorithm $minPts = 460$, $maxSplit = 20$ and $jitter = 0.2$.

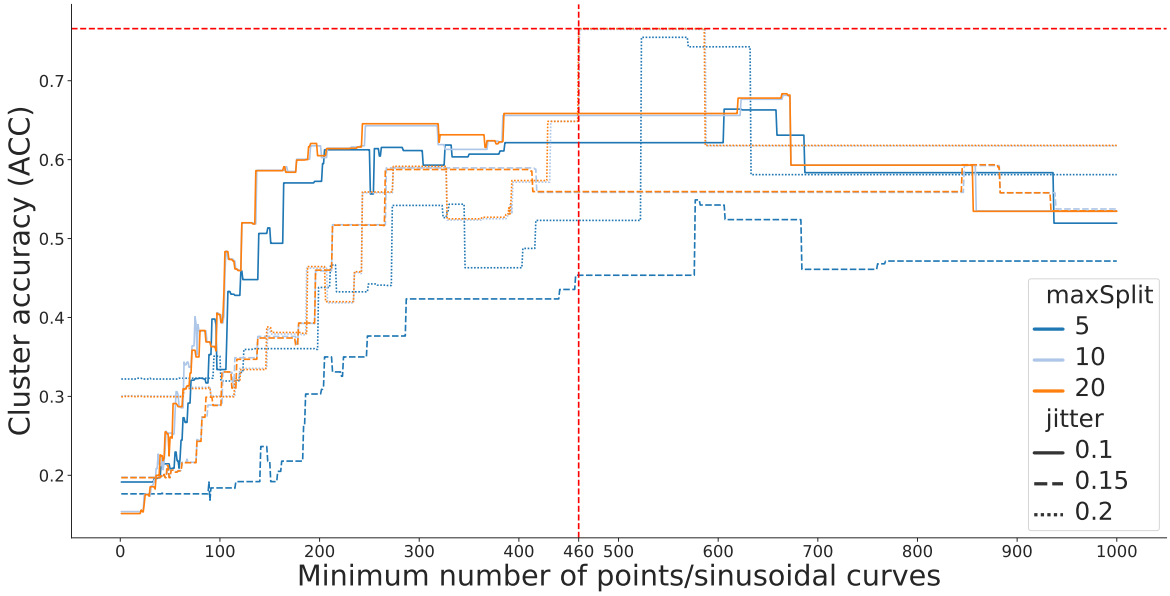


Figure 5.14: Dependencies between *minPts*, *maxSplit* and *jitter* parameters and the Cluster Accuracy (ACC). The two red dashed lines indicate the set of parameters with the highest cluster accuracy, $minPts = 460$, $maxSplit = 20$ and $jitter = 0.2$.

The number of true clusters has been correctly identified, further the algorithm demonstrates its capability to find subspace clusters with correlated structures even if they are intersected by other clusters. However, the CASH algorithm can not entirely separate overlapping areas with a different inherent correlation structure similar to the results with the CK-means algorithm, for example, cluster 0 (blue colour) and cluster 1 (orange colour) that are noticeable in the 3D view and the x-z projection for x values smaller 0.4.

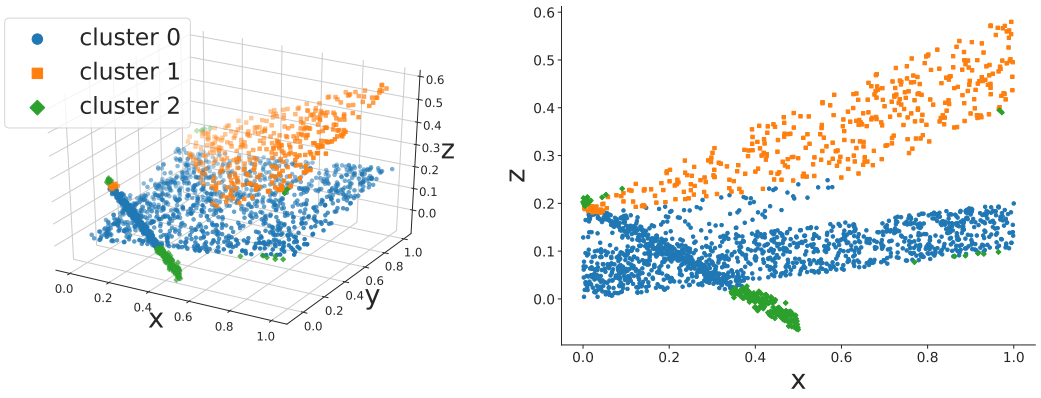


Figure 5.15: Example CASH clustering result; cluster accuracy $\sim 77\%$, $minPts = 460$, $maxSplit = 20$ and $jitter = 0.2$.

Table 5.2 summarizes the clustering results for the synthetic data applying the CoExDBSCAN algorithm, the DBSCAN algorithm the CK-means and PCK-means algorithms and the CASH algorithm. Without the prior knowledge of the true number of clusters or partially labelled data, CoExDBSCAN is outperforming the original DBSCAN algorithm as a baseline approach and the state-of-the-art CASH algorithm in terms of Cluster Accuracy (ACC), Adjusted Rand Index (ARI) and the ability to separate clusters with different inherent correlated structures. To reach a similar accuracy as the CoExDBSCAN algorithm, the semi-supervised algorithms CK-means and PCK-means require around half of the true labels known a priori or around 2,000 pairwise constraints in addition to the true number of clusters. Although given this amount of knowledge to the algorithms in advance, the clustering results can not clearly separate the clusters with different inherent correlations.

	ϵ	$minPts$	δ	noise	cluster	ARI	ACC
CoExDBSCAN	0.03	20	5	435	3	0.60	0.82
	0.09	20	4	**20**	**3**	0.59	**0.80**
DBSCAN	0.03	20	-	405	1	0.37	0.70
	0.02	10	-	428	3	0.37	0.70
	0.08	50	-	21	3	0.20	0.61
	k	known labels			cluster	ARI	ACC
CK-means	3	55%			3	0.49	0.80
PCK-means	3	2,200			3	0.46	0.78
	maxSplit	minPts	jitter		cluster	ARI	ACC
CASH	20	460	0.2		3	0.44	0.77

Table 5.2: Summary of clustering results for the synthetic data using the adjusted Rand index (ARI) and Cluster Accuracy (ACC) metrics.

Real-World Data

To evaluate the proposed CoExDBSCAN algorithm on the real-world MUSICA IASI dataset in the stipulated area of interest without a priori known labels, there can be a comparison made between the DBSCAN algorithm, the CASH algorithm and the CoExDBSCAN algorithm in terms of their ability to identify correlated structures that are geographically close and that are expected to follow theoretical assumptions, see Chapter 2 (Section 2.2).

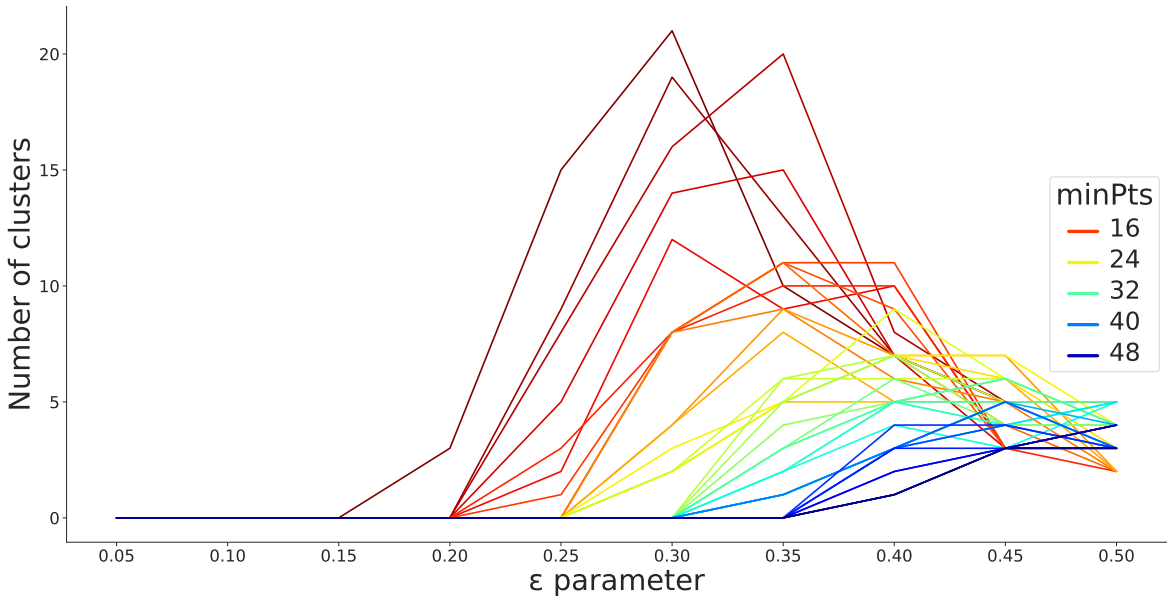


Figure 5.16: Dependencies between ϵ and $minPts$ parameters and the number of clusters identified by DBSCAN.

Applying the DBSCAN algorithm to an arbitrarily chosen time snapshot of the data, i.e. one day, in the area of interest in the parameter space $\epsilon \in [0.05, 0.5]$ with a step size of 0.05 and $minPts \in [10, 50]$ with a step size of 1 results in zero up to 21 clusters, with the number of clusters peak at ϵ around 0.3 and $minPts$ around 10, as illustrated in Figure 5.16. To compare the selected algorithms on the same level of granularity for the number of clusters, the parameter set that produces 20 clusters has been chosen, i.e. $\epsilon = 0.35$ and $minPts = 12$. The feature space for the clustering algorithm consists of the latitude and longitude values as well as the natural logarithm of the H_2O values and the natural logarithm of the δD values divided by 1,000 plus one, $\ln(\frac{\delta D}{1000} + 1)$, assuming a linear correlation between $\ln(H_2O)$ and $\ln(\frac{\delta D}{1000} + 1)$. This assumption is expected to be verifiable by correlated structures in the $\{H_2O, \delta D\}$ value space. All features have been standardized by removing the mean and scaling to unit variance to avoid individual features dominating the distance calculations.

Figure 5.17 shows the clustering result for the DBSCAN algorithm on the defined and standardized feature space of the real-world data snapshot with the selected parameter set in the geo-referenced latitude and longitude space. The result shows mostly spatial discrete and concise clusters, except the first large cluster; points that do not belong to any cluster (noise) are not shown. The same pattern can be observed in the $\{H_2O, \delta D\}$ feature space, see Figure 5.18, which shows primarily discrete and concise clusters, except for the first large cluster. However, there are no particular correlated structures recognizable.

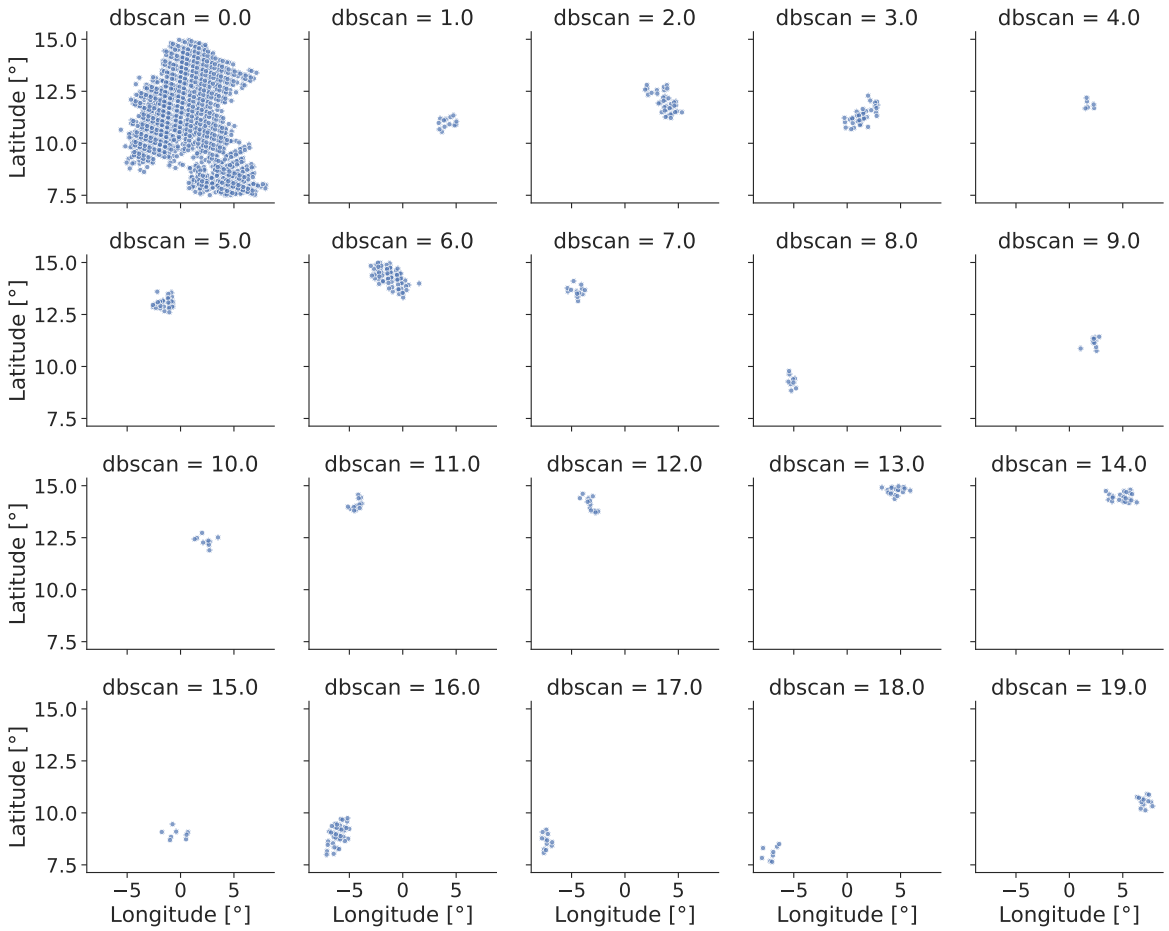


Figure 5.17: Example DBSCAN clustering result for the real-world data at 2016-06-08 in the area of interest. Each plot shows a single cluster in the geo-referenced latitude/longitude space, 20 clusters in total.

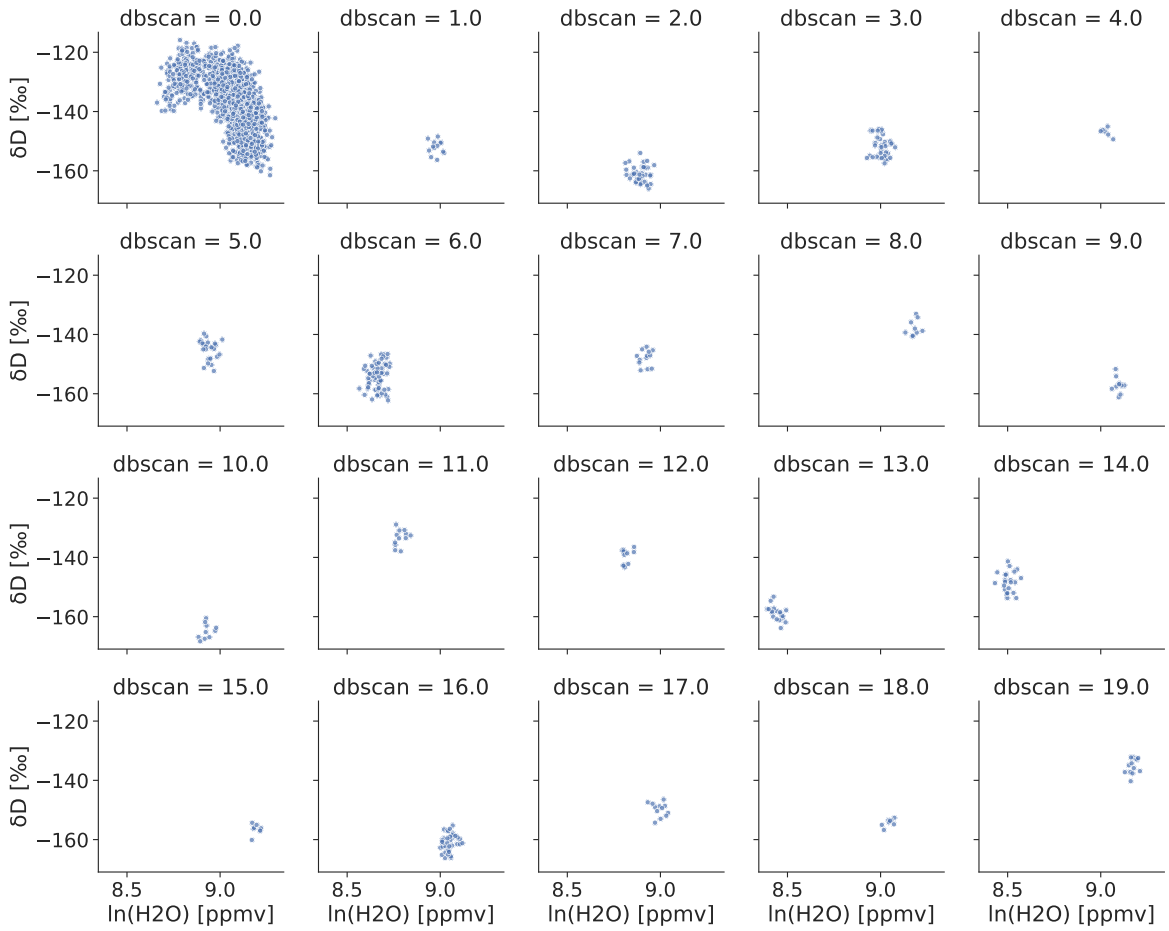


Figure 5.18: Example DBSCAN clustering result for the real-world data at 2016-06-08 in the area of interest. Each plot shows a single cluster in the $\{H_2O, \delta D\}$ value space, 20 clusters in total.

This visual intuition can be qualified by looking at the distribution of the correlation coefficient values and their p-values, see Figure 5.19 and Figure 5.20. The values of the Pearson correlation coefficient, either negative or positive, for the DBSCAN clusters are weak correlations with p-values well above an assumed cutoff at 0.05 as indicated by the red dashed line in Figure 5.20.

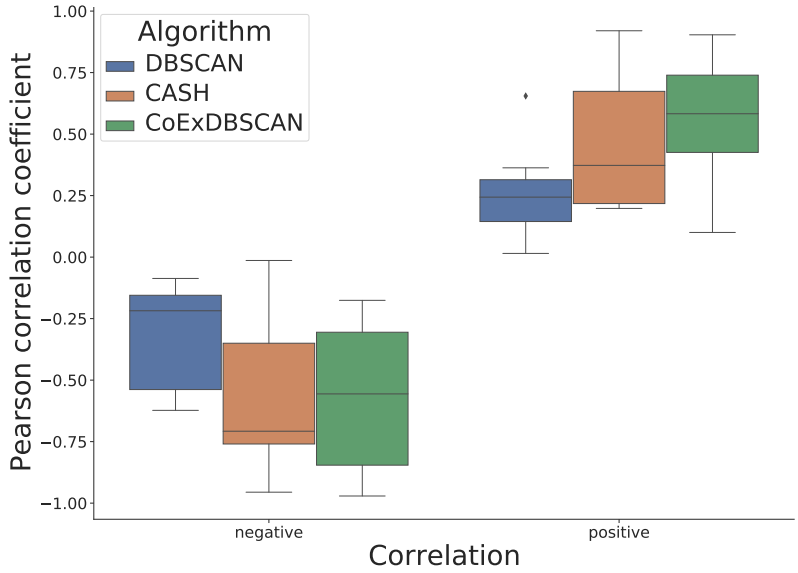


Figure 5.19: Distribution of correlation coefficient values for the DBSCAN, CASH and CoExDBSCAN algorithms.

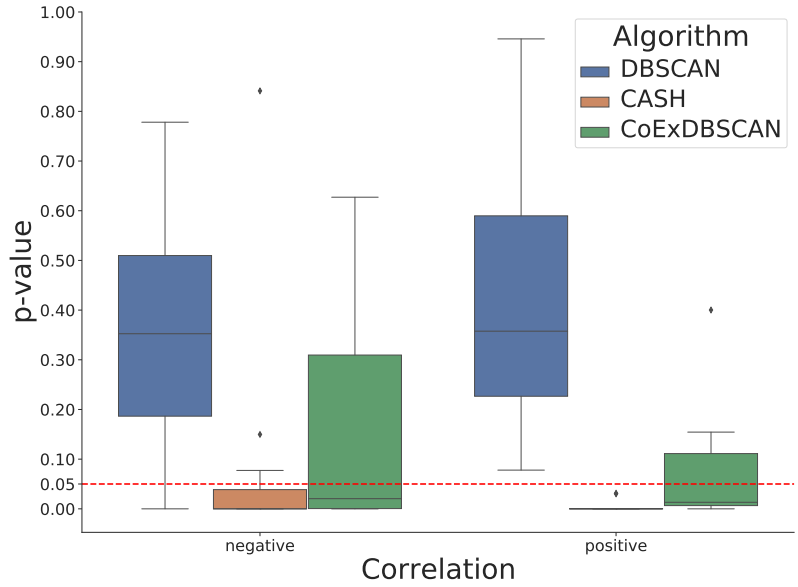


Figure 5.20: Distribution of p-values for the DBSCAN, CASH and CoExDBSCAN algorithms.

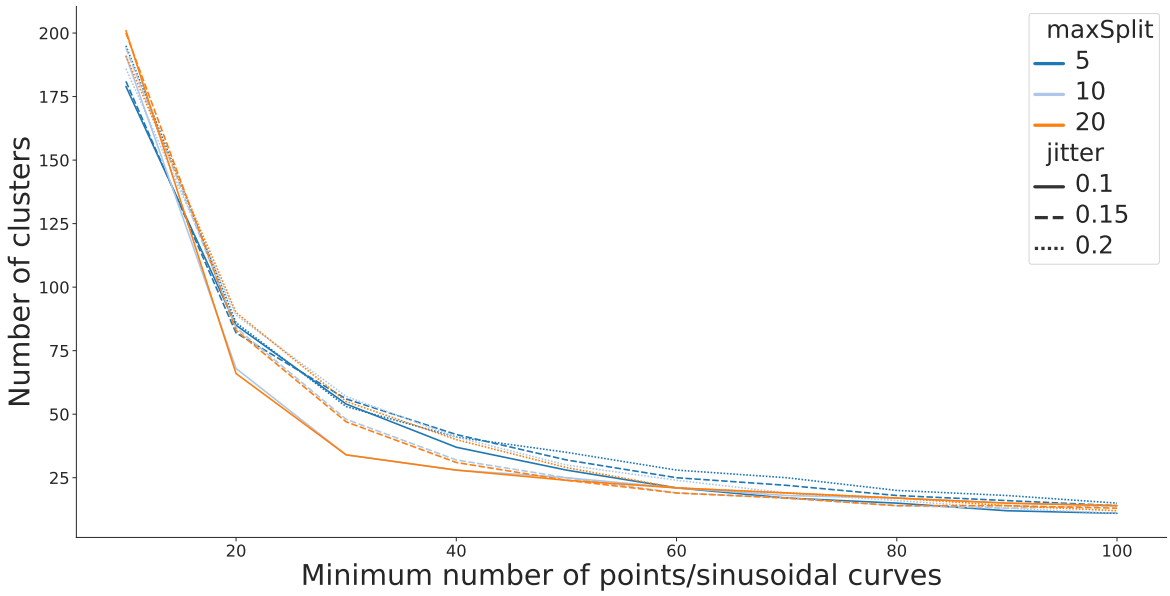


Figure 5.21: Dependencies between $minPts$, $maxSplit$ and $jitter$ parameters and the number of clusters identified by CASH.

In comparison, the CASH algorithm can produce clustering results with the same level of granularity, i.e. the same number of clusters, with the parameters $minPts = 80$, $maxSplit = 5$ and $jitter = 0.20$, that have been determined via grid search on the parameter space $minPts \in [10, 100]$, $maxSplit \in \{5, 10, 20\}$ and $jitter \in \{0.10, 0.15, 0.20\}$, see Figure 5.21.

The clusters in the geo-referenced latitude and longitude space for the CASH algorithm illustrated in Figure 5.22 are not as concise and spatial discrete as the clusters identified by the DBSCAN algorithm. However, in the $\{H_2O, \delta D\}$ feature space, the clusters found by the CASH algorithm exhibit noticeably more of the correlated structures than the clusters found by the DBSCAN algorithm. The values of the Pearson correlation coefficient, either negative or positive, for the CASH clusters are in general higher than the values for the DBSCAN clusters, as shown in Figure 5.19. The p-values are all below an assumed cutoff at 0.05, and therefore can be assumed to be significant, as indicated by the red dashed line in Figure 5.19, except for negative correlations where there are some values in the upper 1.5× Interquartile range (IQR) above 0.05.

