Navigating Diverse Datasets in the Face of Uncertainty

Alejandro Álvarez Ayllón

Programa Oficial de Doctorado en Ingeniería Informática (Computer Engineering)

Dirigida por: Dr. Juan Manuel Dodero Beardo y Dr. Manuel Palomo Duarte

Departamento de Ingeniería Informática
Universidad de Cádiz
España
April 27, 2023

Resumen

Uno de los mayores problemas del *big data* es el origen diverso de los datos. Un investigador puede estar interesado en agregar datos provenientes de múltiples ficheros que aún no han sido pre-procesados e insertados en un sistema de bases de datos, debiendo depurar y filtrar el contenido antes de poder extraer conocimiento.

La exploración directa de estos ficheros presentará serios problemas de rendimiento: examinar archivos sin ningún tipo de preparación ni indexación puede ser ineficiente tanto en términos de lectura de datos como de tiempo de ejecución. Por otro lado, ingerirlos en un sistema de base de datos antes de entenderlos introduce latencia y trabajo potencialmente redundante si el esquema elegido no se ajusta a las consultas que se ejecutarán. Afortunadamente, nuestra revisión del estado del arte demuestra que existen múltiples soluciones posibles para explorar datos *in-situ* de manera efectiva.

Otra gran dificultad es la gestión de archivos de diversas procedencias, ya que su esquema y disposición pueden no ser compatibles, o no estar correctamente documentados. La mayoría de las soluciones encontradas pasan por alto esta problemática, especialmente en lo referente a datos numéricos e inciertos, como, por ejemplo, aquellos relacionados con atributos físicos generados en campos como la astronomía.

Nuestro objetivo principal es ayudar a los investigadores a explorar este tipo de datos sin procesamiento previo, almacenados en múltiples archivos, y empleando únicamente su distribución intrínseca.

En esta tesis primero introducimos el concepto de *Equally-Distributed Dependencies (EDD)* (Dependencias de Igualdad de Distribución), estableciendo las bases necesarias para ser capaz de emparejar conjuntos de datos con esquemas diferentes, pero con atributos en común. Luego, presentamos PresQ, un nuevo algoritmo probabilístico de búsqueda de *quasi-cliques* en hiper-grafos. El enfoque estadístico de PresQ permite proyectar el problema de búsqueda de EDD en el de búsqueda de quasi-cliques.

Por último, proponemos una prueba estadística basada en *Self-Organizing Maps (SOM)* (Mapa autoorganizado). Este método puede superar, en términos de poder estadístico, otras técnicas basadas en clasificadores, siendo en algunos casos comparable a métodos basados en *kernels*, con la ventaja adicional de ser interpretable.

Tanto PresQ como la prueba estadística basada en SOM pueden impulsar descubrimientos serendípicos.

Abstract

When exploring big volumes of data, one of the challenging aspects is their diversity of origin. Multiple files that have not yet been ingested into a database system may contain information of interest to a researcher, who must curate, understand and sieve their content before being able to extract knowledge.

Performance is one of the greatest difficulties in exploring these datasets. On the one hand, examining non-indexed, unprocessed files can be inefficient. On the other hand, any processing before its understanding introduces latency and potentially unnecessary work if the chosen schema matches poorly the data. We have surveyed the state-of-the-art and, fortunately, there exist multiple proposal of solutions to handle data *in-situ* performantly.

Another major difficulty is matching files from multiple origins since their schema and layout may not be compatible or properly documented. Most surveyed solutions overlook this problem, especially for numeric, uncertain data, as is typical in fields like astronomy.

The main objective of our research is to assist data scientists during the exploration of unprocessed, numerical, raw data distributed across multiple files based solely on its intrinsic distribution.

In this thesis, we first introduce the concept of *Equally-Distributed Dependencies*, which provides the foundations to match this kind of dataset. We propose PresQ, a novel algorithm that finds quasi-cliques on hypergraphs based on their expected statistical properties. The probabilistic approach of PresQ can be successfully exploited to mine EDD between diverse datasets when the underlying populations can be assumed to be the same.

Finally, we propose a two-sample statistical test based on Self-Organizing Maps (SOM). This method can outperform, in terms of power, other classifier-based two-sample tests, being in some cases comparable to kernel-based methods, with the advantage of being interpretable.

Both PresQ and the SOM-based statistical test can provide insights that drive serendipitous discoveries.

Agradecimientos

Este trabajo habría sido imposible sin el apoyo y ánimos de muchas personas de las que me siento afortunado por tener cerca. A todos ellos mi agradecimiento, y en especial

A Juanma y Manolo, por vuestra guía, vuestros consejos, y vuestra atención al detalle.

A mis padres, por todos vuestros sacrificios, por vuestra paciencia estirada hasta el límite, por la educación que me habéis dado. . . en fin, por todo.

A Fátima, por haber iluminado mi vida, por tu apoyo y paciencia, por escucharme cuando necesitaba hablar, y, sobre todo, por acompañarme en todo momento.

Y, por supuesto, a mis hijos Sara y Nikos. El mundo es mejor con vosotros.

Style reference

In this document, the following style guide is used:

> *Literal quotations from other authors are indented and in italics.*

The name of algorithms and software products are written with Small Caps; names of variables, filenames, etc., are displayed with a `fixed-width font`.

```python
def code_is_inlined():
    """
    With a fixed-width font and syntax highlighting whenever possible
    """
    pass
```

Contents

List of Figures

List of Tables

List of Codes

List of Acronyms

***k*NN** *k*-Nearest Neighbors 64, 72, 76, 85, 88, 89, 91, 92, 97, 104, 105, 109, 110, 116

AFDS Aircraft Fuel Distribution System 69, 73, 110

AQP Approximate Query Processing 27, 32, 37, 117

BMU Best Matching Unit 87, 88

CDF Cumulative Distribution Function 53

CERN European Organization for Nuclear Research 8, 14

Cosmos Cosmic Evolution Survey 152, 153

CRISP-DM CRoss Industry Standard Process for Data Mining 1, 13, 115

CSV Comma-separated Values 31, 143, 145

DBMS Database Management Systems 28, 31, 32

EDD Equally-Distributed Dependency 44, 51, 55–58, 60, 61, 63, 66–69, 73–75, 79, 81, 83, 86, 105, 107–110, 112, 116–118, 120, 121, 150

EMD Earth-Mover Distance 48

ESOM Emergent Self-Organizing Maps 88

FITS Flexible Image Transport System 145

FK Foreign-Key 48, 49

FWHM Full Width at Half Maximum 155

HEP High Energy Physics 14

IDE Interactive Data Exploration 7, 8, 10, 13, 40, 117

IMD Indices of Multiple Deprivations 99, 100

IND Inclusion Dependency 44–50, 52–54, 56, 57, 63, 67, 83, 116, 118, 120

KDD Knowledge Discovery in Databases 1

KiDS Kilo Degree Survey 152–154

KL Divergence Kullback-Leibler Divergence 94, 97

MMD Maximum Mean Discrepancy 91

PK Primary-Key 48

SDSS Sloan Digital Sky Survey 41, 42, 152–154

SNR Signal-to-Noise Ratio 97

SOM Self-Organizing Map 4, 50, 85, 86, 88–90, 92, 94, 97, 100–106, 110, 111, 113, 116, 118, 121

SVM Support Vector Machine 49

uEDD Unary Equally-Distributed Dependency 76

Il semble que la perfection soit atteinte non quand il n'y a plus rien à ajouter, mais quand il n'y a plus rien à retrancher.

— Antoine de Saint Exupéry

Chapter 1

Introduction

Nowadays, it is not uncommon for many types of users — from proficient data scientists to enthusiasts without formal training, from finance to physics — to dive into overwhelming data sets looking for any relevant pattern they can find. This data may consist of files that have not yet been ingested into a database system. The researcher must curate these files, understand and sieve their content, and extract and communicate information. This activity is known as *data exploration*, and it is an integral part of a new, data-intensive process of doing science that can be considered a new paradigm of scientific exploration, the fourth after the experimental, theoretical, and computer-simulation paradigms [BHS09; HTT09].

Data exploration is also known as Knowledge Discovery in Databases (KDD) because "knowledge" is the product of this process [Pia91; FPS96]. *Data Mining* is sometimes used as a synonym or as an integral part, as shown in figure 1.1 [FPS96; Rei99]. The latter interpretation is preferred for this work.

The CRoss Industry Standard Process for Data Mining (CRISP-DM) [She00] proposes a model for the data mining step, composed of six phases, shown in figure 1.2:

Business Understanding Definition of the requirements and objectives of a data mining project from the business (or domain) perspective.

Data Understanding Familiarization with the data collection. Domain knowledge is needed to understand the data, but the original project can be

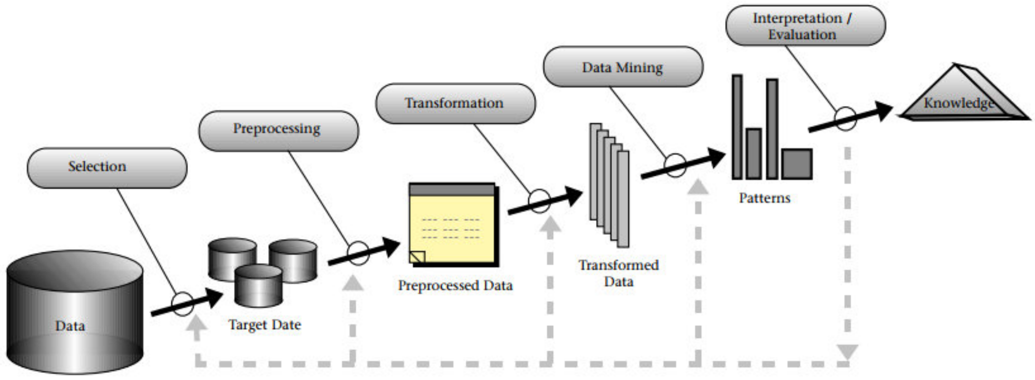


Figure 1.1: Knowledge Discovery in Databases (KDD).

refined as the data is best understood.

Data Preparation Attribute selection, cleaning, imputation, . . . are applied over the raw data.

Modeling Various modeling techniques are implemented, calibrated, and assessed. Different models may require different data preparation — for instance, cleaning, imputation, or normalization.

Evaluation The proposed models need to be thoroughly reviewed to ensure they meet the required quality and achieve the stated objectives.

Deployment The new knowledge has to be useful and actionable. Depending on the original objectives, the model can be integrated or transformed into an automatic system; or "simply" summarized into a report.

In this thesis, we focus on the **Data Understanding** phase, where the user interactively explores the data, gaining insight, and generating hypotheses during the process.

When starting the initial analysis, the data may be in a *raw* format: unprocessed files not optimized for access. Even worse, their schema may be inconsistent or poorly documented, and they may originate from different sources. These factors combined make the task of the data scientist more difficult:

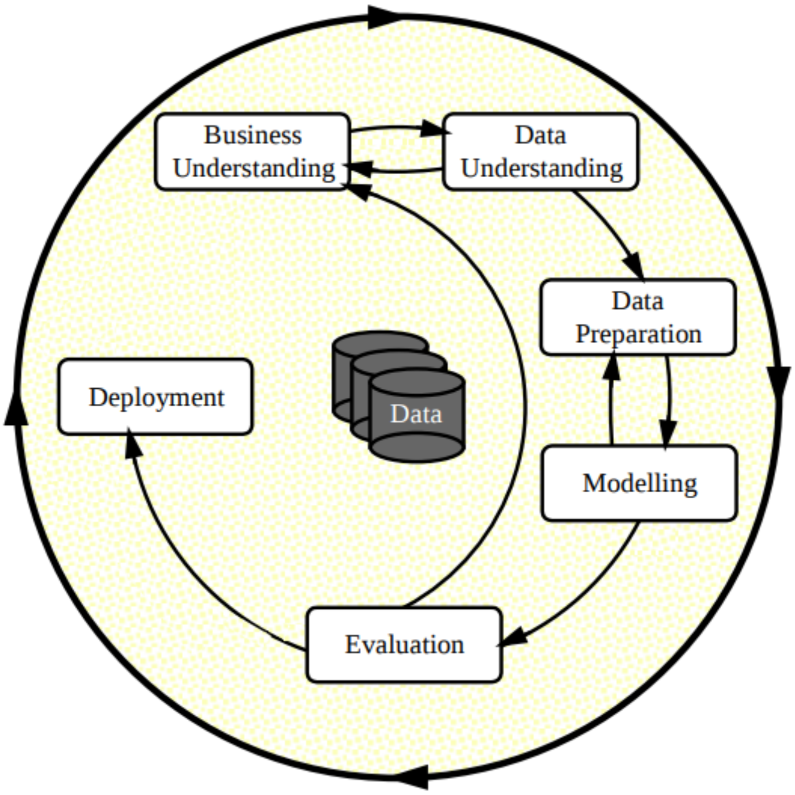


Figure 1.2: CRoss Industry Standard Process for Data Mining (CRISP-DM).

- Ingestion into a "proper" database introduces latency. Since the data is not well understood, any early design decision will soon become obsolete [Ker11]. Techniques for *in-situ* exploration try to overcome this difficulty by allowing direct examination of the data files performantly [Idr11].

- The data may be split into multiple files [Bau12], and these files may not follow the same schema [Ala14]. Data profiling and schema-matching tools can be helpful for this type of problem.

Astronomy is an example of a scientific discipline with vast amounts of digitized data readily accessible to the scientist, and, therefore, where *Data Mining* has been gaining more momentum [BB10]. A considerable portion of this data is made available by the community itself as independent files with little to no coordination in terms of schema consistency [Pep14]. Unfortunately, existing *in-situ* techniques leave out schema-matching, while the existing data profiling approaches require either relational data from discrete domains or are restricted to matching based on a single attribute. This motivates our research.

1.1 Objectives

Given the problem stated above, the main objective of this thesis is:

> To assist data scientists while exploring unprocessed, numerical, raw data distributed across multiple files based solely on its intrinsic distribution.

To make this objective attainable, we define the following sub-objectives:

1. **Find existing techniques** that help users to explore the data *in-situ*. A survey of the literature help users by directing them to algorithms and tools suitable for their use case.

2. **Identify gaps in the coverage of the existing techniques**, helping to direct the effort of present and future research into areas that need better coverage, widening the options available to users.

3. **Design new algorithms** tailored to numerical and uncertain data that cover part of the identified gaps, putting new tools at the disposal of data scientists.

1.2 Structure of this document

First, we described the methodology followed for this thesis in chapter 2. Chapter 3 contains a systematic literature mapping of the *in-situ* processing of scientific data. Then, chapter 4 identifies gaps in the literature regarding the exploration of diverse numerical datasets and summarizes some initial prototypes that remain open for further research. Chapter 5 proposes an algorithm suitable for schema matching tailored to scientific data. Chapter 6 outlines a statistical test based on Self-Organizing Maps (SOMs), which can bridge the gap between schema matching and *in-situ* access. Chapter 7 discusses our contributions and analyses the threats to the validity of the present thesis. Finally, chapter 8 summarizes our findings and contributions and proposes potential future lines of work.

Chapter 2

Methodology

For defining the methodology of our research, we follow the guidelines from *Researching Information Systems and Computing* [Oat06], a well-regarded reference text in information systems and computing research areas. Oates describes six fundamental aspects of research using the mnemonic ‘6P’:

Purpose A research project needs a well-defined objective to be able to define what it means to succeed — either totally or partially.

First, a *PhD* thesis must *increase the body of knowledge* of the chosen research area. Second, we want to *contribute to the solution of an existing problem*: In our case, as discussed in chapter 1, we target the Interactive Data Exploration (IDE) research area.

Products Oates lists five different types of contributions to the body of knowledge based on an existing classification from Davis & Parker [Oat06; DP79]: Evidence, Methodology, Analysis, Theories, and Computer-based products.

Note that improvements are also considered a contribution under this classification.

For the current thesis, we aim to produce a new solution — which encompasses multiple contributions such as theories and computer-based products — for IDE, understanding as such novel algorithms and techniques. Additionally, the literature review is also a contribution (analysis).

Process

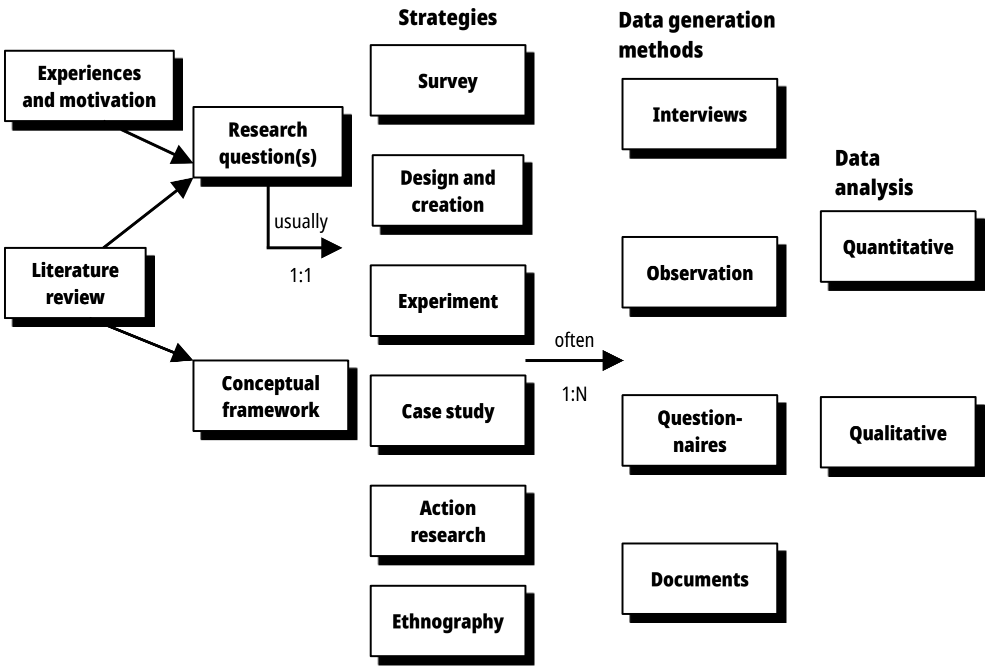


Figure 2.1: Model of the research process [Oat06].

Figure 2.1 shows the model of the research process. The time working at European Organization for Nuclear Research (CERN) and the *Astronomy Department of the University of Geneva* set up the *experiences and motivations* for this research: IDE on physical measurements, which have an intrinsic uncertainty. In chapter 1, we introduced the research questions for this thesis. In this chapter, we define the methodology for the literature review, the research strategy, the data generation methods, and the data analysis.

Participants The direct participants of the current research are the researcher, the tutors, and the thesis supervisors. Journal editors and reviewers are indirect participants.

Paradigm The paradigm is the philosophical model that frames the research. It defines what the researcher considers the nature of reality (*ontology*), how the researcher interacts with knowledge (*epistemology*), and how knowledge is acquired (*methodology*). By definition, this framework can not be proven [Gub90; GL94].

We can find different classifications of several paradigms depending on their views on these questions. For instance, Oates and Chua [Chu86] consider three branches:

- *Positivism*, where reality is objective and the researcher neutral.
- *Interpretative*, where truth is subjective and subject to the context.
- *Critic research*, where the social structure is the main focus.

Shull *et al.* [SSS08] extends this classification with a fourth paradigm: the *pragmatism*. In this paradigm, knowledge is evaluated based on its utility and is considered, in any case, approximate.

Considering our purpose and objective, and given the restricted list of participants, we follow the paradigm of *pragmatism* for this research project.

Presentation The main results from our research are compiled into the present thesis and published in peer-reviewed journals. Drafts have been published in `arXiv`. All the relevant source code is publicly available.

2.1 Literature review

A systematic mapping study is a process for the exploration of the situation of a wide research area with a high level of granularity, allowing us to identify parts of the domain that may be interesting to explore in more detail [KC07]. Because we are trying to obtain an overview of the situation of the research on data exploration techniques and identify where additional work may be required, we decided to follow this approach, and, more specifically, the guidelines proposed in *Systematic Mapping Studies in Software Engineering* [Pet07]. For completeness, we include in figure 2.2 the diagram of the process for a systematic mapping study, as defined by Petersen *et al.* .

2.2 Research strategy

From the list of strategies previously shown in figure 2.1, we follow *Design and Creation* and *Experiments*.

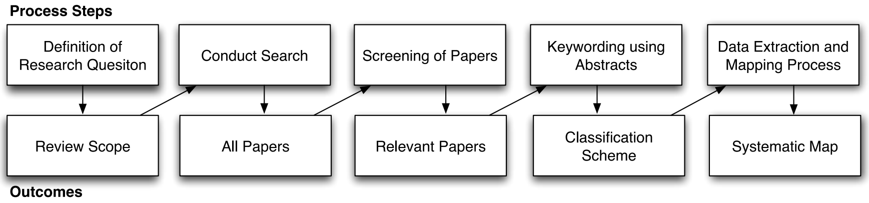


Figure 2.2: The Systematic Mapping process.

Design and Creation is an adequate strategy since we aim for a computer-based product. The resulting artifacts should be carefully studied, developed, and result from a careful engineering approach. Thus, we follow *Engineering Design*:

> *Engineering design is the systematic, intelligent generation and evaluation of specifications for artifacts whose form and function achieve stated objectives and satisfy specified constraints [DB12].*

Figure 2.3 summarizes the different stages for this method. We want to emphasize that this method is inherently iterative since each step provides feedback to the precedent stages. This work results from many such iterations: the research plan defined the initial task: IDE of raw scientific data. We performed a systematic literature mapping to identify solution principles and possible use cases. As a result of this literature review, we refined the initial task: data exploration of multiple files with related raw scientific data but incomplete metadata.

Experiments *Engineering Design* incorporates the development of multiple preliminary products, which need to be compared and evaluated before refining them into the final deliverable. Thus, experiments are a central aspect of this method.

2.3 Data generation methods

From the proposed data generation methods, we use **documents**, such as scientific papers to obtain datasets for the experiments.

With these datasets, we run experiments and **observe** the results, using performance metrics to compare different algorithms and their parameterizations.

