universität wien

DISSERTATION / DOCTORAL THESIS

Titel der Disseratation / Title of the Doctoral Thesis

„Clustering and Anomaly Detection from Heterogeneous Data“

verfasst von / submitted by

Sahar Behzadi Soheil

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Doktor der Technischen Wissenschaften (Dr.techn.)

Wien, 2020 / Vienna, 2020

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA A 786 880
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Informatik
Betreut von / Supervisor:	Univ.-Prof. Dipl.-Inform.Univ. Dr. Claudia Plant

Abstract

Most data mining algorithms (e.g. clustering algorithms) are designed for single–type data sets when attributes consist of only a specific data type, e.g. pure numerical or pure categorical attributes. However, many applications generate a variety of different mixed–type data sets where attributes might be of different natures. It is already well–understood that a simple transformation of a data type into another one is not sufficient since, in this case, relationships between values (such as a certain order among variables) are artificially introduced. Thus, a possible challenge in this respect is to appropriately integrate various data types such that one could efficiently analyze objects without any accuracy or information loss. Therefore, in this thesis, we aim at introducing effective and efficient algorithms dealing with heterogeneous (mixed–type) data sets Considering various data mining tasks. In this regard, we utilize interesting characteristics of every data type, e.g. a natural conceptual hierarchy among categorical information, to introduce novel data mining algorithms. Thereby, we try to integrate attributes of different data types and preserve the original form of information instead of converting data types.

Zusammenfassung

Die meisten Algorithmen aus dem Bereich des Data Mining (z. B. Clustering Algorithmen) sind für Datensätze mit ein und demselben Typ ausgelegt, das heißt die Attribute bestehen nur aus einem bestimmten Datentyp, z. B. aus rein numerischen oder rein kategorischen Attributen. Viele Anwendungen erzeugen jedoch eine Vielzahl verschiedener gemischter Datensätze, bei denen die Attribute unterschiedlicher Natur sein können. Es ist allgemein bekannt, dass eine einfache Transformation eines Datentyps in einen anderen nicht ausreicht, da in diesem Fall Beziehungen zwischen Werten (wie z.B. eine bestimmte Reihenfolge zwischen Variablen) künstlich eingeführt werden. Daher besteht eine mögliche Herausforderung in dieser Hinsicht darin, verschiedene Datentypen angemessen zu integrieren, so dass man Objekte effizient und ohne Genauigkeits- oder Informationsverlust analysieren kann. Das Ziel in dieser Arbeit ist es, effektive und effiziente Algorithmen für den Umgang mit heterogenen (gemischten) Datensätzen unter Berücksichtigung verschiedener Aufgaben des Data Mining einzuführen. In dieser Hinsicht nutzen wir interessante Eigenschaften jedes Datentyps, z.B. eine natürliche konzeptuelle Hierarchie zwischen kategorialen Informationen, um neuartige Algorithmen im Data Mining einzuführen. Dabei versuchen wir, Attribute verschiedener Datentypen zu integrieren und die ursprüngliche Form der Information zu erhalten, anstatt Datentypen zu konvertieren.

Acknowledgments

First and foremost, I would like to express my gratitude to my supervisor, Prof. Claudia Plant, for her continuous support, encouragement, constructive discussions, and her ideas during my Ph.D. time. You understood my situation and created an inspirational and friendly atmosphere which helped me to progress in an enjoyable environment.

My sincere thanks also go to Prof. Christian Böhm who provided me a great opportunity to join his team (Data Mining in Medicine at LMU) as a guest researcher, and who gave me access to the office and research facilities. It was my pleasure to meet your group and cooperate with you. Besides my supervisor, I would like to thank Prof. Allan Hanbury and Prof. Andreas Züfle for reviewing my thesis and their insightful comments.

I would also like to show my deep appreciation to Ben and Martin, it was a great sharing office with you guys during the last four years. You have been always there, especially when things did not go smoothly. Thanks for your proofreading my thesis, you helped me finalize my project. I would like to thank Kateřina for opening up a new interesting research topic for me. You supported me and offered deep insight into the study. To other colleagues, Lukas, Lena, Ylli, and Max, thank you very much for your advice and help. All of you played positive roles in my development towards a Ph.D. degree. I wish to acknowledge the help provided by the technical and support staff, especially Ewald, in the Data Mining research group of the University of Vienna.

I am truly grateful to my family: my parents and to my brother and sister for supporting me spiritually throughout writing this thesis and my life in general. Especial thanks to my lovely daughter, Awa, who is the sunshine in my life and showed me how beautiful life can be.

And finally, last but by no means least, I am greatly indebted to my husband, my love, my advisor, and my sport trainer, Mehdi. You have been always cheering me up, supporting me when there was a failure, and celebrating with every success. Thanks for being so kind and supportive. Without your help, this thesis would never have seen its conclusion.

Bibliographic Note

Most of the results of this thesis were already published in conference proceedings and journal articles. Therefore, the chapters of this thesis are based on the following publications and manuscripts:

- **Paper A:** Sahar Behzadi, Nikola Müller, Claudia Plant, and Christian Böhm. "Clustering of Mixed-type Data Considering Concept Hierarchies". In: *The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. 2019, Chapter 6.
- **Paper B:** Sahar Behzadi, Nikola Müller, Claudia Plant, and Christian Böhm. "Clustering of Mixed-type Data Considering Concept Hierarchies: problem specification and algorithm". In: *International Journal of Data Science and Analytics*. 2020, Chapter 6.
- **Paper C:** Sahar Behzadi, M. A. Ibrahim, and Claudia Plant. "Parameter Free Mixed-Type Density-Based Clustering". In: *International Conference on Database and Expert Systems Applications (DEXA)*. 2018, Chapter 7.
- **Paper D:** Sahar Behzadi, Hermann Hinterhauser, and Claudia Plant. "ITGC: Information-theoretic grid-based clustering". In: *International Conference on Extending Database Technology (EDBT)*. 2019, Chapter 8.
- **Paper E:** Sahar Behzadi, Kateřina Hlaváčková-Schindler, and Claudia Plant. "Dependency anomaly detection for heterogeneous time series: A Granger-Lasso approach". In: *IEEE International Conference on Data Mining (ICDM) workshops*. 2017, Chapter 9.
- **Paper F:** Sahar Behzadi, Kateřina Hlaváčková-Schindler, and Claudia Plant. "Granger Causality for Heterogeneous Processes". In: *The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. 2019, Chapter 10.
- **Paper G:** Sahar Behzadi, Niklas Preschern, Kateřina Hlaváčková-Schindler, and Claudia Plant. "Anomaly Detection in Heterogeneous Time Series by Causality Mining". In: *Knowledge and Information Systems, submitted for publishing*. 2020, Chapter 10.
- **Paper H:** Sahar Behzadi, Benjamin Schelling, and Claudia Plant. "ITGH: Information-theoretic Granger Causal Inference on Heterogeneous Data". In: *The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. 2020, Chapter 11.
- **Paper I:** Sahar Behzadi, Benjamin Schelling, and Claudia Plant. "Information-theoretic Granger Causal Inference on Heterogeneous Data: Problem specification and algorithm". In: *International Journal of Data Science and Analytics, submitted for publishing*. 2020, Chapter 11.

Papers written during the time of my PhD as second author which do not fit the scope of this thesis:

- **Paper J:** Benjamin Schelling, Lena G. M. Bauer, Sahar Behzadi, and Claudia Plant. "Utilizing Structure-rich Features to improve Clustering". In: *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases 2020*. 2020.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Data mining is a particular step of a wider process, *Knowledge Discovery in Databases* (KDD). As Figure 1.1 shows, a KDD process involves using a database along with any required selection, pre–processing, sub–sampling, and transformations of the data; applying data mining methods (algorithms) to enumerate patterns from it; and evaluating the products of data mining to identify the subset of the enumerated patterns deemed knowledge [23]. Essentially, data mining consists of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns (or models) over the data [23].

Basic data mining tasks comprise clustering, classification, association rule mining and frequent pattern mining, regression and anomaly (outlier) detection. Among them, clustering is one of the interesting data mining tasks which groups data objects in the way that objects in the same groups (clusters) are more similar (based on some criteria) to each other than to those in other groups (clusters). Clustering algorithms usually differ significantly in their understanding of what constitutes a cluster and how to efficiently find them. After applying a clustering algorithm and detecting meaningful groups, finding objects, that are considerably different from other objects, leads to more accurate data analysis. This process is one of the data mining tasks which is called anomaly (or outlier) detection.

Most data mining algorithms have been designed only for pure homogeneous data sets.

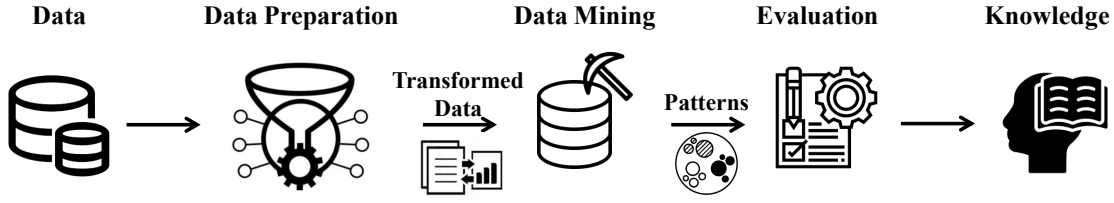


Figure 1.1: Overview of a KDD process.

However, many applications, e.g. population or statistical surveys, climatological reports, generate a mixture of data objects consisting of attributes from different natures (distributions). Mining such complex data sets is a non–trivial task and typically is not achieved by well–known algorithms designed for a special data type. Complex data might be interpreted in different ways. An important type of complex data is in the form of graphs. Another form of complexity is from data that are non–i.i.d. (*independent and identically distributed*), e.g. time series. However, in most domains, the objects of interest are not independent of each other and are not of a single type. We call this kind of complex data, which is of interest in this thesis, heterogeneous data.

When mining heterogeneous data sets, one of the basic and straightforward approaches is to homogenize the data as much as possible. This might be achieved by converting a data type to another one. However, it is already well–understood that this approach has some severe drawbacks. Most of the time, a simple conversion or any specific assumptions might lead to information loss. Moreover, relations between values, such as a certain order among objects, are artificially defined. Thus, our main approach in this thesis is to ***preserve*** original characteristics of every data type and try to ***integrate*** data of different natures as much as possible in order to avoid any information loss. In this respect, we incorporate useful characteristics of every data type and introduce an integrative approach to cope with the aforementioned drawbacks.

The remainder of this cumulative Ph.D. thesis is structured as follows. Chapter 2 defines relevant terms and the background one needs to follow various concepts in this thesis. Chapter 3 specifies the research problem, highlights challenges, and states the research questions. Chapter 4 lists all scientific papers that have been published and details the scientific contributions that have been made in the course of this thesis. Finally, Chapter 5 concludes this thesis and gives an outlook on potential future works.

CHAPTER 2

Background

2.1 Clustering

Clustering is one of various data mining tasks which groups data objects in the way that objects in the same group (cluster) are more similar (based on some criteria) to each other than to those in other groups (clusters). Clustering algorithms usually differ significantly in their understanding of what constitutes a cluster and how to efficiently find them. In the following, we introduce some of the most familiar clustering approaches. In the end, we give an introduction about one of the useful clustering criteria which we often employ in our proposed algorithms.

2.1.1 Clustering Approaches

- **Grid–based clustering**
  One of the well–known clustering approaches is grid–based clustering where any data set is partitioned using a set of grid–cells and data points are assigned to an appropriate grid cell. Grid–based methods [1], [43], [38] quantize the object space into a finite number of cells (hyper–rectangles) and then perform the required operations on the quantized space. The main advantage of grid–based methods is their fast processing time which depends on the number of cells in the grid. In other words, no distance computation is required and the clustering is performed on summaries and not on the individual objects. Thus, the complexity of grid–based algorithms is usually *O(number of populated grid cells)* and not *O(number of objects)*.

  Beyond their ability to deal with noisy data sets, grid–based clustering algorithms are able to identify clusters irrespective of their shapes. Unlike most of the clustering algorithms which require an often initialization phase, algorithms in this category are insensitive to the order of input records and therefore are deterministic.

- **Partition–based clustering**
  Among various clustering approaches, some of them attract a lot of attention because of

their advantages. Partition–based clustering algorithms are popular due to their simplicity and their relative efficiency [29], [3]. K–means [29] is a well–know and well–studied representative for this approach where initially the data is partitioned into k non–empty sets (clusters) and iteratively the data points are assigned to their nearest cluster. Despite the mentioned advantages, clustering algorithms in this group suffer from some drawbacks. Often in this category, the number of partitions (clusters) k should be specified in the beginning and results are not deterministic because of their sensitivity to the initialization. Moreover, they are not suitable to discover clusters with non–convex shapes. As a subset of this group, model–based clustering algorithms consider a specific distribution model to represent data sets. Among them, Expectation–Maximization (EM) algorithm interprets the data as a mixture of Gaussian distributions [20].

- **Density–based clustering**
  Algorithms in this category (e.g. [22], [4]) are appropriately designed to deal with arbitrary shaped clusters. Unlike partition–based algorithms, algorithms in this category are able to deal with noisy data sets. However, we usually need to specify some parameters representing characteristics of dense regions which, mostly, are not straightforward to specify. Additionally, density–based algorithms are not designed to efficiently deal with clusters with various densities.

2.1.2 Parameter–free clustering

Most clustering algorithms require to specify input parameters which are usually difficult to estimate. However, information–theoretic approaches have been proposed to avoid the difficulty of estimating input parameters. These algorithms regard the clustering as a data compression problem by incorporating the *Minimum Description Length* (MDL)–principle. The cluster model of these algorithms comprises joint coding schemes supporting numerical and categorical data. The MDL–principle allows us to balance model complexity and goodness–of–fit. In the following, we elaborate on this principle.

Minimum Description Length Principle

MDL [7] is a well–known model selection approach to evaluate various models and find the most accurate one considering the minimum description length criterion. MDL–principle regards the model selection challenge to a data compression problem in the sense that more accurate models lead to less compression cost. More precisely, let $\mathscr{M}$ denote a set of various candidate models representing the data. Following the two–part MDL [7], the best fitting model $M \in \mathscr{M}$ is the one which minimizes

$$DL(D, M) = DL(D|M) + DL(M) \tag{2.1}$$

where $DL(D|M)$ concerns the description length of the data set D encoded by means of the model M and $DL(M)$ represents the model complexity, i.e. cost of encoding the model itself. In MDL–principle, we incorporate the model complexity to avoid any over–fitting caused

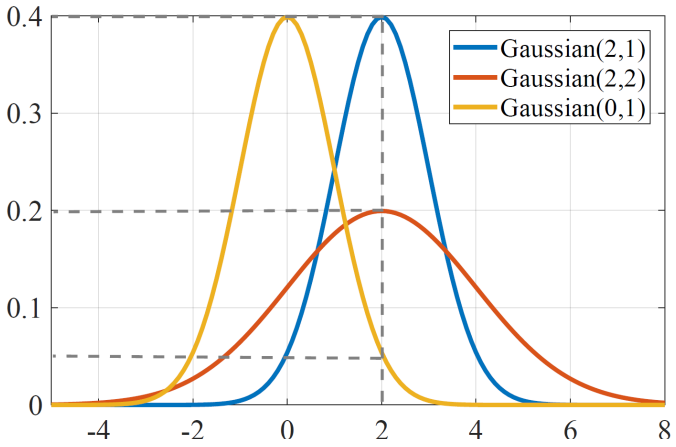


Figure 2.1: Various accurate and inaccurate fitted PDFs for a synthetic data x generated by a $Gaussian(2, 1)$ model.

by too complicated models. Therefore, we encode not only the data but also the model used in the coding process.

We consider $DL(D, M)$ as a model selection indicator. That is, employing a coding scheme, the number of bits required to encode the data indicates the accuracy of the model used in the coding process. According to the Shannon coding theorem [37], the ideal code length is related to the likelihood and is bounded by the entropy. More precisely, for an outcome a the number of bits required for coding is defined by $\log_2 \frac{1}{P(a)}$, where $P(.)$ shows the probability of a with the assumption that $\lim_{P(a)\to 0^+} P(a) \log_2(P(a)) = 0$. This coding scheme is also known as *log loss*. As a consequence, we assign shorter bit strings to outcomes with higher probability and longer bit strings to outcomes with lower probability.

To elaborate the concept, assume a synthetically generated data following Gaussian distribution, i.e. $x \sim Gaussian(2, 1)$. Figure 2.1 shows the *probability density function* (PDF) w.r.t. the true model ($G_1 := Gaussian(2, 1)$, the blue line) and two other PDFs corresponding to models with the lower accuracy, i.e. $G_2 := Gaussian(2, 2)$ (the red line) and $G_3 := Gaussian(0, 1)$ (the orange line). Applying Shannon's theorem, we compute the compression cost of the outcome $a = 2$ w.r.t. three models as follows:

$$
\begin{aligned}
-\log_2 PDF_{G_1}(2) &= -\log_2(0.4) = 1.32 \\
-\log_2 PDF_{G_2}(2) &= -\log_2(0.2) = 2.32 \\
-\log_2 PDF_{G_3}(2) &= -\log_2(0.05) = 4.32
\end{aligned}
$$

Thus, the compression cost is in an inverse relationship with the likelihood of an outcome. The better the model fits the data (G_2 in our example), the more likely the observations are, and hence the lower the compression cost is. Moreover, PDF is a relative likelihood for every outcome and it is not necessarily less than or equal to 1. Thus, in order to avoid the negative

number of bits caused by $PDF(.) > 1$, we consider a resolution parameter γ in the sense that the coding cost is $-\log_2 PDF(a) \cdot \gamma$. The parameter γ is a constant real number ensuring that the coding cost is always positive. Therefore, specifying γ is straightforward and needs to be set in the way that $\forall a \in D$, $PDF(a).\gamma \leq 1$. Then, by setting $\gamma \geq \max_{\forall a \in D} PDF(a)$ we make sure that the coding cost is always positive.

2.2 Anomaly Detection

Anomaly (referred to as outliers, noise, deviations or exceptions) detection is a mining task where the major task is to identify rare items, events, or observations that differ significantly from the majority of the data. Anomalies can be some kind of problem such as bank fraud, a structural defect, medical problems, or errors in a text. Recently, there is a significant interest in anomaly detection among time series in the data mining community. In this respect, we introduce regression models and regularization techniques that are well–known to model time series. Moreover, Granger causality is investigated as one of the useful approaches to capture temporal dependencies among time series which can be helpful to detect dependency anomalies.

2.2.1 Linear regression

Let $y^{1:n} = \{y^1, \ldots, y^n\}$ denote the response (dependent) time series of length n and $\{x_1^{1:n}, \ldots, x_p^{1:n}\}$ be the information set, i.e. the set of all observations w.r.t. regressors $x_1, \ldots, x_p$.

- **Simple linear regression:** The linear model for regression is the most simple regression model which involves a linear combination of the input variables, i.e. at any time point $t, t = 1, \ldots, n$, the response variable y^t is defined as:

$$y^t = x_1^t.\beta_1 + \cdots + x_p^t.\beta_p + \epsilon^t = \sum_{j=1}^{p} x_j^t.\beta_j = \boldsymbol{x^t}.\boldsymbol{\beta} + \epsilon^t \tag{2.2}$$

  where $\boldsymbol{x^t} = (x_1^t, \ldots, x_p^t)$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ are regressor and coefficient vectors, respectively. ϵ^t, called error (noise) variable, denotes a random variable that adds noise to the linear relationship between the response variable and regressors. In a matrix formulation, one can stack all n equations together as:

$$\boldsymbol{y} = \boldsymbol{X}.\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{2.3}$$

  where $\boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$, $\boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$ and $\boldsymbol{X} = \begin{pmatrix} x_1^1 & \ldots & x_p^1 \\ \vdots & & \vdots \\ x_1^n & \ldots & x_p^n \end{pmatrix}$ is the information matrix. Most of the time, the error vector $\boldsymbol{\epsilon}$ is assumed to be the withe noise, i.e. following a Gaussian distribution with mean value 0 and standard deviation 1, in a linear regression.

- **Autoregression:** Usually, an autoregressive model (AR) is used to describe time series processes in nature, economics, etc. It specifies that the output variable depends linearly on its own lagged values in a time series. More precisely, the autoregressive model of order d for a time series $x^{1:n}$ at time point t is defined as:

$$x^t = x^{t-d} \cdot \beta_d + ... + x^{t-1} \cdot \beta_1 + \epsilon^t = \sum_{i=1}^{d} x^{t-i}.\beta_i + \epsilon^t \tag{2.4}$$

  Moreover, AR model can be considered as a special case of linear regression when the regressors are lagged observations of the response variable.

- **Vector Autoregression:** A vector autoregressive (VAR) model is an extended version of AR model when more than one time series are involved in the model. Essentially, it captures the linear interdependencies among multiple time series. Let $x_1, \ldots, x_p$ denote p time series of the length n. Thus, a VAR model of order d is defined as:

$$\boldsymbol{x^t} = \boldsymbol{x^{t-1}}.A_1 + \cdots + \boldsymbol{x^{t-d}}.A_d + \boldsymbol{\epsilon^t} \tag{2.5}$$

  where A_i, $i = 1, \ldots, d$ is a $p \times p$ coefficient matrix w.r.t. $i - th$ equation.

2.2.2 Regularization

Ordinary least square (OLS) is a common approach to estimate linear regression coefficients. In this approach, we minimize the sum of the squares of the differences between the observed dependent variable (values of the variable being observed) in the given data set and those predicted by the linear function. However, this approach might lead to over–fitting while regularization seems a reasonable solution for it. In other words, since the optimization problem can be ill–posed, regularization by a penalty function provides an efficient and sparse solution leading to less complex models. More precisely, let the regression model (Equation 2.2) be given, the regularized optimization problem is as follows:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{t=1}^{n} (y^t - \sum_{j=1}^{p} x_j^t.\beta_j)^2 + \lambda R(\boldsymbol{\beta}) \tag{2.6}$$

where $R(.)$ is the penalty function and λ is the regularization parameter. Here, we introduce some of well–known regularization methods.

- **Lasso regression:** *Least Absolute Shrinkage and Selection Operator* (Lasso) [40] is one of the well–known regularization as well as variable selection approaches. In this approach, one adds L_1–norm (denoted by $||.||_1$) of coefficients as penalty term to the loss function when estimating parameters in a regression model. Thus, the optimization problem is defined as:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{t=1}^{n} (y^t - \sum_{j=1}^{p} x_j^t.\beta_j)^2 + \lambda ||\boldsymbol{\beta}||_1 \tag{2.7}$$

When λ is zero, Equation 2.7 is equivalent to OLS. Setting very large values for λ leads to under–fitting since Lasso shrinks the less important coefficients to zero. Therefore, Lasso is usually well–known for feature (variable) selection in the case, a huge number of features (variables) is given. However, one of the limitations of Lasso is that if $p >> n$, Lasso selects at most n features.

- **Adaptive Lasso:** As a variant of Lasso, adaptive Lasso [46] assigns adaptive weights for penalizing the L_1–norm of the regression coefficients, i.e.

$$R(\boldsymbol{\beta}) := \sum_{j=1}^{p} w_j |\beta_j| \quad , \quad w_j = \frac{1}{|\hat{\beta}_j^{(mle)}|^{\omega}} \tag{2.8}$$

where w_j is the weight vector for some $\omega > 0$ and $\hat{\beta}_j^{(mle)}$ is the maximum likelihood estimate of the parameters. Adaptive Lasso is an appropriate variant of Lasso since its consistency as well as its oracle properties are proven [46]. Despite the efficiency of Lasso approach, the consistency of this approach is not ensured [1].

- **Ridge regression:** In this approach we add L_2–norm (denoted by $|| \cdot ||_2$) of coefficients as penalty term to the loss function, i.e.

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{t=1}^{n} (y^t - \sum_{j=1}^{p} x_j^t.\beta_j)^2 + \lambda ||\boldsymbol{\beta}||_2 \tag{2.9}$$

Similar to Lasso, if λ is zero Equation 2.9 is equivalent to OLS and a very large amount for λ leads to under–fitting resulted by adding too much weight. But unlike Lasso, ridge regression penalizes the coefficients if they are too far from zero enforcing them to be small in a continuous way instead of forcing them to be exactly zero. Thus, it decreases the model complexity while keeping all variables in the model.

- **ElasticNet:** Since variable selection with Lasso can be too dependent on data and thus unstable, ElasticNet, first, was introduced as a remedy for this issue. Essentially, in this approach, the solution is to combine the penalties of both lasso and ridge regression to get the best out of them. More precisely, ElasticNet is a convex combination of Ridge and Lasso when the optimization problem is defined as:

[1] i.e. the resulting sequence of estimates does not have to converge in probability to the optimal solution for variable selection under certain conditions (Section 2 in [46]).

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{t=1}^{n} (y^t - \sum_{j=1}^{p} x_j^t.\beta_j)^2 + \lambda_2 ||\boldsymbol{\beta}||_2 + \lambda_1 ||\boldsymbol{\beta}||_1 \tag{2.10}$$

where the λ_1 controls the sparseness of a model and λ_2 removes the limitation on the number of selected variables and stabilizes the regularization path.

2.2.3 Generalized Linear Model

Generalized Linear Model (GLM), introduced by Nelder and Baker in [31], is a natural extension of the linear regression to the case where time series can have any distribution from the exponential family. Therefore, the response variable is not anymore a simple linear combination of covariates but its mean value is related to the covariates by a *link function*. More precisely, let $\eta = \boldsymbol{X}.\boldsymbol{\beta}$ be a linear predictor for the random component y where $\boldsymbol{X}$ denotes the covariate (information) matrix and $\boldsymbol{\beta}$ is the coefficient vector. We assume, the distribution of y belongs to the exponential family and μ denotes its mean value, i.e. $\mu = E[y|\boldsymbol{X}]$. Thus, in a GLM framework the relation between these two components is not anymore linear but defined as:

$$\mu = g(\eta)$$

where g is the link function, a monotone twice differentiable function given by a user. Corresponding to every distribution, there is an appropriate canonical link function (e.g. $g = log(.)$ for Poisson and $g = \frac{1}{(.)}$ for Gamma distribution) [31]. Table 2.1 summarizes well–known distributions from exponential family providing the appropriate canonical link function w.r.t each distribution. GLM relaxes Gaussian assumptions about the involved processes and the error term. Therefore, the regression error does not necessarily follow a standard Gaussian distribution and it might have any distribution from the exponential family leading to more accurate models.

Distribution	*Link function*
Gaussian	$\mu = \boldsymbol{X}.\boldsymbol{\beta}$
Exponential/Gamma	$\mu = \frac{1}{\boldsymbol{X}.\boldsymbol{\beta}}$
Inverse Gaussian	$\mu = \frac{1}{\boldsymbol{X}.\boldsymbol{\beta}^2}$
Poisson/Countable	$\mu = exp(\boldsymbol{X}.\boldsymbol{\beta})$
Bernoulli/Bi(Multi)nomial	$\mu = \frac{exp(\boldsymbol{X}.\boldsymbol{\beta})}{1+exp(\boldsymbol{X}.\boldsymbol{\beta})}$

Table 2.1: Common link functions for various distributions where $\boldsymbol{X}$ is the covariates matrix, μ is the mean and $\boldsymbol{\beta}$ is the coefficient vector.

2.2.4 Granger Causality

Granger causality, introduced by Granger in the area of economics [25], is a well–known notion for causal inference among time series. Granger causality captures the temporal causal relations

among time series. However, it is not meant to be always equivalent to the true causality since the question of "true causality" is deeply philosophical. This notion of causality is defined based on two principles [21]:

- The cause happens prior to its effect;
- The cause has unique information about the future values of its effect.

The first assumption is intuitively acceptable since the past influences the future, not other way around. On the other hand, the second assumption sounds plausible as well in the sense that without considering the cause no information about the effect is available. Now, let $x^{1:n} = \{x^t | t = 1, \dots, n\}$ and $y^{1:n} = \{y^t | t = 1, \dots, n\}$ denote two stationary time series x and y up to time n, respectively. Moreover, let $\mathscr{I}(t)$ be all the information accumulated since time t and $\mathscr{I}_{\neg y}(t)$ denote all the information apart from the specified time series y up to time t. Now considering two above assumptions, Granger proposed the following definition for a causal effect [25]:

Definition 2.2.1. ***Granger Causality:*** *Given two time series x and y, y Granger–causes x if including previous values of y along with x improves the predictability of x, i.e.*

$$\mathscr{P}(x^t | \mathscr{I}_{\neg y}(t-1)) < \mathscr{P}(x^t | \mathscr{I}(t-1)) \tag{2.11}$$

where $\mathscr{P}$ denotes the predictability.

In another point of view, let Model 1 denote the *autoregressive* (AR) model of order d (the lag) corresponding to time series x. Moreover, let Model 2 denote the augmented AR model w.r.t. x including the lagged observations of x and y.

$$x^t = x^{t-d} \cdot \gamma_{t-d} + \dots + x^{t-1} \cdot \gamma_{t-1} + \epsilon^t \tag{Model 1}$$

$$\begin{aligned} x^t = {} & x^{t-d} \cdot \alpha_{t-d} + \dots + x^{t-1} \cdot \alpha_{t-1} \\ & + y^{t-d} \cdot \beta_{t-d} + \dots + y^{t-1} \cdot \beta_{t-1} + \epsilon^t \end{aligned} \tag{Model 2}$$

Thus, y Granger–causes x if the second model improves the accuracy when predicting x.

The concept of Granger causality is extendable to more than two time series. Let $x_1, x_2, \dots, x_p$ be p time series where $\forall i \in \{1, \dots, p\}, x_i = \{{x^t}_i | t = 1, \dots, n\}$. The VAR model of order d w.r.t. all the time series is defined as Model 3 in the following:

$$X^t = X^{t-d} \cdot B_{t-d} + \dots + X^{t-1} \cdot B_{t-1} + \epsilon^t \tag{Model 3}$$

where $X^t = ({x_1}^t, \dots, {x_p}^t)$ is the concatenated vector of all time series at time point t. In this model B_t is a $p \times p$ matrix of the regression coefficients where the i–th row corresponds

to the coefficients w.r.t. x_i at time t. Essentially, the matrix formulation is an abstract form to illustrate the temporal dependencies among all the time series.

Basic definition of the Granger causality has certain assumptions about the distribution of time series. More precisely, the processes are assumed to be Gaussian distributed time series in Model 1,2 and 3 and hence a linear model is considered overall. Moreover, in a linear model the error term (ϵ^t) is an additive Gaussian white noise with mean 0 and variance 1.

2.3 Evaluation Strategies

Let $P = \{P_1, \ldots, P_r\}$ denote the ground truth w.r.t. a data mining task, e.g. clustering or classification where the data set contains N data objects and $C = \{C_1, \ldots, C_k\}$ be the achieved result. Thus, for each pair of data objects x_i and x_j, there are four different cases:

- x_i and x_j belong to the same category of C and the same category of P
- x_i and x_j belong to the same category of C but different categories of P
- x_i and x_j belong to different categories of C but the same category of P
- x_i and x_j belong to different categories of C and different categories of P

Let a, b, c, d correspond to the number of pairs for the first to fourth cases and L is the total number of pairs, i.e. $L = a + b + c + d$.

- **Precision:** It is also called positive predictive value and is the fraction of relevant instances among the retrieved instances defined as:

$$Precision = \frac{a}{a+d} \tag{2.12}$$

  Moreover, considering adjacency matrices as outputs, let A^* and $\hat{A}$ denote the true and the output adjacency matrix, respectively. We distinguish between two entries in the adjacency matrix A, $A[i,j]$ and $A[j,i]$. Thus, the evaluation measures for time series $x_1, ..., x_p$ are defined as:

$$Precision = \frac{|\{(i,j) \in P : \hat{A}[i,j] = A^*[i,j]\}|}{|\{(i,j) \in P : \hat{A}[i,j] = 1\}|} \tag{2.13}$$

- **Recall:** It is also known as sensitivity and is the fraction of the total amount of relevant instances that were actually retrieved:

$$Recall = \frac{a}{a+b} \tag{2.14}$$

  In case of matrices, the recall is defined as:

$$Recall = \frac{|\{(i,j) \in P : \hat{A}[i,j] = A^*[i,j]\}|}{|\{(i,j) \in P : A^*[i,j] = 1\}|} \tag{2.15}$$

- **F–measure:** There is clearly a trade–off between precision and recall as the goal of prediction where F–measure tries to balance the overall quality of prediction.

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.16}$$

- **Rand Index:** Rand index (RI) is one of the most popular external clustering validation indices which is a measure of the similarity between two clustering results.

$$RI = \frac{a + d}{L} \tag{2.17}$$

- **Categorical Utility:** In order to evaluate clustering results in terms of categorical attributes we apply *categorical utility* (CU) criterion. CU attempts to maximize both the probability that two patterns in the same cluster have attribute values in common and the probability that patterns from different clusters have different values:

$$CU = \sum_{k} (\frac{C_k}{\mathscr{D}\mathscr{B}} \sum_{A \in \mathscr{A}} \sum_{j} [P(A = A_j|C_k)^2 - P(A = A_j)^2]) \tag{2.18}$$

  where $P(A = A_j|C_k)$ is the conditional probability that a categorical attribute A has the value A_j given cluster C_k, and $P(A = A_j)$ is the overall probability of attribute A having the value A_j in the entire data set. Obviously, the higher the CU value, the better the clustering performs.

- **Normalized Mutual Information:** *Normalized mutual information* (NMI) [41] is an information–theoretic evaluation measure for clustering results. NMI numerically evaluates pairwise mutual information between ground truth and resulted clusters and continues normalizing by means of the entropy of either original or resulted clusters. NMI scales between zero and one representing a random and a perfect clustering, respectively.

  Let $H(P)$ and $H(C)$ denote the entropy of P and C, respectively, defined as:

$$H(P) = -\sum_{i=1}^{r} p(P_i). \log(p(P_i)) \tag{2.19}$$

  where $p(P_i)$ shows the probability of the category P_i. Moreover, let $I(P, C)$, the mutual information of P and C, i.e. the amount of information they have in common, be defines as:

$$I(P, C) = \sum_{i=1}^{r} \sum_{j=1}^{k} p(P_i \cap C_j). \log(\frac{p(P_i \cap C_j)}{p(P_i).p(C_j)}) \tag{2.20}$$

  Thus, NMI is defines as follows:

$$NMI(P, C) = \frac{I(P, C)}{\sqrt{H(P).H(C)}} \tag{2.21}$$

CHAPTER 3

Problem Statement and Research Challenges

3.1 Problem Specification

Essentially, our focus in this thesis is mining heterogeneous data sets and facing challenges when analyzing such data. Heterogeneity could mean different when considering various domains. Thus, we distinguish between mixed–type and single–type heterogeneous data sets in this thesis.

A mixed–type heterogeneous data set consists of attributes from different data types. As an example, dealing with statistical surveys, heterogeneous data could consist of categorical attributes(e.g. marital status) and numerical attributes(e.g. the amount of income). To elaborate on the issue, let us consider the following mixed–type data consisting of three different clusters illustrated by different shapes (rectangle, circle, cross) in Figure 3.1. The data set comprises two numerical attributes concerning the position of data objects in a 2D space and a categorical

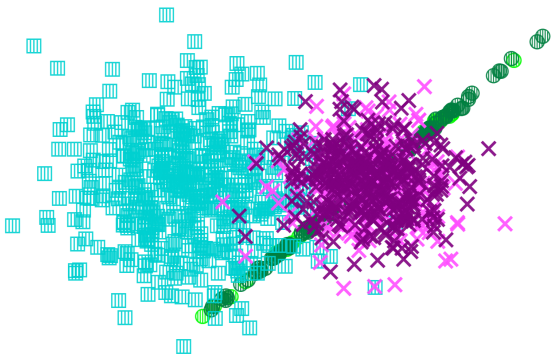


Figure 3.1: A synthetic example for a mixed–type heterogeneous data.

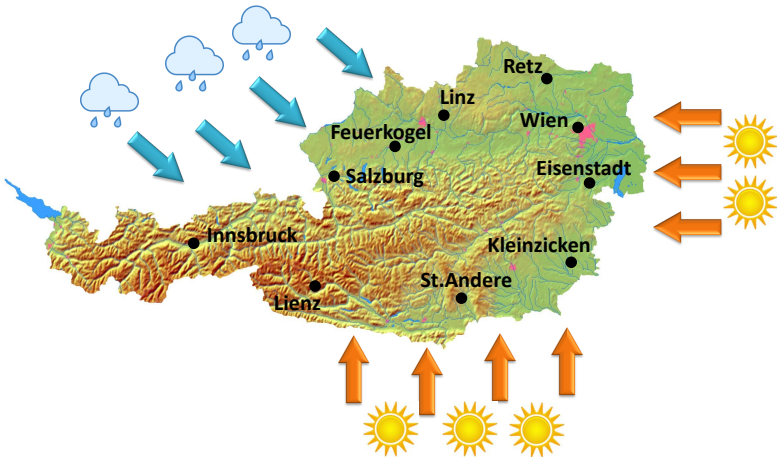


Figure 3.2: Selected stations of meteorological measurements in Austria.

attribute representing the color of data points (rose, purple, light green, dark green and cyan). Therefore, a data object in this data set looks like, for example, (1, 2, purple).

On the other side, single–type heterogeneous data comprise a specific data type, e.g. numerical time series, where attributes might be of different natures. An example of this category is a mixture of Poisson distributed time series (e.g. measuring the number of sunny days) and Gaussian distributed time series (e.g. measuring the amount of precipitation) when observing the weather characteristics in different stations in a climatological data set is (Figure 3.2).

In the following, we address some of the challenges one needs to face when mining heterogeneous data sets.

3.2 Research Challenges

Among the top 10 challenging problems in data mining, identified by [44], there are three challenges related to complex data indicating the importance of the issue;

- **Mining complex knowledge from complex data;**
- **Developing a unifying theory of data mining;**
- **Mining sequence data and time series data;**

The first challenge is related to mining complex data and finding interesting patterns where various characteristics of every attribute are preserved. In order to elaborate the issue, we consider the generated mixed–type heterogeneous data set illustrated in Figure 3.1. As mentioned, when mining such data, one of the basic and straightforward approaches is to homogenize the data as much as it is possible. This might be achieved by converting a data type to another one. That is, we simply convert the categorical attribute *Color* to a numerical attribute by mapping numbers to various colors, e.g. cyan=1, light green =2, rose=3, and so on. Employing NMI

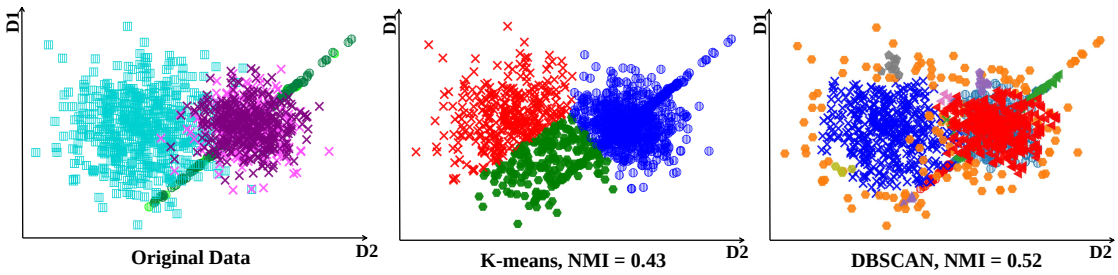


Figure 3.3: Clustering results after converting categorical attribute *Color* to a numerical one.

[41] as an evaluation measure, we apply two popular clustering algorithms, K–means [29] and DBSCAN [22] on the homogenized data set, to find interesting patterns in this example. These clustering algorithms are essentially designed for pure numerical data sets when distance measures play a key role. Figure 3.3 shows the low performance of applying them on the converted data when maximum NMI (achieved by DBSCAN) is 0.52. As a justification, the distance between various colors is artificially defined after a data type conversion and it is not meaningful anymore. Therefore, it might disturb clustering algorithms to find the correct clusters. Thus, our first research question is as follows;

Question 1
How can the effect of artificially defined relationships caused by a simple conversion of data types be avoided when mining heterogeneous data sets?

Although the topic of clustering mixed–type data represented by numerical and categorical attributes attracted attention, e.g. CFIKP [45], CAVE [27], CEBMDC [26], most of the algorithms are designed based on the algorithmic paradigm of k–means, e.g. k–Prototypes [28], SpectralCAT [19], and CoupledMC [42]. Often in this category, not only the number of clusters k has to be specified by a user, but also the weighting between numerical and categorical attributes in clustering. Among them, K–means–mixed (KMM) [2] avoids weighting parameters by an optimization scheme learning the relative importance of the single attributes during runtime. However, it still needs the number of clusters k as an input parameter.

Model–based clustering algorithms have been also proposed for mixed–type data by incorporating a mixture of Gaussian distributions. In between, clustMD [30] is developed using a latent variable model and employing an expectation maximization (EM) algorithm to estimate the mixture model. Yet, this algorithm has certain Gaussian assumptions that do not have to be necessarily fulfilled. On the other hand, clustering algorithms designed for mixed–type data often do not properly model dependencies and are limited to modeling meta–Gaussian distributions. Copulas, that provide a modular parameterization of joint distributions, can model a variety of dependencies but their use with discrete data remains limited due to challenges in parameter inference. Authors in [34] use Gaussian mixture copulas to model complex dependencies beyond those captured by meta–Gaussian distributions for clustering. However,

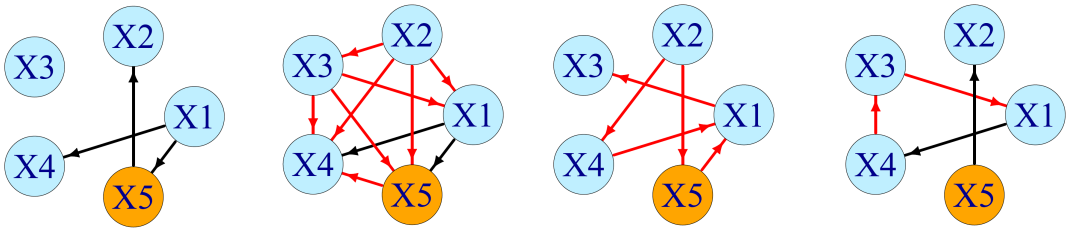


Figure 3.4: Synthetic heterogeneous example. Results of applying existing Granger causal inference algorithms designed for homogeneous data sets on heterogeneous data. Red edges show the wrongly detected causal relations and black edges show the correct causal directions.

this approach may not only result in information loss but also fail to capture the discriminative information between objects.

Thus, most clustering algorithms, although designed for mixed–type data, require a user to specify parameters which are not straightforward to be set. Therefore, our next research question arises as follows;

Question 2
How can the data be analyzed without a user having to specify some parameters, i.e. parameter–free data analysis?

The second challenge implies that most data mining algorithms are "ad–hoc". Many techniques are either designed for a specific data type (e.g. pure numerical data) or consider individual cases, such as clustering or anomaly detection. But there is no unifying framework. In another point of view, most algorithms avoid spurious correlations which are sometimes related to the problem of mining for "deep knowledge", e.g. the hidden causes for many observations [44]. As an example, there might be a strong correlation between the number of sunny days in "Eisenstadt" in Figure 3.2 and the amount of precipitation in "Wien" when investigating the climatological measurements in Austria. Considering such information might help to improve the accuracy of data mining algorithms. But the term "deep knowledge" might have different interpretations. Here our focus is on the discovery of causal networks from observational data, where no certain information about their distribution is provided. This is a fundamental problem with many applications in science. Thus, the next question appears in this regard;

Questions 3
How is it possible to increase the accuracy through "deep knowledge" when mining heterogeneous data?

Regarding the third challenge, mining sequence data (and time series) is challenging since they are non–i.i.d and there is usually a strong correlation among various observations. This case gets even more complicated when time series are of different natures (heterogeneous single–type data). Despite the efficiency of homogeneous algorithms designed for causal inference on time series, they lead to information loss and inaccuracy when applying on heterogeneous data. As a reason, homogenizing the data which in this case means transforming a time series to another one with a specific distribution, leads to inaccuracy. On the other side, applying an algorithm designed for homogeneous data sets on heterogeneous data does not guarantee high performance. To elaborate, we generated a heterogeneous data set consisting of 4 Poisson (blue circles in Figure 3.4) and a Gamma (orange circle) distributed time series and applied some well–known algorithms designed for Granger causal inference on homogeneous data sets, i.e. GT [25] (short for Granger test), CUTE [17], TCML [5]. As it is explicitly clear in Figure 3.4, none of them perform effectively on this data set in terms of *F–measure*. GT assumes a Gaussian distribution and hence a linear relation among time series which leads to inefficiency. On the other hand, CUTE needs to binarise time series as it is designed for event sequences where Bernoulli distributed time series are assumed. It is already well–understood that discretization and specially binarising the data decreases the accuracy since the distribution of the time series is not preserved. Thus, our third question shows up;

Question 4
Is it possible to avoid information loss caused by specific assumptions when mining heterogeneous data?

CHAPTER 4

Contributions and Research Results

In this chapter, we elaborate on the different steps of our research approach in Section 4.1. Afterward, various contributions achieved in the course of this thesis are introduced in Section 4.2. Finally, we address research results containing scientific papers in Section 4.3.

4.1 Research Approach

Our general research approach in this thesis comprises several steps. First, we specify the problem and state the motivation which is essentially avoiding drawbacks of already existing approaches. In the next step, we address objectives of a possible efficient and effective solution. A comprehensive solution tries to move towards a parameter–free data analysis when fewer number of parameters is preferred. After specifying characteristics of an effective solution, we start a loop of iterative design and evaluation. That is, we first propose an algorithm, considering objectives of the problem and the solution. Then, we evaluate the algorithm in various aspects by conducting several experiments on synthetic and real–world data sets. If the evaluation does not seem satisfactory, we go a step backward and modify the design as long as it leads to more convincing results. Finally, when the proposed algorithm sounds promising in various aspects, e.g. properties of the problem and the solution, design, and evaluation, we try to publish the paper in outstanding conferences and journals. Figure 4.1 and 4.2 show different steps of our research approach taken in this thesis in detail.

4.2 Contributions

Generally speaking, our main contribution is to avoid drawbacks of a data type conversion as well as the inaccuracy caused by specific assumptions when mining heterogeneous data sets. In this respect, we aim at preserving the original data and utilizing useful characteristics of every data type. In particular, we focus on the clustering of mixed–type heterogeneous data where a mixture of categorical and numerical attributes is given. As already mentioned,

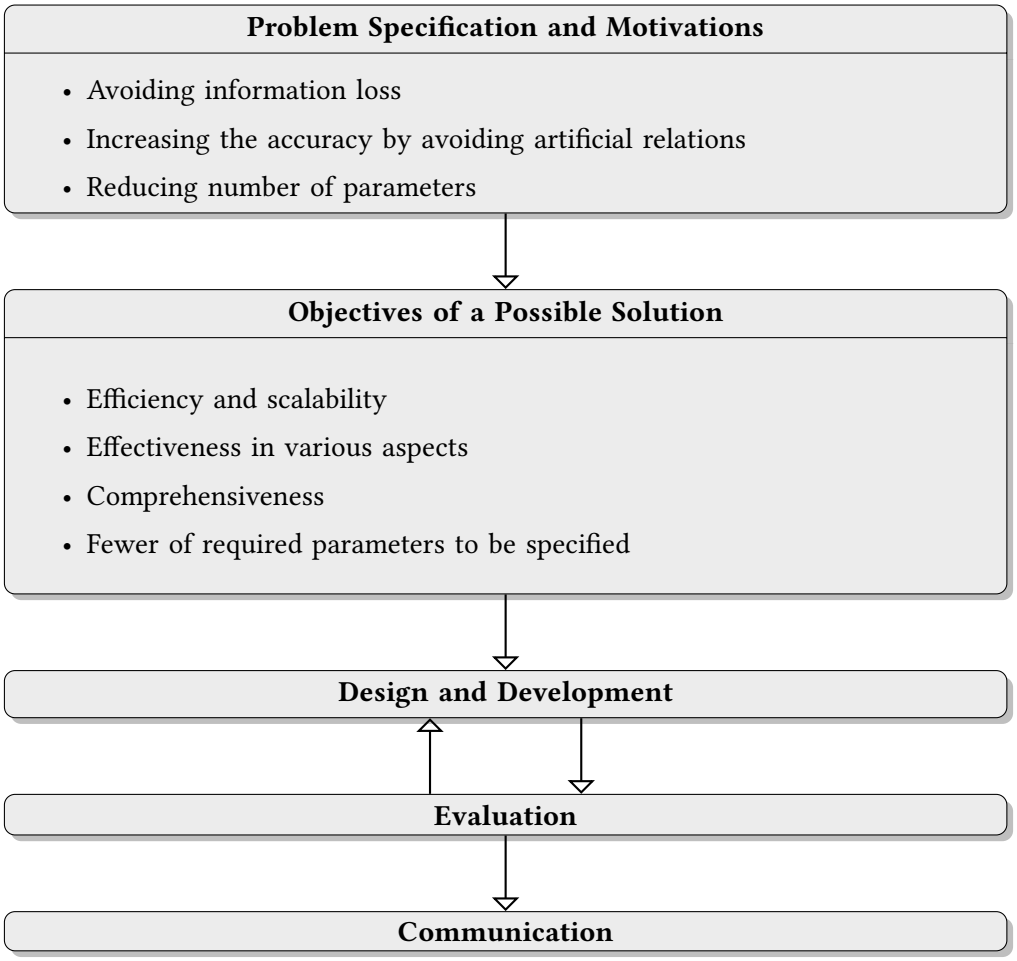


Figure 4.1: First two steps of our research approach in this thesis.

most existing clustering algorithms are designed for pure numerical attributes and applying them to a mixed–type data leads to inaccuracy and information loss. Thus, we try to avoid a data conversion and preserve heterogeneity of data by employing *Concept Hierarchies*. An interesting characteristic of categorical data that one could easily utilize is the natural hierarchy among various categories. To elaborate, let Figure 4.3a show the introduced mixed–type data set (Section 3.1) where two numerical attributes (D_1 and D_2) show the position of every data point in a 2D space and the third attribute shows its color. This data set consists of three different clusters illustrated in Figure 4.3 by different shapes (rectangle, circle, cross). Considering the standard scalable range of colors, one can categorize different colors as illustrated in Figure 4.3b when frequency of every color is assigned to its corresponding node. During clustering of such data, one might find the concept "pink" more representative for a detected cluster consisting of points with colors purple and rose. We utilize such conceptual hierarchies to summarize

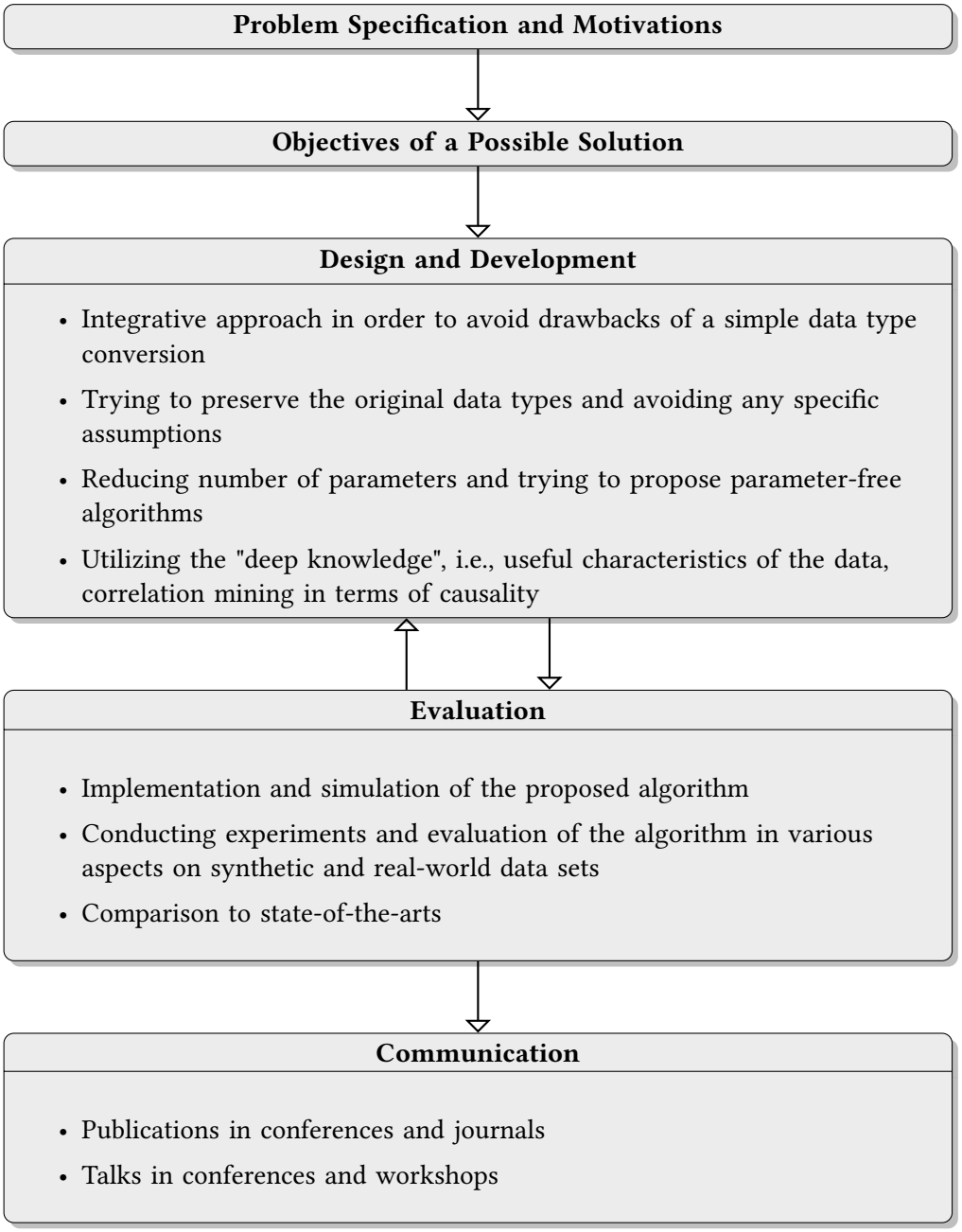


Figure 4.2: Last three steps of our research approach in this thesis.