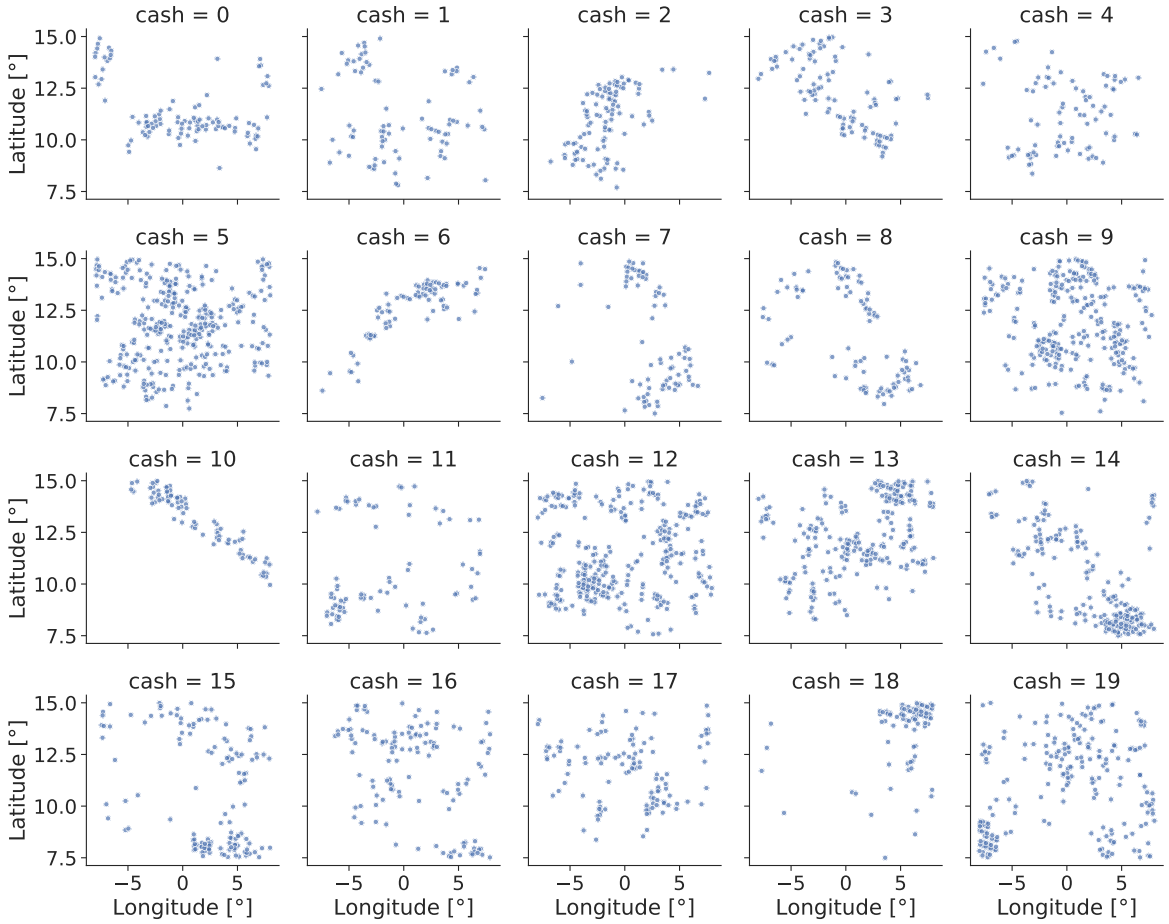


Figure 5.22: Example CASH clustering result for the real-world data at 2016-06-08 in the area of interest. Each plot shows a single cluster in the geo-referenced latitude/longitude space, 20 clusters in total.

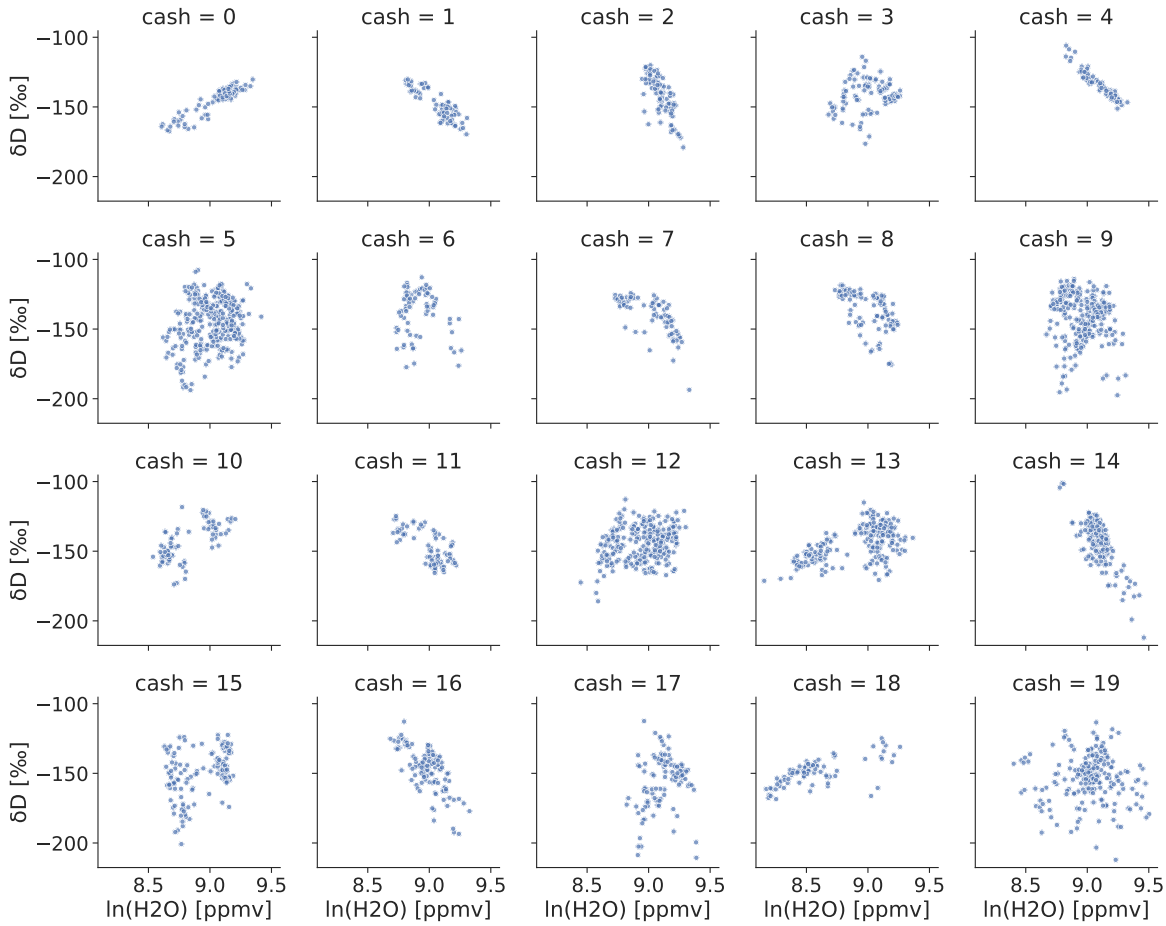


Figure 5.23: Example CASH clustering result for the real-world data at 2016-06-08 in the area of interest. Each plot shows a single cluster in the $\{H_2O, \delta D\}$ value space, 20 clusters in total.

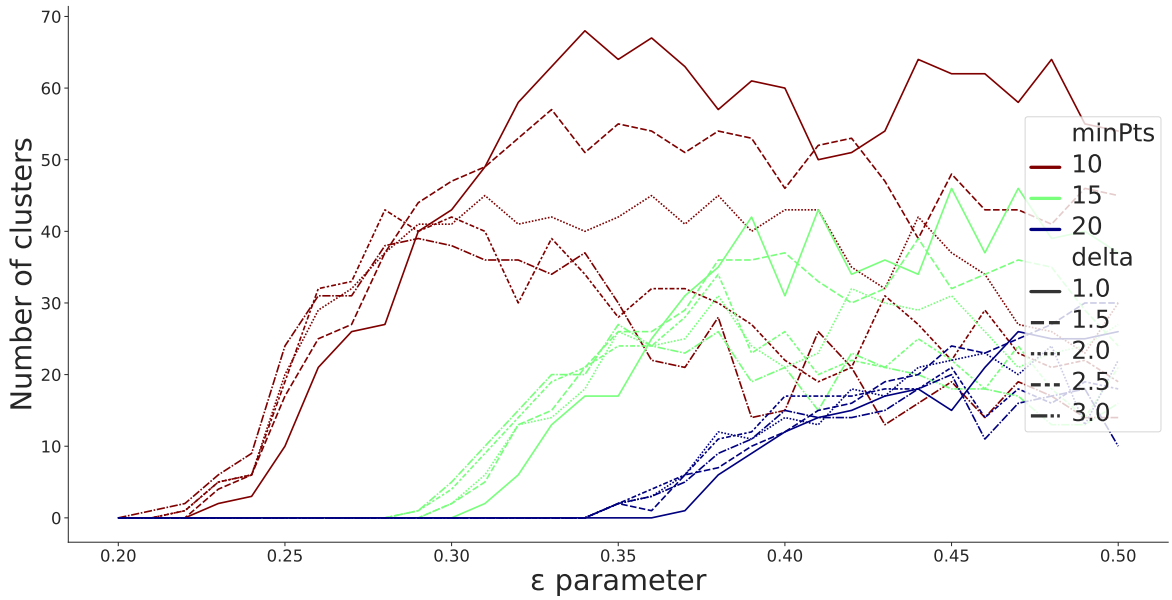


Figure 5.24: Dependencies between ϵ, $minPts$ and δ parameters and the number of clusters identified by CoExDBSCAN.

The CoExDBSCAN clustering results with the same level of granularity can be achieved with the parameter set $\epsilon = 0.25$, $minPts = 10$ and $\delta = 2.0$. These parameters have been determined via grid search on the parameter space $\epsilon \in [0.20, 0.50]$ with a step size of 0.01, $minPts \in [10, 20]$ with a step size of 5 and $\delta \in [1.0, 3.0]$ with a step size of 0.5, see Figure 5.24.

The clusters in the geo-referenced latitude and longitude space for the CoExDBSCAN algorithm illustrated in Figure 5.25 are very concise and spatial discrete; further, the clusters exhibit the correlated structures to a high degree in the $\{H_2O, \delta D\}$ feature space. For positive correlations, the values of the Pearson correlation coefficient for the clusters are higher (median, lower and upper quartile) than the coefficient values for the CASH and DBSCAN clusters. For negative correlations, the median of the Pearson correlation coefficients for the clusters is below the median for the CASH correlation coefficients, but higher than the median for the DBSCAN correlation coefficients. The median p-values for both negative and positive correlations are below an assumed cutoff at 0.05, and therefore can be assumed to be significant (Figure 5.19). In comparison to the clusters found by DBSCAN, the clusters found by CoExDBSCAN are spatially separated as well, but better represent the a priori assumed linear correlation in the $\{H_2O, \delta D\}$ feature space.

Compared to the clusters found by CASH, some represented correlations are not as strong and significant as the represented correlations by the CASH clusters. However, the spatial coherence outweighs this fact for the objective of the analysis to identify correlated structures that are geographically close and that are expected to follow theoretical assumptions.

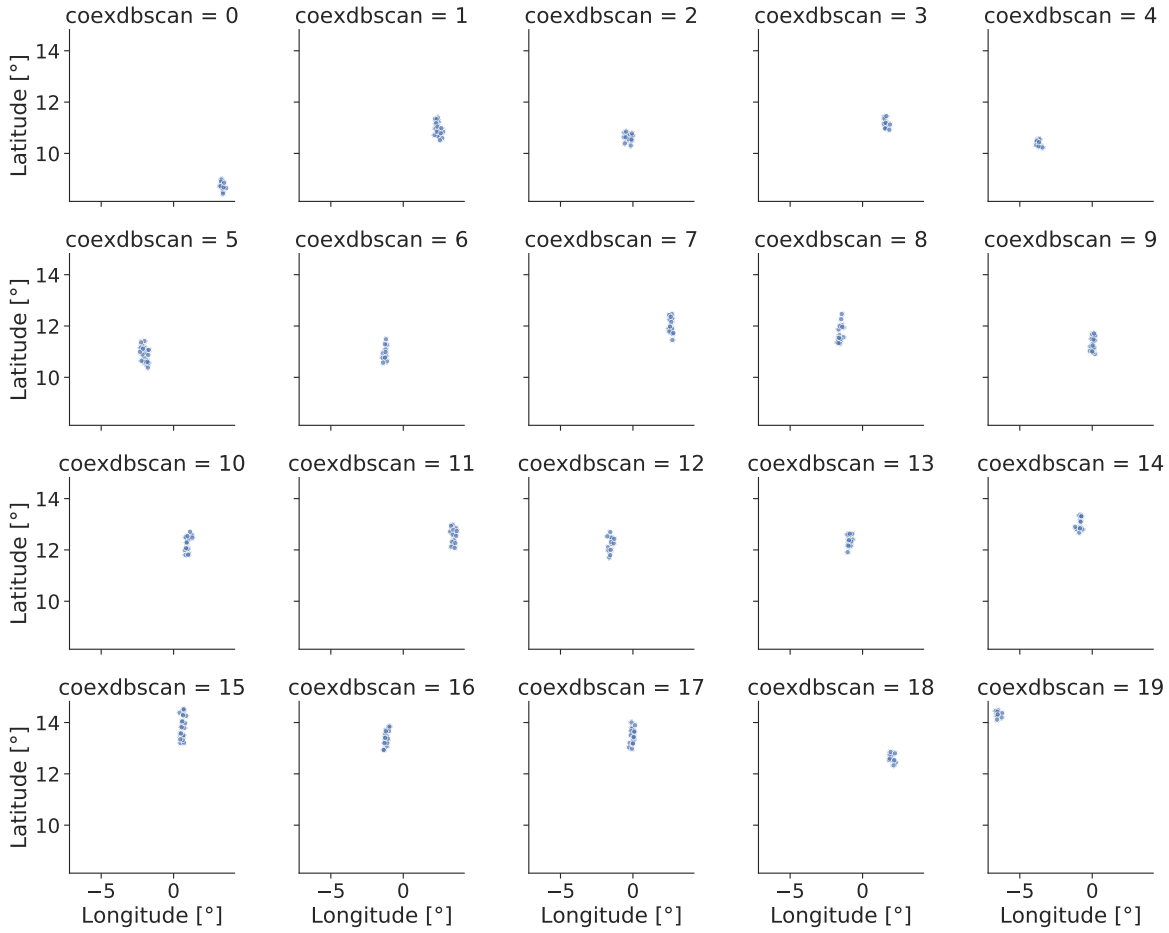


Figure 5.25: Example CoExDBSCAN clustering result for the real-world data at 2016-06-08 in the area of interest. Each plot shows a single cluster in the geo-referenced latitude/longitude space, 20 clusters in total.

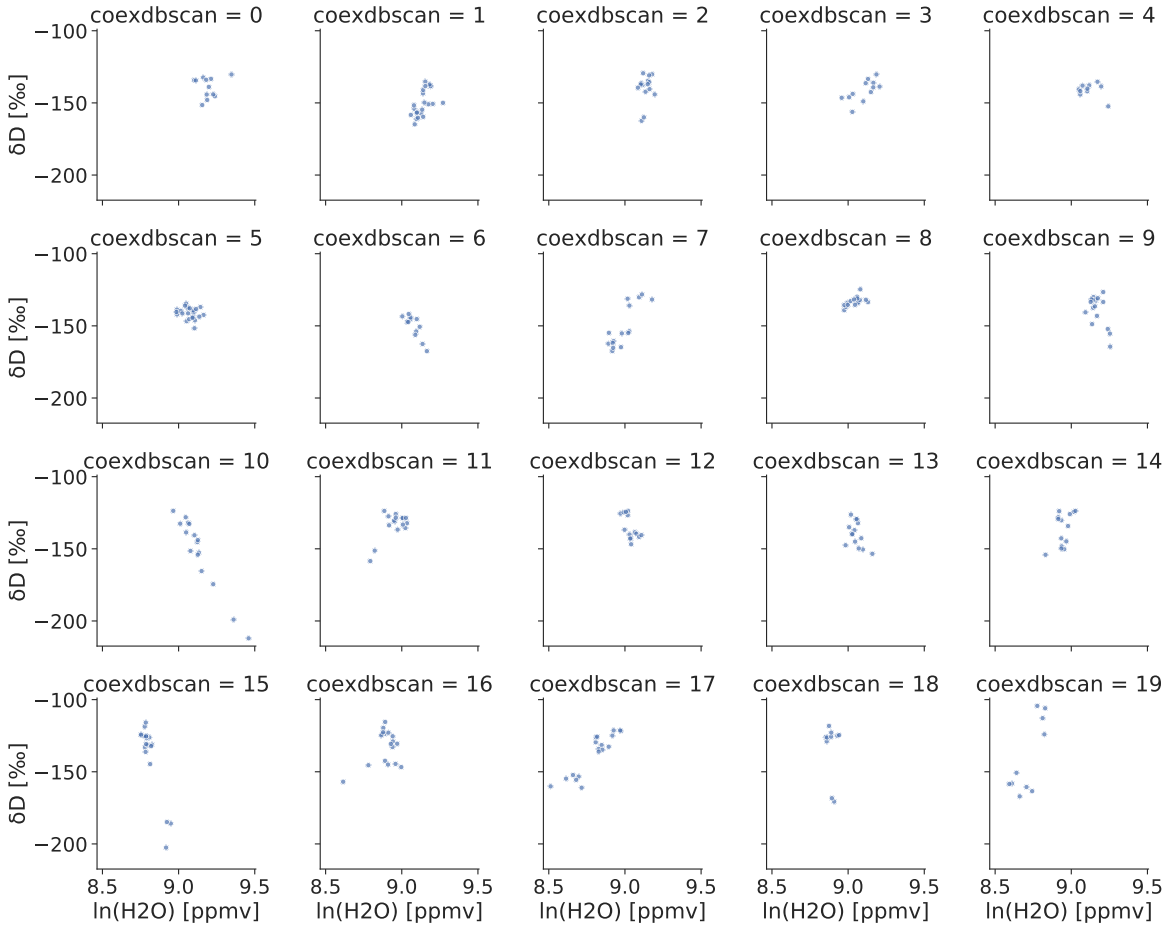


Figure 5.26: Example CoExDBSCAN clustering result for the real-world data at 2016-06-08 in the area of interest. Each plot shows a single cluster in the $\{H_2O, \delta D\}$ value space, 20 clusters in total.

5.4 Summary

This chapter introduced a new density-based clustering algorithm with constrained cluster expansion, namely CoExDBSCAN. CoExDBSCAN uses DBSCAN to find density-connected clusters in a defined subspace of features and restricts the expansion of clusters to a priori constraints. Incorporating a priori knowledge into the clustering process can significantly improve the clustering results. It can align the outcome of the clustering process with the objective of the data analysis, as demonstrated in Section 5.3 for synthetic and real-world data. The proposed approach combines different techniques from subspace, correlation and constrained cluster analysis. In particular, two user-defined parameters are introduced to the original DBSCAN algorithm, one to define the dimensions of the subspace to be used to discover density-based clusters, and one to define the dimensions of the subspace to be used to apply constraints to the cluster expansion, i.e. restricting the cluster expansion step in the original DBSCAN algorithm to the provided user-defined constraints.

The validation of the algorithm on the synthetic and real-world dataset demonstrates that CoExDBSCAN is especially suited for spatio-temporal data, where one subspace of features defines the spatial extent of the data and another the correlations between features. In the presented verification, CoExDBSCAN outperforms the DBSCAN algorithm and the CASH algorithm in terms of Cluster Accuracy (ACC) and Adjusted Rand Index (ARI) on the synthetic dataset. To achieve a similar accuracy with the semi-supervised clustering algorithms CK-means and PCK-means on the presented synthetic dataset, around 55% of true labels or around $2,200$ pairwise constraints have to be provided a prior. In addition, the CK-means and PCK-means algorithms need to know the number of clusters to form in advance, whereas the CoExDBSCAN is able to explore the given dataset without this kind of restriction. The presented verification on the real-world climatological dataset demonstrates that CoExDBSCAN is better suited to identify correlated structures that are geographically close than the original DBSCAN algorithm as well as the CASH algorithm.

Schubert et al. (2017) showed that the original DBSCAN algorithm continues to be relevant even for high-dimensional data, although the parameters become hard to choose in high-dimensional data due to the loss of contrast in distances. Since CoExDBSCAN is based on DBSCAN and overcomes the issue of loss of contrast in distances by utilizing a user-defined subspace for the distance measure, CoExDBSCAN also remains relevant for high-dimensional data beyond the presented low-dimensional verification and evaluation datasets. However, finding and expressing suitable constraints is a challenging task. A generic constraint that allows the algorithm to expand clusters for arbitrarily correlated data points has been proposed in the presented verification.

Generic constraints can avoid overfitting the clustering algorithm, i.e. avoid constraining the cluster expansion to the generating process. Whereas specially tailored constraints, for example, if the information about the functions that generate the dependent y and z variables in the example synthetic dataset is given as constraints, a perfect match to the true labels of the dataset can be achieved, but the generality of the algorithm would be lost. To simplify the process of defining constraints, methods from the field of active learning can be included in the data analysis workflow (Zhu, 2005; Settles, 2009) to provide appropriate constraints to the CoExDBSCAN algorithm. Furthermore, a machine-learning-based selection of suitable constraints can additionally aid the user in applying the algorithm to new datasets.

Besides the challenge of finding and expressing suitable constraints, finding the right parameters for the CoExDBSCAN algorithm remains another challenge, especially for high-dimensional data. In addition to the parameters of the DBSCAN algorithm, the dimensions for the spatial- and constraint-subspace have to be determined by the user. In the presented verification, the parameters of the CoExDBSCAN algorithm have been selected based on hyperparameter optimization, in particular grid search, while varying the selected dimensions based on domain knowledge and the expected outcome of the analysis.

Chapter 6

Semi-Supervised Time Point Clustering and Trajectory Segmentation

The semi-supervised clustering algorithm CoExDBSCAN introduced in the previous chapter effectively identifies correlated structures in spatio-temporal datasets by forming partitions of data complying with a priori constraints in full value space or value subspaces. Moreover, constraining the cluster extension to the correlation of time point values in multivariate time-series enables the CoExDBSCAN algorithm to form clusters of segments with similar correlations. This approach can be utilized for semi-supervised time point clustering as well as trajectory segmentation which is described in the following sections.

The results presented in this chapter have been published in parts in Ertl et al. (2021a) and Ertl et al. (2021b).

6.1 Motivation

The increasing amount of data produced over time by a variety of sensors and scientific instruments available through new technologies and increasing storage capacity provides unique opportunities to discover characteristics and structures reflected by meaningful clusters in such time-series. Especially recurring subsequences in streams of multiple measurements that can be organised as multivariate time-series can be interpreted as recurring events or actions. These recurring events can be used to discover repeating patterns, understanding trends, detect anomalies and in general, better interpret large and high-dimensional datasets (Hallac et al., 2017).

By utilizing the proposed CoExDBSCAN algorithm for such time-series and constraining the cluster extension to the correlation of time point values, clusters of segments with similar correlations can be identified. This novel semi-supervised approach for subsequence time-series clustering follows the pairwise semi-supervision approach, see Section 3.2.2, and extends the concept to a novel concept of cluster-wide constraints.

6.2 CoExDBSCAN Adaptation

This section provides the necessary definitions and details about the adaptation of the CoExDBSCAN algorithm, as well as the formulation of the constraints to restrict the cluster expansion.

Rodpongpun et al. (2012) provide the following definitions on subsequence time-series clustering, i.e. Definition 15 and 16.

Definition 15. *A time-series T of size m is an ordered sequence of real-value data, where $T = (t_1, t_2, ..., t_m)$.*

Definition 16. *A subsequence $T_{i,n}$ of length n of time-series T is $T_{i,n} = (t_i, t_{i+1}, ..., t_{i+n-1})$, where $1 \leq i \leq m - n + 1, n < m$.*

Definition 16 can be extended to allow elements to be omitted as following.

Definition 17. *A subsequence T_S of length n of time-series T is an arranged sequence of data that omits some elements without changing the order of the remaining elements. $T_S = (t_{s_1}, t_{s_2}, ...t_{s_n})$, where $|T_S| = n$ and $\forall i \in [1, n] : s_i < s_{i+1}$.*

The CoExDBSCAN algorithm extends the definition of the ϵ-neighbourhood of the original DBSCAN algorithm, see Definition 1, to the CoExDBSCAN ϵ-neighbourhood of a point, see Definition 12. The transition from Definition 1 to Definition 12 allows to define the temporal order of the data points as the spatial subspace and to provide a constraint function that is evaluated in another subspace for the clustering algorithm. With this configuration of the CoExDBSCAN algorithm, the ϵ-neighbourhood describes a neighbourhood of lagged points, similar to a time window, where the maximum lag in time for the initial data points is defined by the ϵ parameter and the minimal amount of data points that are required to form a cluster is defined by the $minPts$ parameter, see Definition 2 for the definition of direct density-reachable points of the original DBSCAN algorithm.

For any data point t_i at time i the algorithm considers all points t_j at times $j \in [i - \epsilon, i + \epsilon]$ as candidates for an initial subsequence. If all constraints are satisfied for any t_j, t_i and t_j belong to the same subsequence, which is further extended at

point t_j. All resulting subsequences follow Definition 17 and all points within each subsequence satisfy all constraints. If any, the omitted elements from one subsequence are either belonging to another overlapping subsequence or are disregarded as noise.