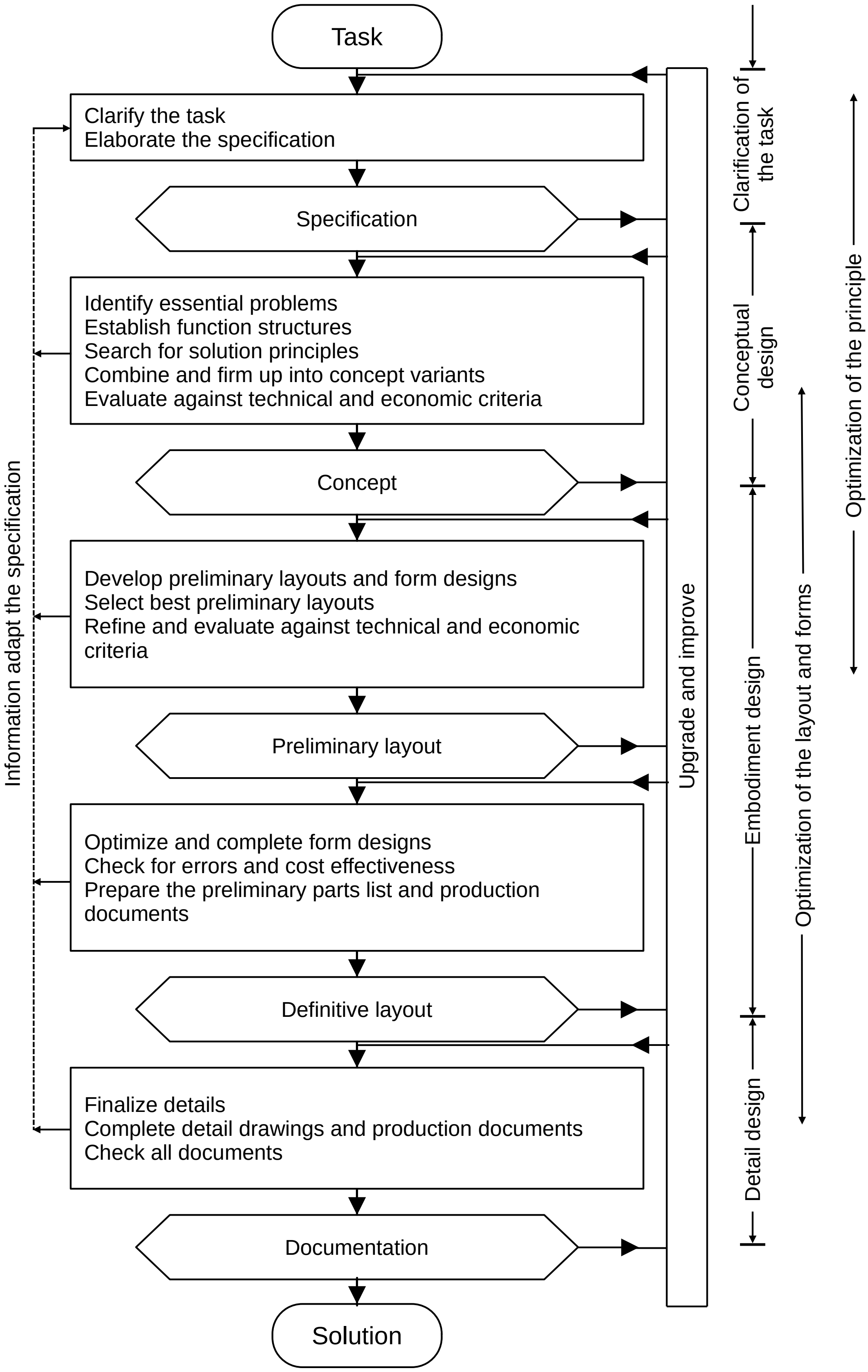

Task
Clarify the task
Elaborate the specification
Specification
Identify essential problems
Establish function structures
Search for solution principles
Combine and firm up into concept variants
Evaluate against technical and economic criteria
Concept
Develop preliminary layouts and form designs
Select best preliminary layouts
Refine and evaluate against technical and economic criteria
Preliminary layout
Optimize and complete form designs
Check for errors and cost effectiveness
Prepare the preliminary parts list and production documents
Definitive layout
Finalize details
Complete detail drawings and production documents
Check all documents
Documentation
Solution
Information adapt the specification
Upgrade and improve
Clarification of the task
Conceptual design
Embodiment design
Detail design
Optimization of the principle
Optimization of the layout and forms

Figure 2.3: Engineering Design [DB12; PBW84].

2.4 Data analysis

We base our comparisons on **quantitative** metrics: run-time, success rate, statistical significance, etc.

2.5 Open Science

This research adheres to the *Open Science* principles [Pon15].

- **Open Access** Papers are published either on *Open Access* journals, or made accessible on pre-print servers.
- **Open Data** The results from our experiments are uploaded to a public server together with the source code.
- **Open Reproducible Research**
  - **Open Notebooks** Notebooks used to summarize the results are included next to the source code.
  - **Open Source** The source code is under a permissive free software license (MIT[1]).
  - **Reproducibility Guidelines** Even if our results become inaccessible, the repositories include a list of the dependencies required to replicate the environment. The procedure followed to generate our results is documented in appendix A.
- **Open Repositories** All the delivered software is in GitHub, and a copy archived in Zenodo [EO13] with an associated DOI. Papers are available in pre-print servers such as arXiv and TechRxiv.

[1] https://opensource.org/licenses/MIT

Chapter 3

Literature Review

Following the *Engineering Design Process*, the initial objectives need to be clarified before searching for the solution principles. For this purpose, we performed a literature review to identify the state of the art.

We summarize in this chapter the stages and results of the *Systematic Mapping of the Literature*, a process described in section 2.1.

Since the *Engineering Design Process* is iterative and includes feedback loops, in section 3.1, we estate the objectives of the *original* literature mapping. These objectives were later refined, incorporating the results of this study. In section 3.2, we describe our method. In section 3.3, we summarize the results of the literature mapping. Finally, in section 3.4, we discuss the interpretation of our findings and insights.

Two surveys were done: the first in mid-2017 and the second at the end of 2022. The results from the 2017 survey appear in the article *Interactive Data Exploration of Distributed Raw Files: A Systematic Mapping Study* [APD19], published on IEEE Access.

3.1 Overview

IDE tools target the Data Understanding phase of CRISP-DM. They have human intuition as a core part of the process, where the user tentatively explores the data, iterating and reformulating the queries as their knowledge and insight change with each iteration.

A system that can be used in such a way needs to be lightweight, adaptive and have reasonably low response times—[Mil68] considers two seconds to be the upper limit for the continuity of thoughts—, helping and assisting without getting in the way of the person involved in the loop.

Because of this exploratory nature, early decisions on data structure, storage, and indexing are inappropriate [Ker11]. They introduce latency and optimize for a pattern that only holds for a brief period of time.

This problem can be tackled at different levels—from the physical layout on disk to the interface interacting with the user. In 2015, Idreos [IPC15] classified several of these solutions depending on their take on the issue. This paper originally attracted our attention due to the potential applications in High Energy Physics (HEP)[1], although the techniques found can be of interest to other scientific domains.

In summary, we need to satisfy three main requirements:

1. *Interactive response times*, as already discussed.

2. *Access to raw data files.* Pre-loading data in main memory is not an option due to the data volume and because we aim for a system that extends and does not replace the existing data management solution.

3. Ideally, *distributed*, since files are stored and replicated by an already existing distributed storage system [Bau12].

The granularity of the access has to be higher than *file level* because scientists normally care about datasets that are defined by the origin of the data — i.e., experiment and year —, and one dataset may be distributed across several files.

[1] This research was initiated while employed at CERN, so HEP was the original target use case.

3.2 Method

Idreos *et al.* [IPC15] propose a classification of different possible approaches to our problem. This study provides an excellent introduction, but we wanted to expand on it by answering three questions that were not covered by the original paper, and we also wished to survey the subsequent evolution of the domain.

3.2.1 Research questions

RQ1. How has the research area evolved?

Given that this is an active research area, it has probably progressed since the tutorial that we are using as a baseline. Therefore, the first question to answer to decide how to focus future research is: How has it evolved since 2015?

RQ2. What is the maturity level of the research area?

How many complete and reliable solutions are available? Are they successfully implemented in practice? How do they improve the users' experience? Identifying publications is not enough, we also want to assess what part of the software life cycle they focus on.

RQ3. How far are we from a tool that solves our three requirements?

The final target of this research is to identify solutions that cover our three requirements. Even though Idreos closed their tutorial by mentioning the importance of interconnection research [IPC15], they do not provide any references or study on this area.

3.2.2 Search strategy

For the retrieval of studies, it is necessary to clearly define how the search is going to be performed. The *2017* survey combined three different strategies:

- Set of known works obtained from [IPC15] because our RQ2 is not covered by the original classification.
- Forward snowballing [WW02] from the known set of publications using Google Scholar.
- For completeness, database searches to improve the coverage of our study.

Jalali and Wohlin [JW12] argue that snowballing and database searches can lead to similar patterns, but they also agree that it is “not easy to draw any general conclusions” about if the conclusions obtained are the same using the two different approaches. Thus, we have opted to follow both.

The set of digital libraries consulted is:

- ACM Digital Library
- Elsevier (Science Direct)
- Springer
- IEEE Digital Library
- Wiley Online Library
- World Scientific Net

Given the fast pace at which the field moves, older papers have probably been superseded or, if still relevant, we expect them to be already included in [IPC15]. Consequently, we have limited the scope in time to studies published from 2010 onwards.

For the *2022 update*, we applied forward-snowballing from the most interesting works identified in the 2017 survey, plus a search in *ACM Digital Library*.

All references obtained by any previous method were imported into the *Zotero Reference Manager*. The definitive lists can be found on two public groups in Zotero.org:

- *Mid-2017* `4517638/ide-in-science`
- *End of 2022* `4966770/ide-in-science-update`

3.2.3 Study selection criteria

We based the initial screening of studies on titles, abstracts, and keywords. In some cases, when the information provided by these fields was insufficient to make a decision, we also considered their conclusions or read the complete study.

We have focused here on finding primary studies related to data exploration. The filtering was performed using the following exclusion criteria:

- *Unsupported language* Studies written in a language different than English, Spanish, or French
- *Incomplete publication* Abstract only, or presentations were excluded
- *Off topic* Out of the data exploration domain
- *Not a primary study* Secondary, tertiary and surveys
- *Duplication* In case of duplication or high similarity for the same set of authors, only the most complete or the most recent one was taken into account.

Those publications that passed the inclusion criteria were reviewed to ensure all their fields were correct. Normally, this should have been done during the previous stage but due to the sheer volume of publications yielded by the search strategy, this step was postponed until the filtering was done. Because only title and abstract were used for the filtering, this did not affect the final result.

3.2.4 Classification

Publications that pass the selection criteria were classified into two axes: data exploration facet and research type.

Category

As mentioned in section 3.2.1, we base our study on the classification done by Idreos *et al.* [IPC15], which is included for convenience in table 3.1. For more details, we refer the interested reader to Idreos' tutorial.

For our purposes, we assigned one single category to each work covered In our study, choosing the most prominent topic when more than one category could fit.

User Interaction			
Data Visualization	Visual Optimizations	Visual Tools	
Exploration Interfaces	Automatic Exploration	Assisted Query Formulation	Novel Query Interfaces
Middleware			
Interactive Performance Optimizations	Data Prefetching	Query Approximation	
Database Layer			
Indexes	Adaptive Indexing	Time Series	Flexible Engines
Data Storage	Adaptive Loading	Adaptive Storage	Sampling

Table 3.1: Categories of Interactive Data Exploration solutions.

Research type

To answer our second research question—the maturity of the area—we follow the classification of research approaches done by [Wie06], as our guidelines for systematic mapping do [Pet07]. We summarize the different research types in table 3.2.

As per this classification, we expect mature solutions that have been implemented in practice to be covered by one or more *Evaluation Research* studies. If, on the contrary, they are in very early stages, then the majority of the related studies will fall into the *Philosophical* or *Opinion* categories.

3.2.5 Data extraction and visualization

At this stage, the papers were filtered and classified. We needed to summarize the obtained data in a way that is useful to answer our research questions.

To answer *RQ1*, we focused on the counting of each category and their visualization on a time series plot.

Research type	Description
Evaluation research	Investigation of a problem or implementation in practice.
Proposal of solution	These papers propose a solution and argue for its relevance without complete validation. A proof-of-concept may be offered.
Validation research	These papers investigate the properties of a solution proposal that has not yet been implemented in practice.
Philosophical papers	These papers sketch a new way of looking at things, a conceptual framework, etc.
Opinion papers	These papers contain the author's opinion.
Personal experience papers	These papers should contain a list of lessons learned by the author from his or her own experience. The evidence can be anecdotal.

Table 3.2: Research type for the Systematic Mapping.

To answer *RQ2*, a bubble plot can help identify the most frequent research type per category. In this way, we can distinguish if one area is more mature than another. Additionally, we labeled publications including some sort of user study, which should prove if any particular solution successfully improves the integration of a human on the loop.

Finally, for *RQ3*, we flag interesting papers classified under *Proposal of Solution* with the three requirements separately, if stated on their abstract or conclusions.

Additionally, while it was not in the original research questions, we extract which publication forums are the most prominent in our results.

3.3 Results

In this section, we describe the outcome of each stage of the systematic mapping.

3.3.1 Study selection

Table 3.3 displays the search queries that were used for each digital library. Note that for the end-of-2022 update, only the *ACM Digital Library* was queried since it indexes the most relevant journals and conferences from 2017 — see table 3.6.

For the 2017 survey, all searches were done on May 16th, 2017, and they yielded a total of 5,525 articles. Additionally, Idreos' tutorial provided 47 papers, and the forward snowballing provided 116. From this total of 5,688, only 242 — 4.25%— were accepted.

For the 2022 survey, all searches were done on the 25th of February 2023, limiting the results to those published between the 1st of June 2017 and the 31st of December 2022. For the *ACM Digital Library search*, we explicitly filtered non-primary studies, although some still were found. Due to limitations of the search engine, only 2,000 articles were found, later reduced to 1,884 after de-duplication[2]. Forward snowballing provided 584 articles, of which 452 are articles referencing BlinkDB [Aga13]! From this total of 2,468 articles, 89 — 3.61% — were accepted.

The details are shown in table 3.4. The rather low hit ratio mostly comes from the online searching of digital libraries because the lack of well-defined, or univocal, keywords makes it difficult to decide what to search for. We do not seem alone in this respect [KB13; JS07]. Even with the keywords defined, and because we must use different search engines, there are few or no commonalities between the way queries can be written and handled between different archives [Bai07; Bre07].

The yield of our systematic mapping is no smaller than those of systematic studies in other fields, which can be as low as 0.3% [Oak03].

3.3.2 Study data extraction

Table 3.5 displays the frequency of publications for each classification cluster proposed by Idreos [IPC15]. It is worth mentioning that four papers on the

[2] De-duplication of *the same* article appearing multiple times on the search results.

Library	**Scope**	**Search**
ACM Digital Library	Full text	`("RAW data" OR "RAW file" OR "ROOT file") AND (query OR exploration)`
ScienceDirect	Title, abstract, keywords (computer science)	`((RAW OR ROOT) AND (query OR exploration))`
Springer	Full text (computer science)	`("RAW data") AND (query OR exploration) + ("RAW file") AND (query OR exploration)`
Wiley Online Library	Abstract	`RAW AND query`
IEEE Digital Library	Abstract	`RAW AND query`
World Scientific Net	Full text (computer science)	`RAW AND query`

Table 3.3: Queries used to obtain the first set of articles for the Systematic Mapping.

Accepted	**Duplicated**	**Not Primary**	**Off Topic**	**Too Old**	**Total**
		2017			
242	9	16	5,295	126	5,688
4.25 %	0.16 %	0.28 %	93.09 %	2.22 %	100 %
		2022			
89	1	19	2,359	0	2,468
3.61 %	0.04 %	0.77 %	95.58 %	0.00 %	100 %

Table 3.4: Accepted and rejected papers count.

Database Layer did not fall into the predefined clusters, given their genericity [Ker11], or as an evaluation of different techniques [Sid17; Zou15; Pal15].

Figure 3.1 displays the frequency of each major cluster against the research type count for each one. In table 3.6, we display the publication forums where more than one study has been published. While there are two main forums, summing 40.18 % of all the publications, most of the papers are spread out in different conferences and journals.

It is worth noting that this table includes gray literature; that is, outside of the formal academic publishing. While one may argue that these papers have not yet been subject to a peer review, they are still included because gray literature can be, and is, a useful source of knowledge for information users [Law15]. In fact, Kitchenham *et al.* [KC07] recommended in their guidelines for systematic reviews to include gray literature in searches.

Category	**2017**	**2022**
User Interaction	**86**	**20**
Assisted Query Formulation	28	2
Visual Optimizations	25	8
Novel Query Interfaces	14	2
Visualization Tools	11	7
Automatic Exploration	7	1
Exploration Interfaces	1	0
Middleware	**48**	**43**
Query Approximation	34	40
Data Prefetching	14	3
Database Layer	**108**	**27**
Adaptive Indexing	26	5
Flexible Engines	16	8
Time Series	16	3
Sampling	15	5
Adaptive Storage	14	2
Adaptive Loading	10	1
Other	11	3

Table 3.5: Frequency of Interactive Data Exploration papers by category.

Publication	**2017**	**2022**
Journal	**55**	**15**
The VLDB Journal	11	3
IEEE Transactions on Knowledge and Data Engineering	3	1
IEEE Transactions on Visualization and Computer Graphics	3	4
International Journal of Cooperative Information Systems	3	
Journal of Big Data	3	
ACM Transactions on Database Systems	2	1
Future Generation Computer Systems	2	
SIGMOD Record	2	
Others	26	3
Conference	**181**	**73**
ACM International Conference on Management of Data	33	12
Proceedings of the VLDB Endowment	30	26
IEEE International Conference on Data Engineering	11	4
Conference on Innovative Data Systems Research	9	
Database Systems for Advanced Applications	5	3
International Conference on Scientific and Statistical Database Management	5	2
IEEE International Conference on Big Data	4	3
International Conference on Extending Database Technology	3	
International Workshop on Data Management on New Hardware	3	
ACM SIGMOD Symposium on Principles of Database Systems	2	
Advances in Visual Computing	2	
Big Data Analytics	2	
Database and Expert Systems Applications	2	
IEEE International Conference on Mobile Data Management	2	
Intelligent Information and Database Systems	2	
International Conference on Advanced Cloud and Big Data	2	
Workshop on Human-In-the-Loop Data Analytics	2	1
Others	62	22
Gray literature	**6**	**1**

Table 3.6: Frequency of papers by publication forum.

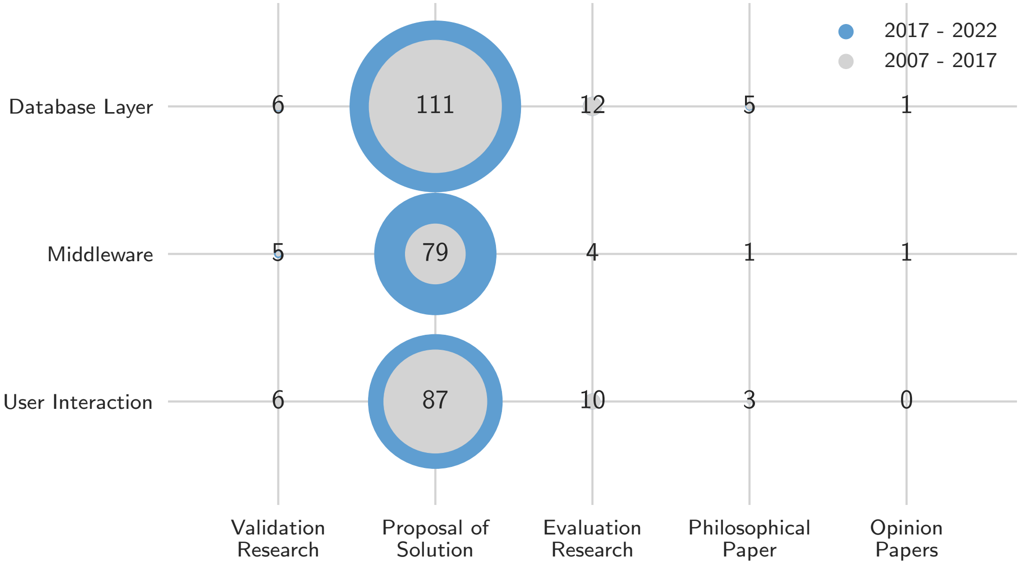


Figure 3.1: Interactive Data Exploration Layer vs Study research type.

3.4 Discussion

With these results, we now answer the three research questions in section 3.4.1. Then, in section 3.4.2 we explain the insights we obtain from these answers. Finally, we enumerate the threats to the validity of this study in section 3.4.3.

3.4.1 Answering the research questions

RQ1. How has the research area evolved?

Figure 3.2 displays the evolution during time of each of the three major classification clusters: user interaction, middleware and database.

Considering our search strategy, most of the results are posterior to 2012. Different approaches seem to be, in general, well balanced—we refer again to table 3.5—, although there is space for more works focused on *exploration interfaces* and *automatic exploration*, which are the less frequent published approaches. Interestingly, studies in the *Middleware Layer* have increase in relative popularity.

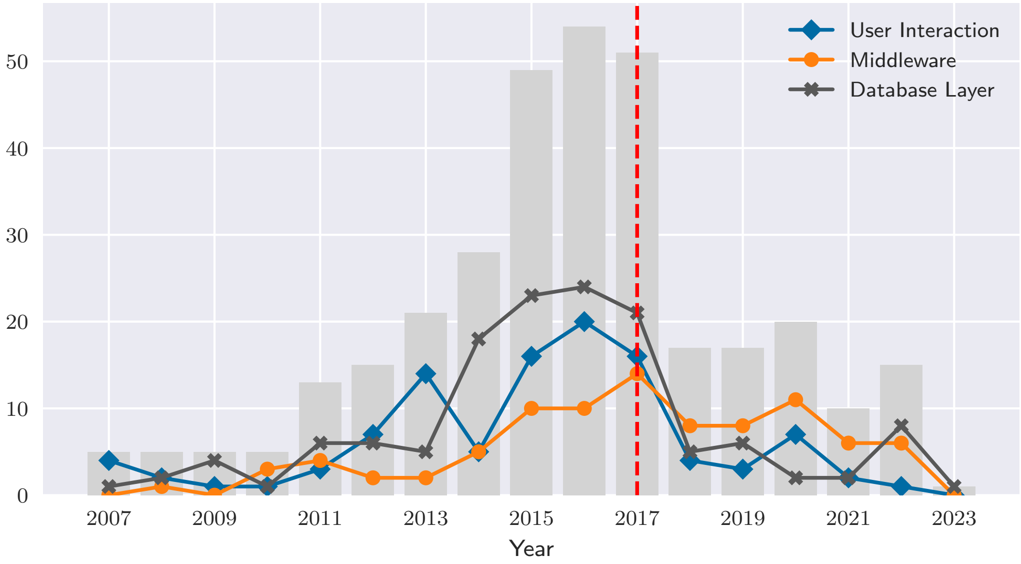


Figure 3.2: Number of papers per layer and year. The red line in 2017 separates the original survey from the update.

RQ2. What is the maturity level of the existing solutions?

We can use the figure 3.1 to answer this question. The vast majority of papers considered by this study—83.69 %—fall within the *proposal of solution* research type.

Meanwhile, *evaluation* and *validation* research are represented just by a 7.85 % and 5.14 %, respectively. Only 37 documents (11.2 %) include some sort of user study: 25 for 'User Interaction', 5 for 'Database Layer', and 5 for 'Middleware'. Research on how different solutions —either existing or proposed— perform in practice is lacking. This observation is true for both the original survey and the update.

These figures are hardly surprising because they seem to have been commonplace in computer science for a long time now [Tic95; ZW97; Sjø05]. For instance, Sjøberg *et al.* survey the status of controlled experiments in software engineering, and the numbers they find are equally low, with only 113 controlled experiments found on 5,453 papers [Sjø05].

It is hard and also out of the scope of this study to make some inferences from these results. Tichy *et al.* [Tic95] mention some potential reasons and measures to improve this situation, namely: difficulty in performing experiments where humans are involved, the lack of common benchmarks, or even that empirical work is not encouraged by the journals and conferences in this area.

RQ3. How far are we from a tool that solves our three requirements?

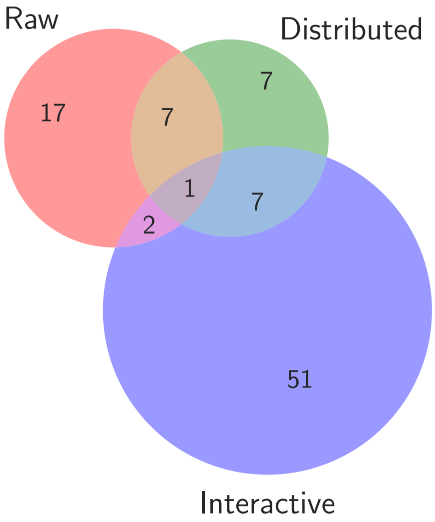


Figure 3.3: Venn diagram with satisfied initial requirements.

We display a Venn diagram with our three requirements in figure 3.3. We can see there is a single study that covers the three requirements: *A Distributed In-situ Analysis Method for Large-scale Scientific Data*, by D.Han *et al.* [HNK17]. While they mention the access over raw files and the fact that it is distributed, they do not explicitly state anything about their interactivity. However, the measured times for selective queries that they report are in the order of a few seconds. Consequently, we decided to consider it to be suitable for interactive usage.

The 2022 update yielded two additional works that target *interactive access* to *raw data*: *FlashView: An Interactive Visual Explorer for Raw Data* [Pan17] and *Resource-Aware Adaptive Indexing for in Situ Visual Exploration and Analytics* [Mar23].

In section 3.5, we summarize the most interesting set of proposals found in our literature mapping.

3.4.2 Study insights

Research in data exploration is very active, and there has been—and there is—a myriad of solutions proposed. This should not come as a surprise: in 2005 Stonebraker [SC05] had already stated this was bound to happen and predicted that there would be an increase in domain-specific tools. This would explain why, of all the classified studies, only one tool satisfies our three prerequisites.