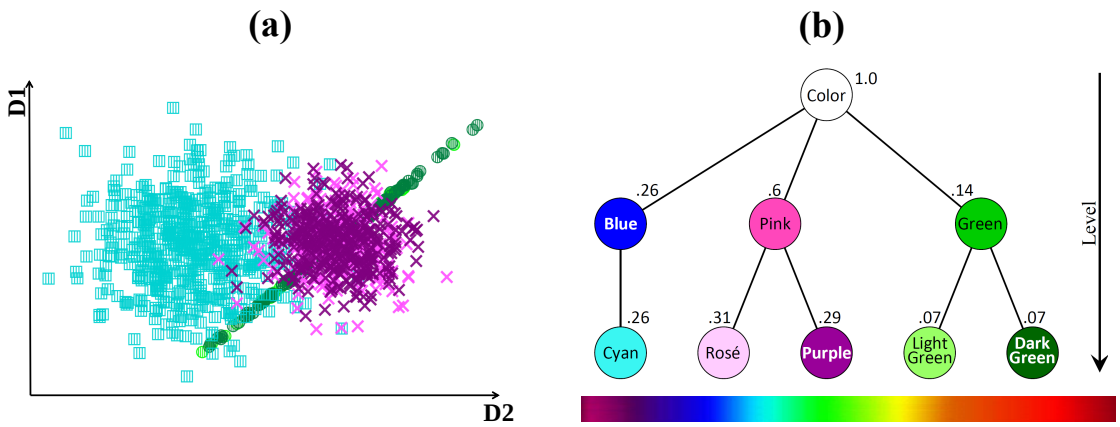


Figure 4.3: a) Synthetically generated mixed–type data, b) A natural hierarchy between colors based on scalable range of colors.

categorical attributes and introduce a model to represent the data when for solely numerical data sets approximating data with a *Probability Distribution Function* (PDF) is well–accepted.

In another point of view, a concept hierarchy provides a meaningful distance measure among various concepts. Dark green and light green, for instance, are more similar compared to purple according to the scalable range of colors (Figure 4.3b). It is also confirmed by the corresponding concept hierarchy since they belong to the same branch of the tree. Most classic clustering algorithms, e.g. DBSCAN, are designed based on a distance metric. Therefore, it sounds reasonable to apply these algorithms on heterogeneous data sets when a meaningful distance measure for both categorical and numerical attributes is considered. That is, we incorporate a unified distance measure for numerical and categorical attributes based on the concept hierarchy. Thus, it brings us to our first contribution as follows;

Contribution 1
Utilizing specific characteristics of every data type in order to preserve original data and avoid the effect of artificially defined relations caused by a simple conversion of data types.

On the other hand, many data mining approaches consider specific assumptions that do not have to be true necessarily. For instance, many algorithms assume a Gaussian distribution and a linear model when dealing with time series. However, there are many processes of a non–Gaussian nature(e.g. Poisson distributed time series) and many applications generating a mixture of time series having different distributions. We already demonstrated various challenges dealing with such data in Section 3.2. Essentially, we try to ***integrate*** data of different natures as much as possible to avoid any information loss. In particular, let us focus on finding causal dependencies between time series in a heterogeneous data set where time series of differ-

ent distributions are considered. Moreover, let Granger causality be the causal notion employed to investigate the existing interactions. As already mentioned in the background (Chapter 2), a basic definition of Granger causality has certain assumptions about the distribution of time series. More precisely, the processes are assumed to be Gaussian distributed time series and hence a linear model is considered overall. Moreover, in a linear model, the error term (ϵ^t) is an additive Gaussian white noise with mean 0 and variance 1 (Model 1 and Model 2 in Section 2.2.4). However, these assumptions are not necessarily true in most applications. Thus, it is crucial to generalize the linear models to the non–linear cases in the sense that we include time series from various distributions and avoid any information loss caused by forcing Gaussian assumptions. Therefore, we employ *Generalized Linear Models* (GLMs) to extend the notion of Granger causality and introduce an integrative framework for causal inference on heterogeneous time series data regardless of their distributions. GLMs allow us to generalize simple autoregressive models to the case where several processes of different distributions from the exponential family are non–linearly related. Altogether, our next contribution is as follows;

Contribution 2
Integrate data of different natures as much as possible to avoid any information loss.

Regarding the third research question and data mining challenges, we are interested in improving the accuracy when mining heterogeneous data utilizing "deep knowledge". One could interpret the term "deep knowledge" in different ways. When mining time series, for instance, one could be interested in discovery of anomalies or outliers. However, classifying multivariate time series data, there are two types of anomalies:

- univariate anomaly: anomalies occur only within individual time series,
- dependency anomaly: anomalies occur due to changes of temporal dependencies among various time series.

Dependency anomalies, are more challenging to detect due to complex temporal structures and interactions among time series. In this regard, the discovery of causal relations among different processes leads to characterize the evolution in time of regular observations. The regular pattern can be used to detect deviated observations (i.e. outliers) in anomaly detection. That is, we incorporate the so–called "deep knowledge" when detecting anomalies in the sense that "deep knowledge" is interpreted as information about causal interactions among time series. Back to the climatological example (introduced in Figure 3.2), now an interesting question would arise: Utilizing the existing temporal dependency between stations could we find any anomalies in terms of the precipitation in a specific station (e.g "Wien") when we have measurements for different stations?

In another point of view, one could interpret incorporating "deep knowledge" as utilizing concept hierarchies to summarize all the information w.r.t. categorical attributes and improving clustering algorithms when dealing with heterogeneous data sets. Therefore, our third contribution is as follows;

Contribution 3
Employing useful characteristics of the data as well as incorporating spurious correlations to improve the accuracy when mining heterogeneous data.

On the other side, most data mining algorithms require a user to specify several parameters. Nevertheless, it is usually non–trivial to find the most appropriate parameter setting. To face this issue, parameter–free algorithms are introduced to make this process automatic. Among various approaches in this regard, we incorporate an effective model selection approach, i.e. Minimum Description Length (MDL) [7] which evaluates various models and find the most accurate one according to the minimum description length criterion. MDL–principle regards the task of model selection to a data compression problem in the sense that more accurate models lead to less compression cost. The better the model fits the major characteristics of the data, the better the result is. Following the MDL–principle, we encode not only the data but also the model itself and minimize the overall description length. Simultaneously, we avoid over–fitting since the MDL tends to a natural trade–off between model complexity and the goodness–of–fit. In the context of clustering, MDL can be employed as a clustering criterion as well as a model selection approach, i.e. an approach to make clustering parameter–free.

Contribution 4
Incorporating the MDL–principle for a parameter–free data mining.

4.3 Research results

Considering the aforementioned challenges and research questions, we focus on the clustering of heterogeneous data sets in this thesis where a parameter–free approach is preferred. Moreover, we address the anomaly detection of such data as a post–processing phase in data mining. First, we list various papers that either have been already published in scientific conferences, workshops, and journals or are currently under review. Then, we elaborate on the way each paper contributes in this thesis to cope the aforementioned challenges.

4.3.1 Publication Overview

- **Paper A:** Sahar Behzadi, Nikola Müller, Claudia Plant, and Christian Böhm. “Clustering of Mixed-type Data Considering Concept Hierarchies”. In: *The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. 2019.

- **Paper B:** Sahar Behzadi, Nikola Müller, Claudia Plant, and Christian Böhm. “Clustering of Mixed-type Data Considering Concept Hierarchies: problem specification and algorithm”.

In: *International Journal of Data Science and Analytics*. 2020.

- **Paper C:** Sahar Behzadi, M. A. Ibrahim, and Claudia Plant. "Parameter Free Mixed-Type Density-Based Clustering". In: *International Conference on Database and Expert Systems Applications (DEXA)*. 2018.

- **Paper D:** Sahar Behzadi, Hermann Hinterhauser, and Claudia Plant. "ITGC: Information-theoretic grid-based clustering". In: *International Conference on Extending Database Technology (EDBT)*. 2019.

- **Paper E:** Sahar Behzadi, Kateřina Hlaváčková-Schindler, and Claudia Plant. "Dependency anomaly detection for heterogeneous time series: A Granger-Lasso approach". In: *IEEE International Conference on Data Mining (ICDM) workshops*. 2017.

- **Paper F:** Sahar Behzadi, Kateřina Hlaváčková-Schindler, and Claudia Plant. "Granger Causality for Heterogeneous Processes". In: *The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. 2019.

- **Paper G:** Sahar Behzadi, Niklas Preschern, Kateřina Hlaváčková-Schindler, and Claudia Plant. "Anomaly Detection in Heterogeneous Time Series by Causality Mining". In: *Knowledge and Information Systems, submitted for publishing*. 2020.

- **Paper H:** Sahar Behzadi, Benjamin Schelling, and Claudia Plant. "ITGH: Information-theoretic Granger Causal Inference on Heterogeneous Data". In: *The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. 2020.

- **Paper I:** Sahar Behzadi, Benjamin Schelling, and Claudia Plant. "Information-theoretic Granger Causal Inference on Heterogeneous Data: Problem specification and algorithm". In: *International Journal of Data Science and Analytics, submitted for publishing*. 2020.

4.3.2 Discussion of Results

There are a wide variety of heterogeneous data sets depending on various applications. In this thesis, we distinguish between mixed–type and single–type heterogeneous data sets as already explained in Section 3.1. Figure 4.4 summarizes our publications w.r.t. various approaches we have taken and different tasks addressed in each paper.

Considering the task of clustering, we focus on mixed–type data sets where a mixture of categorical and numerical attributes is given. Essentially, our approach is to avoid data type conversion when clustering a mixed–type data set. That is, we aim at preserving the original characteristics of data and integrate different data types where a non–parametric

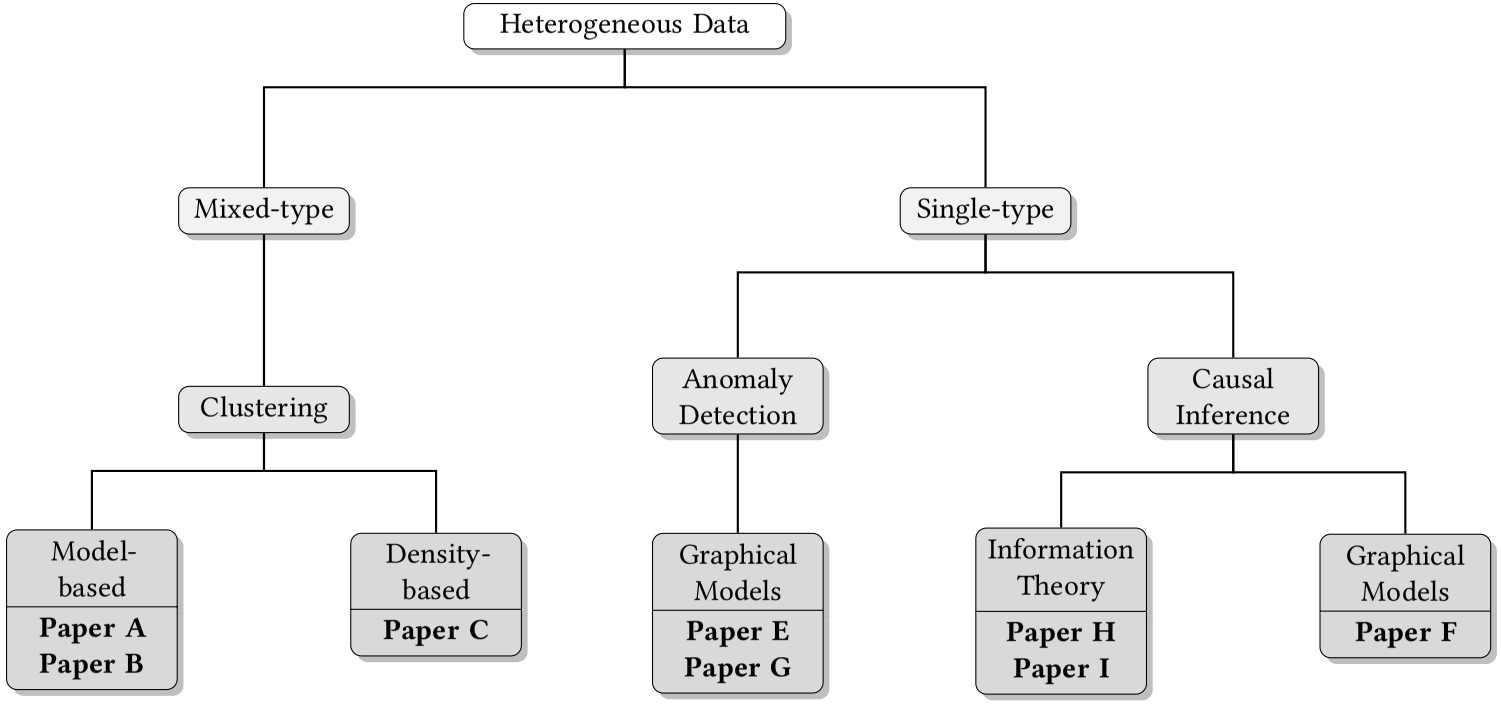


Figure 4.4: Overview of publications with respect to their contributions.

approach is preferred. Thus, we first investigate how to incorporate the MDL–principle as a clustering criterion in Paper D [8] when dealing with pure numerical attributes. In this paper, we proposed a parameter–free **I**nformation–**T**heoretic **G**rid–based **C**lustering (ITGC) algorithm utilizing MDL. That is, we regard the clustering task as a data compression problem such that the best clustering is linked to the strongest data compression. First, an adaptive grid is constructed corresponding to the statistical characteristics of any data set and non–empty cells are considered as single clusters. Then, we combine the concept of density and grid–based methods, and employing our MDL–based objective function, we start merging clusters with their neighbor grid cells only if it pays off in terms of the compression cost. In this paper, we address Contribution 4 although the main focus is not on heterogeneous data sets. That is why we do not include Paper D in Figure 4.4.

Later, we employ our experiences after Paper D for a parameter–free clustering but considering mixed–type data sets in Paper A [12] and Paper B [13]. Here, we again incorporate the MDL–principle as a clustering criterion. MDL allows integrative clustering by relating the concepts of likelihood and data compression while for any attribute a representative model is required. For solely numerical data sets a PDF represents an approximation of data. For categorical attributes, we incorporate concept hierarchies among various categories to summarize the categorical information. Beyond the clustering approaches, detecting the most relevant attributes during this process improves the quality of clustering. However, considering a data set with an unknown distribution where only a few attributes in the data space are relevant to characterize a cluster, it is not trivial to recognize the cluster–specific attributes. Thus, in Paper A, we introduce a parameter–free **CL**ustering algorithm for mixed–type data **I**ncluding **CO**ncept **T**rees, shortly *ClicoT* which ensures that only the truly relevant attributes are marked as cluster–specific attributes. The compression–based objective function employed by ClicoT avoids over–fitting, enhance the interpretability and guarantee the validity of the result. Hence, we address Contribution 1 and 4 introducing ClicoT which is a model–based clustering algorithm. Paper B is an extended version of Paper A where we investigate more aspects of ClicoT.

Another application of concept hierarchies is to employ them as a meaningful distance measure for both categorical and numerical attributes when dealing with mixed–type data sets. Back to the synthetic mixed–type example introduced in Section 4.2, Figure 4.5a shows the corresponding distance hierarchy to the categorical attribute *Color* while labels are related to the weights. In this example, we assume the same weight for all the links, nevertheless, one could assign different weights having more information about the data. To compute the distance between categorical values, we utilize the distance hierarchy. In this example, for instance, Rose and Purple are more similar than Rose and Cyan according to the corresponding distance hierarchy. It is also confirmed by the nature of colors since Rose and Purple are derivations of Pink. Employing the same structure, Figure 4.5b depicts a distance hierarchy corresponding to a numerical attribute. It has only two nodes (i.e. maximum and minimum in the corresponding range of a numerical attribute) and returns the Euclidean distance as the distance between two values.

Benefiting from our experience with distance hierarchies, in Paper C [11], we address Contribution 1 and introduce a general framework appropriate for clustering algorithms that

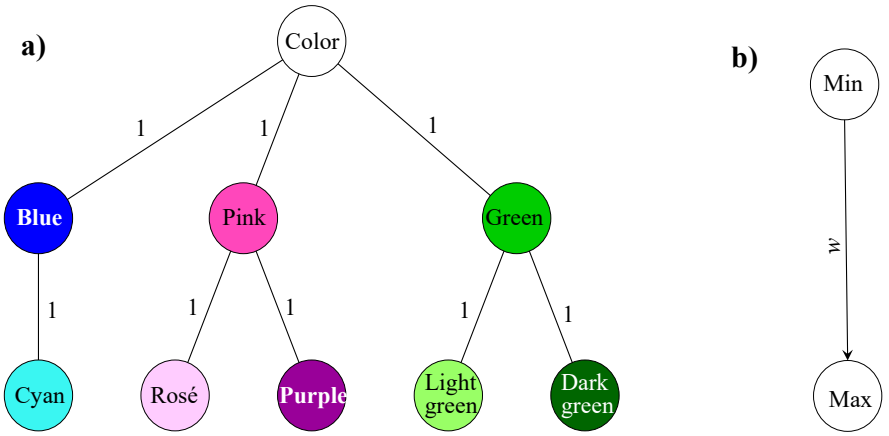


Figure 4.5: a) Distance hierarchy w.r.t. the categorical attribute *Color*, b) Distance hierarchy for a numerical attribute.

need a distance measure as one of the prerequisites. Thus, many well–known clustering algorithms, e.g. DBSCAN [22], could be applied to mixed–type data sets. Here, we focus on DBSCAN as one of the most effective representatives for the density–based clustering approach which captures dense groups of objects as clusters. DBSCAN requires two parameters, a positive real number ϵ and a natural number *MinPts* showing the radius and the density of a neighborhood, respectively. Although DBSCAN is well–known due to its performance, setting appropriate parameters is challenging. To face this challenge, we propose a parameter–free algorithm incorporating the MDL–principle. That is, we fix *MinPts* and try a range of various radius. We apply DBSCAN for every parameter setting and find the best parameters in terms of the compression cost. Therefore, in Paper C, contribution 4 is presented as well.

We concentrate on anomaly detection of single–type heterogeneous data sets for the rest of the publications. Especially, dependency anomalies in time series are of interest in this thesis that are caused by changes in temporal dependencies and causal relations. Therefore, we, first, focus on causal inference on heterogeneous data sets and investigate causal interactions to capture possible temporal dependencies between time series of different distributions. Among several notions of causality, we incorporate Granger causality [25] which is a popular method for causal inference in time series due to its computational simplicity. Essentially, it states that a cause improves the predictability of its effect in the future. Hence, various methods for causal inference differ in the way how they measure the predictability.

In this respect, graphical Granger models are well–known due to their intuitive interpretation and computational simplicity. They employ a penalized VAR model to the Granger concept [5], [6], [18], [39]. Non–zero coefficients in the corresponding VAR model w.r.t. a time series reveal a Granger causal relation. Since this problem can be ill–posed, penalizing the VAR model by means of a penalty function provides an efficient and sparse solution when the convergence to the true causal graph is ensured. Thus, in Paper F [10], we introduce a penalized VAR–based algorithm to detect the **H**eterogeneous **G**raphical **G**ranger **M**odel (HGGM) by employing generalized linear models (GLMs) [10]. GLMs allow us to generalize simple autoregressive models

to the case where several processes of different distributions from the exponential family are non–linearly related. Thereby, we introduce an integrative model to detect causal relations among a large number of heterogeneous time series addressing Contribution 2. Similar to the other graphical models, we assume that the interactions among the involved processes are additive. In order to ensure the convergence of HGGM to the true causal graph (i.e. consistency), we employ the well–know penalization approach, adaptive Lasso, with oracle properties [46].

In another point of view, information theory can be employed for a parameter–free measurement of the predictability addressing Contribution 4. In this respect, we regard the challenge of causal inference as a data compression problem in Paper H [16] and I [15]. In other words, employing the MDL–principle, time series y causes x if considering the past of y together with x decreases the number of bits required to encode x. More deviation in compression cost reveals stronger causal dependency among two time series. Unlike other information–theoretic approaches (e.g. entropy–based algorithms [36]), we incorporate the complexity of models employing the MDL–principle. Thereby, it leads to a natural trade–off among model complexity and goodness–of–fit while avoiding over–fitting. To avoid any information loss, we integrate processes of various distributions without any transformation or certain assumptions. That is, we utilize GLMs to extend the notion of Granger causality for heterogeneous time series data regardless of their distributions. Thereby, Contribution 2 is also presented in Paper H and I in the sense that we introduce an integrative information–theoretic framework for causal inference on time series while preserving the original distribution of every time series. Moreover, unlike many other algorithms in this category, we aim at detecting causal networks. To the best of our knowledge, almost all of the existing algorithms are designed based on a pairwise testing approach. This approach is inefficient when dealing with large causal networks. To avoid this issue, we propose our MDL–based greedy algorithm (ITGH) to detect heterogeneous Granger causal relations in a GLM framework. Paper I is an extended version of Paper H when we assess ITGH conducting various extentive synthetic and real–world experiments.

We investigate dependency anomalies in Paper E [9] and Paper G [14] utilizing temporal dependencies among time series, i.e. here, we interpret the term "deep knowledge" as causal interactions between time series. Thereby, Contribution 3 is addressed in both papers while a graphical Granger approach has been employed. We find the most accurate statistical model that captures the generation process of the normal (non-anomalous) data, then, investigate any deviation from this normal pattern. That is, we estimate the likelihood of a new observation based on the captured model and specify the data as an anomaly if the likelihood is below some thresholds. More precisely, we assume the training data to be non–anomalous and we name the model corresponding to the training data the normal pattern. In the next step, we investigate the test data observations and specify significant deviations to the normal pattern as anomalies. In particular, we propose an anomaly detection framework for heterogeneous time series in Paper E which consists of three main building blocks:

- detecting the temporal causal relations,
- identifying an appropriate anomaly score,
- introducing an efficient approach to specify anomalies.

Discovery of causal relations has been done in Paper E by employing a modification of the Granger–Lasso algorithm for heterogeneous data sets where a GLM framework is considered. Granger–Lasso is a well–known $L1$ penalization approach that deals with only Gaussian time series. The Granger–Lasso optimization problem is solved by using the least square cost function with the Lasso penalty for appropriately transformed input time series. We consider the same anomaly detection framework In Paper G [14], but here, we incorporate our proposed heterogeneous graphical Granger model (HGGM) for the discovery of temporal causal relations and introduce a new anomaly detection algorithm (AD–HGGM) for mixed time series. However, in both papers, we follow objectives of our second contribution and aim at proposing an integrative approach to capture the causal relations among time series of different distributions without enforcing any specific assumptions.

When all temporal dependencies are detected, the next step is to specify anomalies comparing the captured models w.r.t. training and test data. That is, we employ an anomaly score to measure the difference between two distributions. Thus, we employ Jensen–Shannon (JS) divergence as an anomaly score in our framework. JS–divergence is symmetric, its square root is metric and can be used as a distance function. These properties of JS–divergence improves efficiency in the sense that JS–divergence saves some computations.

To specify dependency anomalies we need a threshold defined based on the non–anomalous part of the data, i.e. training data. One could consider the entire training data to capture an anomaly threshold. However, inspired by the AD–GGM algorithm [33], we slide a window over training data and find an anomaly threshold w.r.t. every time window to give more insights about the exact position of the anomaly. That is, we compute the anomaly score introduced in the previous section for every window and approximate the distribution of anomaly scores for non–anomalous data. Employing a significance level α, the $\alpha - -quantile$ of this distribution is considered as threshold cutoff (refer to Section B in [33]).

CHAPTER 5

Conclusion

In this chapter, we revisit 4 research questions formulated in Section 3.2. Then, we investigate the relation between every research question and the corresponding contributions. Moreover, we mention research results w.r.t. every research question addressing various publications in that respect. Finally, we conclude with a discussion of open topics and potential future works in Section 5.2.

5.1 Revisiting Research Challenges

Figure 5.1 summarizes relations among research questions (formulated in Section 3.2), contributions (Section 4.2) as well as publications (Section 4.3). In the following, we elaborate how these central questions have been addressed by scientific contributions as well as corresponding publications.

Research Question 1

How can the effect of artificially defined relationships caused by a simple conversion of data types be avoided when mining heterogeneous data sets?

- **Contribution 1.** Utilizing specific characteristics of every data type in order to preserve original data and avoid the effect of artificially defined relations caused by a simple conversion of data types.
- **Contribution 3.** Employing useful characteristics of the data as well as incorporating spurious correlations to improve the accuracy when mining heterogeneous data.

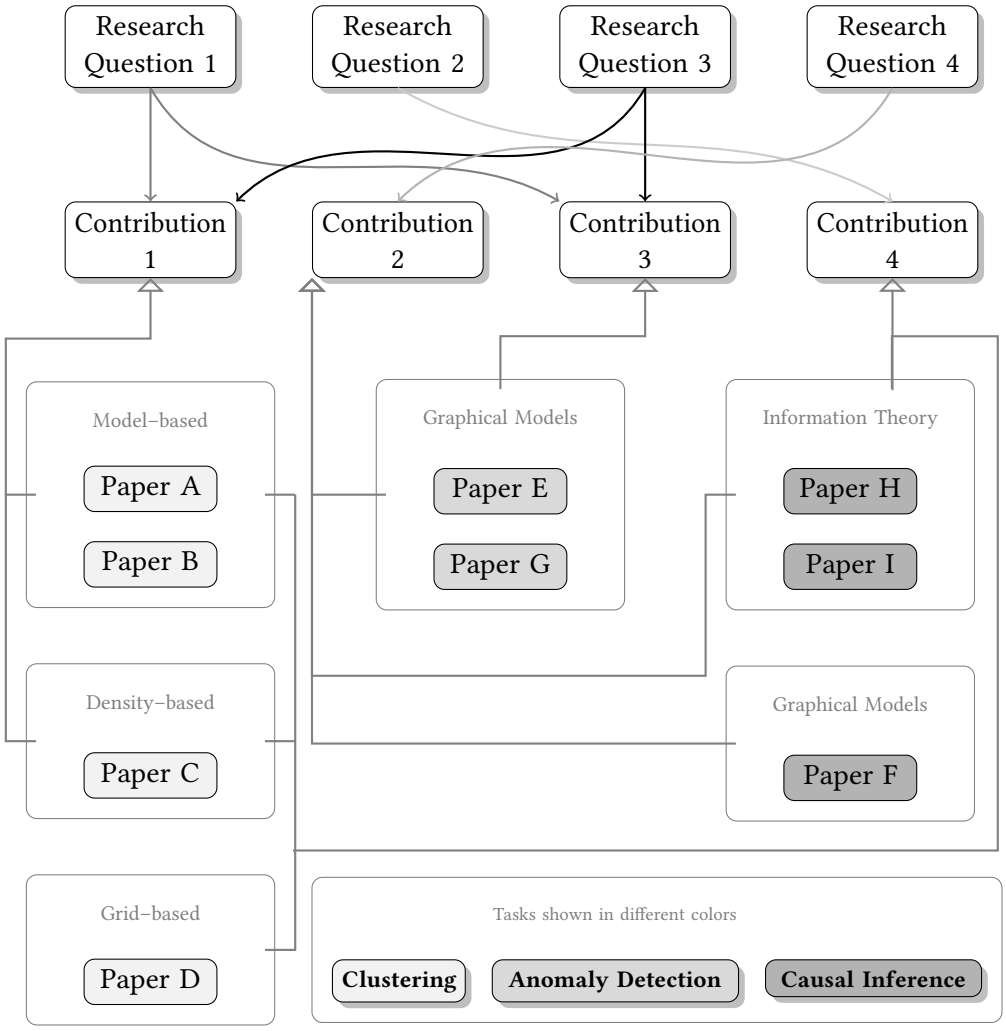


Figure 5.1: Summary of research questions, contributions and publications.

This research question has been addressed in Contribution 1 and 3 where Contribution 3 is a general case of Contribution1. Here, the key point is to avoid the drawbacks of a simple data type conversion. Every data type has several useful characteristics that one could utilize in various applications. These characteristics might be interpreted as "deep knowledge". Addressing this research question, our main contribution is to preserve original data and avoid effects of artificially defined relations caused by data type conversions employing "deep knowledge". In Paper A and B, we propose a model–based clustering algorithm incorporating the MDL–principle where for every attribute a representative model is required. For numerical data sets, a PDF represents an approximation of data. Here, we have two options for categorical attributes: either converting categorical attributes to numerical ones and considering a PDF to represent the data, or finding comprehensive models representing categorical attributes when preserving the original data. In this respect, we utilize the natural hierarchy among categorical values and introduce concept hierarchies to summarize categorical information. A concept could be the color of an object (e.g. light green or cyan), marital status (e.g. married), or the continent where a country is located (e.g. Asia). More precisely, a concept is a categorical value showing some characteristics of every data object. Concept hierarchies allow us to express conceptual interchangeable values by selecting an inner node of a concept hierarchy to describe a cluster. They not only capture more relevant categories for each cluster but also help to interpret the clustering results appropriately.

Considering the fact that almost always there is a natural hierarchy w.r.t. categorical values, we employ concept hierarchies in another way to introduce distance hierarchies as a distance measure available for both types of attributes (i.e. categorical and numerical). A distance hierarchy extends the definition of concept hierarchies by associating a weight to any link. Distance hierarchies are also applicable for numerical attributes resulting in a distance function similar to the weighted Euclidean distance. Thereby, we are allowed to integrate categorical and numerical attributes without any conversion in this framework. Thus, we, again, employ this useful characteristic of attributes and introduce an integrative framework to map an object in a mixed–type data set to a point in the associated distance hierarchy. Finally, in Paper C, we define a distance function applicable for both data types.

Research Question 2

How can the data be analyzed without a user having to specify some parameters, i.e. parameter–free data analysis?

- **Contribution 4.** Incorporating the MDL–principle for a parameter–free data mining.

This research question has been addressed in Contribution 4 resulting in various publications. That is, we employ the MDL–principle for different applications. In the context of clustering, the MDL–principle can be applied either as a clustering criterion or as a model selection approach. In both cases, the goal is to move toward non–parametric clustering algorithms.

Considering the first application of MDL, we regard the task of clustering as a data compression problem such that the best clustering is linked to the strongest data compression. Incorporating the MDL–principle, we cluster the data by relating the concept of likelihood to data compression where for every attribute a representative model is required. Given the appropriate model corresponding to any attribute, MDL leads to an intuitive clustering result employing the compression cost as a clustering criterion. The better the model matches major characteristics of the data, the better the clustering result is. Following the MDL–principle, we encode not only the data but also the model itself and minimize the overall description length. Simultaneously, we avoid over–fitting since the MDL–principle tends to a natural trade–off between model complexity and goodness–of–fit. Thereby, in Paper D, we introduce a grid–based clustering algorithm (ITGC) incorporating a MDL–based objective function. Although ITGC is proposed for pure numerical data sets, we investigate how to incorporate the MDL–principle for clustering purposes. Later, in Paper A and B, we introduce a model–based clustering algorithm (ClicoT) for mixed–type data sets considering data compression as an optimization goal. Essentially, MDL allows a unified view on various data types given an appropriate model for every attribute. Thus, ClicoT flexibly learns the relative importance of the two different sources of information (i.e. categorical and numerical attributes) for clustering without requiring a user to specify input parameters which are usually difficult to estimate.

The second application of MDL is to employ this concept as a model selection tool. That is, we evaluate various models and find the most accurate one in terms of the minimum description length criterion. Analogous to the previous application of MDL, here, we regard the model selection challenge to a data compression problem in the sense that more accurate models lead to less compression cost. In Paper C, we incorporate this concept and propose a parameter–free version of DBSCAN, i.e. MDBSCAN, for clustering of mixed–type data sets. That is, we fix one of the required parameters *MinPts*, i.e. minimum number of points in a neighborhood, and try a range of various radius. Every parameter setting is assumed as a specific model. Then, we execute DBSCAN for every parameter setting and store the clustering result. Finally, we employ MDL and evaluate various parameter settings (models) to find the best model in terms of the compression cost.

In another perspective, we incorporate the MDL–principle considering its second application to a non–parametric Granger causal inference from single–type data sets. That is, in Paper H and I, we propose ITGH regarding causality detection as a data compression problem where any improvement in the predictability is measured in terms of the compression cost. In other words, employing an information–theoretic indicator, time series y Granger–causes x if considering the past of y together with x decreases the number of bits required to encode x.

Research Question 3

How is it possible to increase the accuracy through "deep knowledge" when mining heterogeneous data?

- **Contribution 1.** Utilizing specific characteristics of every data type in order to preserve original data and avoid the effect of artificially defined relations caused by a simple conversion of data types.
- **Contribution 3.** Employing useful characteristics of the data as well as incorporating spurious correlations to improve the accuracy when mining heterogeneous data.

This research question has been addressed in Contribution 1 and 3. The main concept is how to incorporate useful characteristics and information about the data such that it leads to more accurate data analysis. This is exactly addressed in Contribution 3. Moreover, Contribution 1 is a specific case of Contribution 3 where we aim at preserving original data and avoiding any data type conversions. Nevertheless, Contribution 3 has a wider range of applications depending on how to interpret the term "deep knowledge". In the context of dependency anomaly detection of time series, "deep knowledge" can be interpreted as temporal causal dependencies among various time series. Dependency anomalies occur due to any changes in temporal dependencies comparing training and test data. Therefore, useful information about the causal interactions among time series improves the captured model w.r.t. time series. Essentially, we determine the temporal relations among a specific time series and others while we employ a causal inference technique. In a normal case, when no anomalies occur, the temporal causal graph is the same for training and test data. Thus, when learning temporal dependencies for test data, we improve accuracy of the model found for the test data by considering the null hypothesis (temporal dependencies in training data) as another constraint. In this respect, we propose an integrative anomaly detection framework for discovery of the dependency anomalies in heterogeneous time series in Paper E. Later, in Paper G, we improve characteristics of the proposed framework and employ our proposed heterogeneous causal inference algorithm (HGGM) for this application.

Research Question 4

Is it possible to avoid information loss caused by specific assumptions when mining heterogeneous data?

- **Contribution 2.** Integrate data of different natures as much as possible to avoid any information loss.

The forth research question has been addressed in Contribution 2. Essentially, here, the goal is to avoid any information loss caused by considering specific assumptions that are not

necessarily true. In this respect, we aim at integrating information from different natures in a heterogeneous data set and avoid any presumptions. In Paper F, H, and I, we deal with Granger causality among time series from different distributions. While the basic definition of Granger causality assumes a Gaussian distribution for all the time series, it does not have to necessarily be true in every application. Thus, we employ GLMs and extend the definition of Granger causality to a general case where time series may have distributions belonging to the exponential family. In fact, GLMs allow us to generalize simple autoregressive models to the case where several processes of different distributions are non–linearly related. That is, the response variable is not anymore a simple linear combination of covariates but its mean value is related to the covariates by a *link function*.

Later, in Paper E and G, we incorporate the proposed heterogeneous Granger causal inference algorithms for detecting dependency anomalies in time series. Thereby, our approach is not anymore restricted to only Gaussian time series and no specific assumptions about the exact distribution of time series is considered.

5.2 Future Works

The research presented in this thesis raised several questions and unlocked a number of important challenges to be investigated in the future. Thus, the potential future works are listed as follows:

- Many biological and microbiological applications generate mixed–type data sets. Applying our proposed clustering algorithms for mixed–type data sets to such data would be interesting. Particularly, clustering, for instance, various types of disease or grouping different gene expressions might reveal significant information.

- Clustering of time series as well as data stream is one of interesting and, at the same time, challenging topics in this context. Therefore, we are interested in assessing the performance of our proposed algorithm applying to a heterogeneous data stream.

- As a followup, clustering of heterogeneous time series utilizing Granger causal information is a potential future work. We might first investigate features in terms of possible causal relations and select most related features in this respect.

- Essentially, Graphs are considered as complex data sets. Among them, attributed graphs are more complex since corresponding to every node some attributes, might be heterogeneous, are associated. Such graphs get more interesting when there are multi relations among nodes. One of possible future works would be to investigate clustering of multi attributed graphs when we try to preserve all the characteristics of the nodes.

- Grid–based clustering algorithms may lead to inefficiency when dealing with huge data sets in terms of the dimensionality. Thus, focusing on our proposed grid–based clus-

tering algorithm (ITGC), a possible future work would be to investigate parallelization approaches in the sense that the required memory to store the grid information could be distributed.

- As another option for the further investigation for ITGC could be to enhance the partitioning procedure in the sense that it results a sparse grid which is cheaper in terms of the memory.

- Focusing on distance hierarchies, there are many different ways to appropriately assign link weights, e.g. [32], [24]. For simplicity in Paper C, we assign a constant weight to all the links uniformly. Other alternatives and a complete investigation on weight assignment approaches is an interesting issue deserving further research in the future.
- One of the avenues for future work is to employ our MDL–based approach (ITGH) to efficiently detect anomalies in heterogeneous data sets.

Bibliography

[1] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. "Automatic subspace clustering of high dimensional data for data mining applications". In: *SIGMOD Conference*. 1998, pp. 94–105.

[2] Amir Ahmad and Lipika Dey. "A k-mean clustering algorithm for mixed numeric and categorical data". In: *Data Knowl. Eng.* 63 (2 2007).

[3] Paul E. Green Anil Chaturvedi and J. Douglas Caroll. "K-modes Clustering". In: *Journal of Classification* 18.1 (2001).

[4] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. "OPTICS: Ordering Points to Identify the Clustering Structure". In: SIGMOD '99. New York, NY, USA, 1999.

[5] A. Arnold, Y. Liu, and N. Abe. "Temporal causal modelling with graphical Granger methods". In: *ACM SIGKDD* (2007).

[6] Mohammad Taha Bahadori and Yan Liu. "Granger Causality Analysis in Irregular Time Series". In: *SDM*. 2012.

[7] A. Barron, J. Rissanen, and B. Yu. "The minimum description length principle in coding and modeling". In: *IEEE Transactions on Information Theory* 44.6 (1998), pp. 2743–60.

[8] Sahar Behzadi, Hermann Hinterhauser, and Claudia Plant. "ITGC: Information-theoretic grid-based clustering". In: *International Conference on Extending Database Technology (EDBT)*. 2019.

[9] Sahar Behzadi, Kateřina Hlaváčková-Schindler, and Claudia Plant. "Dependency anomaly detection for heterogeneous time series: A Granger-Lasso approach". In: *IEEE International Conference on Data Mining (ICDM) workshops*. 2017.

[10] Sahar Behzadi, Kateřina Hlaváčková-Schindler, and Claudia Plant. "Granger Causality for Heterogeneous Processes". In: *The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. 2019.

[11] Sahar Behzadi, M. A. Ibrahim, and Claudia Plant. "Parameter Free Mixed-Type Density-Based Clustering". In: *International Conference on Database and Expert Systems Applications (DEXA)*. 2018.

[12] Sahar Behzadi, Nikola Müller, Claudia Plant, and Christian Böhm. "Clustering of Mixed-type Data Considering Concept Hierarchies". In: *The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. 2019.

[13] Sahar Behzadi, Nikola Müller, Claudia Plant, and Christian Böhm. "Clustering of Mixed-type Data Considering Concept Hierarchies: problem specification and algorithm". In: *International Journal of Data Science and Analytics*. 2020.

[14] Sahar Behzadi, Niklas Preschern, Kateřina Hlaváčková-Schindler, and Claudia Plant. "Anomaly Detection in Heterogeneous Time Series by Causality Mining". In: *Knowledge and Information Systems, submitted for publishing*. 2020.

[15] Sahar Behzadi, Benjamin Schelling, and Claudia Plant. "Information-theoretic Granger Causal Inference on Heterogeneous Data: Problem specification and algorithm". In: *International Journal of Data Science and Analytics, submitted for publishing*. 2020.

[16] Sahar Behzadi, Benjamin Schelling, and Claudia Plant. "ITGH: Information-theoretic Granger Causal Inference on Heterogeneous Data". In: *The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. 2020.

[17] Kailash Budhathoki and Jilles Vreeken. "Causal Inference on Event Sequences". In: *SDM*. 2018.

[18] Dehua Cheng, Mohammad Taha Bahadori, and Yan Liu. "FBLG: A Simple and Effective Approach for Temporal Dependence Discovery from Time Series Data". In: *KDD*. 2014.

[19] Gil David and Amir Averbuch. "SpectralCAT: Categorical spectral clustering of numerical and nominal data". In: *Pattern Recognition* 45.1 (2012), pp. 416 –433. ISSN: 0031-3203.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1 (1977), pp. 1–38.

[21] Michael D Eichler. "Causal inference in time series analysis". In: 2012.

[22] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise." In: *KDD Conference*. 1996.

[23] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. "From Data Mining to Knowledge Discovery in Databases". In: *AI Magazine* 17.3 (1996).

[24] P. Ronkainen G. Das H. Mannila. "Similarity of attributes by external probes". In: *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining* (1998), pp. 23–29.

[25] Clive WJ Granger. "Investigating causal relations by econometric models and cross-spectral methods". In: *Econometrica: Journal of the Econometric Society* (1969), pp. 424–438.

[26] Zengyou He, Xiaofei Xu, and Shengchun Deng. "Clustering Mixed Numeric and Categorical Data: A Cluster Ensemble Approach". In: *CoRR* abs/cs/0509011 (2005).

[27] Chung-Chian Hsu and Yu-Cheng Chen. "Mining of mixed data with application to catalog marketing". In: *Expert Syst. Appl.* 32.1 (2007), pp. 12–23.

[28] Zhexue Huang. "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values". In: *Data Min. Knowl. Discov.* 2 (3 1998).

[29] J. B. Macqueen. "Some methods of classification and analysis of multivariate observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. 1967, pp. 281–297.

[30] Damien Mcparland and Isobel Claire Gormley. "Model Based Clustering for Mixed Data: ClustMD". In: *Adv. Data Anal. Classif.* 10.2 (2016).

[31] J. A Nelder and R. J. Baker. "Generalized linear models". In: *Encyclopedia of statistical sciences* (1972).

[32] Faloutsos C. Palmer C.R. *Electricity Based External Similarity of Categorical Attributes*. Advances in Knowledge Discovery and Data Mining. PAKDD, 2003.

[33] H. Qiu, Y. Liu, N. A Subrahmanya, and W. Li. "Granger causality for time-series anomaly detection". In: *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE. 2012, pp. 1074–1079.

[34] Vaibhav Rajan and Sakyajit Bhattacharya. "Dependency Clustering of Mixed Data with Gaussian Mixture Copulas". In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. IJCAI'16. New York, New York, USA: AAAI Press, 2016, 1967–1973. ISBN: 9781577357704.

[35] Benjamin Schelling, Lena G. M. Bauer, Sahar Behzadi, and Claudia Plant. "Utilizing Structure-rich Features to improve Clustering". In: *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases 2020*. 2020.

[36] Thomas Schreiber. "Measuring information transfer". In: *Physical review letters* 85.2 (2000), p. 461.

[37] C. E. Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.4 (1948), pp. 623–56.

[38] Gholamhosein Sheikholeslami, Surojit Chatterjee, and Aidong Zhang. "WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases". In: VLDB '98. San Francisco, CA, USA, 1998.

[39] A. Shojaie and G. Michailidis. "Discovering graphical Granger causality using the truncating lasso penalty". In: *Bioinformatics* 26.18 (2010), pp. i517–i523.

[40] R. Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), pp. 267–288.

[41] Nguyen X. Vinh, Julien Epps, and James Bailey. "Information theoretic measures for clusterings comparison: is a correction for chance necessary?" In: *ICML*. 2009.

[42] Can Wang, Chi-Hung Chi, Wei Zhou, and Raymond Wong. "Coupled Interdependent Attribute Analysis on Mixed Data". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI'15. Austin, Texas: AAAI Press, 2015, 1861–1867. ISBN: 0262511290.

[43] Wei Wang, Jiong Yang, and Richard R. Muntz. "STING: A Statistical Information Grid Approach to Spatial Data Mining". In: VLDB '97. San Francisco, CA, USA, 1997, pp. 186–195.

[44] Qiang Yang and Xindong Wu. “10 CHALLENGING PROBLEMS IN DATA MINING RESEARCH”. In: *International Journal of Information Technology amp; Decision Making (IJITDM)* 05.04 (2006).

[45] Jian Yin and Zhifang Tan. “Clustering Mixed Type Attributes in Large Dataset”. In: *ISPA*. 2005, pp. 655–661.

[46] H. Zou. “The adaptive lasso and its oracle property”. In: *Journal of the Am. Stat. Assoc.* (2008), pp. 1418–1429.

CHAPTER 6

Paper A & Paper B: Clustering of mixed–type data considering concept hierarchies: problem specification and algorithm

This chapter comprises two publications concerning clustering of mixed-type heterogeneous data, Paper A [12] and its extended journal version Paper B [13]. Here, the journal version is included which consists of also the conference paper (Paper A).

Authors Contributions:

- **Sahar Behzadi.** Cooperation on the main idea, developing the algorithm, implementation, and writing the paper. Conducting experiments.
- **Nikola S. Müller.** Cooperation on the main idea, developing the algorithm, implementation, and writing the paper.
- **Claudia Plant.** Supervision during development and evaluation of the algorithm as well as writing the paper.
- **Christian Böhm.** Supervision during development and evaluation of the algorithm as well as writing the paper.

International Journal of Data Science and Analytics
https://doi.org/10.1007/s41060-020-00216-2

REGULAR PAPER

Clustering of mixed-type data considering concept hierarchies: problem specification and algorithm

Sahar Behzadi[1] · Nikola S. Müller[2] · Claudia Plant[1,4] · Christian Böhm[3]

Received: 8 November 2019 / Accepted: 27 March 2020

Abstract
Most clustering algorithms have been designed only for pure numerical or pure categorical data sets, while nowadays many applications generate mixed data. It raises the question how to integrate various types of attributes so that one could efficiently group objects without loss of information. It is already well understood that a simple conversion of categorical attributes into a numerical domain is not sufficient since relationships between values such as a certain order are artificially introduced. Leveraging the natural conceptual hierarchy among categorical information, concept trees summarize the categorical attributes. In this paper, we introduce the algorithm *ClicoT* (**CL**ustering mixed-type data **I**ncluding **CO**ncept **T**rees) as reported by Behzadi et al. (Advances in Knowledge Discovery and Data Mining, Springer, Cham, 2019) which is based on the minimum description length principle. Profiting of the conceptual hierarchies, ClicoT integrates categorical and numerical attributes by means of a MDL-based objective function. The result of ClicoT is well interpretable since concept trees provide insights into categorical data. Extensive experiments on synthetic and real data sets illustrate that ClicoT is noise-robust and yields well-interpretable results in a short runtime. Moreover, we investigate the impact of concept hierarchies as well as various data characteristics in this paper.

Keywords Mixed-type data · Information-theoretic clustering

1 Declarations

– *Availability of data and material* We used *MPG*, *Automobile* and *Adult* data sets from the UCI Public Data Repository [7] as well as *Airport* data set from the public project *Open Flights* (http://openflights.org/data.html).

– *Code availability* Our algorithm is implemented in Java and the source code as well as the data sets are publicly available here: https://tinyurl.com/ucp8289.

2 Introduction

Clustering mixed data is a non-trivial task and typically is not achieved by well-known clustering algorithms designed for a specific type. It is already well understood that converting one type to another one is not sufficient since it might lead to information loss. Moreover, relations among values (e.g., a certain order) are artificially introduced. In order to elaborate the issue, we generate a synthetic mixed-type data and investigate the impact of converting a categorical data type to a numerical one while applying well-known clustering algorithms.

Let Fig. 1 show a synthetically generated mixed-type data consisting of three different clusters illustrated by different shapes (rectangle, circle, cross), i.e., shapes are cluster IDs or ground truth. Thus, there are two Gaussian-shaped clusters where one of them (points with the shape rectangle) includes

Sahar Behzadi
sahar.behzadi@univie.ac.at

Nikola S. Müller
nikola.mueller@helmholtz-muenchen.de

Claudia Plant
claudia.plant@univie.ac.at

Christian Böhm
boehm@ifi.lmu.de

[1] Faculty of Computer Science, Data Mining, University of Vienna, Vienna, Austria

[2] Helmholtz Research Center, Munich, Germany

[3] University of Munich, Munich, Germany

[4] ds:Univie, Vienna, Austria

Published online: 25 April 2020

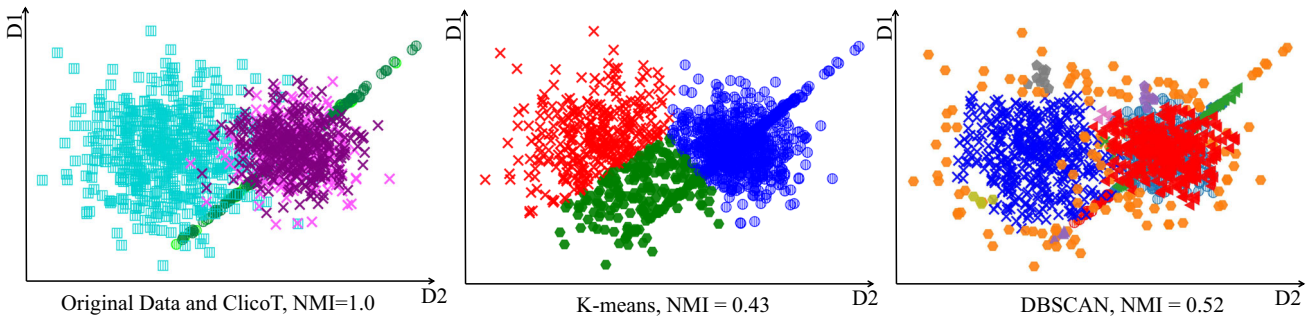


Fig. 1 Clustering results after converting categorical attribute *Color* to numerical (color figure online)

only data points having cyan as their color and the other cluster (points with the shape cross) includes data points having purple and rose as their color. The last cluster (points with the shape circle) is a line-shaped cluster consisting of dark green and light green data points.