A constraint formulation that has been empirically proven to be especially suited for correlated data can be determined as follows, see Definition 18.

Definition 18. *A point t_j belongs to a subsequence T_S of length n of a time-series T of length m, with $T_S = (t_{s_1}, t_{s_2}, ...t_{s_n})$ iff*

$$(Y_{t_j} - \hat{Y}_{t_j})^2 < \delta \cdot \frac{1}{n} \sum_{k=s_1}^{s_n} (Y_{t_k} - \hat{Y}_{t_k})^2 \tag{6.1}$$

where Y and $\hat{Y}$ are the dependent variable and fitted value of the linear regression respectively.

For each evolving subsequence, the residuals of an ordinary least squares linear regression are computed, and neighbouring points are included in this subsequence if and only if the square of the residual of a neighbouring point deviates from the mean of the square of the residuals of the current points in the subsequence only by a certain factor δ. This δ has to be determined either via parameter selection, see Claesen and De Moor (2015) for examples on current hyperparameter search approaches, or via a priori knowledge about the nature of the time-series.

After splitting the time-series into subsequences, all sequences with less than required data points ($reqPts$) are labelled as noise; such sequences can appear if the $minPts$ parameter has been set to a small number and the sequence could not be expanded due to the given constraint. Second, the regression coefficients for each remaining subsequence is computed, and sequences with equal or slightly different regression coefficients for the dependent variable are grouped into the same cluster. The threshold for different coefficients has to be determined the same way as the δ parameter, either via parameter selection, for example, grid-search, or via a priori knowledge about the nature of the time-series. This process can be repeated for multiple time-series and will result in clusters of subsequences as following.

Definition 19. *A cluster C is a set of subsequences of one or multiple time-series, $C = \{T_S^{(l)}\}$ for $1 \leq l \leq N$, where N is the total number of subsequences, where each time point in every subsequence satisfies the conditions formulated in Definition 2 and Definition 18, and each subsequence of points $T_S^{(l)}$ satisfies the following conditions:*

1. $\forall T_S^{(l)} : \left|T_S^{(l)}\right| \geq reqPts$ *(subsequences with more than reqPts)*
2. $\forall T_S^{(l)}, T_S^{(o)} \in C : \|\beta_l - \beta_o\| < \theta$ *(regression coefficients close),*

where β_l, β_o are the regression coefficients of a linear regression of all time points in subsequences $T_S^{(l)}, T_S^{(o)}$ respectively for a threshold θ.

It should be noted that the constraint definition (Definition 18) has been specifically formulated to cluster linear segments with similar regression coefficients for a given time-series. This makes the presented approach especially suited for time-series that exhibit such inherent characteristics, for example, correlated events in the feature space. However, the approach could be used to find non-linear segments as well by designing an appropriate constraint or multiple constraints, i.e. by utilizing a non-linear regression instead of the applied linear regression.

6.3 Evaluation

6.3.1 Setup

This section details the experimental studies to evaluate the proposed model for semi-supervised time point clustering and semi-supervised trajectory segmentation.

Following Definition 15 to 19 the general approach for semi-supervised time point clustering and semi-supervised trajectory segmentation for multivariate time-series can be summarised in four steps.

1. Compute the CoExDBSCAN clustering result for each time-series with the time dimension as the spatial subspace and the correlated features as the correlation subspace with the constraint formulated in Definition 18. Each cluster is equivalent to a subsequence.

2. Label all subsequences with less than the minimum required number of time points as noise, if any.

3. Compute the linear regression coefficients between the correlated features for each subsequence.

4. Group all subsequences with equal or close regression coefficients up to a certain threshold into one cluster. The resulting clusters, see Definition 19, contain segments of one or multiple time-series that are similar to each other in terms of temporal proximity and correlation of the comprising time points.

Semi-Supervised Time Point Clustering

For verification and comparison, a synthetic dataset is generated that has known correlations between two features and known temporal subsequences. Because the correlations and the order of subsequences are known, all methods can be evaluated against the ground truth. The Adjusted Rand Index (ARI) and the Cluster Accuracy (ACC) are used as metrics, see Chapter 2 (Section 2.4.4).

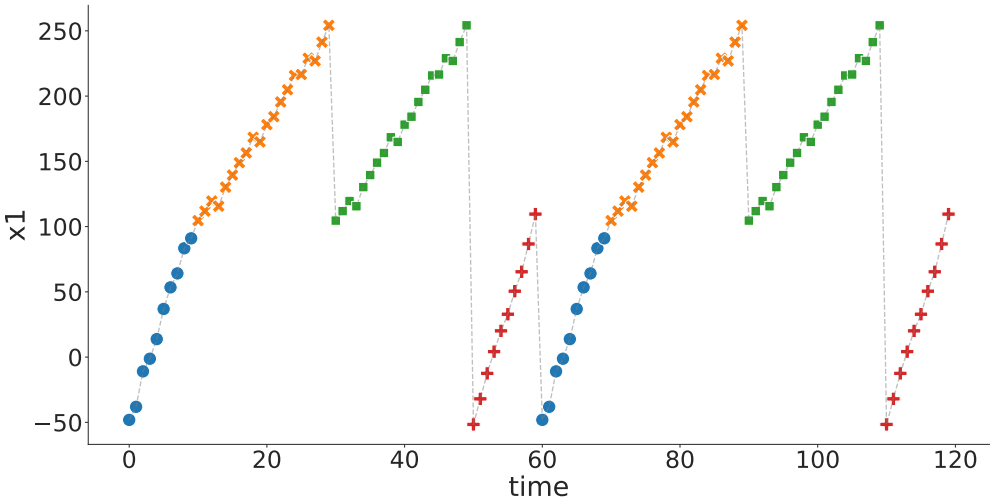
The synthetic dataset has four temporal sequences that are repeated once: "1, 2, 3, 4, 1, 2, 3, 4", illustrated in Figure 6.1a and Figure 6.1b. Each sequence has two correlated features (x1 and x2) that are generated according to Table 6.1.

For each sequence, one feature is evenly distributed on the given interval plus some randomly distributed noise, and the other feature follows a specific linear equation plus some randomly distributed noise that leads to overlapping areas in the joint feature space as depicted in Figure 6.1c. This overlap in feature space is particularly challenging for cluster algorithms, since distance-based and density-based algorithms can not distinguish between the overlapping clusters without a priori information.

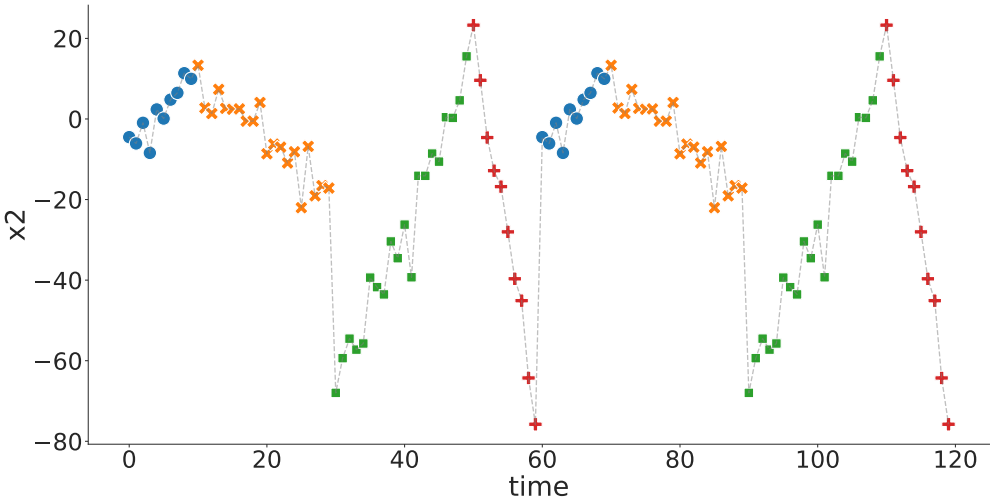
To demonstrate the significance of the approach for real-world data, the LÍngua BRAsileira de Sinais (LIBRAS) movement dataset has been chosen (Dias et al., 2009) to cluster similar partial movements across different movement patterns. The LIBRAS movement dataset is available from the University of California, Irvine (UCI) Machine Learning Repository (Dua and Graff, 2017) and contains 15 classes of 24 instances each, where each class references to a hand movement type in the LIBRAS Brazilian sign language. All movements were tracked from video analysis, where in each frame, the centroid pixels of the segmented objects (the hand) are found, which compose the discrete version of a curve with 45 points. All curves are normalized in the unitary space and mapped in a representation with 90 features, representing the coordinates of the movement (Dua and Graff, 2017). Each hand movement can be further segmented in distinct sub-movements, e.g. upper left to lower right, therefore makes it suitable for subsequence and time point clustering.

Sequence	Points	Feature x1	Feature x2	Noise
1	10	$x1 \in [-50, 100] + \xi$	$x2 = \quad 0.1 \cdot x1 + \xi$	$\xi \sim \mathcal{N}(0, \sigma^2)$
2	20	$x1 \in [100, 250] \; + \xi$	$x2 = -0.2 \cdot x1 + 39.65 + \xi$	$\xi \sim \mathcal{N}(0, \sigma^2)$
3	20	$x1 \in [100, 250] \; + \xi$	$x2 = \quad 0.5 \cdot x1 - 106.94 + \xi$	$\xi \sim \mathcal{N}(0, \sigma^2)$
4	10	$x1 \in [-50, 100] + \xi$	$x2 = -0.6 \cdot x1 + 4.52 + \xi$	$\xi \sim \mathcal{N}(0, \sigma^2)$

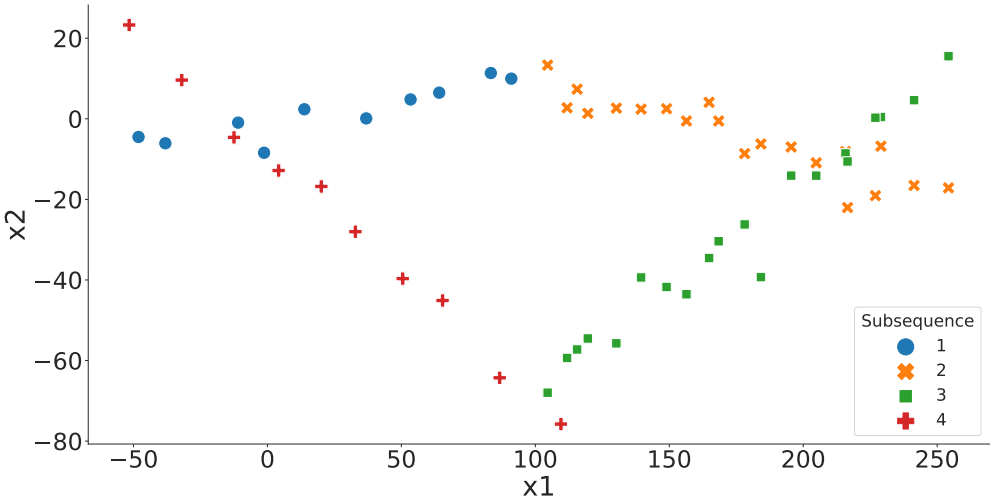
Table 6.1: Value range and generation methods.



(a) Time-series for feature x1.



(b) Time-series for feature x2.



(c) Joint feature space {x1,x2}.

Figure 6.1: Example synthetic dataset with noise $\xi \sim \mathcal{N}(0, 4^2)$; labels are true labels.

Algorithm 3: Semi-Supervised Time Point Clustering

input : set of time-series T
input : time radius ϵ
input : density threshold $minPts$
input : residual threshold δ
output : point labels $labels_T$ initially $undefined$

1 **foreach** *time-series* t *in time-series* T **do**
2 $\quad labels_t$ = CoExDBSCAN($t.time$, $t.\{x_1, ..., x_n\}$, ϵ, $minPts$, δ);
3 $\quad labels_t$ = labelOnePointClusterAsNoise($labels_t$);
4 $labels_T$ = groupByCoefficient(T, $\{labels_t\}$);

However, there are no true labels available for individual segments of the LIBRAS movement dataset, and thus external clustering validation with the ARI or the ACC metrics can not be performed for the real-world data.

Algorithm 3 gives a pseudocode representation of the semi-supervised time point clustering approach. The algorithm takes a set of multivariate time-series as input, as well as the parameters and subspace selections for the CoExDBSCAN algorithm. For each individual time-series, Line 1, the algorithm computes the CoExDBSCAN clustering result, Line 2, with the time dimension ($t.time$) as the spatial subspace and the correlated features ($t.\{x_1, ..., x_n\}$) as the correlation subspace. In addition, the DBSCAN parameters ϵ and $minPts$ are provided, as well as the residual threshold parameter δ for the constraint formulated in Definition 18. After the labels for a specific time-series have been computed, all clusters, i.e. subsequences, with less than the minimum required number of time points are labelled as noise (see Line 3). The last step of the algorithm (Line 4) takes the set of time-series T and all labels for each individual time-series $\{labels_t\}$ and groups similar subsequences into final clusters labelled by $labels_T$. The similarity can be expressed, for example, based on a threshold on the absolute difference of coefficient values or by quantization of the coefficients.

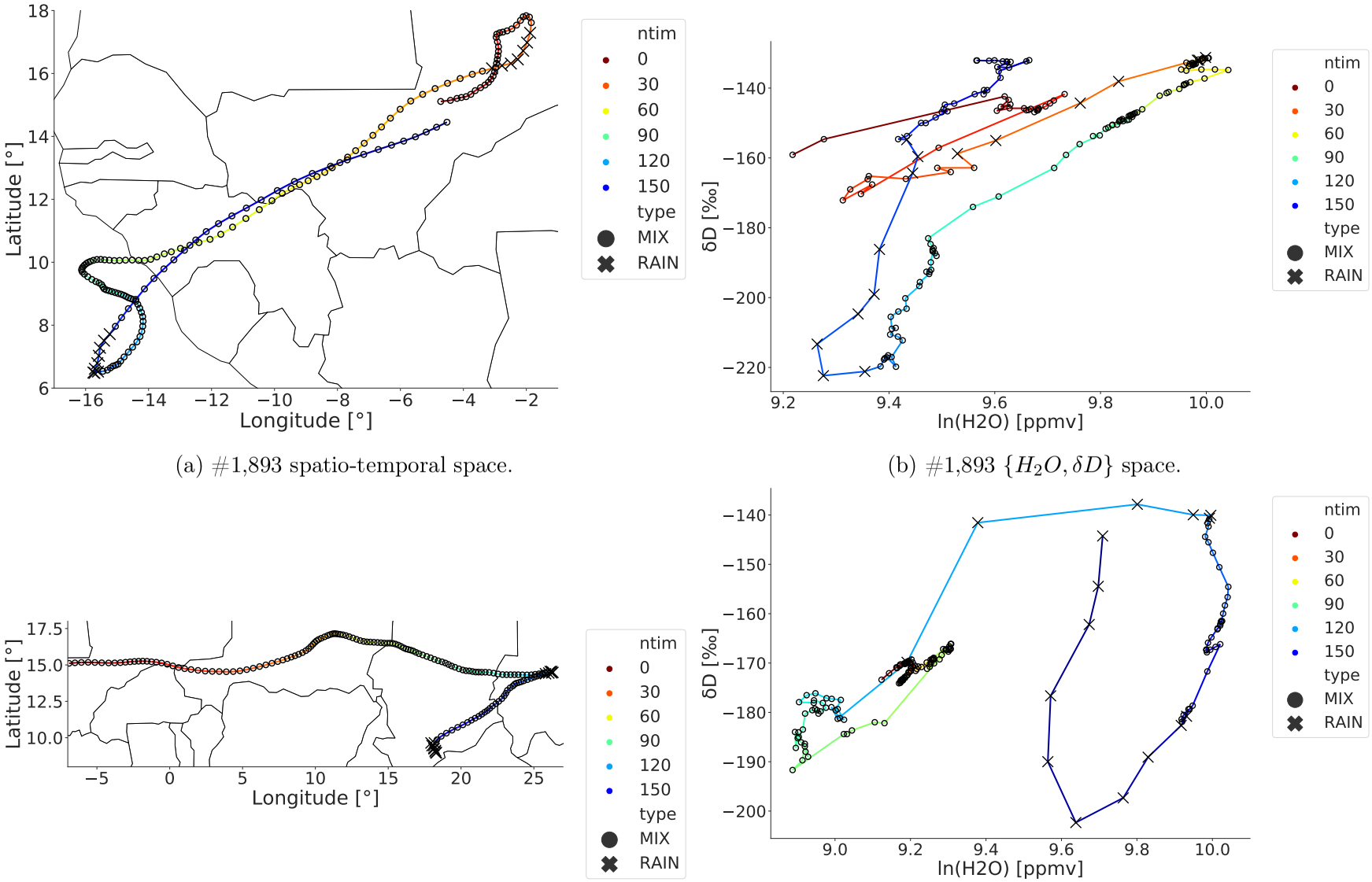
Semi-Supervised Trajectory Segmentation

Similar to the approach for semi-supervised time point clustering detailed in the previous subsection, the CoExDBSCAN algorithm can be adopted for semi-supervised trajectory segmentation as well. In the following, a trajectory is considered as a time-series according to Definition 15, and a subsequence is considered an arranged sequence of data that omits some elements without changing the order of the remaining elements according to Definition 17.

For the demonstration of this approach, trajectory data from a climatological model has been chosen. The data contains $\{H_2O, \delta D\}$ pairs modelled along Lagrangian air parcel trajectories. As climatological model the high-resolution data from the regional isotope-enabled atmospheric model COSMO-iso (Pfahl et al., 2012) is used and the trajectories are determined with the tool LAGRANTO (Sprenger and Wernli, 2015). The trajectories' calculation setup is oriented towards the overpass times and altitudes representative for the MUSICA IASI data, presented in Chapter 2 (Section 2.2). Analysing the model data allows revealing the kind of moisture processes that can be observed in the MUSICA IASI $\{H_2O, \delta D\}$ pair data. Theoretical and observational findings by Noone et al. (2011) can be utilised in the experimental evaluation to identify atmospheric moisture processes that correspond to different $\{H_2O, \delta D\}$ pair distribution for different segments of the trajectory data. The focus for the segmentation of the $\{H_2O, \delta D\}$ pairs modelled along Lagrangian air parcel trajectories is to identify different rain events, where Rayleigh pseudo adiabatic, Super-Rayleigh, and reversible moist adiabatic processes affect the $\{H_2O, \delta D\}$ pair distribution, in contrast to non-rain events, where air mass mixing processes are dominating the $\{H_2O, \delta D\}$ pair distribution. Example trajectories are illustrated in Figures 6.2a to 6.3d in spatial space and value space. Each trajectory has rain phases (marked with crosses) and non-rain/mixing phases (marked with circles). The timely order of single data points is coloured from deep blue (168 hours before arrival) to dark red (0 hours, time of arrival).

Algorithm 4 gives a pseudocode representation of the semi-supervised trajectory segmentation approach. The algorithm takes a set of trajectories as input, as well as the parameters and subspace selections for the CoExDBSCAN algorithm. The identified trajectories with a subset of rain events are sorted by time in ascending and descending order; see Line 2 and 4 in Algorithm 4. For each time ordering the labels are computed using CoExDBSCAN with the time dimension, $timePoints.time$, as the spatial subspace and the correlated features ($\{x_1, ..., x_n\}$) as the correlation space. In addition, the DBSCAN parameters ϵ and $minPts$ are provided, as well as the residual threshold parameter δ for the constraint formulated in Definition 18.

Computing the segmentation in both temporal orders is necessary because the outcome of the linear regression of the constraint depends on the deviation of the residuals from the current cluster points, which can be different following the trajectory points in ascending or descending temporal order. The final segmentation of the trajectory, or the subset of time points within a trajectory, is the selection of phases from the ascending and descending CoExDBSCAN run where the outcome with the squared residual sum is lowest, Line 6 to 11 in the algorithm.



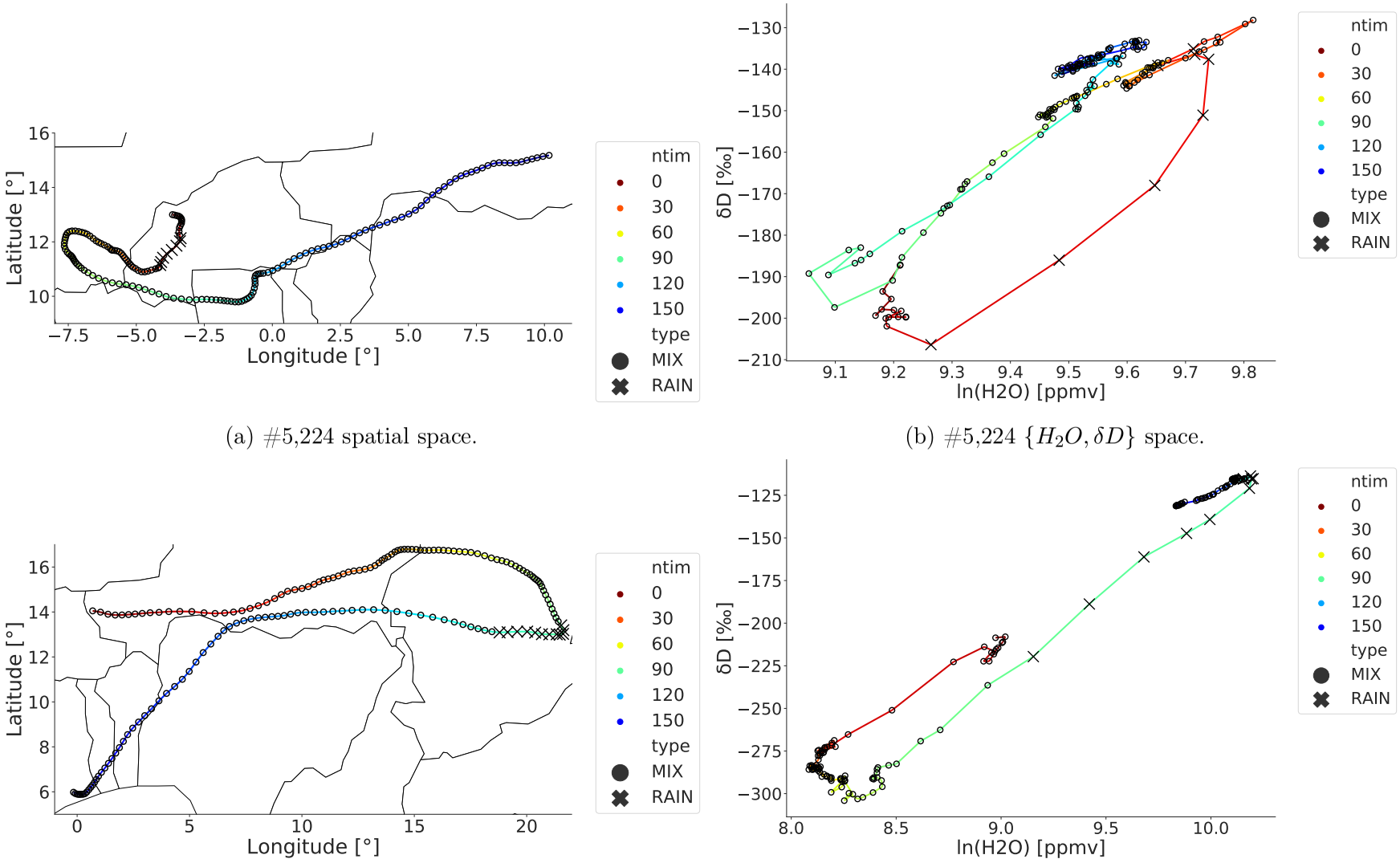
(a) #1,893 spatio-temporal space.

(b) #1,893 $\{H_2O, \delta D\}$ space.

(c) #2,011 spatial space.

(d) #2,011 $\{H_2O, \delta D\}$ space.

Figure 6.2: Example trajectories #1,893 and #2,011.



(a) #5,224 spatial space.

(b) #5,224 $\{H_2O, \delta D\}$ space.

(c) #11,003 spatial space.

(d) #11,003 $\{H_2O, \delta D\}$ space.

Figure 6.3: Example trajectories #5,224 and #11,003.

Algorithm 4: Semi-Supervised Trajectory Segmentation

input : trajectories T
input : time radius ϵ
input : density threshold $minPts$
input : residual threshold δ
output : point labels per trajectory $label_t$ initially $undefined$

```
1  foreach trajectory t in trajectories T do
2      timePoints = sortByTime(t, ascending);
3      phasesAscending = CoExDBSCAN(timePoints.time,
         timePoints.{x_1, ..., x_n}, ε, minPts, δ);
4      timePoints = sortByTime(t, descending);
5      phasesDescending = CoExDBSCAN(timePoints.time,
         timePoints.{x_1, ..., x_n}, ε, minPts, δ)
6      foreach phaseAscending in phasesAscending do
7          foreach phaseDescending in phasesDescending do
8              if sum(OrdinaryLeastSquares(phaseAscending).residuals^2) < sum(
                 OrdinaryLeastSquares(phaseAscending).residuals^2) then
9                  label_t ← phaseAscending;
10             else
11                 label_t ← phaseAscending;
```

6.3.2 Results

Semi-Supervised Time Point Clustering

To verify the proposed approach for semi-supervised time point clustering detailed in Section 6.3.1, the results of the presented method are compared to a baseline k-means clustering for time-series data with Dynamic Time Warping (DTW), a Gaussian Mixture Model (GMM) and the Toeplitz Inverse Covariance-based Clustering (TICC) method that are presented in Chapter 3 (Section 3.1.2) in more detail.