In general, several systems and approaches have been proposed, which could, perhaps, be seen as building blocks. Not all combinations necessarily make sense, but there seem to be research opportunities in this direction, depending on the specific needs to be covered.

For instance, in our particular case, we could consider combining distributed access over raw files, as Han [HNK17] does, but using approximate query processing to reduce the response times.

Code generation is a popular approach for querying raw data files, and approximation-aware code generation has been noted as a challenge yet to be addressed [Moz17]. Another trend for Approximate Query Processing (AQP) research is Deep Learning. As examples, we can find Recurrent Neural Networks [SPF22; MCS21] for predicting future queries, Long Short-Term Memory networks for learning the relationship between query elements and query results, optimizations for querying Deep Neural Network models [Kan21], or deep generative models for aggregate queries [Thi20].

On an orthogonal consideration, since the generation of data volume will likely not slow down, the trend for more tools covering specific niches will probably continue. This diversity of tools is a challenge in many respects, for example: How do we choose the right solution? What is the cost of making the wrong choice? What happens if the chosen tool goes unmaintained in the future and there is no community around it? Will it be hard to maintain? Of course, these questions are not new in software engineering, but typically there are not many choices when deciding on traditional data storage systems, such as

Database Management Systems (DBMS). In the last decade, there has been an increase in available options (relational, object-oriented, schema-less, key-value, etc.) and, while opting for a DBMS has become harder, it has remained rather manageable. However, looking at the results of this study, the difficulty for users to decide will likely become more challenging.

3.4.3 Threats to validity

Search bias

The gaps identified may be covered in journals and conferences associated with the user domain—e.g., astrophysics—, rather than with computer science and engineering. The forward snowballing step reduces this risk because these hypothetical publications would most likely cite the original proposal of solution. Considering that our research method has allowed us to find even gray literature, we consider this risk to be low.

Filtering of articles

Given the huge number of papers that resulted from the search, a first filtering was done just based on the title and abstract. This is a difficult challenge. Unlike in other disciplines, sometimes abstracts do not contain enough information about the paper, and keywords can be inconsistent between journals and authors [Bud08; Bre07; JW12]. As recommended by [Bre07], we took into consideration the conclusions to cover this issue.

Classification

Another concern about these classifications is the bias of the researcher's own interpretation [Mac05]. For instance, *Jorgensen and Shepperd* report on a disagreement over 39% of the reviewed papers in their systematic review [JS07] due to different interpretations of the description of each category. We have been careful in this respect to guarantee the internal validity of the study, although some misclassification may still exist.

Additionally, it can be hard to identify if a solution covers or not one of the three predefined requirements based just on a paper. They may not have been explicitly mentioned if the authors did not consider them relevant for their publication. Therefore, there may have been false negatives.

The present paper documents our process and the resulting publication list has been made publicly available—see section 3.2.2—, so any interested reader can replicate and/or validate our results.

3.5 Discussion of relevant methods

Included for completeness is a summary of each of the nine publications that cover at least two out of the three requirements.

3.5.1 All three requirements

As already mentioned, the only solution that covers the three requirements is documented on the paper *A Distributed In-situ Analysis Method for Large-scale Scientific Data* [HNK17], classified as "adaptive loading".

The authors build on top of SciDB [Sto11], a distributed array-based scientific database, and focus on HDF files [HDF]. To avoid the overhead of data preloading, they leverage the flexible architecture of this database engine, providing their scan operator to read the data directly from the raw files when needed, which needs to be adapted to the internal representation of SciDB . This adaptation is made in two different stages: local and global mapping.

During the local mapping, they read on demand the data that matches the filters associated to the query, adapting it to the SciDB chunk representation: pieces of array data that are distributed together based on some policy - e.g., hashing, range partitioning.

At the global mapping stage, the resulting chunks are redistributed across the storage nodes following the SciDB policies.

Although not relevant to our use case, it is worth mentioning that they also merge small files to reduce the performance penalty of processing of them. This approach is interesting as it compartmentalizes well the logic required to access the raw data from the file distribution and the query engine.

However, the paper notably misses information about the network traffic caused by their global mapping stage since the network overhead depends on how the actual data distribution matches SciDB expectations.

3.5.2 Distributed access to raw files

DiNoDB [Tia14] is oriented towards the interactive development of data aggregation algorithms, where the user needs to quickly move between the batch processing stage and the interactive evaluation of the quality of the results.

It is deployed with Hadoop, and it generates the auxiliary metadata using user-defined functions executed by the reducers during the batch-processing stage. Therefore, the metadata ends up stored together with the raw data - the output of the reducers, and will also be replicated by the Hadoop Distributed File System (HDFS) across the cluster. Additionally, the output data may be cached optionally in memory - via ramfs or the filesystem cache.

For the interactive stage, on each HDFS Data Node, it is deployed an instance of a customized PostgresRaw [Ala12] database, a modified version of PostgreSQL with additional support for raw files based on positional maps - positions of attributes within the file.

With this architecture deployment, the client 1) issues the query to each node separately; 2) PostgresRaw uses the indices to retrieve the offsets of the relevant records and the positional maps to find the fields within the raw file; and 3) the client aggregates the results.

This approach gets good response times for the interactive stage, but the latency significantly increases when the output data does not fully fit into memory.

ARMFUL (Analysis of Raw data from Multiple Files) [Sil16b], probably has the most strict requirement set of all the analyzed papers. Its authors need to access raw data generated during the execution of a workflow and collect their provenance with high granularity. While other tools keep track of the data provenance at the file level - leaving it to the user the cross-match of records stored in different files - they can associate related data entries contained in the raw data files at the record level.

To do so, the authors formally define two additional workflow algebraic data operators [Oga11], which allows to address specific records stored on a file within a data flow: *Raw Data Extraction* - read, tokenize, filter, parse - and *Raw Data Indexing*. These operators can be composed with the existing ones, as *Map* or *Filter* - for instance, a user could map a list of file names to their content and then filter records with a specific threshold, keeping track of the provenance of the data during the process.

The indexing can rely on external tools and two implementations are provided: one based on bitmap indexes generated by FastBit [Wu09], and another one on positional maps, implemented following RAW's approach [Kar14].

Since this study particularly focuses on raw data access during simulations, the interactivity only applies to the queries made to the provenance database.

QUIS [CKJ17] is a *flexible engine* that provides its own query language as an extension of SQL, a set of adapters for a variety of data sources — Comma-separated Values (CSV), MS Excel, DBMS, etc. — and a query engine. The SQL extensions allow the user to specify the data sources and their schema.

ArrayUDF [Don17]. User-Defined Functions (UDF) allow developers to specify operations on *single* elements of a dataset: a tuple within a table or a cell in an array. ArrayUDF extends this possibility to functions over a neighboring range of elements, where the neighboring relationship can be flexibly specified — i.e., not limited to a rectangular window. The system is aware of the physical layout of the data and takes it into account when scheduling the operations across multiple parallel processing elements in order to maximize locality.

Diraq [Lak18] integrates indexing and compression of floating point numerical data, improving the efficiency of range queries and reducing the storage footprint at the same time. It does so by exploiting the IEEE floating point format, where the leading bytes —containing sign, exponent, and most significant bits of the mantissa— generally exhibit low cardinality and can be efficiently used for binning. When the system is comprised of multiple systems, an index is computed per *group of cores* as a well-balanced compromise between a per-core and a per-cluster index.

In **Distributed caching for processing raw arrays** [Zha18], its authors propose a distributed caching system aware of the necessary data locality for performant query computation: i.e., tuples from multiple, distributed, raw files are cached together and close to a compute node if it is likely that it will run future queries that need that data.

PS^3 (Partition Selection with Summary Statistics) [Ron20] improves the performance and accuracy of AQP using summary statistics to perform weighted sampling of the raw datasets: instead of randomly sampling the data, PS^3 builds sketches for each data partition, using them at query time to select which set of partitions to sample maximizing the accuracy while reducing the overhead. At initialization, the system samples a set of known, existing queries, and computes some summary statistics and the contribution of each partition to the answer. With this information, PS^3 trains a model that learns which subset of statistics best discriminates the contribution of a given partition. With this, given a query, the system can predict how much a given partition will contribute and will weigh it accordingly for the sampling.

3.5.3 Interactive access to raw files

FlashView: An interactive visual explorer for raw data [Pan17] is a *visualization tool* that does not aim at fully replacing DBMS, but rather at helping the user decide which data is worth loading into one. The user needs to provide a description of the data schema, which FlashView uses to sample the file when the first query arrives. Queries are treated hierarchically: samples obtained for an already processed query can be used to provide a first approximation of a new query if it is derived —i.e., additional filtering—, while the system takes additional samples. These new samples are streamed into the running queries following the hierarchy, building indexes as the queries are being processed, similar to the database *cracking* techniques [IKM07].

Resource-aware adaptive indexing for in situ visual exploration and analytics [Mar23] proposes two novel adaptive indexing techniques to improve the performance for visual exploration of raw data: *Categorical Exploration*

Tree (CET) for categorical attributes, and *Visual Exploration Tile-Tree Index* (VETI) that combines CET and tiling based on numerical or spatial attributes. On the first query, the file is parsed, and each tuple is assigned to a tile and indexed with VETI — the attributes used for the spatial partitioning need to be known. The categorical attributes of the tuples within a tile are indexed with a CET. During exploration, tiles may be split or merged to improve performance. The most interesting aspect is that to keep the index size within the allocated resources, the indexing is treated as an optimization problem. Roughly speaking, tiles and attributes are assigned an expectation of *utility* based on their probability of being requested by a future query, and CET trees are assigned a *cost* based on their memory requirement. Finally, there is an available *budget* for building the index. The problem can finally be mapped to the Knapsack Problem.

3.5.4 Distributed and interactive

Six out of the seven proposals are classified as “query approximation”, and the remaining, even though labeled as “visual optimization”, relies heavily on query approximation as well.

It would seem that to get fast responses, some compromises on precision must be made. This makes sense intuitively as processing fewer data will reduce the processing time at the cost of less accuracy. Additionally, some nodes may be offline, unresponsive, or overloaded on a distributed system. The results need to be aggregated within a reasonable deadline to keep the latency low, even if parts of the system have not responded yet.

It is worth noting that most of these papers also match the “sampling” category, but since sampling is just an aspect of the overall solution and their authors normally use “query approximation” to refer to their methods, we have decided to classify them as such.

BlinkDB [Aga13] allows users to perform SQL-like aggregation queries on data stored on HDFS, specifying time or error constraints. First, the authors base their system on the assumption - supported by evidence - that the column sets used for the aggregation queries are predictable, regardless of the actual grouping value. With this information, they perform a stratified sampling [Loh09] to avoid the under-representation of rare subgroups. Finally, the system chooses suitable samples based on the query constraints provided by the user, profiling them at run time so it can improve the execution plan for later queries.

ScalaR [BSC13] improves the performance of the visualization of big data sets dynamically reducing the size of the response returned to the front-end layer. Its authors provide an intermediate layer that consumes the queries issued by the user and uses the statistics computed by the database back-end to evaluate in advance the expected size of the result set. If this size is above a given threshold, the query is rewritten to either aggregate, sample or filter the data, generating a smaller approximate response that can be more performantly displayed.

Although their solution is back-end agnostic, their proposed implementation relies on SciDB [Sto11]. It quickly comes to mind that this could potentially be integrated with the previous method by Han *et al.* [HNK17], resulting in a visual exploration tool for raw data files.

The authors of **DICE** (Distributed and Interactive Cube Exploration) [Kam14] attack the problem on three fronts: speculative query execution, online data sampling, and an exploration model - *faceted* cube exploration - that limits the number of possible queries, improving the efficacy of the speculative execution.

Probably, the most interesting idea from this paper is the notion of the exploration being done in "sessions": The authors do not attempt to optimize for any possible query, but only for those that are likely to follow from the state of the current session. Predicting a set of potential following queries is made possible thanks to their exploration model, which restricts the possible number of "transitions" from the current state for a session.

The predicted queries are then ranked based on their likelihood and accuracy gain. Those most likely and providing the highest accuracy gain will be speculatively executed in advance, populating the cache. This way, when the final query arrives, the response can be built from the content of the cache if the predictions were successful. Otherwise, it will be scheduled to the underlying nodes.

For more information about "data cubes", we refer to the DICE paper, or the original proposal [Gra97].

AccuracyTrader [Han16] is a distributed approximate processing system comprised of two components: one online and another offline.

First, the offline part reduces the dimensionality of the original data using Single Value Decomposition - so it only supports numerical values. Then, it groups similar entries using an R-Tree, where each node represents an aggregated data point, and all nodes at the same level correspond to a "synopsis". This tree is flattened into an index at a level that balances the number of leaves under each aggregated data point and the selectivity of the tree at that level. Finally, it aggregates the data for each index entry using the original dimensions of the indexed points and stores this aggregated data in the "synopsis".

When a query arrives, the online part uses these "synopses" to produce an approximate result with an accuracy estimation. It then iterates using the detailed data points to improve the response accuracy until the deadline specified by the user expires.

In the paper, the authors prove that the system scales well in terms of tail latency and accuracy when the number of requests increases for a "search engine"-like workload. However, the data has to be aggregated into the synopsis beforehand.

KIWI [Kim15] is a SQL front-end built on top of Hadoop that aims to provide both batch processing and interactive analytics via approximate query processing. It generates both vertical (column) and horizontal (row) samples, and re-writes the queries to use these samples instead of the original data. However, it is hard to assess the technical soundness of this solution, since the paper is very short - 2 pages including citations - and we have not been able to find any later citations, nor do the authors cite other papers about the same tool.

Wang *et al.* [WCA15] introduce a framework based on the map-reduce paradigm. Instead of the traditional batch processing approach where the analysis is performed on big chunks of data, their system executes the analysis logic iteratively on samples, updating an estimator in each round until a stop condition is satisfied - both the estimator and condition are provided by the user. When the termination condition is satisfied, the remaining jobs are canceled, saving computing cycles and reducing the latency. Similarly to other analyzed solutions, they use stratified sampling to ensure good accuracy and coverage of rare cases. The sampling is done without replacement, so in each iteration, new data points are taken into account, improving the selectivity of the method.

Kayak [MT17] is a framework that defines its own set of *primitives* — insert dataset, search dataset, outlier detection — composed of reusable *tasks*, etc. — profiling, joinability computation, etc. Finally, tasks are decomposed in atomic *steps*. Individual steps have an associated cost function, and they can be executed to provide exact or approximate answers together with a confidence value. When the user performs a query, they can specify an acceptable error range. Knowing the system load, the target tolerance, and the estimated cost of each individual step, Kayak can use different strategies to optimize for confidence, time-to-first response, etc. Kayak leverages Apache Spark for the processing of large datasets, and Metanome [Pap15] for data profiling.

3.5.5 Summary

We can see some commonalities by looking at the underlying techniques used by the solutions described above:

First, for providing access to raw files, code generation, and positional mapping seem to provide a good solution. Both are implemented either directly — PostgresRaw — or used via integration with an existing implementation — DiNoDB. Isolating the raw data access as a database operator composes well for all studied solutions regardless of the framework of reference - workflow, PostgreSQL or SciDB .

Second, to provide interactivity on a distributed system, the engine needs to approximate the results using a deadline or an accuracy requirement as a stop condition. The resiliency and the low latency are achieved by being capable of processing only parts of the data via sampling — BlinkDB —, pre-computed summaries — AccuracyTrader — or both. In either case, error estimation becomes an important part of the system, both internally and as part of the interface exposed to the user. Looking at the 2022 update of the survey, BlinkDB's popularity — 867 citing papers on Google Scholar — indicates steady interest and relevance of sampling techniques for AQP.

Finally, since Deep Learning excels at pattern recognition and data summarization, these techniques are becoming more popular for AQP: for estimating the answer [RRS21; Thi20], predicting future queries [SPF22; MCS21], query optimization [Bi22], etc.

3.6 Conclusions

In this systematic mapping study, we have detailed the method that we followed to gather and filter papers related to *data exploration*, searching for solutions that tackle big data volumes stored in a distributed way and with a low latency. This process has produced 242 papers, which we have classified according to their approach [IPC15] on one axis and to their research type [Wie06] on another.

The results suggest that plenty of solutions have been proposed by researchers. However, there is rarely any follow-up, at least published, on their practical implementation, be it to confirm a successful introduction to users or to evaluate other tools already in place. Unfortunately, this is not different from the state of other areas of the computing sciences.

We have found evidence that code generation is a well-proven approach for accessing raw data files, although most solutions have not been generalized onto a distributed environment. Additionally, Deep Learning is becoming increasingly popular in AQP, given its summarization and pattern identification capabilities.

Finally, and as the main takeaway, we realized that most solutions treat files as separate, independent relations, leaving it to the end-user to work out how they are related, an observation shared by other authors [Sil16a]. One exception is Kayak [MT17], which integrates Metanome [Pap15] to extract metadata useful to link relations. Nevertheless, as we will see in the next chapter, the techniques from Metanome are not sufficient for numeric, uncertain data.

Chapter 4

Identifying gaps

As a result of the survey presented in chapter 3, we found that most solutions treat files as separate, independent relations. These files may not have yet been ingested into a database system and the schema may be unfamiliar and not adequately documented or even be composed of multiple files with heterogeneous schemes [Ala16; ZZ15]. We consider that Idreo's classification misses a category for this problem: *Schema Homogenization*. This new category belongs to the *Middleware* layer.

In *Data Mining in Astronomical Databases*, Borne describes how data exploration of this kind of diverse dataset is relevant since it can drive serendipitous discoveries [Bor00]. He proposes two groups of data mining approaches in this respect: event based and relationship based:

- **Event based**
  - *Known events / known algorithms* Use physical models to locate known phenomena of interest spatially or temporally within a large database.
  - *Known events / unknown algorithms* Use pattern recognition and clustering to discover new relationships between known phenomena.
  - *Unknown events / known algorithms* Use predictive models to predict the presence of unseen events within a large and complex database.
  - *Unknown events / unknown algorithms* Use thresholds to identify transient or unique events.

- **Relationship based**
  - *Spatial* Identify objects in the same location.
  - *Temporal* Identify events occurring within the same time period.
  - *Coincidence* In general, apply clustering techniques to identify objects that are co-located within a multidimensional space.

Borne then enumerates a list of science requirements for data mining:

- *Object Cross-Identification* between catalogs. Similar to the natural join in relational algebra, but based on spatial or multidimensional co-location.
- *Object Cross-Correlation* comparing sets of attributes over the full set of objects. For instance, identify remote galaxies as those that are *not* present on the ultraviolet spectrum.
- *Nearest-neighbor identification* or, in general, application of clustering algorithms in multidimensional spaces.
- *Systematic Data Exploration* via event- and relationship-based queries to a database hoping to make serendipitous discoveries.

Following the methodology described in chapter 2, we identified an essential problem: exploring multiple files with uncertain numerical data is a neglected aspect in the IDE. With this insight, we returned to the clarification stage and used Borne's description of data exploration in astronomy to better understand data exploration in this context.

In section 4.1, we use an existing astronomy database to expand our understanding of how users explore datasets. Then, in section 4.2, we refine and concretize the objectives of the present thesis. In section 4.3, we identify solution principles from the literature. Finally, in section 4.4, we list two proposals of solution and refer to the respective chapters that document them.

4.1 Examining real use cases

We looked for concrete examples of queries mentioning *astronomy* on the 242 articles classified on the systematic literature mapping from chapter 3.

It is soon evident that the Sloan Digital Sky Survey (SDSS) [Abo18] is popular as a test set since it is readily available and well documented [Gra02]. Furthermore, there are easily accessible sample queries [SDSa] and real ones [SDSb]. Listing 1 shows an example of how to obtain a list of queries performed by users.

```sql
    SELECT clientIP, seq, statement, elapsed
    FROM SQLlog
    WHERE yy=2018 AND mm>=10 AND rows>0 AND dbname LIKE 'BestDR14%'
```

Code 1: Example of how to obtain a list of queries performed by users during the end of 2018 over the 14th data release.

In total, 25 articles (10.3%) use SDSS as a test dataset. Table 4.1 classifies these 25 articles following the same schema as described in section 3.2.4.

Category	**Total**	**SDSS**	**%**
Exploration Interfaces	34	3	8.8%
Indexes	58	5	8.6%
Storage	39	11	28.3%
Data Visualization	36	2	5.6%
Interactive Performance Optimizations	48	4	8.3%

Table 4.1: Classification of the articles that use the data from the Sloan Digital Sky Survey.

To obtain an overview of the type of usual utilization of this database, we processed the results from query 1. We extracted the columns, relations, and filters usually affected by the queries. Table 4.2 shows the most frequent queried combinations of relations.

Interestingly, introspection queries are widespread, indicating that users spend

Tables	**Count**	**Percentage**
fGetNearbyObjEq, PhotoPrimary	264,785	37.62%
DBObjects†	150,458	21.37%
PhotoTag, fGetObjFromRectEq	93,094	13.23%
IndexMap†	41,265	5.86%
Galaxy, fGetNearbyObjEq	31,130	4.42%
sppParams, PhotoTag, fGetObjFromRectEq	29,805	4.23%

Table 4.2: Combination of relations most frequently queried. Tables marked with (†) are meta-data tables (i.e., describe the schema).

a considerable time familiarizing themselves with a complex schema. This has led to attempts at reducing this friction by methods such as context-aware auto-completion [Kho10].

4.2 Refining objectives

We summarize here our insights after the literature survey and examination of real use cases:

Support for data distributed across multiple files is generally neglected by *in-situ* data exploration solutions [Sil16a; Ala16]. However, indexing, storage, and interactivity are well covered, as seen in chapter 3.

Exploring the database schema itself, as the SDSS query logs suggest, is non-negligible user activity. Our observation is consistent with an IBM study that finds that even data architects can spend up to 70% of their time just discovering the metadata of databases [Wu08].

Our refined question is: can we help users to navigate datasets split across multiple files, with unknown schema, facilitating relationship-based mining? We can rely on name matching when metadata is present, but what can be done when it is missing, or if correspondences are not unambiguous?

4.3 Solution principles

Our use case looks like the following: an astronomer facing several data files containing raw astronomical measurements with little or no explanation about their schema. These files may come from different surveys or different sets of observations from the same region of the sky, and the user can only make the following educated guesses:

- The populations are likely the same, or at least very similar (i.e., stars).
- A subset of the attributes is shared between the relations (i.e., brightness on different electromagnetic bands).
- The measurements have an associated uncertainty [Sto09], either explicitly stated or not (i.e., random errors, instrument accuracy, floating point precision).

To help cross-matching the files, the first intuition would be to run a statistical test between all possible pairs of columns, such as the Kolmogorov-Smirnov [Hod58] or Wilcoxon [Wil45] tests. However, as figure 4.1 exemplifies, this information is not enough to do a cross-match.

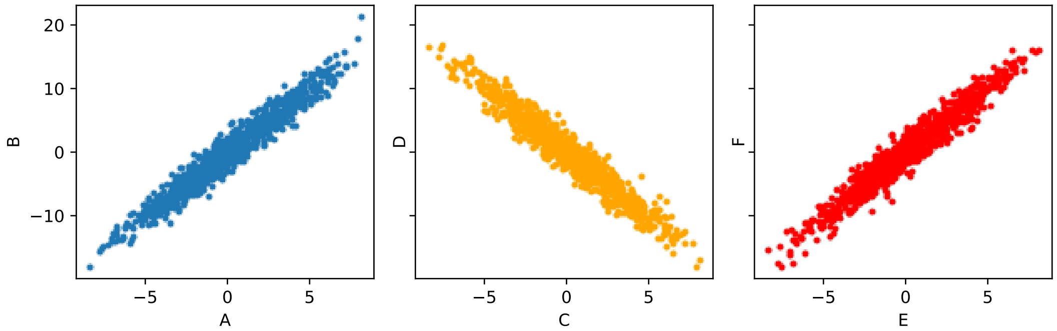


Figure 4.1: Example of a 2D distribution where the pairwise matching would not be accurate enough. Pairwise tests would tell us that A matches C and E; and that B matches D and F. However, A, B does not match C, D.

Therefore, a solution to our research question must take multidimensionality into account.