The data set comprises two numerical attributes concerning the position of data objects and a categorical attribute representing the color of data points (rose, purple, light green, dark green and cyan). Therefore, a data object in our synthetic data looks like, for example (1, 2, purple). Numerical attributes are generated following various random Gaussian distributions. We simply converted the color to a numerical attribute by mapping numbers to various colors. Considering the *normalized mutual information* (NMI) [17] as an evaluation measure, Fig. 1 depicts the inefficiency of applying *K*-means and DBSCAN, two popular clustering algorithms, on the converted data. Therefore, integrating categorical and numerical attributes without any conversion is required since it preserves the original format of any attribute.

Utilizing the *minimum description length* (MDL) principle, we can regard the clustering task as a data compression problem such that the best clustering is linked to the strongest data set compression. MDL allows integrative clustering by relating the concepts of likelihood and data compression while for any attribute a representative model is required. Although for solely numerical data sets a *probability distribution function* (PDF) represents an approximation of data, finding an appropriate approximation for categorical attributes is not straightforward. Considering the natural hierarchy among categorical values, *concept hierarchies* are introduced to summarize the categorical information. Back to the running example, assuming pink as a higher-level hierarchy for the objects in the cluster consisting of rose and purple, points with the shape $\times$ more accurately represent the characteristics of the cluster.

Beyond the clustering approaches, detecting the most relevant attributes during this process improves the quality of clustering. However, considering a data set with an unknown distribution where only few subgroups in the data space are actually relevant to characterize a cluster, it is not trivial to recognize the cluster-specific attributes. Thus, we introduce an information-theoretic greedy approach to specify the most relevant attributes. As a result, the novel parameter-free **CL**ustering algorithm for mixed-type data **I**ncluding **CO**ncept **T**rees, shortly *ClicoT*, provides a natural interpretation. The approach consists of several contributions:

- *Integration* ClicoT integrates two types of information considering data compression as an optimization goal. ClicoT flexibly learns the relative importance of the two different sources of information for clustering without requiring the user to specify input parameters which are usually difficult to estimate.
- *Interpretation* In contrast to most clustering algorithms, ClicoT not only provides information about *which* objects are assigned to which clusters, but also gives an answer to the central question *why* objects are clustered together. As a result of ClicoT, each cluster is characterized by a signature of cluster-specific relevant attributes providing appropriate interpretations.
- *Robustness* The compression-based objective function ensures that only the truly relevant attributes are marked as cluster-specific attributes. Thereby, we avoid overfitting, enhance the interpretability and guarantee the validity of the result.
- *Usability* ClicoT is convenient to be used in practice since the algorithm scales well to large data sets. Additionally, the compression-based approach avoids difficult estimation of input parameters, e.g., the number or the size of clusters.

Moreover, in this paper we elaborate the concept hierarchies and investigate the impact of them on the performance of ClicoT. We also address whether or not various characteristics of data sets, e.g., proportion of categorical and numerical attributes, have any influence on the effectiveness of ClicoT via extensive experiments.

3 Clustering mixed data types

To design a mixed-type clustering algorithm, we need to address three fundamental questions: How to model numerical attributes to properly characterize a cluster? How to model categorical attributes? And finally how to efficiently integrate heterogeneous attributes when the most relevant attributes are specified? In principle, a PDF summarizes values by approximating meaningful parameters. However, the idea of using a background PDF for categorical attributes is not intuitive at first; therefore, we employ concept hierarchies.

3.1 Concept hierarchy

In this paper, a concept could be color of an object (e.g., light green or cyan), marital status (e.g., married) or the continent where a country is located (e.g., Asia). More precisely, a concept is a categorical value showing some characteristics of every data object. As mentioned, concept hierarchies allow us to express conceptual interchangeable values by selecting an inner node of a concept hierarchy to describe a cluster. Concept hierarchies not only capture more relevant categories for each cluster but also help to interpret the clustering result appropriately. Let $\mathcal{DB}$ denote a database consisting of n objects. An object o comprises m categorical attributes $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$ and d numerical attributes $\mathcal{X} = \{x_1, x_2, \ldots, x_d\}$. For a categorical attribute A_i, we denote different categorical values by ${A_i}^{(j)}$. An *Element* represents a categorical value or a numerical attribute and we denote the number of all *Element*s by E. Considering the natural hierarchy between different categories, for each categorical attribute A_i a concept hierarchy is already available as follows:

Definition 1 *Concept Hierarchy* Let $T_{A_i} = (N, \mathcal{E})$ be a tree with root A_i denoting the concept hierarchy corresponding to the categorical attribute A_i with the following properties:

1. T_{A_i} consists of a set of nodes $N = \{n_1, n_2, \ldots, n_s\}$ where any node is corresponding to a categorical concept. $\mathcal{E}$ is a set of directed edges $\mathcal{E} = \{e_1, e_2, \ldots, e_{(s-1)}\}$, where n_j is a parent of n_z if there is an edge $e_l \in \mathcal{E}$ so that $e_l = (n_j, n_z)$.
2. The level $l(n_j)$ of a node n_j is the height of the descendant sub-tree. If n_j is a leaf, then $l(n_j) = 0$. In a concept, tree leaf nodes are categorical values existing in the data set. The root node is the attribute A_i which has the highest level, also called the height of the concept hierarchy.
3. Each node $n_j \in N$ is associated with a probability $p(n_j)$ which is the frequency of the corresponding category in a data set.

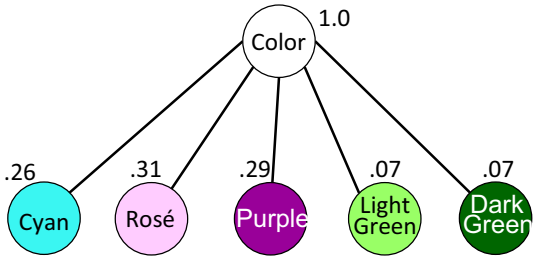


Fig. 2 A flat concept tree for the categorical attribute color (color figure online)

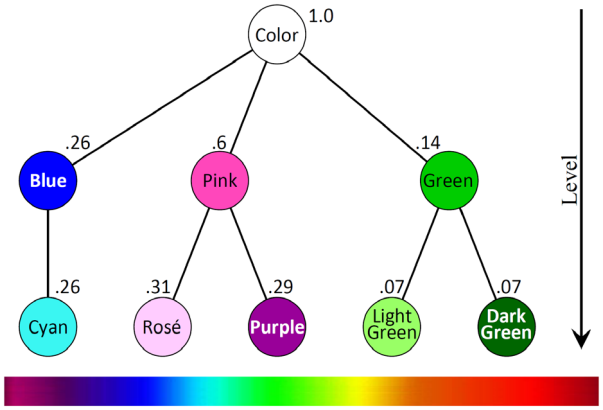


Fig. 3 Concept tree corresponding to the running example w.r.t. the natural hierarchy among various colors (color figure online)

4. Each node n_j represents a sub-category of its parent; therefore, all probabilities of the children sum up to the probability of the parent node.

To elaborate, let us consider the synthetic example illustrated in Sect. 2 (see Fig. 1). In this example, the only categorical attribute is color, while two other attributes are numeric. Therefore, $\mathcal{A} = \{A_1\}$ and $\mathcal{X} = \{x_1, x_2\}$ where A_1 is color and X_1, X_2 show X and Y-axis in a two-dimensional space. Every data point in this data set can have one of the following colors: cyan, rose, purple, light green and dark green. Thus, the set of categorical values w.r.t. A_1 is { cyan, rose, purple, light green and dark green}. As a default, we assume a flat concept tree to summarize the frequency of categorical values, especially, when there is no meaningful hierarchy among different categories. A flat hierarchy consists of a one level tree including all the categories in the leaf level without any hierarchy. Figure 2 depicts a default flat concept tree corresponding to the running example. However, usually, for each categorical attribute a concept hierarchy is available due to the natural hierarchy among different categories. For instance, considering the natural scalable range of colors, one can categorize different colors as illustrated in Fig. 3. Here, the height is 2 showing another concept level (level 1) which categorizes the color of data points, e.g., green categorizes

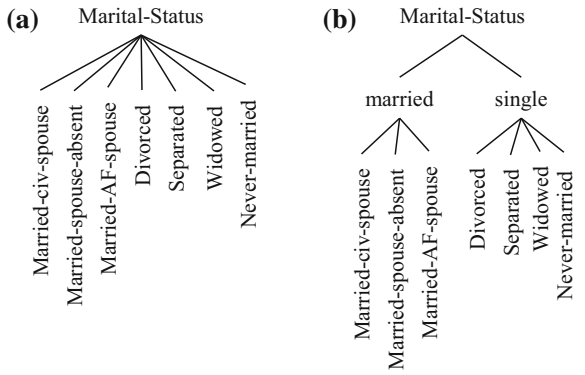


Fig. 4 Concept tree for categorical attribute *marital status* w.r.t. Adult data set

dark green and light green in a same category based on the natural scalable range of colors. The node labels show background probabilities $p(n_j)$ (i.e., frequency) for each node. This initialization of the background distribution is once performed before assigning objects to clusters.

Categorical attributes are more often observed in real applications, e.g., population surveys. As an example, we focus on *Adult* data, a real-world data set from the UCI repository [7]. Adult data set without missing values, extracted from the census bureau database, consists of 48,842 instances of 11 attributes. The class attribute Salary indicates whether the salary is over 50K or lower. Categorical attributes consist of different information about people in this survey, e.g., work-class, education, occupation and marital status. Focusing on the categorical feature marital status, every person belongs to a unique category including divorced, never married, married-spouse-absent, etc. The leaf level shows various marital status one could have in both concept trees illustrated in Fig. 4. The left concept tree without any hierarchy (Fig. 4a) shows the default flat hierarchy can be considered in the beginning. However, we can categorize various status based on whether or not a person is married. In this case, three different categories, i.e., married-civ-spouse, married-spouse-absent and married-AF-spouse, fall in the same category, married. All other categorical values, i.e., divorced, separated, widowed and never-married, cannot be located in the same category since people having these status are single. Therefore, we consider another category, single, which seems more plausible for those status. Thus, Fig. 4b shows one of the possible concept hierarchies one can assume w.r.t. marital status for Adult data. We investigate this data set in more detail in Sect. 6.

ClicoT profits the concept hierarchy to provide more interpretable results. But also non-hierarchical categorical attributes can be regarded as a simple flat concept hierarchies with height one. We claim our algorithm performs appropriately in comparison with other algorithms for this case as well.

3.2 Cluster-specific elements

Besides an efficient clustering approach, finding relevant attributes to capture the best fitting model is important. Usually, the clustering result is disturbed by irrelevant attributes. To make the model for each cluster more precise, we distinguish between relevant and irrelevant attributes. Each cluster c is associated with a subset of the numerical and categorical relevant elements denoted by *cluster-specific elements*. Categorical cluster-specific elements are represented by a specific concept hierarchy which diverges from the background hierarchy (i.e., the concept hierarchy of the entire database).

Definition 2 *Cluster* A cluster c is described by:

1. A set of objects $\mathcal{O}_c \subset \mathcal{DB}$.
2. A cluster-specific subspace $I = \mathcal{X}_c \cup \mathcal{A}_c$, where $\mathcal{X}_c \subseteq \mathcal{X}$ and $\mathcal{A}_c \subseteq \mathcal{A}$.
3. For any categorical attribute $A_i \in \mathcal{A}_c$, the corresponding cluster-specific concept hierarchy is a tree ${T^c}_{A_i} = (N_c, \mathcal{E}_c)$ with nodes and edges as specified in Definition 1. $N_c \subset N$ indicates the cluster-specific nodes. For computing the probabilities associated with the cluster-specific nodes instead of all n objects, only the objects $\mathcal{O}_c$ in cluster c are applied, i.e., $p(n_j) = \frac{|n_j|}{|\mathcal{O}_c|}$.

The idea of cluster-specific nodes allows to specify an inner node as a representative for each cluster. ClicoT aims at finding a partition of $\mathcal{DB}$ into clusters, and simultaneously at discovering the cluster-specific subspace for each cluster.

3.3 Integrative objective function

Given the appropriate model corresponding to any attribute, MDL allows a unified view on mixed data. The better the model matches major characteristics of the data, the better the result is. Following the MDL principle [16], we encode not only the data but also the model itself and minimize the overall description length. Simultaneously, we avoid overfitting since the MDL principle tends to a natural trade-off between model complexity and goodness-of-fit.

Definition 3 *Objective Function* Considering cluster c the description length (DL) corresponding to this cluster is defined as:

$$\mathrm{DL}(c) = \mathrm{DL}_n(\mathcal{X}) + \mathrm{DL}_c(\mathcal{A}) + \mathrm{DL}(model(c)) \tag{1}$$

The first two terms, i.e., DL_n and DL_c, represent coding costs concerning numerical and categorical attributes, respectively. The last term ($\mathrm{DL}(model)$) denotes the model encoding cost. Essentially, numerical and categorical attributes contribute simultaneously and in the same way. We incorporate the

required coding cost for both types, numerical and categorical, without any data type conversion. Thus, instead of data type conversion we integrate all attributes avoiding information loss. Our proposed objective function minimizes the overall description length of the database which is defined as:

$$\mathrm{DL}(\mathcal{DB}) = \sum_{c \in \mathcal{C}} \mathrm{DL}(c) \tag{2}$$

Coding Numerical Attributes Considering Huffman coding scheme, the description length of a numerical value o_i is defined by $-\log_2 \mathrm{PDF}(o_i)$. We assume the same PDF to encode the objects in various clusters and clusters compete for an object while the description length is computed by means of the same PDF for every cluster. Therefore, any PDF would be applicable and using a specific model is not a restriction [4]. For simplicity, we select Gaussian PDF, $\mathcal{N}(\mu, \sigma)$. Moreover, we distinguish between the cluster-specific attributes in any cluster c, denoted by $\mathcal{X}_c$, and the remaining attributes $\mathcal{X} \setminus \mathcal{X}_c$ (Definition 2). Let μ_i and σ_i denote the mean and variance corresponding to the numerical attribute x_i in cluster c. If x_i is a cluster-specific element ($x_i \in \mathcal{X}_c$), we consider only cluster points to compute the parameters otherwise ($x_j \in \mathcal{X} \setminus \mathcal{X}_c$) the overall data points will be considered. Thus, the coding cost for numerical attributes in cluster c is provided by:

$$DL_n(\mathcal{X}) = \sum_{x_i \in \mathcal{X}} \sum_{o_i \in \mathcal{O}_c} -\log_2 \left(\mathcal{N}(\mu_i, \sigma_i) \right) \tag{3}$$

Coding Categorical Attributes Analogously, we employ Huffman coding scheme for categorical attributes. The associated probability to a category is its frequency w.r.t. either the specific or the background hierarchy (Definition 1). Similar to numerical attributes, we assume $\mathcal{A}_c$ as the set of cluster-specific categorical attributes and $\mathcal{A} \setminus \mathcal{A}_c$ for the rest. Let o_j denote a categorical object value corresponding to the attribute A_j. We define $f(A_j, o_j)$ as a function which maps o_j to a node in either a specific or a background hierarchy depending on A_j. In summary, $f(.)$ is defined as:

$$f(A_j, o_j) = \begin{cases} n_j \in T^c{}_{A_j} & A_j \in \mathcal{A}_c \\ n_j \in T_{A_j} & A_j \in \mathcal{A} \setminus \mathcal{A}_c \end{cases}$$

Thus, the categorical coding cost for a cluster c is given by:

$$\mathrm{DL}_c(\mathcal{A}) = \sum_{A_j \in \mathcal{A}} \sum_{o_j \in \mathcal{O}_c} -\log_2 \left(p(f(A_j, o_j)) \right) \tag{4}$$

Model Complexity Without taking the model complexity into account, the best result will be a clustering consisting of singleton clusters. This result is completely useless in terms of the interpretation. Focusing on cluster c, the model complexity is defined as:

$$\begin{aligned} \mathrm{DL}(model(c)) = {} & idCosts(c) + SpecificIdCosts(c) \\ & + paramCosts(c) \end{aligned} \tag{5}$$

The idCosts are required to specify which cluster is assigned to a object while balancing the size of clusters. Employing the Huffman coding scheme, idCosts are defined by $|\mathcal{O}_c| \cdot log_2 \frac{n}{|\mathcal{O}_c|}$ where $|\mathcal{O}_c|$ denotes the number of objects assigned to cluster c. Moreover, in order to avoid information loss we need to specify whether an attribute is a cluster-specific attribute or not. That is, given the number of specific elements s in cluster c, the coding costs corresponding to these elements, *SpecificIdCosts*, is defined as:

$$SpecificIdCosts(c) = s \cdot \log_2 \frac{E}{s} + (E - s) \cdot \log_2 \frac{E}{(E - s)} \tag{6}$$

Following fundamental results from information theory [16], the costs for encoding the model parameters are reliably estimated by:

$$paramCosts(c) = \frac{numParams(c)}{2} \cdot \log_2 |\mathcal{O}_c| \tag{7}$$

For any numerical cluster-specific attribute, we need to encode its mean and variance while for a categorical one the probability deviations to the default concept hierarchy need to be encoded, i.e., $numParams(c) = |\mathcal{X}| \cdot 2 + \sum_{A_i \in \mathcal{A}} |N_c|$. Moreover, we need to encode the probabilities associated with the default concept hierarchy, as well as the default (global) means and variances for all numerical attributes. However, these costs are summarized to a constant term which does not influence our subspace selection and clustering technique.

4 Algorithm

Together with the main building blocks of ClicoT, two other steps are required to achieve an appropriate parameter free clustering: (1) recognizing the cluster-specific elements and (2) probability adjustments.

4.1 How to specify cluster-specific elements?

The optimization process in the objective function tends to mark an element with the most cost saving as a cluster-specific. Let the *specific coding cost* denote the cost where an element is marked as specific and the *non-specific coding cost*

Algorithm 1 Cluster-specific elements

```
Deviation (Element e_i)
scc := specific coding cost
nscc := non-specific coding cost
dev := deviation in terms of coding cost
if e_i is numerical then
    // case 1: e_i is specific
    // find N(μ_i, σ_i) w.r.t. O_c and compute DL_n(.)
    s = s + 1
    scc = Σ_{c∈C} DL(c)

    // case 2: e_i is not specific
    // find N(μ_i, σ_i) w.r.t. DB and compute DL_n(.)
    nscc = Σ_{c∈C} DL(c)

else if e_i is categorical then
    // case 1: e_i is specific
    A_j := categorical attribute w.r.t. e_i
    T_{A_j} := concept tree w.r.t. A_j
    // adjust T_{A_j} and get T^c_{A_j}
    for all Vertex V in T_{A_j} do
        ProcessHierarchy(V)
    end for
    // find DL_c(.) w.r.t. specific concept tree T^c_{A_j}
    P(o) = P(n) where n ∈ T^c_{A_j}
    s = s + 1
    scc = Σ_{c∈C} DL(c)

    // case 2: e_i is not specific
    // find DL_c(.) w.r.t. background concept tree T_{A_j}
    P(o) = P(n) where n ∈ T_{A_j}
    nscc = Σ_{c∈C} DL(c)
end if

// find the deviation
dev = |nscc − scc|
return dev
```

Algorithm 2 Concept tree updates

```
ProcessHierarchy (Vertex V)
ssp := sum of specific probabilities
sup := sum of unspecific probabilities
if V is a leaf then
    if V is specific then
        return (V.probability, 0)
    end if
    return (0, V.backgroundProbability)
end if
// now V is not a leaf
(ssp, sup) := (0, 0)
for all C in children(V) do
    (s, u) := processHierarchy(C)
    (ssp, sup) := (ssp + s, sup + u)
end for
if V is specific or root then
    factor := (V.probability − ssp)/sup
    for all C in children(V) do
        propagateDownFactor(C, factor)
    end for
    return (V.probability, 0)
end if
return (ssp, sup)
```

indicates the cost otherwise. Consulting the idea that cluster-specific elements have the most deviation of specific and non-specific cost and therefore saves more coding costs, we introduce a greedy method to recognize them. Algorithm 1 summarizes how to find the coding cost deviation w.r.t. every element e_i. We sort the elements according to their deviations and specify the first element as a cluster-specific element. We continue marking elements until marking more elements does not pay off in terms of the coding cost. Note that different nodes of a concept hierarchy have the same opportunity to be specific. Additionally marking a categorical element influences the specific concept hierarchy; therefore, we have to consider the adapted probabilities (next section).

4.2 Probability adjustment

To adjust the probabilities for a numerical cluster-specific attribute, we can safely use mean and variance corresponding to the cluster. In contrast, learning the cluster-specific concept hierarchy is more challenging since we need to maintain the integrity of a hierarchy. According to Definition 1, we assure that node probabilities of siblings in each level sum up to the probability of the parent node. Moreover, node probabilities should sum up to one for each level.

Algorithm 2 summarizes the adjustment process where *ProcessHierarchy*() is a recursive procedure to update the concept tree assuming marked cluster-specific elements. It, firstly, determines all probabilities in a concept hierarchy starting from the following configuration: Initially, all nodes are assigned to the background probability of the overall data set (*V.backgroundProbability*). An arbitrary number of (internal and/or leaf) nodes are marked as *cluster-specific* and assigned to different probabilities, taken from the currently considered cluster (*V.probability*). The recursive procedure is always started at the root node. When descending the concept hierarchy recursively, for each node we keep track of two sums, that of the specific probabilities inside the complete subtree (ssp) and that of the unspecific ones (sup). When returning from a recursion, we pass exactly these two variables to the caller, enabling him to determine how much the remaining probabilities must be adjusted. Whenever we return from the recursion and reach a cluster-specific node, we determine an adjustment factor according to the formula

$$factor = \frac{V.probability - ssp}{sup} \tag{8}$$

which is the factor correcting the deviation between all cluster-unspecific nodes in the sub-tree from the probability which we have in the current specific node. This factor is propagated down the concept hierarchy using the procedure PropagateDownFactor() in Algorithm 3 which is again

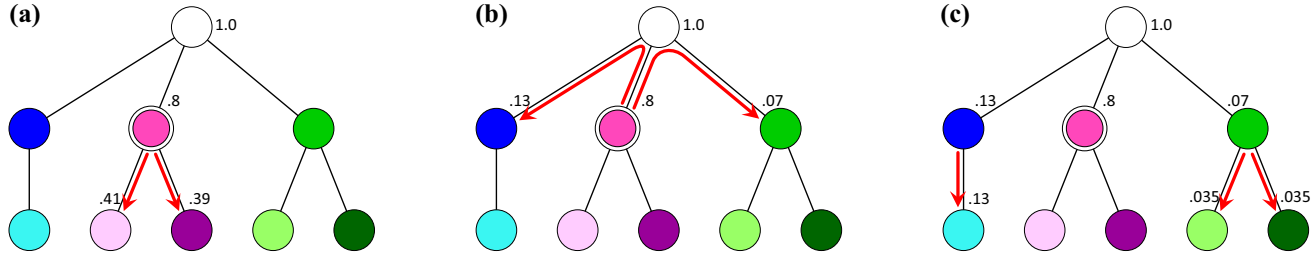


Fig. 5 Update concept hierarchies considering *pink* as a cluster-specific node (color figure online)

recursive and descends the sub-tree only in the non-cluster-specific nodes, because only for those we can adapt the probabilities. If ConceptTreeUpdate() returns to a cluster-unspecific node, it only sums up ssp and sup and delivers this information to its caller without directly down-propagating anything. Overall, the recursive method ProcessHierarchy() visits every node of the concept hierarchy once (and only once). During this whole recursive procedure, it is also guaranteed that PropagateDownFactor() also visits every node at most once. Thus, the method is linear in the number of nodes.

Algorithm 3 Down-propagation of the adjustment factor

```
1: PropagateDownFactor (Vertex V, double factor)
2: if V is unspecific then
3:     V.probability := V.probability · factor
4:     if V is not leaf then
5:         for all C ∈ V.children do
6:             PropagateDownFactor(C, factor)
7:         end for
8:     end if
9: end if
```

To clarify, let Fig. 5 show the procedure on the concept hierarchy corresponding to the running example (Fig. 1) where labels denote the frequencies. Moreover, let pink be a cluster-specific node for the cluster with the shape $\times$. The adjustment starts with the root node and processes its children. Then, it continues computing the relative probabilities for the specific concept hierarchy rather by background probability fraction (Fig. 5a). 80% relative probability should be distributed between two children, rose and purple, based on the computed propagation factor. During the next step, the remaining 20% probability is assigned level-wise to blue and green to assure that probabilities in each level sum up to 1 (Fig. 5b). Again each parent propagates down its probability (Fig. 5c). The result is a concept hierarchy best fitting to the objects when the background distributions are preserved.

4.3 ClicoT algorithm

ClicoT is a top-down parameter-free clustering algorithm. That is, we start from a cluster consisting of all objects and

Algorithm 4 ClicoT

```
1: input DB
2: learn background distributions of each attribute
3: C' = {C_0} with C'_0 = O_i ∈ DB
4: repeat
5:     // try to split until convergence
6:     C = C'
7:     cost = DL(DB|C) // current cost
8:     C' = {C'_1 ... C'_{k-1}} split worst C_i ∈ C to {C'_i, C'_k}
9:     while clustering C' changes do
10:        C'_i = {O_j : min_i DL(O_j|C'_i)} // assign objects
11:        Select cluster-specific elements by a greedy method for each
           cluster and compute costs
12:        Update each attribute of C'_i
13:    end while
14:    cost' = DL(DB|C') // split cost
15: until cost > cost'
16: k = |C|
17: return C, k
```

iteratively split down the most expensive cluster c in terms of the coding cost to two new clusters $\{c'_a, c'_b\}$. Then, we apply a k-means-like strategy and assign every point to closest cluster which is nothing else than the cluster with the lowest increase in the coding cost. Employing the greedy algorithm, we determine the cluster-specific elements and finally we compute the compression cost for clustering results in two cases, before and after splitting (Definition 3). If the compression cost after splitting, i.e., $\mathcal{C}'$ with $|\mathcal{C}'| = k + 1$, is cheaper than the cost of already accepted clustering $\mathcal{C}$ with $|\mathcal{C}| = k$, then we continue splitting the clusters. Otherwise the termination condition is reached and the algorithm will be stopped.

5 Related work

Driven by the need of real applications, the topic of clustering mixed-type data represented by numerical and categorical attributes has attracted attentions, e.g., CFIKP [19], CAVE [10], CEBMDC [8]. In between, most of the algorithms are designed based on the algorithmic paradigm of k-means, e.g., k-Prototypes [11]. Often in this category not only the number of clusters k but also the weighting between numerical

and categorical attributes in clustering has to be specified by the user. Among them, K-means-mixed (KMM) [1] avoids weighting parameters by an optimization scheme learning the relative importance of the single attributes during runtime, although it needs the number of clusters k as input parameter. KMM employs data conversion and discretize numerical attributes into categorical ones and then calculate the interactions in terms of categorical ways. Almost similarly, SpectralCAT [6] and CoupledMC [18] both conduct k-means clustering on continuous features and use a validity index to choose clustering label as new continuous features. These methods calculate the couplings based on discretized numerical data which may lead to information loss due to failure in capturing the distribution of the continuous data.

Following a mixture of Gaussian distributions, model-based clustering algorithms have been also proposed for mixed-type data. In between, ClustMD [13] is developed using a latent variable model and employing an expectation maximization (EM) algorithm to estimate the mixture model. Yet, this algorithm has a certain Gaussian assumption which does not have to be necessarily fulfilled. On the other hand, clustering algorithms for mixed-data often do not properly model dependencies and are limited to modeling meta–Gaussian distributions. Copulas, that provide a modular parameterization of joint distributions, can model a variety of dependencies, but their use with discrete data remains limited due to challenges in parameter inference. Authors in [15] use Gaussian mixture copulas, to model complex dependencies beyond those captured by meta–Gaussian distributions, for clustering. However, this approach may not only result in information loss but also fail to capture the discriminative information between objects.

Some of the approaches utilize the unique characteristics of any data type to avoid the drawbacks of converting a data type to another one. Profiting of the concept hierarchy, these algorithms introduce an integrative distance measure applicable for both numerical and categorical attributes. The algorithm DH [9] proposes a hierarchical clustering algorithm using a distance hierarchy which facilitates expressing the similarity between categorical and numerical values. As another method, MDBSCAN [2] employs a hierarchical distance measure to introduce a general integrative framework applicable for the algorithms which require a distance measure, e.g., DBSCAN. On the other hand, information-theoretic approaches have been proposed to avoid the difficulty of estimating input parameters. These algorithms regard the clustering as a data compression problem by hiring the minimum description length (MDL). The cluster model of these algorithms comprises joint coding schemes supporting numerical and categorical data. The MDL principle allows balancing model complexity and goodness-of-fit. INCONCO [14] and Integrate [5] are two representative for mixed-type clustering algorithms in this family. While Integrate has been designed for general integrative clustering, INCONCO also supports detecting mixed-type attribute dependency patterns.

Recently, deep neural networks are widely used for clustering. Among them, authors in [12] propose an auto-instructive representation learning scheme to enable margin-enhanced distance metric learning for a discrimination-enhanced representation. Finally, they feed the learned representation into both partition-based (k-means) and density-based (DBSCAN) clustering methods.

6 Evaluation

In this section, we assess the performance of ClicoT compared to other clustering algorithms in terms of NMI which is a common evaluation measure for clustering results. NMI numerically evaluates pairwise mutual information between ground truth and resulted clusters scaling between zero and one. We conducted several experiments evaluating ClicoT in comparison with KMM [1], INCONCO [14], DH [9], ClustMD [13], Integrate [5] and MDBSCAN [2]. In order to be fair in any experiment, we input the corresponding concept hierarchy to the algorithms which are not designed for dealing with it. That is, we encode the concept hierarchy as an extra attribute so that categorical values belonging to the same category have the same value in this extra attribute. Our algorithm is implemented in Java, and the source code as well as the data sets is publicly available[1].

6.1 Mixed-type clustering of synthetic data

In order to cover all aspects of ClicoT, we first consider a synthetic data set. Then, we continue experiments by comparing all algorithms in terms of the noise-robustness. Finally, we will discuss the runtime efficiency.

Clustering Results In this experiment, we evaluate the performance of all the algorithms on the running example (Fig. 1) while all parametric algorithms are set up with the right number of clusters. The data have two numerical attributes concerning the position of any data point and a categorical attribute showing the color of the points. Figure 6 shows the result of applying the algorithms where different clusters are illustrated by different colors. As it is explicitly shown in this figure, ClicoT, with NMI 1, appropriately finds the initially sampled three clusters where green, pink and blue are cluster-specific elements. Setting the correct number of cluster and trying various Gaussian mixture models, ClustMD results in the next accurate clustering. Although MDBSCAN utilizes the distance hierarchy, it is not able to capture the pink and green clusters. KMM cannot distinguish among various colors. Since two clusters pink and green

[1] https://tinyurl.com/ucp8289.

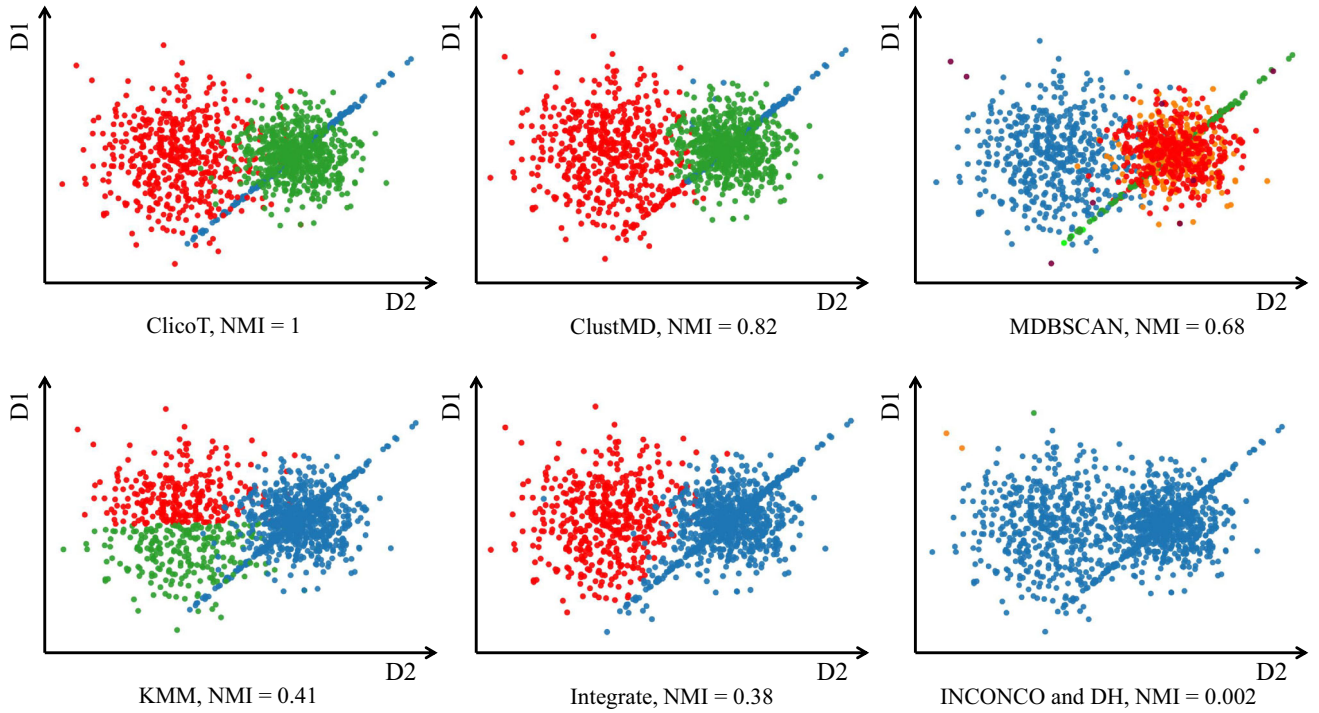
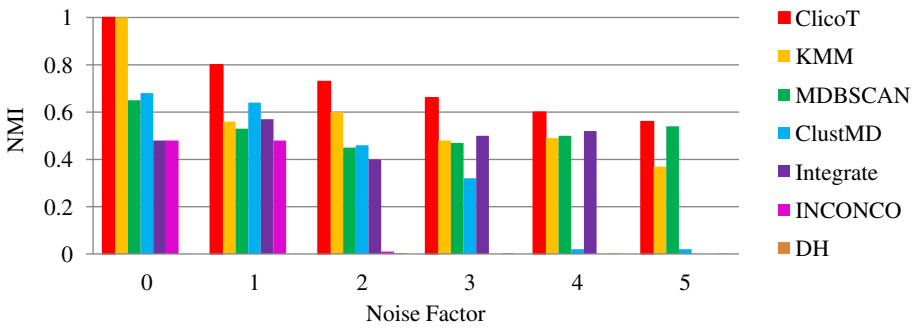


Fig. 6 Clustering results on the running example

Fig. 7 Comparing noise-robustness of ClicoT to other algorithms



are heavily overlapped, Integrate cannot distinguish among them. DH and INCONCO result on this data set inefficiently finding almost only one cluster.

Noise-robustness In this section, we benchmark noise-robustness of ClicoT w.r.t the other algorithms in terms of NMI by increasing the noise factor. To address this issue, we generate a data set with the same structure as the running example when adding another category, brown, to the categorical attribute color as noise. Regarding numerical attributes, we increase the variance of any cluster. We start from 5% noise (noise factor = 1) and iteratively increase the noise factor ranging to 5. Figure 7 clearly illustrates noise-robustness of ClicoT compared to others.

Flat Hierarchy In this section, we investigate the case when no appropriate hierarchy is considered. That is, we assume a flat concept tree with no hierarchy (e.g., Fig. 2) and run the following two experiments. Firstly, we focus on the running example introduced in Sect. 2 (see Fig. 1) and assume a flat hierarchy for the categorical attribute *Color* where no higher level concept categorizes the colors (Fig. 2).

As expected also observed from Fig. 8, ignoring a meaningful hierarchy for categorical attributes decreases the performance of ClicoT. However, ClicoT-flat (NMI = 0.60) is still comparable to MDBSCAN and more effective than KMM, Integrate, INCONCO and DH. In this data set, Cluster 3 (the line shape cluster illustrated by green circles in Fig. 1) highly overlaps two other clusters at some points. The data points in this cluster have the colors light green and dark green. As it is observed from the result of ClicoT-flat (Fig. 8), ignoring a meaningful hierarchy for the colors leads to an inefficiency in the sense that numerical attributes get cluster-specific and hence important while clustering. Therefore, parts of Cluster 3 which overlap with two other clusters (middle part and tail of Cluster 3) are wrongly grouped.

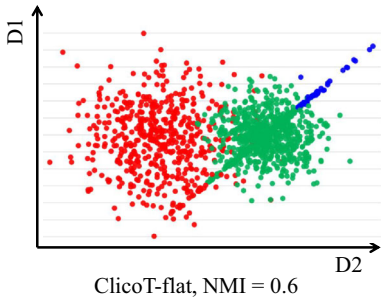


Fig. 8 Result of ClicoT applied on running example assuming a flat concept tree for colors (Fig. 2) (color figure online)

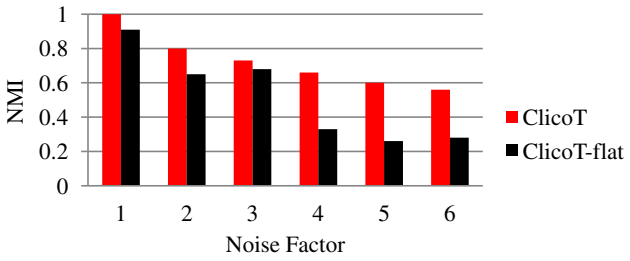


Fig. 9 Comparing ClicoT and ClicoT-flat assuming various noise factors

In the next investigation, we repeat the noise experiment applying ClicoT-flat. Here, the goal is to compare ClicoT and ClicoT-flat in various cases when the data set gets more noisy iteratively. As the plot in Fig. 9 depicts, ClicoT is always more effective in comparison with ClicoT-flat, although the performance of ClicoT-flat is still comparable in the beginning when the noise factor is smaller. It again approves the role of a meaningful hierarchy in order to increase the efficiency.

Scalability To evaluate the efficiency of ClicoT w.r.t the other algorithms, we generated a 10-dimensional data set (5 numerical and 5 categorical attributes) with three Gaussian clusters. Then, respectively, we increased the number of objects ranging from 2000 to 10,000. In the other case, we generated different data sets of various dimensionality ranging from 10 to 50 where the number of objects is fixed. Figure 10 depicts the performance of all algorithms in terms of the runtime complexity. Regarding the first experiment on the number of objects, ClicoT is slightly faster than others while increasing the dimensionality Integrate performs faster. However, the runtime of this algorithm highly depends on the number of clusters k initialized in the beginning (we set $k = 20$). That is, this algorithm tries a range of k and outputs the best results. Therefore, by increasing k the runtime is also increasing.

Proportion How would ClicoT behave when various proportions of categorical and numerical attributes are considered in the data sets? What happens when the majority of attributes are numerical and vice versa? In this experiment, we address the mentioned questions and generate various synthetic data sets each of which having a different proportion of categorical and numerical attributes. The x-axis in Fig. 11 shows the proportion factor, while for factor 1, for example, we generate 2 numerical and 2 categorical attributes. In Fig. 11, the yellow bins show the case when we increase the number of numerical attributes while the categorical attributes are set two, e.g., factor $3 = \frac{6 \quad numerical \quad attributes}{2 \quad categorical \quad attributes}$. For the categorical attributes, we assume a flat hierarchy with 3 various categories in every experiment. Analogously the green bins in Fig. 11 illustrate the results of applying ClicoT when the proportion factor is achieved by $proportion = \frac{\#categorical \quad attributes}{\#numerical \quad attributes}$.

As observed in Fig. 11 having various number of numerical or categorical attributes as well as different proportions does not influence the performance of our proposed algorithm. ClicoT is very well designed to deal with any kind of data structures since it always utilizes cluster-specific attributes and marks the most relevant attributes as specific.

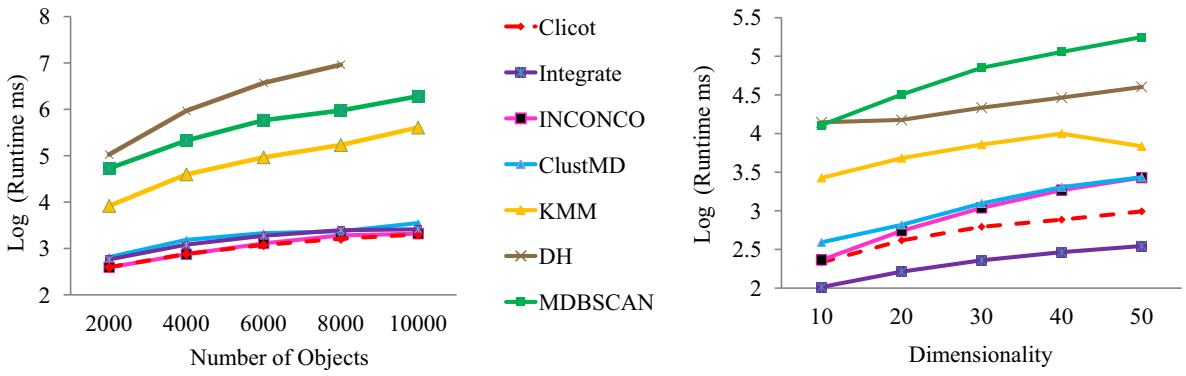


Fig. 10 Investigating the runtime efficiency of ClicoT in comparison with other algorithms. Two various cases are considered: **a** when the number of objects is increasing while the dimensionality is fixed, **b** when the number of objects is fixed and the dimensionality (number of categorical and numerical attributes) is increasing

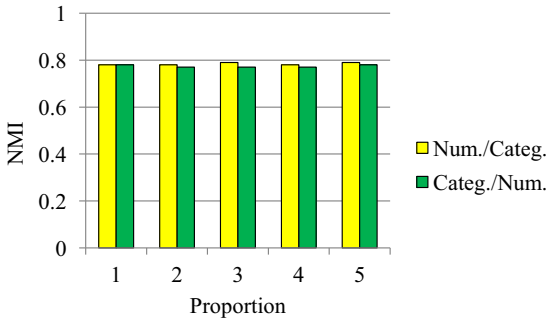


Fig. 11 Synthetic experiments to investigate the impact of various proportions of categorical and numerical attributes

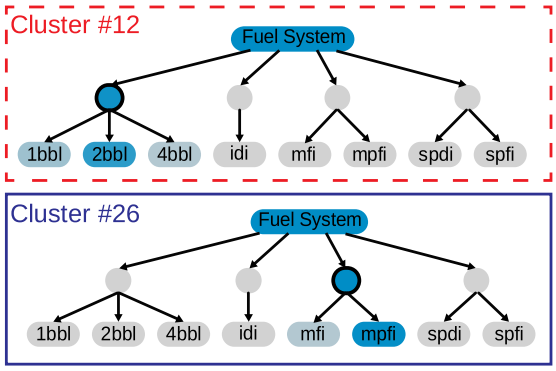


Fig. 12 Cluster-specific categories for Cluster 12 and Cluster 26 w.r.t. the categorical attribute *Fuel System*

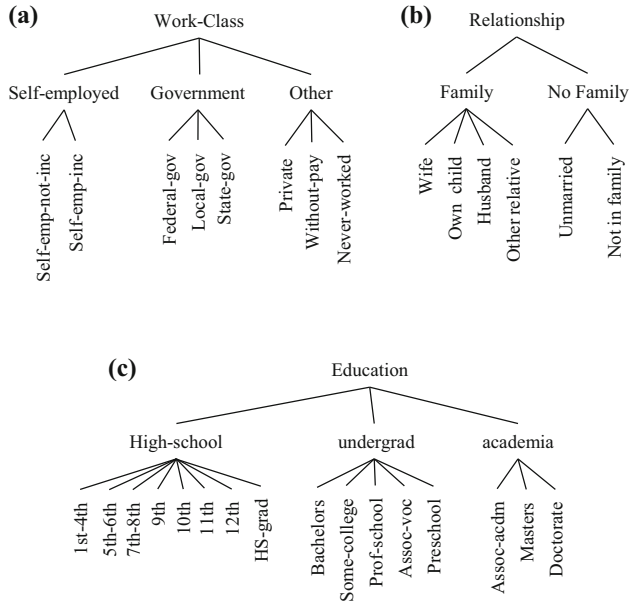


Fig. 13 Concept tree for 3 categorical attributes of Adult data set

Table 1 Cluster specific attributes for attribute *Relationship* based on the most deviation

	C_2	C_3
Family	− 0.24	0.359
Wife	0.025	− 0.047
Own child	0.111	− 0.154
Husband	− **0.398**	**0.59**
Other relative	0.02	− 0.028
No family	0.24	− 0.359
Unmarried	0.074	− 0.105
Not in family	0.165	− 0.253

Bold numbers in the table show maximum deviations corresponding to each attribute

6.2 Experiments on real-world data

Finally, we evaluate clustering quality and interpretability of ClicoT on real-world data sets. We used *MPG*, *Automobile* and *Adult* data sets from the UCI Repository [7] as well as *Airport* data set from the public project *Open Flights*[2].

MPG MPG is a slightly modified version of the data set provided in the StatLib library. The data concern city-cycle fuel consumption in miles per gallon (MPG) in terms of 3 categorical and 5 numerical attributes consisting of different characteristics of 397 cars. We consider MPG ranging from 10 to 46.6 as the ground truth and divide the range to 7 intervals of the same length. Considering a concept hierarchy for the name of cars, we group all the cars so that we have three branches: European, American and Japanese cars. Moreover, we divide the range of model year attribute to three intervals: 70–74, 75–80 and after 80. We leave the third attribute as a flat concept hierarchy since there is no meaningful hierarchy between variation of cylinders. Comparing ClicoT (NMI = 0.4) to the other algorithms INCONCO (0.17), KMM (0.37), DH (0.14), MDBSCAN (0.02), ClustMD (0.33) and Integrate (0), ClicoT correctly finds 7 clusters each of which is compatible with one of the MPG groups. Cluster 2, for instance, is compatible with the first group of MPGs since the frequency of the first group in this cluster is 0.9. In this cluster, American cars with the frequency of 1.0 and cars with 8 cylinders with the frequency of 1 and model year in first group (70–74) with the frequency of 0.88 are selected as cluster-specific elements.

Automobile This data set provides 205 instances with 26 categorical and numerical attributes. The first attribute defining the risk factor of an automobile has been used as class label. Altogether there are 6 different classes. Due to many missing values, we used only 17 attributes. Comparing the best NMI captured by every algorithm, ClicoT

[2] http://openflights.org/data.html.

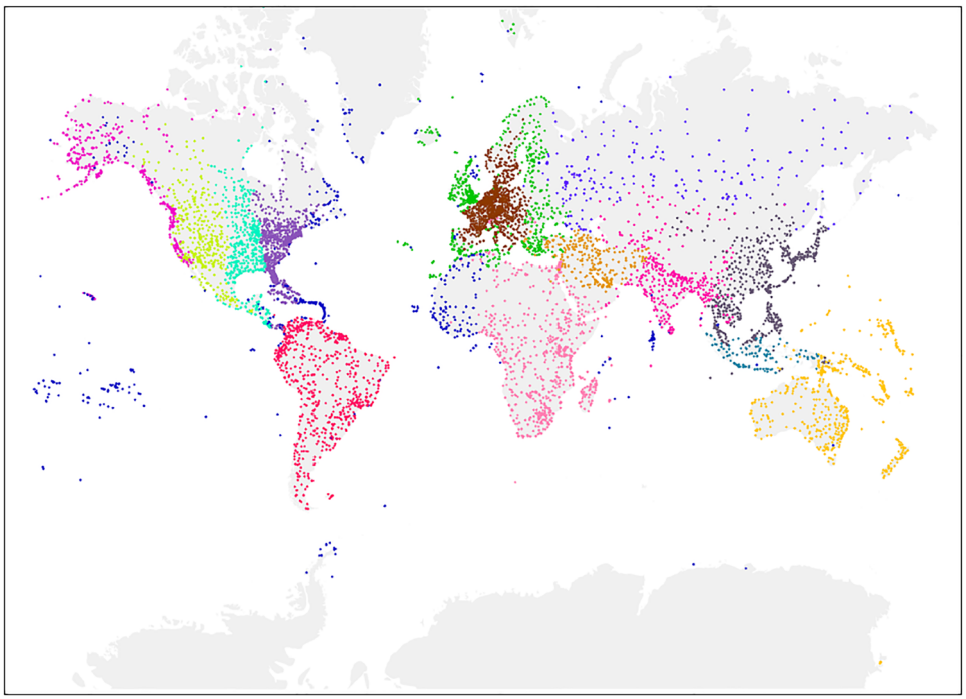


Fig. 14 Result of ClicoT on Open Flights data set

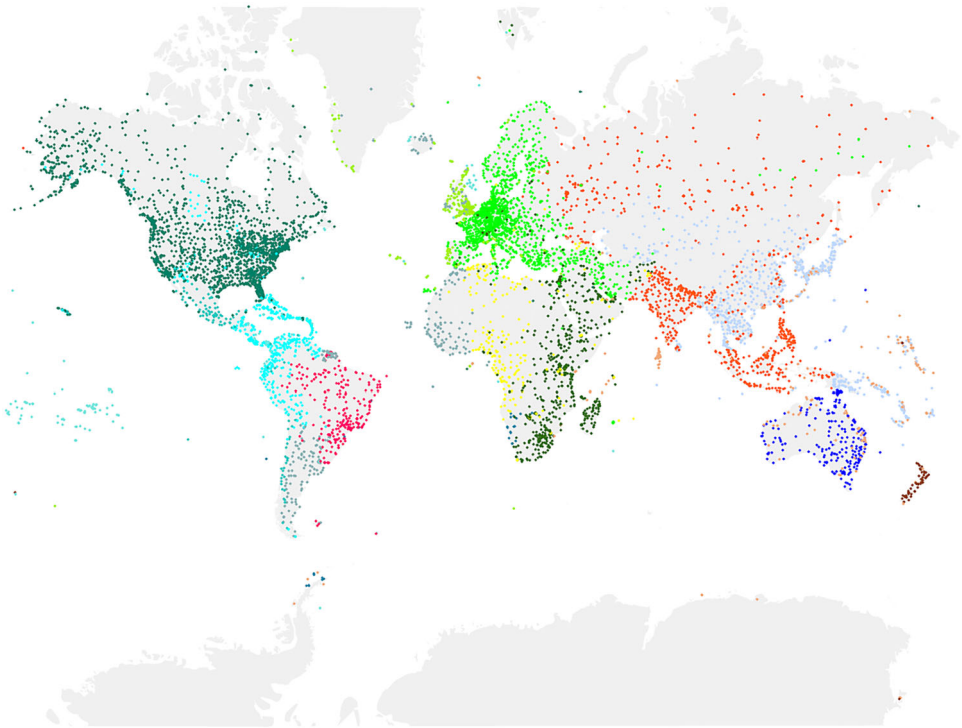


Fig. 15 Result of KMM on Open Flights data set

(NMI = 0.38) outperforms KMM (0.23), INCONCO (0.20), Integrate (0.17), DH (0.04), ClustMD (0.16) and MDBSCAN (0.02). Furthermore, ClicoT gives an insight into the interpretability of the clusters. As illustrated in Fig. 12, Cluster 12, for instance, is characterized mostly by the fuel system of *2bbl*, but also by *1bbl* and *4bbl*. Also we see that Cluster 26 is consisting of both *mpfi* and slightly of *mfi*, too. Concerning the risk analysis this clustering serves, ClicoT allows to recognize which fuel systems share the same insurance risk.