The k-means with DTW algorithm is a well established similarity-based method for time-series and time point clustering and is available in a variety of programming languages. The Gaussian Mixture Model is a general, model-based approach that provides a sound mathematical-based approach for statistical modelling of a wide variety of random phenomena as detailed in Chapter 2 (Section 2.4.2). For a state-of-the-art comparison, the Toeplitz Inverse Covariance-based Clustering (TICC) method has been chosen (see Chapter 3, Section 3.1.2) since the authors have shown that their method outperforms a range of model-based and distance-based clustering methods for clustering multivariate time-series in subsequences. The implementation of the modified CoExDBSCAN algorithm is based on Python, the implementation of the k-means with DTW algorithm is provided by the tslearn machine learning toolkit for time-series data in Python (Tavenard et al., 2020) and the GMM implementation is provided by the scikit-learn machine learning package in Python (Pedregosa et al., 2011); the code of the TICC method is also provided in Python by the authors Hallac et al. (2017).

Experiments showed that varying the standard deviation of the noise for the synthetic data leads to different performances of the compared algorithms. Figure 6.4 and Figure 6.5 show the Cluster Accuracy (ACC) and Adjusted Rand Index (ARI) for the clustering results of the CoExDBSCAN algorithm, the GMM method, the k-means with DTW algorithm and the TICC algorithm for the synthetic data with varying noise. The standard deviation for the noise distribution $\xi \sim \mathcal{N}(0, \sigma^2)$ varies in the range $[0.01, 5.0]$ with a step size of 0.5. The lower limit is set slightly above zero to prevent numerical errors with the TICC algorithm, while increasing the standard deviation beyond 5.0 shows the same trend for all algorithms. All results are averaged over ten consecutive runs with negligible variance.

As illustrated in Figure 6.4 and Figure 6.5 the CoExDBSCAN algorithm achieves the highest accuracy on the synthetic dataset, ranging from 94.17% to 98.33% accuracy, with the highest ARI score, ranging from 0.91 to 0.96.

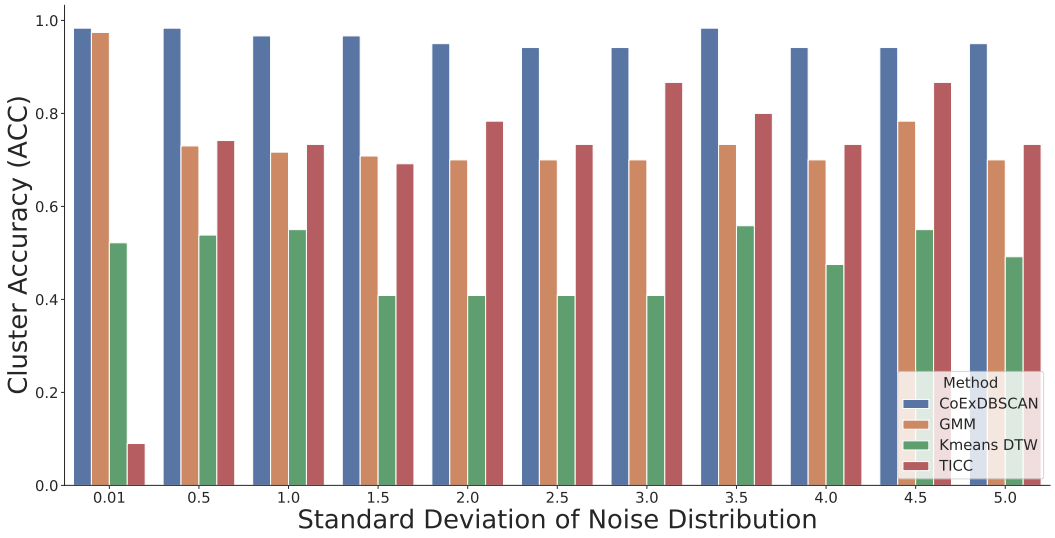


Figure 6.4: Cluster Accuracy (ACC) of the compared algorithms depending on the noise standard deviation for the synthetic dataset.

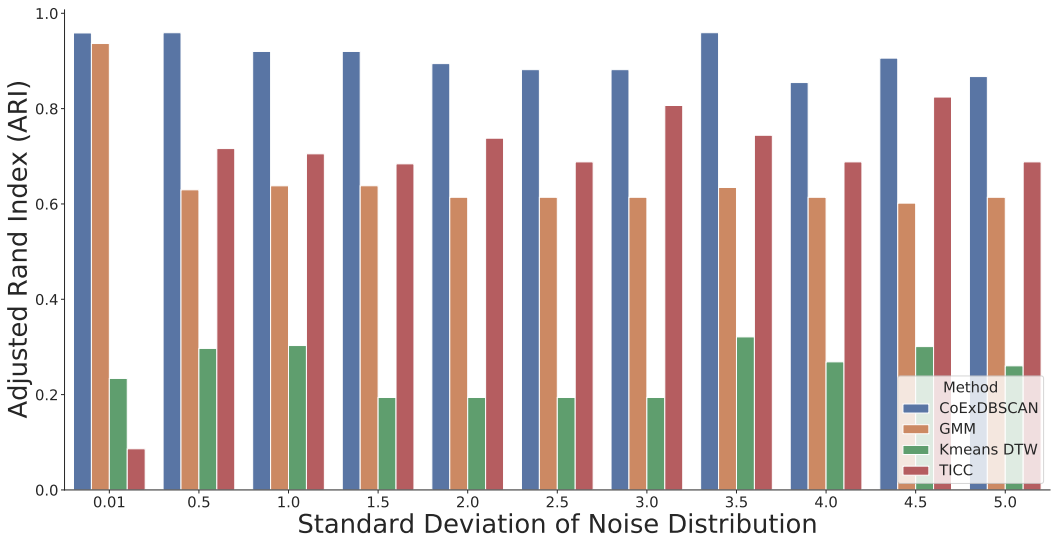


Figure 6.5: Adjusted Rand Index (ARI) of the compared algorithms depending on the noise standard deviation for the synthetic dataset.

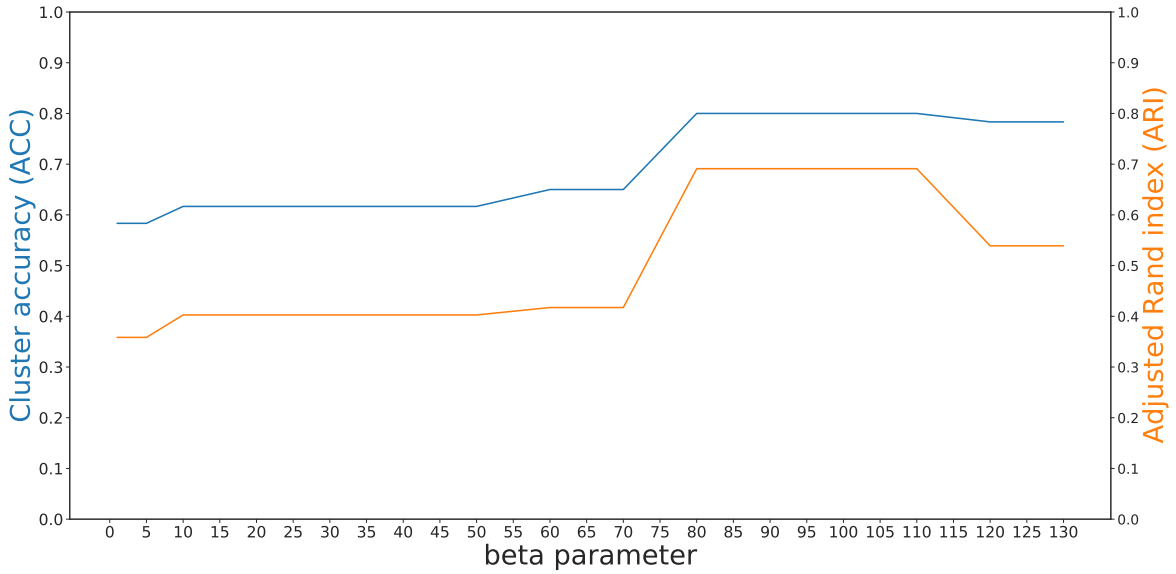


Figure 6.6: Example ACC and ARI for the TICC clustering algorithm depending on the smoothness penalty parameter β; with a noise distribution for the synthetic data $\xi \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 4.0$.

The GMM method shows a similarly high accuracy for the lowest standard deviation of the noise distribution (97.42% accuracy) but performs significantly worse for higher standard deviation values (70.00% to 78.33% accuracy). The k-means with DTW algorithm performs with a continuously low accuracy, between 40.83% to 55.83%, and with a continuously low ARI score, between 0.19 to 0.32. The TICC algorithm performs poorly for the lowest standard deviation of the noise distribution (9.00%) and close to the range of the GMM method for higher standard deviation values, 69.17% to 86.76% accuracy and 0.68 to 0.82 ARI.

The parameters for the compared algorithms have been determined from the result with the highest ACC via grid search for each run. Since k-means, GMM and TICC require the number of clusters as a parameter; this parameter has been fixed to the true number of subsequences. The number of initializations for the GMM method has been set to ten iterations with the best results to keep, and the window size for TICC that has been set to one, both parameters due to empirical evaluation. The smoothness penalty parameter β for the TICC algorithm has been evaluated in the range of $[1, 200]$ with a step size of one, whereas values greater than 130 yielded numerical errors. It should be noted that without a temporal consistency constraint, see Hallac et al. (2017), the β parameter should be kept at zero to comply with Definition 17, however the best accuracy for the TICC algorithm can be achieved with $\beta \in [80, 110]$, see Figure 6.6.

The parameters for the adapted version of the CoExDBSCAN algorithm have been evaluated for ϵ in the range $[1, 5]$ (step size one), $minPts$ in the range $[1, 5]$ (step size one) and δ in the range $[1, 10]$ (step size one) which empirically showed the best results and cover the main variability of the ARI and ACC values. More details on the empirical evaluation of the parameters are given in Appendix B.1. In comparison, the modified and adapted version of CoExDBSCAN yields the best clustering result for all generated synthetic data regardless of the noise distribution's standard deviation. This approach significantly outperforms the baseline k-means with DTW, the general clustering approach with GMM and also the state-of-the-art TICC algorithm, see Table 6.2.

Figure 6.7, Figure 6.8 and Figure 6.9 illustrate the clustering results of the CoExDBSCAN semi-supervised time point clustering method for the synthetic dataset with noise distribution $\xi \sim \mathcal{N}(0, 4^2)$ and the parameters $\epsilon = 2$, $minPts = 1$ and $\delta = 6$. The subsequences have been accurately identified and clustered together, with four data points labelled as noise ($\sim 3\%$ of all data points).

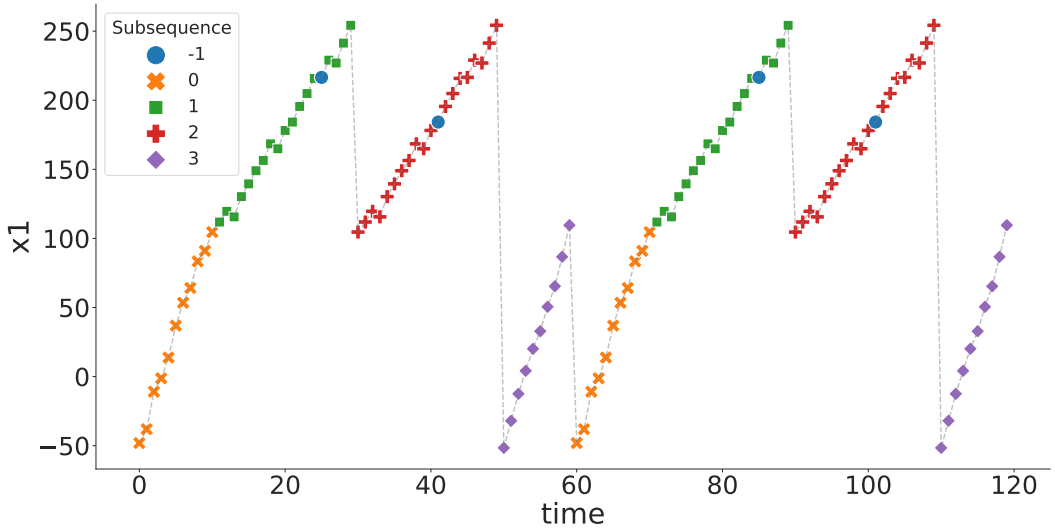


Figure 6.7: Semi-supervised time point clustering with the modified CoExDBSCAN algorithm; time-series for feature x1 with predicted labels.

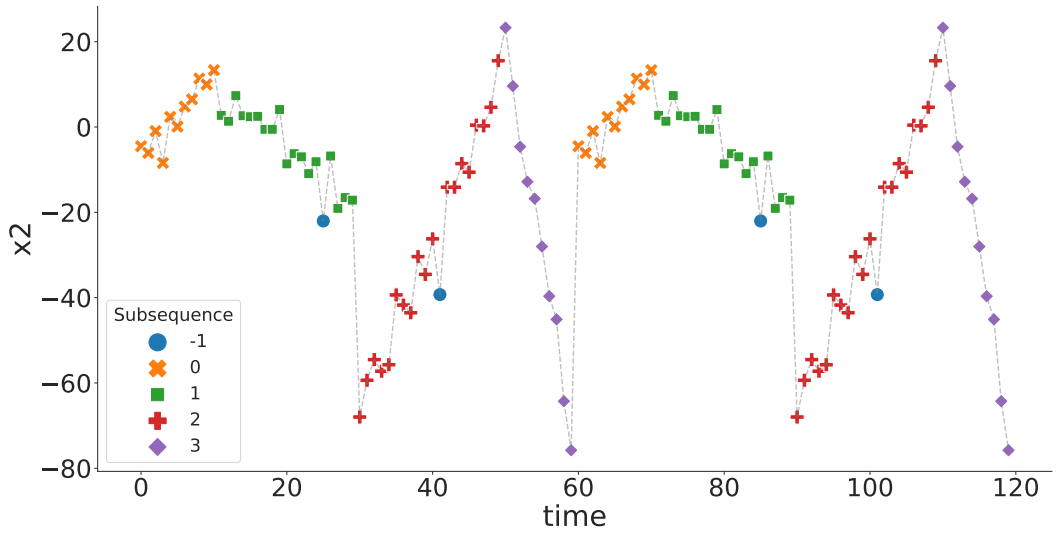


Figure 6.8: Semi-supervised time point clustering with the modified CoExDBSCAN algorithm; time-series for feature x2 with predicted labels.

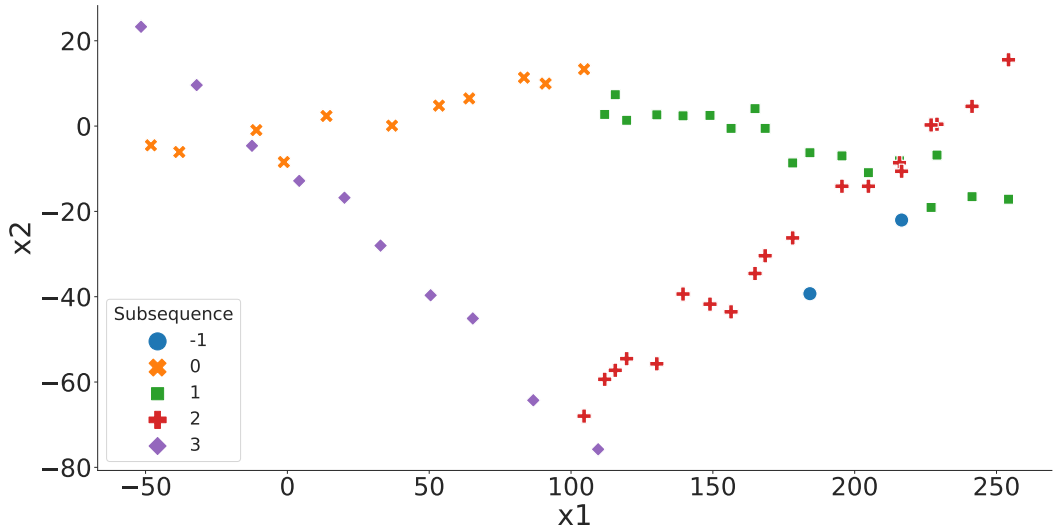


Figure 6.9: Semi-supervised time point clustering with the modified CoExDBSCAN algorithm; joint feature space {x1,x2} with predicted labels.

standard deviation	method	ACC	ARI
0.01	CoExDBSCAN	0.983333	0.958709
	GMM	0.974167	0.936560
	Kmeans DTW	0.521667	0.233889
	TICC	0.090000	0.086261
0.50	CoExDBSCAN	0.983333	0.959186
	GMM	0.730000	0.629611
	Kmeans DTW	0.538333	0.296864
	TICC	0.741667	0.716166
1.00	CoExDBSCAN	0.966667	0.919901
	GMM	0.716667	0.637987
	Kmeans DTW	0.550000	0.302922
	TICC	0.733333	0.705104
1.50	CoExDBSCAN	0.966667	0.919901
	GMM	0.708333	0.638106
	Kmeans DTW	0.408333	0.193940
	TICC	0.691667	0.684092
2.00	CoExDBSCAN	0.950000	0.894413
	GMM	0.700000	0.613985
	Kmeans DTW	0.408333	0.193940
	TICC	0.783333	0.737931
2.50	CoExDBSCAN	0.941667	0.881699
	GMM	0.700000	0.613985
	Kmeans DTW	0.408333	0.193940
	TICC	0.733333	0.688192
3.00	CoExDBSCAN	0.941667	0.881699
	GMM	0.700000	0.613985
	Kmeans DTW	0.408333	0.193940
	TICC	0.866667	0.806535
3.50	CoExDBSCAN	0.941667	0.881699
	GMM	0.700000	0.613985
	Kmeans DTW	0.491667	0.221350
	TICC	0.866667	0.806535
4.00	CoExDBSCAN	0.941667	0.854677
	GMM	0.700000	0.613985
	Kmeans DTW	0.475000	0.268655
	TICC	0.733333	0.688192
4.50	CoExDBSCAN	0.941667	0.905909
	GMM	0.783333	0.601686
	Kmeans DTW	0.550000	0.300687
	TICC	0.866667	0.824262
5.00	CoExDBSCAN	0.950000	0.867239
	GMM	0.700000	0.613985
	Kmeans DTW	0.491667	0.260299
	TICC	0.733333	0.688192

Table 6.2: Summary of clustering results for the synthetic data using the adjusted Rand index (ARI) and cluster accuracy (ACC) metrics for different noise distributions.

Figure 6.10, Figure 6.11 and Figure 6.12 illustrate the clustering result for the TICC algorithm for the same synthetic dataset. Varying the TICC input parameter β yields results with an ACC between 0.58 and 0.80 and an ARI score between 0.36 and 0.69. While the algorithm is able to distinguish between the second, third and fourth sequence, the first and second as well as fourth and first sequence cannot accurately be separated. The β parameter for the illustrated example is 80, and the true number of clusters is provided.

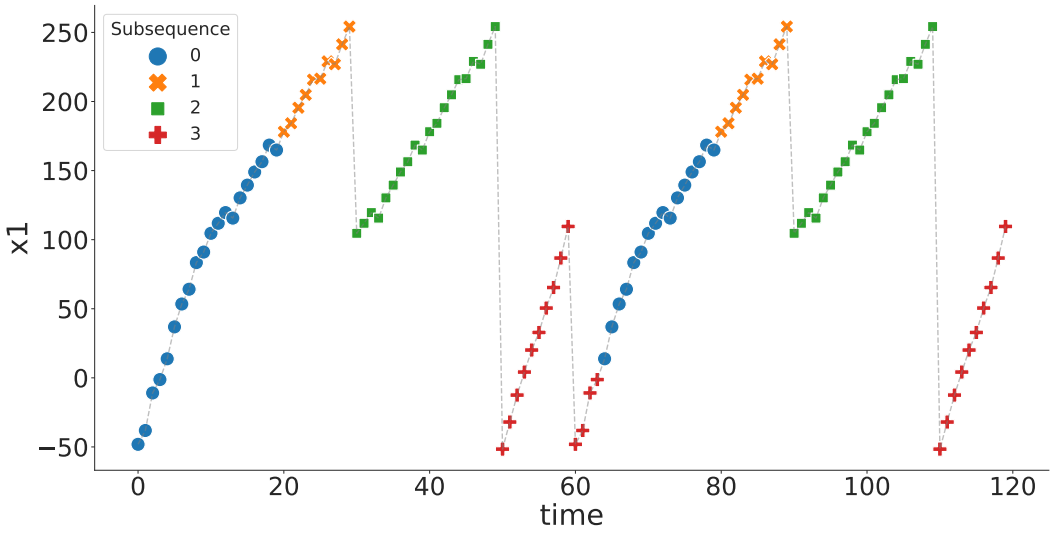


Figure 6.10: Semi-supervised time point clustering with the TICC algorithm; time-series for feature x1 with predicted labels.

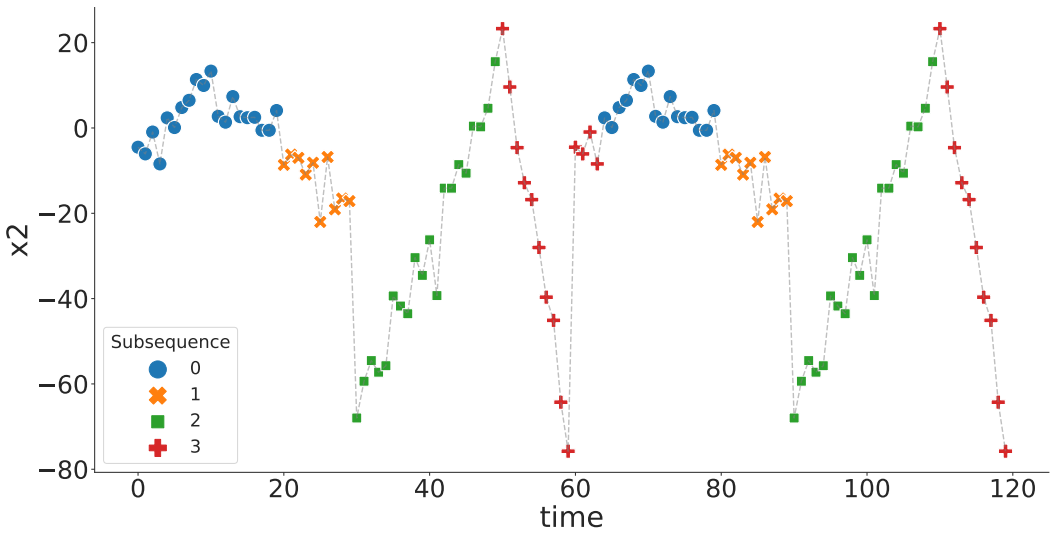


Figure 6.11: Semi-supervised time point clustering with the TICC algorithm; time-series for feature x2 with predicted labels.

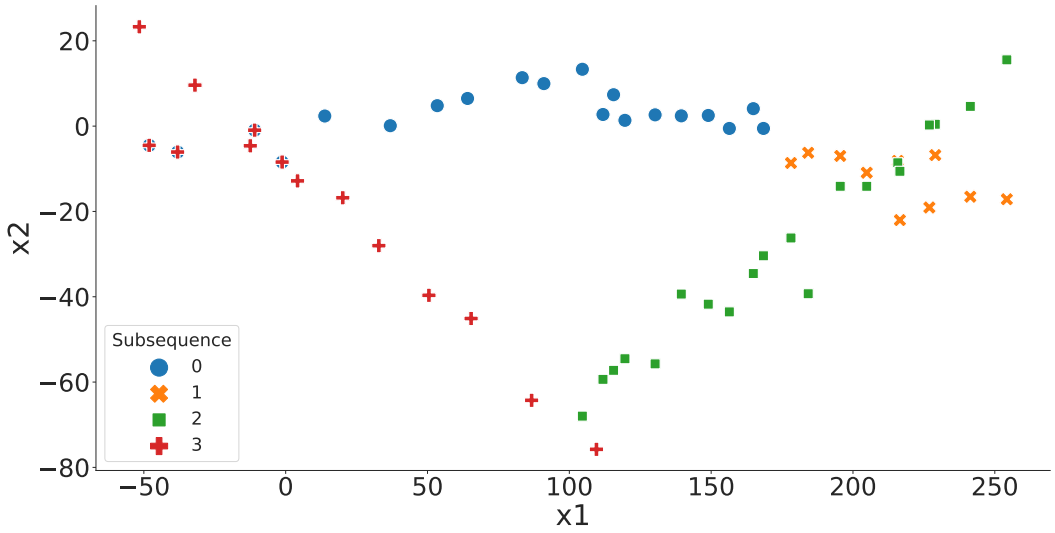


Figure 6.12: Semi-supervised time point clustering with the TICC algorithm; joint feature space {x1,x2} with predicted labels.