To bridge this gap, we propose the concept of Equally-Distributed Dependencies (EDDs), which is inspired by the idea of Inclusion Dependencies (INDs) from the relational algebra:

> *An inclusion dependency between column A of a relation R and column B of a relation S, written $R.A \subseteq S.B$, or $A \subseteq B$ when the relations are clear from the context, asserts that each value of A appears in B. Similarly, for two sets of columns, X and Y , we write $R.X \subseteq S.Y$, or $X \subseteq Y$, when each distinct combination of values in X appears in Y [AGN15]*

The definition of IND is based on set theory, which is not directly applicable to numeric data where measures are in the real domain (e.g., spatial coordinates or flux measurements) and usually have an associated uncertainty that may or may not be explicitly stored.

However, this definition can be naturally reformulated in terms of equality of distribution $X \overset{d}{=} Y : F_X(x) = F_Y(x)\ \forall x$, where F_X and F_Y are the cumulative distribution functions of X and Y, respectively:

Definition 1 *An equally-distributed dependency between a set of columns X from of relation R and a set of columns Y of relation S, written $R.X \overset{d}{=} S.Y$ or $X \overset{d}{=} Y$ asserts that the values of X and Y follow the same probability distribution.*

The term *arity* refers to the cardinality of the sets of attributes X and Y. For instance, if $|X| = 1$, we talk about unary EDDs; if $|X| = 2$, binary or 2-EDDs; and, in general, for $|X| = n$, n-ary EDDs.

Finding high arity INDs is an NP-hard problem [Kan92]. For instance, for two sets of n attributes in R and S, there are $n!$ different possible permutations to check. In comparison, finding unary INDs seems a relatively simple problem, as the worst case has complexity $O(n^2)$. Nonetheless, testing over real files may require expensive input/output operations. Furthermore, as we will see later, false positives at this stage can quickly make finding high arity INDs unfeasible. This is because the search space tends to grow exponentially with the number of one-attribute matches, making unary INDs search time much less important than reducing the number of false positives.

We used a published experimental evaluation of IND finding techniques [Dür19] as a starting point for assessing how adequate existing solutions are for our problem. The authors carried out a set of experiments with thirteen IND algorithms, of which seven are for unary INDs, four for n-ary INDs, and two for both types. A more recent survey confirms that this work contains the current state-of-the-art for Inclusion Dependencies [KPN22]. It is worth mentioning that this evaluation is based on Metanome, the library used by Kayak to profile the datasets.

We now briefly describe the n-ary finding algorithms evaluated by the authors and discuss their suitability for our needs.

4.3.1 n-IND finding algorithms

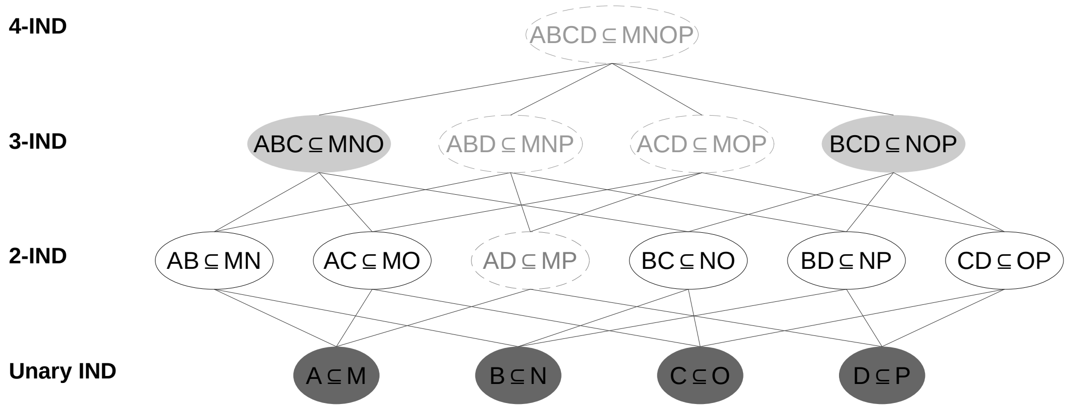


Figure 4.2: Example structure of the search space as a lattice for an initial set of 4 unary INDs. As an illustration, if the 2-INDs surrounded by a solid line were valid, a bottom-up traversal would only need to check the validity of the 3-INDs with a gray background since the others could not be valid.

Given two relations R and S, with attributes A and B respectively, a unary Inclusion Dependency (uIND) exists if $R.A \subseteq S.B$. More generally, for two sets of attributes X and Y, both of cardinality n, an n-ary Inclusion Dependency (nIND) exists if every combination of values in X appears in Y [DLP02; AGN15].

Given a set U of valid uINDs, the search space for higher-arity candidates is defined by its power set and a partial order relation called specialization [DLP02]:

Definition 2 *Let $I_1 = R[X] \subseteq S[Y]$ and $I_2 = R'[X'] \subseteq S'[Y']$. I_1* ***specializes*** *I_2 (denoted $I_1 \prec I_2$) iff*

1. *$R = R'$ and $S = S'$.*

2. *X and Y are sub-sequences of X' and Y', respectively.*

We can also say that I_2 ***generalizes*** *I_1.*

Example 1 $(R[AB] \subseteq S[EF]) \prec (R[ABC] \subseteq S[EFG])$*. However,* $R[AB] \subseteq S[DE]) \not\prec (R[ACD] \subseteq S[DFG])$

This partial order enables us to structure the search space as a lattice, as exemplified in figure 4.2. Most solutions leverage this property to explore the search space bottom-up —from level k to k+1— or top-down —from level k to k–1— order.

Mind [DLP02] is a bottom-up approach: it starts from a set of known, satisfied unary INDs and builds higher arity candidates combining them. These new candidates are then validated against the database, and those satisfied are used for computing the next-level candidates until no more candidates are available.

Zigzag [DP03] starts with a Mind bottom-up approach up to a given arity $n \geq 2$. Then, it uses all satisfied INDs to initialize a *positive border* and the non-satisfied to initialize a *negative border*. The set of satisfied INDs is used to generate the set of candidates with the highest arity possible, called *optimistic border*, which is then validated against the database. This is the bottom-up part of the search. Valid candidates are directly added to the positive border. Invalid candidates are treated depending on how many tuples are different between relations. Those above a given threshold (too many different tuples) are added to the negative border. Those below are top-down traversed, from level n to n–1, validated, and then added to the positive border if they are satisfied. The algorithm then iterates, building a new optimistic border until it is impossible to generate new INDs. The optimistic approach can prune the search space aggressively when there are high-arity INDs, but when most arities are low, MIND may perform better.

FIND2 [KR03] is based on the equivalence between finding n-INDs and finding cliques on n-uniform hypergraphs (a generalization of the concept of a graph where each edge connects n nodes). Each unary IND corresponds to a node, and an n-IND corresponds to an edge on an n-uniform hypergraph. Once such a graph is built, each IND corresponds to a clique and maximal INDs correspond to maximal cliques. They present the HYPERCLIQUE algorithm, capable of finding maximal cliques performantly, which can be mapped back to candidate maximal INDs. These are finally validated using database queries. As ZIGZAG, FIND2 starts with a bottom-up approach to look for maximal cliques (i.e., potential maximal INDs). The invalid ones are used to generate a new $(n+1)$ uniform graph. This is a stage that corresponds to the top-down traversal.

While these three algorithms were evaluated on INDs between relational datasets and with attributes that can be directly compared (i.e., from discrete domains), their traversal of the search space and their validation steps are well decoupled. They can be easily adapted to the equality-of-distribution statistical tests.

Furthermore, the reference benchmark shows that MIND, FIND2 and ZIGZAG have a comparable run-time, sometimes even faster than the alternatives. While FAIDA is generally faster, its validation strategy requires computing hashes over the attributes and their combinations, which is inapplicable for continuous data that can very possibly have an associated uncertainty.

From the three suitable candidates, MIND's bottom-up approach can be performant enough for relatively low arity IND relations. However, it has one substantial disadvantage: it requires an exponential number of tests, prohibitive for higher arity INDs. Both ZIGZAG and FIND2 overcome this limitation by alternating between *optimistic* (top-down) and *pessimistic* (bottom-up) traversals. Finally, FIND2 maps the search of INDs to the search of maximal cliques. We know that using statistical tests will introduce unavoidable false negatives, which would translate into missing edges. A clique with missing edges is a quasi-clique, and finding quasi-cliques, while at least as hard as finding cliques, is doable.

4.3.2 Foreign Key Discovery

We briefly survey this area since we consider it complementary to IND discovery. A Foreign-Key (FK) constraint on an attribute A over a Primary-Key (PK) B implies that all values present on A must also be present on B. Therefore, there exists an inclusion dependency between A and B. However, the reverse is not necessarily true. For instance, two auto-increment attributes from two different relations may have an accidental IND with no semantic meaning.

To distinguish between accidental and meaningful INDs, Rostin *et al.* [Ros09] propose to train machine learning models over a set of features extracted from positive PK/FK relations and negative, non-meaningful INDs. However, their proposal is limited to unary INDs.

Zhang *et al.* [Zha10] present an algorithm capable of handling multi-column PK/FK relations. They define the concept of *Randomness Test*, which assumes that an FK is a representative sample of a PK and, therefore, should follow a similar distribution. They use an approximation of the Earth-Mover Distance (EMD) —the cost of transforming one distribution into another— to measure the similarity between PK and FK. Their algorithm ranks PK/FK candidates by distance —closest first— and selects the top $X\%$, where X must be chosen to balance precision and recall.

More recently, Jian *et al.* [JN20] introduced an approach that identifies both PK and FK holistically. They validate Zhang's concept of *Randomness* and propose a simplified estimator that treats each attribute separately. They do not need the PKs to be known but require a list of INDs as input.

It is worth noting that even though the latter two publications use the idea of the FK being a random sample of the PK, their methods use the *distance* between distributions for ranking candidates [Zha10] or as a feature [JN20]. Our method is based, however, on statistical hypothesis testing[1].

[1] While the EMD could be used as a test statistic, it would be computationally expensive [HMS21].

4.3.3 Complementarity

ReDiscover [Ala16] uses machine learning techniques, such as Support Vector Machines (SVMs), to identify matching columns between scientific tabular data. The author defines different ways in which datasets may be related: containment, partial containment, augmentation, completion, equality, . . . This solution focus does not only focus on correspondences (i.e., A is a subset of B), but it also pays attention to the semantics of the *relationship* (i.e., A is a selection of B). Yet, this system focuses mainly on the correspondence between *individual columns*, which is insufficient for spatial and coincidence associations, as they are multidimensional. We are left only with a set of pairwise correspondences that may not be enough to cross-match tuples between files.

4.3.4 Schema Homogenization

For completeness, table 4.3 classifies these solutions under the proposed *Schema Homogenization* category, expanding on the result from the literature mapping shown in chapter 3.

Middleware			
Schema Homogenization	IND [DLP02; DP03; Koe02]	FK [Ros09; Zha10; JN20]	Complementarity [Ala16]

Table 4.3: Expansion of Idreos [IPC15] classification with *Schema Homogenization.*

4.4 Proposed solution

The algorithms described in section 4.3.1 require discrete data, and they often need to be able to answer definitely whether a given IND is satisfied — i.e., Mind, Zigzag, Find2. While there have been proposals to enhance Find2 with approximate, heuristic methods, the data is still expected to be discrete, and the probabilistic aspect of the validation is decoupled from the inference of new INDs from known INDs. In the following chapter, we propose a novel algorithm, PresQ, that can recover multidimensional correspondences between datasets with different, undocumented — or unknown — schemas. PresQ relies on statistical tests to accept or discard these correspondences and takes into account the uncertainty when inferring new INDs, generalizing the problem of IND finding, and making it applicable to new domains.

To complement this method, chapter 6 describes a multidimensional statistical test based on SOM offering additional interpretability. It can be used with PresQ for studying with more details accepted or rejected correspondences, or afterward over the merged dataset for clustering or as a pre-processing for nearest-neighbor coincidence search [SD11].

Appendix B briefly describes the preliminary ideas that were discarded during the phase of *Embodiment Design*.

Chapter 5

Discovery of Multidimensional Dependencies via Quasi-Cliques on Hypergraphs

Most of the content in this chapter appears in the article PresQ*: Discovery of Multidimensional Equally-Distributed Dependencies via Quasi-Cliques on Hypergraphs* [APD22], published on *IEEE Transactions on Emerging Topics in Computing.*

PresQ is a statistically robust algorithm for finding EDDs, as described in definition 1.

Following the *Engineering Design* process, PresQ is the *output* of the *embodiment design* phase of the design process, and the present chapter is its documentation. In section 5.1, we list the set of rules needed to be able to define a search space for EDDs and introduce the concepts of hypergraph and quasi-clique. In section 5.2, we propose a novel algorithm based on quasi-cliques to infer common equally-distributed multidimensional attributes. In section 5.3, we show experimental results that prove that PresQ successfully finds dependencies in a reasonable amount of time. Finally, in section 5.4, we compile the conclusions and propose areas for further work.

5.1 Definitions

5.1.1 Equally Distributed Dependencies

An Inclusion Dependency exists if all combinations of values from a given set of attributes in one relation are contained within a set of attributes from another. However, we will hardly ever find a strict subset relation between two attributes in the real domain. Measurements may have associated uncertainty, and even floating-point representation may vary (i.e. 32 vs 64 bits). It is generally a flawed idea to compare floating-point numbers with strict equality.

Instead, we can use $R.X \overset{d}{=} S.Y$ as an approximation, meaning that the two sets of attributes are equally distributed. This relation is, unlike the subset relation, symmetrical.

Following the parallelism with IND finding, we say that the dataset d satisfies the relation defined by equality of distribution $\overset{d}{=}$ when a statistical test fails to reject the null hypothesis

$$H_0 : P(R[X]) = P(S[Y]) \tag{5.1}$$

Three inference rules can be used to derive some additional INDs from an already known set of INDs. They are defined using sets and subsets [CFP84], but they translate to the equality of distribution:

Reflexivity

$R[X] \overset{d}{=} R[X]$

Permutation and projection

If $R[A_1, \ldots, A_n] \overset{d}{=} S[B_1, \ldots, B_n]$ then $R[A_{i_1}, \ldots, A_{i_m}] \overset{d}{=} S[B_{i_1}, \ldots, B_{i_m}]$ for each sequence $i_1, \ldots, i_m$ of distinct integers from $\{1, \ldots, n\}$

Transitivity

$R[X] \overset{d}{=} S[Y] \wedge S[Y] \overset{d}{=} T[Z] \implies R[X] \overset{d}{=} T[Z]$

The reflexivity, permutation, and transitivity rules are well-known to hold for $\overset{d}{=}$ [RW79].

For the projection, we need to prove that if two sets of random variables $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$ are equally distributed, so are any of their possible sub-sequences.

Proof 1 *Let X' and Y' be the sequences $X_1, \ldots, X_m$ and $Y_1, \ldots, Y_m$ with $m < n$. Their corresponding Cumulative Distribution Function (CDF) are just the* marginal *CDF:*

$$\begin{aligned} F_{X_1,\ldots,X_m}(x_1,\ldots,x_m) =& F_{X_1,\ldots,X_m,X_{m+1},X_n}(x_1,\ldots,x_m,x_{m+1},\ldots,x_n) \\ F_{Y_1,\ldots,Y_m}(y_1,\ldots,y_m) =& F_{Y_1,\ldots,Y_m,Y_{m+1},Y_n}(x_1,\ldots,x_m,x_{m+1},\ldots,x_n) \\ & \forall (x_1,\ldots,x_m) \in \mathbb{R}^m \textit{ and } x_i \to \infty \ \forall i > m \end{aligned} \tag{5.2}$$

By definition 1, the right hand-side of both equations must be the same. By transitivity,

$$\begin{aligned} & F_{X_1,\ldots,X_m}(x_1,\ldots,x_m) = F_{Y_1,\ldots,Y_m}(y_1,\ldots,y_m) \\ \implies & X_1,\ldots,X_m \overset{d}{=} Y_1,\ldots,Y_m \end{aligned} \tag{5.3}$$

Thanks to the validity of these rules, particularly the permutation and projection, we can use the specialization relation seen in definition 2 when dealing with distributions.

With these rules, we have defined the search space similar to the one from IND discovery. The last requirement is a property that allows the pruning of the search space as illustrated in figure 4.2.

Let $I = R[X] \overset{d}{=} S[Y]$. A dataset d *satisfies* I iff a statistical test fails to reject $H_0 : P(R[X]) = P(S[Y])$ given a significance level α. This is denoted as $d \models I$.

Property 1 *Given $I_1 \prec I_2$:*

1. *If $d \models I_2$, then $d \models I_1$ (Accepting H_{0_2} implies accepting H_{0_1}[1])*

2. *$d \not\models I_1$ with a probability α when $d \models I_2$ (Rejecting H_{0_1}* does not *imply the rejection of H_{0_2})*

This property is similar to that proposed for INDs [DLP02], with the exception that even if $d \models I_2$, there is a probability to falsely reject I_1 bound by the significance level α.

Example 2 *If we have two sets of* 10 *attributes that are equally distributed, the number of 3-dimensional projections (specializations) that must be equally distributed will be* $\binom{10}{3} = 120$. *If we have a significance level of* $\alpha = 0.1$, *the expected number of falsely rejected 3-dimensional equalities is then* 12.

5.1.2 Uniform n-Hypergraphs and quasi-cliques

A hypergraph is a generalization of a graph where the edges may connect any number of nodes. It is defined as a pair $H = (V, E)$, with V the set of nodes and E the set of edges. An edge $e \in E$ is a set of distinct elements from V.

Definition 3 *Given the hypergraph* $H = (V, E)$, H *is a* n-hypergraph *iff all of its edges have size* n.

A clique or hyper-clique on a n-hypergraph $H = (V, E)$ is a set of nodes $V' \subseteq V$ such that every edge defined by the permutations of distinct n nodes from V' exists in E [KR03].

A quasi-clique or hyper-quasiclique (sometimes named pseudo-clique) is a generalization of a clique where a given number of edges can be missing. The exact definition can be based on the ratio of missing edges or based on the node degrees. Another option is to combine both measures [BHB07], which is our preferred method.

We generalize the definition of quasi-cliques to k-uniform hypergraphs :

Definition 4 *Given a k-uniform hypergraph* (V, E), *and two parameters* $\lambda, \gamma \in [0, 1]$ *the sub-graph* $H' = (V', E')$ *induced by a subset* $V' \subseteq V$ *is a* $(\lambda - \gamma)$ *quasi-clique iff:*

$$|E'| \geq \gamma \cdot \binom{|V'|}{k} \tag{5.4}$$

[1] Strictly speaking, *not rejecting* H_{0_2} implies that we can not reject H_{0_1}.

$$\forall v \in V' : deg_{V'}(v) \geq \lambda \cdot \binom{|V'| - 1}{k - 1} \tag{5.5}$$

Where $deg_{V'}(v)$ represents the degree of v, and E' is a subset of E such that $\forall e \in E' : e \subseteq V'$

In other words, condition 5.4 allows for some edges to be missing, while condition 5.5 enforces a lower bound on the degree of each node. Intuitively, the latter is essential to avoid quasi-cliques where most nodes are densely connected and a handful of nodes are connected only to a few.

The hyper-clique problem is a particular case when either $\lambda = 1$ or $\gamma = 1$.

5.2 Inferring common multidimensional data

The first required step to identify multidimensional EDDs is to find a set of unary EDDs, for which a naive approach would mean quadratic complexity. To reduce the complexity, we propose an algorithm based on interval trees in section 5.2.1. In section 5.2.2, we discuss the difficulties of the existing adaptable algorithms when dealing with uncertainties. Finally, in section 5.2.3, we propose a novel algorithm, based on quasi-cliques, which is more resilient to both false positives and false negatives.

5.2.1 Uni-dimensional EDDs

The first required step for any of the three algorithms is to find a set of valid unary EDDs on the datasets. i.e., attribute pairs that follow the same distribution. It can be done with the non-parametric Kolmogorov-Smirnov (KS) two-sample test [Hod58]. More formally, for a possible pair of attributes A and B from two different relations, the null hypothesis H_0 for the KS test is $A \overset{d}{=} B$. As for any statistical test, this null hypothesis is accepted or rejected with a significance level $\alpha \in [0, 1]$, which is the probability of falsely rejecting H_0 (false negative).

Consider a dataset containing the relations $R_1, R_2, \ldots, R_n$ with a total number of attributes $N = \sum_{i=1}^{n} |R_i|$. A naive approach to finding unary EDDs requires $N - 1$ statistical tests for each attribute. Since the EDD relation is symmetric ($A \overset{d}{=} B \iff B \overset{d}{=} A$) half of the tests can be avoided, bounding the total number of tests by the quadratic expression $(N \times (N-1))/2$.

We propose using an interval tree built over the complete dataset to reduce the number of tests. The building of the tree can be performed in $O(N \log(N))$ time, and each query done in $O(\log(N) + m)$, where m is the number of overlapping intervals for a given attribute. In case of N being much higher than m, which we expect to be generally the case, the number of operations can be thus reduced to $O(N \log(N) + M)$, where M is the total number of overlapping pair of attributes. Note that $M \leq (N \times (N-1))/2$, so the worst-case remains quadratic.

However, the cost of the tests themselves is almost negligible when compared to the cost of finding n-ary EDDs, which is exponential with the number of unary EDDs. Therefore, a low significance level α for finding unary EDDs will considerably increase the cost at later stages.

5.2.2 Multidimensional EDDs

Once we have a set of unary matches, we need to find which, if any, higher dimensional sets of attributes are shared between each pair of relations. As discussed in section 4.3, only three of the existing IND finding solutions are not strongly dependent on discrete types: MIND, ZIGZAG, and FIND2. However, replacing the *inclusion* tests with statistical tests affects their behavior.

MIND traverses the search space bottom-up. Thus, for two relations with a single multidimensional EDD with n attributes, every combination of k nodes from $k = 2$ to $k = n$ must be tested, as shown in equation 5.6.

$$\sum_{k=0}^{n} \binom{n}{k} = \sum_{k=0}^{n} \frac{n!}{n!(n-k)!} \tag{5.6}$$

Since statistical tests are not exact, the chances of having at least one false rejection in the validation chain increases with the maximal EDD arity, introducing discontinuities in the search space. This makes its traversal more difficult.

The search algorithm of Find2 is capable of finding maximal INDs with fewer tests. As an input, it requires a set of valid unary and n-ary IND relations. A k-uniform hypergraph $G(V, E)$ is then constructed, where the set of accepted unary INDs are mapped to the set of vertices V, and the set of accepted k-INDs to the set of edges E. Given this initial hypergraph, the authors of Find2 prove that finding higher arity INDs can be mapped to the problem of finding cliques, since all the generalized k-ary INDs *must* appear as edges.

However, this is not always true for the EDD finding problem. The statistical test will yield some false positives and some false negatives, a combination that makes it difficult for Find2 to find the true relations. Cliques will likely be *broken* due to the false rejections, and there will be spurious edges due to false positives. Higher arity EDDs do not appear as cliques in this scenario.

Finally, Zigzag can not recover well from missing EDDs. Any rejected EDD is added to the negative border and will not be considered any further. Additionally, some early experiments with Zigzag indicated that the combination of false positives and false negatives makes the algorithm run close to its worst-case complexity (factorial).

5.2.3 PresQ algorithm

As we have discussed, EDD finding does not map well to the clique-finding problem due to missing and spurious edges caused by the statistical tests. We propose instead an algorithm based on quasi-cliques as described in definition 4. This approach is better suited to the uncertainties associated to hypothesis testing.

Finding quasi-cliques *seeds*: Some initial experiments with Find2 showed that, by only modifying the validation strategy to use statistical tests, the algorithm was able to find relatively high arity EDDs regardless of the missing edges.

The modified Find2 maps the initial set of EDDs into a graph and lets the Hyperclique algorithm find the set of maximal cliques, then maps them back to EDDs and validates the inferred EDDs. Therefore, if FIND2 finds high arity EDDs is because HYPERCLIQUE finds maximal cliques close to the maximal quasi-clique. This makes sense since, generally, a quasi-clique contains smaller but denser sub-graphs [SDT18] and a clique is denser than a quasi-clique.

Therefore, we use a modified version of HYPERCLIQUE to search for quasi-clique *seeds*, accepting a candidate if it is a quasi-clique, as per the joint definition of equations 5.4 and 5.5. We combine both definitions since limiting only the number of missing edges tends to accept quasi-cliques with too many vertices.