Adult Data Set Adult data set without missing values, extracted from the census bureau database, consists of 48,842 instances of 11 attributes. The class attribute Salary indicates whether the salary is over 50K or lower. Categorical attributes consist of different information, e.g., work-class,

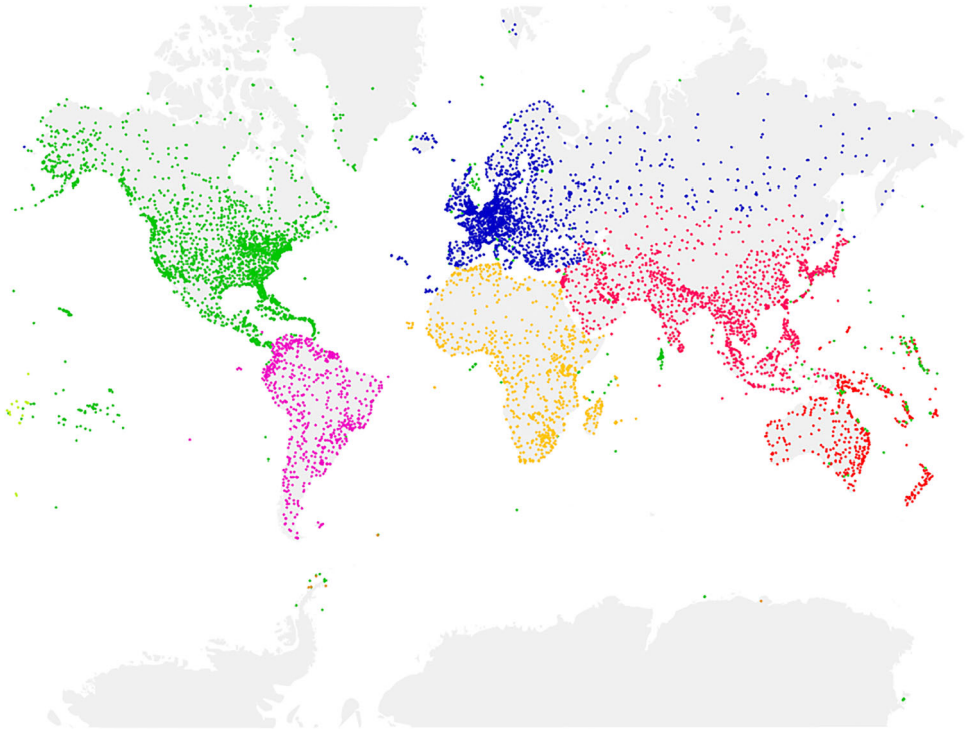


Fig. 16 Result of MDBSCAN on Open Flights data set

education, occupation. A detailed concept hierarchy is provided in Fig. 13. Although compared to INCONCO (0.05), ClustMD (0.0003), MDBSCAN (0.004), DH (0) and Integrate (0), our algorithm ClicoT (0.15) outperforms all other algorithms except KMM (0.16) which is slightly better. In order to give more insights into discovered clusters, we use two other evaluation measures, *Categorical Utility* (CU) and *Rand Index*, and compare the result of ClicoT to KMM which in this experiment is slightly more efficient in terms of NMI.

Before any comparison, we briefly explain about new evaluation strategies. Rand index is one of the most popular external clustering validation indices. Assuming P as the true clustering of data set with N data objects and C as clustering result, for each pair of data objects x_i and x_j, there are four different cases:

- *Case 1* x_i and x_j belong to the same clusters of C and the same category of P
- *Case 2* x_i and x_j belong to the same clusters of C but different categories of P
- *Case 3* x_i and x_j belong to different clusters of C but the same category of P
- *Case 4* x_i and x_j belong to different clusters of C and different categories of P

Let a, b, c, d correspond to number of pairs for the first to fourth cases and L is the total number of pairs ($L = a + b + c + d$). Thus, *Rand index* is defined as follows, with larger values indicating better results:

$$Rand\ index = \frac{a+d}{L}$$

On the other side, in order to evaluate the clustering result in terms of categorical attributes we apply the *categorical utility* criterion. CU attempts to maximize both the probability that two patterns in the same cluster have attribute values in common and the probability that patterns from different clusters have different values:

$$\text{CU}=\sum_k \left(\frac{C_k}{\mathcal{DB}} \sum_{A\in\mathcal{A}} \sum_j [P(A=A_j|C_k)^2 - P(A=A_j)^2] \right)$$

where $P(A = A_j|C_k)$ is the conditional probability that attribute A has the value A_j given cluster C_k, and $P(A = A_j)$ is the overall probability of attribute i having A_j in the entire data set. Obviously, the higher the CU value, the better the clustering performs.

Considering the *Rand index* as the metric, ClicoT (0.592) performs almost the same as KMM (0.604). However, ClicoT (0.41) slightly outperforms KMM (0.39) in terms of CU. Meaning that, clusters resulted by ClicoT are more efficiently distinguished in terms of categorical attributes compared to KMM.

On the other side, a deeper look to the clusters found by ClicoT shows interesting and interpretable results. ClicoT

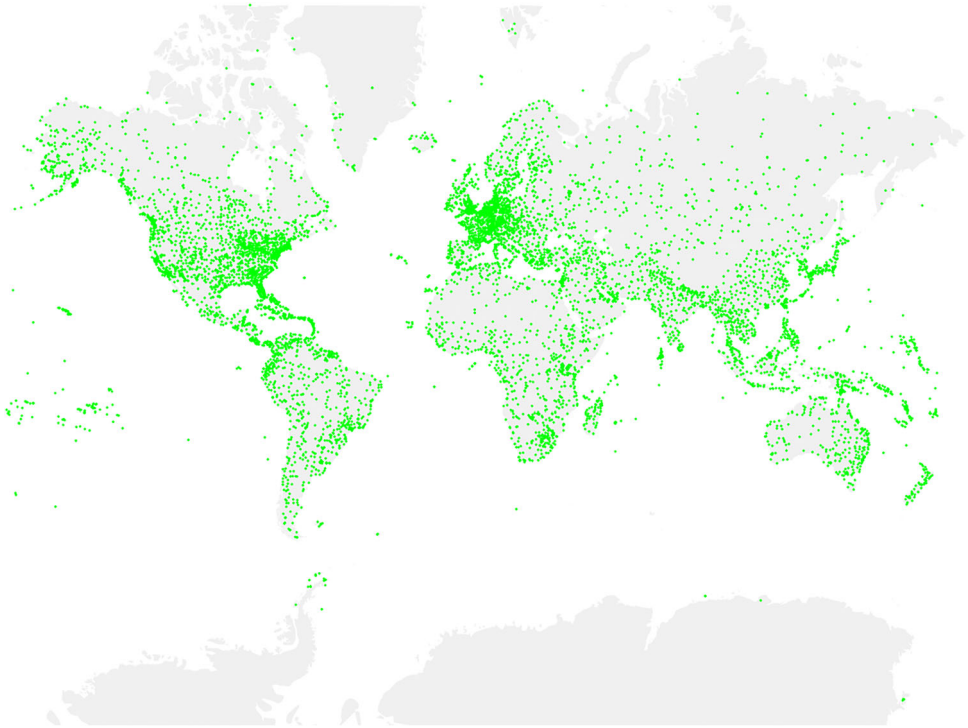


Fig. 17 Result of INCONCO and integrate on Open Flights data set

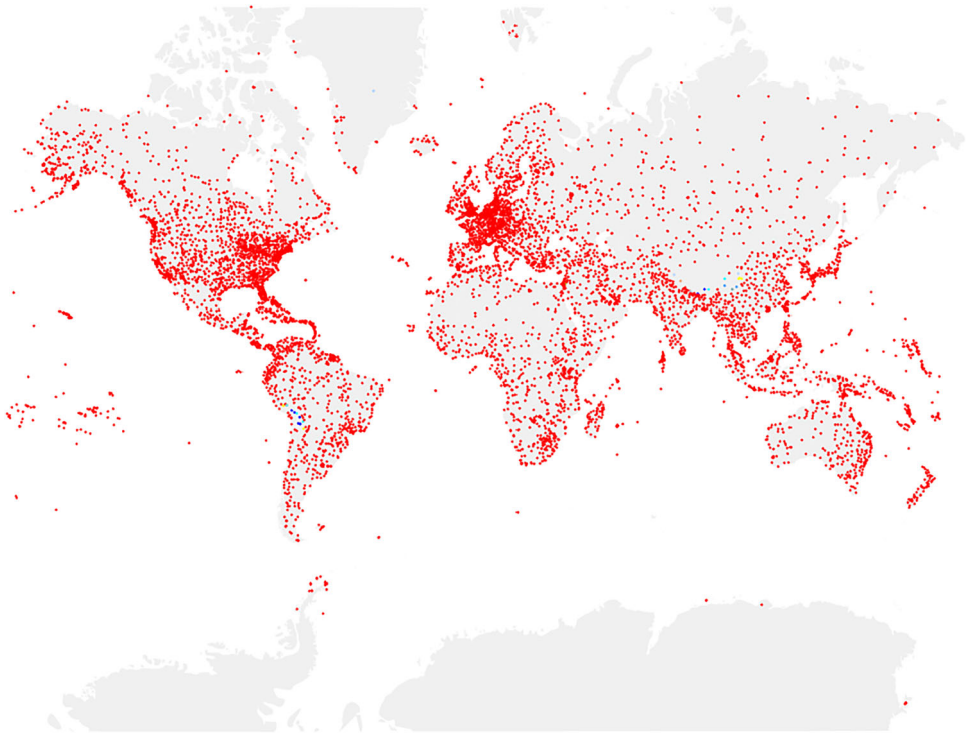


Fig. 18 Result of DH on Open Flights data set

finds 4 clusters in which Cluster 2, the biggest cluster, consists of almost 56% of objects. As Table 1 shows, in this cluster *Husband* is specified as the cluster-specific element, since it has the most deviation in terms of coding cost, but negative. The probability of instances having *Husband* as categorical value and the salary $<= 50K$ is zero in this cluster. Therefore, along with the negative deviation this means that in Cluster 2 persons with the role as husband in a family earn more than $50K$.

According to this table, for Cluster 3 *Husband* is cluster-specific as well. It has the most positive deviation and also the highest probability in this cluster, 0.99 which approves specifying this categorical value as cluster specific. In this cluster, almost 60% of persons having *Husband* as a role

earn more than 50k per year which is compatible with the overall distribution of the salary in Cluster 2 (Table 1).

Open Flights Data Set The public project Open Flights provides worldwide information about airports, flights and airlines. Here, we consider instances of airports in order to carry out a cluster analysis. The data set consists of 8107 instances each of which represents an airport. The numeric attributes show the longitude and latitude , the sea height in meters and the time zone. Categorical attributes consist of the country, where the airport is located and the day light saving time. We constructed the concept hierarchy of the country attribute so that each country belongs to a continent. Since there is no ground truth provided for this data set, we interpret the result of ClicoT (Fig. 14) and illustrate the result of applying other algorithms (Figs. 15, 16, 17, 18). INCONCO, Integrate and DH found almost only one cluster which makes any interpretation for this result nonsense (Figs. 17 and 18).

Clustering results illustrated in Fig. 14 consist of 15 clusters showing that ClicoT appropriately grouped almost geographically similar regions in the clusters. Therefore, we set the number of clusters for the other algorithms which required a user to specify it as 15. Starting from west to east, North American continent divided into five clusters. Obviously here the attribute of the time zone was chosen as specific because the clusters are uniquely made according to this attribute. In comparison with ClicoT, KMM found almost one cluster here and grouped all airports with different time zones together (Fig. 15). On the other hand, MDBSCAN groups all the airports continentally ignoring the time zone while the same concept hierarchy as ClicoT is given (Fig. 16).

Moving to the south, ClicoT pulled a plausible separation between South and North America. Considering South America as cluster-specific element and due to the rather low remaining airport density of South America ClicoT combined almost all of the airports to a cluster (red). In Western Europe, there are some clusters, which can be distinguished by their geographic location. Additionally, many airports around and in Germany are be grouped together.

7 Conclusion

To conclude, we have developed and demonstrated that ClicoT is not only able to cluster mixed-typed data in a noise-robust manner, but also yielded most interpretable cluster descriptions. By using data compression as the general principle ClicoT automatically detects the number of clusters within any data set without any prior knowledge. Moreover, the experiments impressively demonstrated that clustering can greatly benefit from a concept hierarchy. Therefore, ClicoT excellently complements the approaches for mining mixed-type data.

Acknowledgements Open access funding provided by University of Vienna.

References

1. Ahmad, A., Dey, L.: A k-mean clustering algorithm for mixed numeric and categorical data. Data Knowl. Eng. **63**, 503–527 (2007)
2. Behzadi, S., Ibrahim, M.A., Plant, C.: Parameter free mixed-type density-based clustering. In: Database and Expert Systems Applications (DEXA) (2018)
3. Behzadi, S., Müller, N.S., Plant, C., Böhm, C.: Clustering of mixed-type data considering concept hierarchies. In: Advances in Knowledge Discovery and Data Mining, pp. 555–573. Springer International Publishing, Cham (2019)
4. Böhm, C., Faloutsos, C., Pan, J., Plant, C.: Robust information-theoretic clustering. In: KDD (2006)
5. Böhm, C., Goebl, S., Oswald, A., Plant, C., Plavinski, M., Wackersreuther, B.: Integrative parameter-free clustering of data with mixed type attributes. In: PAKDD (1), pp. 38–47 (2010)
6. David, G., Averbuch, A.: Spectralcat: categorical spectral clustering of numerical and nominal data. Pattern Recognit. **45**(1), 416–433 (2012)
7. Frank, A., Asuncion, A.: UCI machine learning repository (2010). http://archive.ics.uci.edu/ml
8. He, Z., Xu, X., Deng, S.: Clustering mixed numeric and categorical data: a cluster ensemble approach. CoRR arXiv:cs/0509011 (2005)
9. Hsu, C.C., Chen, C.L., Su, Y.W.: Hierarchical clustering of mixed data based on distance hierarchy. Inf. Sci. **177**(20), 4474–4492 (2007)
10. Hsu, C.C., Chen, Y.C.: Mining of mixed data with application to catalog marketing. Expert Syst. Appl. **32**(1), 12–23 (2007)
11. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min. Knowl. Discov. **2**, 283–304 (1998)
12. Jian, S., Hu, L., Cao, L., Lu, K.: Metric-based auto-instructor for learning mixed data representation (2018)
13. Mcparland, D., Gormley, I.C.: Model based clustering for mixed data: ClustMD. Adv. Data Anal. Classif. **10**(2), 155–169 (2016)
14. Plant, C., Böhm, C.: Inconco: interpretable clustering of numerical and categorical objects. In: KDD, pp. 1127–1135 (2011)
15. Rajan, V., Bhattacharya, S.: Dependency clustering of mixed data with gaussian mixture copulas. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16, pp. 1967–1973. AAAI Press (2016)
16. Rissanen, J.: A universal prior for integers and estimation by minimum description length. Ann. Stat. **11**(2), 416–31 (1983)
17. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: is a correction for chance necessary? In: ICML (2009)

18. Wang, C., Chi, C.H., Zhou, W., Wong, R.: Coupled interdependent attribute analysis on mixed data. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, p. 1861–1867. AAAI Press (2015)
19. Yin, J., Tan, Z.: Clustering mixed type attributes in large dataset. In: ISPA, pp. 655–661 (2005)

CHAPTER 7

Paper C: Parameter Free Mixed-type Density-based Clustering

Authors Contributions:

- **Sahar Behzadi.** Proposing the main idea, developing the algorithm; Cooperation on the implementation and conducting experiments; Writing the paper.
- **Mahmoud A. Ibrahim.** Cooperation on developing the algorithm; Implementation; Conducting experiments.
- **Claudia Plant.** Supervision during development and evaluation of the algorithm as well as writing the paper.

Parameter Free Mixed-Type Density-Based Clustering

Sahar Behzadi[1(✉)], Mahmoud Abdelmottaleb Ibrahim[1], and Claudia Plant[2]

[1] University of Vienna, Vienna, Austria
{sahar.behzadi,a1309720}@univie.ac.at
[2] ds:UniVie, University of Vienna, Vienna, Austria
claudia.plant@univie.ac.at

Abstract. Nowadays many applications generate mixed data objects consisting of numerical and categorical attributes. Simultaneously dealing with mixed objects is more challenging and various approaches convert one type to another one to face this issue. But in many cases this leads to information loss. Therefore integrating categorical and numerical attributes sounds reasonable since it keeps the original format of any attribute. In this paper we focus on clustering and especially density-based clustering as one of the well-known clustering approaches well-performed on arbitrary shape clusters. Density-based clustering algorithms require a distance measure to discover dense regions. Therefore we introduce the distance hierarchy as a distance measure appropriate for both categorical and numerical attributes. However setting the parameters regarding any parametric clustering algorithm could be another issue. Therefore we employ minimum description length principle to automate this process.

Keywords: Density-based clustering · Distance hierarchy
Parameter free clustering · Minimum description length

1 Introduction

Clustering is one of the various data mining tasks which groups most similar data objects together. Many well-known clustering algorithms (e.g. K-means [10] or DBSCAN [11]) measure the euclidean distance as a similarity measure in the sense that the closest object to a specific object is the most similar one. Although this approach sounds reasonable for pure numerical data, considering a mixture of categorical and numerical attributes might challenge its efficiency. However many applications generate a mix of data objects consisting of numerical and categorical attributes.

It is already well-understood that converting one type to another one is not sufficient since it might lead to information loss. Moreover relations between values such as a certain order are artificially introduced. For instance, assuming various regions such as China or United States one can't really define an order or the distances between them.

S. Hartmann et al. (Eds.): DEXA 2018, LNCS 11030, pp. 19–34, 2018.
https://doi.org/10.1007/978-3-319-98812-2_2

In the meantime integrating categorical and numerical attributes without any conversion seems reasonable since it keeps the original format of any attribute. Considering the fact that almost always there is a natural hierarchy regarding categorical values we introduce distance hierarchy as a distance measure available for both types of attributes. A distance hierarchy extends the concept hierarchy by associating a weight to any link [8]. Also one could assume a distance hierarchy corresponding to any numerical attribute resulting in the euclidean distance.

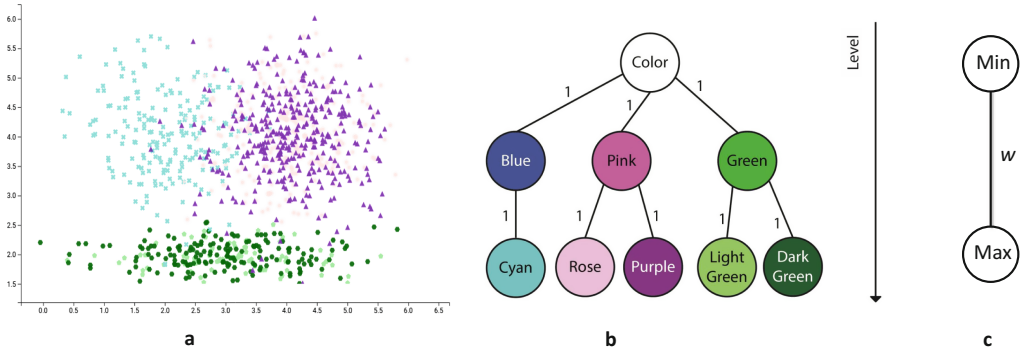


Fig. 1. Synthetic dataset. (a) Three generated clusters with two numerical and one categorical attributes (color). (b) A natural hierarchy between colors. (c) A distance hierarchy corresponding to numerical attributes. (Color figure online)

Figure 1a illustrates a generated dataset comprised of two numerical attributes showing the position of each object and a categorical attribute containing several colors. With respect to the natural hierarchy among various colors Fig. 1b shows the corresponding distance hierarchy to the categorical attribute Color while labels are related to the weights. In this example we assume the same weight for all the links however one could assign different weights due to more information on dataset. To compute the distance between categorical values we utilize the distance hierarchy in the sense that a distance hierarchy provides insights of objects. For instance Rose and Purple are more similar than Rose and Cyan w.r.t. the distance hierarchy. However it is confirmed by the nature of colors since Rose and Purple are derivations of Pink. Preserving the same structure Fig. 1c depicts a distance hierarchy corresponding to the numerical attribute in this example. It has only two nodes and returns the euclidean distance as the distance between two numerical values.

By profiting the distance hierarchy we introduce a general framework appropriate for clustering algorithms which need a distance measure as one of the prerequisites. There are many existing clustering approaches e.g. partition-based, density-based, hierarchical clustering to mention a few. In between density-based algorithms are well-known due to their performance on different (even arbitrary shaped) datasets. DBSCAN is one of the most effective representatives for this approach which captures dense groups of objects as clusters. The basic idea is that if a particular point belongs to a cluster, it should be near to lots of other points in that cluster. In this paper we select DBSCAN to compare results of

the proposed framework with state-of-the-art algorithms dealing with mixed-data types.

DBSCAN requires two parameters, a positive real number ϵ and a natural number MinPts showing the radius and the density of a neighborhood respectively. Although DBSCAN is well-known due to its performance, setting appropriate parameters that meet all the aspects of a dataset could be challenging. To face this challenge we propose a parameter free approach by means of *Minimum Description Length* (MDL) principle which links the best clustering with the strongest compression of data.

In this paper we develop a parameter-free mixed-type clustering algorithm modifying DBSCAN which is based on an optimization strategy utilizing MDL. Our contributions consist of:

- **An integrated framework:** We introduce distance hierarchy as a distance measure suitable for both categorical and numerical attributes in the sense that we integrate both types.
- **DBSCAN for mixed-data:** We modify DBSCAN, a well-known density-based clustering algorithm, so that it is applicable for mixed-type data.
- **Parameter-free clustering:** Utilizing MDL principle we introduce a fast noise-robust algorithm without specifying parameters.

In Sect. 3 we introduce the problem specification and introduce a framework suitable for mixed datasets by means of distance hierarchies. In the following we modify DBSCAN in Sect. 4. Section 5 defines a non-parametric version of MDBSCAN following principles of MDL. Finally in Sect. 6 we evaluate our algorithm comparing to others.

2 Related Work

Nowadays to analyze many real applications one need to deal with mixed-type data represented by numerical and categorical attributes. For example, the approaches K-Means-Mixed (KMM) [1], k-Prototypes [9], INCONCO [15], Integrate [3], CFIKP [18], CAVE [7], DH [4] as well as CEBMDC [19].

Most of these approaches use the algorithmic paradigm of k-Means. Often, e.g. in k-Prototypes, not only the number of clusters k, but also the weighting between numerical and categorical attributes should be specified.

The algorithm KMM needs the number of clusters k as input parameter but avoids weighting parameters by an optimization scheme learning the relative importance of the single attributes during runtime. To avoid the difficulty of estimating input parameters, information-theoretic approaches have been proposed. These algorithms (e.g. INCONCO and Integrate) are based on the idea of data compression. The cluster model of these algorithms comprises joint coding schemes supporting numerical and categorical data. The MDL principle allows balancing model complexity and goodness-of-fit. While Integrate has been designed for general integrative clustering, INCONCO also supports detecting mixed-type attribute dependency patterns. The algorithm DH [4] proposes

a hierarchical clustering algorithm using a distance based on the concept hierarchy which facilitates expressing the similarity between categorical values and also unifies distance measuring of numerical and categorical values.

3 Mixed-Data Framework for Clustering

Clustering is the task of grouping different objects of a dataset $\mathcal{DB}$ into k clusters $\mathcal{C} = \{C_1, C_2, ..., C_k\}$. Objects in the same group (cluster) are more similar to each other than to those in other groups (clusters). As mentioned before there are many efficient clustering algorithm that require a distance measure as one the prerequisites. However finding an appropriate distance measure applicable for both categorical and numerical values at the same time is not a trivial task. In this section we introduce a distance measure based on the natural concept hierarchy related to any attribute. A distance hierarchy avoids loss of information by preserving the natural original orders.

Considering a mixed-type data we assume an object O consists of m categorical attributes $\mathcal{A} = \{A_1, A_2, ..., A_m\}$ and d numerical attributes $\mathcal{X} = \{X_1, X_2, ..., X_d\}$. For a categorical attribute A_i, we denote its domain by $Dom(A_i)$ and different categorical values by ${A_i}^j$. According to the natural hierarchy within categorical or numerical values we assume a distance hierarchy corresponding to any attribute. In the following we define the distance hierarchy and introduce a general framework for clustering.

3.1 Distance Hierarchy

Basically a concept hierarchy (tree) consists of concept nodes and links, in which leaf nodes represents more specific concepts while parent nodes are general concepts. Regarding the i-th attribute a distance hierarchy, denoted by $DH_i = (N, \mathcal{E}, W)$, extends the corresponding concept tree by associating a weight to each link. The definition of distance hierarchy in this paper is inspired by [8].

A DH_i has the following properties:

1. DH_i consists of a set of nodes $N = \{n_1, n_2, ..., n_s\}$ and a set of edges $\mathcal{E} = \{e_1, e_2, ..., e_{(s-1)}\}$, where n_j is a parent of n_z if $(n_j, n_z) \in \mathcal{E}$. W denotes the set of weights assigned to any link to facilitate the computation of distance between values.
2. Each node n_j is a concept and represents a sub-category of its parent. Usually data objects are distributed in leaf level. The root node represents associated attribute respectively.
3. The level $l(n_j)$ of a node n_j is the height of the descendant sub-tree. If n_i is a leaf node (e.g. categorical values), then $l(n_j) = 0$. The root node is the attribute A_i which has the highest level, also called the height of the concept hierarchy.

There are many different ways to assign link weights of a distance hierarchy [6,13]. For simplicity in this paper we assign uniformly a constant weight to all links. Other alternatives and a complete investigation on weight assignment approaches is an interesting issue deserving further research in the future.

Let $x = (N_x, d_x)$ denote a point in a distance hierarchy comprising an *anchor* and a positive real value *offset* represented by N_x and d_x respectively. The anchor is a leaf node and the offset represents the distance from the root to x. In the following we explain how to compute the distance between any two categorical or numerical values.

A DH_i regarding to a numerical attribute X_i consists of only two nodes, a root Min and a leaf Max (e.g. Fig. 1c). The associated link weight w equals to the range of X_i, i.e., $w_i = (max_{X_i} - min_{X_i})$. Let $p = (Max, d_p)$ denote a point in a numerical distance hierarchy. Therefore the anchor is always Max and the offset d_p is the distance from the point to the root Min.

3.2 Distance Function and Framework

To clearly define the distance function we need the following definitions:

Definition 1. ***Ancestor.*** *A point p is an ancestor of q if p is one of the nodes existing on the path from q to the root in the corresponding distance hierarchy.*

Definition 2. ***Lowest Common Ancestor*** *(LCA). Considering two nodes p and q in a distance hierarchy the lowest common ancestor or $LCA(p, q)$ is defined as p if p is an ancestor of q otherwise the deepest tree node that is an ancestor of p and q.*

Definition 3. ***Lowest Common Point*** *(LCP). If $p = q$ the lowest common point or $LCP(p, q)$ is defined as p (or q) otherwise $LCA(p, q)$.*

Now we are well-equipped to introduce the distance function. Let $dist(p, q)$ denote the distance between two points p and q w.r.t. the distance hierarchy where p and q could be either categorical or numerical values.

$$dist(p, q) = d_p + d_q - 2d_{LCP(p,q)} \tag{1}$$

where $LCP(p, q)$ is the lowest common point of p and q according to the distance hierarchy and $d_{LCP(p,q)}$ is the distance between the least common point and the root.

Figure 2 depicts an example how to compute the distance between two points $W = (Rose, 2), X = (Purple, 2)$ by means of the distance hierarchy corresponding to the categorical attribute color. W and X belong to the same category and as illustrated in Fig. 2a Pink is the lowest common ancestor of W and X. Therefore $LCP(p, q) = LCA(p, q)$ and $d_{LCP(p,q)} = 1$ w.r.t. Equation 1 and finally $dist(X, W) = |2 + 2 - 2 * 1| = 2$.

However considering $Y = (Cyan, 2)$ and $W = (Rose, 2)$ existing in different categories they naturally should have bigger distance which is approved by our

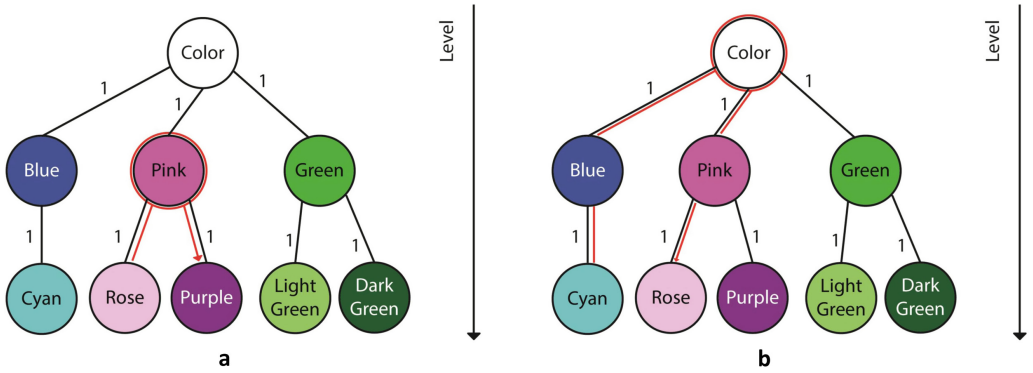


Fig. 2. Computing the distance between colors. (a) Distance between Rose and Purple, $LCA(Rose, Purple) = Pink$. (b) Distance between Cyan and Rose, $LCA(Rose, Cyan) = Color$. (Color figure online)

distance metric. Looking at Fig. 2b one could easily find the least common point of Y and W which is the root node (Color). Therefore the distance between W and Y is $|2 + 2 - 2 * 0| = 4$.

To introduce a general framework for clustering we require mapping objects to distance hierarchies. As mentioned before any attribute of a data object is associated with a distance hierarchy. Let $o = [o_1, o_2, ..., o_n]$ denote an object with n categorical and numerical attributes and let $DH = \{DH_1, DH_2, ..., DH_n\}$ be the set of distance hierarchies associated to any attribute. o_i could be either a categorical value belonging to $Dom(A_i)$ or a numerical value. A categorical attribute A_i associates with DH_i in the way that the set of domain values of A_i corresponds to the leaf nodes of DH_i. As explained before a numerical attribute X_j associates with DH_j which is a degenerated hierarchy (See Sect. 3.1).

Any attribute value o_i is mapped by means of a mapping function h_i to a point in its associated distance hierarchy. For instance let o_i be a categorical value then the mapping $h_i(o_i)$ maps o_i to a leaf node $p = (o_i, d_{o_i})$ in the corresponding distance hierarchy DH_i. For a numerical value o_j the mapping $h_j(o_j)$ maps o_j to $p = (Max, o_j - Min_j)$ in DH_j.

Finally the distance between two mixed-type objects $o_1 = [o_{11}, o_{12}, ..., o_{1n}]$ and $o_2 = [o_{21}, o_{22}, ..., o_{2n}]$ is measured as follows:

$$d(o_1, o_2) = \left(\sum_{i=1,n} w_i (o_{1i} - o_{2i})^L \right)^{1/L} = \left(\sum_{i=1,n} w_i (h_i(o_{1i}) - h_i(o_{2i}))^L \right)^{1/L} \quad (2)$$

where by $L = 2$ the distance is similar to a weighted Euclidean distance.

4 Mixed-Type Density-Based Algorithm

During the previous section we introduced a framework to map a mixed-type object to a point in the associated distance hierarchy and finally we defined a

distance function to compute the distance between objects. As mentioned before the proposed framework is a general framework applicable for any clustering algorithm dealing with mixed-type datasets while it requires a distance metric inside its algorithm. For instance k-means algorithm [10], an efficient clustering algorithm to find Gaussian clusters, assign any object to the closest centroid in terms of distance between the point and any centroid. However applying such algorithms on mixed data types requires either converting attributes which lead to information loss or investigating an appropriate distance metric for both categorical and numerical values. Utilizing the proposed framework enables us to apply an efficient clustering algorithm, designed for pure types of attributes, for a hybrid case.

There are many efficient clustering algorithms dealing with only one type of attributes. In this paper we focus on density-based approaches due to their performance on different datasets (even arbitrary shaped). Particularly we modify DBSCAN [11] and call it **MDBSCAN**. DBSCAN is one of the well-known representatives for this approach which captures dense groups of objects as clusters. The basic idea is that if a particular point belongs to a cluster, it should be near to lots of other points in that cluster.

More specifically, DBSCAN needs a positive value ϵ showing the radius of a neighborhood around a point p and the minimum number of points in this ϵ-neighborhood denoted by $MinPts$. Then we start from a random point, find all the points within its ϵ-neighborhood then if the number of those points are bigger than $Minpts$ we build a cluster and keep adding points to it by considering the same procedure for each point in this neighborhood.

To capture the points belonging to the ϵ-neighborhood of an object p we need to compute the distance between all other objects w.r.t. p. This is exactly where our framework plays a role so that DBSCAN would be applicable for mixed-type datasets while Euclidean distance is not a suitable distance any more.

5 Parameter Free Clustering Algorithm

As explained in Sect. 4 DBSCAN requires two parameters, a positive value ϵ showing the radius of a neighborhood and the minimum number of points in this ϵ-neighborhood denoted by $MinPts$. Selection of a higher $Minpts$ leads to more dense clusters. Simultaneously a smaller ϵ forces points to be closer to each other so that they could be considered as a part of a cluster. Although DBSCAN is an efficient clustering algorithm, its efficiency highly depends on parameters while they are hard to specify in advance. Therefore investigating an approach to make this algorithm parameter-free tends to a more effective DBSCAN.

We regard this challenge as a data compression problem applying the principle of Minimum Description Length (MDL). In the following we explain how to make MDBSCAN parameter-free by utilizing (MDL) principle.

5.1 Minimum Description Length (MDL)

MDL is a well-known principle for estimating statistical informations and compressing of data. Regarding clustering as a data compression problem allows us a unifying view, naturally balancing the influence of categorical and numerical attributes in clustering. MDL allows integrative clustering by relating the concepts of likelihood and data compression. To maximize the data compression we assign a shorter description length to regular data objects and longer descriptions to outliers w.r.t. a coding scheme. Following the MDL principle [16], we encode not only the data but also the model itself and minimize the overall description length. The less number of clusters exist in a model the weaker the model fits to the data. On the other side, a model with more clusters tends to be more complex, but has a better fit to the data [3]. The MDL principle finds a natural trade-off between model complexity and goodness-of-fit and thereby avoids over-fitting. In this paper we refer to clustering results as a model associated with data. After clustering to find the compression measure or description length regarding the model we apply MDL as follows:

Definition 4. ***Description Length.*** *Let $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ denote the clustering result consisting of k clusters. The overall description length (DL) of the dataset DB is defined as:*

$$DL(\mathcal{DB}) = \sum_{C_i \in \mathcal{C}} DL(C_i)$$

where $DL(C_i)$ denotes the description length w.r.t. the cluster C_i and is defined as:

$$DL(C_i) = DL_c(\mathcal{X}) + DL_c(\mathcal{A}) + DL(model(C_i)) \tag{3}$$

The first two terms represent coding costs necessary for encoding the numerical and categorical attributes respectively using a specific coding scheme. The last term is the cost of model encoding.

As mentioned we need a coding scheme to find the coding or description length corresponding to any cluster. Huffman coding is one of the well-known coding schemes where the description length of a value o_i is defined by:

$$PDF(o_i).log_2 PDF(o_i)$$

where PDF stands for *Probability Distribution Function*. The output can be interpreted as the numbr of bits necessary to transfer information from a sender to a receiver via a communication chanel. Since the PDF is part of the codebook, any distribution function can be applied [2].

5.2 Coding Numerical Values

Since any distribution function is applicable in the proposed coding scheme and since selection of a specific PDF is not a severe restriction, for simplicity we select Gaussian distribution to illustrate numerical attributes. Thus, for any numerical attribute $X_i \in \mathcal{X}$ we assume $PDF(X_i) = \mathcal{N}(\mu_i, \sigma_i)$. More precisely for a numerical value x the Gaussian probability distribution function is defined as follows:

$$PDF(x) = \frac{1}{\sigma_i \sqrt{2\pi}} exp\left(-\frac{(x-\mu i)^2}{2\sigma i^2}\right).$$

where μ_i and σ_i are mean and variance computed by means of data objects in cluster C_i. Therefore the coding cost corresponding to numerical attribute X_i is:

$$DL(X_i) = -\sum_{x \in X_i} PDF(x).\log_2 PDF(x)$$

Finally based on Huffman coding scheme the first term of Eq. 3 (numerical coding cost) w.r.t. the cluster C_i is provided by:

$$DL_{C_i}(\mathcal{X}) = \sum_{X_i \in \mathcal{X}} DL(X_i)$$

5.3 Coding Categorical Values

The proposed Huffman coding scheme is applicable on categorical attributes as well. However we consider the frequency of any categorical value (leaf nodes in a distance hierarchy) as the associated probability distribution function to the categorical attribute. More precisely considering a categorical attribute $A_i \in \mathcal{A}$ then for any categorical value $A^j{}_i \in Dom(A_i)$ we define $PDF(A^j{}_i)$ in a cluster C_z as $\frac{|A^j{}_i|}{|C_z|}$. Thus, based on the coding scheme the cost of coding categorical attributes in a specific cluster C_z is defined as:

$$DL_{C_i}(\mathcal{A}) = \sum_{A_i \in \mathcal{A}} \sum_{A^j{}_i \in A_i} -PDF({A_i}^j).\log_2 PDF({A_i}^j)$$

5.4 Coding the Model

As mentioned considering MDL principles we encode not only the data but also the model itself and minimize the overall description length. So far we have explained how to encode the data consisting of two parts; categorical and numerical part. In this section we elaborate how to encode the model associated with the data so that, back to our example, a receiver has enough information to decode the data. Thus decoding the model one needs to know to which cluster

an object belongs and which parameters specify the cluster model. The first concept is called cluster id denoted by $IDCost$ and the last one is parameter cost denoted by $ParamCost$. The $IDCost$ follows the principle of Huffman coding which implies that we assign shorter bits to the larger clusters. Therefore the cluster id cost w.r.t. the cluster C_z is provided by $\log_2 \frac{|\mathcal{DB}|}{|C_z|}$ where $|C_z|$ shows the number of objects in cluster C_z.

Following the theory of MDL [17] and focusing on a specific cluster C_z the parameter cost to model all the objects in this cluster can be approximated by $\frac{P_z}{2}.\log_2|C_z|$. P_z denotes the number of parameters in cluster C_z required to encode the model. We already assumed a Gaussian distribution to model numerical attributes. Therefore any numerical attribute X_i is described by two parameters: μ_i and σ_i. Regarding categorical attributes we need to encode all the probabilities corresponding to categorical values. Therefore for an attribute A_i with $|A_i|$ categorical values the number of parameters required to be coded is $|A_i| - 1$.

Algorithm 1. Parameter free MDBSCAN

```
Min − DL :=0;
foreach radius ϵ ∈ range R do
    result = MDBSCAN(ϵ,4);
    DL := Compute the description length for results;
    if (DL < Min − DL) then
        Min − DL= DL;
        BestResult = result;
return BestResult;
```

5.5 Algorithm

As explained MDBSCAN requires two parameters to be specified: ϵ and $MinPts$. Referring to original DBSCAN algorithm [11] authors employ $k - dist$ graph to find the best parameter setting. However they claim that for $k > 4$ the $k - dist$ graphs do not differ significantly from the $4 - dist$ graph. Therefore they recommend to set the $MinPts$ as 4. In this paper we stay with the same strategy and fix $MinPts$ as 4 but varying the radius of a ϵ-neighborhood.

Algorithm 1 summarizes our proposed non-parametric MDBSCAN algorithm. Considering $MinPts = 4$ we apply MDBSCAN iteratively for a specific range of ϵ. At the end of each iteration the overall description length DL will be computed following the principle of MDL. Thus for various parameter setting (models) we achieve a comparison score by means of DL. Finally, at the end of iterations we select the parameter setting with the minimum DL i.e. the most compressed model resulting the best clustering.

6 Evaluation

To assess the efficiency and effectiveness of our non-parametric MDBSCAN we compare our proposed algorithm to state-of-the-art mixed-type clustering

algorithms. We selected K-Means-Mixed (KMM) [1], INCONCO [15] and DH [8] so that we can cover all aspects of MDBSCAN. In this section extensive experiments on synthetic and real datasets will demonstrate the advantages of MDBSCAN over other clustering algorithms. All algorithms are implemented in Java and the source code as well as the datasets are available here: https://tinyurl.com/ybqq35xc.

Normalized mutual information (NMI) [12] is an information theoretic evaluation measure for clustering results. In this paper we employ NMI to assess our algorithm in comparison to others. NMI numerically evaluates pairwise mutual information between ground truth and resulted clusters and continues normalizing by means of the entropy of either original or resulted clusters. NMI scales between zero and one representing a random and a perfect clustering, respectively.

6.1 Synthetic Data Experiments

By synthetically generating various datasets we aim to evaluate MDBSCAN covering different aspects e.g. effectiveness, noise-robustness, scalability (Fig. 3).

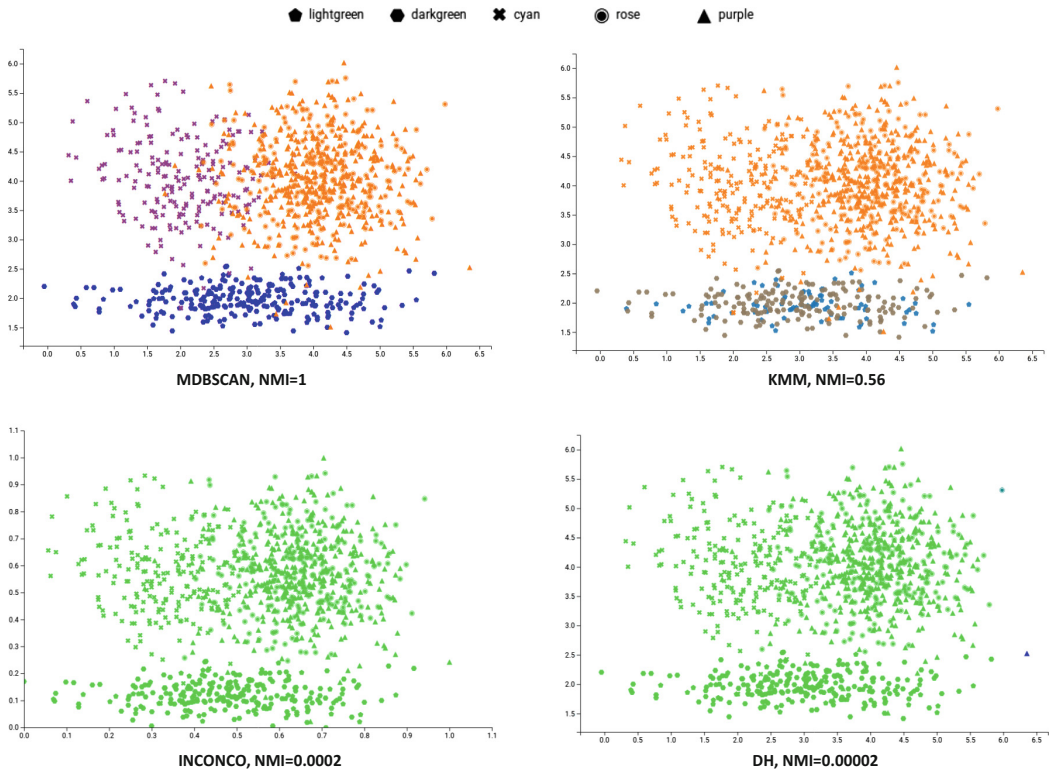


Fig. 3. Clustering results. MDBSCAN, INCONCO, KMM as well as DH.

– **Effectiveness:** In this experiment we consider the same dataset as the running example illustrated in Fig. 1a. Also NMI is applied to assess the effectiveness of a clustering algorithm. As mentioned before a dataset with 3 clusters

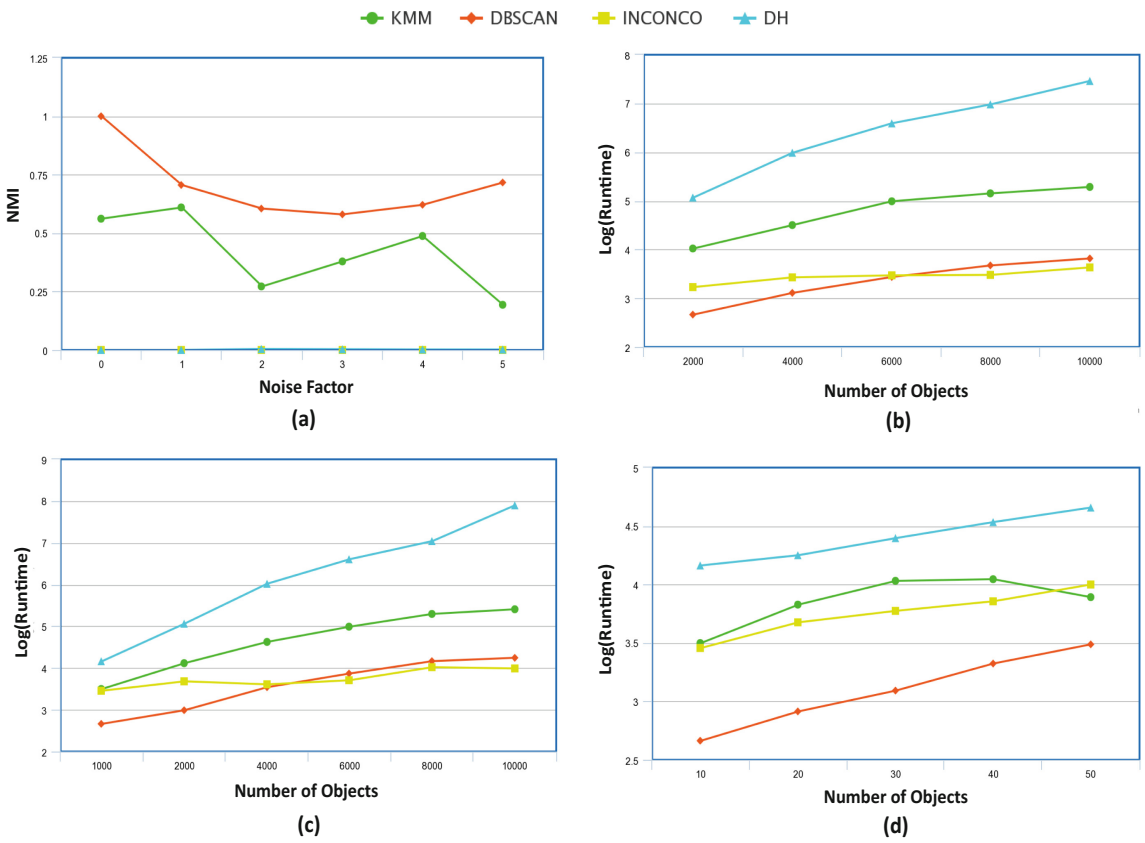


Fig. 4. Comparison in terms of noise-robustness and runtime complexity. (a) Noise-robustness experiment. (b) Runtime experiment by increasing the number of objects while the dimension is 3. (c) Runtime experiment for 10 attributes. (d) Runtime experiment by increasing the dimensionality. (Color figure online)

which consists of 1000 objects with 2 numerical attributes showing the position of any object and a categorical attribute denoting the colors. (Different clusters are illustrated with different shapes in Fig. 1a).
Figure 4 clearly demonstrates that MDBSCAN perfectly outperforms other algorithms in terms of NMI while NMI = 1 shows the best clustering result one could achieve. We illustrate different clusters with different colors to make the comparison more clear. Colors are shown with various shapes. MDBSCAN has been able to assign all Purple and Rose objects to a cluster although some of Purples are positioned in another clusters. (See triangle points in the blue cluster.)

– **Noise-robustness:** To address noise-robustness we introduce a noise factor for both types of attributes. In this experiment we generated a relatively same synthetic dataset as what was generated for the previous experiment. Considering the related distance hierarchy we introduce another category Brown as noise objects distributed in all clusters. We start from 5%. $|C_i|$ noise objects inside any cluster then keep increasing this factor from 1 to 5 times. Moreover for numerical attributes we increase the variance of each numerical attribute by the same factor ranging from 1 to 5 in order to cover

all disturbing aspects. Increasing the variance tends to more mixed clusters. Figure 4a shows the results in terms of NMI running all competitors. As it is clear MDBSCAN outperforms other algorithms regarding any factor of noise while it is non-parametric and finds the exact number of clusters during any experiment.

– **scalability:** We utilize two approaches to address the scalability of our algorithm in comparison to other algorithms. By first approach we increase the number of objects ranging from 2000 to 10000 while the number of attributes is fixed. We considered two different cases in this approach: 1 - the number of attributes is 3 (the data set with almost the same structure as the running example), 2 - the number of attributes is set to 10 (d numerical dimension and 5 categorical). Figure 4a, b indicates that MDBSCAN in both cases is faster than KMM and DH. But in comparison to INCONCO it is faster in the beginning but after almost 5000 objects they have relatively the same run time. However according to the next experiment one could come to the conclusion that MDBSCAN is faster. The other approach is dealing with dimensionality i.e. while the number of objects is fixed the dimensionality is increasing iteratively ranging from 10 to 50. Figure 4d illustrates clearly what we have claimed for MDBSCAN.

6.2 Real Data Experiments

Finally, we evaluate our proposed algorithm MDBSCAN in comparison to other algorithms on real world datasets. As real world problems we used *Teaching Assistant Evaluation* and *Contraceptive Method Choice* from UCI repository [5] and *Airport* dataset from the public project Open Flights [14].

– **Teaching Assistant Evaluation:** The data is provided by [5] and consists of evaluations of teaching performance over three regular semesters and two summer semesters *teaching assistant* (TA) assignments. There are 3 roughly equal-sized categories (low, medium and high) illustrating the scores. The data has 151 objects each of which concerns teaching performance in terms of 4 categorical attributes (Whether the TA is a native English speaker or not, Course instructor, Course, Summer or regular semester) and one numerical (Class size) attribute. In this experiment we consider a flat hierarchy since there is no meaningful hierarchy among categorical values.
  Based on the experimental results MDBSCAN (0.25) outperforms significantly the other algorithms in terms of NMI: INCONCO (0.006), KMM (0.02) and DH (0.02). As mentioned the ground truth is the scores divided to 3 clusters however MDBSCAN found 5 clusters. Although the number of clusters found by MDBSCAN differs from the released labels, it captured characteristics of the dataset better than other algorithms comparing in terms of their mutual information (NMI).
– **Contraceptive Method Choice:** This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey [5]. Samples are married women who were either not pregnant or do not know if they were at

the time of interview. The data consists of 7 categorical and 2 numerical attributes. In this experiment we consider a natural hierarchy for categorical attributes when it makes sense. For instance categorical values corresponding to "Standard-of-living index" consist of 1, 2, 3 and 4. We consider "1" as the lowest standard, "2" and "3" as medium and finally "4" as the highest standard of living. Analogously one could consider the same natural hierarchy for Wife's and Husband's education. For the rest we assume a flat hierarchy.
The target attribute which is used as a ground truth during our experiments is the contraceptive method used by women. This attribute is supposed to group women based on their demographic and socio-economic characteristics. Applying all competitors MDBSCAN (0.35) outperforms other algorithms KMM (0.03), INCONCO (0.04) and DH (0.0001) in terms of NMI.