The third best clustering results are obtain by the Gaussian Mixture Model with all dimensions included in the clustering process, see Figure 6.13, Figure 6.14 and Figure 6.15. Similar results have been shown by Hallac et al. (2017) in their comparison for the same order of subsequences. While the third and fourth sequences can be distinguished, the first and second subsequences have been aggregated into one cluster. Moreover, the GMM makes a distinction between the first occurrence of sequences one and two (green squares) and the second occurrence of sequences one and two (orange crosses) and therefore does not correctly cluster the sequences.

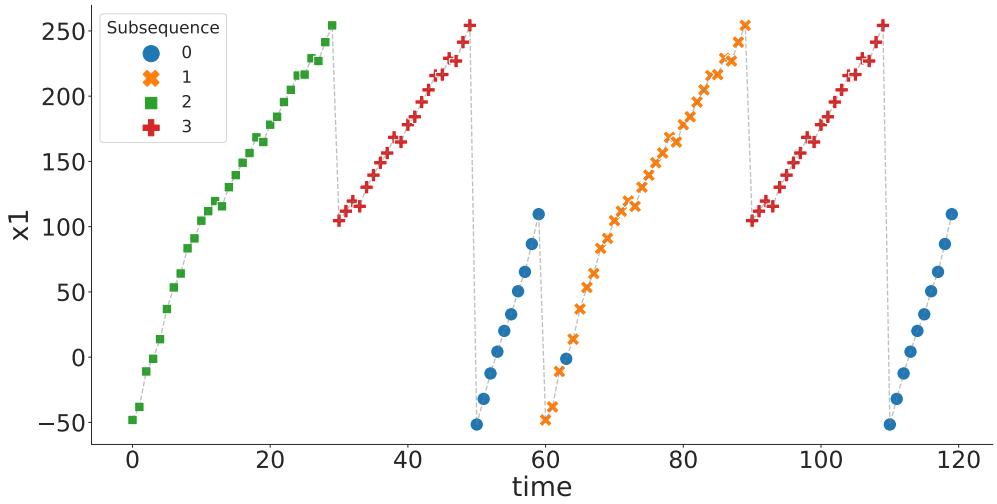


Figure 6.13: Semi-supervised time point clustering with the Gaussian mixture model (GMM); time-series for feature x1 with predicted labels.

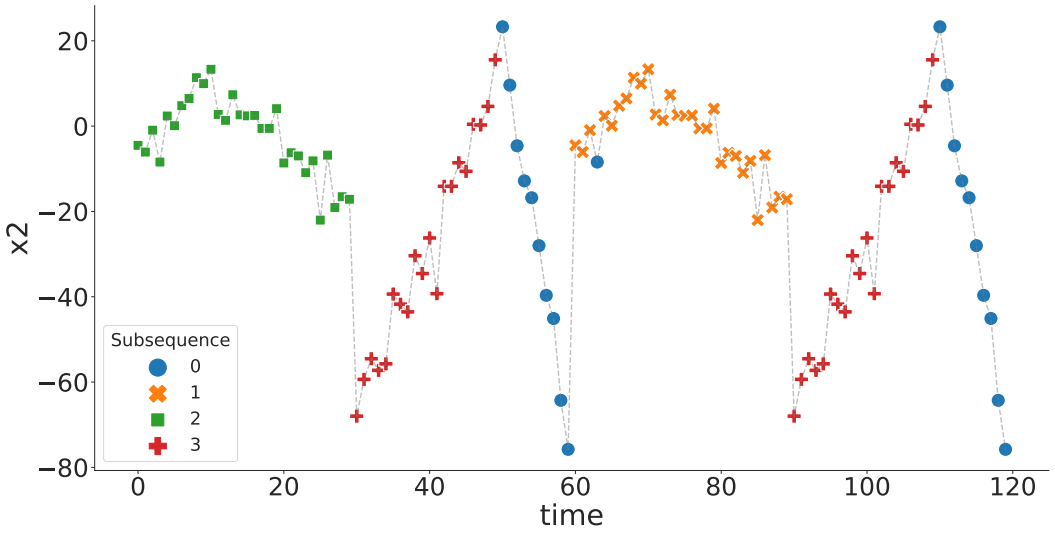


Figure 6.14: Semi-supervised time point clustering with the Gaussian mixture model (GMM); time-series for feature x2 with predicted labels.

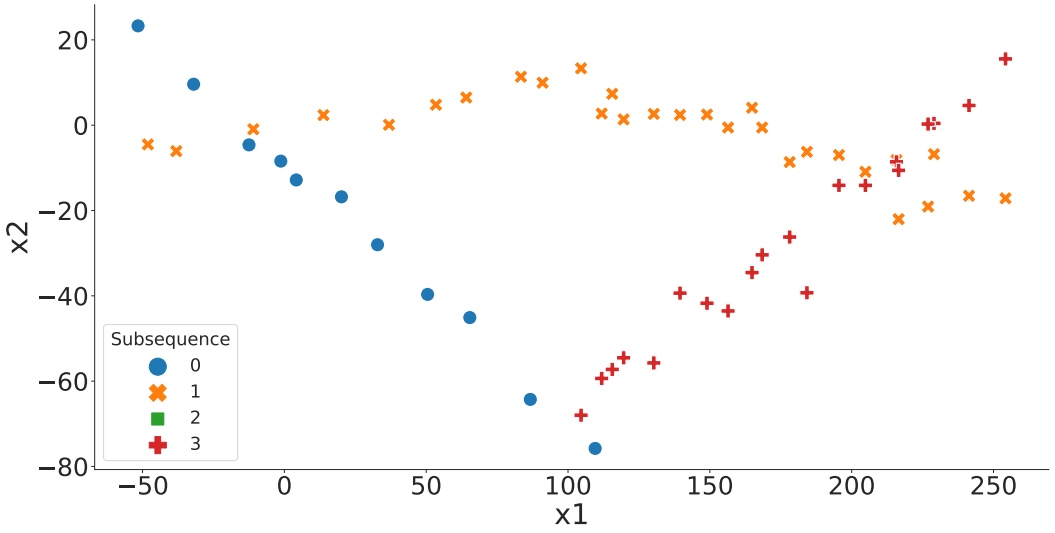


Figure 6.15: Semi-supervised time point clustering with the Gaussian mixture model (GMM); joint feature space {x1,x2} with predicted labels.

The results for the k-means with DTW algorithm are illustrated in Figure 6.16, Figure 6.17 and Figure 6.18. Each sequence is split into two parts and the respective parts clustered together across sequences, e.g. the parts of the first and fourth sequence into cluster zero and two (blue and green colour) and the parts of the second and third sequence into cluster one and three (orange and red colour).

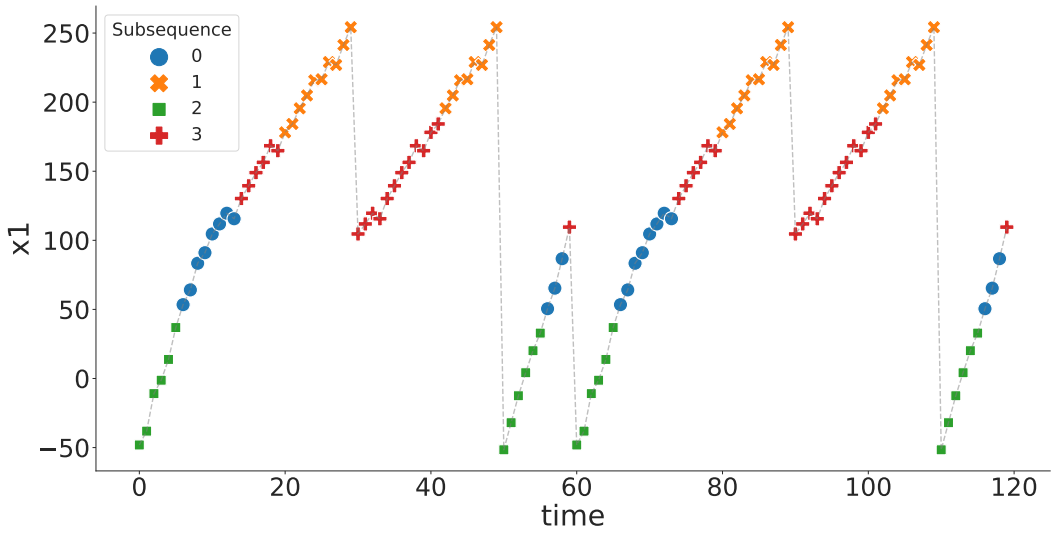


Figure 6.16: Semi-supervised time point clustering with the k-means with DTW algorithm; time-series for feature x1 with predicted labels.

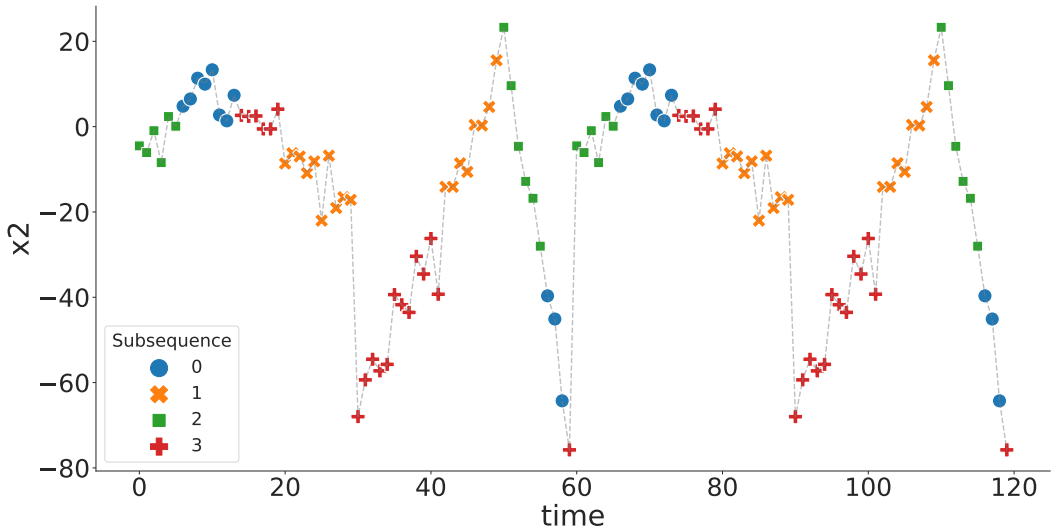


Figure 6.17: Semi-supervised time point clustering with the k-means with DTW algorithm; time-series for feature x2 with predicted labels.

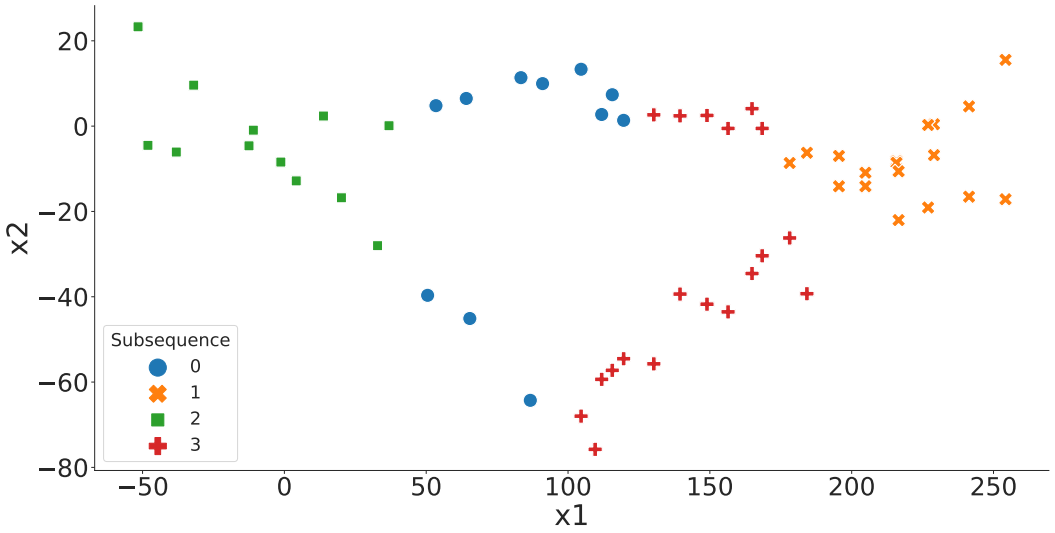


Figure 6.18: Semi-supervised time point clustering with the k-means with DTW algorithm; joint feature space {x1,x2} with predicted labels.

The modified CoExDBSCAN does not require the number of clusters as an input parameter, which is usually unknown a priori and should instead be discovered in the clustering process. The other parameters are intuitively comprehensible, with ϵ corresponding to the time window, δ corresponding to the factor of maximum deviation from the residuals mean of the linear regression and θ corresponding to the similarity threshold of the linear regression coefficients for each subsequence. Given these parameters either through empirical evaluation or expert knowledge, the approach captures the inherent structure of the data best compared to the selected algorithms, which is further demonstrated in the following for the real-world dataset.

For the evaluation of the selected algorithms on a real-world dataset, the vertical zigzag class of the LIBRAS dataset provides an intuitively accessible example. The vertical zigzag class contains 24 instances referencing a hand movement from either the upper left or upper right to the lower right or lower left, respectively, following a zigzag motion. Figure 6.19 illustrates an example of the LIBRAS dataset with nine samples from the vertical zigzag class out of 24 instances.

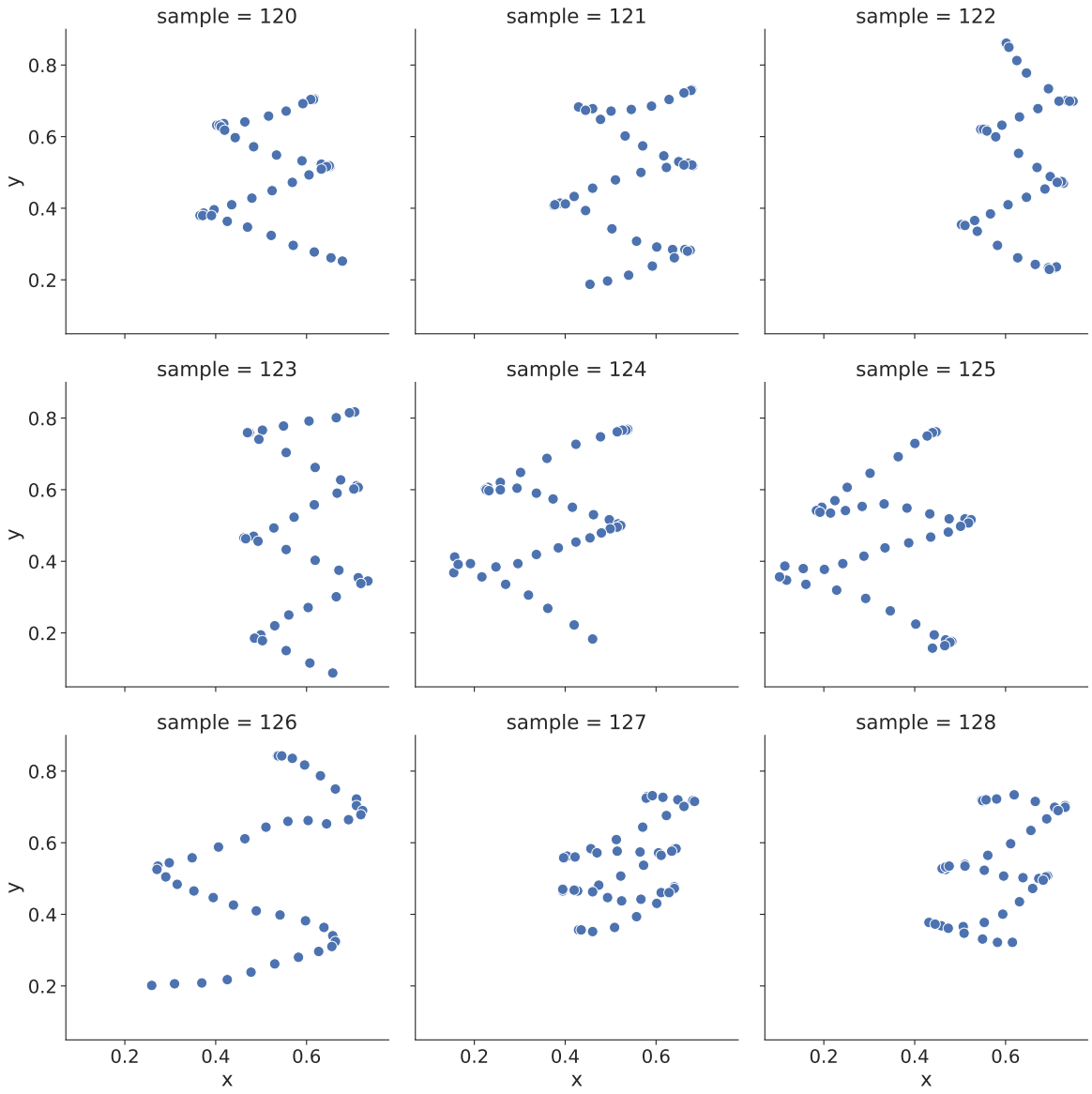
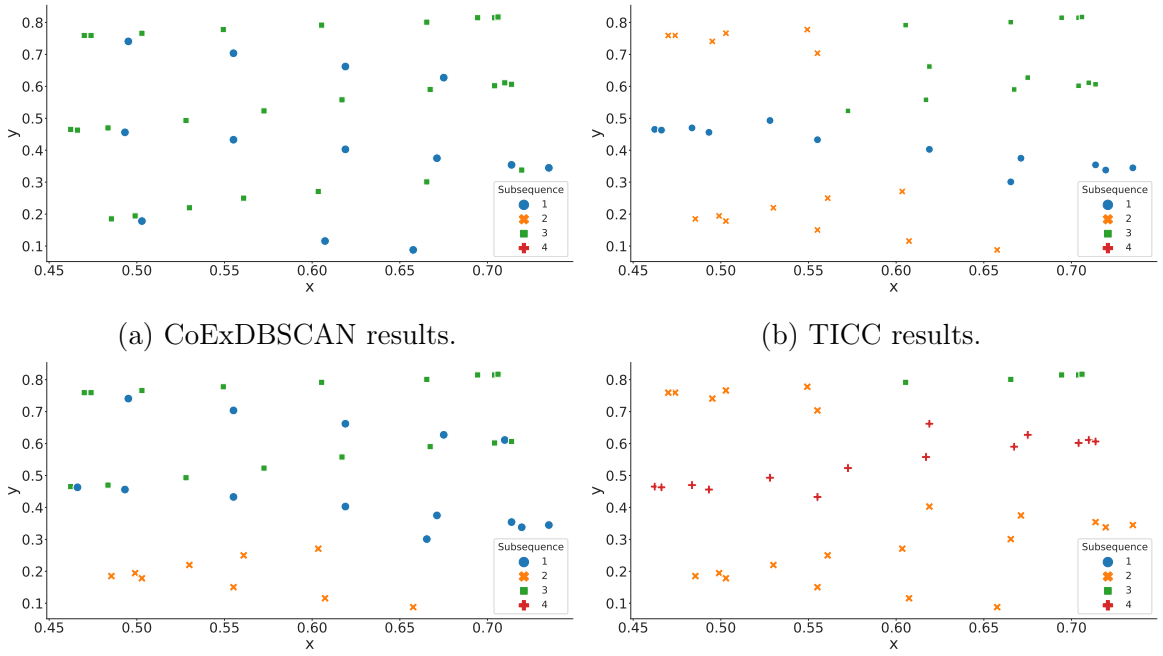


Figure 6.19: Example of the LIBRAS dataset with nine samples from one class (vertical zigzag).

Figure 6.20a to Figure 6.20d provide a visual comparison of the CoExDBSCAN, TICC, GMM and k-means with DTW algorithms on the LIBRAS dataset on an arbitrary individual sample from the vertical zigzag class, see sample number 123 in Figure 6.19. For each algorithm, the discretization of the linear regression coefficients for each cluster has been performed by grouping the coefficients into equal-sized buckets based on their quantiles. The visual comparison shows that CoExDBSCAN provides the best qualitative result. With CoExDBSCAN similar partial motions are grouped into the same category, see Figure 6.20a, while with the TICC algorithm (Figure 6.20b), the GMM method (Figure 6.20c) and the k-means with DTW algorithm (Figure 6.20d),



(a) CoExDBSCAN results.

(b) TICC results.

(c) GMM results.

(d) kmeans with DTW results.

Figure 6.20: Comparison of clustering algorithms on the LIBRAS dataset on an individual sample from the vertical zigzag class, see sample 123 from Figure 6.19; points labelled as noise are omitted.

partial motions with a visual clear change of course could not be separated. Furthermore, GMM, TICC and k-means with DTW require the number of clusters to form as input parameter, which depends on the class of motion and number of partial motions to identify. Therefore, these algorithms can merely express the inherent structure of the dataset, but the modified CoExDBSCAN method can independently discover the intrinsic properties of the data.

Figure 6.21 exemplifies the result of the experimental evaluation, depicting all of the nine sample time series illustrated in Figure 6.19, without loss of generality, after applying the modified CoExDBSCAN algorithm on each series. The discretization of the linear regression coefficients for each cluster results in three clusters that correspond to similar partial motions. The blue dot time points indicate a motion staring left in the coordinate space and ending in a lower right position of the coordinate space. The green square time points indicate a mirrored motion that starts right in the coordinate space and ends in a lower left position. The third class of subsequences, orange crosses, indicate a motion similar to the blue dot sequences but with a less steep slope, e.g. oriented to a horizontal motion.

The extracted subsequences are also identifiable across classes. Figure 6.22 shows nine arbitrary samples from the horizontal zigzag class.

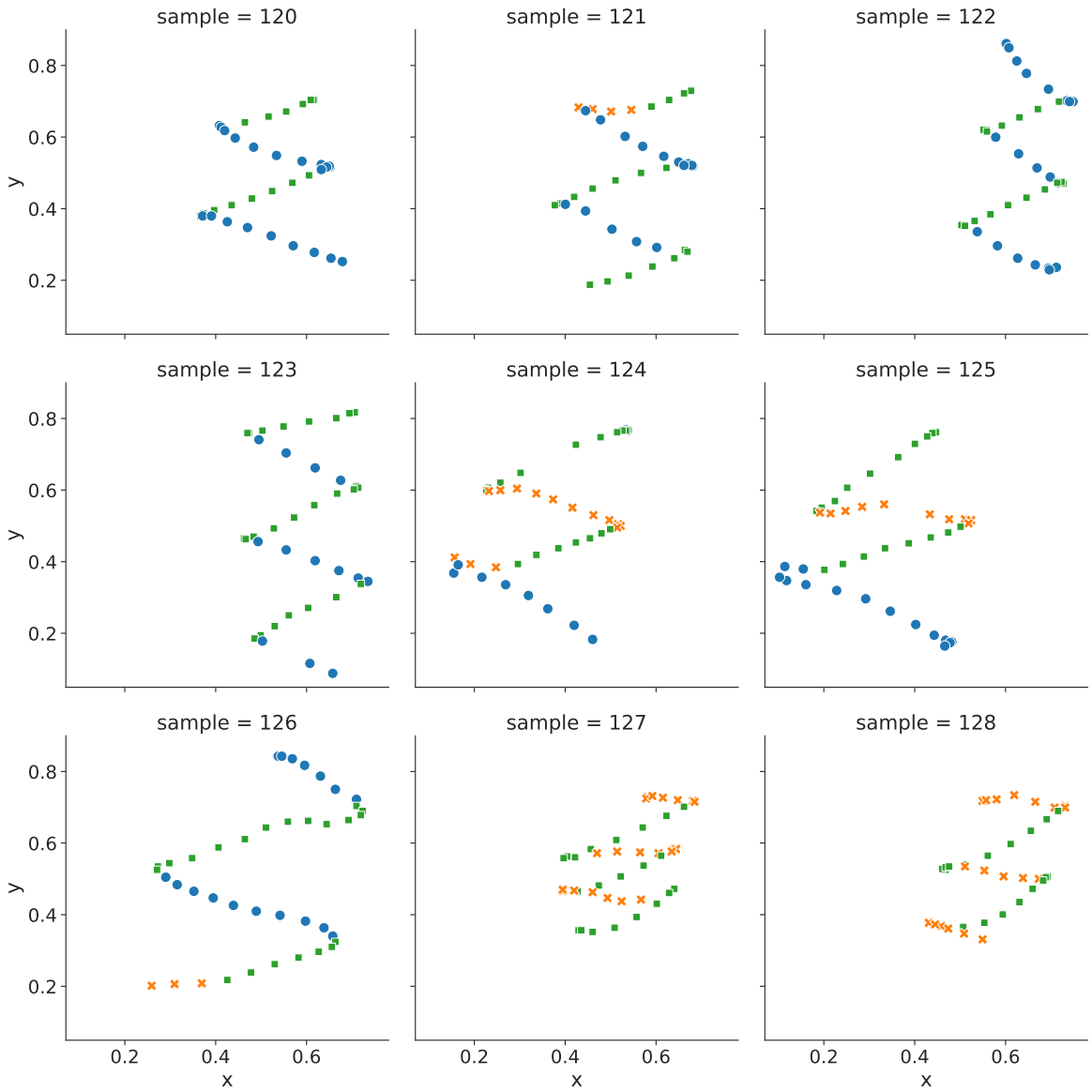


Figure 6.21: Semi-supervised time point clustering with the modified CoExDBSCAN algorithm on the LIBRAS dataset. Nine samples from the vertical zigzag class illustrate the successful segmentation of each time-series into similar motions.