Growing the quasi-clique *seeds*: This is similar to KERNELQC's idea [SDT18], but based on a quasi-clique enumeration algorithm. Given a quasi-clique *seed* from the first stage, candidates are *grown* following a tree-shaped, depth-first traversal [Uno10].

Let v be a node on a graph $G[V]$ with a degree lower or equal to the average degree. The density (i.e. γ) of $G[V \backslash v]$ is no less than the density of $G[V]$. In other words, if we remove from a γ-quasi-clique a node v with a degree lower than the average degree, the resulting graph is still *at least* a γ-quasi-clique. This is consistent with the observation that a quasi-clique contains denser subgraphs [SDT18].

Consequently, removing the vertex with the lowest degree means that the resulting quasi-clique is still a γ-quasi-clique. In the case of a tie, we can choose the vertex by its index (or name). This node is named $v^*(V)$.

Finally, a quasi-clique K' is considered a child of another quasi-clique K if and only if $K' \backslash K = v^*(K')$, i.e a quasi-clique K' is a child of K if it has one additional node that is the first node when sorted in ascending order by degree and index. This defines a strict parent-to-child relationship between quasi-cliques, which can be modeled and traversed like a tree.

The original algorithm [Uno10] is exclusively oriented towards γ-quasi-cliques, and this traversal would include many candidates that are not λ-quasi-cliques. To prune the search space and avoid branches that will not yield any valid quasi-clique, at each recursion step, we compute the degree that the nodes on K' should have, so that K' is a λ-quasi-clique. When adding a node, the expected minimum degree may increase. By knowing this value, we can ignore all nodes with a degree lower than the threshold *in the input graph*, as no matter how many more nodes we were to add afterward, no child candidate would satisfy the λ threshold.

This step successfully increases the number of quasi-cliques found. However, the number of maximal cliques is bound in general by an exponential expression of the form $\Omega(a^{|V|/b})$, where a, b are two constants that depend on the rank of the hypergraph [Tom81]. Since cliques are a particular case of quasi-cliques, we can expect the lower bound for the maximum number of quasi-cliques also to be exponential. Even if enumerating quasi-cliques can be done in polynomial time *per quasi-clique* [Uno10], the total run-time has a worst-case exponential complexity for dense hypergraphs. Therefore, it would be advisable to disable this stage for datasets with attributes hard to differentiate at low dimensionality or restrict it to the top-k seeds found.

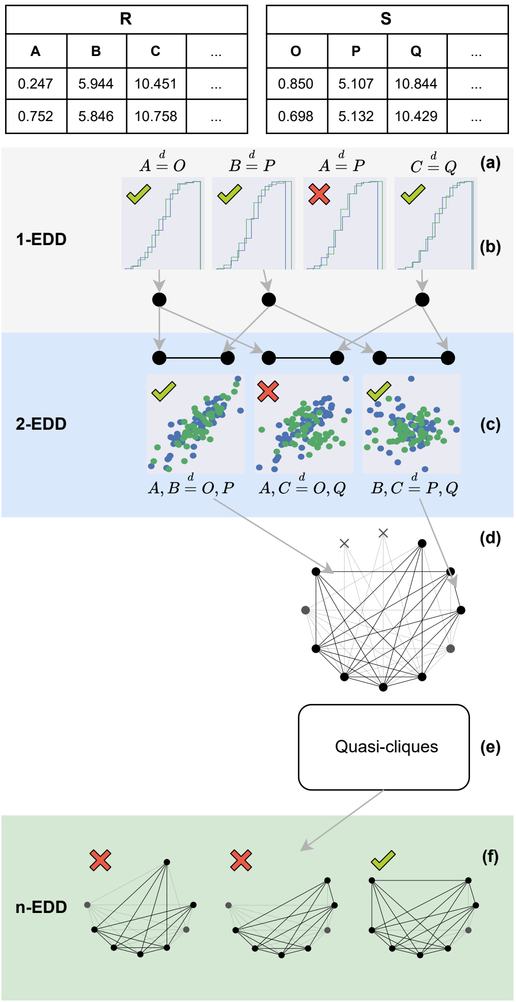


Figure 5.1: Simplified schematic of PresQ.

Given datasets R and S,

(a) Candidate 1-EDDs are found applying the interval-tree as described in section 5.2.1.

(b) Those for which the Kolmogorov-Smirnov finds a significant difference are discarded, and the rest are mapped to nodes.

(c) All pairwise combinations are tested, and those equally distributed are (d) mapped to edges on a 2-hypergraph. The algorithm works with hypergraphs of any rank (e.g., triplets mapped to edges on a 3-hypergraph).

(e) PresQ searches for quasi-cliques as described in section 5.2.3.

(f) A quasi-clique of cardinality n corresponds to an n-EDD, which is then validated by a statistical test. Those rejected are decomposed to generate the edges for a 3-hypergraph, which are verified (c), used to build a 3-hypergraph (d) and finally passed as input back to (e).

The graph above displays spurious nodes and edges (light grey, dotted) and false negatives (missing edges between dark nodes) based on attribute names. The full graph is not a valid quasi-clique because the hypergeometric test on the node degree (eq. 5.8) prunes the two nodes shown with crosses. Three candidates of arity 8 are generated given the constrain on the number of edges (eq. 5.4). Two of them are rejected by the n-dimensional statistical test and used to compute the edges of the 3-hypergraph.

Parameters

Before explaining how to tailor the parameterization of the quasi-clique finding for the purpose of searching EDDs, we need to remind that, given two sets of attributes $R[X]$ and $S[Y]$, our algorithm builds on the null hypothesis $H_0 : P(R[X]) = P(S[Y])$. In other words, it is based on the *assumption* that any EDD candidate is valid.

Let α be the significance level chosen by the user before running the algorithm. Let G be the initial k-uniform hypergraph and let K be a quasi-clique candidate. Under H_0, K represents a $|K|$-ary EDD, and by the projection rule, all possible edges between the nodes in K are also valid k-ary EDDs. If we run null hypothesis tests over these k-ary specialized EDDs, by the definition of type-I error, we can expect as many as $\alpha \times \binom{|K|}{k}$ false rejections. In other words, under H_0, we can expect a ratio of α missing edges. This is equivalent to setting the threshold for equation 5.4 as:

$$\gamma = 1 - \alpha$$

Adjusting λ is less straightforward: a high threshold will reject good candidates. A low one will accept spurious ones, triggering unnecessary tests. Even worse, the spurious quasi-cliques tend to have a high cardinality. Once rejected, they will cascade and cause an increase in lower-arity EDDs to be tested as much as $\binom{n}{k+1}$, where n is the arity of the EDD candidate, and k is the current level of the bottom-up exploration.

To solve this dilemma, we propose to use an adaptive value for λ based on the quasi-clique being checked: under H_0, there is no reason to think that any particular subset of the edges from the clique has a higher probability of having missing members. In other words, if a given node has an unexpected low degree, it is most likely connected by spurious edges.

Let N be the number of edges and n the maximum degree of the nodes on a clique with $|V'|$ nodes. Under this null hypothesis, the degree of the nodes should roughly follow a hypergeometric distribution:

$$\Pr(\text{Degree}(v) = d) = \frac{\binom{|E'|}{d}\binom{N-|E'|}{n-d}}{\binom{N}{n}}, \text{ for } v \in V' \tag{5.7}$$

This fact allows us to perform a statistical test and accept or reject our quasi-clique candidate with a given significance level. Figure 5.2 shows some examples of this distribution for a quasi-clique with 29 nodes and the critical value for a one-tail test with $\alpha = 0.05$. In other words, if the degree of a node within a quasi-clique candidate is less than the critical value, we can reject the null hypothesis and accept that the set of edges connecting the node are spurious.

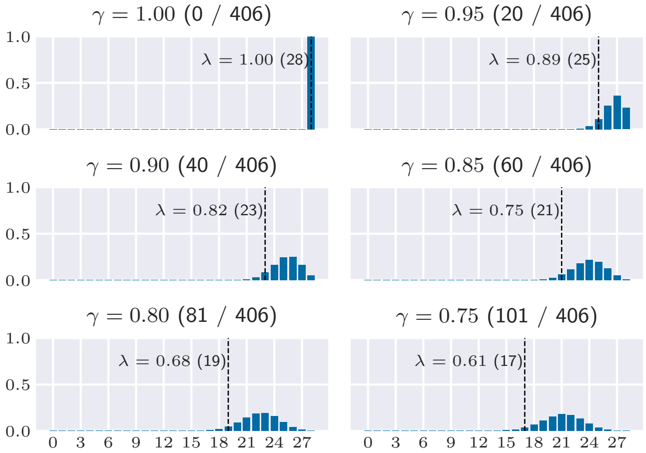


Figure 5.2: Distribution of the degree of the nodes under the null hypothesis that the missing edges on the quasi-clique are due to the expected false negative rate of the statistical test. The vertical line corresponds to the one-tail test with $\alpha = 0.05$.

In summary, as a constant number of missing edges could be considered too restrictive [BHB07], we consider a fixed ratio to be limiting as well, and harder to make sense of —i.e., why choose $\lambda = 0.6$ and not $\lambda = 0.7$?. We propose that instead, replacing equation 5.5 with equation 5.8 could be a more intuitive and flexible approach.

$$\forall v \in V' : \text{CDF}(\text{Degree}(v)) \geq \Lambda \tag{5.8}$$

Where $0 \leq \Lambda \leq 1$. As with γ and λ, a value of 1 would only accept regular cliques.

The proposed parameterization for γ and Λ are internally consistent since they are both constructed under H_0.

Figure 5.1 visually summarizes the stages of PresQ algorithm, and the effects of the parameters γ and Λ on the quasi-clique finding stage.

In the following section, we will show that adapting FIND2 clique validation with ours is enough to improve its performance in run-time and results. The *growing* step improves the efficacy (i.e. more maximal EDDs found) at the cost of a higher run-time.

5.3 Experiments

We have implemented in Python a version of FIND2 that validates candidates with statistical tests, and the proposed PRESQ. Both share most of the code, including initialization and statistical tests. Any difference in run-time is only because the modified version searches for quasi-cliques instead of full cliques.

We focus on comparing these algorithms for two main reasons: 1) To prove that quasi-clique finding can outperform clique finding both in run-time and results when the data is noisy, an advantage not necessarily exclusive to EDD finding; 2) While INDs are targeted towards inferring foreign-key relationships and generally of low arity, we expect EDDs to be of high arity —*co-located within a multidimensional space*—, and FIND2 performs well when the arity is high [Dür19].

5.3.1 Experimental design

We have performed two different sets of experiments: one exclusively benchmarks the quasi-clique search, while the other runs over real-world datasets.

(Quasi-) clique search

This experiment decouples the testing of the quasi-clique search from the uncertainty associated with the data. The test accepts as parameters the rank for the hyper-graph k, the cardinality for the clique n, the number of additional nodes N, the fraction of missing edges α and the fraction of *spurious* edges β. With these parameters, the test performs the following initialization procedure:

1. Create n nodes belonging to the clique

2. Create N additional nodes
3. Create the set E of $\binom{n+N}{k}$ edges connecting *all nodes*
4. Create the set Q of $\binom{n}{k}$ edges belonging to the clique
5. Obtain the set of all edges not belonging to the clique $C = E \setminus Q$

With these sets, and to obtain an estimation of the distribution of the target measurement, it then repeatedly generates noisy versions of the original clique through the following steps:

6. Remove $\alpha \times |Q|$ random edges from the original full clique Q
7. Add $\beta \times |C|$ random edges from C
8. Run Find2 and PresQ over the resulting graph

The parameters α and β simulate the effect of type I and type II errors respectively.

PresQ is configured with $\gamma = 1 - \alpha$ and $\Lambda = 0.05$. The number of additional nodes is fixed to half the number of nodes in the clique: $N = \frac{n}{2}$.

This experiment measures, in a controlled manner, the capability of the algorithms to find the *true* clique and how their run-time is affected by the number of missing and spurious edges. Since the inputs are randomized, some will unavoidably run with exponential complexity, the worst case for all the algorithms. To avoid spending too much time on these extreme cases, the test also accepts a timeout parameter. We describe the measurements we have taken in table 5.1, and the different parametrizations in table 5.2.

Real-world datasets

For the statistical tests, we use a non-parametric multivariate test based on k-Nearest Neighbors (kNN) [Hen88; Sch86], but any other multivariate test could be used. However, regardless of the chosen test, there will always be a number of false negatives bound by the significance level. In any case, the techniques here discussed remain relevant.

Table 5.1: Set of measurements taken for the quasi-clique finding problem.

Recovery ratio	For each quasi-clique Q' found, we compute the Jaccard index for each found quasi-clique, $J(Q, Q') = \|Q \cap Q'\| \div \|Q \cup Q'\|$. From all the obtained values, we report the maximum. A value of 1 signals a perfect match.
Time	Wall-clock time.
Timeouts	How many runs exceeded the timeout.

Table 5.2: Combination of parameters for the quasi-clique find problem.

Rank	α	β	**Timeout (s)**
2	$[0.05, 0.30]$, step 0.05	0.0	240
	0.1	$[0.0 - 0.8]$ step 0.2	
3	$[0.05, 0.30]$, step 0.05	0.0	300
	0.1	$[0.0 - 0.8]$ step 0.2	1200
4	$[0.05, 0.30]$, step 0.05	0.0	1200
	0.1	$[0.0 - 0.8]$ step 0.2	3000

The initialization stage of the test is as follows:

1. We load two separate datasets.
2. The constant columns, where every tuple has the same value —including *null*— or only a handful of different values, are dropped. FAIDA authors followed a similar procedure to reduce the number of columns to check [Kru17].
3. A random sample is taken from both relations (it defaults to 200).
4. The algorithm described in section 5.2.1 is used to find a set of valid unary EDDs.
5. *All* possible n-EDDs (for $n \in \{2, 3\}$) are generated and validated. The tests begin at different arities in order to compare the resiliency of FIND2 and PRESQ for different initial conditions.
6. Valid n-EDDs are used to create the initial graph passed as input to PRESQ.

The fifth step is performed at different significance levels of $\alpha \in \{0.05, 0.10, 0.15\}$ to verify how the number of missing and spurious edges affects the search algorithms. Typically, MIND would generate the graph (i.e. 3-EDDs are generated from valid 2-EDDs). Nonetheless, we start with all possible n-EDDs for simplicity: it is easier to model and understand how many missing edges are expected as a function of α.

The input for both search algorithms is, thus, identical at every run. However, since there is an unavoidable effect of the randomization of the sampling in step 3 and the N-dimensional permutation tests, we repeated the experiment. As a result, we are confident that the difference is significant and not due to chance.

While FIND2 has no parameters beyond the initial set of EDDs, PRESQ requires a value for both γ and Λ. As we mentioned earlier, it makes sense to bind γ to the expected number of missing edges (false negatives): $\gamma = 1 - \alpha$. For Λ, we tested with the values 0.05 and 0.1 since lower values yield too many accidental quasi-cliques, while higher values defeat the tolerance introduced by γ.

Table 5.3: Set of measurements taken from individual runs.

Time	Wall-clock time, without accounting for the initialization stage, as this is shared.
Number of tests	Time spent looking for quasi-cliques and validating the candidates. Tests can be potentially expensive, so we measure how many statistical tests are necessary.
EDD count	Without removing non-maximal EDDs.
Maximal EDD count	Removing non-maximal EDDs.
Timeouts	The execution time has a time limit of 3000 seconds. We report the percentage of runs that could not finish within the allocated time window.
Highest arity	The maximum EDD arity found.

To measure the efficacy (EDDs finding) and efficiency (run-time) of the algorithms, we took the measurements summarized in tables 5.3 and 5.4.

Given the variability and the number of dimensions, it can be hard to assess the quality of the results. As a general guideline, we consider:

- The higher the match ratio, the better: the highest arity EDD is potentially the most interesting and selective candidate for cross-matching.
- For a similar match ratio, the lower the run-time, the better.

For a similar match ratio, a higher number of maximal EDDs is desirable. Arguably not for the IND discovery —after all, a few good candidates may suffice—, but it proves the capacity of finding maximal quasi-cliques.

It is important to note that some of these measures are interdependent. For instance, if a maximal EDD with a higher arity is found, the number of EDDs should generally decrease. Conversely, if a true, high-arity candidate is rejected, multiple generalizations will be considered and possibly accepted, increasing the number of unique EDDs. Similarly, finding more maximal EDDs implies running more statistical tests, so the run-time will be worse. Ultimately, it is up to the user to decide what is more important and parameterize the algorithm accordingly.

Table 5.4: Set of measurements derived over the complete set of runs.

Match ratio It is a ratio between the maximum arity of the maximal quasi-clique found and the *true* maximum EDD possible to find on each separate run. This *truth* is solely based on attribute names. The algorithms can find higher arity EDDs when the values are taken into account. This is proof of success: the metadata would not have sufficed to capture this trait.

Accuracy Measured as the number of total returned EDDs, divided by the number of statistical tests executed. A ratio of 1 (best) means that every candidate was accepted by the statistical test, while a ratio of 0 (worst) means that all candidate quasi-cliques were rejected. This value is also affected by the power of the statistical test as a function of dimensionality.

We ran the tests disabling the limitation on the degree ($\Lambda = 0$) and the limitation on the total number of edges ($\gamma = 0$). In this manner, we can evaluate if there is any difference when using one, the other, or both.

Table 5.5: Summary of the datasets used for validation.

Dataset	**Tables**	**Rows**	**Columns**	**1-EDD**
Mortgage/Treasury	2	1k + 1k	16 + 16	26
Ailerons/Elevators	2	14k + 17k	41 + 19	44
DC2	2	198k + 193k	39 + 33	279
AFDS	4	172 × 4	8 × 4	63
Waveform	2	5k + 5k	22 + 41	145
KEEL	43	43 — 41k	444	972
ChEMBLDB	79	5 — 19M	418	599

Datasets: To test the algorithms, we ran them over two pairs of relations from the KEEL regression datasets [Alc11], the training and test catalog from the *Euclid photometric-redshift challenge* [Des20], and a set of sensor measurements from an aircraft fuel distribution system [Ghe19]. For the scalability tests, we have used the full KEEL regression dataset, two variants from the Waveform

Database Generator [DG17; Bre84], and versions 29 and 30 of the ChEMBL database [Gau16].

Some statistics about these datasets are summarized in table 5.5.

Mortgage / Treasury, from KEEL, contain the same data, permuted by rows and by columns. These datasets are an example of data de-duplication.

Ailerons / Elevators, also from KEEL, share their origin (control of an F16 aircraft) but have different sets of attributes. These datasets are an example of data fusion.

DC2 comes from a single catalog of astronomical objects split based on the sky coordinates. The authors masked some of the attributes of the training set (i.e., coordinates and the target attributes red-shift). Therefore, both catalogs share some of the attributes but from different sources. A naive one-to-one schema matching will easily mistake these attributes for small sample sizes. In contrast, for bigger samples, some true correspondences will be falsely rejected. These datasets require some more resilient methods capable of working on a multidimensional space. These datasets are an example of schema inference/matching and automatic feature discovery.

Aircraft Fuel Distribution System (AFDS) comprises five different files, all sharing the same schema but containing sensor measurement values for different scenarios: one nominal, and four abnormal. Our implementations of Find2 and PresQ can process the five files at the same time.

Waveform Database Generator We use version 1, with 21 attributes, and version 2, which shares the same 21 attributes and adds 19 extra features that are just Gaussian noise. This 21-ary EDD between the datasets goes beyond the maximum 7-ary evaluated in previous works [Dür19]. Additionally, the number of attributes and their distribution similarity generates many false positives at low dimensionality, stressing the capability of processing noisy, dense, graphs.

ChEMBL Database We use versions 29 and 30 of the ChEMBL database, each of size 20GiB. They are stored on BeeGFS, a clustered filesystem. We evaluate the scalability with respect to the number of columns, adding tables progressively. In this scenario, the overhead introduced by the sampling becomes significant.

The two pairs from KEEL (i.e. Mortgage/Treasury and Ailerons/Elevators) were found running over the whole KEEL dataset initial versions of the algorithms described in this paper, proving their capabilities. We report the performance of this exercise, together with the other two scalability tests, in section 5.3.3.

5.3.2 Environment

The tests were run on a cluster, where each node is fitted with an Intel(R) Xeon(R) Gold 6240 CPU at 2.60GHz with 36 virtual cores, running on a standard CentOS Linux 7.9. The default memory allocation per core was 3 GB.

For the (quasi-)clique search, we submitted one job with as many tasks as parameter combinations described in table 5.2 and 1 CPU per task, for cliques of size $10, 20$, and 30. We chose the time limit based on the measured run-time from early test runs.

For the real dataset tests, we submitted jobs with 8 tasks and 1 CPU per task, limited to 24 hours. The objective of concurrent runs was to increase the number of data points since the code was not parallelized.

Finally, we executed ten randomized runs for each increment on the number of columns for the scalability tests.

5.3.3 Results

In this section, we summarize the results of our test setup.

(Quasi-) clique search

We summarize the *wall-time* and *recovery ratio* metrics by estimating their distribution mean and its associated standard error following the Bootstrap method. The *timeout* is measured by counting how many runs fail to find a quasi-clique within the allocated time window.

While the *wall-time* distribution is far from Gaussian, we consider that randomizing the input, pruning the long-running cases, and averaging the results of a few short-running iterations is a valid usage of the algorithms. This makes comparing the means a reasonable assessment.

Influence of spurious edges: We show in figure 5.3 the performance of the algorithms for 3-hypergraphs and different ratios of spurious edges. The exponential worst-case complexity becomes more apparent the more connected nodes there are. FIND2 is the most affected, but at some point, PRESQ performance also degrades significantly and eventually also fails to finish on time. These results confirm that spurious edges influence the *run-time* of these algorithms very negatively [KR06].

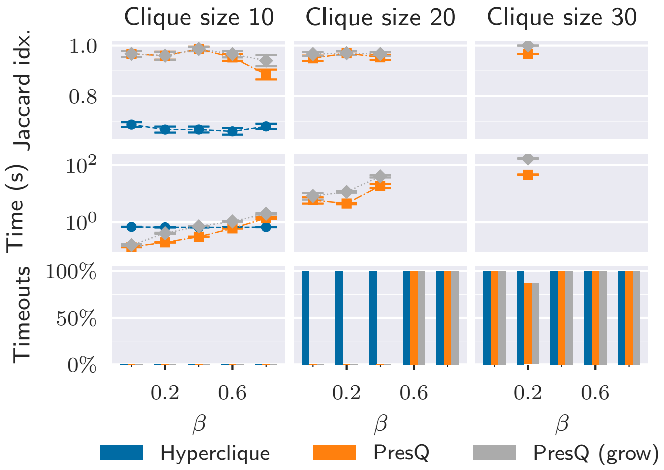


Figure 5.3: Recovery ratio and run-times for cliques on uniform 3-hypergraphs for different ratios of spurious edges (β). Each data point corresponds to 15 runs.

Influence of missing edges: Figure 5.4 shows that our proposal generalizes for hypergraphs. PRESQ with the growing stage enabled, oscillates very close to the original clique even when 30% of the edges are missing. However, the number of timeouts increases given that the algorithm needs to traverse more levels from the seed to the maximal quasi-clique. Interestingly, there is an inverse correlation between the number of missing edges and run time.

Influence of correlated ratios: In a more realistic scenario — i.e., when using statistical tests — as the number of missing edges increases, the number of spurious edges should decrease. We have run tests with the growing stage enabled for different parametrizations on the node degree threshold. This in-

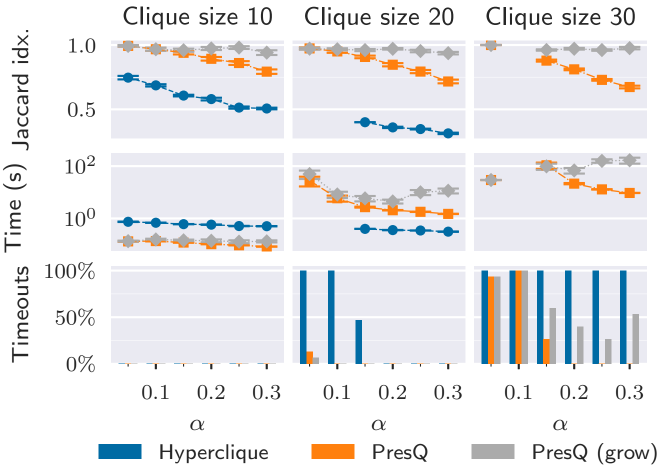


Figure 5.4: Recovery ratio and run-times for cliques on uniform 3-hypergraphs for different ratios of missing edges (α). Each data point corresponds to 15 runs.

cludes a regular λ parameter with a value of 0.8 chosen based on good empirical results we obtained during early iterations of this work. The correlation between β and α is based on the empirical statistical power of the kNN test as a location test on k dimensions and a sample size of 100. In all cases, $\gamma = 1 - \alpha$.