– **Open Flights Dataset:** The public project Open Flights provides information about airports distributed worldwide. The data consists of 8107 instances each of which has numeric attributes showing the longitude and latitude, the sea height in meters and the time zone. Moreover each object consists of categorical attributes denoting the country, where the airport is located, and the day light saving time. We constructed the concept hierarchy of the country attribute so that each country belongs to a continent. Again three other comparison algorithms (KMM, INCONCO and DH) were applied to this dataset. INCONCO and DH could not find hidden clusters and both of them found only one cluster for this dataset which is not meaningful.
Since there is no ground truth regarding the *Airport* dataset, we first run MDBSCAN and then set the number of clusters required by KMM as the number of clusters found by MDBSCAN in the sense that we could compare them. Figure 5 depicts the discovered clusters after applying MDBSCAN and KMM. MDBSCAN reasonably finds 6 main clusters corresponding to 6 main continents (we consider Russia as European country in distance hierarchy) and two smaller clusters illustrated as roughly noise clusters. However the result of KMM algorithm seems random finding 6 clusters in Asia. For another

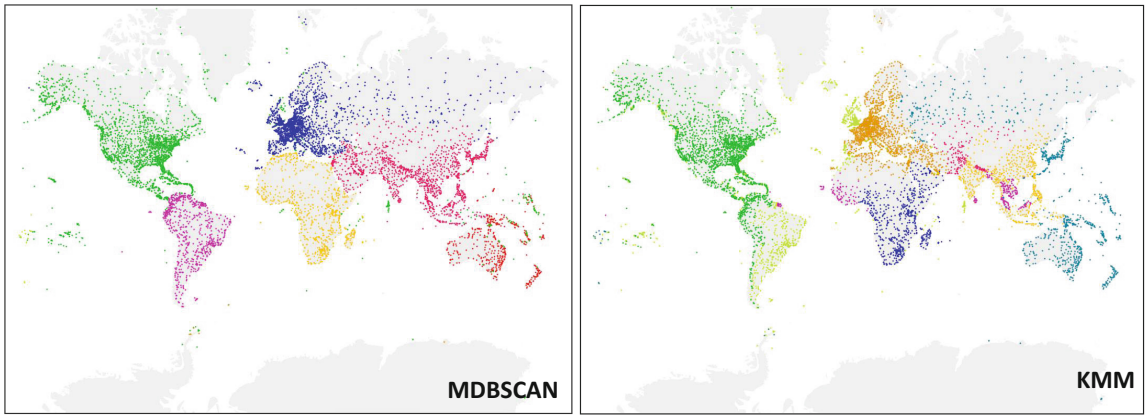


Fig. 5. Clustering results on airport dataset. Comparing MDBSCAN and KMM on Airport dataset.

example KMM groups all the airports located in Australian continent, parts of Eastern Asia and Russia as one cluster while it is hard to interpret this result.

7 Conclusion

To conclude, we introduced an integrative framework to cluster mixed-type datasets consisting of categorical and numerical attributes. In this framework we defined a distance measure, applicable for both types, by means of distance hierarchy. Utilizing this distance measure we avoid converting a data type to another one which tends to information loss and artificially introduced certain orders. Moreover we modified DBSCAN, one of the most efficient and effective density-based clustering algorithm, so that it is able to deal with mixed-type data. Employing MDL principles, we introduced a compression-based approach to score various models and to make MDBSCAN parameter-free. Finally the experiments on synthetic and real datasets indicate the advantages of MDBSCAN in comparison to other state-of-the-art clustering algorithms. However due to Gaussian assumptions considered during the parameter-free procedure may lead to an inaccurate model when the original dataset is non-Gaussian.

References

1. Ahmad, A., Dey, L.: A k-mean clustering algorithm for mixed numeric and categorical data. Data Knowl. Eng. **63**, 503–527 (2007)
2. Böhm, C., Faloutsos, C., Pan, J., Plant, C.: Robust information-theoretic clustering. In: KDD (2006)
3. Böhm, C., Goebl, S., Oswald, A., Plant, C., Plavinski, M., Wackersreuther, B.: Integrative parameter-free clustering of data with mixed type attributes. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010. LNCS (LNAI), vol. 6118, pp. 38–47. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13657-3_7
4. Hsu, C.C., Chen, C.L., Su, Y.W.: Hierarchical clustering of mixed data based ondistance hierarchy. Inf. Sci. **177**(20), 4474–4492 (2007)
5. Frank, A., Asuncion, A.: UCI machine learning repository (2010). http://archive.ics.uci.edu/ml
6. Das, G., Mannila, H., Ronkainen, P.: Similarity of attributes by external probes. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, pp. 23–29 (1998)
7. Hsu, C.C., Chen, Y.C.: Mining of mixed data with application to catalog marketing. Expert Syst. Appl. **32**(1), 12–23 (2007)
8. Hsu, C.C., Chen, C.L., Su, Y.W.: Hierarchical clustering of mixed data based on distance hierarchy. Inf. Sci. **177**(20), 4474–4492 (2007)
9. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min. Knowl. Discov. **2**, 283–304 (1998)
10. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1: Statistics, pp. 281–297. University of California Press, Berkeley (1967). https://projecteuclid.org/euclid.bsmsp/1200512992

11. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of KDD 1996. AAAI Press (1996)
12. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: is a correction for chance necessary? In: ICML (2005)
13. Palmer, C.R., Faloutsos, C.: Electricity based external similarity of categorical attributes. In: Whang, K.-Y., Jeon, J., Shim, K., Srivastava, J. (eds.) PAKDD 2003. LNCS (LNAI), vol. 2637, pp. 486–500. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36175-8_49
14. Patokallio, J.: Open Flights (2016). http://openflights.org/data.html
15. Plant, C., Böhm, C.: Inconco: interpretable clustering of numerical and categorical objects. In: KDD, pp. 1127–1135 (2011)
16. Rissanen, J.: A universal prior for integers and estimation by minimum description length. Ann. Stat. **11**(2), 416–31 (1983)
17. Rissanen, J.: An introduction to the MDL principle. Technical report, Helsinkin Institute for Information Technology (2005)
18. Yin, J., Tan, Z.: Clustering mixed type attributes in large dataset. In: Pan, Y., Chen, D., Guo, M., Cao, J., Dongarra, J. (eds.) ISPA 2005. LNCS, vol. 3758, pp. 655–661. Springer, Heidelberg (2005). https://doi.org/10.1007/11576235_66
19. He, Z., Xu, X., Deng, S.: Clustering mixed numeric and categorical data: a cluster ensemble approach. CoRR, abs/cs/0509011 (2005)

CHAPTER 8

Paper D: ITGC: Information-theoretic Grid-based Clustering

Authors Contributions:

- **Sahar Behzadi.** Proposing the main idea, developing the algorithm; Cooperation on the implementation and conducting experiments; Writing the paper.
- **Hermann Hinterhauser.** Cooperation on developing the algorithm; Implementation; Conducting experiments.
- **Claudia Plant.** Supervision during development and evaluation of the algorithm as well as writing the paper.

ITGC: Information-theoretic Grid-based Clustering

Sahar Behzadi
Faculty of Computer Science
University of Vienna
Vienna, Austria
sahar.behzadi@univie.ac.at

Hermann Hinterhauser
Faculty of Computer Science
University of Vienna
Vienna, Austria
a00946522@unet.univie.ac.at

Claudia Plant
Faculty of Computer Science and
ds:Univie
University of Vienna
Vienna, Austria
claudia.plant@univie.ac.at

ABSTRACT

Grid-based clustering algorithms are well-known due to their efficiency in terms of the fast processing time. On the other hand, when dealing with arbitrary shaped data sets, density-based methods are most of the time the best options. Accordingly, a combination of grid and density-based methods, where the advantages of both approaches are achievable, sounds interesting. However, most of the algorithms in these categories require a set of parameters to be specified while usually it is not trivial to appropriately set them. Thus, we propose an **I**nformation-**T**heoretic **G**rid-based **C**lustering (ITGC) algorithm by regarding the clustering as a data compression problem. That is, we merge the neighbour grid cells (clusters) when it pays off in terms of the compression cost. Our extensive synthetic and real-world experiments show the advantages of ITGC compared to the well-known clustering algorithms.

1 INTRODUCTION

Among various clustering approaches some of them attract more attentions because of their advantages. Partition-based clustering algorithms are popular due to their simplicity and the relative efficiency [7], [2]. K-means [7] is a well-know and well-studied representative for this approach where initially the data is partitioned into k non-empty sets and iteratively the data points are assigned to their nearest cluster. Despite the mentioned advantages, the clustering algorithms in this group suffer from some drawbacks. For instance, the number of clusters k should be specified in the beginning and the results are not deterministic because of their sensitivity to the initialization. Moreover, they are not suitable to discover clusters with non-convex shapes. As a subset of this group, model-based clustering algorithms consider a specific distribution model to represent the data sets. Among them, Expectation-Maximization (EM) algorithm interpret the data as a mixture of Gaussian distributions [5]. On the other hand, density-based clustering algorithms [6], [3] are appropriately designed to deal with clusters having an arbitrary shape. Unlike the partition-based algorithms, the algorithms in this approach are able to deal with noisy data sets. However, in order to find dense regions we need to specify two parameters representing the radius and the density of a neighborhood. Additionally, density-based algorithms are not designed to efficiently deal with clusters with various densities. Spectral clustering [9] is another approach which has become popular due to its simple implementation and its performance in many graph-based clustering. It can be solved efficiently by any standard linear algebra software. However, this approach is expensive for the large data sets since the Computing eigenvectors is the bottleneck.

Another well-known approach is grid-based clustering where any data set is partitioned using a set of grid-cells and data points are assigned to the appropriate grid cell. Grid-based methods [1], [14], [11] quantize the object space into a finite number of cells (hyper-rectangles) and then perform the required operations on the quantized space. The main advantage of grid-based methods is their fast processing time which depends on the number of cells in the grid. In the other word, no distance computation is required and the clustering is performed on summaries and not on the individual objects. Thus, the complexity of grid-based algorithms is usually *O(number of populated grid cells)* and not *O(number of objects)*. Beyond their ability to deal with noisy data sets, grid-based clustering algorithms are able to identity clusters irrespective of their shapes. Unlike most of the clustering algorithm which require an initialization phase, the algorithms in this category are insensitive to the order of input records and therefore are deterministic.

Despite the valuable advantages of grid-based clustering algorithms, to the best of our knowledge, all of them are parametric algorithms where a user is required to specify the parameters. However, most of the time it is not trivial to appropriately set them. Thus, utilizing the principle of Minimum Description Length (MDL) we propose a non-parametric **I**nformation-**T**heoretic **G**rid-based **C**lustering algorithm where we regard the clustering task as a data compression problem so that the best clustering is linked to the strongest data compression. First, an adaptive grid is constructed corresponding to the statistical characteristics of any data set and non-empty cells are considered as single clusters. Then, we combine the concept of density and grid-based methods and employing our compression-based objective function we start merging clusters with their neighbour grid cells only if it pays off in terms of the compression cost.

In this paper we propose an information-theoretic clustering algorithm offering the following contributions:

- **Adaptive partitioning:** We utilize the statistical characteristics of any data set, e.g. local and global dispersion, in order to introduce an adaptive partitioning of the data.
- **Non-parametric clustering:** Employing the MDL-based objective function, we iteratively merge clusters when it pays off in terms of the compression cost automatically. Thus, no parameter needs to be specified.
- **Insensitivity to the shape of clusters:** ITGC employs the concept of density-based methods in order to select the next merging candidate. Thus, it is insensitive to the shape of clusters whether they are Gaussian, arbitrary or even having various density regions.
- **Scalability:** Analogous to other grid-based clustering algorithm, the complexity of ITGC depends on the number of cells not on the number of objects which leads to a scalable algorithm in terms of the number of objects.

2 INFORMATION-THEORETIC GRID-BASED CLUSTERING

In order to introduce a grid-based clustering algorithm we need to address two fundamental questions: (1) how to find a specific appropriate partitioning (grid) corresponding to any data set; (2) how to efficiently merge the cells to discover the hidden clusters. Thus, our proposed algorithm ITGC consists of two main building blocks: (1) finding a suitable grid corresponding to specific characteristics of any data set and (2) employing MDL principle to effectively and efficiently merge the cells without any parameter to be specified.

2.1 Partitioning the Data

Finding a suitable partitioning with respect to the data is a crucial task in a grid-based clustering algorithm. Inspired by [8], we utilize the characteristics of any data set to introduce the best fitting partition. That is, we are looking for a partition which leads to high internal homogeneity in the cells and high external heterogeneity of each cell with respect to its neighbors for every single cell. Thus, for any cell C_j consisting of n_j data points the statistical indicators are defined as:

$$\bar{X}_j = \frac{\sum_{i=1}^{n_j} X_{ij}}{n_j} \quad and \quad S_j = \sqrt{\frac{\sum_{i=1}^{n_j}(X_{ij} - \bar{X}_j)^2}{n_j - 1}} \tag{1}$$

Where X_{ij} is the distance of the $i-th$ data point in C_j to the center of this cell. Thus, $\bar{X}_j$ is the average distance of data points to the center of C_j and S_j is the standard deviation of the cell. These are statistical indicators on the local level (each individual cell), similar indicators are calculated on the global level (the entire grid) as:

$$\mu = \frac{\sum_j \bar{X}_j}{N} \quad and \quad \sigma = \sqrt{\frac{\sum_j(\bar{X}_j - \mu)^2}{N - 1}} \tag{2}$$

Where μ is the average center - distance of all cells, N is the total number of cells and σ is the standard deviation of all cells. Based on these indicators, we define $CV_{Local}(j)$ and CV_{Global} as the coefficient of variation (CV) corresponding to any cell C_j and the global variation, respectively. That is,

$$CV_{Local(j)} = \frac{S_j}{\bar{X}_j} \quad and \quad CV_{Global} = \frac{\sigma}{\mu} \tag{3}$$

In another point of view, the above scores show how widespread the data points are indicating the relative dispersion at the local (cell) and global (grid) levels. Finally the partitioning cost is defined as:

$$gridCost = \frac{CV_{Global}}{avg \quad CV_{Local}(j)} \tag{4}$$

Considering the cost corresponding to any grid size $k \times k$, we iteratively increase k starting from 1 until it pays off. However, it is not trivial to justify a terminal for this process without observing its trend. Initially, the grid cost increases sharply by increasing k then it slows down quickly and continues linearly. Observing this common trend, we conduct a simple linear regression on various costs with respect to various k values. The regression line is expected to fit more through the higher ks where the costs have lower deviations. Thus, the optimal partitioning can be set to the first k where the grid cost deviated from the fitted line lower than the average.

On the other side, by increasing the size of grid the area of non-empty cells decreases. Thus, it is reasonable to assume this trend to continue with even smaller cells, but the descending trend slows down while decreasing cell sizes. Visualizations of the collected area reveals a common trend which is reverse to the previous one. The area starts at a maximum value, decreases very sharply at lower k and keeps decreasing at a lower gradient. In order to find the optimum value for k, we analogously fit a linear regression through the data set rejecting the low k values which deviate larger than average from the fitted line. The following steps summarize this procedure.

- Step 1: The grid is divided into $k \times k$ cells where the initial size for k is 1.
- Step 2: The grid cost as well as the area of non-empty cells are determined and the values are stored.
- Step 3: We iteratively increase k ranging from 1 to a max_k and repeat the previous steps
- Step 4: Now the optimum partitioning is determined employing two different criteria.

2.2 MDL-based Objective Function

Utilizing the Minimum Description Length (MDL) principle [10] we regard the clustering task as a data compression problem so that the best clustering is linked to the strongest data compression. Given the appropriate model corresponding to any attribute, MDL leads to an intuitive clustering result employing the compression cost as a clustering criterion. The better the model matches major characteristics of the data, the better the result is. Following the MDL principle, we encode not only the data but also the model itself and minimize the overall description length. Simultaneously, we avoid over-fitting since the MDL tends to a natural trade-off between model complexity and the goodness-of-fit. That is, for a given cluster C_i the corresponding compression cost is defined as:

$$MDL(C_i) = CodingCost(C_i) + ParamCost(C_i) + IDCost(C_i) \tag{5}$$

where *CodingCost* shows the cost of coding the data points in cluster C_i by means of a coding scheme. The next two terms illustrate the model complexity where the model itself needs to be encoded. In this paper we employ the Huffman coding scheme to encode the data considering an appropriate model. That is, given the corresponding *Probability Distribution Function* (PDF) to any attribute, the coding cost of any object x is determined by $-log_2 PDF(x)$. Any PDF would be applicable and using a specific model is not a restriction [4] for our algorithm. In this paper, we consider Gaussian PDF for simplicity. In the following we elaborate our objective function more concretely.

- **Objective Function:** The overall MDL-based objective function is the summation of the all compression costs with respect to various clusters. That is,

$$MDL(\mathcal{D}) = \sum_{C_i \in C} MDL(C_i) \tag{6}$$

  where $\mathcal{D}$ is the entire data set and $C = \{C_1, ..., C_k\}$ is the set of all clusters.

- **Data Coding Cost:** Let $\mathcal{X} = \{X_1, ..., X_d\}$ denote the set of all attributes. For any object $x = (x_1, ..., x_d)$ the corresponding coding cost is the sum of encoding any attribute value x_i. Putting all together, the coding cost corresponding to cluster C_i is given by:

$$CodingCost(C_i) = -\sum_{X_j \in \mathcal{X}} \sum_{x \in C_i} \log_2 PDF_j(x) \quad (7)$$

where $PDF_j(.)$ is the Gaussian model with respect to $j-th$ attribute X_j.

- **Model Complexity:** Without taking the model complexity into account, the best result will be a clustering consisting of singleton clusters. This result is completely useless in terms of the interpretation. In order to specify the associated cluster with any data object, we need to encode the cluster IDs. Thus, the IDCosts are required to balance the size of clusters and defined as:

$$IDCost(C_i) = |C_i|.log_2 \frac{|C_i|}{|\mathcal{D}|} \quad (8)$$

Following the fundamental results from the information theory [10], for any attribute X_j the parameters corresponding to model employed to encode the data need to be encoded as well. That is, concerning any Gaussian distribution PDF_j with respect to the attribute X_j, the mean value and the standard deviation need to be encoded, i.e.

$$ParamCost(C_i) = \frac{1}{2}.(2|\mathcal{X}|).log_2|C_i| \quad (9)$$

2.3 Algorithm

As mentioned, ITGC consists of two main building blocks. Algorithm 1 summarizes our grid-based algorithm ITGC. First, an optimal grid is constructed following the steps mentioned in Section 2.1, i.e. the procedure *FindOptimumGrid(.)*. Then, we start merging the cells if it pays off in terms of our objective function (Section 2.2). Initially every cell is considered as a cluster while empty cells are ignored. The cluster with the most number of data points is chosen in the sense that at the end the results are deterministic. We compute the coding cost with respect to the selected cluster MDL_{before} and merge this cluster with one of its neighbors and compute the cost after merging two clusters MDL_{after}. If the cost after merging is smaller than the cost before, we merge two clusters and continue the merging process. Otherwise, the visited cluster is marked. Finally the algorithm terminates if no unmarked non-empty cell exists.

3 EXPERIMENTS

In this section we assess the performance of ITGC comparing to other clustering algorithms in terms of *Normalized Mutual Information* (NMI) which is a common evaluation measure for clustering results [13]. NMI numerically evaluates pairwise mutual information between ground truth and resulted clusters scaling between zero and one.

We conducted several experiments evaluating our algorithm on synthetic and real-world data sets. In order to investigate the effectiveness of ITGC we generated various data sets and compared to the base-line clustering algorithms, i.e. k-means [7] and DBSCAN [6]. While the insensitivity of ITGC to the shape of clusters as well as its effectiveness is illustrated by synthetic experiments, we extended the comparison to the wider range of well-known clustering algorithms. Our algorithm is implemented in Java and the source code as well as the data sets are publicly available [1].

[1] https://tinyurl.com/y85gglpx

Algorithm 1: Information-theoretic grid-based clustering

```
ITGC (D)
G = FindOptimumGrid(D);
C = {C_1, ..., C_k}        // Non-empty cells in G
seeds := non-visited clusters;
while (seeds != empty) do
    C_i := the cluster with the most data points in C
    C_i is visited
    while (MDL_before > MDL_after) do
        MDL_before = MDL(C_i)
        C_j := a random non-visited neighbor cell w.r.t C_i
        C_m := the cluster after merging C_i and C_j
        MDL_after = MDL(C_m)
        if MDL_before > MDL_after then
            remove C_j and C_i from C .
            add C_m to C
        end if
    end while
    seeds := non-visited clusters;
end while
return (C)
```

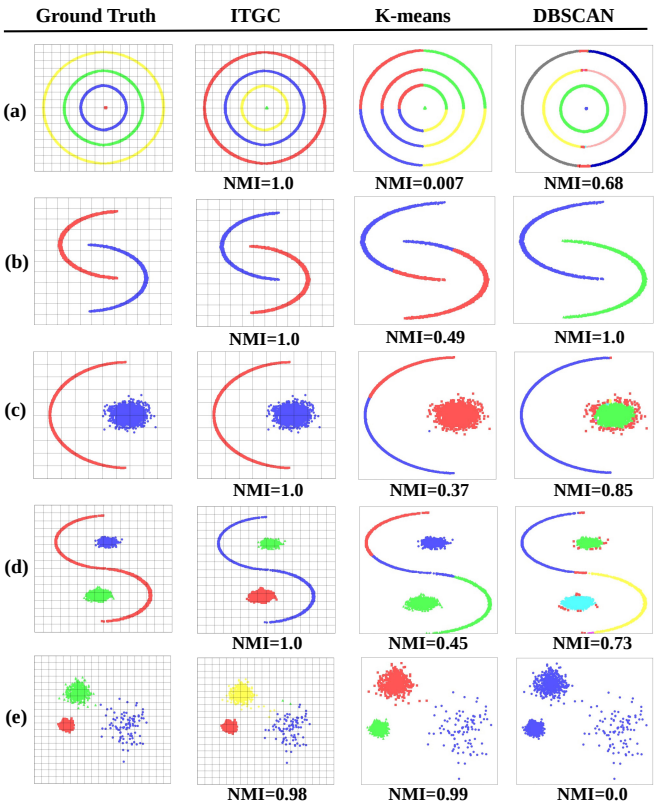


Figure 1: Comparison on various synthetic data sets.

3.1 Synthetic Experiments

In order to cover all aspects of ITGC, we investigate the performance of the algorithms considering various synthetic data sets including arbitrary shaped data sets as well as clusters with different densities. Then, we continue experiments by comparing all algorithms in terms of the scalability.

Performance: Most of the time any clustering algorithm is designed for a specific kind of data sets. For instance, k-means

Dataset	*Attr./Objects*	*ITGC*	*k-means*	*DBSCAN*	*EM*	*Spectral C.*	*CLIQUE*	*Single L.*
Iris	4/150	**0.66**	0.53	0.59	0.60	0.6	0.00	0.59
Occupancy Detection	7/20560	**0.61**	0.56	0.00	0.31	0.00	**0.61**	-
Breast Cancer	9/286	**0.47**	0.32	0.41	0.45	0.45	0.39	0.27
User Knowledge	5/403	0.24	**0.27**	0.01	**0.27**	**0.27**	0.00	0.01

Table 1: Comparison on real data sets.

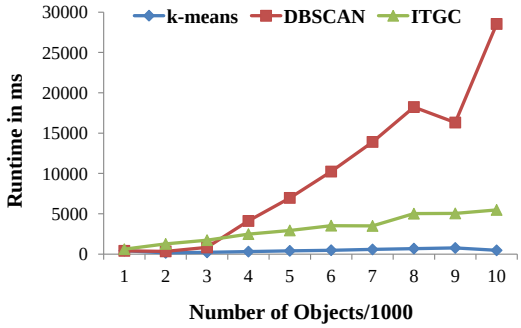


Figure 2: Scalability of the algorithms by increasing the number of objects.

appropriately deals with Gaussian shaped data sets and its performance dramatically decreases when the clusters have no specific shape. On the other hand density-based clustering algorithms are sensitive to different densities w.r.t. various clusters. In order to evaluate the performance of ITGC concerning various shapes of clusters, we synthetically generated arbitrary shaped clusters in a combination with some Gaussian clusters. Figure 1 shows the effectiveness and the insensitivity of ITGC considering various cases. As expected, k-means fails when the clusters are not Gaussian (Figure 1a,b,c,d). On the other hand, DBSCAN is not able to discover the clusters with various densities (Figure 1d,e)

Scalability: To evaluate the efficiency in terms of the runtime complexity we generated 5 dimensional synthetic data sets where we iteratively increased the number of data objects ranging from 1,000 to 10,000. Figure 2 shows the result of this experiment. As expected, k-means is the fastest algorithm while DBSCAN is the worse since its complexity highly depends on the number of objects. Although ITGC is not able to outperform k-means, its corresponding execution time is still reasonable and more efficient than DBSCAN.

3.2 Real Experiments

In this section we extend our experiments to the wider range of clustering algorithms including EM [5], Single link [12], spectral clustering [9] and CLIQUE [1] as the well-known representatives for any clustering approach. We evaluate clustering quality of ITGC on real-world data sets. We used *Iris*, *Occupancy Detection*, *User Knowledge* and *Breast Cancer* data sets from the UCI Repository [2]. Table 1 shows the characteristics of any data set and the results of applying various algorithms in terms of NMI. Concerning any data set the best NMI is high lighted and when getting "Out Of Memory" error we inserted "-" in the table. As Table 1 illustrates ITGC outperforms other algorithms considering the first 3 real-world data sets. Interestingly, in this experiment we outperform CLIQUE which is a well-known grid and density-based clustering algorithm (the results are similar on the *Occupancy* data set). Although some of the comparison methods perform slightly better than ITGC on *User Knowledge* data set, our result is still comparable and we outperform DBSCAN, CLIQUE and Single link.

[2]http://archive.ics.uci.edu/ml/index.php

4 CONCLUSION AND FUTURE WORKS

In this paper we propose an information-theoretic clustering algorithm, ITGC, utilizing the MDL-principle. Firstly, We employ the statistical characteristics of any data set to appropriately partition the data without any presumptions. Then, an MDL-based objective function is proposed to iteratively merge the neighbour clusters when it pays of in terms of the compression cost of the clusters. Our experiments on synthetic and real-world data sets show the advantages of our proposed algorithm compared to other well-known clustering algorithms. Similar to other grid-based clustering algorithms, our algorithm may lead to inefficiency when dealing with huge data sets in terms of the dimensionality. Thus, a possible future work would be to investigate the parallelization approaches in the sense that the required memory to store the grid information could be distributed. As another option for the further investigation could be to enhance the partitioning procedure in the sense that it results a sparse grid which is cheaper in terms of the memory.

REFERENCES

[1] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. 1998. Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD Conference*. 94–105.
[2] Paul E. Green Anil Chaturvedi and J. Douglas Caroll. 2001. K-modes Clustering. *Journal of Classification* 18, 1 (2001).
[3] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering Points to Identify the Clustering Structure *(SIGMOD '99)*. New York, NY, USA.
[4] Christian Böhm, Christos Faloutsos, Jia Pan, and Claudia Plant. 2006. Robust information-theoretic clustering. In *KDD*.
[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* 39, 1 (1977), 1–38.
[6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.. In *KDD Conference*.
[7] J. Macqueen. 1967. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*. 281–297.
[8] Joel D. Melo, Edgar M. Carreno, Aida Clavino, and Antonio Padilha-Feltrin. 2014. Determining spatial resolution in spatial load forecasting using a grid-based model. *Electric Power Systems Research* 111 (2014), 177–184.
[9] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2001. On Spectral Clustering: Analysis and an Algorithm *(NIPS'01)*.
[10] Jorma Rissanen. 2005. *An Introduction to the MDL Principle*. Technical Report. Helsinkin Institute for Information Technology.
[11] Gholamhosein Sheikholeslami, Surojit Chatterjee, and Aidong Zhang. 1998. WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases *(VLDB '98)*. San Francisco, CA, USA.
[12] R. Sibson. 1973. SLINK: An optimally efficient algorithm for the single-link cluster method. *Comput. J.* 16, 1 (1973), 30–34.
[13] Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2009. Information theoretic measures for clusterings comparison: is a correction for chance necessary?. In *ICML Conference'09*.
[14] Wei Wang, Jiong Yang, and Richard R. Muntz. 1997. STING: A Statistical Information Grid Approach to Spatial Data Mining *(VLDB '97)*. San Francisco, CA, USA, 186–195.

CHAPTER 9

Paper E: Dependency Anomaly Detection for Heterogeneous Time Series : A Granger-Lasso Approach

Authors Contributions:

- **Sahar Behzadi.** Cooperation on the main idea and developing the algorithm as well as writing the paper; Implementation; Conducting experiments.
- **Kateřina Hlaváčková-Schindler.** Cooperation on the main idea, developing the algorithm, experiments as well as writing the paper.
- **Claudia Plant.** Supervision during development and evaluation of the algorithm as well as writing the paper.

2017 IEEE International Conference on Data Mining Workshops

Dependency anomaly detection for heterogeneous time series: A Granger-Lasso approach

Sahar Behzadi
University Of Vienna
Vienna, Austria
Email: sahar.behzadi@univie.ac.at

Katerina Hlaváčková-Schindler
University Of Vienna
Vienna, Austria
Email: katerina.schindlerova@univie.ac.at

Claudia Plant
University Of Vienna
Vienna, Austria
Email: claudia.plant@univie.ac.at

***Abstract*—The special characteristics of time series data, such as their high dimensionality and complex dependencies between variables make the problem of detecting anomalies in time series very challenging. Anomalies and more precisely dependency anomalies ensue from the temporal causal dependencies. Furthermore the graphical Granger causal models provide an appropriate environment to capture all the temporal dependencies in Gaussian time series. However many production systems are characterized by a high degree of complex stochastic processes consisting of heterogeneous time series. Considering this situation discovery of dependency anomalies would be more challenging since almost all the current algorithms are dealing with the homogeneous cases. Granger-Lasso algorithm is a well-known $L1$ penalization algorithm which copes with the temporal causality detection only for Gaussian time series. Inspired by this algorithm and considering the incremental heterogeneous time series generated in many different industries, we propose a modification for Granger-Lasso algorithm in the sense that it would be applicable for a larger class of heterogeneous time series. To introduce this algorithm we are motivated by generalized linear models. Moreover based on the proposed algorithm for discovery temporal dependencies we introduce its application in anomaly detection considering time series followed by distributions from exponential family, e.g. Poisson, binomial or multinomial distribution. The Granger-Lasso procedure is solved by using least square cost function with Lasso penalty for appropriately transformed input time series. The experimental results illustrate the performance and efficiency of the proposed algorithm on the synthetic and other datasets. We evaluated the proposed method on causality testing on different examples.**

I. Introduction

Time series play an incremental role in production processes in scientific measures e.g. medical or climatological measurements. Due to their temporal nature, these data can provide an insight into the complex systems which they measure in the sense that one can detect production anomalies at the early stages. Detecting anomalies is a crucial problem in many fields providing us with a large amount of data from sensors, logs and so on. There are two types of anomalies in multivariate time-series data:
1. The anomaly occurring only within individual variables (univariate anomaly),
2. The anomaly occurring due to changes of temporal dependencies (dependency anomaly).

The dependency anomaly is much more challenging to investigate. However it is more common in the reality. Profiting the temporal and autocorrelation property of time series one can detect the temporal dependencies and consequently dependency anomalies. Different methods have been developed to infer temporal causal relationships from time series data, including dynamic Bayesian Networks [1] and Granger causality [2].

Variations of Granger graphical models were proposed to detect temporal Granger dependencies between variables in multivariate time series data with Gaussian distribution [3], [4], [5], [6]. Among them penalized methods often provide better prediction accuracy simultaneously with providing sparse models. Particularly when one deals with a numerous number of time series, sparse models perform more efficient. They often provide better prediction accuracy since the performance of the Granger causality methods depend on the length of the time series as well as the sample size. One of the well-known algorithms to discover the structure of graphical models based on the concept of Granger causality is the Granger-Lasso (or $L1$ penalty) introduced by Arnold *et al.* [3].

Granger-Lasso is an efficient and effective algorithm dealing with a large number of time series. Qiu et al. used this algorithm to the dependency anomaly detection in time series [7]. One of the major assumptions by Granger-Lasso as well as by anomaly detection algorithm using this approach is that all the time series are Gaussian. They assume that the density model in all of the experiments are linear Gaussian models. However in principle it could be an arbitrary statistical model since it is more likely in the realistic cases to have various time series from different distributions.

In this paper, we deal with this question: How to efficiently detect temporal causalities and consequently anomalies for heterogeneous multivariate time series. More precisely a heterogeneous time series consists of various time series from different distributions. To make it more clear, let's imagine an stochastic process consisting of three time series. Figure 1 illustrates such a situation so that we are given a system of

2375-9259/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDMW.2017.155

mixed time series. In this example x_1 and x_3 are Gaussian and x_2 is a Gamma distributed time series. Edges in this figure show the intended causal relations among x_1, x_2 and x_3. Now the issue is how to modify the Granger-Lasso which is appropriate for numerous Gaussian time series in the sense that it is applicable for discovery temporal dependencies in a mixed or heterogeneous cases. Finding such a modification provides us an effective tool to detect dependency anomalies as well.

In this paper we mainly address two mentioned issues: we first introduce an algorithm by means of Granger-Lasso to discover all the Granger causal relations among heterogeneous time seies and then applying this algorithm we introduce another algorithm to detect the dependenecy anomalies in that case. Inspired by the idea of generalized linear models (GLM) and also in order to extend the application of the Granger causality concept to non-Gaussian time series, we propose a model for finding the causal interactions. And then we extend the idea of [7] for anomaly detection in multivariate time series with distributions from exponential family.

The major contributions of our approach are as follows:

- **Heterogeneous graphical Granger models:** As a modelling method we utilize graphical Granger models in the GLM framework so that the model will be applicable to other than Gaussian time series. By applying appropriate transformations of the input time series we introduce a heterogeneous graphical Granger model which allows additive interactions among time series with distributions from the exponential family.
- **Modified Granger-Lasso:** We modify Granger-Lasso algorithm and more precisely its objective function so that it could be applicable for not only estimation of Granger causalities in linear Gaussian models but also non-linear cases from the exponential family e.g. Poisson.
- **Anomaly detection:** Utilizing the modified Granger-Lasso algorithm for the heterogeneous time series we detect all dependency anomalies as a natural consequence of the Granger causality detection.
- **A symmetric information theoretic score:** In order to efficiently assess the anomalies we introduce an information-theoretic anomaly score which is symmetric. This anomaly score helps us to avoid computationally expensive steps of the anomaly detection algorithm.

The paper is organized in the way that we proceed with explaining two first contribution as a base line for the last two ones. After introducing our model we will describe how to use this useful tool to detect the heterogeneous anomalies. In Section V we report some experimental results on two datasets investigating the performance of causality detection part of the paper. The evaluation of anomaly detection will remain for the future works.

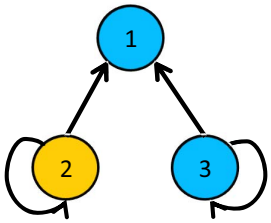


Fig. 1. **Temporal Feature Causal Network** Temporal causal network illustrated for two Gaussian time series (x_1, x_3) and a Gamma distributed one (x_2).

II. Granger-Lasso: A temporal causal detection algorithm

One of the well-known approaches to detect causalities among time series is Granger causality. It is based on the intuition that the cause helps to predict its effects in the future. More precisely a feature x is said to Granger-cause y, if the autoregressive model for y in terms of past values of both x and y is statistically significantly more accurate than that one based on just the past value if y [2]. Graphical Granger models extend the notion of Granger causality among two variables to many variables. There are many methods how to discover the structure of graphical models based on the concept of Granger causality [8], [3]. Arnold *et al.* applied the lasso penalty and proposed a temporal dependency learning algorithm only for Gaussian time series [3]. In this section we recall some preliminaries required to introduce our model.

This section is organized as follows: first of all we review Granger causality and graphical Granger models and then lasso estimation to find causalities. Finally we recall GLM so that we are well-equipped to introduce our model for heterogeneous time series (time series of various distributions).

A. Granger causality and graphical Granger models

First we recall the definition of Granger causality between two univariate time series. Let $x^{1:T} = \{x\}_{t=1}^{T}$ and $y^{1:T} = \{y\}_{t=1}^{T}$ denote time series up to time T. Consider the following two regression models:

$$y^T = Ay^{1:T-1} + Bx^{1:T-1} + \varepsilon^T \quad (1)$$

$$y^T = Ay^{1:T-1} + \varepsilon^T \quad (2)$$

Then x is said to be Granger-causal for y if the model (1) results in statistically significantly better regression model than with (2).

Graphical Granger models extend the notion of Granger causality among two variables to p variables. Let $x_1, \ldots, x_p$ be p time series and X define the rearrangement of p given time series at time t into a vector time series, i.e. $\mathbf{X}^t = (x_1^t, \ldots, x_p^t)'$. Thus, we define the Graphical Granger model:

$$\mathbf{X}^T = \mathbf{A}^1\mathbf{X}^{T-1} + \ldots + \mathbf{A}^{T-1}\mathbf{X}^1 + \varepsilon^T \quad (3)$$

where $\mathbf{A}_1, \ldots, \mathbf{A}_{T-1}$ are p-dimensional vectors and ε^T is the vector of errors. One can imagine this model (3) as a *Directed Acyclic Graph* (DAG) over the time series. Assume $G = (V, E \subset V \times V)$ be a DAG, when V is the set of time series $\mathbf{X^1}, \ldots \mathbf{X^T}$ and E the set of associations among them. In graphical models E corresponds to the $p \times p$ adjacency matrix achieved by $\mathbf{A}_1, \ldots, \mathbf{A}_{T-1}$.

Causal relationships among different time series in the graphical model (3) can be represented by DAGs. x_j^t is said to be Granger-causal for x_i^T if the corresponding coefficient, $A_{j,i}^t$ is statistically significant [5]. This corresponds graphically to the fact that there exists an edge $x_j^t \rightarrow x_i^T$ in the graphical model with $T \times p$ nodes.

Lasso-type estimates of DAGs can therefore be used in the context of graphical Granger models in order to estimate the effects of time series on each other. In the next section we will address this estimation.

B. Granger-Lasso causality detection algorithm

Applying an $F - test$ or any other statistical significance tests is a straight-forward approach for detecting Granger causalities between two time seires. This approach is time consuming for more than two time series and sensitive to the number of observations. Moreover it guarantees only a suboptimal solution. Additionally Zou and Feng investigated Granger causality comapring to the other methods and concluded that the performance of any approach depends on the length of the time series as well as the sample size [9]. Therefore we need sparse models. particularly in case where the sample size is small. Granger-Lasso is a temporal dependency learning algorithm introduced by Arnold *et al.* [3] to achieve the neighborhood sparsity based on the lasso (L_1) penalty.

Consider a graphical model with p time series, observed over T time points, and let d be the lag. Let ${X^{Lagged}}_{T,d}$ be the concatenated vector of all the lagged variables up to time T,i.e. $\{{x_i}^{T-t} | i = 1, ..., p; t = 1, ..., d\}$. The graphical Granger model based on (3) for lag d can be reformulated as p minimization problems:

$$\gamma_i(\eta_i) = \arg \min_{\gamma_i^T \in \mathbb{R}^p} \sum_{T=d+1}^{n} (x_i^T - X_{T,d}^{Lagged} \gamma_i^T)^2 + \eta_i L_i(\gamma_i) \quad (4)$$

for a given number of different samples n and $\eta_i > 0$ the regularization parameter. η_i indicates the strength of shrinkage and variable selection, which, in moderation can improve both prediction and interpretability. In this study we assume the regularization parameters given however there are many heuristics to find an appropriate amount for this parameter. Generally selecting it well is essential to the performance of lasso therefore we post pone investigating the influence of different heuristics to find η_i.

$$L_i(\gamma_i) = \sum_{j=1}^{p} |\gamma_{ij}| \quad (5)$$

is L_1 type penalty function. For each time series x_i, γ_i is a p-dimensional vector called Lasso coefficients which minimizes the average squared errors of regressing for x_i with respect to the Lasso penalty. Thus, x_j Granger causes x_i if and only if at least one of the corresponding coefficients γ_i is non-zero.

It is well-known that the L_1-penalized least square regression, as targeted by the Lasso, is a convex problem, making it possible to attain the global maximum. Profiting of L_1 regularized regressions, we are able to discover sparse Granger causality relations.

Granger-Lasso algorithm works appropriately for Gaussian time series. However this is more realistic to have a complex system with many time series from various distributions. Refer to Figure 1, where three time series are given, one is Gamma distributed and the two other ones Gaussian, is Granger-Lasso still applicable?

In the following we investigate the performance of Granger-Lasso in the heterogeneous case where we are given many time series of various distributions. Without loss of generality we only assume time series from exponential family e.g. Poisson, Gamma and so on. The algorithm proposed in this paper requires the notion of regression for GLMs in the sense that we provide a framework to apply Granger-Lasso on heterogeneous case. In the following sections we recall GLM and finally introduce our model for causality detection on heterogeneous time series.

C. Generalized Linear Model

The idea of generalized linear models (GLM) was introduced by Nelder and Wedderburn[10]. It is a natural extension of linear regression to the cases when the considered regressed time series are not necessarily Gaussian by allowing the linear model to be related to the response variable via a link function. The GLM framework provides a tool for implementing model prediction using standard software.

Mainly GLM consists of three components: the random component which belongs to the exponential family of distributions of with mean value $\mu = E[y|\mathbf{x}]$, a linear predictor $\eta = X\beta$, and the link function g, a monotone twice differentiable function given by a user. The link function associates the linear predictor with the mean value via $g(\mu) = \theta$.

Table I shows common distributions with canonical link functions.

III. Causality Detection for Heterogeneous Time Series: The Model

With respect to previous works (e.g. [3]) this question will arise whether it is possible to discover temporal dependencies between heterogeneous time series or not. Considering heterogeneous time series, in this section we will propose a model in the *Generalized Linear Model* (GLM) framework to detect the temporal causalities. In the following we introduce our model for heterogeneous time series.

	Link Function	Mean Function
Normal	$X\beta = \mu$	$\mu = X\beta$
Exponential/ Gamma	$X\beta = \mu^{-1}$	$\mu = X\beta^{-1}$
Inverse Gaussian	$X\beta = \mu^{-2}$	$\mu = X\beta^{\frac{-1}{2}}$
Poisson/ Countable	$X\beta = ln(\mu)$	$\mu = exp(X\beta)$
Bernoulli/ Categorical/ Multinomial	$X\beta = ln\frac{\mu}{1-\mu}$	$\mu = \frac{exp(X\beta)}{1+exp(X\beta)}$

TABLE I. COMMON LINK FUNCTIONS IN GLM FRAMEWORK.

Let's assume we have found all the temporal dependencies between various time series applying Granger-Lasso algorithm. If we assume x_j Granger causes x_i, then we will find this relationship between mean value of both time series x_j and x_i. In another point of view Granger-causal relationships will be preserved even if we apply Granger-Lasso algorithm on the mean values. Inspired by this fact we introduce a model in GLM framework. We transform each time series at time T to its mean value by means of the link function so that we provide the linear relationships. Consequently a Granger-Lasso algorithm will be applicable since we are dealing with the linear regressions. In the following we first propose a graphical Granger model over the heterogeneous time series and then we explain how to find the Granger causalities by applying Granger-Lasso algorithm:

A. Graphical Granger model for heterogeneous time series

Considering the mentioned graphical Granger model 3 we assume p time series $x_i, i = 1, \ldots, p$. Let time lag d be given. Each time series is generated from a concrete distribution function in the exponential family, for example with Poisson or binomial distribution. Profiting of the GLM properties, we consider the following model in which we suppose that the mean of each time series depends on its own history up to the lag d and the past values of concurrent time series:

$$E[x_i^T] := \mu_i(T|X_{T,d}^{Lagged}) = g_i^{-1}(X_{T,d}^{Lagged}\gamma_i^T) \tag{6}$$

more precisely we can write:

$$E[x_i^T] = g_i^{-1}(\sum_{j=1}^{p}\sum_{t=1}^{d} x_j^{T-t}\gamma_{ij}^t) \tag{7}$$

where $E[x_i^T]$ is the expected value of x_i^T. g_i is an invertible link function corresponding to time series x_i for each $i = 1, \ldots, p$. Basically we can have a multivariate link function i.e. for each i there is a particular form of link function, $g = (g_1, \ldots, g_p)$.

As mentioned we transform all the time series to their mean value in the sense that we provide linearity and apply Granger-Lasso over the constructed graphical model. W_i^T denotes the transformation of x_i^T by means of the corresponding link function:

$$W_i^T := g_i(E[x_i^T]) \tag{8}$$

when $E(.)$ is the arithmetic mean value by time T:

$$E(x_i^T) := \frac{1}{T}\sum_{t=1}^{T} x_i^t \tag{9}$$

Now we are well equipped to construct a graphical Granger model over the transformed time series:

$$G := (W, H) \tag{10}$$

where W is the set of temporal nodes $W^T = (W_1^T, \ldots, W_p^T)$ and H is the set of edges in the graph corresponding to the oriented temporal causal connections among the nodes.

B. Granger-Lasso estimation of the graphical model

Via the link function applied on the mean values, a linear relationship among the time series is established. Now we apply Granger-Lasso estimators to the constructed graphical Granger model (10). To estimate the proposed model, we define p non-overlapping Lasso optimization problems for $i = 1, \ldots, p$:

$$\gamma_i(\eta_i) = \arg\min_{\gamma_i^T \in \mathbb{R}^p} \sum_{T=d+1}^{n} (W_i^T - X_{T,d}^{Lagged}\gamma_i^T)^2 + \eta_i L_i(\gamma_i) \tag{11}$$

For some r, s between $1, \ldots, p$ the process x_r Granger–causes process x_s, denoted by $x_r \to x_s$, if and only if at least one of the corresponding coefficients of γ_{sr} in (11) for any T is non-zero.

In contrast to the linear regression, the GLM framework does not enforce any assumption on the distribution of the time series nor its errors. Moreover, the GLM using maximum likelihood estimation in comparison to Lasso has high computational time and a in general guarantees only a local optimum (generally non-convex). However it is well-known that Lasso estimation is a convex and efficient method providing the possibility to attain the global maximum.

IV. TEMPORAL CAUSAL ANOMALY DETECTION

As already mentioned, detecting temporal anomalies is much more challenging and complicated, but common in the real application. Especially for the heterogeneous case when one can consider time series with various distributions, the problem is even more complex. But such a scenario seems to be realistic, since it is more likely that interacting stochastic processes do not have the same distributions. Based on what we introduced in the previous sections, we are able to detect all the temporal dependencies between or within the heterogeneous time series.

Before explaining how to use the temporal dependencies in order to find the anomalies, we formally define the task of anomaly detection in this section. Then we introduce the anomaly detection algorithm and consequently an information-theoretic score.

One of the anomaly detection approaches is to build a statistical model that captures the generation process of the normal (non anomalous) data, then estimate the likelihood of a new observation based on this model and predict the data as an anomaly if the likelihood is below some thresholds. The problem of anomaly detection for multivariate time-series data can be defined as follows:

Given p time series, $x_1, \ldots, x_p$, we want to find data points (indexed by time) that significantly deviate from the "normal" pattern of the data sequence (normal does not mean necessarily Gaussian). Without loss of generality, we can reformulate the problem into the following one: Given two data sequences $X^{(r)} = \{x_i^{t,(r)}, i = 1, \ldots, p, t = 1, \ldots, T_r, \}$ and $X^{(te)} = \{x_i^{t,(te)}, i = 1, \ldots, p, t = 1, \ldots, T_{te}, \}$. $X^{(r)}$ and $X^{(te)}$ is the reference and test set, respectively. We will compute the anomaly score for $X^{(r)}$ with respect to $X^{(te)}$ for each variable to determine whether and how much each variable contributes to the difference between both time series.

A. Heterogeneous temporal causal anomalies

During the previous sections we introduced a useful and efficient tool to discover all the temporal dependencies with respect to the heterogeneous cases. Inspired by Qiu *et al.* [7] we apply this tool on the mixed time series in the sense that we capture a statistical model that fits better to the generation process of the normal data. We suppose that there is no anomaly in the training dataset therefore we capture the normal statistical model by means of using the proposed Granger-Lasso algorithm on $X^{(r)}$. The solutions for the corresponding optimization problem 11 are the Lasso coefficient which are used as the null hypothesis for the next step.

In the next step we aim to find data points that significantly deviate from the normal pattern. The null hypothesis in this step is that the temporal causal graphs of reference set and test set are the same. Therefore we apply this hypothesis as another constraint in the optimization problem so that we preserve all the temporal dependencies related to the training data:

$$\gamma_i^{(te)}(\eta_i) = \arg \min_{\gamma_i^{(te),T} \in \mathbb{R}^p} \sum_{T=d+1}^{n} (W_i^{(te),T} - X_{T,d}^{(te),Lagged} \gamma_i^{(te),T})^2 + \eta_i L_i(\gamma_i^{(te)}) + \eta_i L_i(\gamma_i^{(r)}) \tag{12}$$

where $\gamma_i^{(r)}$ is the Lasso coefficient and corresponds to the temporal dependencies in the training dataset.

B. Information theoretic anomaly score

After all the temporal dependencies were found, the next step is finding an anomaly score so that we are able to detect any anomaly. From the information theoretic perspective, the most natural difference measure between two distributions is the Kullback-Leibler (KL) divergence. This is a measure of to which extent one probability distribution diverges from the second probability distribution. This measure is not symmetric and therefore not a distance in the metric sense.

Qiu *et al.* in [7] used KL divergence as an anomaly score for a particular time-series (or feature) x_i. As the drawback of KL divergence not to be a distance metric, they had to compute both scores from training to test and vice versa and then to find the maximum score between these two scores. Jensen-Shannon divergence is symmetric and the square root of the Jensen-Shannon divergence is a metric, therefore can be used as a distance function. We take this natural alternative since the symmetry helps to avoid some computations. For a particular time series x_i Jensen-Shannon anomaly score is defined by:

$$JSD(x_i) = \frac{1}{2} D(P^{(r)} || M) + \frac{1}{2} D(M || P^{(te)}) \tag{13}$$

where $D(.)$ is the KL measure and $M = \frac{1}{2}(P^{(r)} + P^{(te)})$:

$$D(P^{(r)} || M) = \sum P^{(r)}(x_i | X^{lagged}{}_{t,d}) \frac{P^{(r)}(x_i | X^{lagged}{}_{t,d})}{P^{(te)}(x_i | X^{lagged}{}_{t,d})} \tag{14}$$

in 13 $P^{(r)}$ and $P^{(te)}$ are the underlying probability distribution function (PDF) in the reference and test datasets, respectively. We assume that the probabilities are in the exponential family e.g. Gaussian, Poisson or Gamma distribution. For example, if the underlying distribution is Poisson then $x^t{}_i \ Poisson(X^{lagged}{}_{t,d} \beta_i)$. To estimate the required parameters we consider the reference dataset for parameters in $P^{(r)}$ and test dataset for parameters in $P^{(te)}$.