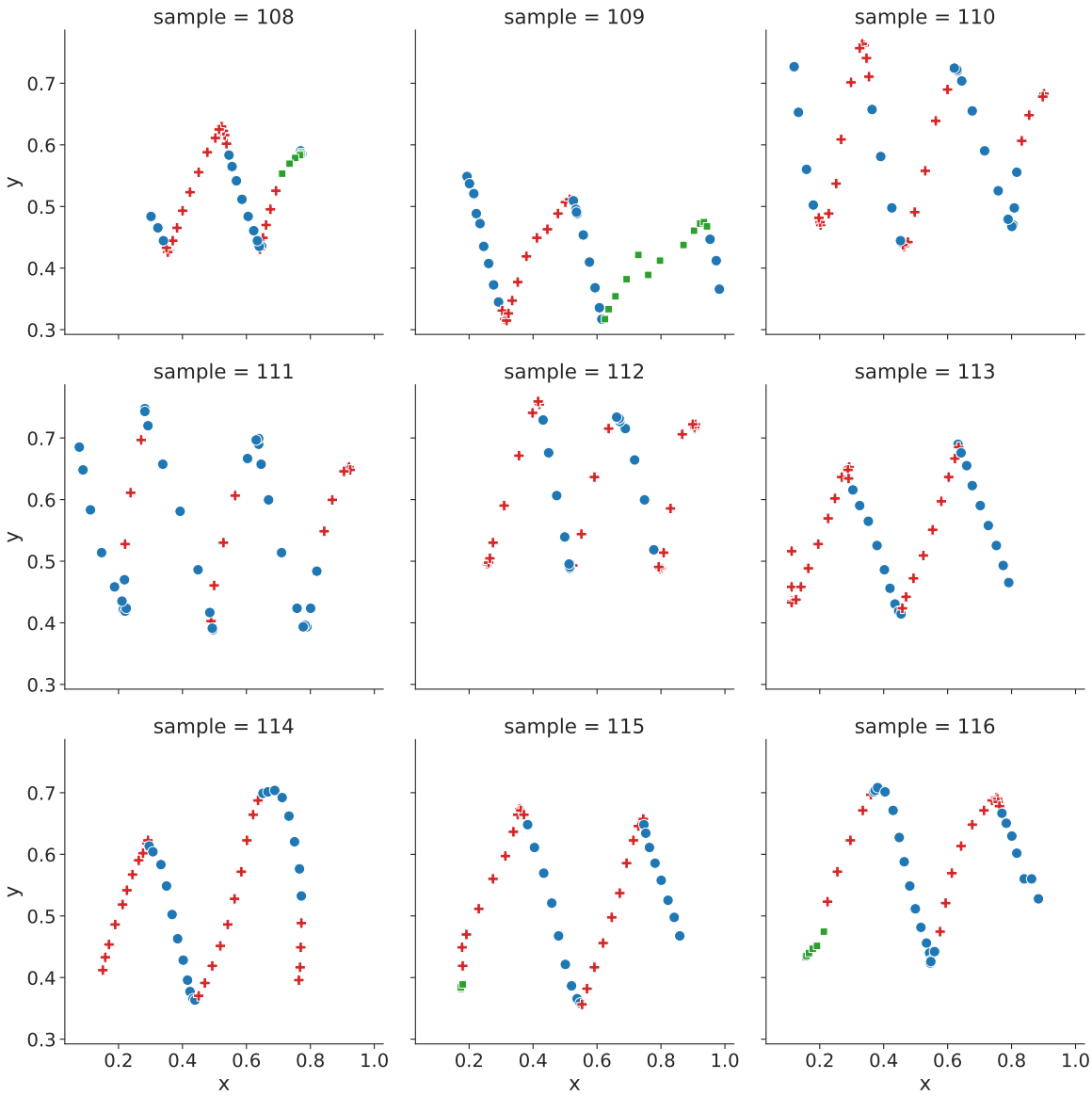


Figure 6.22: Semi-supervised time point clustering with the modified CoExDBSCAN algorithm on the LIBRAS dataset. Nine samples from the horizontal zigzag class illustrate the successful segmentation of each time-series into similar motions.

It is apparent that the blue partial motions from Figure 6.21 are also present in Figure 6.22, indicating a hand motion starting in the upper left and ending in the lower right from the starting position.

In addition to the clustering of similar subsegments, the clustering results can be utilized to serve as an effective basis for time-series classification by comparing the distributions of coefficients. For example, the Kolmogorov-Smirnov statistic (Hodges, 1958) can be computed for each learned distribution against an unseen distribution. This approach can classify time-series that exhibit similar subsequences up to an accuracy of $\sim 67\%$. Similar accuracy can be achieved by *k-nearest neighbors vote* ($\sim 79\%$) or a time-series specific *Support Vector Classifier* ($\sim 68\%$) from the tslearn toolkit without parameter optimization.

Semi-Supervised Trajectory Segmentation

Following the experimental setup described in Section 6.3.1, CoExDBSCAN can be utilized to segment trajectories similar to the approach for semi-supervised time point clustering. The experimental trajectory data consists of $\{H_2O, \delta D\}$ pair distributions that are segmented into multiple phases, which can be associated with atmospheric moisture processes. Identifying such processes is an essential scientific task to infer the dynamics of cloud-circulation systems. Investigating the atmosphere from a cloud-circulation system perspective is essential to address the significant uncertainty of climate predictions (Bony et al., 2015).

To differentiate multiple phases of individual trajectories, the constraint on the $\{H_2O, \delta D\}$ pair distribution has been conceptualised together with domain experts and the constraint parameter δ empirically determined, see Definition 18. The expansion of clusters e.g. subsequences, is constraint by including a neighbouring point in the ϵ-neighbourhood only if the residuals of an ordinary least squares linear regression for the current cluster points without the neighbouring point deviates only by a certain factor δ from the ordinary least squares linear regression including the neighbouring point, formulated in Definition 18.

To exemplify the approach and demonstrate the flexibility to segment likewise only a subset of individual trajectories, the experimental study focuses on rain events within trajectories without any loss of generality. These events have been identified by retrieving the time points where the moving average over three consecutive time points for the relative humidity are above a certain threshold and extending the event to at least three time points if necessary.

For the experimental evaluation, the Lagrangian air parcel trajectory dataset has been organised as multivariate time-series, e.g. backward trajectories for individual air

parcels, with a focus on an area of interest with the arrival of all trajectories above West Africa at pressure levels 575 and 625 Hectopascal (hPa). The trajectories are calculated daily for the local morning (9 am) and evening (9 pm) times during the period from June 8, 2016, to July 30, 2016, resulting in 12,720 individual trajectories (11,853 after filtering) with 169 time-steps each with a time delta of one hour; each trajectory comprises a time frame of 7 days.

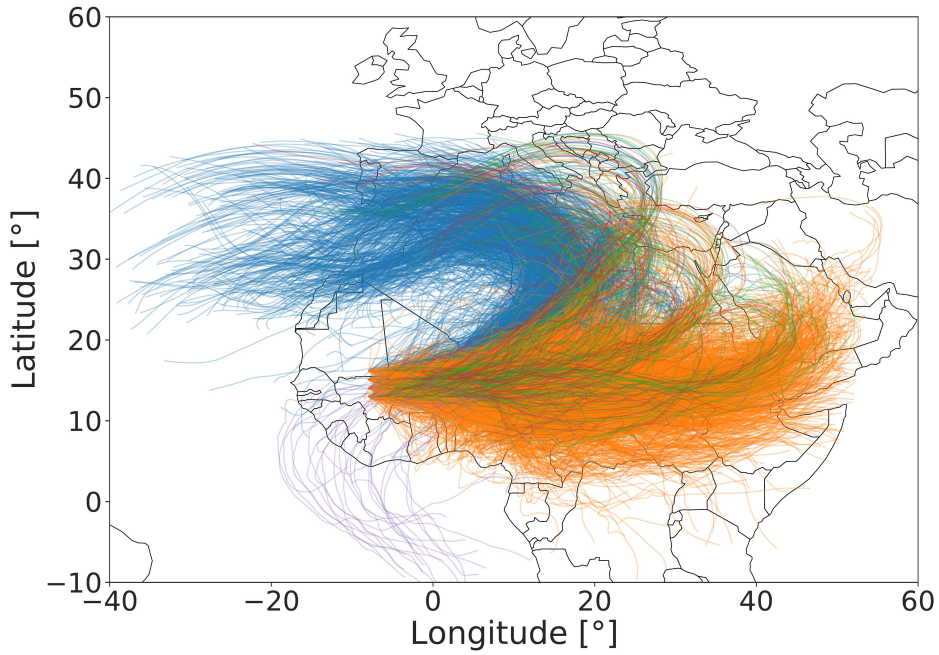
Figure 6.23a illustrates the trajectory dataset, depicting 3,194 individual trajectories out of 11,853 that have been coloured according to their similarity, using DBSCAN on a pre-computed distance matrix. For the pre-computed distance matrix, each trajectory has been converted to a $4 \cdot 169 = 676$ dimensional vector; the latitudinal (1) and longitudinal (2) difference for each time point to the arrival coordinates, the bearing (3) for each consecutive point and the scaled height difference (4) for each consecutive point, 169 points each. This initial clustering is only done to associate and compare individual segments as a result of the CoExDBSCAN segmentation, see Figure 6.23b, and is entirely independent of the trajectory segmentation approach.

Following Algorithm 4 in Section 6.3.1, that describes the semi-supervised trajectory segmentation approach applied to the experimental data, the identified trajectories with a subset of rain events are sorted by time in ascending and descending order, see Line 2 and 4 in Algorithm 4.

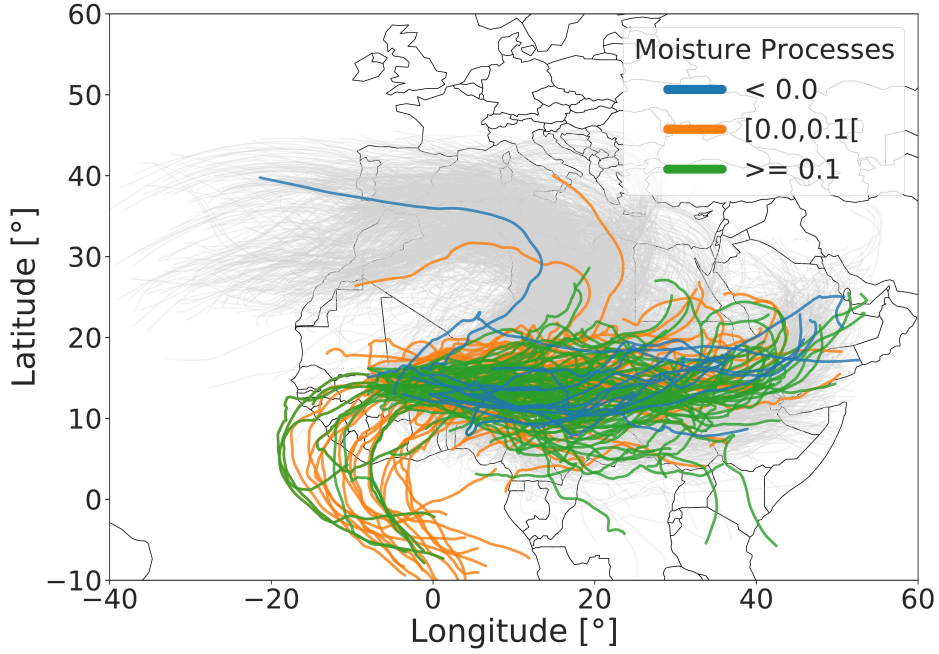
For each time ordering the labels are computed using CoExDBSCAN with the time dimension, `timePoints.time`, as the spatial subspace and the natural logarithm of the water vapour values together with the natural logarithm of the isotopologue ratio value divided by 1,000 plus one, `timePoints.{ln(H2O),ln(deltaD/1000 + 1)}`, as the constraint subspace, see Line 3 and 5. This subspace defines the correlation subspace between $\ln(H_2O)$ and $(\ln(HDO) - \ln(H_2O))$, which is a proxy for δD as shown in the following equations.

$$\delta D = [\frac{HDO}{H_2O}/R - 1] * 1000 \tag{6.2}$$

$$\ln(\frac{\delta D}{1000} + 1) = \ln(\frac{HDO}{R}/H_2O) \tag{6.3}$$



(a) Example DBSCAN clustering result.



(b) Example moisture processes.

Figure 6.23: (a) Illustration of 3,194 trajectories out of 11,853 that have been coloured according to their geographical closeness, bearing similarity and height difference along each individual trajectory. (b) Illustration of the association of individual trajectories to different moisture processes as a result of the CoExDBSCAN segmentation.

Equation 6.3 can be approximated for standardized HDO by multiplying the above equation with $\frac{1}{R}$ with $R = 3.1152 \cdot 10^{-4}$, the Standard Mean Ocean Water (SMOW) constant, which results in the δD proxy.

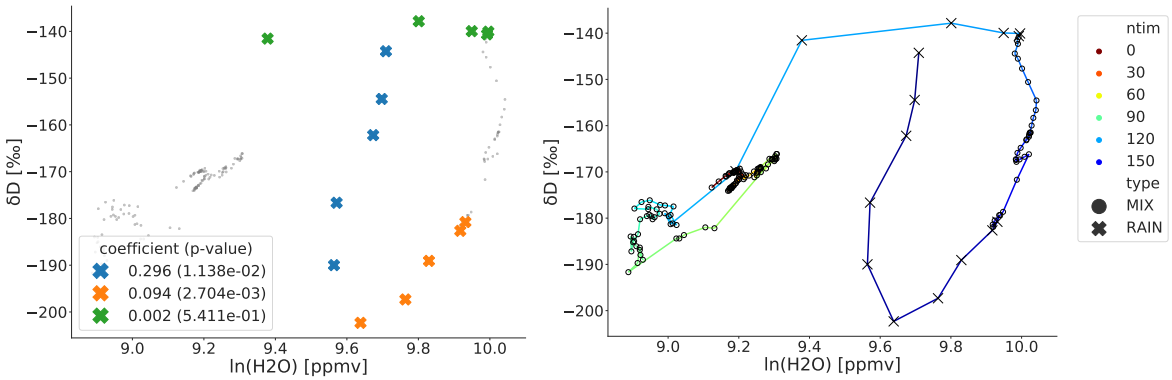
$$\ln(\frac{HDO}{H_2O}) = (\ln(HDO) - \ln(H_2O)) \tag{6.4}$$

The remaining parameters have been empirically determined and proposed by domain experts and set to $\epsilon = 2$, $minPts = 3$ and $\delta = 2$ in this example. Computing the segmentation in both temporal orders is necessary because the outcome of the linear regression of the constraint depends on the deviation of the residuals from the current cluster points, which can be different following the trajectory points in ascending or descending temporal order. The final segmentation of the trajectory, or the subset of time points within a trajectory, is the selection of phases from the ascending and descending CoExDBSCAN run where the outcome with the squared residual sum is lowest, Line 6 to 11 in the algorithm.

Figure 6.24 shows the segmentation result of two example trajectories. Figure 6.24a and Figure 6.24c depict the results of the semi-supervised segmentation approach with dots indicating non-rain events (air mass mixing) and crosses indicating rain events, which have been segmented into different coloured phases with different regression coefficients. Figure 6.24b and Figure 6.24d visualize the temporal order of the corresponding trajectories' events coloured according to the number of hours before arrival from 168 to 0 (dark blue to dark red). The first example trajectory in Figure 6.24a has two rain events with three distinct phases, where the first event from hour 168 to hour 159 before arrival (see Figure 6.24b) has been segmented into two different phases with different regression coefficients with statistical significance (p-value < 0.05, blue crosses and orange crosses); the second event starts at hour 124 and ends at hour 119 before arrival with a steady regression coefficient, however not statistically significant (green crosses). The second example trajectory in Figure 6.24c shows again two rain events, the first from hour 140 to hour 133 before arrival and the second from hour 38 to hour 32 before arrival.

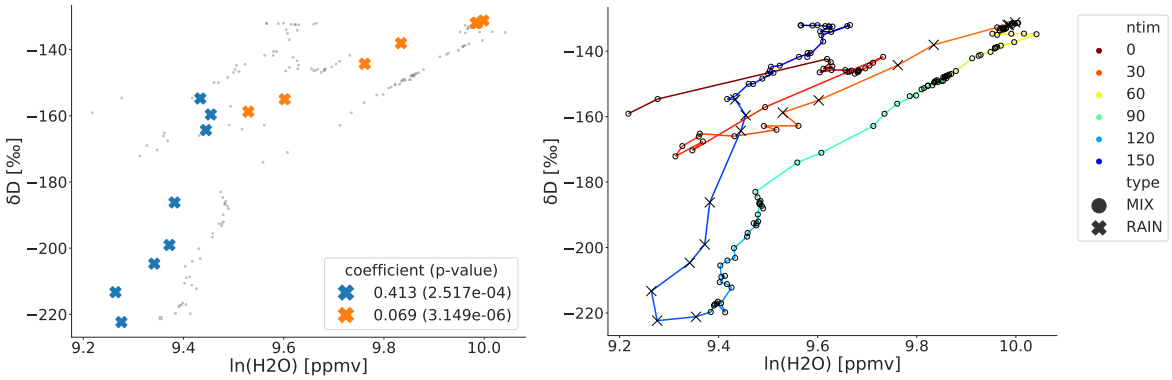
The regression coefficients in Figure 6.24c are unvarying with statistical significance (blue crosses and orange crosses). The different slopes of the linear regression lines fitted to the two different rain events are clearly observable.

Comparing the rain segments from both example trajectories with the theoretical evolution of δD as a function of H_2O, see Section 2.2, the segmented event in Figure 6.24a (blue and orange crosses) and the two timely separated phases in Figure 6.24c (blue and orange crosses) can be interpreted as different kind of rain processes.



(a) Three phases example trajectory.

(b) Timely order of events for (a).



(c) Two phases example trajectory.

(d) Timely order of events for (c).

Figure 6.24: Example trajectories with two rain sequences, which can be identified by the crosses; (a) with additional segmentation (c) without additional segmentation (blue and orange crosses in (c) correspond to temporally separated rain events). (b) and (d) illustrate the timely order of events according to the number of hours before arrival from 168 to 0 (dark blue to dark red).

For example, the first phase in Figure 6.24c (blue crosses) is in line with super-Rayleigh processes, i.e. there is some interchange between condensed moisture and vapour. The second phase (orange crosses) is close to a Rayleigh process, i.e. can be explained by condensation and direct rainout, without significant interchange between condensed moisture and vapour.

These two example trajectories demonstrate that this semi-supervised approach for trajectory segmentation can **(1) identify timely separated events** as well as **(2) split timely connected events** with varying regression coefficients. The latter gives novel opportunities for identifying details of the related rain processes. Distinguishing these fine-grained structures in large volumes of data emphasises the benefit of applying this method. Analysing the distributions of regression coefficients for all segments allows identifying different moisture processes in the atmosphere.

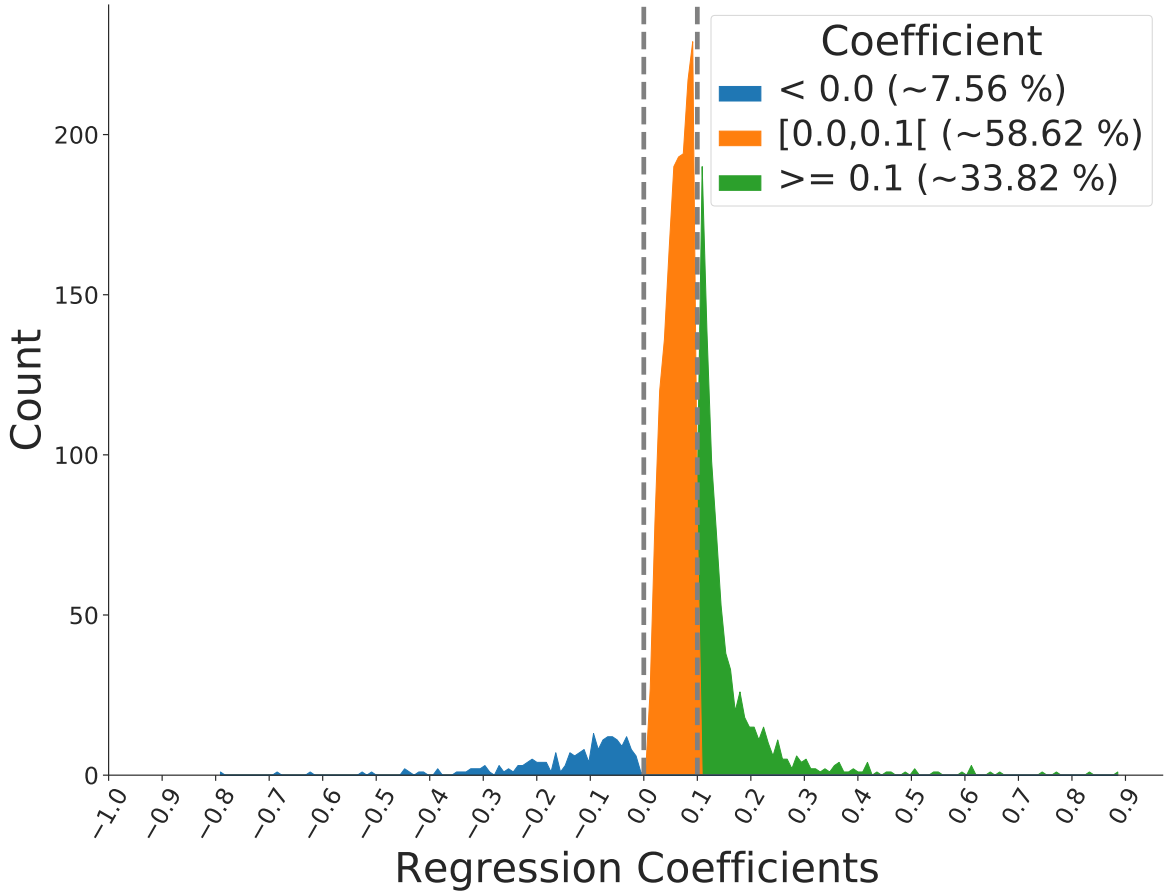


Figure 6.25: Histogram of regression coefficients (slope of linear regression line) for all rain segments as a result of the semi-supervised trajectory segmentation with statistical significance (p-value < 0.05).

The histogram in Figure 6.25 outlines the distribution of regression coefficients with statistical significance (p-value < 0.05) for all rain segments as a result of the semi-supervised trajectory segmentation, analysing all 11,853 trajectories, clipped to the interval $[-1.0, 1.0]$ (omitting four segments < -1.0 and three segments > 1.0 overall). Of all rain events, 7.6% have a negative slope, 58.6% have a slope between 0.0 and 0.1, and 33.8% have a slope greater than 0.1.

Figure 6.26 incorporates these distributions of regression coefficients with the distribution of the $\{H_2O, \delta D\}$ pairs modelled in the area of interest and the time of arrival of the trajectories at atmospheric pressure levels between 575 - 625 hPa, i.e. about 3.5 - 4.5 km a.s.l that is the altitude for which the MUSICA IASI $\{H_2O, \delta D\}$ pair data generally have the best sensitivity. The grey dots show all model data, i.e. represent the same data points presented in Section 2.2 (grey dots in Figure 2.3), but modelled instead of measured data. The lines are 50% contour lines.

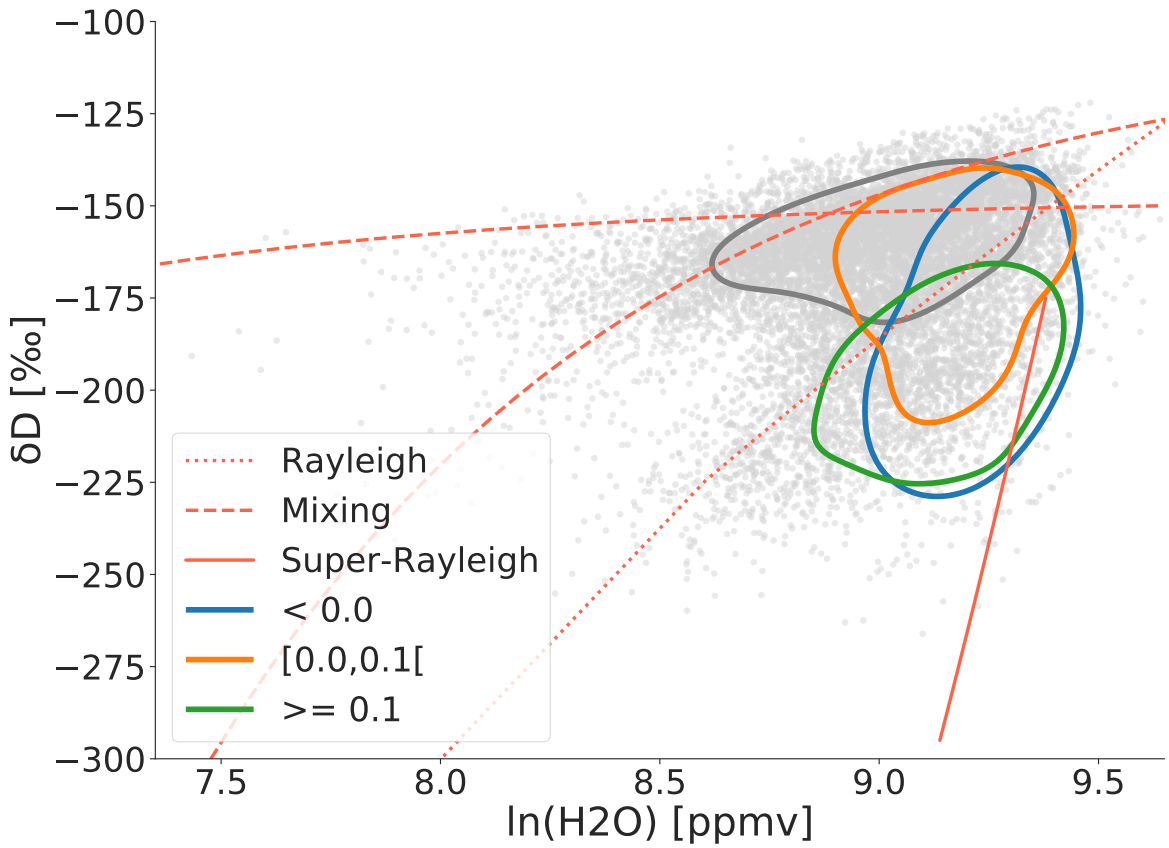


Figure 6.26: $\{H_2O, \delta D\}$ distributions of the model data in the area of interest for all data points (grey dots and contour line) and for data points representing air masses that experienced rain events (blue, orange and green colours are as in Figure 6.25 for rain events being characterised by different regression coefficients); contour levels are at 50%.