Figure 5.5 summarizes the results. A hand-picked parameter of $\lambda = 0.8$ can perform well for some hypergraphs but quickly underperforms as the hypergraphs become noisier. On the contrary, our proposal based on the hypergeometric distribution remains stable. However, disabling the degree limitation performs better for this particular setup. This makes sense since there is no correlation between missing edges.

Real-world datasets

The initial randomized state heavily influences the proposed performance measurements. Their distribution can not be assumed normal. Purely comparing their means is not enough to assess the validity of our proposal, and we also need an estimation of variability.

The metric of choice used to compare our measurements is the *percent difference* between sample means, being its sample estimator [CT19]:

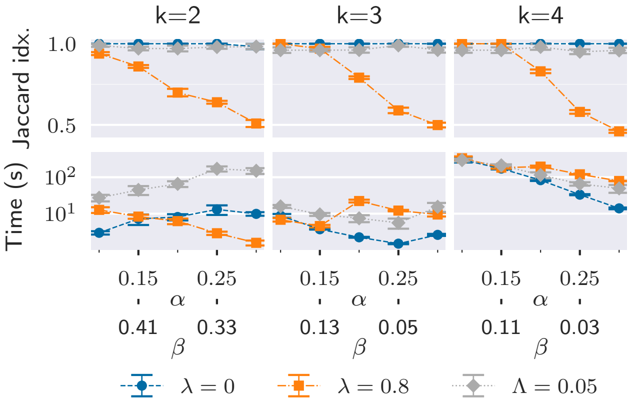


Figure 5.5: Recovery ratio and run-times for cliques of size 20 on uniform $(2, 3, 4)$-hypergraphs. In this setup, there were no timeouts. Each data point corresponds to 5 runs.

$$\hat{\phi} = \frac{\hat{\mu}_{\text{PresQ}} - \hat{\mu}_{\text{Find2}}}{\hat{\mu}_{\text{Find2}}} \tag{5.9}$$

The distribution of $\hat{\phi}$ can be estimated using bootstrapping. In this manner, we obtain the estimated population mean and standard deviation. Finally, we compute the 95% confidence interval $\hat{\mu_\phi} \pm 1.96\hat{\sigma_\phi}$

Figure 5.6 shows this confidence interval for match ratio, unique EDDs, number of tests and wall time (columns) for a significance level of 0.10, against the different datasets (rows).

The DC2 case is particularly interesting. The attributes of the datasets are relatively numerous —compared to the others— and very similar in their distributions. A low initial significance level will generate very dense graphs, with a few missing edges, and many spurious, which impacts the performance considerably. This is a known issue of Find2 [KR06]. Increasing the significance level reduces the number of spurious edges at the cost of missing true ones. Consequently, the efficiency is improved at the cost of the efficacy. PresQ allows us to increase the significance level without sacrificing much efficacy.

For the AFDS dataset, when comparing the maximum EDD arity found per pair of files, scenarios two and three are the most similar, as seen in figure 5.7. We can obtain this insight without even knowing what the schema nor the content of the files are. After seeing this result, we checked the original paper from

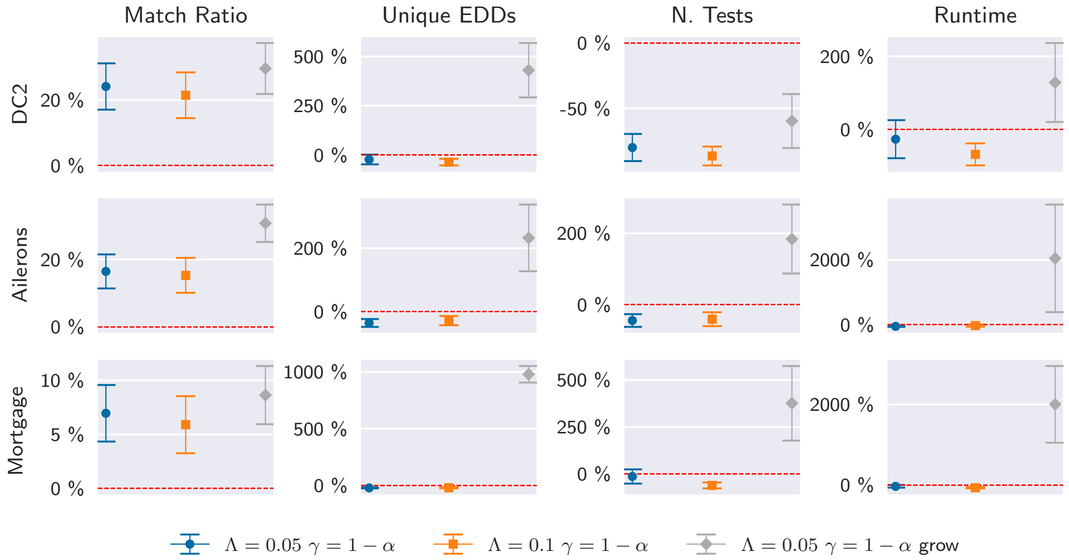


Figure 5.6: 95% confidence intervals for the percent difference (equation 5.9) between FIND2 and three parameterizations of PRESQ for the DC2, Ailerons vs. Elevator, and Mortgage vs. Treasury datasets. Intervals that do not intersect the horizontal dashed line at 0% show a statistically significant result. For ratio, higher is better. For tests and run-time, lower is better. Unique is harder to assess since the results also depend on the statistical power of the chosen test. Since the growing stage can generate many candidates, a low-powered test will accept many false EDDs.

where the dataset was obtained, verifying that, indeed, they are "two closely related scenarios" [Ghe19]. We consider this another proof of the utility of the proposed techniques.

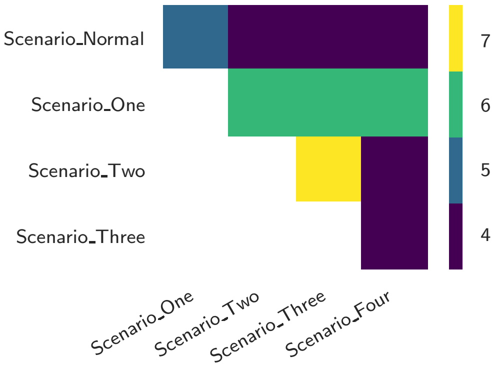


Figure 5.7: Pairwise max arity found on the AFDS dataset for each pair of scenarios.

Table 5.6 (page 77) summarizes the overall results when we execute our tests over the datasets *Mortage vs Treasury*, *Ailerons vs Elevators* and *DC2* for different values of γ and Λ — note that FIND2 is equivalent to either of the two parameters set to 1.0. For run-time, match ratio, and the number of unique EDDs, we provide the first and third quartiles. The *precision* column shows how many candidates are accepted by the statistical test. A value of 1 means that all candidates were valid EDDs.

When the search algorithm looks for cliques (the first entry for each dataset), the precision is high since almost all candidates were accepted. However, these candidates are, on average, of lower arity. This is visible on the *Match* columns. As the potential maximal arity becomes higher —e.g., *DC2*— the chance of having missing edges increases, thus making the search more resource intensive.

On the other hand, in a too-permissive setup where only γ constrains the quasi-cliques (second entry), the algorithm is too eager and accepts EDD candidates later rejected either by the statistical test or by the limitation of not accepting duplicated columns. The precision is low, and the search time increases as well.

Our proposed Λ parameter, based on the *expectation* on the number of missing edges, is more effective at constraining the set of candidates even when used alone (third entry). The precision increases and the run time is reduced. When combined with γ (fourth and fifth entries), the precision increases and fewer tests are required.

As an illustration of this balance, let us examine in more detail the consequences of the different Λ parameterization following the process shown in figure 5.1 when running over the DC2 dataset. The first four stages are unaffected by this parameter:

(a) As described in section 5.2.1, an interval tree is built over the attributes from both relations. Only overlapping ranges are compared, reducing by 27% the number of tests required.

(b) 810 KS tests need to be done. 49 pairwise combinations are considered equally distributed (Unary Equally-Distributed Dependencies (uEDDs)).

(c) $(n \times (n-1)) \div 2 = 1176$ edges are build combining all uEDDs and validated using the kNN test. 612 edges are considered valid.

(d) The initial graph has half as many edges as the complete graph. Since we know the ground truth, we can extract the sub-graph induced by the set of true uEDD and find the number of missing edges to be ≈ 0.10 on average, as we expected.

The following table exemplifies the consequences of different values of Λ — see equation 5.8 — on the count and size of the found quasi-cliques (e) and the number validated by the kNN test (f). Those invalid are 'decomposed' into candidate 3-EDDs, validated, and used to build a 3-hypergraph (d) feedback to stage (e) for the next iteration.

	Quasicliques	**Valid**	**Median size**
$\Lambda = 0.00$	2385	292	19
$\Lambda = 0.05$	107	64	12
$\Lambda = 1.00$[2]	53,053	52,291	6

For $\Lambda = 0$, the search algorithm is too greedy and accepts quasi-cliques that are poor candidates. Too many are invalid and need to be feedback to the

[2] Equivalent to clique finding

algorithm, increasing run-time. For $\Lambda = 1$, the search algorithm is too restrictive. Its precision is high, but it spends a long time enumerating small cliques. $\Lambda = 0.05$ provides the right balance, improving the result and performance.

Finally, the growing stage increases the number of candidates of all arities. This requires a more exhaustive traversal of the search space and the execution of more tests, increasing the total run-time. While we run the growing stage over *all* found seeds, this stage could be restricted only to a subset of the most interesting *seeds* —e.g., highest cardinality.

Table 5.6: Summary of run-time, matching ratio (based on name), and number of maximal quasi-cliques found accepted by the statistical test. The significance level is $\alpha = 0.1$. N corresponds to the number of randomized runs. PresQ(G) identifies PresQ with the growing stage.

	Λ	γ	**Time (s)**		**Match**		**Unique**		**Prec.**	**N**	**Timeouts**
			Q1	*Q3*	*Q1*	*Q3*	*Q1*	*Q3*			
					Mortgage vs Treasury						
Find2			0.44	0.64	0.75	1.00	12	21	0.99	527	0.0%
PresQ	**0.00**	0.9	44.86	459.04	0.94	1.00	11	15	0.06	212	0.0%
PresQ	0.05	**0.0**	0.71	11.08	0.88	1.00	11	17	0.49	535	0.0%
PresQ	0.05	0.9	0.76	10.61	0.88	1.00	11	17	0.57	507	0.0%
PresQ	0.10	0.9	0.73	1.99	0.84	1.00	11	17	0.75	503	0.0%
PresQ(G)	0.05	0.9	47.10	247.39	0.88	1.00	125	262	0.22	503	0.0%
					Ailerons vs Elevators						
Find2			5.63	48.41	0.78	1.00	142	291	0.98	93	0.0%
PresQ	**0.00**	0.9	8.82	36.75	1.00	1.22	88	174	0.24	128	0.0%
PresQ	0.05	**0.0**	22.68	52.87	1.00	1.22	113	239	0.16	126	0.0%
PresQ	0.05	0.9	7.51	20.12	1.00	1.11	86	198	0.35	60	0.0%
PresQ	0.10	0.9	6.83	22.40	1.00	1.11	89	205	0.41	60	0.0%
PresQ(G)	0.05	0.9	57.86	674.26	1.11	1.25	321	1062	0.22	60	0.0%
					DC2						
Find2			74.94	805.71	0.60	0.71	73	150	0.90	53	34.0%
PresQ	**0.00**	0.9	681.51	1536.19	0.68	0.69	102	200	0.01	16	87.5%
PresQ	0.05	**0.0**	40.07	189.45	0.80	0.93	46	115	0.10	21	47.6%
PresQ	0.05	0.9	25.57	214.27	0.76	0.89	46	113	0.14	53	13.2%
PresQ	0.10	0.9	18.61	144.98	0.76	0.87	42	98	0.18	52	23.1%
PresQ(G)	0.05	0.9	458.26	1881.02	0.81	0.93	518	798	0.23	52	50.0%

Table 5.7: Summary of run-time, matching ratio (based on name), and number of maximal quasi-cliques found. Significance level is $\alpha = 0.1$, initial arity $k = 3$.

Mortgage vs Treasury

	Λ	γ	**Time (s)**		**Match**		**Unique**		**Prec.**	**N**	**Timeouts**
			Q1	*Q3*	*Q1*	*Q3*	*Q1*	*Q3*			
Find2			8.75	20.54	0.75	1.00	86	144	1.00	160	16.3%
PresQ	**0.00**	0.9	55.00	569.01	1.00	1.00	67	102	0.17	160	0.0%
PresQ	0.05	0.9	8.43	19.45	0.81	1.00	82	126	0.99	160	0.0%
PresQ	0.10	0.9	9.04	19.60	0.81	1.00	83	130	0.99	160	0.0%
PresQ(G)	**0.00**	0.9	6.71	902.57	0.87	1.00	98	264	0.29	160	47.5%
PresQ(G)	0.05	0.9	36.83	358.53	0.86	1.00	204	395	0.98	160	0.6%
PresQ(G)	0.10	0.9	43.23	287.59	0.81	1.00	198	376	0.98	160	0.0%
Ailerons vs Elevators											
Find2			24.66	1072.74	0.89	1.00	474	947	1.00	114	40.4%
PresQ	**0.00**	0.9	28.55	140.93	1.13	1.33	276	627	0.45	114	1.8%
PresQ	0.05	0.9	26.59	83.72	1.11	1.22	339	656	0.95	114	14.0%
PresQ	0.10	0.9	25.55	105.93	1.00	1.14	357	680	0.96	110	14.6%
PresQ(G)	**0.00**	0.9	171.81	940.15	1.19	1.33	498	1044	0.34	114	9.7%
PresQ(G)	0.05	0.9	174.88	670.84	1.13	1.29	629	1330	0.95	111	18.9%
PresQ(G)	0.10	0.9	205.63	718.51	1.12	1.29	661	1245	0.95	109	19.3%
DC2											
Find2			1050.56	1050.56	1.00	1.00	560	560	0.97	83	98.8%
PresQ	**0.00**	0.9	599.20	599.20	0.88	0.88	830	830	0.06	32	96.9%
PresQ	0.05	0.9	207.28	2013.45	0.81	0.89	351	747	0.56	81	88.9%
PresQ	0.10	0.9	71.85	926.94	0.80	0.93	380	791	0.72	78	93.6%
PresQ(G)	**0.00**	0.9								32	100.0%
PresQ(G)	0.05	0.9	340.72	599.10	1.02	1.21	775	888	1.00	80	97.5%
PresQ(G)	0.10	0.9	413.58	670.53	1.00	1.13	808	947	1.00	76	97.4%

Table 5.7 (page 78) summarizes the performance measures when FIND2 and PRESQ are run over an initial 3-hypergraph. The precision is considerably higher than when starting on a 2-hypergraph. This is due to the higher power of the statistical test at dimension 3, so fewer spurious edges are introduced. However, the overall run-time suffers because the number of edges on a hypergraph is $\binom{|V|}{k}$ where $|V|$ is the number of nodes and k the rank of the hypergraph. Therefore, for a fixed number of nodes, the number of edges is generally higher for hypergraphs of higher rank.

Scalability tests

For measuring the scalability of our algorithm, we executed the algorithm over the KEEL, Waveform, and ChEMBL datasets, progressively adding columns, measuring run-time; the number of 1, 2, and n-EDDs; and the number of unary tests saved by the interval tree. In all cases the chosen parameterization is $\alpha = 0.1$, $\Lambda = 0.05$, $\gamma = 1 - \alpha$ and 200 samples. We set the run-time limit at 3000 seconds. The relations and their attributes are consistently added in alphanumeric order. The accepted 1-EDDs are used to compute all the possible 2-EDDs, while the accepted 2-EDDs define the initial edges for the n-EDD finding. Figure 5.8 (page 81) summarizes our results.

The *KEEL* dataset contains 43 different relations. The interval tree saves around 45% of the tests since many columns do not overlap. The number of EDDs increments in 'bursts' when a relation that matches a previous one enters the pool. This is due to the existence of high arity EDDs (16, 12, 7, and 6). The high number of 2-EDDs makes the growing stage eventually impractical.

For the *ChEMBL* databases, we have used a naive random sampling that only requires a single pass over the entire database. Even then, the reading time is little with respect to the rest of the EDD finding algorithm. The number of 1-EDDs steadily increments as relations are added, but 2 and n-EDDs remain relatively stable —the corresponding error bars overlap—, and so does the run-time. The arities are lower than those from KEEL, the maximum being 6 for the tables `molecule_dictionary` and `compound_properties`. The interval tree saves 57.9% of tests for 1-EDDs.

Finally, while the *Waveform Generator* datasets are the smallest in size, it is the dataset for which the algorithm shows the worst performance. This is due to the maximum arity possible (up to 20) and because their attributes are hard to distinguish — the interval tree can not save even one test. The number of 2-EDDs grows super-linearly with respect to the number of attributes, resulting in a very dense and noisy initial graph with many possible quasi-cliques.

From these experiments, we can conclude that the algorithm scales well with respect to the number of relations and columns and that the sampling has a low impact even for big datasets. However, when the statistical test has low power for the input data, the run-time significantly degrades even for moderate input sizes since the search space is combinatorial and little pruning is possible. In this case, the user can choose a different test or increase the sample size to maximize the power. Figure 5.9 exemplifies the effect the sample size has on the result set and, therefore, run-time for the complete Waveform dataset. The power of the test is low when considering only a few attributes, and the initial graph becomes rather dense.

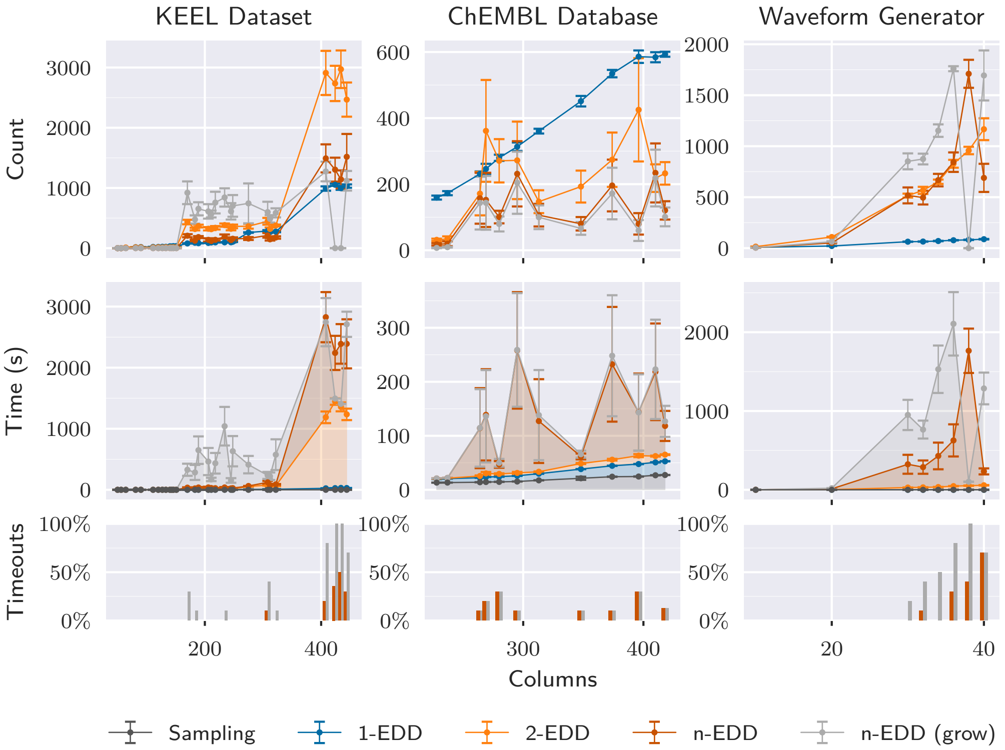


Figure 5.8: Scalability of PresQ with respect to the number of columns. The top row corresponds to the number of EDDs with arities 1, 2, and $n \geq 3$. The middle row shows the time spent on each stage: sampling, searching, and testing for the different arities. Note that the two n-EDDs variants are stacked over the previous stages, displaying the total run-time. The last row shows the percentage of runs timed out at 50 minutes. Each data point summarizes between 10 and 13 randomized runs.

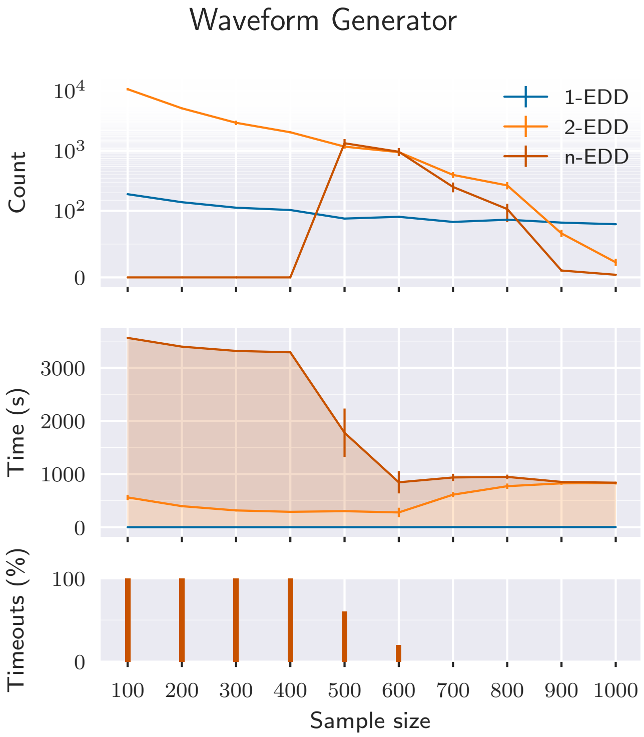


Figure 5.9: Scalability of PresQ with respect to the number of samples for the Waveform datasets. Note that for the top row, the y axis is linear between 0 and 100, and logarithmic afterward. Each data point summarizes 10 randomized runs.

5.4 Conclusions

Finding sets of equally-distributed dependencies between numerical datasets is a similar problem to that of finding Inclusion Dependencies between tables in a relational model. However, the statistical nature of tests, with their potential uncertainties, can make their finding more complicated and considerably degrade the performance of existing algorithms. This problem can be mapped to finding quasi-cliques, as the IND problem can be mapped to finding full cliques.

In this chapter, we introduced the concept of EDD, similar to the IND from the relational domain. We proposed PresQ, a new algorithm based on the search of maximal quasi-cliques on hyper-graphs. We proved that by limiting the quasi-cliques by the number of missing edges and the degree of the nodes, PresQ can successfully identify shared sets of attributes.

In general, comprehensive approaches will be needed to find very high arity EDDs, given the complexity of the IND/EDD discovery problem. In chapter 8, we discuss possible future research directions.

All the necessary code to reproduce our tests, our measurements, and figures, are publicly available[3].

[3] https://doi.org/10.5281/zenodo.6865856

Chapter 6

Two-sample test based on Self-Organizing Map

6.1 Introduction

A classification task can be seen as a sort of two-sample statistical test. If a classifier trained over two samples can effectively distinguish between them with better-than-chance performance, the classifier rejects the null hypothesis that both samples come from the same distribution [Fri03]. While a formal statistical test can be more suitable when the alternative hypothesis is known, machine learning classifiers can be a good alternative when the data is complex and abundant [Kir20; Kim21; Liu20]. Furthermore, the representation "learned" by some classifiers can be helpful to examine how the samples differ [Fri03; LO17], or it could be used later for other purposes, such as directly classifying future samples.