C. Algorithm

In this section we describe more in detail, how to detect temporal causal anomalies. Algorithm 1 shows different steps of the proposed anomaly detection procedure based on Granger-Lasso for heterogeneous time series. As mentioned in our model (Section III) we need to transform each time series to its mean value in the sense that we will be able to use the Granger-Lasso type objective function for detecting of the temporal dependencies. Thus, for each i we first transform time series x_i time to $W^r{}_i$ so that it will be linear. Procedure 2 shows the transformation in more detail. Then we solve the corresponding Granger-Lasso optimization problem (11) to find the temporal dependency graph and the corresponding adjacency matrix respectively in reference dataset.

Function $JSD(x_i)$ measures an appropriate anomaly score for $i = 1, \ldots, p$ described in the last section based on the comparisons of every two observations in reference and test dataset.

In order to detect the anomalies more precisely we slide a window over the test data to detect not only if any anomaly occurs but also more accurately in which time window with the particular length (WS) it happens. For any specific window we again transform the time series ($W^{window}{}_i$) and solve the Granger-Lasso optimization problem. Similar to what we did on the reference dataset we compute the Jensen-Shannon divergence as an anomaly score. After sliding the window over

all the time series we are well equipped to find the anomalies. In order to do this we need to decide if the score of an observation is an anomaly with respect to a given threshold.

Algorithm 1: Heterogeneous Anomaly Detection

input : $X_i, i = 1, ..., p$; $Reference\ Data := {X_i}^r$; $Test\ Data := {X_i}^{te}$; $Window\ Size := WS$
output: list of temporal anomalies
foreach x_i **do**
  // Learn Lasso-Granger graph for reference data
  ${W^r}_i = Transform({x^r}_i)$;
  ${\beta_i}^r = LassoGranger({X^{lagged}}_{t,d}, {W^r}_i, \lambda)$;
  // Compute anomaly scores for each window
  $JSD(x_i, {\beta_i}^r)$;
  // Slide a window with size WS over the test data
  foreach *window* **do**
    ${W^{window}}_i = Transform({x_i}^{window})$;
    ${\beta_i}^{window} = LassoGranger(X_{window lagged}, W_i^{window}, \lambda)$;
    //Compute anomaly scores for each window
    $JSD(x_i^{window})$;
  end
end

Procedure 2: Transformation

Transform $(Time\ Series\ x_i)$
foreach t **do**
  //g_i is the link function corresponding to x_i e.g. g_i is log for Poisson
  ${W_i}^t = g_i(E({x_i}^t))$;
end

V. Experimental Results

The main contribution of this paper is the causality and anomaly detection dealing with heterogeneous time series i.e. many time series with various distributions from the exponential family. This section consists of experimenal results to assess the performance of the first part of the contribution, causality detection. First we explain about the data generation process and the evaluation measures. Then we report the results of applying our algorithm on two datasets. The first dataset is generated artificially by means of introduced data generation process and the second one is the dataset used by Kim *et al.* in their paper [11]. Then we compare the results of the proposed causality detection algorithm comparing to the algorithm introduced by Kim *et al.*.

A. Synthetic data generation

In this section we clarify how we generated the synthetic data since it is not straightforward to generate an appropriate dataset so that it fulfils all the constraints. In this paper we assume that we are dealing with the time series whose means depend on the past values of all time series through the link function corresponding to each distribution as we mentioned before (Table I). We define this kind of dependency between the mean values and the linear combinations via some equations. These equations are consistent to the distributions.

To generate the equations we proceed from the end of the algorithm to the beginning. The output of Granger-Lasso algorithm over heterogeneous time series is an adjacency matrix illustrating the causal relationships between different time series. The graph corresponding to this adjacency matrix is called *feature causal graph*. Therefore we start with a random adjacency matrix associated to the random feature causal graph. Then we randomly assign directed edges between the nodes showing whether any Granger-causal relations between two specific nodes (time series) exists or not. Figure 1 demonstrates the random graphical model over three time series as an example.

Having formed the feature causal graph, we then generated a graphical Granger model and the associated graph in the temporal variable space that is consistent with it. To accomplish this step we only need to choose a random lag for any established edge between two time series in feature causal graph. We selected the lag associated to each edge according to a uniform distribution within a prescribed range. For instance randomly choosing lag k for the edge $x_i \rightarrow x_j$ in the feature graph means that there is an edge $x_i^{T-k} \rightarrow x_j^T$ in graphical Granger graph. Finally we randomly assign each edge a weight, sampled from a specified range.

Now we have the equations and we need to generate the observation for each time series. After initialization we used the procedure 3 to iteratively generate the observations.

Procedure 3: Data Generation Process

Data Generator $(meanvalueequations)$
$Series$:= data matrix consisting of all the observations for all the time series;
Initialize the first observations randomly;
foreach t **do**
  Compute the mean value for each time series based on the given equations;
  foreach *time series* x_i **do**
    $Series(i,t) = t * \mu(i,t) - (t-1) * \mu(i,t-1)$;
  end
end
return $(Series)$

B. Evaluation measures

As described the output of the proposed algorithm is feature causal graph illustrating all the Granger causal relations. We now describe how to quantify the similarity between the target causal graph used to generate the synthetic data (ground truth) and the output causal graph. In this paper the metrics of *Precision*, *Recall* and $F1 - measure$, commonly used in the

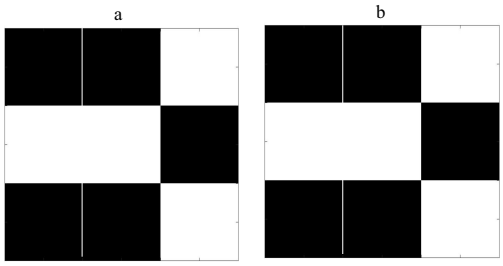


Fig. 2. **Synthetic data result** a) Adjacency matrix for the feature causal graph of the ground truth b) Achieved adjacency matrix corresponding to the feature causal graph applying our algorithm.

machine learning and information retrieval literature, are used to the problem of predicting the 0,1-label in the adjacency matrix representation of the graph.

Since the output graph is a directed one showing the causal relation in both directions, therefore we distinguish between two entries in the adjacency matrix A, $A[i,j]$ and $A[j,i]$. For instance predicting a bi-directional edge between x_i and x_j, when there is actually a directed edge from $x_i \rightarrow x_j$, would entail one correct prediction and one prediction error.

Let A^* and $\hat{A}$ denote the true adjacency matrix and the output adjacency matrix respectively. Also based on our model defined in Section III we consider $W \times W$ as the set of time series pairs. Now we define the evaluation measure:

$$Precision = \frac{|\{(i,j) \in W \times W : \hat{A}[i,j] = A^*[i,j]\}|}{|\{(i,j) \in W \times W : \hat{A}[i,j] = 1\}|}$$

$$Recall = \frac{|\{(i,j) \in W \times W : \hat{A}[i,j] = A^*[i,j]\}|}{|\{(i,j) \in W \times W : A^*[i,j] = 1\}|}$$

$$F1 - measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

There is clearly a trade-off between precision and recall as the goal of prediction, and the $F1 - measure$ tries to balance the overall quality of prediction [3].

C. First synthetic dataset

The synthetic dataset consists of three time series. x_1 and x_3 are Gaussian time series and x_2 has a Gamma distribution. We generated 3000 observations for each time series whose means depend on the past values through the corresponding link function of each distribution as we mentioned before. The following equations show how mean values depend on various time series.

$$\mu_1(T|X_{T,d}^{Lagged}) = 1 + 0.25 * x_2^{T-1},$$

$$\mu_2(T|X_{T,d}^{Lagged}) = \frac{1}{(1 + 0.125 * x_2^{T-1})},$$

$$\mu_3(T|X_{T,d}^{Lagged}) = 1 + 0.25 * x_3^{T-1} + 0.25 * x_1^{T-1}.$$

We assumed x_1 and x_3 are Gaussian time series, therefore there is a linear relation between the mean value and the linear combination of the lagged values. But considering Gamma distribution for x_2 enforces us to apply a suitable link function to link the mean value and the linear combinations. In this case as mentioned in Table I we select the inverse link function. In the following we apply the Procedure 3 to generate observations. Figure 1 is a demonstration of the current synthetic dataset.

D. Results for the first synthetic dataset

Figure 2 illustrates the result of proposed algorithm on the synthetic dataset. The left adjacency matrix Figure 2a is the ground truth and the right one (Figure 2b) is the output. As it is obviously illustrated our algorithm finds completely all the Granger causal relations between heterogeneous time series of mixed Gamma and Gaussian distributions. As we expected profiting GLM properties, Granger-Lasso algorithm is applicable for the heterogeneous case. Precision, recall and $F1 - measure$, all equal 1. Considering other advantages of Granger-Lasso e.g. time complexity and finding global optimum, our proposed algorithm efficiently works on the synthetic data.

E. Second dataset

To be able to assess all aspects of our algorithm we compare the proposed algorithm to the algorithm introduced by Kim *et al.*. Authors in [11] introduced a general statistical framework for assessing Granger causality in general and even for heterogeneous cases. In the following we call this algorithm as *GSF-Granger*.

Kim *et al.* generated a synthetic dataset to evaluate their framework which consists of three heterogeneous time series. We applied our algorithm on this dataset so that we can compare our results to them. The dataset consists of three time series. x_1 and x_3 are Gamma distributed time series and x_2 has a Poisson distribution. The following equations show how mean values depend on various time series.

$$\mu_1(T|X_{T,d}^{Lagged}) = \frac{1}{1 + 0.5 * x_1^{T-1} + x_2^{T-1} + 0.5 * x_3^{T-2}},$$

$$\mu_2(T|X_{T,d}^{Lagged}) = exp(1 + x_1^{T-1}),$$

$$\mu_3(T|X_{T,d}^{Lagged}) = \frac{1}{2 + 0.5 * x_3^{T-1}}.$$

x_1 and x_3 are Gamma distributed time series, therefore there is an inverse relation between the mean value and the linear combination of the lagged values. But considering Poisson distribution for x_2 a suitable link function (logarithm) is selected to link the mean value and the linear combinations. Analogously we applied Procedure 3 to generate the observations. Figure 3a is a demonstration of the feature causal graph related to the current synthetic dataset.

F. Results for the second dataset

Figure 3 illustrates the result of proposed algorithm for the synthetic dataset in comparison to GSF-Granger. The left adjacency matrix Figure 3b is the result of GSF-Granger which is compatible with the ground truth. The right one (Figure 3c) is the output of our algorithm.

The proposed algorithm was successful to find four relations among all five Granger causal relations. Precision $= 1$ shows that all the causal relations in the output feature causal graph are correctly established. Recall $= 0.8$ indicates that not all of the relations are found and Our algorithm was not able to find the Granger causal relation between x_2 and x_1 in both direction. GSF-Granger found all the Granger causalities as it is reported in the original paper. However referring to the $F1 - measure$ our algorithm still works efficient. Anyhow $F1 - measure = 0.89$ which indicates the overall quality of the prediction is reasonable.

VI. Consistency of Lasso

Recall that an estimator (the method providing the estimator) is prediction error consistent if the estimator converges to the optimal solution w.r.t. the increasing size of the data set. Chatterjee [12] proved the mean squared prediction consistency of a general estimator using Lasso under some assumptions:

- the observations are independent of the other observations
- are upper bounded
- have Gaussian errors with expectation zero and finite variances.

Authors in [13] generalized the mean squared prediction consistency to the case when the errors do not have to be normal errors with expectation zero but still with bounded finite variances. This allows using Lasso regression for large number of non-Gaussian variables. In this case Lasso is under some conditions still a consistent estimator.

In order to fulfil this constraint we standardized the time series by shifting each observation by means of the overall mean value so that at the end we have mean value zero and finite variances. It implies the errors will have bounded variance. In practice it is hard to guarantee that errors will have mean zero, but applying the mentioned transformation at least we make sure that the errors have bounded variance. Especially for the heterogeneous cases when we are dealing with time series from the exponential family e.g. Poisson, there is no guarantee to fulfil the constraint on the mean value.

For the future work we would try to investigate the consistency of Lasso in more theoretic manner so that we can guarantee applying Lasso on the proposed model for heterogeneous time series would be consistent.

VII. Related Work

Anomaly detection refers to the problem of finding anomalies in data. There are other synonyms of anomaly, such as outliers, contaminants, exceptions, etc. Anomaly detection algorithms find a direct application in many areas, such as insurance or health care, fraud detection for credit cards, fault detection, cyber-security or others. Most of the recent literature is written under the term outlier detection [14], [15]. Recent approaches to anomaly detection in time series range from point anomaly detection algorithms to change-point detection algorithms.

For example, Twitter'ds approach in [16] enjoys a high precision and recall and is fast, however it is specific to the use-case of Twitter. Some open-source point anomaly detection techniques are for example packages "'extremevalues"' [17], or package for outliers from [18].

To detect temporal anomalies, Rogge-Solti in [19] used probabilistic inference and proposed a Bayesian model that can be inferred from the Petri net representation of a business process. A concrete anomaly detection algorithm is usually applicable to only a specific use-case. Chandola *et al.* in [20] provided an overview of the state-of-art anomaly detection methods sorted by categories of use, concluding that only a set of anomaly models are most appropriate for a given anomaly category of interest. Laptev *et al.* in [21] created a collection of anomaly detection and forecasting models for time series called EGADS (Extensible Generic Anomaly Detection System). It is stand-alone platform that can be used as a library in larger systems. The system has a hierarchical architecture and working modules. As a subcategory, the system also considers non-normal time series. The basic idea of the applied algorithm is to find low density regions of the deviation metric distribution, for which the algorithm such as Local Outlier Factor (LOF) [22] is applied.

Qiu et al. in [7] proposed Granger graphical models as an effective and scalable approach for anomaly detection whose results can be readily interpreted. Specifically, they focused on Granger graphical models as a family of graphical models that exploit the temporal dependencies between variables by applying $L1$-regularized learning to Granger causality. The concept of using penalized auto-regression to detect the temporal Granger causality was introduced by Arnold et al. [3]. Granger-Lasso is an efficient and effective algorithm dealing with a large number of time series.

Authors in [11] introduced a general statistical framework for assessing Granger causality in general and even for heterogeneous cases. In their paper, the Granger causality from a time series x_2 to a time series x_1 is assessed based on the relative reduction of the likelihood of x_1 by the exclusion of x_2 compared to the likelihood obtained using all the time series.

VIII. Conclusion and Future Work

In this paper we proposed a modification of well-known Granger-Lasso algorithm so that it is applicable for the hetero-

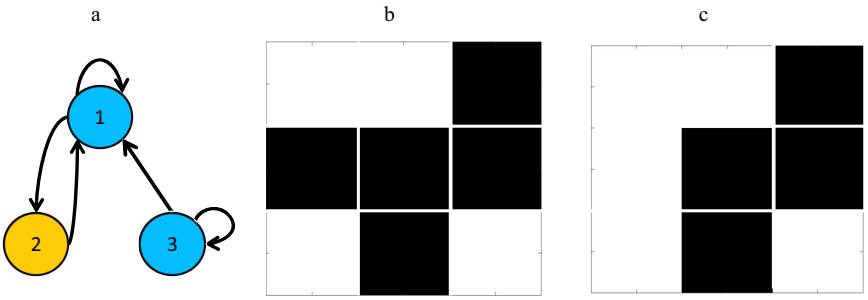


Fig. 3. **Comparing our algorithm to GSF-Granger.** (a) Feature Causal Graph (Ground truth). (b) Adjacency matrix applying GSF-Granger (c) Adjacency matrix applying the proposed algorithm.

geneous time series. In order to provide the required conditions to apply Granger-Lasso for such a complex system we utilized the GLM advantages. We generated our graphical Granger model over the transformed time series in the GLM framework considering this fact that all the Granger causal relationships between the original time series will be preserved if we apply the Granger-Lasso on the transformed time series. In this case we can deliver one of the Lasso consistency constraint that is linearity.

Finally we proposed an application of the new algorithm in anomaly detection where we were interested to find dependency anomalies in heterogeneous time series by means of Granger-Lasso algorithm. We also introduced Jensen-Shannon divergence as an information theoretic anomaly score which is symmetric and saves time complexity by avoiding extra computations.

At the end we investigated the consistency of Lasso algorithm under different conditions. As mentioned Lasso algorithm is consistent under the assumptions which are in practice hard to guarantee particularly for non-Gaussian time series.

Due to time limitation we have done no experimental result on the real datasets so far. For the future we will evaluate the proposed causality detection algorithm on different real datasets where we are able to interpret the results meaningfully. We are also interested to assess the application of temporal causal detection algorithm in discovery heterogeneous anomalies. To cope with the consistency problem we will try different variations of Lasso which they are more appropriate to apply on the non-Gaussian time series.

REFERENCES

[1] K. P. Murphy and S. Russell, "Dynamic bayesian networks: representation, inference and learning," 2002.

[2] C. W. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica: Journal of the Econometric Society*, pp. 424–438, 1969.

[3] A. Arnold, Y. Liu, and N. Abe, "Temporal causal modelling with graphical granger methods," *ACM SIGKDD*, 2007.

[4] A. C. Lozano, N. Abe, Y. Liu, and S. Rosset, "Grouped graphical granger modeling for gene expression regulatory networks discovery," *Bioinformatics*, vol. 25, no. 12, pp. i110–i118, 2009.

[5] A. Shojaie and G. Michailidis, "Discovering graphical granger causality using the truncating lasso penalty," *Bioinformatics*, vol. 26, no. 18, pp. i517–i523, 2010.

[6] K. Hlavackova-Schindler, V. Naumova, and J. Pereverzyev, S., "Multi-penalty regularization for detecting relevant variables." in *Recent Applications of Harmonic Analysis to Function Spaces, Differential Equations, and Data Science*, ser. Novel Methods in Harmonic Analysis, I. Pesenson, Q. L. Gia, A. Mayeli, H. Mhaskar, and D.-X. Zhou, Eds. Basel: Birkhäuser Basel, June 2017, vol. 2, pp. 889–916.

[7] H. Qiu, Y. Liu, N. A. Subrahmanya, and W. Li, "Granger causality for time-series anomaly detection," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 2012, pp. 1074–1079.

[8] H. Zou, "The adaptive lasso and its oracle properties," *Journal of the American statistical association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[9] C. Zou and J. Feng, "Granger causality vs. dynamic bayesian network inference: a comparative study," *BMC bioinformatics*, vol. 10, no. 1, p. 122, 2009.

[10] J. A. Nelder and R. J. Baker, "Generalized linear models," *Encyclopedia of statistical sciences*, 1972.

[11] S. Kim and E. N. Brown, "A general statistical framework for assessing granger causality," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2222–2225.

[12] S. Chatterjee, "Assumptionless consistency of the lasso," *arXiv preprint arXiv:1303.5817*, 2013.

[13] K. Schindlerova, "Prediction consistency of lasso regression does not need normal errors." *British Journal of Mathematics and Computer Science*, vol. 19, pp. 1–7, October 2016.

[14] C. C. Aggarwal and S. Sathe, *Outlier Ensembles: An Introduction*. Springer, 2017.

[15] A. Anderson, J. Kleinberg, and S. Mullainathan, "Assessing human error against a benchmark of perfection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 11, no. 4, p. 45, 2017.

[16] O. Vallis, J. Hochenbaum, and A. Kejariwal, "A novel technique for long-term anomaly detection in the cloud," in *6th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 14)*. Philadelphia, PA: USENIX Association, 2014.

[17] M. P. van der Loo, "Detection of outliers with the extremevalues package," in *useR2010*, 2010.

[18] L. Komsta, "Outliers: Tests for outliers. r package version 0.14."

[19] A. Rogge-Solti and G. Kasneci, *Temporal Anomaly Detection in Business Processes*. Cham: Springer International Publishing, 2014, pp. 234–249.

[20] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.

[21] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: ACM, 2015, pp. 1939–1947.

[22] M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *PROCEEDINGS OF THE 2000 ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA*. ACM, 2000, pp. 93–104.

CHAPTER 10

Paper F & Paper G: Anomaly Detection in Heterogeneous Time Series by Causality Mining

This chapter comprises two publications concerning causal inference on single-type heterogeneous data (Paper F: Granger Causality for Heterogeneous Processes [10]) and its application for detecting dependency anomalies among time series (Paper G: Anomaly Detection in Heterogeneous Time Series by Causality Mining [14]). Paper G is is included in this chapter which is an extended journal version of Paper F.

Authors Contributions for Paper F:

- **Sahar Behzadi.** Cooperation on the main idea and developing the algorithm as well as writing the paper; Implementation; Conducting experiments.
- **Kateřina Hlaváčková-Schindler.** Cooperation on the main idea, developing the algorithm, experiments as well as writing the paper.
- **Claudia Plant.** Supervision during development and evaluation of the algorithm as well as writing the paper.

Authors Contributions for Paper G:

- **Sahar Behzadi.** Cooperation on the main idea and developing the algorithm as well as writing the paper; Implementation; Conducting experiments.

- **Niklas Preschern.** Cooperation on developing the algorithm; Implementation; Conducting experiments.

- **Kateřina Hlaváčková-Schindler.** Cooperation on writing the paper.

- **Claudia Plant.** Supervision during development and evaluation of the algorithm as well as writing the paper.

Noname manuscript No.
(will be inserted by the editor)

Anomaly Detection in Heterogeneous Time Series by Causality Mining

Sahar Behzadi · Niklas Preschern · Kateřina Hlaváčková-Schindler · Claudia Plant

Received: date / Accepted: date

Extension Statement: This paper is an extended version of the recently published paper "Granger Causality for Heterogeneous Processes" [4]. In this paper [4], we proposed a graphical Granger model (HGGM) to discover temporal dependencies among time series from different distributions, i.e. heterogeneous data sets. Hereby, we incorporate HGGM in order to introduce a general anomaly detection framework applicable for heterogeneous complex data sets. The proposed framework and the algorithm are describe in Section 6. We conducted various new extensive experiments concerning different aspects of our algorithm in Section 6.4. Following results of the experiments, our proposed algorithm (AD-HGGM) outperforms other state-of-the-arts in this respect.

S. Behzadi
Faculty of Computer Science, Data Mining, University of Vienna
Vienna, Austria
E-mail: sahar.behzadi@univie.ac.at

N. Preschern
Faculty of Computer Science, Data Mining, University of Vienna
Vienna, Austria
E-mail: a01246683@unet.univie.ac.at

K. Hlaváčková-Schindler
Faculty of Computer Science, Data Mining, University of Vienna
Vienna, Austria
E-mail: katerina.schindlerova@unet.univie.ac.at

C. Plant
Faculty of Computer Science, Data Mining, University of Vienna and ds:Univie
Vienna, Austria
E-mail: claudia.plant@univie.ac.at

Abstract Detecting anomalies among time series have recently attracted attention of the data mining community. Among various types of anomalies, dependency anomalies, resulted by changes of temporal dependencies, are more challenging to detect due to complex temporal structures and causal interactions. Among various causal notions graphical Granger causality is well-known due to its intuitive interpretation and computational simplicity. Most of the current graphical approaches are designed for homogeneous data sets i.e. the interacting processes are assumed to have the same probability distribution. Since many applications generate time series of heterogeneous nature, the question arises how to leverage graphical Granger models to detect temporal causal dependencies among them. Profiting from the generalized linear models, we introduce an efficient **H**eterogeneous **G**raphical **G**ranger **M**odel (HGGM) for detecting causal relations among time series having a distribution from the exponential family which includes a wider common distributions e.g. Poisson, gamma. To guarantee the consistency of the algorithm, adaptive Lasso as a variable selection method is employed. Incorporating the introduced algorithm (HGGM), we propose a general anomaly detection framework (AD–HGGM) to specify dependency anomalies in heterogeneous data sets. Moreover, we introduce a symmetric information-theoretic anomaly score to measure anomalous deviations. Extensive experiments on synthetic and real data confirm the effectiveness and efficiency of HGGM as well as AD–HGGM.

Keywords Time Series · Anomaly detection · Granger causality · Heterogeneous data sets

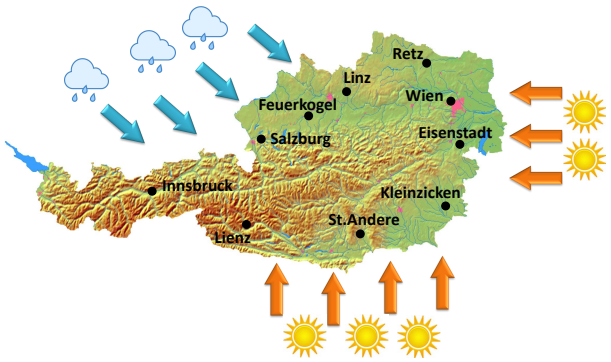


Fig. 1 Meteorological stations and three major weather systems influencing Austria.

1 Introduction

Recently there is a significant interest in anomaly detection among time series in data mining community. Classifying multivariate time series data, there are two types of anomalies:

- univariate anomaly: anomalies occur only within individual time series,
- dependency anomaly: anomalies occur due to changes of temporal dependencies among various time series.

As expected, the second type of anomalies, dependency anomalies, are more challenging to detect due to complex temporal structures and interactions among time series. In this regard, usually, discovery of causal relations among different processes leads to characterize the evolution in time of regular observations. The regular pattern can be used to detect the deviated observations or outliers in anomaly detection [19]. A number of methods has been developed to infer causal relations from time series data by Granger causality [10] which is a popular method due to its computational simplicity. The presumption of this approach is that a cause helps to predict its effects in the future. Most of the existing methods in this area assume additive causal interactions among time series following a specific data type or a certain distribution. The well-know causality notion, Additive Noise Models (ANMs), have been proposed for either continuous [22] or discrete [18] time series. Moreover, most of the probabilistic approaches are designed for homogeneous data sets [6], [5]. However, in reality the interacting processes do not have to be homogeneous (having the same distribution). Such situations can occur, for example, in climatology when various measurements are provided for different meteorological stations. Figure 1 shows 10 weather stations and three major weather systems in Austria. The monthly amount of precipitation as well as the number of sunny days have been measured for every station, each of which with a non-Gaussian distribution. One can be interested in investigating how the number of sunny days in a station, influenced by one of the weather systems, can impact the amount of precipitation in the other locations.

Applying existing algorithms on such data sets can result an inaccurate Granger causal model since they have been designed for specific homogeneous data types. Moreover, the small set of algorithms, which are supposed to cope with the heterogeneity, mostly employ an exhaustive pairwise testing. This leads to inefficiency in a causal network discovery specially when the number of interacting processes is increasing. In between, graphical Granger models are popular due to their efficiency and effectiveness. They employ a penalized Vector Autoregression (VAR) to the Granger concept [1], [3], [9], [23]. However, to the best of our knowledge, so far they have been designed only for homogeneous data sets. Thus, in this paper we introduce a penalized VAR-based algorithm to detect the **H**eterogeneous **G**raphical **G**ranger **M**odel (HGGM) by employing generalized linear models (GLMs) [4]. Similar to the other graphical models, we assume that the interactions among the involved processes are additive. In order to ensure the convergence of HGGM to the true causal graph (i.e. consistency), we employ the well-know penalization approach, adaptive Lasso, with oracle properties [26].

Moreover, detected temporal causal graphs can be used to specify dependency anomalies in early stages. That is, employing our graphical Granger approach (HGGM), we propose a general framework (AD–HGGM) to detect anomalies among time series in heterogeneous data sets. AD–HGGM consists of three main building blocks:

- Discovery of causal dependencies,
- Measuring the level of anomalies,
- Efficiently specifying anomalies.

In order to measure the level of anomalies, we employ *Jensen-Shannon Divergence* (JSD) as a symmetric anomaly score. The paper brings the following contributions:

- **Heterogeneity:** Applying the GLM methodology, we introduce a heterogeneous graphical Granger model to discover the causal interactions among a wide variety of heterogeneous time series from the exponential family (HGGM [4]). Moreover, employing HGGM to detect dependency anomalies leads to an effective approach (AD–HGGM) for heterogeneous cases ;

- **Consistency:** Assessing the causal relations via adaptive Lasso ensures consistency of HGGM;

– **Scalability:** Unlike other existing algorithms, HGGM avoids an exhaustive pairwise causality testing by penalized estimation of VAR models. Due to the computational simplicity of HGGM, it is convenient to be used in practice. Moreover, its reasonable runtime complexity makes the algorithm scalable for the large data sets consisting of long time series. As a consequence, AD–HGGM (designed based on HGGM) is also reasonably efficient in terms of runtime complexity and comparing to others;

– **Effectiveness:** Following results of the extensive experiments on synthetic and real data sets, HGGM is an effective algorithm even by detecting sparse causal graphs. Moreover, our experiments concerning the effectiveness of AD–HGGM confirms advantages of employing HGGM to detect anomalies in heterogeneous data sets.

– **Efficiency:** Due to our proposed symmetric anomaly score based on Jensen-Shannon Divergence, AD–HGGM avoids inessential computations which leads to an efficient algorithm compared to others.

In the following we specify the problem and the theoretical background and propose our HGGM model. Section 2 presents the related work. In Section 3, we introduce the problem and our proposed framework to deal with heterogeneous data. In Section 4 we introduce our integrative algorithm HGGM and the theoretical considerations of it. Extensive experiments on synthetic and real data are demonstrated in Section 5. In Section 6, we incorporate HGGM and introduce an effective anomaly detection algorithm (AD–HGGM) appropriate for dependency anomaly detection among heterogeneous time series. Our conclusion is in Section 7.

2 Related Work

Anomaly detection, referring to the problem of finding anomalies in data, attracted much attention in various applications in many areas, such as health care, fraud detection for credit cards, cyber-security an so on. Recent approaches to anomaly detection in time series range from point anomaly detection algorithms to change-point detection algorithms [25], [13].

Authors in [20] used probabilistic inference and proposed a Bayesian model that can be inferred from the Petri net representation of a business process to detect temporal anomalies. Chandola *et al.* in [8] provided an overview of the state-of-art anomaly detection methods sorted by categories of use. Laptev *et al.* in [12] created a collection of anomaly detection and forecasting models for time series in which the basic idea is to find low density regions of the deviation metric distribution.

On the other side, Qiu *et al.* employed graphical Granger models as an effective and scalable approach for anomaly detection which their results can be readily interpreted [19]. They focused on Granger causality to exploit temporal dependencies among time series applying L_1-regularization. Granger causality [10] is well-known due to its simplicity and computational efficiency. It states that a cause efficiently improves the predictability of its effect. There are various approaches depending on how to assess the predictability. Probabilistic approaches interpret it as the improvement in the likelihood (i.e. probability). However, several methods in this group are distinguished based on the way how they employ probability. Information-theoretic methods detect the causal direction by introducing some indicators. Among them, compression-based algorithms apply the Kolmogorov complexity and define a causal indicator by mean of the Minimum Description Length (MDL) [6], [5], [7]. Essentially, these algorithms are designed to infer the pairwise causal relations. Therefore, employing them for discovery of causal networks leads to inefficiency, especially when the number of processes increases. Moreover, to the best of our knowledge, almost all the algorithms in this category deal with homogeneous data sets except *Crack* [14], the most recent compression-based algorithm to deal with multivariate and heterogeneous processes. Beside the pairwise testing and its drawbacks, this algorithms lacks the accurate causal relations since there is no lag parameter considered in this approach. Transfer entropy, shortly TEN, is another approach among information-theoretic methods which is based on Shannon's Entropy [21]. In this approach it is more likely that the causal direction with the lower entropy corresponds to the true causal relation. Given a lag variable, TEN can detect both linear and non-linear causal relations. However, due to pairwise testing and its dependency on the lag variable, the computational complexity of this algorithm is exponential in the lag parameter. Moreover, similar to compression-based methods, TEN is not designed to deal with bidirectional causalities. As another method in this category, the authors in [11] employ the log-likelihood ratio to detect any causal relations among processes. They propose a statistical framework (SFGC) for mixed type data and assessing the causal relations between multiple time series is accomplished by the false discovery rate (FDR). The statistical power of the FDR based methods rapidly decreases with increasing number of hypotheses and these methods are com-

putationally intensive. As a consequence, the statistical efficiency of SFGC decreases for the increasing number of investigated time series. Another approach to assess the predictability is the graphical Granger method where a penalized VAR model is supposed to be estimated [1], [23]. Graphical Granger method is popular for its simplicity and efficiency since employing a penalized VAR model we avoid the pairwise testing. However most of the algorithms in this category are designed for Gaussian processes. Utilizing the advantages of this approach we introduced a graphical Granger algorithm for heterogeneous processes.

3 Theory

3.1 Granger Causality

Granger causality is a well-studied causality notion introduced by Granger in the field of econometric [10]. Granger causal inference captures the temporal dependencies among time series providing useful information although it is not meant to be equivalent to the true causality. In a bivariate case, let $x^{1:n} = \{x^t | t = 1, \dots, n\}$ and $y^{1:n} = \{y^t | t = 1, \dots, n\}$ denote two non-stationary time series up to time n. Moreover, let Model 1 represent autoregressive (AR) model corresponding to time series y and Model 2 show the augmented AR model taking past observations of x into consideration.

$$y^T = \alpha_1 y^1 + \cdots + \alpha_{T-1} y^{T-1} + \varepsilon^T \tag{1}$$

$$y^T = \alpha_1 y^1 + \cdots + \alpha_{T-1} y^{T-1} + \gamma_1 x^1 + \cdots + \gamma_{T-1} x^{T-1} + \varepsilon^T \tag{2}$$

Granger causality states that x Granger–causes y if the AR Model 2 significantly improves the predictability of y comparing to the Model 1. The concept of Granger causality can be extended to more than two time series. Let $x_1^{1:n}, \dots, x_p^{1:n}$ denote p time series up to time n and X^T be the concatenated vector of all time series at time T, i.e. $X^T = (x_1^T, ..., x_p^T)$. Thus, the VAR model w.r.t. X^T is given by:

$$X^T = A_1 X^1 + \cdots + A_{T-1} X^{T-1} + \varepsilon^T \tag{3}$$

where A_t is a matrix of the regression coefficients at time $t = 1, \dots, T-1$ and ε^t is an additive white noise. Thus, x_j Granger–causes x_i if at least one of the $(i,j)th$ elements in the coefficient matrices $A_1, \dots, A_{T-1}$ is non-zero.

3.2 Causal Inference by Penalization

In order to detect the causal relations between several time series, estimating coefficients of the VAR model introduced in the last section is essential. This problem can be ill-posed. Therefore, penalizing VAR models of order d (a time window) by means of a penalty function provides an efficient and sparse solution while the convergence to the true causal graph is ensured (e.g. [1], [23]). This approach is referred to as variable selection as well since only features with strong dependencies can survive. Thus, for time series $x_i, \quad i = 1, ..., p$, we consider the VAR model including all p time series and slide the window of size d over time series and get the corresponding VAR model. The fact is, best regressors with the least squared error for every specific time series will have non-zero coefficients in corresponding VAR model only for the dependent time series. More precisely, Let $X_{T,d}^{Lag} = \{{x_i}^{T-t} | i = 1, ..., p; \quad t = 1, ..., d\}$ denote the concatenated vector of all the lagged variables up to time T for a given time window of length d. In this paper, we consider the same lag d for each time series for simplicity. Therefore, the penalized least square estimation of coefficients, i.e. the variable selection problem for the time series x_i is given by:

$$\hat{\beta}_i = \arg\min_{\beta_i} \sum_{T=d+1}^{n} \left(x_i^T - X_{T,d}^{Lag} \beta_i\right)^2 + \lambda R(\beta_i) \tag{4}$$

where λ is the regularization parameter and $R(.)$ is the penalty function. $\hat{\beta}_i = (\beta_1, ..., \beta_p)$ is the concatenated vector of regression coefficients $\beta_1, ..., \beta_p$ w.r.t. to time series $x_1, ..., x_p$. Considering the definition of Granger causality, x_j Granger–causes x_i when at least one of the coefficients in β_j is non-zero.

3.3 Adaptive Lasso

One of the well-known penalization approaches as well as variable selection methods is Lasso [24]. The Lasso penalty function considered in Equation 4 is the L_1 norm of the coefficients, i.e. $R(\beta_i) = ||\beta_i||_1$. Despite the efficiency of Lasso, consistency [1] of this approach is not guaranteed. Here, we incorporate adaptive Lasso [26], modified Lasso penalty function, as the variable selection method in our model due to its consistency as well as its oracle properties. In adaptive Lasso, adaptive weights are assigned to penalize the L_1 norm of

[1] I.e. the resulting sequence of estimates does not have to converge in probability to the optimal solution for variable selection under certain conditions (Section 2 in [26]).

coefficients. Thus, the penalty function is given by:

$$R(\beta_i) := \sum_{j=1}^{p} w_j|\beta_j| \quad where \quad w_j = \frac{1}{|\hat{\beta}_j^{(mle)}|^{\omega}} \tag{5}$$

where w_j is the weight vector for some $\omega > 0$ and $\hat{\beta}_j^{(mle)}$ is the maximum likelihood estimate of the parameters. The consistency of adaptive Lasso is proven under some mild regularity conditions in the following theorem [26]:

Theorem 1 *Let $\mathcal{A} = \{i : \hat{\beta}_i \neq 0\}$ be the set of all non-zero coefficient estimates. Suppose that $\lambda/\sqrt{n} \to 0$ and $\lambda n^{\frac{(\omega-1)}{2}} \to \infty$ then under some mild regularity conditions adaptive Lasso must be consistent for the variable selection.*

3.4 Heterogeneous Granger Causality

Most Granger causal inference approaches consider certain Gaussian assumptions for the interacting time series. However, in the reality this assumption leads to an inaccurate causal model since there are many processes which do not follow a Gaussian distribution. Moreover, mostly the VAR penalization approaches are consistent under additional specific conditions on Gaussian time series, see e.g. [1]. Incorporating GLM frameworks, we introduce a general integrative model to infer causal relations among a large number of time series from various distributions.

GLM was first introduced by Nelder *et al.* in [17] and it is a natural extension of linear regression. In this case the regressed variables do not have to necessarily follow a Gaussian distributions but they (time series) can have any distribution from the exponential family. Thus, the relation among the response variable and the covariates in a regression is defined by a link function g, a monotone twice differentiable function depending on concrete distribution functions from the exponential family, and do not have to be linear any more.

In our model, we assume the mean value of each time series at time T depends on its own history and the past values of the concurrent time series such that:

$$E(x_i^T) = g_i^{-1}(X_{T,d}^{Lag}.\beta_i). \tag{6}$$

Thus, our general objective function is defined as:

$$\hat{\beta}_i = \arg\min_{\beta_i} \sum_{T=d+1}^{n} \big[- x_i^T (X_{T,d}^{Lag}.\beta_i) + g_i^{-1}(X_{T,d}^{Lag}.\beta_i)\big]$$
$$+\lambda. \sum_{j=1}^{p} w_j|\beta_j|. \tag{7}$$

When x_i follows binomial and Poisson distribution, respectively, the concrete form of our proposed objective function (7) is defined as:

$$\hat{\beta}_i = \arg\min_{\beta_i} \sum_{T=d+1}^{n} \big[-x_i^T (X_{T,d}^{Lag}.\beta_i) + log(1+e^{(X_{T,d}^{Lag}.\beta_i)})\big]$$
$$+\lambda. \sum_{j=1}^{p} w_j|\beta_j|, \tag{8}$$

$$\hat{\beta}_i = \arg\min_{\beta_i} \sum_{T=d+1}^{n} \big[- x_i^T (X_{T,d}^{Lag}.\beta_i) + \exp(X_{T,d}^{Lag}.\beta_i)\big]$$
$$+\lambda. \sum_{j=1}^{p} w_j|\beta_j|. \tag{9}$$

4 HGGM Algorithm

Algorithm 1 summarizes the proposed method, HGGM. In the beginning, the overall lagged matrix X^{Lag} is constructed by sliding a window of size d over each time series. Then, we need to optimize the introduced objective function by solving the optimization problem (Equation 7) for each time series. This can be done by calling the procedure $GLM - penalize()$ [15] where it employs Fisher scoring algorithm estimating the regression coefficients. Incorporating the cross-validation to find the best regularization parameter, we set the maximum λ as an input of $GLM - penalize()$.

Essentially, in a GLM framework, one needs to know the distribution of the time series x_i in advance to apply an appropriate link function g w.r.t $x_i s$. However, in the reality, it is not straightforward to guess the correct distribution. Thus, we utilize a statistical fitting procedure to find the most accurate distribution for every time series. That is, we incorporate Akaike Information Criterion (AIC) and assign the distribution from the exponential family with the least AIC to every time series. Finally, based on the definition of Granger causality we get pairwise Granger-causal relations among p time series out of which we construct the adjacency matrix corresponding to the final causal graph.

Consistency: Considering adaptive Lasso for variable selection, the consistency of this approach has been proven under some mild regularity conditions (Section 3). Thus, incorporating the adaptive Lasso for GLMs leads to the following statement about the consistency of HGGM.

Algorithm 1 Causal Detection by HGGM

HGGM $(x_i, g_i, i = 1, \ldots, p; d; \lambda_{max})$
Adj := adjacency matrix of the output graph
X^{lag} := lagged matrix of all temporal variables
// find Granger causalities for each feature
for all x_i **do**
  // solve the penalized optimization problem considering lagged variables
  $\beta_i = GLM - penalize(X^{Lag}, x_i, g_i, \lambda_{max}, d)$; // β_i := coefficients w.r.t x_i
  for all ${\beta_i}^j$ sub-vectors of β_i **do**
    $Adj(j, i) = 0$ //discover Granger-causalities
    if $(\exists t, 1 < t < d$ such that ${\beta_i}^j(t) > 0)$ **then**
      $Adj(j, i) = 1$
    end if
  end for
end for
return (Adj)

Corollary 1 *Assume G be a true Granger causal graph corresponding to p time series, each of length n. Let the regularization parameter λ fullfils the conditions of Theorem 1. Then taking p time series as input, HGGM outputs a causal graph which converges to the true graph G with probability approaching 1 as $n \to \infty$.*

Proof When $n \to \infty$ the conditions of Theorem 1 are fulfilled. Therefore it follows that the procedure *GLM–penalize(.)* in Algorithm 1 converges to the true Granger causal graph. Thus, HGGM is consistent as well.

Computational Complexity: In order to investigate causal relationships for every time series $x_i, i = 1, \ldots, p$, we need to fit and find the most accurate VAR model following the proposed objective function (7). that is, we have p regression models each of which consisting of d lagged variables corresponding to $x_1, \ldots, x_p$ at any time. As mentioned in HGGM we employ Fisher scoring approach to estimate the parameters of VAR models and hence, the number of computations required to solve a VAR of order d is $\mathcal{O}(d^2)$. Thus, the computational complexity of HGGM is in order of $\mathcal{O}(np^2d^2)$.

5 Experimental Results

Here, we investigate efficiency and effectiveness of HGGM comparing to other Granger causal inference algorithms. As an evaluation measure we employ *F-measure* which takes both precision and recall into account. Although there are many approaches designed for Granger causal inference, only few of them are applicable for heterogeneous data sets where a mix of time series of various distributions are given. Therefore, we focus on three algorithms which are applicable to mixed time series, i.e. transfer entropy, shortly TEN [21], Crack [14] and SFGC [11] and compare our algorithm to them in various aspects.

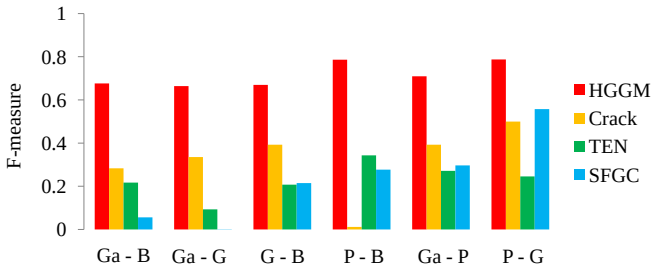


Fig. 2 Comparing performance of HGGM to others in various heterogeneous data sets. Ga: Gamma, G: Gaussian, B: Bernoulli, P: Poisson.

Conducting extensive experiments on synthetic and real–world data sets, we investigate the effectiveness and efficiency of HGGM comparing to other algorithms. HGGM is implemented in MATLAB and the source code and data sets are publicly available at: `https://bit.ly/2FkUB3Q`. We use the publicly available implementations and recommended parameter settings for other comparison methods.

5.1 Synthetic Heterogeneous Data Sets

In the beginning, the effectiveness of HGGM is investigated in comparison to other algorithms in terms of *F-measure*. That is achieved by conducting various experiments each of which concerning a unique characteristic. In the following, we focus on the scalability of HGGM varying the number of interacting time series and the length of them. We report the average performance of 50 iterations performed on different data sets with the given characteristics in every synthetic experiment. Unless otherwise mentioned, we generated time series are of length 1,000 (except for the experiment on increasing the length). In order to be fair in every experiment, we run the algorithm for various lags and take the average F-measure as the final result for the algorithms which require a user to specify the lag variable.

Effectiveness: HGGM is designed to deal with Gaussian as well as non- Gaussian time series having a distribution from the exponential family. In this experiment we generated time series with various combinations of Gaussian and non-Gaussian distributions in order to assess HGGM in various cases. Figure 2 shows that HGGM outperforms other algorithms in various combinations of Gaussian – non-Gaussian distributions and discrete – continuous time series. It confirms effectiveness of the GLM-based objective function cop-

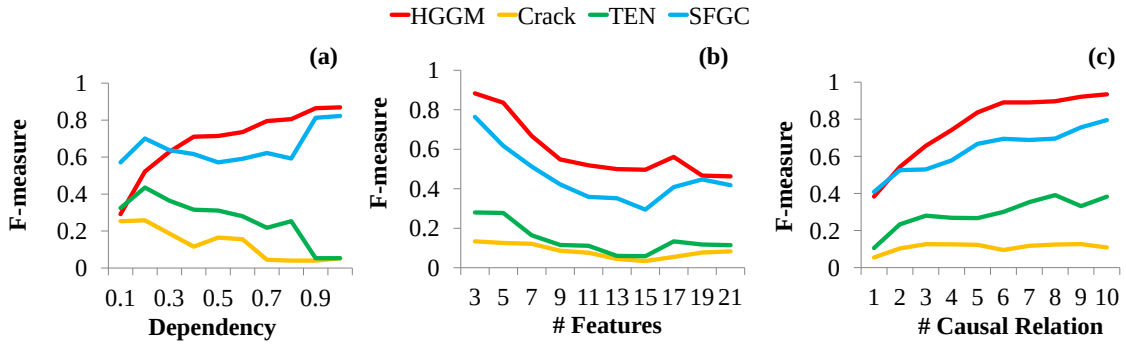


Fig. 3 Investigating the performance of HGGM comparing to others in various aspects.

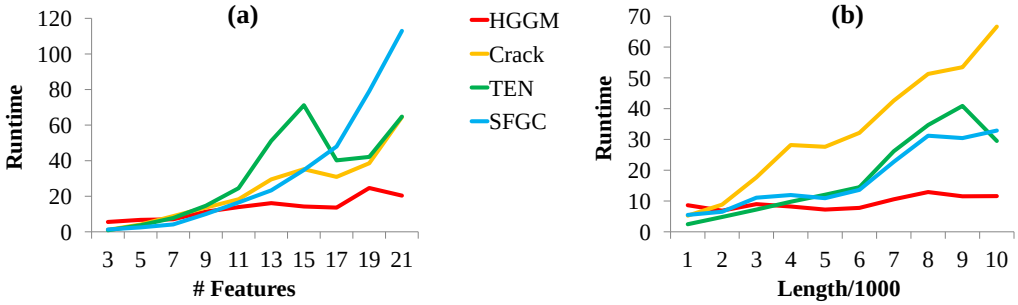


Fig. 4 Investigating computational efficiency of HGGM comparing to others in terms of runtime in seconds.

ping with heterogeneity of time series in comparison to others. We focus on synthetic data sets for the rest of the experiments where a mixture of Poisson – Gaussian time series are generated as a representative for heterogeneous data sets.

Dependency: We refer to the coefficients in VAR model as dependency among interacting time series. In this experiment, the performance of HGGM in compared to others when the dependency is increasing ranging from 0.1 to 1. Figure 3 a illustrates how various algorithms perform in this experiment. When increasing dependency, as expected, HGGM and SFGC have an ascending trend while the effectiveness of Crack and TEN is decreasing surprisingly. Although the performance of HGGM is less than SFGC and TEN in a very early stage, it outperforms other algorithm for the dependencies higher than 0.3 with a wide margin.

Increasing the Number of Time Series: Does the number of involved time series in a causal graph influences the performance of algorithms? Investigating this question, we increase the number of time series (features) iteratively in order to compare the performance of the algorithms when many time series are involved i.e. corresponding causal graphs are more complex. Figure 3 b confirms a descending trend in terms of F-measure for every algorithm while HGGM is still more efficient than others in any case. There is a big gap among the performance of two algorithms, Crack and TEN, comparing to HGGM in this figure. As a justification, these algorithms are not designed to deal with bidirectional causalities. Thus, by increasing the number of time series, it affects the performance more and more.

Causal Relations: How does the sparsity of true causal graph affects the performance of various algorithms? In this experiment, we address this question varying the number of causal relations among 5 mixed time series from Poisson – Gaussian combination. As expected, the effectiveness is in a direct relation with the density of the true causal graph. That is, the performance of algorithms is increasing when the density is increasing too. However, Figure 3 c illustrates the superiority of our algorithm in terms of performance comparing to others even for sparse graphs.

Scalability: We investigate algorithms in terms of scalability conducting two various experiments. First, the number of interacting time series is increasing iteratively where the length is set to 1,000 i.e. $n = 1,000$. In the other experiment, we, every time, generate four time series while the length of them n is ranging from 1,000 to 10,000. The superiority of HGGM is shown in Figure 4 a for the first experiment when the number of time series (features) is bigger than 6 comparing to Crack and TEN and bigger than 9 comparing to SFGC.

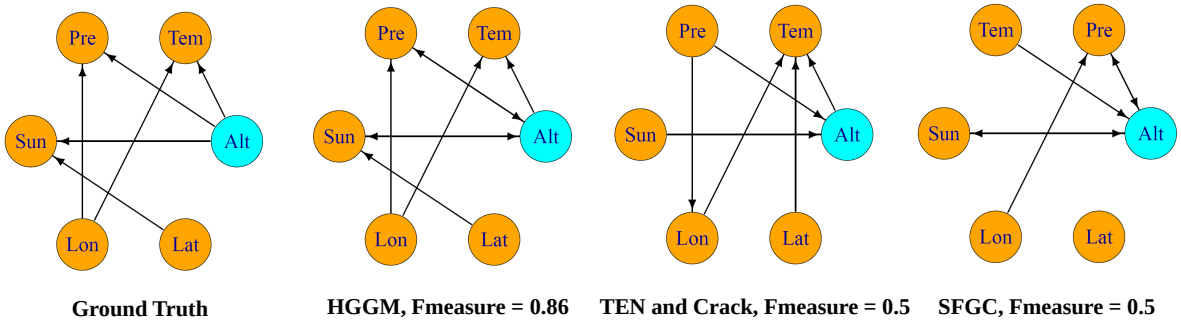


Fig. 5 Comparing HGGM to others on German weather data set.

As shown in Figure 4 b, focusing on the result of the second experiment, increasing the length of time series, HGGM is the fastest algorithm almost always for the time series longer than 2,000. Thus, the efficiency of the proposed algorithm (HGGM) is confirmed considering both experiments.

5.2 Real-world Applications

We conducted the experiments on publicly available real data sets considering two cases, whether a ground truth is given or not. In order to be fair in the real experiments we set $d = 15$ for all the algorithms which require a lag variable.