The grey contour line is for the whole dataset, the blue, orange, and green contour lines are for data points representing air masses that experienced rain events during the last five days before arrival. The colours are according to the coefficients that characterise the rain events (i.e. they are in line with Figure 6.25). If an air mass experienced rain events with two different characteristics, the respective data point belongs to both groups. The orange contour line is for air masses that experienced a rain process with a slope for $\ln(H_2O)$ versus $(\ln(HDO) - \ln(H_2O))$ as a proxy for δD between 0.0 and 0.1. This slope is typical for a Rayleigh process and the data points at arrival for atmospheric pressure levels between 575 - 625 hPa group around the theoretical Rayleigh line (red dotted line). The green contour line outlines air masses with steeper slopes than what can be expected from a Rayleigh process, i.e. slopes greater or equal than 0.1. Such large slopes indicate a Super-Rayleigh process, an interchange between condensed water (cloud droplets or rain) and water vapour. Such

interchange is typical for convective processes, and at the arrival point, the respective $\{H_2O, \delta D\}$ pair generally lies below the theoretical Rayleigh line and close to the illustrated example Super-Rayleigh line (solid red line). The blue contour line indicates air masses with negative slopes. Such slopes can be explained by the mixing of two air masses, where one air mass has a $\{H_2O, \delta D\}$ pair distribution above the Rayleigh line and the other air mass has a $\{H_2O, \delta D\}$ pair distribution below the Rayleigh line. Respective air might also be related to convective processes. The grey contour line represents the entire dataset, and it is situated above the Rayleigh line. This contour line is dominated by non-rain events, indicating that such events are dominated by mixing processes (red dashed lines). In summary, this study with simulated data shows that the $\{H_2O, \delta D\}$ distribution at the arrival point gives insight into the moisture transport process that a respective air mass has experienced. This leads to the conclusion that the $\{H_2O, \delta D\}$ distribution as observed in the MUSICA IASI dataset can be utilised as a good proxy for characterising the moisture processes that has been experienced by different air mass as well.

6.4 Summary

This chapter proposes two adaptations of the CoExDBSCAN algorithm introduced in Chapter 5, a new approach for semi-supervised time point clustering for multivariate time-series and a novel semi-supervised approach for trajectory segmentation.

The approach for semi-supervised time point clustering adopts and modifies the CoExDBSCAN algorithm and applies the algorithm to time-series data to find temporal neighbourhoods, whose expansions are restricted to a priori constraints. In Section 6.2 a constraint formulation has been proposed to restrict the cluster expansion to the correlation of time point values. This constraint is defined by the deviation of time point residuals from the mean of residuals of time points within a subsequence, based on linear regression. Forming groups of segments with similar correlations results in clusters of subsequences, which can be interpreted as recurring events or actions; for example, similar movements as shown in the experimental evaluation in Section 6.3.2. The verification on the synthetic dataset demonstrates that the proposed method based on CoExDBSCAN significantly outperforms the baseline k-means with DTW approach, the general clustering approach with GMM and also the state-of-the-art TICC algorithm. The validation and application to the real-world LIBRAS movement dataset demonstrates that this approach can accurately identify subsequences of similar motions and is able to extract distributions of coefficients for each subsequence towards an effective classification approach.

The approach for semi-supervised trajectory segmentation demonstrates the application of the CoExDBSCAN algorithm to identify different processes in the atmosphere. The CoExDBSCAN algorithm can be adopted for trajectory segmentation by formulating a constraint on the deviation from the ordinary least squares regression residuals of combined or split segments. By further extracting information about the regression coefficients for each segment and comparing the distribution of coefficients to theoretical values, it is possible to identify corresponding atmospheric processes. This approach is demonstrated in the experimental evaluation for Lagrangian air parcel trajectories together with data from the regional isotope-enabled atmospheric model COSMO-iso (Pfahl et al., 2012). The experimental evaluation in Section 6.3.2 demonstrates that the approach can successfully segment time-series, e.g. trajectories or subsequences, into sequences that contain close temporal points which follow a priori constraints. With the constraint formulated in Definition 18 each segment is differentiated by the deviation from the ordinary least squares regression residuals, which in effect splits sequences if their individual linear regression is a better fit than their combined linear regression. This constraint is particularly useful for the research objective to analyse $\{H_2O, \delta D\}$ pair distributions that follow theoretical linear relations. However, in general, this constraint restricts the cluster expansion to the correlation of time point values and can be transferred to different domains and datasets. For example, motion data in 2D or 3D space can be segmented with the same approach and constraint formulation, identifying recurring, similar motions that follow similar linear correlations.

Segmenting the Lagrangian air parcel trajectories enables researchers to better understand the complex dynamics that cause the $\{H_2O, \delta D\}$ pair distribution observable in the MUSICA IASI dataset introduced in Section 2.2. Since the proposed approach is able to distinguish fine-grained structures in large volumes of data, it is an effective data-driven analysis method for this purpose. For instance, with the grouping of trajectories, researchers can draw conclusions on atmospheric moisture transport patterns. Ongoing research in the area of observational atmospheric data orientated towards the combination of different sensors might soon allow the global detection of the vertical distribution of $\{H_2O, \delta D\}$ pairs. A method for such a synergistic combination has recently been demonstrated using CH_4 as an example by Schneider et al. (2021a). By using state-of-the-art reanalysis datasets for calculating the trajectories, the presented approach is expected to be able to directly identify different moisture processes in the measured $\{H_2O, \delta D\}$ fields.

Chapter 7

Conclusion and Outlook

This chapter summarizes the main contributions of this thesis and gives possible future research directions.

7.1 Conclusion

The research in this thesis evolves around the main challenge to better understand the natural structure of spatio-temporal data that is reflected by meaningful clusters. The conducted cluster analysis of spatio-temporal data by applying existing clustering algorithms shows that the proposed data-driven approach allows to learn the intrinsic structure of the data and to analyse the evolution of geo-referenced distributions over time. However, the presented approach does not necessarily describe the underlying process that generated the data accurately. Therefore the interpretation of the results additionally relies on domain knowledge associating cluster and cluster changes to processes.

This conclusion leads up to the main contribution of this thesis, the development of a novel semi-supervised clustering algorithm for spatio-temporal data, CoExDBSCAN, a density-based clustering algorithm with constrained expansion. The CoExDBSCAN algorithm is based on the DBSCAN algorithm to find density-connected clusters in a defined subspace of the data and restricts the expansion of clusters to a priori constraints. A priori knowledge in the form of constraints for the clustering algorithm can significantly improve the clustering results and align the outcome of the clustering process with the objective of the data analysis. This assertion is demonstrated for spatio-temporal data in general and for climatological data in particular.

CoExDBSCAN significantly outperforms the original DBSCAN algorithm and the algorithms that CoExDBSCAN has been compared to, i.e. the CASH algorithm as a state-of-the-art comparison, the constrained k-means algorithm (CK-means) and the

pairwise constrained k-means algorithm (PCK-means), and is better suited to identify correlated structures that are geographically close than the compared algorithms. The real-world evaluation demonstrates that CoExDBSCAN can also capture the underlying data generating processes more accurately than the proposed data-driven approach.

The introduction of the novel semi-supervised clustering algorithm CoExDBSCAN enables a novel method for semi-supervised time point clustering of multivariate time-series and a novel semi-supervised approach for trajectory segmentation. Both methods adopt the CoExDBSCAN algorithm to find temporal neighbourhoods in multivariate time-series, i.e. trajectories represented as multivariate time-series, and restrict the expansion of the clusters to the correlation of time point values. Forming groups of segments with similar correlations results in clusters of subsequences, i.e. trajectory segments, which can be interpreted as recurring events or actions.

In the case of semi-supervised time point clustering, CoExDBSCAN significantly outperforms the baseline k-means with DTW approach, the general clustering approach with GMM and also the state-of-the-art TICC algorithm on synthetic data and the LIBRAS movement dataset. The validation and application to the real-world LIBRAS movement dataset demonstrate that the proposed approach can accurately identify subsequences of similar motions. The approach further allows to extract distributions of coefficients for each subsequence that can serve as an effective basis for classification, for example, by computing the Kolmogorov-Smirnov statistic for each learned distribution of coefficients against an unseen distribution of coefficients, which yields a similar accuracy compared to methods like k-nearest neighbours vote or a time-series specific Support Vector Classifier without parameter optimization.

In the case of the semi-supervised segmentation of Lagrangian air parcel trajectories, CoExDBSCAN is able to identify different processes in the atmosphere by extracting information about the regression coefficients for each segment and comparing the distribution of coefficients to theoretical values. This approach allows to unveil fine-grained structures in large volumes of data and to extract segments from subsequences of continuous temporal events successfully. These segments can further be compared to the theoretical evolution of the dependent variable δD as a function of H_2O.

The presented segmentation of the Lagrangian air parcel trajectories helps better understand the complex dynamics that cause specific $\{H_2O, \delta D\}$ pair distributions that the Metop-A, B and C satellites observe and are available in the MUSICA IASI dataset. Specifically, distributions that are discovered by the segmentation approach can be compared to $\{H_2O, \delta D\}$ distributions in the observational data to conclude the origin and formation of different water transport pathways in the atmosphere.

7.2 Outlook

The presented research in this thesis provides several starting points for future research.

The presented approach for analysing the evolution of geo-referenced distributions over time allows investigating the application of **different Bayesian models** in exchange for the Gaussian mixture model for future work. Different Bayesian models could incorporate more a priori domain-specific knowledge into the clustering process and better catch the underlying processes that generated the data.

One of the uppermost tasks for future research should address the challenge of **finding and expressing suitable constraints** for the CoExDBSCAN algorithm. While generic constraints can avoid overfitting the clustering algorithm, i.e. avoid constraining the cluster expansion to the generating process, specially tailored constraints can lead to a loss of generality. Possible solutions to simplify the process of defining constraints could rely on methods from the field of active learning or apply machine-learning-based methods for the selection processes.

Another task for future work would be to **integrate** the **CoExDBSCAN** algorithm with standard data mining and data analysis tool-kits/libraries, e.g. the Python scikit-learn library or the ELKI data mining software.

With regards to the presented semi-supervised time point clustering approach, the **classification aspect** of the proposed method by extracting distributions of coefficients for subsequences shows an interesting approach that requires more detailed comparison studies with other algorithms in the field of subsequence and time point classification for multivariate time-series.

Considering the case of the semi-supervised segmentation of Lagrangian air parcel trajectories, future work could improve the **extraction of specific** $\{H_2O, \delta D\}$ **sequences** in the model domain and use the model data of rain, cloud or others to relate specific $\{H_2O, \delta D\}$ distributions to distinct moist processes clearly. One application for this improved segmentation can demonstrate that the proposed method can be used to **distinguish between air masses that experienced shallow convection or deep convection**. The long-term perspective for this approach would be to generate multisensor $\{H_2O, \delta D\}$ observation data, i.e. to create observation data that offer $\{H_2O, \delta D\}$ pair information at different altitudes. Then use trajectories calculated from state-of-the-art reanalysis fields and apply the method directly to the observational data. Data-driven analysis of observation data could **significantly improve the insight into the dynamics of such cloud-circulation systems** and would help to improve the significant uncertainty of climate predictions.

Appendix A

Appendix

A.1 CoExDBSCAN Parameter Evaluation

This appendix details further the parameter evaluation of the CoExDBSCAN and DBSCAN algorithm presented in Section 5.3.2 for the synthetic dataset.

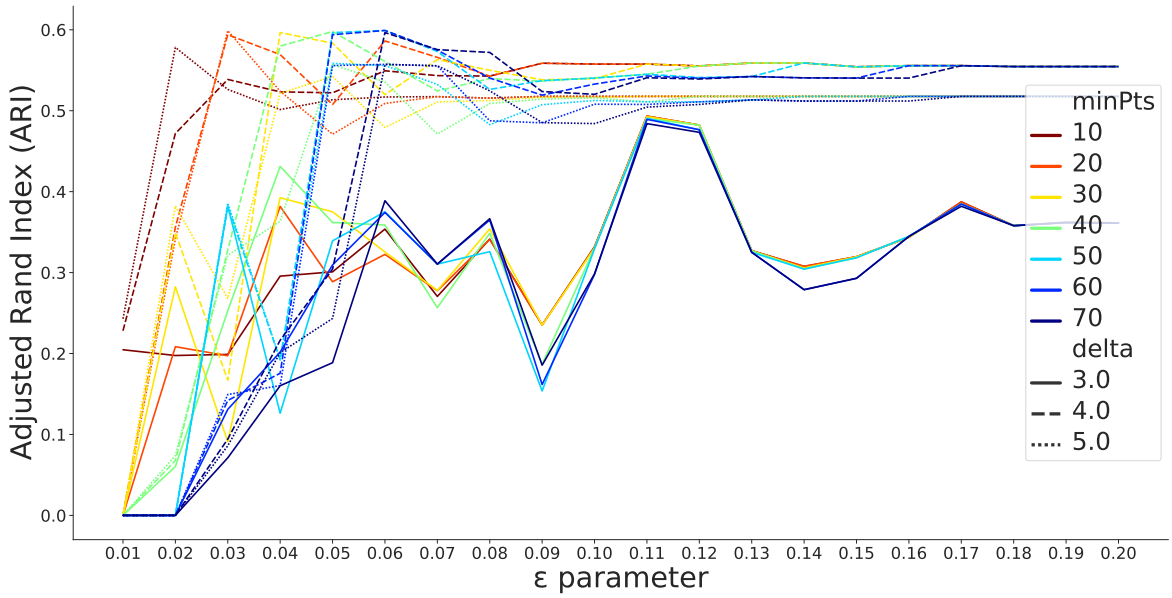


Figure A.1: Dependencies between ϵ, $minPts$ and δ parameters and the Adjusted Rand Index (ARI) for the CoExDBSCAN algorithm.

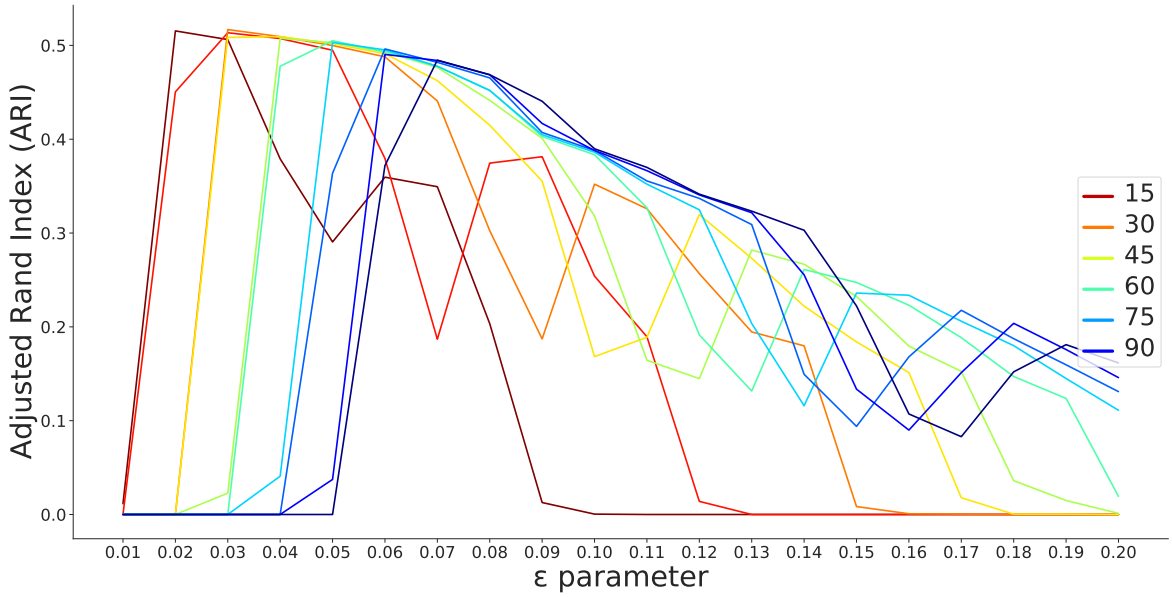


Figure A.2: Dependencies between ϵ and $minPts$ and the Adjusted Rand Index (ARI) for the DBSCAN algorithm.

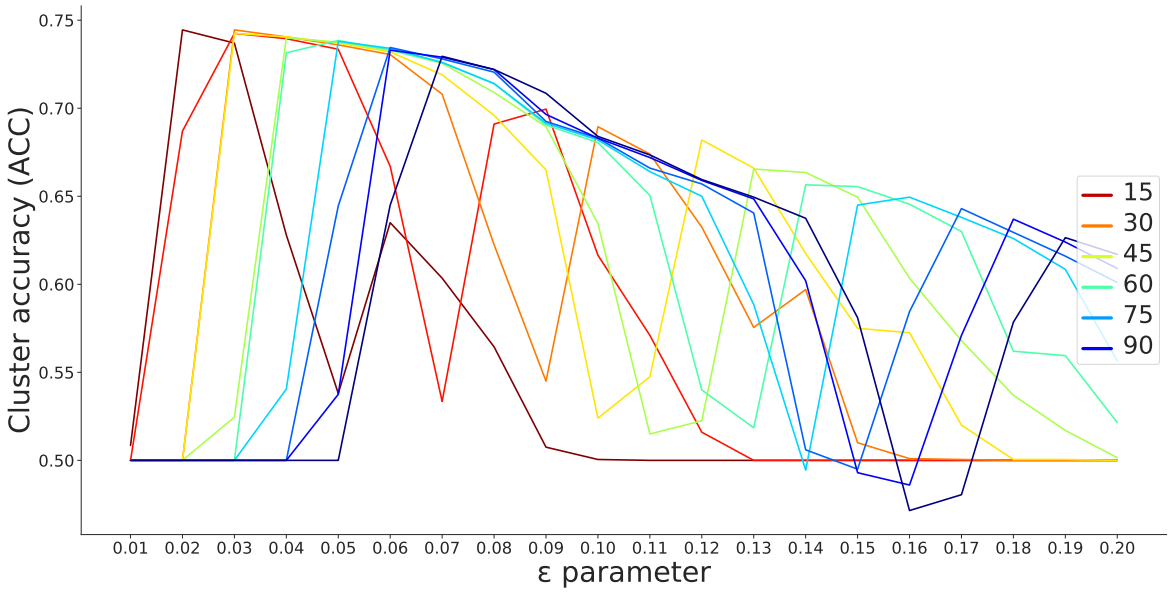


Figure A.3: Dependencies between ϵ and $minPts$ and the Cluster Accuracy (ACC) for the DBSCAN algorithm.

Appendix B

Appendix

B.1 Semi-Supervised Time Point Clustering Parameter Evaluation

This appendix details further the parameter evaluation of the CoExDBSCAN and DBSCAN algorithm presented in Section 6.3.2 for the synthetic dataset.

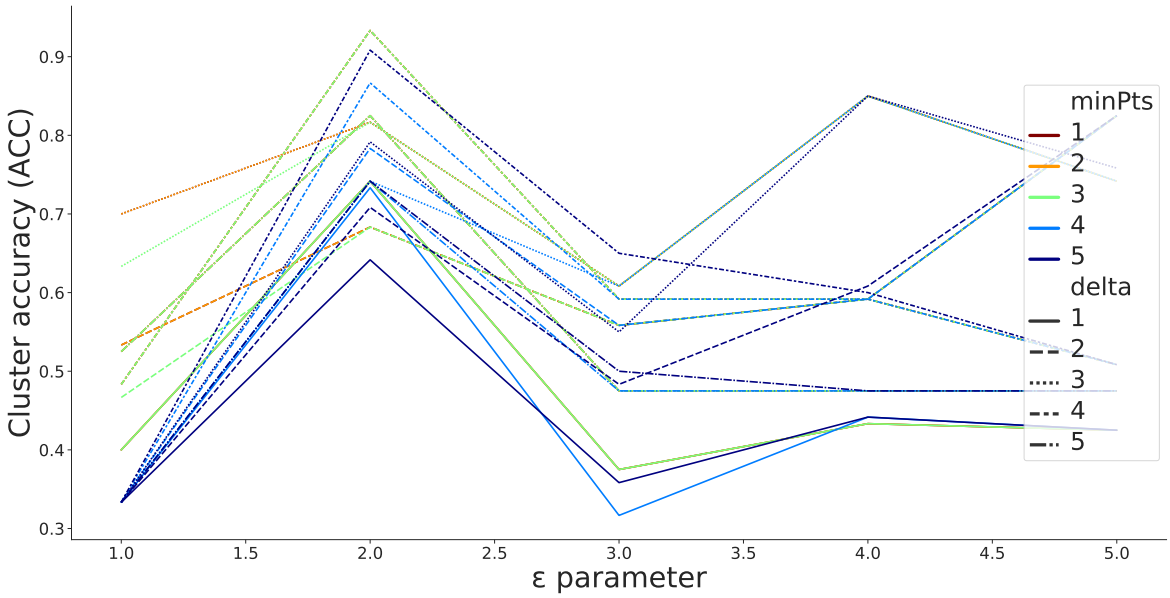


Figure B.1: Dependencies between ϵ, $minPts$ and δ parameters and the Cluster Accuracy (ACC) for the CoExDBSCAN algorithm.

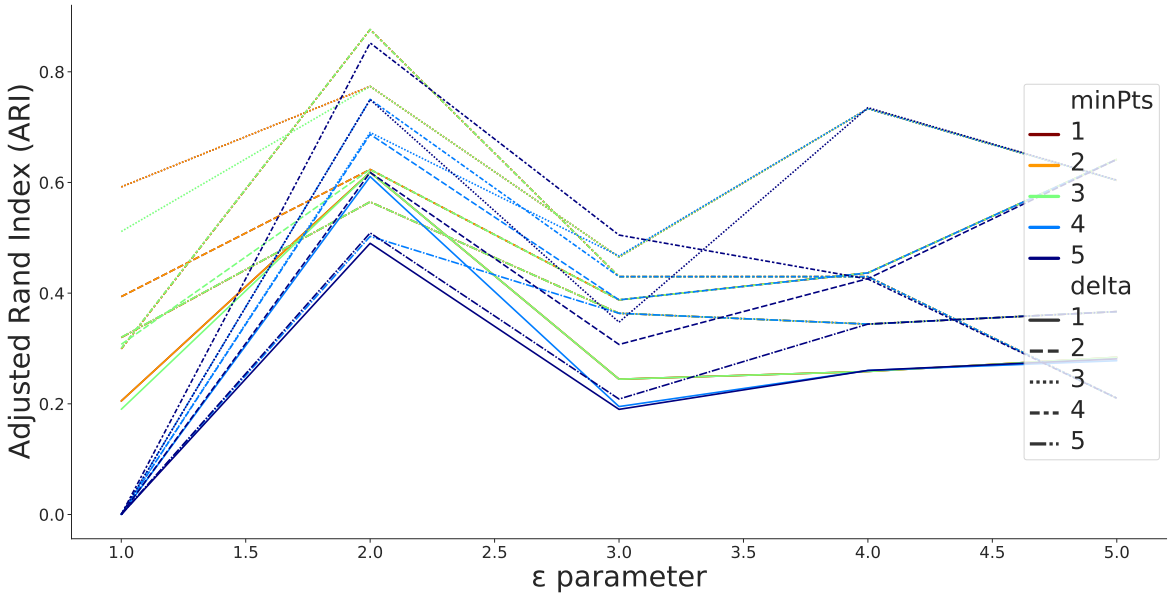


Figure B.2: Dependencies between ϵ, $minPts$ and δ parameters and the Adjusted Rand Index (ARI) for the CoExDBSCAN algorithm.