SOM is a technique for dimensionality reduction. It is a type of neural network that learns a low-dimension representation - generally 2D - of the original high-dimensional space while maintaining the topological layout of the original data [Koh82; Vil99]. The learned map can be directly used for unsupervised clustering and classification tasks if the training data is labeled [UM05; Ult07]. Thus, SOMs can be used as a building block of an ML-based two-sample test, similar to kNN or *Neural Network* classifiers, with the valuable addition of producing a representation that can be visualized.

In this chapter, we propose a multivariate statistical test based on SOMs that shows performance comparable to other techniques based on machine learning models and even better for medium to big sample sizes. In addition to the p value for $H_0 : P = Q$, the test also outputs a trained SOM model that can be used for other tasks, such as classification or visualization. Our proposal uses a χ^2 statistic to compare the densities of both samples on the projected plane instead of relying on a training-testing split. This allows us to fully utilize the sample data. To our knowledge, and based on the results from three exhaustive surveys, using Self-Organizing Maps to perform multidimensional two-sample testing has not been proposed before [KKK98; OKK03; PHK06].

This statistical set can potentially be used with PresQ to validate high-dimensional EDDs. The resulting projection can be used to identify objects from multiple datasets that are co-located in the matched multidimensional space, satisfying two of Borne's science requirements for data mining: *Object Cross-Correlation* and *Nearest-neighbor identification*.

In section 6.2, we introduce classifier two-sample tests and Self-Organizing Map. Then, in section 6.3, we describe our proposal for a multidimensional non-parametric statistical test based on SOMs. In section 6.4, we describe the experimental setup we used to validate our proposal — including the parametrization of the existing techniques evaluated as a baseline —. In section 6.5, we present the results. Finally, in section 6.6, we compile our conclusions and propose areas for future work.

6.2 Definitions

Classifier two-sample tests A binary classifier can be seen as a two-sample test. If a classifier has a better-than-chance performance, it can be inferred that the two classes do not originate from the same underlying population [Fri03].

More formally, let $X = \{x_0, x_1, \ldots, x_n\}$ be a sample from P, and $Z = \{z_1, z_2, \ldots, z_m\}$ a sample from Q. A test statistic $\hat{t} \sim T$ is used to "summarize" the difference between both samples and, depending on a pre-established significance level α used to reject the null hypothesis $H_0 : P = Q$ if $\alpha > P(T \geq \hat{t} | H_0)$.

When using a binary classifier for performing a statistical test, both samples are pooled together $U = \{u_i\}_{i=1}^{n+m} = \{x_i\}_{i=1}^{n} \cup \{z_i\}_{i=1}^{m}$. The samples originating from P are labeled $y_i = 1$, and the samples originating from Q, $y_i = -1$.

The original proposal trains a classifier on the *complete* pooled sample. This classifier is then used to score each data point, generating a set of scores for the first sample S_+ and for the second S_-. The multi-dimensional comparison is thus reduced to a regular univariate two-sample test problem [Fri03].

Another approach is to split the pooled dataset $\{u_i\}_{i=1}^{n+m}$ into training and testing sets. A classifier is then trained on the former, and the accuracy is measured for the latter. The accuracy becomes the test statistic $\hat{t}$, which follows asymptotically $N(\frac{1}{2}, \frac{1}{4n_{test}})$ [LO17]. Alternatively, a permutation test can be used [Kim21]. Two disadvantages of these kinds of tests are that they can not use the whole sample for computing the test statistic — therefore, they are not suitable for small datasets — and they are underpowered due to the discrete nature of the test statistic [Ros19].

Self-Organizing Maps is an unsupervised machine-learning algorithm that learns a projection from a high-dimension input space into a low-dimension output space, generally two-dimensional, to aid visualization. The output space is modeled as a grid of *neurons* — a neural map — that *responds* to a set of values from the input space [Koh82]. The output model preserves the topology of the input space: any continuous changes in the input data cause a continuous change on the neural map [Vil99]. In other words, input values close in the original high-dimensional space trigger *neurons* that are close in the low-dimensional projection [Koh13].

The output space W has to be defined before the training phase. The user needs to define the shape of the grid — square or hexagonal —, its size, and whether the map *wraps around* (toroidal maps). Each neuron i from the model has an associated weight vector with the same dimensionality as the input space, $w_i(t)$, where t corresponds to the *epoch* of the training stage. The initial values $w_i(t_0)$ can be randomly assigned or based on Principal Component Analysis [Koh13].

During the training, at each epoch t, each point x from the training set — or a batch — is mapped to its Best Matching Unit (BMU), which is just the neuron whose weight vector is the closest given a distance metric d:

$$\mathrm{bmu}(x) = \underset{w_i \in W}{\mathrm{argmin}}\ d(x, w_i) \tag{6.1}$$

Once this is done, the BMU and the weight of the neighboring neurons are updated, so they become closer to the input data point:

$$w_i(t+1) = w_i(t) + \alpha h_{i,b}(t)(x - w_b(t)) \tag{6.2}$$

Where $0 \leq \alpha \leq 1$ is a learning factor that may or may not depend on t, and $0 \leq h_{i,b} \leq 1$ is the neighborhood function, usually with a Gaussian shape that shrinks at each epoch, such as the function shown in equation 6.3 [Vil99; Wit17]. This process can be repeated for multiple epochs or until convergence.

$$h_{i,b}(t) = \exp(-\frac{||w_i - w_b||}{\delta(t)}) \tag{6.3}$$

SOMs display emergent properties when the grid is large enough: they can be directly used for clustering, classification, and other machine learning techniques. These are referred as Emergent Self-Organizing Maps (ESOM) [UM05]. This combination of emergence and visualization capabilities motivates our proposal of a statistical test based on SOMs: a test that rejects the null hypothesis that two samples are equally distributed can also provide insights into how they differ.

6.3 χ^2 test on the projection over a Self-Organizing Map

Thanks to the topology preservation of SOMs, a classifier can be trained on the output space rather than the input space. For instance, for a kNN approach, neurons can be labeled using the training data and a majority rule. Later, test data can be assigned the label from its BMU. This is almost equivalent to a kNN classifier with $k = 1$. Furthermore, neurons belonging to sparse regions can be left unlabeled, so test data projected into them can be labeled as *unknown class* [UM05; SD11].

While a SOM-based classifier could be used in place of the neural or kNN classifiers proposed originally [LO17], we propose a different approach that does not require splitting the input data into training and testing sets, leveraging the distribution of the data on the output space instead. The intuition behind this is that if two samples are equally distributed on the input space, they must be equally distributed on the output space.

More specifically, our method works as follows:

1. We train a SOM M of size (w, h) over $U = X \cup Z$
2. We project X and Z separately over the SOM M
3. We compute how many points from X and how many from Z are mapped to a given neuron n_i

$$R_i = \sum_{x \in X} [\text{bmu}(x) = i] \qquad\qquad S_i = \sum_{z \in Z} [\text{bmu}(z) = i] \tag{6.4}$$

4. Finally, we perform a a χ^2 two sample test comparing the counts for both samples on the output space

$$\chi^2 = \sum_{i=1}^{w \times h} \left\{ \frac{(K_1 R_i - K_2 S_i)^2}{R_i + S_i} [R_i + S_i > 0] \right\} \tag{6.5}$$

Where K_1 and K_2 are two constants used to adjust for different sample sizes:

$$K_1 = \sqrt{\frac{|Y|}{|X|}} \qquad\qquad K_2 = \sqrt{\frac{|X|}{|Y|}} \tag{6.6}$$

Note that we ignore the neurons where no objects are mapped. Under the null hypothesis, the test statistic χ^2 follows a χ^2 distribution with $k - c$ degrees of freedom, where k is the number of cells where $R_i + S_i > 0$, and $c = 1$ if the sample sizes are equal, or $c = 0$ otherwise [HA93].

As with any test based on binning, its main disadvantages are that its results may depend on the binning (in this case, size of the SOM) and that it requires more data points. On the other hand, since the SOM adapts to the topology of the original data, it is less susceptible to artifacts than a simple 2D histogram due to the binning.

As with the classifier tests, as a side effect of the test, we are left with a trained model that can be used for (1) visualization; and (2) for big enough SOM and samples, even for clustering [UM05]. Unlike other classifier tests, with our proposal, the whole dataset can be used for computing the statistic [Kir20]. Additionally, thanks to the regularization terms shown in equation 6.6, it also works with unbalanced sample sizes, an advantage over most kernel-based methods [SC21].

We implemented our proposal using Somuclu, a parallel tool for training self-organizing maps on large data sets [Wit17]. In the following section, we describe the experimental setup used to evaluate our proposal. Later, in section 6.5, we report the results of our tests.

6.4 Experimental setup

6.4.1 Evaluated alternatives

We considered four different two-sample tests based on machine learning techniques. All of them have in common the merging of both samples into a single set Z, labeled with 1 if the sample comes from X or -1 if it comes from Y.

Nearest neighbor type coincidences. The assumption under H_0 is that on the neighboring area of any point, the number of samples belonging to X and to Y should be similar, while if $f \neq g$, then there will be areas with a higher density of objects coming from one of the two sets [Hen88; Sch86].

To perform the test, consider a neighbor r of a sample $z_i \in Z$. We set

$$\begin{aligned} I_i(r) &= 1, \text{ if the label of } z_i \text{ and of } z_r \text{ match} \\ I_i(r) &= 0, \text{otherwise} \end{aligned} \tag{6.7}$$

The statistical test:

$$T_{n,k} = \sum_{i=0}^{n} \sum_{r=0}^{k} I_i(r) \tag{6.8}$$

n is the total number of samples, and k the number of neighbors considered. The distribution of the statistic is empirically obtained by applying a permutation test.

Classifier two-sample tests. We implemented the classifier two-sample test as described in section 6.2 using `scikit-learn` [Ped11] neural classifier[1], and kNN classifier[2] with their default parameterization. Table 6.1 summarizes the differences with the original proposal. However, these differences should not significantly affect the performance [LO17].

Parameter	**Revisiting...**	`scikit-learn`
	C2ST-NN	
Number of hidden layers	1	1
Number of neurons	20	100
Activation	ReLU	ReLU
Optimizer	Adam	Adam
Epochs	100	200
	C2ST-kNN	
k	$\lvert X\rvert/2$	5

Table 6.1: Differences between our parameterization (`scikit-learn` defaults) and the one used in the original proposal [LO17].

Kernel Methods are based on computing the Maximum Mean Discrepancy (MMD) between the samples, which is the distance between their expected features in a Reproducing Kernel Hilbert Space. In the original proposal [Gre12], however, is computationally expensive to compute the test statistic — $O(N^2)$ — and to approximate its distribution under H_0 — $O(N^2)$ or $O(N^3)$ depending on the method [ZGB13]. A proposed alternative, MMD-B [ZGB13], splits the input data into blocks, computes the original, unbiased MMD statistic on each

[1] `sklearn.neural_network.MLPClassifier`

[2] `sklearn.neighbors.KNeighborsClassifier`

block — which are i.i.d —, and averages the results. Because of the central limit theorem, this average asymptotically follows a normal distribution. Song *et al.* [SC21] propose another test statistic based on MMD-B that allows unbalanced sample sizes and is more robust to the chosen kernel bandwidth (i.e., σ on a Gaussian kernel).

6.5 Results

We performed five experiments to evaluate the performance of our SOM two-sample test proposal.

For the first three setups - Normal, DC2, and the *Open University Learning Analytics* dataset - we set the significance level $\alpha = 0.1$. We then measure the run-time and empirical type I and type II error rates over 200 repeated tests for all the evaluated tests: (1) SOM (our proposal), (2) the kNN permutation test [Sch86], and (3) two classifier tests (kNN and Neural Network) [LO17]. To obtain the 95% confidence interval, we used the Wilson score interval [Edw27].

These values are measured for: (1) a fixed sample size of $n = m = 500$ and variable dimension; and (2) for variable sample sizes and full dimensionality.

We used a *K-Best* feature selection to decide the order in which dimensions are added, from more to less informative. Therefore, increasing the dimensionality is expected to have a diminishing return.

6.5.1 Normal distribution

Figure 6.1 shows the error rates and run-time for a location test of two multivariate Gaussian distributions with $D = 1000$. For the first distribution, all dimensions have a mean of 0, while for the second distribution, the first dimension has a mean of 1, and the rest have a mean of 0.

All tests can easily reject H_0 within a reasonable run-time. For a high number of samples, however, the kNN permutation test worsens its run-time performance, probably due to the imbalance of the KD-Tree.

Figure 6.2 shows the same variables for a scale test of two multivariate Gaussian

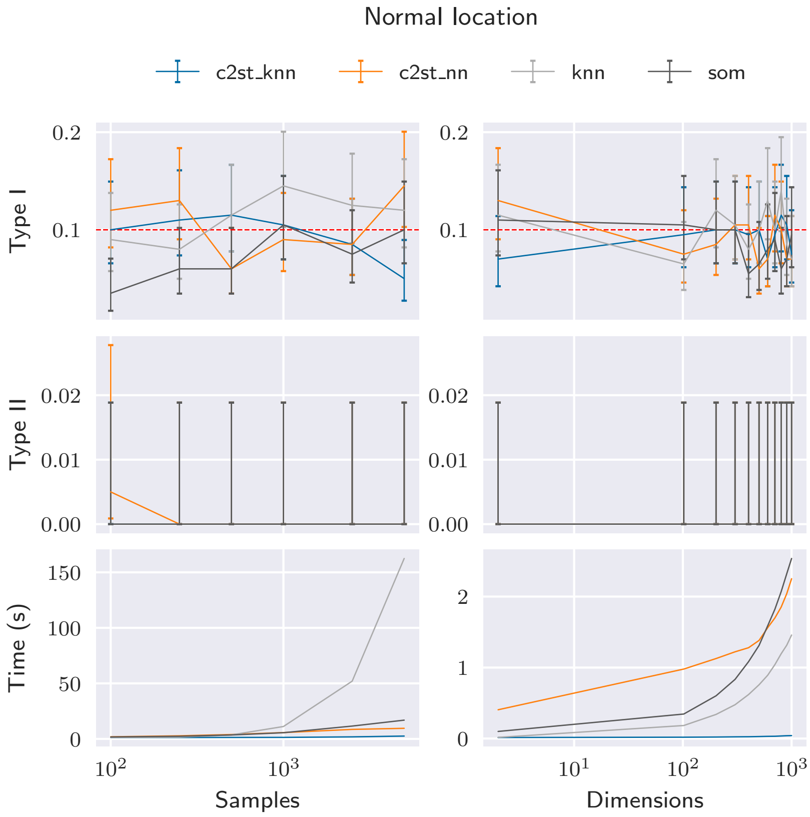


Figure 6.1: Location test for two multivariate Gaussian distributions with $D = 1000$.

distributions with $D = 1000$ and two random co-variances matrices samples from the Wishart distribution [SH72]. In this case, while the type I errors are well bounded by the significance level, both algorithms based on neural networks (C2ST-NN and SOM) require a higher number of samples to be able to reject H_0, with respect to the other methods.

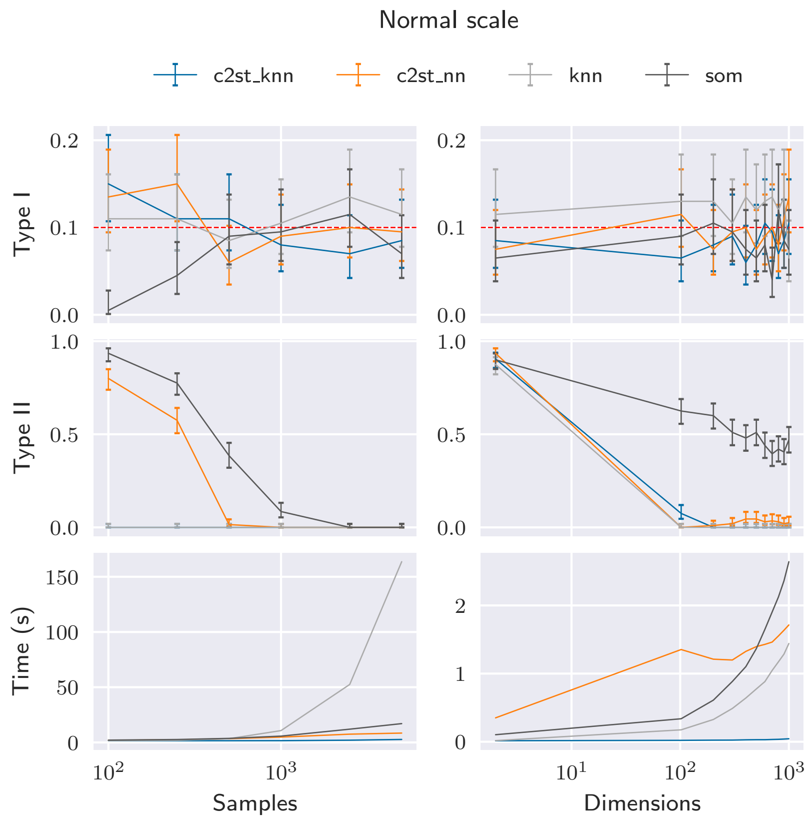


Figure 6.2: Scale test for two multivariate Gaussian distributions with $D = 1000$.

Ramdas *et al.* [Ram15] have argued that a “fair” evaluation of the power of multivariate non-parametric tests as the dimensionality increases is to keep the amount of information fixed. i.e. the Kullback-Leibler Divergence (KL Divergence) between both distributions should remain constant. For the location test, this can be achieved by two multivariate Gaussians that only differ on the first dimension, i.e. $(1, 0, \ldots, 0)$ vs $(0, 0, \ldots, 0)$.

For completeness, figure 6.3 shows the performance of the classifier two-sample tests being evaluated under this condition. We can see that they all fail to improve their type II error as the dimensionality increase. Finally, figure 6.4

shows the performance of the different tests when only the scale of the first dimension differs, as Ramdas *et al.* propose for a fair scale test. In this case, it is more evident that the type II error of all tests worsens as the number of dimensions increases.

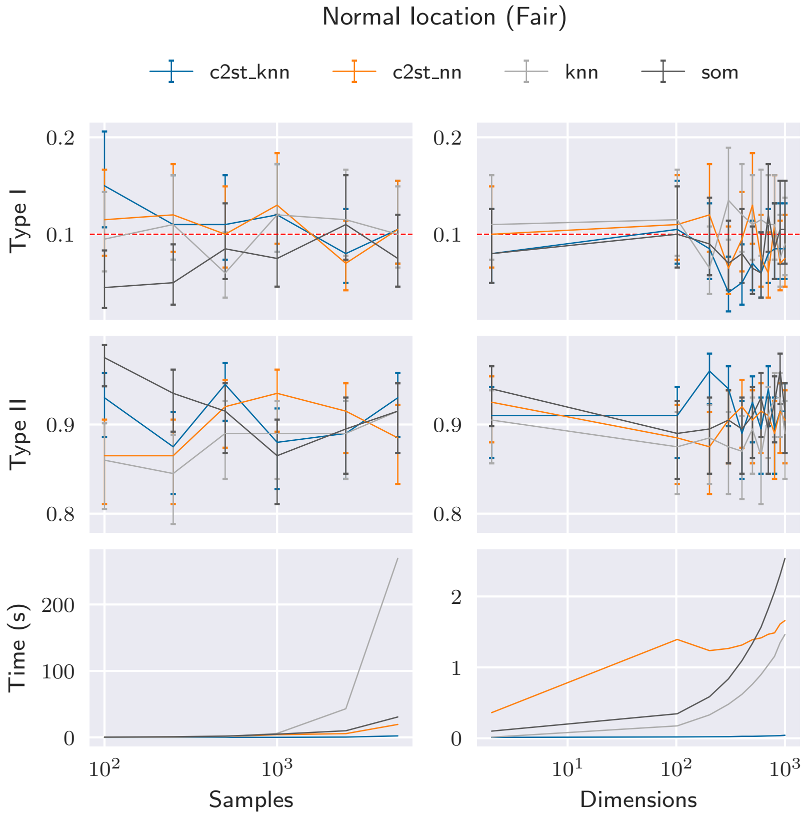


Figure 6.3: Fair location test for two multivariate Gaussian distributions with $D = 1000$.

We consider that Ramdas *et al.* raise a valid point: given the same amount of available information, additional dimensions do not help. However, for our PresQ use case, this is an unrealistic scenario: we are *discovering* the matching dimensions, and each additional feature will help discriminate whether two samples follow the same distribution.

6.5.2 DC2 Dataset

The datasets from this challenge come from a single catalog of astronomical objects split based on the sky coordinates [Des20].

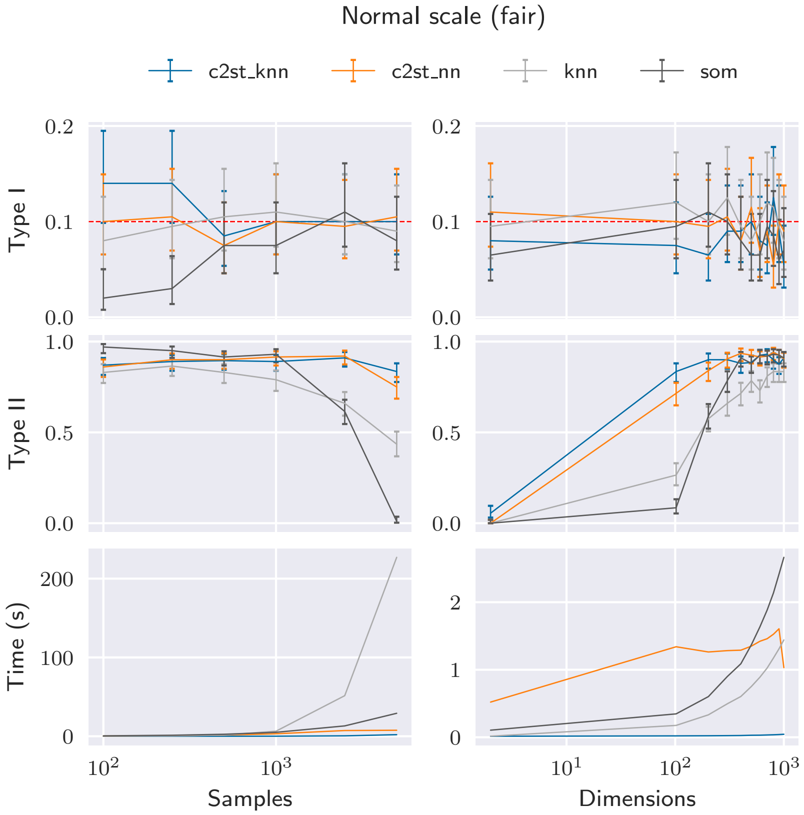
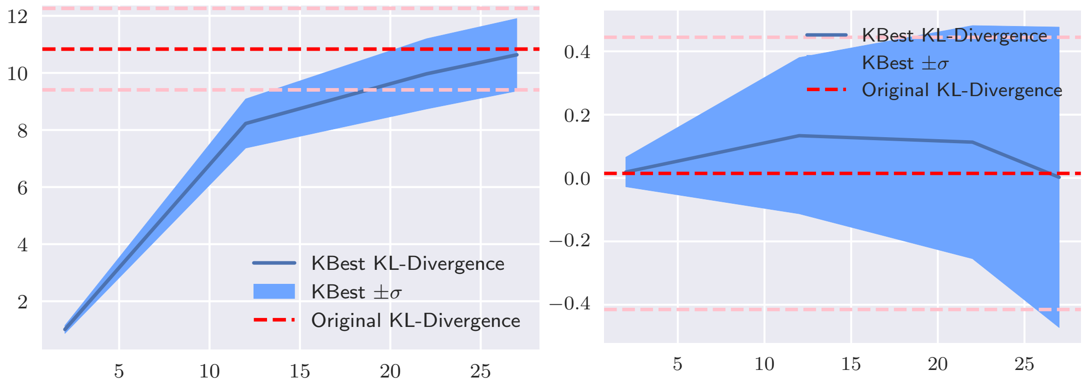


Figure 6.4: Fair scale test for two multivariate Gaussian distributions with $D = 1000$.