Weather in Germany: The first data set *DWD* [2] is a climatological data consisting of 6 measurements, temperature, sunshine hours, altitude, precipitation, longitude and latitude for 394 weather stations all over Germany. The altitude measurement is already provided in a discrete time series while all other measurements are continuous. Applying the statistical fitting procedure (Section 4), we assign Gaussian distribution for all continuous time series and the Poisson distribution for the altitude. The ground truth is available in [16] which is provided by pairwise causal relations. In order to be fair by evaluating the results of the algorithms, we do not consider the causal interactions where no information is provided. Figure 5 shows the performance of HGGM comparing to other algorithms in terms of $F - measure$. HGGM ably finds all the existing causal relations. However, it detects causal relations where sunshine and temperature cause altitude.

Marks: The next two data sets together with the corresponding ground truth are publicly available [3]. *Marks* data set concerns the examination marks of 88 students on five different topics. The given true causal graph reveals any impacts the grades of a topic could have on the other topics. We assign Poisson distribution to any topic. In this experiment HGGM ($F - measure = 0.74$) was able to outperform TEN (0.55), Crack (0.6) and SFGC (0.71).

Gaussian: The Gaussian data set is a simulated data showing the causal interactions among 7 Gaussian time series. The time series are of the length 5,000. HGGM ($F - measure = 0.4$) performs more accurately comparing to other algorithms, TEN (0), Crack (0.14) and SFGC (0.14), although non of the algorithms was able to capture all the causal relations in the ground truth.

Austrian climatological data set: As a real world application we investigate causal spatio-temporal interactions among climatological phenomena for 10 sites uniformly distributed in Austria (Fig. 1).

For any site we used the monthly measurements of precipitation and of the number of sunny days for 26 months. Employing the statistical fitting, we consider a Gamma distribution for the precipitation and a Poisson distribution for the number of sunny days. Figure 6-7 show the complete Granger causal graphs running various algorithms. However, we focus on the station *Feuerkogel* to better interpret the results. Moreover, the real meteorological data set is publicly available [4]. Essentially, Austrian weather is influenced by three climatic systems while any system has its own characteristics. Concerning the interpretation of results for the selected station, we concentrate on the Atlantic maritime climate from the north-west which is characterized by low-pressure fronts, mild air from the Gulf Stream, and precipitation [2]. The northern slopes of the Alps, the Northern Alpine Foreland, and the Danube valley are influenced by the Atlantic weather system.

[2] http://www.dwd.de/DE/Home/home_node.html
[3] Http://www.bnlearn.com/documentation
[4] Https://www.zamg.ac.at

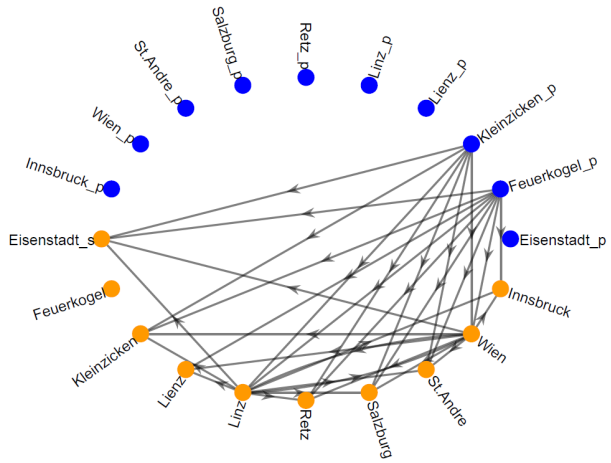


Fig. 6 Experiment on the Austrian data. Result of HGGM algorithm.

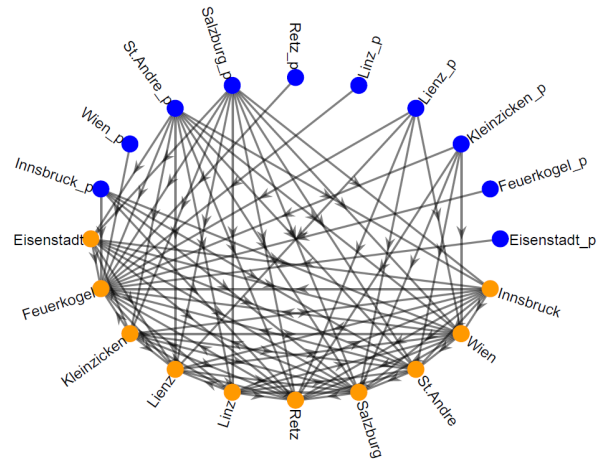


Fig. 7 Experiment on the Austrian data. Result of Crack algorithm.

The next weather system is continental climate which is mostly characterized by low pressure fronts with precipitation in summer and high pressure systems with cold and dry air in winter. Mainly eastern Austria (*Retz*, *Wien*, *Eisenstadt*) is affected by the continental weather system. The last weather system is Mediterranean from the south with few clouds and increasing the number of sunny days. This weather system influences the southern slopes of the Alps i.e. *Lienz*, *St.Andere*, *Kleinzicken*.

Fig. 9 shows the causal graph discovered by HGGM, TEN and Crack. SFGC was not able to detect any causal relation therefore we exclude its result. Considering the impact of the Atlantic weather system, one expects the influence on the neighbour sites of *Feuerkogel* and the sites in eastern Austria. The sites in southern slope cannot be influenced by this system since the Alps are located in between. Comparing HGGM to other algorithms, HGGM is successful to detect more influenced sites by finding the correct causal direction among *Linz*, *Salzburg*, *Retz*, *Wien* and *Eisenstadt*. However it detects an interaction between *Feuerkogel* and *Lienz* which is not likely due to the large mountain area between the sites. Regarding Crack, although the only causal relation discovered by this algorithm sounds reasonable, there are other stations, e.g. *Linz* and *Salzburg*, where it is plausible to consider a causal interaction among them. On the other hand, TEN discovers a dense causal graph among all 20 time series and *Feuerkogel* which is hard to interpret. Moreover considering the Atlantic weather system, there is no interpretation for the causal direction from *Retz* to *Feuerkogel* detected by TEN since its direction is exactly in the opposite.

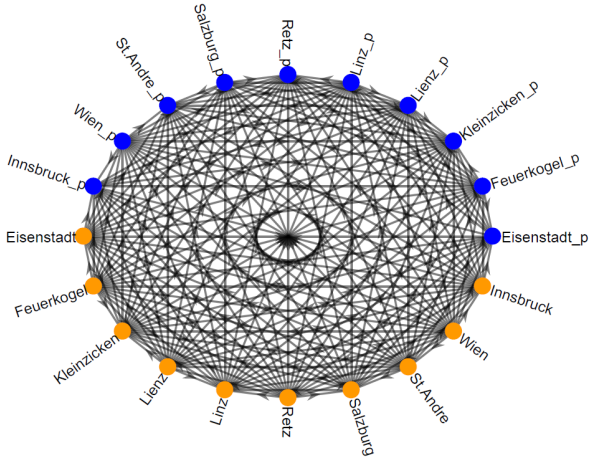


Fig. 8 Experiment on the Austrian data. Result of TEN algorithm.

6 Anomaly Detection of Heterogeneous Time series

As mentioned, discovery of causal relations among different processes leads to specify any evolution in time of regular instances. The regular pattern can be used to detect the deviated observations in anomaly detection. Among various types of anomalies (e.g. uni-variate anomalies etc.) we focus on dependency anomalies where an anomaly occurs due to changes of temporal dependencies. Detecting dependency anomalies is more challenging and complicated, but common in the real application. Given a heterogeneous data set, this task gets even more complex although such a scenario seems to

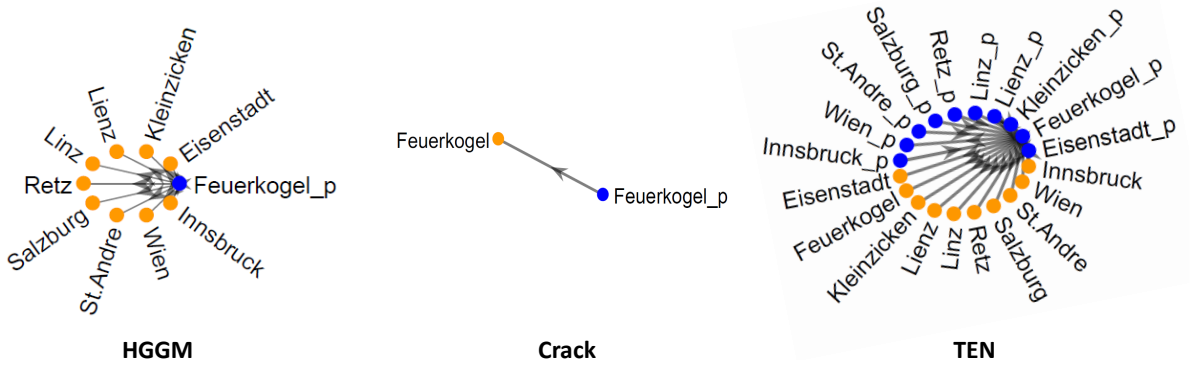


Fig. 9 Experiment on the Austrian climatological data. blue circles: amount of precipitation and orange circles: number of sunny days.

be realistic. In this section we introduce our anomaly detection algorithm designed for heterogeneous cases based on what we introduced in the previous sections. Firstly, we investigate all the temporal dependencies between or within the heterogeneous time series. Then, we utilize this useful information in order to detect dependency anomalies in a general case.

Before explaining how to use the temporal dependencies in order to find the anomalies, we formally define the task of anomaly detection in the following. Then, we introduce our anomaly detection framework designed for heterogeneous data sets.

6.1 Detecting Anomalies in Time Series

One of the approaches to detect anomalies in time series is to find the most accurate statistical model that captures the generation process of the normal (non-anomalous) data, then, investigate any deviation from this normal pattern. That is, we estimate the likelihood of a new observation based on the captured model and specify the data as an anomaly if the likelihood is below some thresholds. More precisely, for a time series $x^{1:n}$, let $x^{(tr)} = \{x^t | t = 1, \dots, T_{tr}\}$ and $x^{(te)} = \{x^t | t = T_{tr}, \dots, n\}$ denote the training data and test data, respectively. We assume the training data to be non-anomalous and we name the model corresponding to the training data the normal pattern. In the next step, we investigate the test data observations and specify any significant deviations to the normal pattern as anomalies. To the best of our knowledge, Qiu *et al.* employed Granger graphical models to detect dependency anomalies (AD-GGM) among time series for the first time [19]. However, their approach is restricted to only Gaussian time series where some specific assumptions about the distribution of time series is considered. Although our approach is inspired by [19], we incorporate our proposed Granger graphical model for heterogeneous time series (HGGM) and introduce a new anomaly detection algorithm (AD–HGGM) for mixed time series.

6.2 Heterogeneous Anomaly Detection Framework

As mentioned, we aim at detecting dependency anomalies in this paper where an anomaly occurs due to changes of temporal dependencies. That is, given p time series $x_1, \dots x_p$ of length n, we, firstly, investigate the test data observations utilizing the temporal causal relations among time series while no assumption about the distribution of time series is considered. Analogous to the previous section, let $X^{(tr)} = \{x_i^t | i = 1, \dots, p \ \ \& \ \ t = 1, \dots, T_{tr}\}$ and $X^{(te)} = \{x_i^t | i = 1, \dots, p \ \ \& \ \ t = T_{tr}, \dots, n\}$ denote the training data and the test data for p time series. Therefore, we compare the normal pattern captured from $X^{(tr)}$ to the pattern captured from $X^{(te)}$. Our proposed assumption-free anomaly detection framework for heterogeneous time series consists of three main building blocks:

- detecting the temporal causal relations,
- identifying an appropriate anomaly score,
- introducing an efficient approach to specify anomalies.

In the following we describe every block in detail.

- **Temporal Dependencies:** Most existing algorithms for Granger causal inference are based on statistical significance tests which is computationally expensive and sensitive to the number of observations. In this paper, also inspired by GGM algorithm, we employ our proposed Granger graphical model (HGGM) to learn temporal dependencies based on penalized regression. That is, we determine the temporal relations among a time series x_i and others by the regularized regression introduced in Equation 7.

Extensive experiments in Section 5 already confirmed effectiveness and efficiency of HGGM. Moreover, our approach is general compared to GGM in the sense that it is applicable for heterogeneous data sets.
In a normal case, when no anomalies occur, the temporal causal graph is the same for training and test data. This is the null hypothesis in dependency anomalies. That is, when learning the temporal dependencies in test data, we consider the null hypothesis, i.e. temporal dependencies in training data, as another constraint. More precisely, let $\hat{\beta}_i^{(tr)}$ be the coefficient vector solving the optimization problem defined in Equation 7 for training data $X^{(tr)}$. Analogously, let $\beta_i^{(te)}$ denote the coefficient vector w.r.t. test data $X^{(te)}$. Thus, when finding $\beta_i^{(te)}$ we incorporate the null hypothesis in the sense that the values of $\beta_i^{(te)}$ should be zero (or nonzero) when the corresponding values of $\beta_i^{(te)}$ are zero (or nonzero). This leads to the following regularized regression the test data:

$$\hat{\beta}_i^{(te)} = \arg\min_{\beta_i^{(te)}} \sum_{T=T_{tr}+d+1}^{n} \big[- x_i^{(te),T}(X_{T,d}^{(te),Lag}.\beta_i^{(te)}) + g_i^{-1}(X_{T,d}^{(te),Lag}.\beta_i^{(te)})\big] + \lambda.\sum_{j=1}^{p} w_j|\beta_j^{(te)} - \hat{\beta}_j^{(tr)}|. \tag{10}$$

Since $\hat{\beta}_j^{(tr)}$ is a constant in Equation 10, therefore the estimation of $\hat{\beta}_i^{(te)}$ is also consistent followed by Corollary 1. In the next step we aim at finding data points that significantly deviate from the normal pattern.

– **Information-theoretic Score:** When all the temporal dependencies are detected, the next step is to specify anomalies comparing the captured model w.r.t. training and test data. That is, we employ an anomaly score to measure any difference between two distributions. From the information-theoretic perspective, the most natural difference measure between every two distributions is the Kullback-Leibler (KL) divergence. This is a distance function of to which extent one probability distribution diverges from the second one. However, KL-divergence is not symmetric and hence not a distance metric. Therefore, we consider Jensen-Shannon (JS) divergence as an anomaly score in our framework. JS-divergence is symmetric, its square root is a metric and can be used as a distance function. These properties of JS-divergence help to save some computations.

Altogether, for a particular time series x_i Jensen-Shannon anomaly score (JSD) is defined by:

$$JSD(x_i) = \frac{1}{2}D(P^{(tr)}||M) + \frac{1}{2}D(M||P^{(te)}) \tag{11}$$

where $P^{(tr)}$ and $P^{(te)}$ are the underlying probability distribution function (PDF) in the reference and test data sets, respectively while it depends on the distribution of the time series x_i. Applying GLMs allow us to consider all the probabilities from the exponential family, e.g. Poisson, Gamma. As an example, if the underlying distribution is Poisson, the conditional probabilities used in JSD are computed considering $x_i|X_{T,d}^{Lag} \sim Pois(X_{T,d}^{Lag}.\hat{\beta}_i)$ where $\hat{\beta}_i$ is the coefficient vector achieved by Equation 7. Moreover, $D(.)$ denotes the KL measure and is defined as follows:

$$D(P^{(tr)}||M) = \sum P^{(tr)}(x_i|X_{T,d}^{Lag})\frac{P^{(tr)}(x_i|X_{T,d}^{Lag})}{P^{(te)}(x_i|X_{T,d}^{Lag})} \tag{12}$$

where $M = \frac{1}{2}(P^{(tr)} + P^{(te)})$.

– **Anomaly Detection:** So far we have elaborated how to specify temporal dependencies in a heterogeneous data to capture the model associated to the data. Moreover, we introduced an anomaly score to measure the deviation of a model to a normal pattern. Now, the question is how to mark anomalies. In order to specify dependency anomalies we need a threshold defined based on the non-anomalous part of the data, i.e. training data. One could consider the entire training data to capture an anomaly threshold. However, inspired by AD-GGM algorithm [19], we slide a window over training data and find an anomaly threshold w.r.t. every time window in order to give more insights about the exact position of the anomaly. That is, for every window we compute the anomaly score introduced in the previous section and approximate the distribution of the anomaly scores for a non-anomalous data. Employing a significance level α, the $\alpha-quantile$ of this distribution is considered as threshold cutoff (refer to Section B in [19]).

6.3 Algorithm

Algorithm 2 summarizes different steps of the proposed anomaly detection procedure based on HGGM algorithm for heterogeneous data sets (AD–HGGM). As already explained, we aim at finding a normal temporal

pattern based on the training data. Then, we slide a window over the test data and specify anomalous windows comparing the corresponding anomaly score for the window to an anomaly threshold.

Before starting the algorithm, we need to specify an anomaly threshold w.r.t. the non-anomalous data, i.e. training data, for every time series $x_i, i = 1, \ldots, p$. Procedure *AnomalyThreshold(.)*, summarized in Algorithm 3, shows different steps of this process. $\beta^{(tr)}$ denotes the coefficient vector of the corresponding VAR model for the entire training data when learning the temporal causal dependencies by HGGM. Then, we slide a window over the training data and learn the temporal dependencies for every window. Finally, we calculate the $JSD(.)$ for every window and find $\alpha - quantile$ of the most fitting distribution w.r.t. anomaly scores.

Now, having an anomaly threshold w.r.t. every time series, we proceed with the same procedure (as explained in Algorithm 3) for the test data. In order to detect the anomalies more precisely, we slide a window over the test data to discover not only if any anomaly occurs but also more accurately in which time window it happens. That is, We slide a window w of size WS over the test data and learn the temporal dependencies (Equation 10). Then, we compare the corresponding anomaly score (JSD_i^w) to the threshold w.r.t. time series x_i, i.e. thr_i. If the anomaly score is higher than the threshold we mark the window as an anomalous window.

Algorithm 2 Heterogeneous Anomaly Detection

AD–HGGM($x_1, \ldots, x_p; WS; \alpha; \lambda_{max}$)

// $x_1, \ldots, x_p$:= time series with a distribution from the exponential family
// WS := window size
// α := significance level
// λ_{max} := maximum λ

$X^{(tr)} = \{x_i^t | i = 1, \ldots, p \ \& \ t = 1, \ldots, T_{tr}\}$ // training data
$X^{(te)} = \{x_i^t | i = 1, \ldots, p \ \& \ t = T_{tr}, \ldots, n\}$ // test data

// find the anomaly threshold for every time series
$thr = AnomalyThrshold(X^{(tr)}, WS, \alpha)$;

for all x_i **do**

  // Learn temporal causal graph for x_i
  ${\beta_i}^{(tr)} = HGGM(X^{(tr)}, \lambda_{max})$;

  // Slide a window w with size WS over the test data
  for all w **do**
    ${\beta_i}^{(te),w} = HGGM(X^{(te),w}, \lambda_{max})$;
    $JSD_i^w = JSD(x_i^{(w)}, {\beta_i}^{(tr)}, {\beta_i}^{(te),w})$
    if $JSD_i^w > thr_i$ **then**
      mark w as an anomalous window;
    end if
  end for

end for

Algorithm 3 Anomaly Thresholds

AnomalyThreshold($X^{(tr)}$, WS, α)
// find anomaly threshold for every time series
thr := a vector of anomaly thresholds w.r.t. time series

// Learn normal temporal causal graph
$\beta^{(tr)} = HGGM(X^{(tr)}, \lambda_{max})$;
for all x_i **do**
  // Slide a window w with size WS over the training data
  for all w **do**
    ${\beta_i}^{(tr),w} = HGGM(X^{(tr),w}, \lambda_{max})$;
    $JSD(x_i^{(w)}, {\beta_i}^{(tr)}, {\beta_i}^{(tr),w})$
  end for
  //fit a distribution to anomaly scores then consider $\alpha -$ *quantile* of it.
  thr_i = fitDistribution($JSD(x_i), \alpha$);
end for
return thr

6.4 Experiments

In this section we conduct various experiments assessing the performance of our proposed heterogeneous anomaly detection algorithm (AD–HGGM) in terms of *F-measure*. In every experiment, we compare our results to AD-GGM [19] in order to investigate the impact of applying a homogeneous algorithm with specific assumptions on a heterogeneous data.

Our expectation is that AD–HGGM should perform better on heterogeneous data sets and especially non-Gaussian time series due to the employed general Granger graphical model (HGGM). Moreover, the introduced anomaly score (JSD) relaxes Gaussian assumptions about the time series when computing the anomaly scores for every time window. First, we elaborate the data generation process. Then, the performance of AD–HGGM is assessed on various homogeneous and heterogeneous synthetic data sets. Afterwards, we will conduct different experiments to see how the performance changes when changing the length of time series, strength of causal relations (dependency), number of dependencies and number of anomalous time series. Both, AD–HGGM and AD-GGM, are implemented in MATLAB and for AD-GGM we use the publicly available implementations and recommended parameter settings. The source code and data sets are publicly available in the following repository: https://tinyurl.com/tnxw7fc

– **Synthetic Data Generation:** In order to investigate dependency anomalies in time series, we generate the corresponding training and test data from different dependency structures, i.e. with different VAR models. Figure 10 shows an example when two dependency structures along with two random

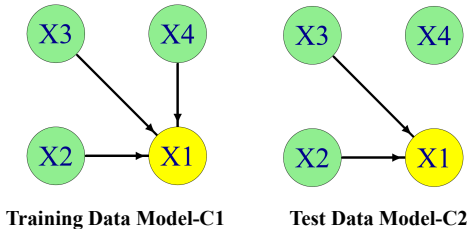


Fig. 10 Dependency structure of generated data.

coefficient matrices, C_1 and C_2, are used to generate a synthetic data set. Given 3 random time series X_2, X_3 and X_4, we generate training and test data w.r.t. X_1 considering a random lag variable and two coefficient matrices C_1 and C_2. $X_2, \ldots, X_4$ could have any distribution from the exponential family. In this example, all the time series consist of 300 observations. The first 200 values of X_1 (training data) are then generated with C_1 and the last 100 (test data) are generated with C_2. Therefore, an anomaly occurs at time 201 meaning that all the anomaly scores for the last 100 values should ideally be above the anomaly score threshold calculated from the training data.

– **Accuracy:** AD–HGGM is designed to deal with homogeneous and heterogeneous data sets consisting of time series with a distribution from the exponential family. Analogous to the previous experiments on HGGM algorithm (Section 5), we focus on 4 distributions: Gaussian, Poisson, Bernoulli and Gamma. We start by trying out all different homogeneous data sets, generated as explained in the previous section, with 4 features and the dependency structure shows in Figure 10. Both algorithms were run on multiple data sets and we took the average F-measure for all of them. As explicitly evident from Figure 11, AD–HGGM outperforms AD-GGM in all of the homogeneous cases with a wide margin. These results, specially results on Gaussian data set, confirm the effectiveness of adaptive Lasso (whose consistency is proven) compared to Lasso approach. Moreover, employing GLM framework instead of assuming a default Gaussian distribution, pays off in various non-Gaussian data sets, as it is observed in Figure 11.
Next, we consider more realistic data sets where a mixture of time series from various distributions is given. We tried out data sets with the following combinations of distributions: Binomial-Gamma, Binomial-Poisson, Gamma-Poisson, Poisson-Gaussian, Gaussian-Gamma, Gaussian-Binomial. As expected (also confirmed by the experiments, Figure 11), AD-GGM is not able to deal with the heterogeneity of data sets leading to poor results compared to homogeneous cases. Not only homogeneity, but also non-Gaussianity of a data set negatively influences the performance of AD-GGM, as, for instance, it results worse on "Bernoulli-Poisson" in comparison to "Gaussian-Gamma" or "Gaussian-Bernoulli" (see Figure 11). On the other side, AD–HGGM effectively handles the heterogeneity of data sets, regardless of the distribution of time series, due to our general assumption-free framework utilizing GMLs.

In the next experiments, we focus on different aspects of data that can potentially be impactful, e.g. the length of the time series or number of anomalous features. The data sets are generated with the same procedure as before. However, we modified the length of time series or the strength of dependencies in some of the experiments. But generally there were two coefficient matrices defined that simulate the temporal anomaly changes. In order to be consistent in the following experiments, the distribution of the data sets is the combination of Bernoulli and Gamma, since both algorithms seem to be well comparable in performance on this combination (AD-GGM results the best on this data in previous experiment, see Figure 11). Further details about the data sets will be discussed in each subsection.

– **Dependency:** We refer to the coefficients in a vector autoregressive model as dependencies. Moreover, coefficients in a VAR model show the strength of temporal causal relations in that model. In this experiment, we aim at assessing the algorithms in the sense that different temporal dependencies with various strength are given. Therefore, we evaluate AD–HGGM compared to AD-GGm ranging the dependency between 0.2 to 1.0. In this case, a more stable trend shows the performance and the ability of the algorithm to deal with even weak temporal dependencies. The experiment is conducted on 4 time series with the length 300 from mixed distribution data sets, with only the first feature exhibiting anomalous behavior. As expected, Figure 12a illustrates the stability of AD–HGGM compared to AD-GGM even for weak causal relations.

– **Number of anomalous features:** Is the algorithm able to detect the anomalies equally well as only looking for one anomaly, if there are more than one anomalous time series? To assess the algorithm in this respect, we continuously increase the number of features as well as the number of anomalous fea-

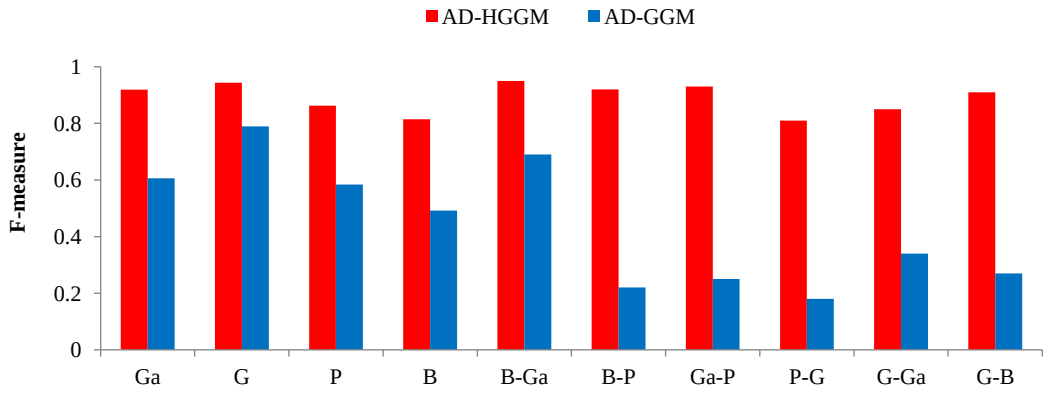


Fig. 11 Accuracy investigation on synthetic homogeneous and heterogeneous data sets. Ga: Gamma, G: Gaussian, B: Bernoulli, P: Poisson.

tures ranging from 2 to 10 in 2 increments. Figure 12b shows the result of AD–HGGM compared to AD-GGM in this experiment. As it is evident, number of anomalous features has no impact on the performance of AD–HGGM confirming the efficiency of our proposed algorithm even in more complex data sets where many anomalous features are interacting.

– **Number of dependencies:** Does the sparseness of a temporal causal graph play any roles when detecting the dependency anomalies? In this experiment, we address the above question while increasing the number of causal relations ranging from 1 to 5. Here, the generated data sets consist of a mixture of 6 Bernoulli-Gamma distributed features of the length 300 with a random dependency structure. Regardless of sparseness of the causal graphs, AD–HGGM outperforms AD-GGM almost always showing a stable trend (see Figure 12c). Thus, sparsity does not play any roles when applying our proposed algorithm. This result confirms the effectiveness of AD–HGGM applying to the data sets with sparse as well as complex dependency structures.

– **Feature length:** In this experiment, we evaluate the algorithms when they deal with different lengths of time series ranging from 200 to 2000. The dependency structure outlined before is the same (a mixture of 4 time series from Bernoulli-Gamma combination of distribution), except the fact that the length of training and test data changes with the length of the time series. Analogous to the previous experiments, where the training data was 200 and test data 100 time points long, we choose to use 60% of the data for training and the rest as test data, on which the algorithm performance is evaluated. As evident in Figure 12d, the performance of AD–HGGM is robust against changing the length of time series. This result confirms the effectiveness of proposed algorithm regardless of the length of the provided data. In contrast, AD-GGM does not show any meaningful trend which is difficult to interpret although the authors claim that with more data AD-GGM is more effective.

7 Conclusions and future work

In this paper we introduced HGGM, a graphical Granger model for discovery of causal relations among a number of heterogeneous processes. Profiting of a GLM framework our approach is general for time series having distributions from exponential family. Moreover to ensure the consistency of HGGM we employ adaptive Lasso with a proven consistency. We investigated the performance of HGGM in terms of effectiveness and efficiency comparing to state-of-the-art methods. Extensive experiments on synthetic and real data sets demonstrates the advantages of HGGM. As one of the interesting applications of HGGM, we utilized it to detect anomalies among heterogeneous time series. Thus, we introduced a general anomaly detection framework to discover dependency anomalies among time series in heterogeneous data sets (AD–HGGM).

8 Compliance with Ethical Standards

– **Funding:** This study was not funded.
– **Conflict of Interest**: None of the authors has received research grants from any company for this article. Sahar Behzadi, Niklas Preschern and Kateřina Hlaváčková-Schindler declare that they have no con-

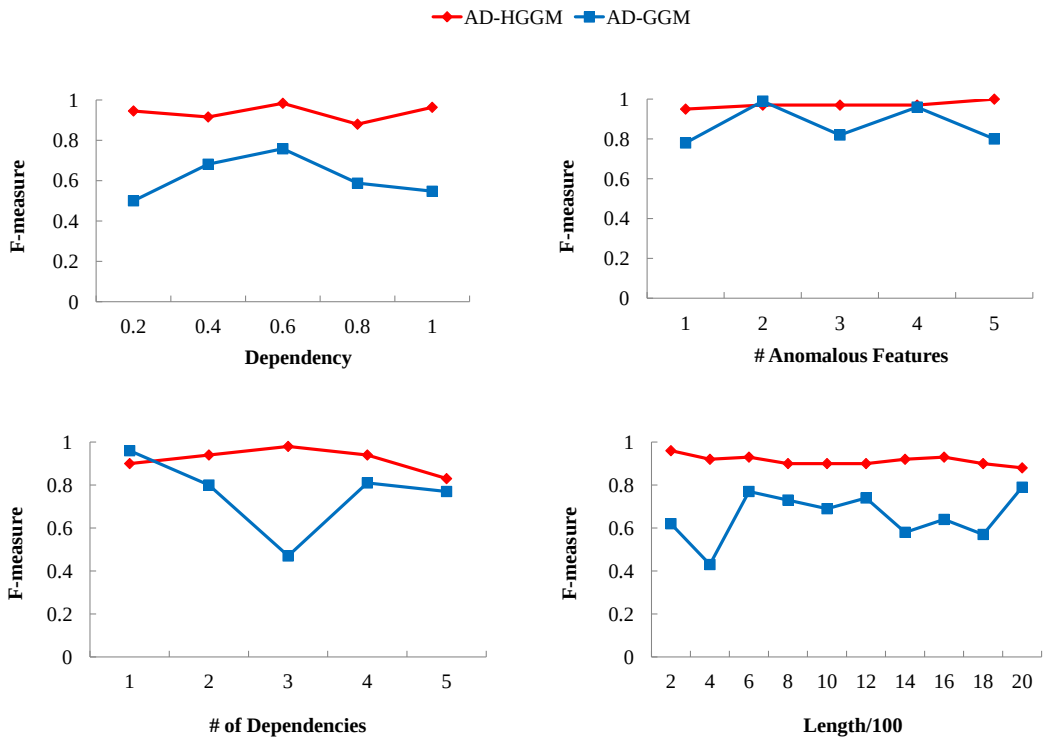


Fig. 12 Investigating various characteristics of AD–HGGM

flict of interest. Prof. Claudia Plant is an Editor in KAIS journal.

– **Ethical approval:** This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Arnold, A., Liu, Y., Abe, N.: Temporal causal modelling with graphical Granger methods. In: KDD (2007)
2. Bacsó, N.: Das Klima des Donauraumes. Geoforum (1971)
3. Bahadori, M.T., Liu, Y.: Granger causality analysis in irregular time series. In: SDM (2012)
4. Behzadi, S., Hlaváčková-Schindler, K., Plant, C.: Granger causality for heterogeneous processes. In: The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) 2019 (2019)
5. Budhathoki, K., Vreeken, J.: Causal inference by compression. In: ICDM (2016)
6. Budhathoki, K., Vreeken, J.: Mdl for causal inference on discrete data. In: ICDM (2017)
7. Budhathoki, K., Vreeken, J.: Causal inference on event sequences. In: SDM (2018)
8. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Comput. Surv. **41**(3), 15:1–15:58 (2009). DOI 10.1145/1541880.1541882
9. Cheng, D., Bahadori, M.T., Liu, Y.: Fblg: A simple and effective approach for temporal dependence discovery from time series data. In: KDD (2014)
10. Granger, C.W.: Investigating causal relations by econometric models and cross-spectral methods. Econometrica pp. 424–438 (1969)
11. Kim, S., Putrino, D., Ghosh, S., Brown, E.: A Granger causality measure for point process models of ensemble neural spiking activity. PLOS Comp. Biology pp. 1–13 (2011)
12. Laptev, N., Amizadeh, S., Flint, I.: Generic and scalable framework for automated time-series anomaly detection. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, pp. 1939–1947. ACM, New York, NY, USA (2015). DOI 10.1145/2783258.2788611
13. van der Loo, M.P.: Detection of outliers with the extremevalues package. In: useR2010 (2010)
14. Marx, A., Vreeken, J.: Causal inference on multivariate mixed-type data by minimum description length. ECMLPKDD (2018)
15. McIlhagga, W.: penalized: A matlab toolbox for fitting generalized linear models with penalties. Journal of Statistical Software, Articles (2016)
16. Mooij, J.M., Peters, J., Janzing, D., Zscheischler, J., Schölkopf, B.: Distinguishing cause from effect using observational data: Methods and benchmarks. Journal of Machine Learning Research (2016)
17. Nelder, J.A., Baker, R.J.: Generalized linear models. Encyclopedia of statistical sciences (1972)
18. Peters, J., Janzing, D., Schölkopf, B.: Causal inference on discrete data using additive noise models. IEEE Trans. patt. an. and mach. intelligence p. 2436—2450 (2011)
19. Qiu, H., Liu, Y., Subrahmanya, N.A., Li, W.: Granger causality for time-series anomaly detection. In: ICDM (2012)

20. Rogge-Solti, A., Kasneci, G.: Temporal Anomaly Detection in Business Processes, pp. 234–249. Springer International Publishing, Cham (2014)
21. Schreiber, T.: Measuring information transfer. Physical review letters **85**(2), 461 (2000)
22. Shimizu, S., Hoyer, P.O., Hyvärinen, A., Kerminen, A.: A linear non-gaussian acyclic model for causal discovery. Journal of Machine Learning Research **7**(Oct), 2003–2030 (2006)
23. Shojaie, A., Michailidis, G.: Discovering graphical Granger causality using the truncating lasso penalty. Bioinformatics (2010)
24. Tibshirani, R.: Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society. Series B (Methodological) pp. 267–288 (1996)
25. Vallis, O., Hochenbaum, J., Kejariwal, A.: A novel technique for long-term anomaly detection in the cloud. In: 6th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 14). USENIX Association, Philadelphia, PA (2014)
26. Zou, H.: The adaptive lasso and its oracle property. Journal of the Am. Stat. Assoc. pp. 1418–1429 (2008)

CHAPTER 11

Paper H & Paper I: Information–theoretic Granger Causal Inference on Heterogeneous Data: problem specification and algorithm

This chapter comprises two publications concerning information-theoretic causal inference on single-type heterogeneous data (Paper H [16]) and its extended journal version (Paper I [15]). Paper I is included in this chapter.

Authors Contributions:

- **Sahar Behzadi.** Cooperation on the main idea and developing the algorithm as well as writing the paper; Implementation; Conducting experiments.
- **Benjamin Schelling.** Cooperation on interpreting results of the experiments as well as writing the paper.
- **Claudia Plant.** Supervision during development and evaluation of the algorithm as well as writing the paper.

Noname manuscript No.
(will be inserted by the editor)

Information–theoretic Granger Causal Inference on Heterogeneous Data: problem specification and algorithm

Sahar Behzadi · Benjamin Schelling · Claudia Plant

Received: date / Accepted: date

Abstract Granger causality for time series states that a cause improves the predictability of its effect. That is, given two time series x and y, we are interested in detecting the causal relations among them considering the previous observations of both time series. Although, most of the algorithms are designed for causal inference among homogeneous processes where only time series from a specific distribution (mostly Gaussian) are given, many applications generate a mixture of various time series from different distributions. We utilize Generalized Linear Models (GLM) to propose a general information–theoretic framework for causal inference on heterogeneous data sets. We regard the challenge of causality detection as a data compression problem employing the Minimum Description Length (MDL) principle. By balancing the goodness–of–fit and the model complexity we automatically find the causal relations. Extensive experiments on synthetic and real–world data sets confirm the advantages of our algorithm ITGH (for **I**nformation–**T**heoretic **G**ranger causal inference on **H**eterogeneous data) compared to other algorithms.

Keywords Heterogeneous data · Information theory · Causal inference · Granger causality

S. Behzadi
Faculty of Computer Science, Data Mining, University of Vienna
Vienna, Austria
E-mail: sahar.behzadi@univie.ac.at

B. Schelling
Faculty of Computer Science, Data Mining, University of Vienna
Vienna, Austria
E-mail: benjamin.schelling@univie.ac.at

C. Plant
Faculty of Computer Science, Data Mining, University of Vienna and ds:Univie
Vienna, Austria
E-mail: claudia.plant@univie.ac.at

1 Declarations

- **Availability of data and material:** We conduct various experiments on publicly available homogeneous and heterogeneous real–world data sets where a valid ground truth is provided. Table 3 summarizes the characteristics of the data sets. Moreover, a climatological data set without a provided ground truth is investigated which is publicly available in [11]. For convenience, here we gathered all the data sets: `https://tinyurl.com/yar5yuoq`.

- **Code availability:** ITGH is implemented in MATLAB and the source code is publicly available at: `https://tinyurl.com/yar5yuoq`

2 Introduction

Discovery of causal networks from observational data, where no certain information about their distribution is provided, is a fundamental problem with many applications in science. The regular patterns, found by investigating the corresponding causal graph, can be used to detect the deviated observations or outliers [17]. Among several notions of causality, Granger causality [9] is a popular method for causal inference in time series due to its computational simplicity. It states that a cause improves the predictability of its effect in the

future. That is, given two time series x and y, considering the previous observations of y together with x improves the predictability of x if y causes x. There are various algorithms in this area depending on how we measure the predictability. Usually, any improvement in the predictability is measured in terms of variance of the prediction errors (known as Granger test, shortly GT).

In this paper we establish our method based on an information–theoretic measurement of the predictability. That is, we regard the challenge of causal inference as a data compression problem. In other words, employing the *Minimum Description Length* (MDL) principle, y causes x if considering the past of y together with x decreases the number of bits required to encode x. More deviation in the compression cost reveals stronger causal dependency among two time series. Unlike the other information–theoretic approaches (e.g. entropy–based algorithms [20]), we incorporate complexity of the models with the MDL–principle. Thus, it leads to a natural trade–off among model complexity and goodness–of–fit while avoiding over–fitting.

Although Granger causality is well–studied, most of the algorithms are designed for homogeneous data sets where time series from a specific distribution are provided. Recently, Budhathoki *et al.* proposed a MDL–based algorithm designed for causal inference on binary time series [7]. Additive Noise Models (ANMs) have been proposed for either continuous [16] or discrete [15] time series. Graphical Granger approaches, which are popular due to their efficiency, mostly consider additive causal relations with a certain Gaussian assumption, e.g. TCML [1] or [2].

Despite the efficiency of homogeneous algorithms, many applications generate heterogeneous data, i.e. a mixture of various time series from different distributions. In climatology, for instance, the measurement of number of wet days does not necessarily follow the same distribution as the amount of precipitation. Moreover, transforming a time series to another time series with a specific distribution leads to inaccuracy. On the other side, applying an algorithm designed for homogeneous data sets on heterogeneous data does not guarantee a high performance. To elaborate, we generated a heterogeneous data set consisting of 4 Poisson (blue circles in Figure 1) and a Gamma (orange circle) distributed time series and applied one of the representatives for any category designed for Granger causal inference on homogeneous data sets. As it is explicitly clear in Figure 1, none of the well–known algorithms perform efficiently on this data set in terms of *F–measure*. GT, for instance assumes a Gaussian distribution and hence a linear relation among time series which obviously leads to inefficiency. On the other hand, CUTE needs to binarise time series as it is designed for event sequences where Bernoulli distributed time series are assumed. It is already well–understood that discretization and specially binarising the data decreases the accuracy since the distribution of the time series is not any more the same.

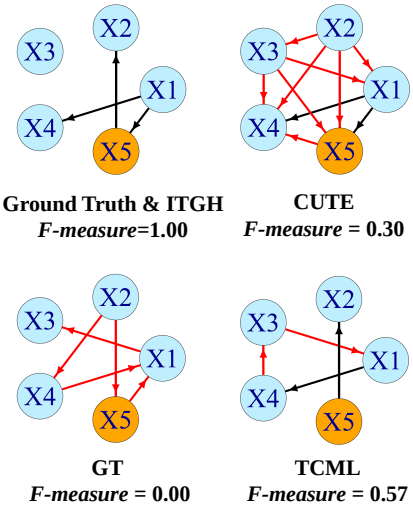


Fig. 1 Synthetic heterogeneous example. Results of applying existing Granger causal inference algorithms designed for homogeneous data sets on heterogeneous data. Red edges show the wrongly detected causal relations and black edges show the correct causal directions.

Thus, integrating processes of various distributions without any transformation or certain assumptions sounds crucial. In this paper, we utilize *Generalized Linear Models* (GLMs) to extend the notion of Granger causality and introduce an integrative information–theoretic framework for causal inference on heterogeneous data regardless of the distributions. GLMs allow us to generalize simple autoregressive models to the case where several processes of different distributions from the exponential family are non–linearly related. Moreover, unlike many other algorithms, we aim at detecting causal networks. To the best of our knowledge, almost all of the existing algorithms are designed based on a pairwise testing approach. This approach is inefficient in causal network discovery when dealing with large causal networks. To avoid this issue, we propose our MDL–based greedy algorithm (ITGH) to detect heterogeneous Granger causal relations in a GLM framework. Our approach consists of the following contributions:

– **Effectiveness:** We introduce a MDL–based indicator for detecting Granger causal relations. Unlike other information–theoretic approaches, we ensure the effectiveness of our MDL–based algorithm by

balancing goodness–of–fit and model complexity;

- **Heterogeneity:** Applying the GLM methodology, we propose our heterogeneous MDL–based algorithm to discover the causal interactions among a wide variety of time series from the exponential family;

- **Scalability:** Due to the greedy approach, we might not find the overall optimal solution, but it makes ITGH scalable and convenient to be used in practice. Moreover, our extensive experiments on synthetic and real data sets confirm the efficiency of the proposed algorithm compared to others;

- **Comprehensiveness:** Our approach is comprehensive in the sense that we avoid any assumptions about the distribution of data by applying an information –theoretic approach.

The paper is organized as follows: First, we present the related work in Section 3. In Section 4, we elaborate the theoretical aspects of ITGH together with the required background. In Section 5, we introduce our greedy algorithm ITGH. Extensive experiments on synthetic and real-world data sets are demonstrated in Section 6.

3 Related Work

Granger causality is a well–known causal notion among time series due to its simplicity, robustness and computational efficiency [9]. It states that a cause (y) efficiently improves the predictability of its effect (x).

There are various approaches to infer the causality depending on how to measure the predictability. Typically, any improvement in the predictability is measured in terms of variance of the error by a hypothesis testing approach [12,18]. Moreover, graphical Granger methods are designed based on a penalized estimation of *vector autoregressive* (VAR) models [1,22]. The intention in this approach is that, if y causes x it has non-zero coefficient in the VAR model corresponding to x. First, Arnold *et al.* [1] proposed a Lasso penalized estimation for VAR models (TCML) to obtain a sparse and robust estimator and hence the causal relations. As an extension, Bahadori and Liu [3] proposed a semi–parametric algorithm for non–Gaussian time series based on the copula approach to retain the scalability of linear VAR. Recently, authors in [5] employed adaptive Lasso to generalize this approach to the heterogeneous cases (HGGM).

As another category, probabilistic approaches interpret the predictability as the improvement in the likelihood. Several methods in this group are distinguished based on the way how they incorporate the probability. Among them, Kim and Brown [10] introduced a probabilistic framework (SFGC) for Granger causal inference on heterogeneous data sets by a pairwise testing of the maximum likelihood ratio. In this framework assessing causal relations between multiple time series is accomplished by the false discovery rate (FDR). The statistical power of the FDR based methods rapidly decreases with increasing the number of hypotheses and these methods are computationally expensive.

As another approach, information–theoretic methods detect the causal direction by introducing a causal indicator. Among them, transfer entropy, shortly TEN, is designed based on Shannon's entropy [20]. In this approach, it is more likely that the causal direction with the lower entropy corresponds to the true causal relation. Given a lag variable, TEN detects both linear and non–linear causal relations. However, due to pairwise testing and its dependency on the lag variable, the computational complexity of this algorithm is exponential in the lag parameter. On the other hand, compression–based algorithms apply the Kolmogorov complexity and define a causal indicator based on the MDL–principle. Unlike the entropy–based approach, we incorporate the complexity of the models in the MDL–principle, as well. Thus, it leads to a natural trade–off among model complexity and goodness–of–fit while avoiding over–fitting. Then, MDL–based approach is more efficient compared to the entropy–based approach. Recently, Budhathoki *et al.* [7] proposed a MDL–based algorithm (CUTE) to infer the Granger causality among event sequences. However, this algorithm is designed only for homogeneous data sets and deals with binary time series. Moreover, CUTE detects the causal relations in an exhaustive pairwise manner which ends with inaccurate networks while detecting Granger causal graphs. To the best of our knowledge, ITGH is the only algorithm in this approach which is designed for discrete and continuous time series and supports the heterogeneity of data sets.

Several other approaches have been proposed for identification of Granger causality in non–linear systems e.g. kernelized regression [13], generalized linear autoregressive models [23], [11]. However these methods are not efficient in high dimensions and do not support the heterogeneity.

4 Theory

How to detect the Granger causal direction among any two time series? How to extend this concept to a general

Symbol	*Description*
$\mathcal{I}(t)$	all the information accumulated since time t
$\mathcal{I}_{\neg y}(t)$	all the information apart from the specified time series y up to time t
${x_i}^t$	time series x_i at time t
X^t	concatenated vector of $x_1, ..., x_p$ at time t
n	length of time series
d	lag variable
g_i	link function w.r.t. x_i
DL	description length
$\mathcal{C}_i$	set of all causal time series w.r.t. x_i
$M_{\mathcal{C}_i}$	prediction model w.r.t. x_i
${e_i}^t$	estimation error $(\hat{x_i}^t - {x_i}^t)$

Table 1 Common symbols used in the paper.

heterogeneous case? Could an information–theoretic approach lead to causal inference? These are fundamental questions we address in this section while providing the required background, simultaneously. Table 1 summarizes the notations used commonly in this paper.

4.1 Granger Causality

Granger causality, introduced by Granger in the area of economics [9], is a well–known notion for causal inference among time series. Granger causality captures the temporal causal relations among time series although it is not meant to be always equivalent to the true causality since the question of "true causality" is deeply philosophical. This notion of causality is defined based on two principles [8]:

- The cause happens prior to its effect;
- The cause has unique information about the future values of its effect.

The first assumption is intuitively acceptable since the past influences the future, not other way around. On the other hand, the second assumption sounds plausible as well in the sense that without considering the cause no information about the effect is available. Now, let $x = \{x^t | t = 1, \ldots, n\}$ and $y = \{y^t | t = 1, \ldots, n\}$ denote two stationary time series x and y up to time n, respectively. Moreover, let $\mathcal{I}(t)$ be all the information accumulated since time t and $\mathcal{I}_{\neg y}(t)$ denote all the information apart from the specified time series y up to time t. Now considering two above assumptions, Granger proposed the following definition for a causal effect [9]:

Definition 1 Granger Causality: Given two time series x and y, y Granger–causes x if including previous values of y along with x improves the predictability of x, i.e.

$$\mathcal{P}(x^t | \mathcal{I}_{\neg y}(t-1)) < \mathcal{P}(x^t | \mathcal{I}(t-1))$$

where $\mathcal{P}$ denotes the predictability.

In another point of view, let Model 1 denote the *autoregressive* (AR) model of order d (the lag) corresponding to time series x and Model 2 denote the *vector autoregressive* (VAR) model w.r.t. x including the lagged observations of x and y.

$$x^t = \gamma_{t-d} \cdot x^{t-d} + ... + \gamma_{t-1} \cdot x^{t-1} + \epsilon^t \qquad \text{(Model 1)}$$

$$\begin{aligned} x^t &= \alpha_{t-d} \cdot x^{t-d} + ... + \alpha_{t-1} \cdot x^{t-1} \\ &+ \beta_{t-d} \cdot y^{t-d} + ... + \beta_{t-1} \cdot y^{t-1} + \epsilon^t \end{aligned} \qquad \text{(Model 2)}$$

Thus, y Granger–causes x if the second model improves the predictability of x.

The concept of Granger causality is extendable to more than two time series. Let $x_1, x_2, ..., x_p$ be p time series where $\forall i \in \{1, ..., p\}, x_i = \{{x^t}_i | t = 1, ..., n\}$. The VAR model of order d w.r.t. all the time series is defined as Model 3 in the following:

$$X^t = X^{t-d} \cdot B_{t-d} + ... + X^{t-1} \cdot B_{t-1} + \epsilon^t \qquad \text{(Model 3)}$$

where $X^t = ({x_1}^t, ..., {x_p}^t)$ is the concatenated vector of all time series at time point t. In this model B_t is a $p \times p$ matrix of the regression coefficients where the i-th row corresponds to the coefficients w.r.t. x_i at time t. Essentially, the matrix formulation is an abstract form to illustrate the temporal dependencies among all the time series.

Basic definition of the Granger causality has certain assumptions about the distribution of time series. More precisely, the processes are assumed to be Gaussian distributed time series in Model 1,2 and 3 and hence a linear model is considered overall. Moreover, in a linear model the error term (ϵ^t) is an additive Gaussian white noise with mean 0 and variance 1. However, these assumptions are not necessarily true in most of the applications. Thus, it is crucial to generalize the linear models to the non–linear cases in the sense that we include time series from various distributions and avoid any information loss resulted by a simple conversion.

4.2 General Causal Framework

As already discussed (Section 2), avoiding the heterogeneity of the data as well as assuming a specific (mostly Gaussian) distribution leads to inaccuracy. In this regard, illustrated results in Figure 1 are proper examples to elaborate the issue. Therefore, in order to avoid any information loss, we extend the Granger causality to a general GLM framework where a wide variety of distributions are included and no transformation is required.