Bibliography

E. Achtert, C. Böhm, J. David, P. Kröger, and A. Zimek. *Robust Clustering in Arbitrarily Oriented Subspaces*, pages 763–774. Society for Industrial and Applied Mathematics, 2008. doi: 10.1137/1.9781611972788.69. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611972788.69.

C. C. Aggarwal and C. K. Reddy. *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC, 1st edition, 2013. ISBN 1466558210.

C. C. Aggarwal and P. S. Yu. Finding Generalized Projected Clusters in High Dimensional Spaces. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, page 70–81, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581132174. doi: 10.1145/342009.335383. URL https://doi.org/10.1145/342009.335383.

S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah. Time-Series Clustering - A Decade Review. *Inf. Syst.*, 53(C):16–38, Oct. 2015. ISSN 0306-4379.

R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. *SIGMOD Rec.*, 27(2): 94–105, June 1998. ISSN 0163-5808. doi: 10.1145/276305.276314. URL https://doi.org/10.1145/276305.276314.

M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering Points to Identify the Clustering Structure. *SIGMOD Rec.*, 28(2):49–60, June 1999. ISSN 0163-5808. doi: 10.1145/304181.304187. URL https://doi.org/10.1145/304181.304187.

H. Attias. Inferring Parameters and Structure of Latent Variable Models by Variational Bayes. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, page 21–30, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606149.

G. H. Ball and D. J. Hall. ISODATA, a novel method of data analysis and pattern classification. Technical report, Stanford research inst Menlo Park CA, 1965.

S. Basu, A. Banerjee, and R. J. Mooney. Semi-supervised Clustering by Seeding. In *Proceedings of 19th International Conference on Machine Learning (ICML-2002)*, pages 19–26, 2002. URL `http://www.cs.utexas.edu/users/ai-lab?basu:ml02`.

S. Basu, A. Banerjee, and R. J. Mooney. Active Semi-Supervision for Pairwise Constrained Clustering. In *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM-04)*, April 2004. URL `http://www.cs.utexas.edu/users/ai-lab?basu:sdm04`.

S. Basu, I. Davidson, and K. Wagstaff. *Constrained clustering: Advances in algorithms, theory, and applications*. CRC Press, Boca Raton, Florida, 01 2008.

N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. In *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, SIGMOD '90, page 322–331, New York, NY, USA, 1990. Association for Computing Machinery. ISBN 0897913655. doi: 10.1145/93597.98741. URL `https://doi.org/10.1145/93597.98741`.

D. J. Berndt and J. Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, page 359–370. AAAI Press, 1994.

K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When Is "Nearest Neighbor" Meaningful? In C. Beeri and P. Buneman, editors, *Database Theory — ICDT'99*, pages 217–235, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. ISBN 978-3-540-49257-3.

D. Birant and A. Kut. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, 60(1):208–221, 2007. ISSN 0169-023X. doi: https://doi.org/10.1016/j.datak.2006.01.013. URL `https://www.sciencedirect.com/science/article/pii/S0169023X06000218`. Intelligent Data Mining.

C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

R. K. Blashfield and M. S. Aldenderfer. *The Methods and Problems of Cluster Analysis*, pages 447–473. Springer US, Boston, MA, 1988. ISBN 978-1-4613-0893-5. doi: 10.1007/978-1-4613-0893-5_14. URL `https://doi.org/10.1007/978-1-4613-0893-5_14`.

C. Böhm, K. Kailing, P. Kröger, and A. Zimek. Computing Clusters of Correlation Connected Objects. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD '04, page 455–466, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138598. doi: 10.1145/1007568.1007620. URL https://doi.org/10.1145/1007568.1007620.

S. Bony, B. Stevens, D. M. W. Frierson, C. Jakob, M. Kageyama, R. Pincus, T. G. Shepherd, S. C. Sherwood, A. P. Siebesma, A. H. Sobel, M. Watanabe, and M. J. Webb. Clouds, circulation and climate sensitivity. *Nature Geoscience*, 8(4):261–268, Apr 2015. ISSN 1752-0908. doi: 10.1038/ngeo2398.

C. Borger, M. Schneider, B. Ertl, F. Hase, O. E. García, M. Sommer, M. Höpfner, S. A. Tjemkes, and X. Calbet. Evaluation of MUSICA IASI tropospheric water vapour profiles using theoretical error assessments and comparisons to GRUAN Vaisala RS92 measurements. *Atmospheric Measurement Techniques*, 11(9):4981–5006, 2018. doi: 10.5194/amt-11-4981-2018. URL https://amt.copernicus.org/articles/11/4981/2018/.

R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-37456-2.

M. Claesen and B. De Moor. Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*, 2015.

T. H. Cormen. *Introduction to algorithms*. MIT Press, Cambridge, Massachusetts, third edition edition, 2009. ISBN 9780262033848. URL http://swbplus.bsz-bw.de/bsz306807912inh.htm.

F. Dahinden, F. Aemisegger, H. Wernli, M. Schneider, C. J. Diekmann, B. Ertl, P. Knippertz, M. Werner, and S. Pfahl. Disentangling different moisture transport pathways over the eastern subtropical North Atlantic using multi-platform isotope observations and high-resolution numerical modelling. *Atmospheric Chemistry and Physics Discussions*, 2021:1–49, 2021. doi: 10.5194/acp-2021-269. URL https://acp.copernicus.org/preprints/acp-2021-269/.

I. Davidson and S. S. Ravi. *Clustering With Constraints: Feasibility Issues and the k-Means Algorithm*, pages 138–149. Society for Industrial and Applied Mathematics, 2005. doi: 10.1137/1.9781611972757.13. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611972757.13.

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)*, 39(1):1–22, 1977.

D. B. Dias, R. C. B. Madeo, T. Rocha, H. H. Biscaro, and S. M. Peres. Hand movement recognition for Brazilian Sign Language: A study using distance-based neural networks. In *2009 International Joint Conference on Neural Networks*, pages 697–704, 2009.

C. J. Diekmann, M. Schneider, B. Ertl, F. Hase, O. García, F. Khosrawi, E. Sepúlveda, P. Knippertz, and P. Braesicke. The global and multi-annual MUSICA IASI $\{H2O, \delta D\}$ pair dataset. *Earth System Science Data*, 13(11):5273–5292, 2021a. doi: 10.5194/essd-13-5273-2021. URL `https://essd.copernicus.org/articles/13/5273/2021/`.

C. J. Diekmann, M. Schneider, P. Knippertz, A. J. de Vries, S. Pfahl, F. Aemisegger, F. Dahinden, B. Ertl, F. Khosrawi, H. Wernli, and et al. A Lagrangian perspective on stable water isotopes during the West African Monsoon. *Earth and Space Science Open Archive*, page 43, 2021b. doi: 10.1002/essoar.10506628.1. URL `https://doi.org/10.1002/essoar.10506628.1`.

P. Diggle. *Statistical analysis of spatial and spatio-temporal point patterns*. Monographs on statistics and applied probability ; 128. CRC Press, Taylor & Francis, Boca Raton [u.a.], 3. ed. edition, 2014. ISBN 9781466560239. URL `http://www.lancaster.ac.uk/staff/diggle/pointpatternbook/`.

D. Dinler and M. K. Tural. *A Survey of Constrained Clustering*, pages 207–235. Springer International Publishing, Cham, 2016. ISBN 978-3-319-24211-8.

H. E. Driver and A. L. Kroeber. *Quantitative expression of cultural relationships*, volume 31. Berkeley: University of California Press, 1932.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

R. O. Duda and P. E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM*, 15(1):11–15, Jan. 1972. ISSN 0001-0782. doi: 10.1145/361237.361242. URL `https://doi.org/10.1145/361237.361242`.

B. Ertl, J. Meyer, A. Streit, and M. Schneider. Application of Mixtures of Gaussians for Tracking Clusters in Spatio-temporal Data. In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge*

Management - Volume 1: KDIR,, pages 45–54. INSTICC, SciTePress, 2019. ISBN 978-989-758-382-7. doi: 10.5220/0007949700450054.

B. Ertl, J. Meyer, M. Schneider, and A. Streit. CoExDBSCAN: Density-based Clustering with Constrained Expansion. In *Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1: KDIR,*, pages 104–115. INSTICC, SciTePress, 2020. ISBN 978-989-758-474-9. doi: 10.5220/0010131201040115.

B. Ertl, J. Meyer, M. Schneider, and A. Streit. Semi-Supervised Time Point Clustering for Multivariate Time Series. *Proceedings of the Canadian Conference on Artificial Intelligence*, 6 2021a. doi: 10.21428/594757db.9fa1eff5. URL `https://caiac.pubpub.org/pub/a3py333z`. https://caiac.pubpub.org/pub/a3py333z.

B. Ertl, M. Schneider, C. Diekmann, J. Meyer, and A. Streit. A Semi-supervised Approach for Trajectory Segmentation to Identify Different Moisture Processes in the Atmosphere. In M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, editors, *Computational Science – ICCS 2021*, pages 264–277, Cham, 2021b. Springer International Publishing. ISBN 978-3-030-77961-0.

M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1):32–40, 1975.

S. Gaffney and P. Smyth. Trajectory Clustering with Mixtures of Regression Models. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, page 63–72, New York, NY, USA, 1999. Association for Computing Machinery. ISBN 1581131437. doi: 10.1145/312129.312198.

S. J. Gaffney, A. W. Robertson, P. Smyth, S. J. Camargo, and M. Ghil. Probabilistic clustering of extratropical cyclones using regression mixture models. *Climate Dynamics*, 29(4):423–440, Sep 2007. ISSN 1432-0894. doi: 10.1007/s00382-007-0235-z.

O. E. García, M. Schneider, B. Ertl, E. Sepúlveda, C. Borger, C. Diekmann, A. Wiegele, F. Hase, S. Barthlott, T. Blumenstock, U. Raffalski, A. Gómez-Peláez, M. Steinbacher, L. Ries, and A. M. de Frutos. The MUSICA IASI CH4 and N2O products and

their comparison to HIPPO, GAW and NDACC FTIR references. *Atmospheric Measurement Techniques*, 11(7):4171–4215, 2018. doi: 10.5194/amt-11-4171-2018. URL https://amt.copernicus.org/articles/11/4171/2018/.

H. Greenspan, J. Goldberger, and L. Ridel. A Continuous Probabilistic Framework for Image Matching. *Computer Vision and Image Understanding*, 84(3):384–406, 2001. ISSN 1077-3142. doi: https://doi.org/10.1006/cviu.2001.0946. URL https://www.sciencedirect.com/science/article/pii/S1077314201909464.

P. D. Grünwald. The Minimum Description Length Principle, Mar 2007. URL https://doi.org/10.7551/mitpress/4643.001.0001.

D. Hallac, S. Vare, S. Boyd, and J. Leskovec. Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 215–223, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348874.

J. L. Hodges. The significance probability of the smirnov two-sample test. *Arkiv för Matematik*, 3:469–486, 1958.

L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1): 193–218, Dec 1985. ISSN 1432-1343. doi: 10.1007/BF01908075. URL https://doi.org/10.1007/BF01908075.

Huidong Jin, Man-Leung Wong, and K. . Leung. Scalable model-based clustering for large databases based on data summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1710–1719, 2005. doi: 10.1109/TPAMI.2005.226.

D. Ienco and R. Interdonato. Deep Multivariate Time Series Embedding Clustering via Attentive-Gated Autoencoder. In H. W. Lauw, R. C.-W. Wong, A. Ntoulas, E.-P. Lim, S.-K. Ng, and S. J. Pan, editors, *Advances in Knowledge Discovery and Data Mining*, pages 318–329, Cham, 2020. Springer International Publishing. ISBN 978-3-030-47426-3.

A. K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651 – 666, 2010. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2009.09.011. URL http://www.sciencedirect.com/science/article/pii/S0167865509002323. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR).

H. Jin, K.-S. Leung, M.-L. Wong, and Z.-B. Xu. Scalable model-based cluster analysis using clustering features. *Pattern Recognition*, 38(5):637–649, 2005. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2004.07.012. URL https://www.sciencedirect.com/science/article/pii/S0031320304003929.

P. Kalnis, N. Mamoulis, and S. Bakiras. On Discovering Moving Clusters in Spatio-temporal Data. In C. Bauzer Medeiros, M. J. Egenhofer, and E. Bertino, editors, *Advances in Spatial and Temporal Databases*, pages 364–381, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31904-7.

L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

E. Keogh and J. Lin. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and Information Systems*, 8(2):154–177, 8 2005. ISSN 0219-3116.

S. Kisilevich, F. Mansmann, M. Nanni, and S. Rinzivillo. *Spatio-temporal clustering*, pages 855–874. Springer US, Boston, MA, 2010. ISBN 978-0-387-09823-4. doi: 10.1007/978-0-387-09823-4_44. URL https://doi.org/10.1007/978-0-387-09823-4_44.

G. Kollios, D. Gunopulos, N. Koudas, and S. Berchtold. Efficient biased sampling for approximate clustering and outlier detection in large data sets. *IEEE Transactions on Knowledge and Data Engineering*, 15(5):1170–1187, 2003. doi: 10.1109/TKDE.2003.1232271.

S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951. doi: 10.1214/aoms/1177729694. URL https://doi.org/10.1214/aoms/1177729694.

J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory Clustering: A Partition-and-Group Framework. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, page 593–604, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595936868. doi: 10.1145/1247480.1247546.

S. Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137, 1982.

P. S. Maciąg. A Survey on Data Mining Methods for Clustering Complex Spatiotemporal Data. In S. Kozielski, D. Mrozek, P. Kasprowski, B. Małysiak-Mrozek, and

D. Kostrzewa, editors, *Beyond Databases, Architectures and Structures. Towards Efficient Solutions for Data Analysis and Knowledge Representation*, pages 115–126, Cham, 2017. Springer International Publishing. ISBN 978-3-319-58274-0.

J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

P. K. Mallapragada, R. Jin, and A. Jain. Non-parametric Mixture Models for Clustering. In E. R. Hancock, R. C. Wilson, T. Windeatt, I. Ulusoy, and F. Escolano, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, pages 334–343, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-14980-1.

G. J. McLachlan, S. X. Lee, and S. I. Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6:355–378, 2019.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass. [u.a.], 2012. ISBN 9780262018258.

A. Moitra. Algorithmic Aspects of Machine Learning. Spring 2015., 2015. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA.

S. Nassar, J. Sander, and C. Cheng. Incremental and Effective Data Summarization for Dynamic Hierarchical Clustering. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD '04, page 467–478, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138598. doi: 10.1145/1007568.1007621. URL `https://doi.org/10.1145/1007568.1007621`.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, page 849–856, Cambridge, MA, USA, 2001. MIT Press.

D. Noone. Pairing Measurements of the Water Vapor Isotope Ratio with Humidity to Deduce Atmospheric Moistening and Dehydration in the Tropical Midtroposphere. *Journal of Climate*, 25(13):4476 – 4494, 2012. doi: 10.1175/JCLI-D-11-00582.1.

D. Noone, J. Galewsky, Z. D. Sharp, J. Worden, J. Barnes, D. Baer, A. Bailey, D. P. Brown, L. Christensen, E. Crosson, F. Dong, J. V. Hurley, L. R. Johnson, M. Strong, D. Toohey, A. Van Pelt, and J. S. Wright. Properties of air mass mixing and humidity

in the subtropics from measurements of the D/H isotope ratio of water vapor at the Mauna Loa Observatory. *Journal of Geophysical Research: Atmospheres*, 116(D22), 2011. doi: 10.1029/2011JD015773.

L. Paci and F. Finazzi. Dynamic model-based clustering for spatio-temporal data. *Statistics and Computing*, 28(2):359–374, Mar 2018. ISSN 1573-1375. doi: 10.1007/s11222-017-9735-9. URL `https://doi.org/10.1007/s11222-017-9735-9`.

C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

L. Parsons, E. Haque, and H. Liu. Subspace Clustering for High Dimensional Data: A Review. *SIGKDD Explor. Newsl.*, 6(1):90–105, June 2004. ISSN 1931-0145. doi: 10.1145/1007730.1007731. URL `https://doi.org/10.1145/1007730.1007731`.

K. Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

D. Pelleg and D. Baras. K-Means with Large and Noisy Constraint Sets. In J. N. Kok, J. Koronacki, R. L. d. Mantaras, S. Matwin, D. Mladenič, and A. Skowron, editors, *Machine Learning: ECML 2007*, pages 674–682, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-74958-5.

F. Petitjean, A. Ketterlin, and P. Gançarski. A Global Averaging Method for Dynamic Time Warping, with Applications to Clustering. *Pattern Recogn.*, 44(3):678–693, Mar. 2011. ISSN 0031-3203.

S. Pfahl, H. Wernli, and K. Yoshimura. The isotopic composition of precipitation from a winter storm – a case study with the limited-area model COSMO$_{iso}$. *Atmospheric Chemistry and Physics*, 12(3):1629–1648, 2012. doi: 10.5194/acp-12-1629-2012. URL `https://acp.copernicus.org/articles/12/1629/2012/`.

P. Pons and M. Latapy. Computing Communities in Large Networks Using Random Walks. In p. Yolum, T. Güngör, F. Gürgen, and C. Özturan, editors, *Computer and Information Sciences - ISCIS 2005*, pages 284–293, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-32085-2.

M. Pourrajabi, D. Moulavi, R. J. G. B. Campello, A. Zimek, J. Sander, and R. Goebel. Model Selection for Semi-Supervised Clustering. In *Proceedings of the 17th International Conference on Extending Database Technology, EDBT 2014, Athens, Greece, March 24-28, 2014*, pages 331–342, Konstanz, 2014. OpenProceedings.org.

W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971a. ISSN 01621459. URL http://www.jstor.org/stable/2284239.

W. M. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971b.

S. J. Rey and L. Anselin. PySAL: A Python Library of Spatial Analytical Methods. *The Review of Regional Studies*, 37(1):5–27, 2007.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(78)90005-5. URL https://www.sciencedirect.com/science/article/pii/0005109878900055.

S. J. Roberts. Parametric and non-parametric unsupervised cluster analysis. *Pattern Recognition*, 30(2):261–272, 1997. ISSN 0031-3203. doi: https://doi.org/10.1016/S0031-3203(96)00079-9. URL https://www.sciencedirect.com/science/article/pii/S0031320396000799.

S. Rodpongpun, V. Niennattrakul, and C. A. Ratanamahatana. Selective Subsequence Time Series clustering. *Knowledge-Based Systems*, 35:361–368, 2012. ISSN 0950-7051.

F. Role, S. Morbieu, and M. Nadif. CoClust: A Python Package for Co-Clustering. *Journal of Statistical Software, Articles*, 88(7):1–29, 2019. ISSN 1548-7660.

P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. URL https://www.sciencedirect.com/science/article/pii/0377042787901257.

H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1): 43–49, 1978.

M. Schneider and F. Hase. Optimal estimation of tropospheric H_2O and δD with IASI/METOP. *Atmospheric Chemistry and Physics*, 11(21):11207–11220, 2011. doi: 10.5194/acp-11-11207-2011. URL https://acp.copernicus.org/articles/11/11207/2011/.

M. Schneider, B. Ertl, C. J. Diekmann, F. Khosrawi, A. N. Röhling, F. Hase, D. Dubravica, O. E. García, E. Sepúlveda, T. Borsdorff, J. Landgraf, A. Lorente, H. Chen, R. Kivi, T. Laemmel, M. Ramonet, C. Crevoisier, J. Pernin, M. Steinbacher, F. Meinhardt, N. M. Deutscher, D. W. T. Griffith, V. A. Velazco, and D. F. Pollard. Synergetic use of IASI and TROPOMI space borne sensors for generating a tropospheric methane profile product. *Atmospheric Measurement Techniques Discussions*, 2021:1–37, 2021a. doi: 10.5194/amt-2021-31. URL `https://amt.copernicus.org/preprints/amt-2021-31/`.

M. Schneider, B. Ertl, C. J. Diekmann, F. Khosrawi, A. Weber, F. Hase, M. Höpfner, O. E. García, E. Sepúlveda, and D. Kinnison. Design and description of the MUSICA IASI full retrieval product. *Earth System Science Data Discussions*, 2021: 1–51, 2021b. doi: 10.5194/essd-2021-75. URL `https://essd.copernicus.org/preprints/essd-2021-75/`.

E. Schubert and A. Zimek. ELKI: A large open-source library for data analysis - ELKI release 0.7.5 "Heidelberg". *CoRR*, abs/1902.03616, 2019. URL `http://arxiv.org/abs/1902.03616`.

E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Trans. Database Syst.*, 42(3), July 2017. ISSN 0362-5915. doi: 10.1145/3068335. URL `https://doi.org/10.1145/3068335`.

G. Schwarz. Estimating the Dimension of a Model. *Ann. Statist.*, 6(2):461–464, 03 1978. doi: 10.1214/aos/1176344136. URL `https://doi.org/10.1214/aos/1176344136`.

D. W. Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

Scott Shaobing Chen and P. S. Gopalakrishnan. Clustering via the Bayesian information criterion with applications in speech recognition. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, volume 2, pages 645–648 vol.2, 1998. doi: 10.1109/ICASSP.1998.675347.

S. Seabold and J. Perktold. statsmodels: Econometric and statistical modeling with Python. In *9th Python in Science Conference*, 2010.

B. Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

M. Sprenger and H. Wernli. The LAGRANTO Lagrangian analysis tool – version 2.0. *Geoscientific Model Development*, 8(8):2569–2586, 2015. doi: 10.5194/gmd-8-2569-2015.

K. Steinhaeuser, N. V. Chawla, and A. R. Ganguly. Comparing Predictive Power in Climate Data: Clustering Matters. In D. Pfoser, Y. Tao, K. Mouratidis, M. A. Nascimento, M. Mokbel, S. Shekhar, and Y. Huang, editors, *Advances in Spatial and Temporal Databases*, pages 39–55, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-22922-0.

H. Steinhaus. Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci*, 1 (804):801, 1956.

R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, and E. Woods. Tslearn, A Machine Learning Toolkit for Time Series Data. *Journal of Machine Learning Research*, 21(118):1–6, 2020. URL `http://jmlr.org/papers/v21/20-091.html`.

K. Toride, K. Yoshimura, M. Tada, C. Diekmann, B. Ertl, F. Khosrawi, and M. Schneider. Potential of Mid-tropospheric Water Vapor Isotopes to Improve Large-Scale Circulation and Weather Predictability. *Geophysical Research Letters*, 48(5):e2020GL091698, 2021. doi: https://doi.org/10.1029/2020GL091698. URL `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020GL091698`. e2020GL091698 2020GL091698.

R. Tryon. Cluster Analysis; Correlation Profiles and Orthometric (factor) Analysis for the Isolation of Unities in Mind and Personality. *Ann Arbor: Edwards*, page 122, 1939.

Q. Tu, F. Hase, T. Blumenstock, M. Schneider, A. Schneider, R. Kivi, P. Heikkinen, B. Ertl, C. Diekmann, F. Khosrawi, M. Sommer, T. Borsdorff, and U. Raffalski. Intercomparison of arctic XH2O observations from three ground-based Fourier transform infrared networks and application for satellite validation. *Atmospheric Measurement Techniques*, 14(3):1993–2011, 2021. doi: 10.5194/amt-14-1993-2021. URL `https://amt.copernicus.org/articles/14/1993/2021/`.

K. Wagstaff and C. Cardie. Clustering with Instance-Level Constraints. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 1103–1110, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained K-Means Clustering with Background Knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 577–584, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

S. Wang, T. Cai, and C. F. Eick. New Spatiotemporal Clustering Algorithms and their Applications to Ozone Pollution. In *2013 IEEE 13th International Conference on Data Mining Workshops*, pages 1061–1068, 2013. doi: 10.1109/ICDMW.2013.14.

T. Warren Liao. Clustering of time series data—a survey. *Pattern Recognition*, 38(11): 1857 – 1874, 2005. ISSN 0031-3203.

A. Weber. *Storage-Efficient Analysis of Spatio-Temporal Data with Application to Climate Research*. PhD thesis, Karlsruhe Institute of Technology, Aug. 2019. URL `https://doi.org/10.5281/zenodo.3360021`.

A. Wiegele, M. Schneider, F. Hase, S. Barthlott, O. E. García, E. Sepúlveda, Y. González, T. Blumenstock, U. Raffalski, M. Gisi, and R. Kohlhepp. The MUSICA MetOp/IASI H_2O and δD products: characterisation and long-term comparison to NDACC/FTIR data. *Atmospheric Measurement Techniques*, 7(8):2719–2732, 2014. doi: 10.5194/amt-7-2719-2014. URL `https://amt.copernicus.org/articles/7/2719/2014/`.

D. Xu and Y. Tian. A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, 2(2):165–193, Jun 2015. ISSN 2198-5812. doi: 10.1007/s40745-015-0040-1. URL `https://doi.org/10.1007/s40745-015-0040-1`.

Q. Zhang, J. Wu, P. Zhang, G. Long, and C. Zhang. Salient Subsequence Learning for Time Series Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2193–2207, 2019.

X. J. Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

S. Zolhavarieh, S. Aghabozorgi, and Y. W. Teh. A Review of Subsequence Time Series Clustering. *The Scientific World Journal*, 2014:312521, 7 2014. ISSN 2356-6140.

J. Zubin. A technique for measuring like-mindedness. *The Journal of Abnormal and Social Psychology*, 33(4):508, 1938.