GLM, introduced by Nelder and Baker in [14], is a natural extension of the linear regression to the case where the time series can have any distribution from the exponential family. Therefore, the response variable is not a simple linear combination of covariates but its mean value is related to the covariates by a *link function*. Corresponding to every distribution, there is an appropriate canonical link function (e.g. $g = log(.)$ for Poisson and $g = \frac{1}{(.)}$ for Gamma distribution) [14]. Table 2 summarizes well–known distributions from exponential family providing the appropriate canonical link function corresponding to each distribution. Thus, we generalize the models introduced in Section 4.1 as follows (Model 1 $\rightarrow$ Model 4 and Model 2 $\rightarrow$ Model 5):

$$E(x^t|x) = g(\gamma_{t-d} \cdot x^{t-d} + ... + \gamma_{t-1} \cdot x^{t-1}) + \epsilon^t \quad \text{(Model 4)}$$

$$E(x^t|x,y) = g(\alpha_{t-d} \cdot x^{t-d} + ... + \alpha_{t-1} \cdot x^{t-1} + \beta_{t-d} \cdot y^{t-d} + ... + \beta_{t-1} \cdot y^{t-1}) + \epsilon^t \quad \text{(Model 5)}$$

where g is the appropriate link function w.r.t. the distribution of time series x. GLM relaxes the Gaussianity assumptions about the involved time series and the error term. Therefore, ϵ^t does not necessarily follow a standard Gaussian distribution and it can have any distribution from the exponential family leading to more accurate models. In the following we denote Model 4 and Model 5 as M_x and M_{xy}, respectively. Thus, we extend the concept of Granger causality to heterogeneous cases by utilizing the advantages of a GLM framework. That is, the time series y Granger–causes x if M_{xy} results in an improvement in the predictability of x compared to M_x. Next, we propose an information–theoretic approach to measure the improvement in the predictability.

Distribution	*Link function*
Gaussian	$\mu = X.\beta$
Exponential/Gamma	$\mu = \frac{1}{X.\beta}$
Inverse Gaussian	$\mu = \frac{1}{X.\beta^2}$
Poisson/Countable	$\mu = exp(X.\beta)$
Bernoulli/Bi(Multi)nomial	$\mu = \frac{exp(X.\beta)}{1+exp(X.\beta)}$

Table 2 Common link functions for various distributions where X is the covariates matrix, μ is the mean and β is the coefficient matrix.

4.3 Information–theoretic measuring of Causal Dependencies

How to measure the predictability? How to employ information theory to infer causal relations? In this paper, we regard measuring the predictability to a compression problem. That is, we employ the description length of time series in the sense that the more predictable a time series is the less number of bits is required to compress and describe it. We focus on MDL [4] and introduce an information–theoretic indicator which reveals the causal dependencies among time series. In the following we introduce MDL–principle and elaborate how we utilize it for causal inference.

4.3.1 MDL–Principle

Essentially, MDL [4] is a well–known model selection approach to evaluate various models and find the most accurate one considering the minimum description length criteria. MDL–principle regards the model selection challenge to a data compression problem in the sense that more accurate models lead to less compression cost. Let $\mathcal{M}$ denote a set of various candidate models representing your data. Following the two–part MDL [4], the best fitting model $M \in \mathcal{M}$ is the one which minimizes

$$DL(D, M) = DL(D|M) + DL(M) \quad (6)$$

where $DL(D|M)$ concerns the description length of the data set D encoded by means of the model M and $DL(M)$ represents the model complexity, i.e. cost of encoding the model itself. In MDL–principle, we incorporate the model complexity to avoid any over–fitting caused by complicated models. Therefore, we encode not only the data but also the model used in the coding process.

We consider $DL(D, M)$ as a model selection indicator. That is, employing a coding scheme, the number of bits required to encode the data indicates the accuracy of the model used in the coding process. According to the Shannon coding theorem [21], the ideal code

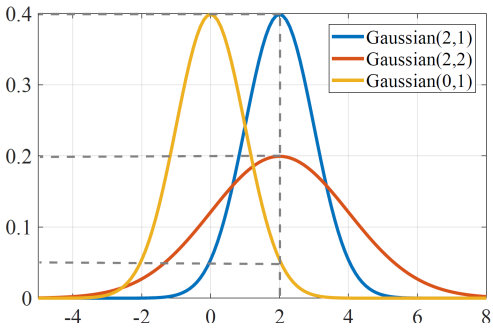


Fig. 2 Various fitted PDFs for a synthetic time series x generated by a $Gaussian(2,1)$ model.

length is related to the likelihood and is bounded by the entropy. More precisely, for an outcome a the number of bits required for coding is defined by $\log_2 \frac{1}{P(a)}$, where $P(.)$ shows the probability of a with the assumption that $\lim_{P(a)\to 0^+} P(a)\log_2(P(a)) = 0$. This coding scheme is also known as *log loss*. As a consequence, we assign shorter bit strings to the outcomes with higher probability and longer bit strings to outcomes with lower probability.

To elaborate the concept, we generate a continuous time series following Gaussian distribution, i.e. $x \sim Gaussian(2,1)$ distribution. Figure 2 shows the *probability density function* (PDF) w.r.t. the true model ($G_1 := Gaussian(2,1)$, the blue line) and two other PDFs corresponding to models with the lower accuracy, i.e. $G_2 := Gaussian(2,2)$ (the red line) and $G_3 := Gaussian(0,1)$ (the orange line). Applying the Shannon's theorem, we compute the compression cost of the outcome $a = 2$ w.r.t. three models as follows:

$$-\log_2 PDF_{G_1}(2) = -\log_2(0.4) = 1.32$$

$$-\log_2 PDF_{G_2}(2) = -\log_2(0.2) = 2.32$$

$$-\log_2 PDF_{G_3}(2) = -\log_2(0.05) = 4.32$$

Thus, the compression cost is in an inverse relationship with the likelihood of an outcome. The better the model fits the data (G_2 in our example), the more likely the observations are and hence the lower the compression cost is. Moreover, PDF is a relative likelihood for every outcome and it is not necessarily less than or equal to 1. Thus, in order to avoid negative number of bits caused by $PDF(.) > 1$, we consider a resolution parameter γ in the sense that the coding cost is $-\log_2 PDF(a)\cdot\gamma$. The parameter γ is a constant real number ensuring that the coding cost is always positive. Therefore, specifying γ is straightforward and needs to be set in the way that $\forall a \in D$, $PDF(a).\gamma \leq 1$. Then, by setting $\gamma \geq \max_{\forall a\in D} PDF(a)$ we make sure that the coding cost is always positive.

4.3.2 Causal Inference by MDL

Granger, in the original paper, measures the predictability in terms of the variance of the error in regression [9]. That is, for two time series x and y if $\sigma^2(x|\mathcal{I}) < \sigma^2(x|\mathcal{I}_{\neg y})$ then y Granger–causes x. Here, we regard measuring the predictability to a compression problem.

Back to Section 4.2, let $P(x^t|x^{t-d},...,x^{t-1})$ denote the predictive model w.r.t. Model 4 showing the probability of an outcome $x^t, t = 1,...,n$ w.r.t. the past observations of x up to time $t-1$. We assume that P belongs to a class of prediction strategies, i.e. $P \in \mathcal{P}$. Thus, following MDL–principle, the coding cost of time series x assuming Model 4 is defined as:

$$DL(x|M_x) = \sum_{t=d}^{n} -\log P(x^t|x^{t-d},...,x^{t-1}) \tag{7}$$

Moreover, let $P(x^t|x^{t-d},...,x^{t-1},y^{t-d},...,y^{t-1})$ denote the predictive model w.r.t. Model 5 outputting the probability of an outcome x^t assuming the past observations of x and y. Analogously, the coding cost of time series x assuming Model 5 is defined as:

$$DL(x|M_{xy}) = \sum_{t=d}^{n} -\log P(x^t|x^{t-d},...,x^{t-1},y^{t-d},...,y^{t-1}) \tag{8}$$

Referring to the generalized definition of Granger causality (Section 4.2), time series y causes x when using M_{xy} instead of M_x improves the predictability of x. That is, if y causes x, including y leads to higher probability for the observations in x, i.e. $P(x^t|x^{t-d},...,x^{t-1}) < P(x^t|x^{t-d},...,x^{t-1},y^{t-d},...,y^{t-1})$. Since higher probabilities (more accurate models) result the smaller number of required bits for encoding the data (Section 4.3.1), therefore $DL(x|M_{xy}) < DL(x|M_x)$.

The next part of MDL incorporates the model complexity. Then, we need to not only encode the data but also the model parameters used for coding in order to avoid complex models. That is, a proper model is the one which improves the predictability and at the same time it is not too complex. Thus, we say, M_{xy} fits the characteristics of the data more appropriately only if it is beneficial both aspects, predictability and model cost, i.e. $DL(x|M_{xy})+DL(M_{xy}) < DL(x|M_x)+DL(M_x)$. In the next section we introduce the model complexity in more detail. But for now, we consider the coefficients as well as the link function used in a model as model parameters which need to be coded. Altogether, assessing the predictability with data compression and employing

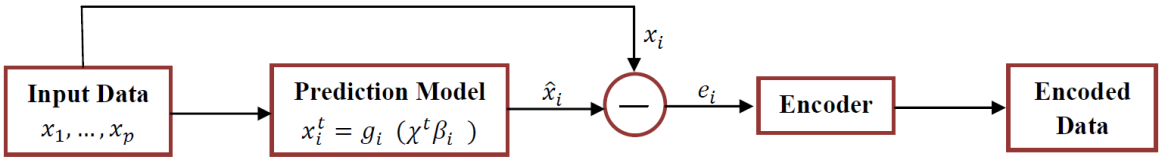


Fig. 3 Predictive coding scheme.

MDL to find the costs, we come up with the following definition for Granger causality:

Definition 2 MDL–based Granger Causality: Given two time series x and y, y Granger–causes x if the description length of x assuming M_{xy} is smaller than the one assuming M_x, i.e. $DL(x, M_{xy}) < DL(x, M_x)$.

Next, we elaborate our heterogeneous MDL–based framework for Granger causal inference as well as the proposed objective.

4.4 Heterogeneous MDL–based Granger Causal Framework

Given p time series $x_1, ..., x_p$, the generalized VAR model of order d w.r.t. x_i is defined as:

$$x_i{}^t = g_i(\mathcal{X}^t \cdot \beta_i) \tag{9}$$

where $\mathcal{X}^t$ is a concatenated vector of lagged observations $X^{t-d}, ..., X^{t-1}$ corresponding to all p time series when d is the lag. β_i is the regression coefficient vector consisting of $p \times d$ coefficients. Now we extend the MDL–based definition of Granger causality (Definition 2) to a general form including p time series $x_1, ..., x_p$.

Definition 3 Multivariate MDL–based Granger Causality: Let $\mathcal{C}_i$ denote the set of all causal time series corresponding to x_i together with x_i itself where $|\mathcal{C}_i| \leq p$ for $i = 1, ..., n$. Then, $M_{\mathcal{C}_i}$ is a generalized VAR model (9) w.r.t. x_i including the lagged observations of time series in $\mathcal{C}_i$. Moreover, Let $M_{\mathcal{C}_i \cup x_j}$ represent the generalized VAR model w.r.t. x_i including all causal time series together with x_j. Then, x_j Granger–causes x_i if $DL(x_i, M_{\mathcal{C}_i \cup x_j}) < DL(x_i, M_{\mathcal{C}_i})$.

In the following we clarify how to encode a time series and compute the corresponding description length ($DL(.)$). In this paper, we utilize the *predictive coding scheme* to encode time series. Moreover, a detailed objective function is defined in Section 4.4.3.

4.4.1 Predictive Coding Scheme

One of the well–known approaches to encode time series is the predictive coding scheme where the prediction error w.r.t. a time series together with the parameters of the corresponding predictive model are encoded and transmitted. Figure 3 illustrates three major components of this scheme for any time series $x_i, i = 1, ..., p$, i.e. a prediction model, the error term and an encoder. As a prediction model for time series x_i, we consider the generalized VAR model as introduced in Definition 3. Let $\hat{x_i}^t$ be the predicted value of x_i at time t. Then, the prediction error $e_i{}^t$ is the difference between the observed value $x_i{}^t$ and the estimated value $\hat{x_i}^t$, i.e. $e_i{}^t = x_i{}^t - \hat{x_i}^t$. Finally the prediction error needs to be encoded by a an encoder and transmitted to the receiver along with parameters of the prediction model.

4.4.2 Fit the Distribution

In MDL–principle the true prediction strategy is assumed to be given in advance. Moreover, the link function in a GLM model depends on the distribution of the response variable. On the other side, only observational data is provided in practice where the true distributions for the time series are not known. Moreover, any certain assumptions about the distribution of the data leads to inaccuracy and information loss as we already mentioned it in Section 2 (see Figure 1). In this paper, we follow the MDL–principle discussed in Section 4.3.1 to find the most fitting predictive model for the data. That is, we utilize an information–theoretic score (compression cost) to find the appropriate distribution for a time series. We assume a set of candidate prediction strategies from the exponential family. Considering every candidate, we estimate the parameters for the generalized AR model (M_x) employing an estimator (e.g. maximum likelihood). Then, we compute the compression cost of the time series for the estimated models. As discussed in Section 4.3, the more a model fits the data, the smaller the description length is. Finally, we select the distribution with the lowest description length as the fitted model.

More precisely, let $\mathcal{P} = \{P_1, ..., P_m\}$ denote the set of the candidate prediction strategies (probability distributions) from the exponential family e.g. Gaussian, Poisson or Gamma. Thus, the optimal predictive model $P \in \mathcal{P}$ w.r.t. x is defined as:

$$P = \min_{\forall P_i \in \mathcal{P}} P_i(x, M_x) \tag{10}$$

4.4.3 Objective Function

Considering the predictive coding scheme, the prediction error needs to be encoded. In order to correctly decode the data, the model as well is required to be coded and transferred. We first focus on the error coding costs then on the model complexity and finally we introduce our integrative objective function for heterogeneous time series.

– **Data Compression:** Following the properties of a GLM framework, the prediction errors can have any distribution from the exponential family [14]. Since the true distribution for the error term is also unknown, we employ our proposed fitting procedure, discussed in the previous section, to find the most accurate distribution w.r.t. the error term. Thus, the coding cost of the prediction error e_i w.r.t. x_i is defined as:

$$DL(x_i|M_{\mathcal{C}_i}) = DL(e_i) = \sum_{t=1}^{n} -\log PDF_e({e_i}^t|{e_i}^{t-1}, ..., {e_i}^{t-d}) \tag{11}$$

where $PDF_e(.)$ is the most accurate model w.r.t. e_i and n is the length of time series x_i.

– **Model Complexity:** As mentioned, $M_{\mathcal{C}_i}$ (Section 4.4) denotes the prediction model corresponding to time series x_i where $\mathcal{C}_i$ consists of the causal time series w.r.t. x_i. Essentially, the parameters in this model are the regression coefficients or β_i (a vector of length $p \times d$) plus g_i, the appropriate link function considering the distribution of x_i. Following a central result from the theory of MDL [19], the parameter costs to model n observations of x_i w.r.t. the prediction model M_i is approximated by:

$$DL(M_{\mathcal{C}_i}) = \frac{m_i}{2} \log n \tag{12}$$

where m_i denote the number of parameters in $M_{\mathcal{C}_i}$, i.e. $m_i = p \times d + 1$. The model costs depend logarithmically on the length of time series x_i. The intention behind this formulation is that for shorter time series the parameters do not need to be coded with very high precision. However, we consider time series with the same length in this paper.

Altogether, for a data set D consisting of time series $x_1, ..., x_p$ our MDL–based objective function is defined as:

$$DL(D, M) = \sum_{i=1}^{p} DL(x_i|M_{\mathcal{C}_i}) + DL(M_{\mathcal{C}_i}) \tag{13}$$

where $M = \{M_{\mathcal{C}_i}|i = 1, ..., p\}$.

5 ITGH Algorithm

Most of the algorithms are designed to detect either x causes y or vice versa by a pairwise testing. Usually a causal indicator is defined and needs to be tested for any possible pair of time series. However, this approach leads to inefficiency by increasing the number of processes when detecting a Granger causal network. To cope with this issues, we propose ITGH (Algorithm 1) consisting of two main building blocks: (1) fitting a distribution to the time series in observational data without any assumption and (2) detecting the Granger causal network in a greedy way.

Algorithm 1 Granger Causal Network Detection by ITGH

```
ITGH (X = [x_1, ..., x_p])
adj = [0] // Output, a p × p adjacency matrix
fitDistribution(X); // explained in Section 4.4.2
for all x_i in X do
    S_i:= Sorted time series according to their dependencies w.r.t. x_i
    C_i = {x_i} // The set of all causal time series w.r.t. x_i
    DL_I = 0 // The cost including the candidate time series
    DL_E = 0 // The cost excluding the candidate
    while DL_I ≤ DL_E do
        x_j := The candidate, the first time series in S_i
        DL_I = DL(x_i, M_{C_i ∪ x_j})
        DL_E = DL(x_i, M_{C_i})
        if DL_I ≤ DL_E then
            adj(i, j) = 1 // x_j causes x_i
            remove x_j from S_i
            C_i = C_i ∪ x_j
        end if
    end while
end for
return (adj)
```

We consider the fitting procedure (*fitDistribution(.)* in Algorithm 1) as a preprocessing phase. That is, once

we find the most accurate fitted distribution w.r.t. every time series as explained in Section 4.4.2. Then, we use this information as an assumption in our greedy algorithm. To be fair, we also input the fitted distributions to other algorithms we compare to. Unlike other compression–based causal inference algorithms (e.g. CUTE [7]), we avoid the drawbacks of an exhaustive pairwise testing by introducing a fast greedy algorithm. That is, corresponding to any time series x_i, we sort $x_1, ..., x_p$ based on their dependencies in a regression model w.r.t. x_i. In fact (also inspired by [1]), the time series with the higher dependency w.r.t. x_i has the higher coefficients in the corresponding regression model. Thus, for any $x_i, i = 1, ..., n$ we iteratively include the time series with the higher dependency w.r.t. x_i in the regression model until this procedure improves the compression cost of x_i. Essentially, for a candidate x_j we compute the description length of x_i (see Definition 3) considering two models $M_{\mathcal{C}_i}$ and $M_{\mathcal{C}_i \cup x_j}$. If including x_j pays off in terms of the compression cost, we keep including the next time series. Otherwise, the procedure terminates when no further causes exist for x_i. The output of this algorithm is an adjacency matrix for the Granger causal network.

5.1 Computational Complexity

Referring to Algorithm 1, the procedure goes through all the time series $x_1, ..., x_p$ and performs the same procedure for any time series (line 4). Moreover, the algorithm ITGH is deterministic in the sense that investigating the causal relations for time series $x_1, ..., x_p$ in any random order leads to the same Granger causal graph. Hence, we focus on complexity analysis of a random time series x_i (line 5–18) and at the end we introduce the general complexity of ITGH.

When detecting the causal relations for the time series x_i we need to sort all the other time series $x_1, ..., x_p$ once in the beginning (line 5). The complexity of a fast sorting algorithm is $\mathcal{O}(p \log(p))$. However, the main part of ITGH, in terms of the runtime complexity, is investigating causal relations for a time series. That is, following the greedy algorithm (line 9–18), for a time series x_i at most p compression–based Granger tests should be considered. Moreover, in order to reveal whether or not there is a causal relation we need to fit two GLM models (line 11 and 12).

There are various methods concerning how to fit a GLM model, i.e. iteratively reweighted least squares (IRLS), maximum likelihood with a Newton approach or Bayesian methods. It is up to the user which method to be employed for fitting a GLM model. Focusing on IRLS, fitting procedure has a complexity of $\mathcal{O}(nc^2)$ where c is $d \times |\mathcal{C}_i \cup x_j|$ in line 11 or $d \times |\mathcal{C}_i|$ in line 12, i.e. $c \leq d \times p$. Thus, in the worst case the runtime complexity of lines 5 to 18 in Algorithm 1 is of order $\mathcal{O}(p \log(p)) + \mathcal{O}(pc^2 n)$. It can also happen that there is no causal relations w.r.t. x_i. In this case the greedy algorithm is of the order $\mathcal{O}(c^2 n)$ and terminates after one iteration.

Altogether, the runtime complexity of ITGH (including line 4) in the best case is $\mathcal{O}(p^2 \log(p)) + \mathcal{O}(pc^2 n)$ and in the worst case is $\mathcal{O}(p^2 \log(p)) + \mathcal{O}(p^2 c^2 n)$. However, mostly in reality $p \ll n$ which means the runtime complexity of ITGH is highly depending on n leading to a complexity of order $\mathcal{O}(c^2 n)$.

6 Experiments

In order to assess the performance of ITGH we compare our algorithm to state–of–the–art algorithms in terms of *F–measure* conducting experiments on synthetic and real–world data sets. Although there are many algorithms to infer the causality among time series, only few of them are applicable to heterogeneous data sets and also fewer deal with Granger causality. Therefore, we compare ITGH to SFGC [10], TEN [20] and HGGM [5] which are designed to deal with heterogeneous data sets. Moreover, we compare our algorithm to TCML [1], CUTE [7] and the basic Granger test (GT) [9] to investigate the effect of assuming a specific (mostly Gaussian) distribution for non–Gaussian processes or transforming time series. ITGH is implemented in MATLAB and for the other comparison methods we used their publicly available implementations and recommended parameter settings. The source code and data sets are publicly available at: `https://tinyurl.com/yar5yuoq`.

6.1 Evaluation Strategy

We utilize $F - measure$ to evaluate the similarity between the target causal graph (ground truth) and the output causal graph. Moreover, we distinguish between two entries in the adjacency matrix A, $A[i, j]$ and $A[j, i]$. Let A^* and $\hat{A}$ denote the true and the output adjacency matrix respectively. Thus, the evaluation measures for time series $x_1, ..., x_p$ are defined as:

$$Precision = \frac{|\{(i,j) \in P : \hat{A}[i,j] = A^*[i,j]\}|}{|\{(i,j) \in P : \hat{A}[i,j] = 1\}|}$$

$$Recall = \frac{|\{(i,j) \in P : \hat{A}[i,j] = A^*[i,j]\}|}{|\{(i,j) \in P : A^*[i,j] = 1\}|}$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall}.$$

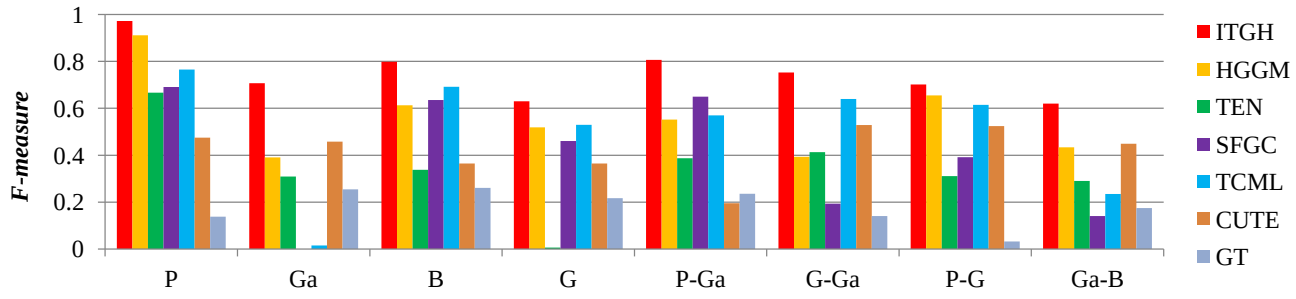


Fig. 4 Accuracy on homogeneous and heterogeneous synthetic data sets. P: Poisson, Ga: Gamma, G: Gaussian, B: Bernoulli

6.2 Synthetic Experiments

To investigate the efficiency and the effectiveness of ITGH compared to other algorithms, we generated various synthetic data sets. In the first series of experiments we aim at assessing the algorithms in terms of their effectiveness considering various homogeneous and heterogeneous data sets. Then, the scalability of ITGH is addressed in this section. In any synthetic experiment, we report the average performance of 50 iterations performed on different data sets with the given characteristics. The length of generated time series is always 1,000 except it is explicitly mentioned. Unless otherwise stated, we assume a random dependency level (strength of causal relations) among time series while generating them. In all the synthetic experiments we input the lag parameter as well as the true distributions used to generate the data to all the algorithms.

1. **Synthetic Data Generator:** In this section we clarify how we generated the synthetic data since it is not straight–forward to generate an appropriate data set. Following the principle of GLMs, we deal with the time series whose means depend on the past values of all time series through a concrete link function. This kind of dependency between the mean values and the other covariates are defined via a generalized VAR model.
   To generate random generalized VAR models, we consider a bottom up approach. That is, we start with a random adjacency matrix associated to a random causal graph. Then we assign directed edges between the nodes showing whether or not a Granger causal relations between two specific nodes (time series) exists. The ground truth in Figure 1 demonstrates a random graphical model over 5 time series as an example.
   Having formed the causal graph, we generate the coefficient matrix showing the dependencies among time series. Unless otherwise stated, we assume a random dependency level in every experiment. Moreover, We consider the lag d associated to each time series and corresponding to every time series we generate a random VAR model of order d assuming the coefficient matrix. The result is a generalized VAR model where outputs the mean values w.r.t. time series $x_i, i = 1, ..., p$ at any time point t. Thus, we randomly generate observations while taking the desired distribution for x_i into account.

2. **Accuracy:** In this experiment we generated various homogeneous and heterogeneous data sets from different distributions. Two discrete (Poisson and Bernoulli) and two continuous (Gamma and Gaussian) distributions were selected to cover some of possible combinations of distributions. Regarding any combination we generated 50 data sets each of which consists of four time series of length 1,000 with three causal relations where in mixed data sets the heterogeneity factor is 70%–30% (e.g. in Figure 4, 3 Poisson and 1 Gaussian time series).
   As it is observable in Figure 4, regardless of the homogeneity or heterogeneity of the data or even the distribution of the time series, ITGH outperforms other algorithms by a wide margin. Interestingly, we outperform TCML on Gaussian data sets although it is designed specifically for Gaussian time series and it performs better than other algorithms on such homogeneous data sets. This confirms the advantages of an MDL approach applied in a GLM framework to generalize the linear regressions. Focusing on the GT algorithm designed for single–type data sets, it is observable that GT is more efficient on homogeneous Gamma and Bernoulli data sets rather than on the mixture of them. The same be-

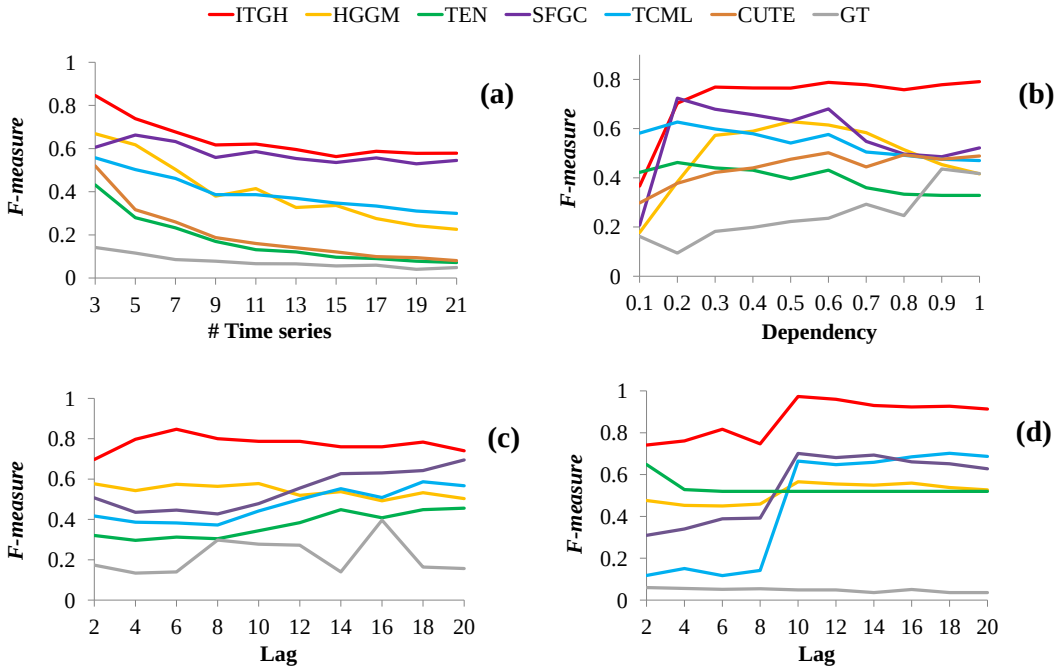


Fig. 5 Various experiments on synthetic data sets concerning the effectiveness.

haviour is also visible considering other combinations (e.g. Poisson–Gaussian).
On the other side, we outperform CUTE on the Bernoulli data set due to the inefficiency of pairwise testing compared to our proposed greedy approach. Although CUTE seems to be efficient on some heterogeneous data sets compared to the homogeneous ones (e.g. Gaussian–Gamma vs. Gamma and Gaussian), but this result does not sound reliable. Since the algorithm deals with only binary data, it is highly likely that binarising time series destroys the distribution of the data leading to some random binary time series. In the following we focus on a mixture of time series having Poisson and Gamma distribution as a representative for heterogeneous data sets.

3. **Effectiveness:** This experiment specifically investigates the effectiveness of the greedy approach in ITGH in terms of *F–measure* when the number of time series is increasing. Here we generate heterogeneous data sets where in any case 70% of the time series are Poisson and 30% are Gamma distributed and the number of causal relations is equal to 0.67% of the number of time series.
It is already expected that the performance of an exhaustive pairwise testing approach is decreasing when dealing with larger graphs. Figure 5a confirms our expectation and illustrates the constantly descending performance of HGGM, TEN and CUTE. As excepted, GT and SFGC are quite stable. However, GT is the worst algorithm in this experiment resulting in a maximum *F–measure* of 0.14. Moreover, this experiment shows the advantages of ITGH and SFGC compared to other algorithms regardless of the number of time series, although in the beginning their performance is affected by growing the causal graph.

4. **Dependency:** We refer to the coefficients of VAR models as the dependency which essentially show the strength of causal relations. In this experiment we investigate the performance of the algorithms concerning various dependencies ranging from 0.1 to 1. Analogously, we focus on data sets where a mixture of 3 Poisson and 1 Gamma time series are generated.
In Figure 5b any ascending or descending trend shows the inefficiency while a constant trend confirms the ability of an algorithm to deal with strong and weak causal relations. ITGH generally outperforms other competitors in terms of *F–measure* and unlike other algorithms, varying the dependency does not influence the performance of our algorithm significantly. Ignoring the starting point, the stable trend of ITGH confirms the efficiency of our algorithm even for lower dependency levels. Unexpectedly, the performance of TCML, SFGC and TEN is slightly descending in this experiment.

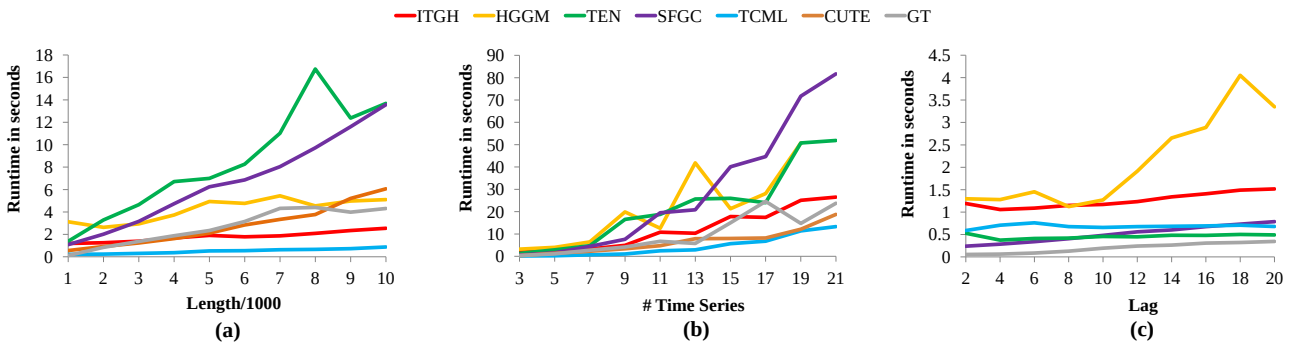


Fig. 6 Investigating the scalability in various experiments.

5. **Lag Experiment:** This experiment is particularly interesting for the real–world data sets where no information about the lag variable is given in advance. In our experiments all the algorithms except CUTE require the user to input the lag variable d. Figure 5c and d illustrate the result of conducting two experiments when random heterogeneous data sets consisting of two Poisson and one Gamma distributed time series of length 500 are generated. In the first experiment, the intention is to investigate how various algorithms behave when applied to various data sets generated with various lag variables ranging from 2 to 20.
   Essentially, an algorithm should perform efficiently regardless of the lag variable corresponding to the data. Therefore, stability of an algorithm seems to be crucial in this experiment. As it is evident in Figure 5c ITGH and HGGM are the most stable algorithms in this competition while the others are either increasing (SFGC, TCML and TEN) or have no trend (GT). Next, we aim at investigating the impact of choosing a lag variable different than the true lag.
   Thus, in the next experiment we generate synthetic data sets setting the lag parameter as 10. Then, we apply various algorithms while increasing the lag from 2 to 20 so that one can observe the result of choosing higher and lower amounts of this parameter. Referring to Figure 5d, almost all the algorithms result in a same trend when the performance is increasing with a big jump after the true lag is reached. Moreover, the performance is almost always stable for a lag variable higher than the true lag.

6. **Scalability:** We conducted three experiments to assess the scalability of ITGH compared to others. In every case we keep the heterogeneity of data sets as already explained (70% Poisson–30% Gamma) where the number of causal relations in every data set is equal to 0.67% of the number of time series. During the first experiment we vary the length of time series ranging from 1,000 to 10,000 when the number of time series is set to five.
   As Figure 6a depicts, ITGH is the second fastest algorithm in this experiment and outperforms all other algorithms designed for heterogeneous data sets, i.e. HGGM, TEN and SFGC. Together with TCML, our algorithm shows a perfect stable trend when increasing the length of time series.
   In the other experiment we iteratively increase the number of time series. As expected, all the algorithms have an increasing trend (Figure 6b). However, we outperform other heterogeneous algorithms in this experiment as well. Finally, the behaviour of the algorithms is investigated when the lag is increasing (same data sets as the experiment in Figure 5d). Except HGGM, all other algorithms are almost stable in this experiment (Figure 6c). Although ITGH seems to be relatively time–consuming compared to others, its runtime is less than 1.5 seconds and still reasonable.

6.3 Real Applications with Ground Truth

We conduct various experiments on publicly available homogeneous and heterogeneous real–world data sets where a valid ground truth is provided. Table 3 summarizes the characteristics of the data sets when the column *Fitted distribution* shows the best fitted distribution for time series by the procedure *fitDistribution(.)*

[1] `http://www.b30-oberschwaben.de/html/tabelle.html`
[2] `https://www.bafu.admin.ch/bafu/de/home/themen`
[3] `https://www.uwo.ca/stats`
[4] `https://webdav.tuebingen.mpg.de/cause-effect/`
[5] `http://archive.ics.uci.edu/ml/datasets/Abalone`
[6] Causal Inference on Event Sequences [7]

Data set	*Fitted distributions*	*Length*	*ITGH*	*SFGC*	*HGGM*	*TEN*	*TCML*	*CUTE*	*GT*
Traffic [1]	1 Poisson,1 Bernoulli	254	**1.00**	0.67	**1.00**	0.00	0.00	**1.00**	0.00
Ozone concentration [2]	2 Gaussian	365	**1.00**	0.50	0.67	0.00	0.40	0.00	0.67
Speed [4]	2 Gamma	202	**1.00**	0.00	**1.00**	0.00	0.00	**1.00**	0.00
Temperature [3]	2 Gaussian	168	**1.00**	0.00	0.00	**1.00**	0.40	**1.00**	0.67
Mooij [4]	2 Gaussian	16382	**1.00**	0.67	0.67	0.00	0.00	**1.00**	0.67
* Moffat [4]	2 Gamma, 1 Gaussian	721	**1.00**	0.33	0.67	0.50	0.45	0.00	0.67
* Abalone [5]	1 Poisson, 3 Gaussian	4177	**1.00**	0.56	0.00	**1.00**	0.00	**1.00**	0.67
* Energy Distributor [4]	1 Poisson, 2 Gaussian	9504	**0.89**	0.00	0.55	0.30	0.56	0.67	0.67
Neural Spike Train [6]	4 Bernoulli	1000	0.62	0.47	0.00	0.57	0.20	**0.76**	0.50

Table 3 Comparison on real–world data sets including a ground truth in terms of *F–measure*.

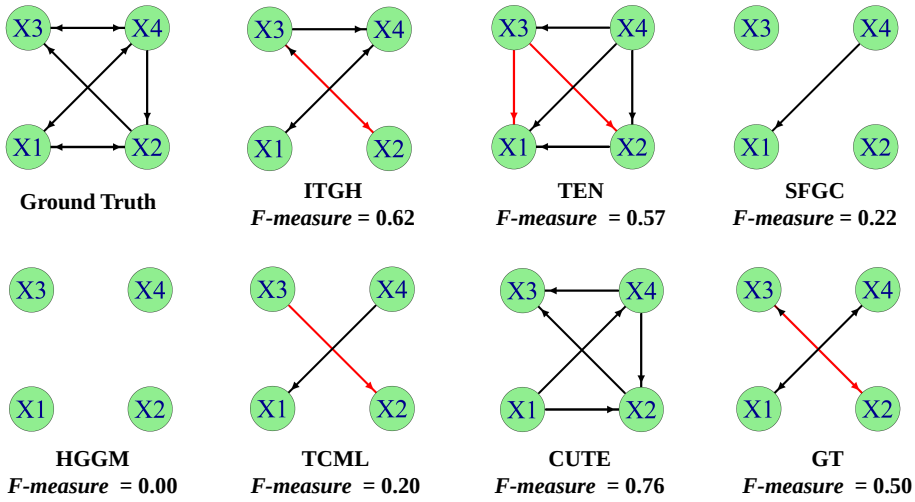


Fig. 7 Results on neural spike training data set. Black arrows show the correct Granger causal relations and the red arrows show the incorrect causal directions.

(see Section 5). We input the same fitted distribution to every algorithm. To be fair in real–world experiments, we investigate the performance of the algorithms considering various lags ranging from 1 to 20 and finally we report the best result for any algorithm in Table 3 in terms of *F–measure*.

The first data set, *Traffic*, consists of two time series of length 254, number of cars per day at different counting stations (a count Poisson distributed time series) and the type of days which can be either Sunday plus holidays or working days (a Boolean time series with the Bernoulli distribution). ITGH, HGGM and CUTE, correctly find the causal relation where the type of days influences the number of cars in stations. As expected, whether a day is a holiday or a working day impacts the traffic on the street.

The data set, *Ozone concentration*, comprises two continuous Gaussian time series of length 365 where the first measurement shows the daily mean values of Ozone (microgram/cubic meter) and the second one shows the temperature (Celsius) of the year 2009 in Switzerland. Referring to Table 3, ITGH is the only algorithm which is able to correctly infer that changes in the Ozone concentration cause changes in temperature, not other way around.

The next data set, *Speed* consists of two continuous Gamma time series. While recording *Speed*, a ball track has been used equipped with two pairs of light barriers. The first pair measures the initial speed and the second pair the speed of a ball at some later positions of the track. ITGH, HGGM and CUTE are able to detect the correct intuitive causal direction where the initial speed of a ball causes the later speed of the ball.

The next two data sets, *Temperature* and *Mooij*, investigate causal relations among measured indoor and outdoor temperatures while *Mooij* has a higher resolution (measured every 5 minutes). The *Temperature* data set consists of two Gaussian time series of the length 168 while Gaussian time series of length 16382 are provided in *Mooij* data set. Again, in this experiment MDL–based algorithms, ITGH and CUTE, correctly infer that the temperature outdoor causes the temperature inside as it is intuitively clear.

For the next three data sets, marked with *, the ground truth is given partially and the information about some interactions is missing in the causal graphs. There-

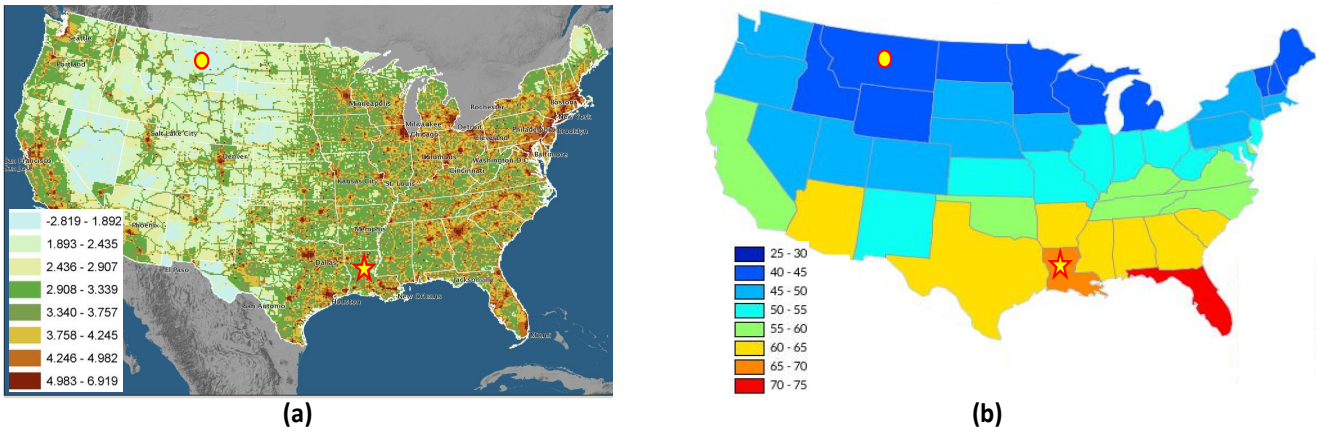


Fig. 8 a) Map of CO2 concentration, b) Annual average temperature of various states

fore, corresponding to any data set we report the average *F–measure* w.r.t. the causal pairs where the true information is given. *Moffat* consists of two Gamma time series (PPFD, a measure of light intensity in terms of photons and PPFDdir, a measure of direct solar light intensity in terms of photons that are available for photosynthesis) and a Gaussian time series (NEP, a measure of the carbon flux). In this experiment, ITGH is the only algorithm that correctly finds the true causal directions where PPFDdir and PPFD cause NEP.

Abalone concerns predicting the age of abalones (large, edible sea snails) based on the physical measurements. We consider 4 measurements, sex (Poisson), length (Gaussian), diameter (Gaussian), height (Gaussian). Obviously, sex influences the other measurements (also confirmed by [6]). As it is explicitly clear from Table 3, ITGH, TEN and CUTE, three information–theoretic–based algorithms, outperform other algorithms on this data set since they find three correct causal directions from sex to the other measurements.

The next data set, *Energy Distributor*, comprises three measurements: hour of the day (Poisson), temperature in degree Celsius (Gaussian) and the total electricity consumption in a region of Turkey (Gaussian). While outperforming other competitors, ITGH correctly infers the causal direction from the hour of the day to the temperature and the electricity consumption confirmed by the common sense that temperature and energy consumption depend on the hour of the day. Moreover, ITGH discovers the intuitive relation among temperature and energy consumption such that temperature affects the use of electricity of humans, while energy consumptions does not directly influence temperature in a region. However, ITGH finds another causal relation where energy consumption causes the hour of the day which is not plausible.

The *neural spike train* data set consists of records from an experiment carried out on a monkey. There are two types of influences in neural spike trains, excitatory (neurons fire more) and inhibitory (neurons fire less). The data set is investigated in [7] where the authors binarised the data resulting in 4 Bernoulli time series of the length 1,000. However, the original data, provided in [18], is not publicly available they obtained the data simulator where spike trains are generated using point process generalised linear models (GLM). Figure 7 illustrates the ground truth as well as resulted causal graph applying all the algorithms. ITGH (*F–measure*= 0.62) outperforms all other algorithms in this experiment except CUTE which performs better than ITGH on this data set with *F–measure*= 0.76. However, CUTE is not designed to deal with feedback loops while ITGH correctly finds the bidirectional causal relations between $X1$ and $X4$, as well as $X1$ and $X2$.

6.4 Application to Climatology

Nowadays, global warming and climate changes are the news headlines all over the world. But, what causes the climate changes? In this experiment, we investigate causal relations between the climate observations and various natural and artificial forcing factors when no ground truth is provided. The data set, provided in [11], is publicly available. We consider the monthly measurements of 11 factors over 13 years (from 1990 to 2002) in two states in the US, i.e. Montana and Louisiana: temperature (TMP), precipitation (PRE), vapor (VAP), cloud cover (CLD), wet days (WET), frost days (FRS), green house gases including Methane (CH4), Carbon Dioxide (CO2), Hydrogen (H2) and carbon monoxide (CO) and solar radiation including global extraterrestrial (GLO).

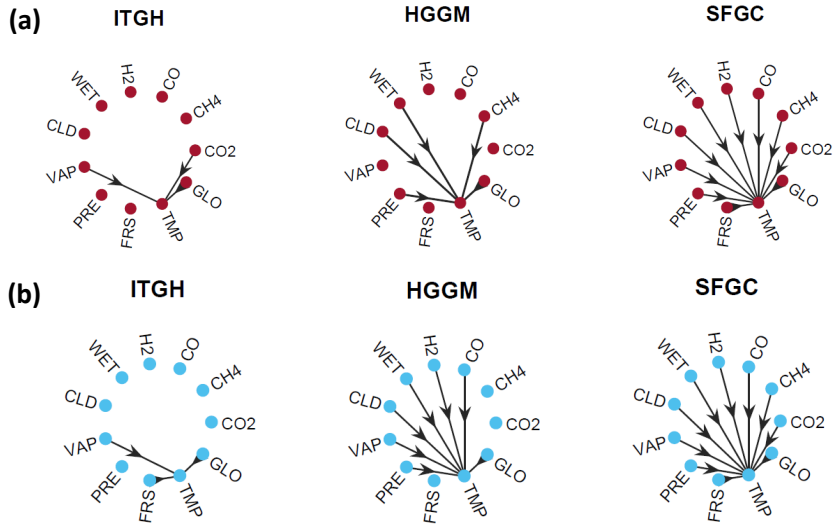


Fig. 9 Application to Climatology. a) resulted causal graphs for temperature in Louisiana state, b) resulted causal graphs for temperature in Montana state.

After fitting the distribution for any time series, we apply ITGH and other heterogeneous methods inputting the most appropriate distribution. The data providers suggested a maximum lag of 4 [11]. However, no exact information about the lag is given. Therefore, we select a random lag when it is smaller than 4. That is, the lag is set to 3 for the first experiment (Louisiana) and 2 for the second experiment (Montana). Since the temperature is the most concerning factor in global warming and also for a better visualization, we focus on the factors which influence the temperature. Green house gases, specially CO2, as well as solar radiation are the most important factors in global warming. Thus, in Figure 8a we provide a map of CO2 concentration as one of the main causes in climate changes[7]. Moreover, depending on where a state is located, cold or warm region, various climate measurements influence the temperature. Figure 8b summarizes the annual average temperature of various states[8].

According to Figure 8, Louisiana (marked by star) is located in the warm region of the US where the CO2 concentration is also high. As Figure 9a shows, ITGH correctly detects CO2 and the solar radiation as causal factors for temperature (confirmed by [11]). Moreover, influencing the temperature by VAP is also interpretable since Louisiana is located in the warm subtropical region. On the other side, the result of SFGC does not sound plausible since it finds a causal relation among all the factors and the temperature, even the frost days per month. HGGM seems more efficient compared to SFGC, However, it does not find any effects caused by one of the most effective factors, i.e. CO2.

Unlike Louisiana, Montana (marked by circle in Figure 8) is located in the cold region. Therefore, the detected causal direction from the frost days and vapor to the temperature in Figure 9b is reasonable (also confirmed by [11]). However, HGGM is not able to find the relation among the frost days and the temperature. According to Figure 8a the CO2 concentration in this state is not high. Therefore, CO2 does not influence the temperature in Montana dramatically. ITGH correctly does not consider a causal relation among CO2 and temperature while SFGC does. On the other side, HGGM is not able to find the effect of frost days, although it correctly recognizes the relation between CO2 and the temperature.

7 Conclusions and future work

In this paper we proposed ITGH, an information–theoretic algorithm for discovery of causal relations in a heterogeneous data set. We regard causality detection as a data compression problem where any improvement in the predictability is measured in terms of compression cost. Following the MDL–principle, we introduced an integrative objective function applicable for heterogeneous data sets. Profiting of a GLM framework our approach is generalized for time series having distributions from the exponential family. Our greedy approach (instead of an exhaustive pairwise causality test) leads to an effective and efficient algorithm without any assumption about the distribution of the data. To the best of our knowledge, there is no other MDL–based algorithm

[7] https://news.uns.purdue.edu/images/+2008/gurneyvulcan1.jpg

[8] https://www.currentresults.com/Weather/US/average-annual-state-temperatures.php

designed for Granger causal inference on heterogeneous data sets. One of the avenues for future work is to employ our MDL–based approach to efficiently detect the anomalies in heterogeneous data sets.

References

1. Arnold, A., Liu, Y., Abe, N.: Temporal causal modelling with graphical Granger methods. In: KDD (2007)
2. Bahadori, M.T., Liu, Y.: Granger causality analysis in irregular time series. In: SDM (2012)
3. Bahadori, M.T., Liu, Y.: An examination of practical granger causality inference. In: SDM (2013)
4. Barron, A., Rissanen, J., Yu, B.: The minimum description length principle in coding and modeling. IEEE Transactions on Information Theory (1998)
5. Behzadi, S., Schindler, K., Plant, C.: Granger causality for heterogeneous processes. In: PAKDD (2019)
6. Budhathoki, K., Vreeken, J.: Mdl for causal inference on discrete data. In: ICDM (2017)
7. Budhathoki, K., Vreeken, J.: Causal inference on event sequences. In: SDM (2018)
8. Eichler, M.D.: Causal inference in time series analysis (2012)
9. Granger, C.W.: Investigating causal relations by econometric models and cross-spectral methods. Econometrica (1969)
10. Kim, S., Putrino, D., Ghosh, S., Brown, E.N.: A granger causality measure for point process models of ensemble neural spiking activity. PLOS Computational Biology (2011)
11. Liu, Y., Niculescu-Mizil, A., Lozano, A., Lu, Y.: Learning temporal causal graphs for relational time-series analysis. In: ICML (2010)
12. Lütkepohl, H.: New introduction to multiple time series analysis. Springer (2005)
13. Marinazzo, D., Pellicoro, M., Stramaglia, S.: Kernel-granger causality and the analysis of dynamical networks. Phys. Rev. E (2008)
14. Nelder, J.A., Baker, R.J.: Generalized linear models. Encyclopedia of statistical sciences (1972)
15. Peters, J., Janzing, D., Schölkopf, B.: Causal inference on discrete data using additive noise models. IEEE transactions on pattern analysis and machine intelligence (2011)
16. Peters, J., Mooij, J.M., Janzing, D., Schölkopf, B.: Causal discovery with continuous additive noise models. J. Mach. Learn. Res. **15**, 2009–2053 (2014)
17. Qiu, H., Liu, Y., Subrahmanya, N.A., Li, W.: Granger causality for time-series anomaly detection. In: ICDM (2012)
18. Quinn, C.J., Coleman, T.P., Kiyavash, N., Hatsopoulos, N.G.: Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. Journal of Computational Neuroscience (2011)
19. Rissanen, J.: A universal prior for integers and estimation by minimum description length. The Annals of Statistics (1983)
20. Schreiber, T.: Measuring information transfer. Phys. Rev. Lett. (2000)
21. Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal **27**(4), 623–56 (1948)
22. Shojaie, A., Michailidis, G.: Discovering graphical Granger causality using the truncating lasso penalty. Bioinformatics (2010)
23. Weissman, T., Kim, Y.H., Permuter, H.H.: Directed information, causal estimation, and communication in continuous time. IEEE Transactions on Information Theory (2013)