We generate three different samples:

1. Samples from the full catalog
2. Samples applying a magnitude cutout ($\text{MAG}_{\text{VIS}} < 22.$)
3. Samples applying a Signal-to-Noise Ratio (SNR) cutout ($\text{VIS}/\text{VIS}_{\text{Error}} > 10.$)

Following Ramdas' paper, in figure 6.5, we report the estimated KL Divergence — computed using a kNN density estimation [Pér08] — between the datasets for an increasing number of dimensions. It can be seen that the amount of available information rapidly increases for the magnitude cutout but barely for the SNR filter, probably because, for the latter, the means of the distributions barely change, but their dispersion significantly does.



(a) KL Divergence for the MAG_{VIS} cutout.

(b) KL Divergence for the SNR cutout.

Figure 6.5: Kullback-Leibler Divergence for the DC2 samples. The original divergence corresponds to the original dimensionality of the dataset.

Figure 6.6 shows the measured performances for the DC2 with the magnitude cutout. Even for a small sample size, the SOM test achieves very low type II errors, significantly better than the tests based on classifiers. This may be because the SOM and kNN tests can use the full sample, while the classifiers must split the data into training and testing sets.

Figure 6.7 shows the same performance metrics for the SNR cutout. In this case, as suggested by the KL Divergence, adding dimensions does not help any of the tests. However, as the sample size increases, the kNN and SOM tests

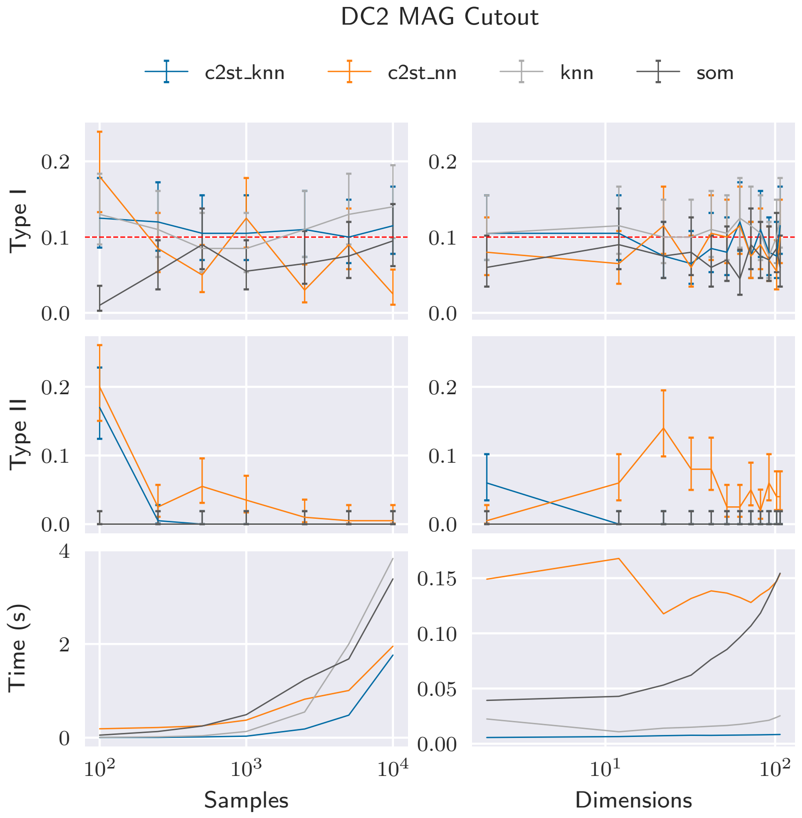


Figure 6.6: Statistical performance vs sample size (left) and dimensionality (right).

improve their type II error rate. Bigger sample sizes do not help the classifier-based tests.

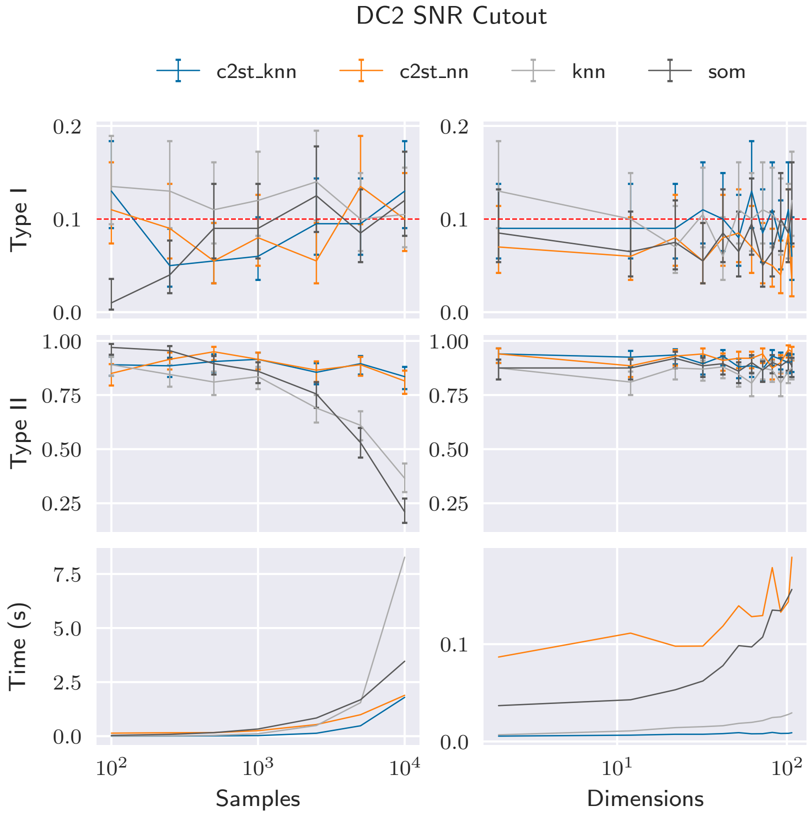


Figure 6.7: Statistical performance vs sample size (left) and dimensionality (right).

6.5.3 Open University Learning Analytics Dataset

The objective of this experiment is to prove that our proposed test can be successfully used to contrast a hypothesis, providing an interpretable result useful for further investigating the data.

We base our test on the *Open University Learning Analytics* dataset, which contains anonymized data about student demographics [KHZ17]. Let us consider the case of a researcher with the hypothesis that gender, age, and region of origin influence a student's economic situation, or perhaps they could be trying to deanonymize the data.

From this dataset, we can use the Indices of Multiple Deprivations (IMD) to

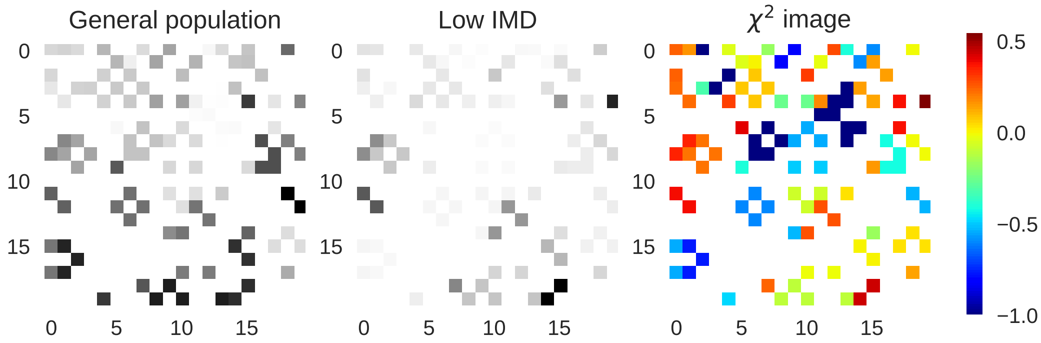


Figure 6.8: Density of samples for the general population (left), poorest segment (center) and relative difference (right). Cells with a value of -1 do not have any low-income students, while those with a value of 0.5 show an "excess" with respect to the general population.

measure poverty. The null hypothesis H_0 would be that a sample from the overall population and a sample from the poorest segment are indistinguishable, i.e., they come from the same distribution. We take all students from the lower end of the poverty line and a sample of the same size from the general population to test this hypothesis. We run the test using a SOM of size 20×20. The null hypothesis is rejected with a p-value of 0.

Unlike other statistical tests, in addition to the p-value, the researcher can use the result of the SOM test to compare the projections of both samples. Figure 6.8 shows the density of samples for each cell for the overall population (left), the density of samples from the low-income students (center), and the relative difference between both (right). The "most different" cells hint at how they are different.

If we pick one of the cells with the most significant bias towards low-income students, we can see it contains only young female students from the *North Western Region*. In figure 6.9, we show the distribution of IMD for the overall population (left) and for this subset (right). Indeed, the income distribution for this demographics is heavily skewed towards the low end. We could obtain this hindsight without prior knowledge of which attributes correlate with the difference, only with the "hunch" that there is a relation. Thus, we prove that our proposed test can be useful for data exploration.

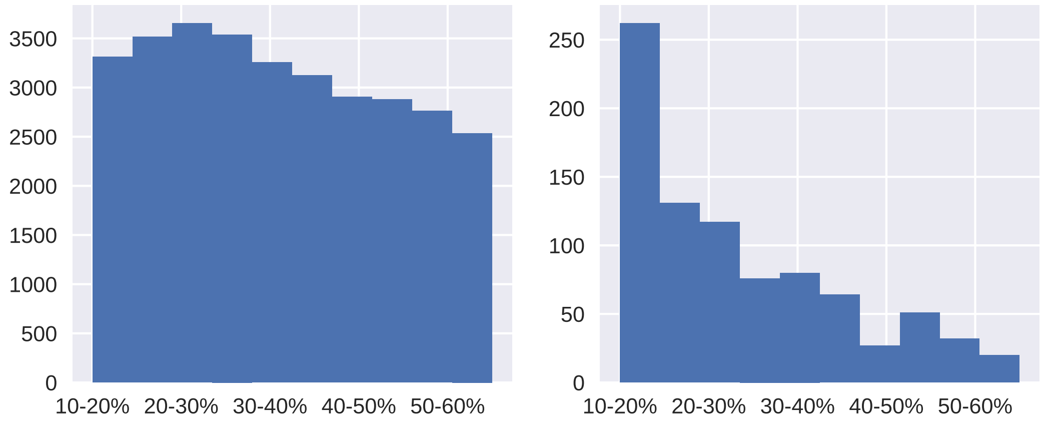


Figure 6.9: *Indices of Multiple Deprivations* for the general population (left) and for young female students from the *North Western Region*.

6.5.4 Eye Movements Dataset

For generating this dataset, 11 subjects were shown a question and a list of ten associated sentences, of which one was the correct answer (C), four relevant (R), and five irrelevant (I). Their eye movement was measured for each possible answer. The overall measurements were summarized into 22 features, together with the appropriate label for the sentence [Sal05].

With this dataset, we aim to prove that our method can be competitive with other start-of-the-art proposals, with the benefit of providing a trained model useful for later purposes.

To evaluate the power of comparing different sets of measures, we replicate Song's set up and run 2000 times[3]the classifiers and SOM tests using a significance level of $\alpha = 0.001$ for different sample sizes. We report their statistical power in table 6.2. The results from Song and MMD-B are extracted from Song's paper [SC21].

The results show that our test has low power for small samples, but it rapidly gains terrain compared to the classifier-based methods, being competitive even with kernel-based techniques. The nearest-neighbors method is remarkably efficient in all cases.

To evaluate the usefulness of the trained model obtained as part of running

[3] Song uses 1000 repetitions.

I vs. C						
m = n	**Song**	**MMD-B**	**KNN**	**C2ST-KNN**	**C2ST-NN**	**SOM**
100	0.826	0.374	**0.973**	0.164	0.079	0.042
200	0.998	0.850	**1.000**	0.565	0.349	0.947
300	**1.000**	0.985	**1.000**	0.863	0.644	**1.000**
400	**1.000**	**1.000**	**1.000**	0.968	0.882	**1.000**

R vs. C						
m = n	**Song**	**MMD-B**	**KNN**	**C2ST-KNN**	**C2ST-NN**	**SOM**
100	0.670	0.236	**0.845**	0.062	0.023	0.007
200	0.969	0.685	**0.996**	0.298	0.139	0.672
300	0.999	0.941	**1.000**	0.558	0.314	0.987
400	**1.000**	0.988	**1.000**	0.811	0.560	**1.000**

Table 6.2: Empirical statistical power for the eye movement datasets. We mark in bold the best results for each sample size.

the statistical test, we use the trained SOMs as classifiers: The neurons can be labeled according to the training data labels mapped into their region using a majority rule. During classification, objects can be labeled according to the label of the neuron into which they are mapped. We performed a 50-fold cross-validation with a sample size of $n = m = 409$, so each training set has $n = m \approx 400$ elements for each label.

The obtained mean accuracy where: C vs. I 72.48%; C vs. R 68.48%; I vs. R 57.60%. Even with relatively small sample sizes, our results are comparable with those reported on the paper from which the dataset was obtained: 65.8% Correct vs. Incorrect (joint of Irrelevant and Relevant) [Sal05].

Finally, as an exercise on interpretability, figure 6.10 shows the value of the two most distinct code-book dimensions. These attributes, related to the regression (re-reading a word) show a sharp distinction that matches the distribution of samples from the Correct and Incorrect samples quite well. This matches the expectations from the original paper that regression features indicate high-level cognitive processing and, therefore, correlate with conscious efforts when choosing a correct answer.

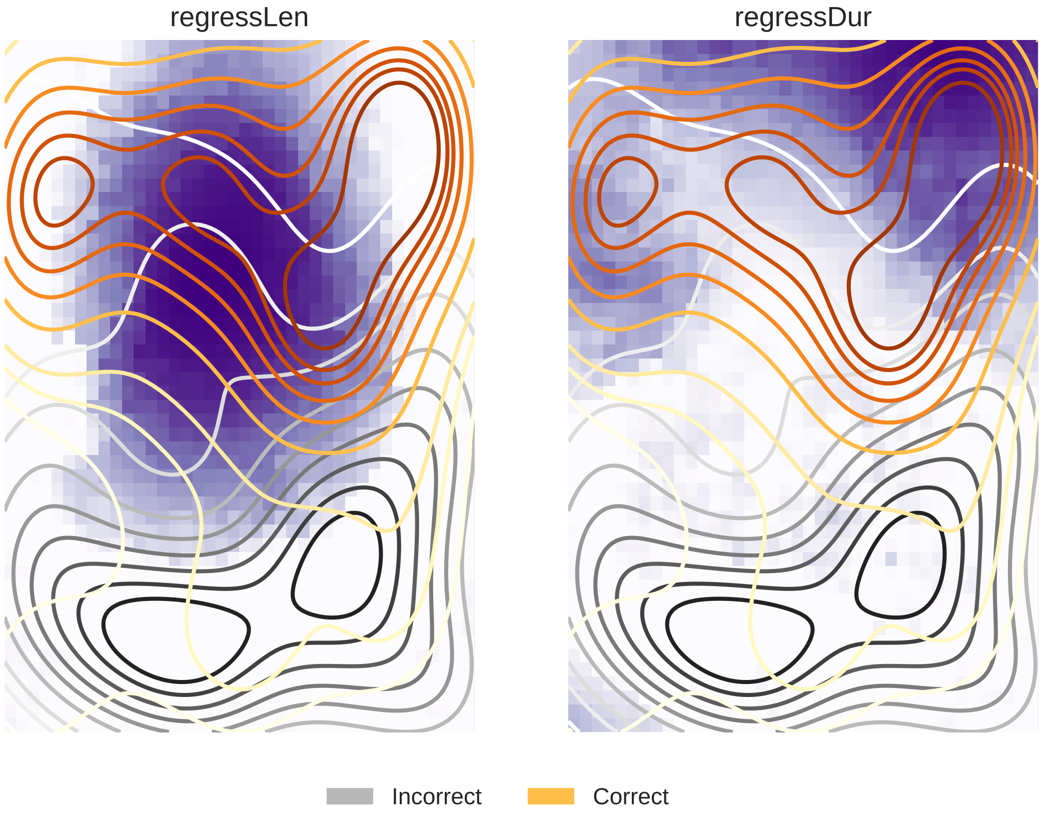


Figure 6.10: Code-book values for the two most distinct features from the Eye dataset. Darker neurons are more sensitive to the given feature. The lines represent the density estimation for the Correct and *Incorrect* samples over the SOM. It is visible that the Incorrect category "wraps" around long regression distances, while the *Correct* category correlates positively with the regression duration.

6.5.5 Age (IMDb Faces) Dataset

The IMDb-WIKI dataset [RTV16] contains 460,723 face images extracted from IMDb. The authors additionally provide a neural network pre-trained to predict the age of the face cutouts. Similarly to Song's experiments [SC21], we ran the neural model over the IMDb faces, extracting the values from the last hidden layer (4096 neurons) as the target multivariate distribution. We then group the samples in age ranges and verify our proposal, comparing 500 samples from consecutive age groups, and repeat 500 times for each experiment. The significance level is also set to 0.001.

Table 6.3 shows the results for the SOM test, together with the results reported by Song*et al.* for their kernel-based method and for MMD-B[ZGB13]. For this particular experiment, Song's proposal outperforms both MMD-B and the SOM test, although our method comes second in statistical power.

Age ranges	**Song**	**MMD-B**	**KNN**	**SOM**
15-20 vs. 20-25	1.000	1.000	1.000	1.000
20-25 vs. 25-30	1.000	0.800	0.984	0.990
25-30 vs. 30-35	0.990	0.790	0.876	0.726
30-35 vs. 35-40	1.000	0.830	0.866	0.812
35-40 vs. 40-45	0.950	0.250	0.784	0.564
40-45 vs. 45-50	0.930	0.400	0.812	0.606

Table 6.3: Statistical performances for the Age dataset.

6.5.6 PresQ results

As we mentioned in the introduction, the motivating example is the unsupervised discovery of shared attributes between multiple datasets using PresQ (chapter 5), obtaining, as a result, the list of sets of attributes and a trained model that can be used to cross-match the datasets.

We run two examples from the PresQ paper using the proposed SOM based test instead of the kNN statistical test. For each of the examples, we measure the following:

Ratio of known shared attributes identified by PresQ.

Overhead Number of tests per unique Equally-Distributed Dependency found.

Time that took PresQ to finish, including the time taken for serializing the SOM models.

Figure 6.11 reports the 95% confidence interval ($\mu \pm 1.96\sigma$) for the *relative differences* obtained when running with SOM vs kNN. Their distribution has been estimated using bootstrapping. The parametrization for PresQ was $\Lambda = 0.1$, $\gamma = 0.95$, $\alpha = 0.05$ and a sample size of 1000.

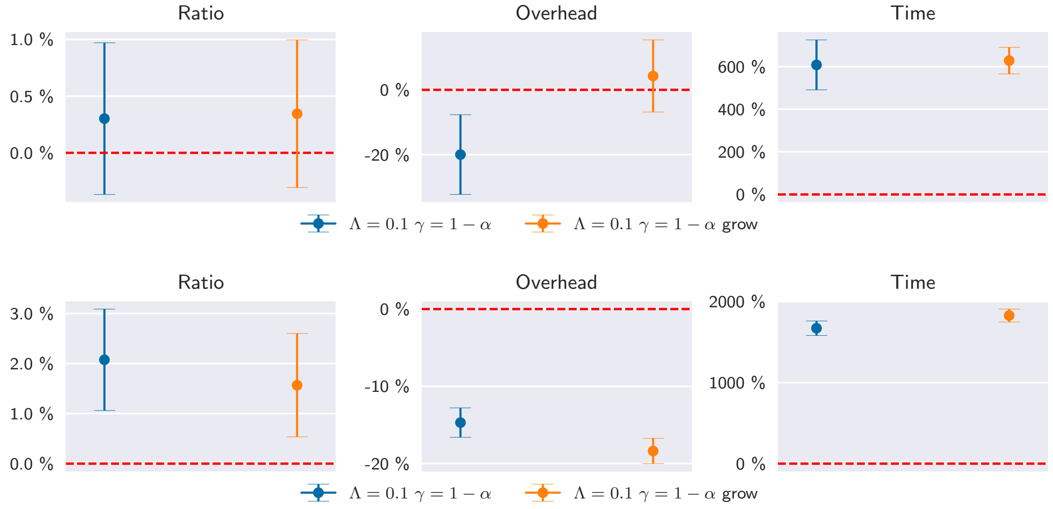


Figure 6.11: Relative difference between PresQ_{SOM} and PresQ_{kNN}. Top: DC2 dataset [Des20]. Bottom: Ailerons / Elevators datasets [Alc11].

The SOM test has a run-time penalty because it is a more complex model to train. However, fewer tests are required for the same number of unique EDD found. This is likely a consequence of the kNN test being slightly more prone to reject the equality of distribution than the SOM test, as we can for instance see in figures 6.1, 6.2.

6.6 Conclusions

As part of interactive data exploration, researchers may need to compare multiple datasets. These datasets can originate from multiple independent files or generative models that need to be compared with reality. When these datasets are of high dimensionality, especially if the exploration is tentative, developing tailored statistical tests can become impractical. In those cases, relying on heuristic approaches based on machine learning techniques, as classifiers, to decide whether two samples are distinguishable becomes a good alternative [Fri03; Kim21].

However, some of these methods, like neural networks, are hard to interpret when rejecting the null hypothesis $H_0 : P = Q$. In other words, they reject that both samples originate from the same underlying distribution but do not further assist the researcher. Other models, such as random forests, are more interpretable [Fri03].

In section 6.3, we proposed another machine learning technique based on Self-Organizing Maps [Koh82] that is understandable and capable of pointing the researcher to where the differences in a multidimensional space are. After all, SOMs were initially proposed as visualization aids. Nonetheless, they display interesting emergent properties and can be used for clustering or classification as well [Ult07].

In section 6.5, we proved that the power of this technique is comparable to other machine learning techniques and even superior for medium-size datasets. We also proved in the experiment 6.5.3 that the output could guide researchers toward refining a hypothesis. Thus, our test can be a valuable asset to the researcher's tool-set and complementary to more formal hypothesis testing whenever considered necessary [Ros19].

For future work, it would be interesting to explore the possibilities of properties of emergent Self-Organizing Maps to assist the researcher in examining the differences. For instance, clustering could help identify whole regions that differ rather than focusing on individual neurons.

Chapter 7

Discussion

In section 1.1, we defined that the main objective of this thesis was *to assist users during the exploration of unprocessed, numerical, raw data distributed across multiple files*. After a systematic literature mapping, described in detail in chapter 3, in section 4.2 we narrowed the objectives to the *multiple files with unknown, heterogeneous schemas, solely using the intrinsic data distribution*.

With this objective in mind, in chapter 4 we proposed the concept of EDD, proposed a novel algorithm, PresQ, to mine these dependencies between multiple datasets — chapter 5 —, and a two-sample, machine-learning statistical test that can visually assist users in examining how the two samples differ — chapter 6.

In section 7.1, we discuss the behavior, performance, and results of PresQ, a tool for the discovery of multidimensional EDDs via Quasi-Cliques on Hypergraphs. In section 7.2, we argue how the proposed solutions are relevant to the stated objectives. In section 7.3, we list the threats to the validity of this work and what measures we have taken to minimize their impact.

7.1 Discovery of Multidimensional Dependencies via Quasi-Cliques on Hypergraphs

Identifying shared attributes between multiple numerical datasets is an interesting problem. It combines the challenging nature of algorithms devised to find Inclusion Dependencies, an NP-hard problem [Kan92], with the unavoidable uncertainty of statistical methods. This uncertainty reflects as falsely rejected EDDs and falsely accepted EDDs.

Find2 is an algorithm that maps inclusion dependencies to hyper-cliques, which generally performs at least as well as the alternatives [Dür19]. It is loosely coupled with the discrete nature of the underlying data. However, its ability — and of most, if not all, of the existing algorithms — to find high arity EDDs will be impaired by the number of false rejections, which a lower rejection threshold could compensate for. Yet, this solution increases the number of false detections, which is a known factor that significantly degrades its performance, similar to other hypergraph-based methods' [KR06]. We experimentally confirmed this problem in section 5.3.3.

We propose a new algorithm based on quasi-cliques, where a candidate is accepted even if some edges are missing. This algorithm has three parameters:

- The ratio of missing edges (γ).
- The tolerance on the number of missing edges connecting a node from the quasi-clique (Λ).
- Whether to use the found quasi-clique as seeds.

We provide a generalization of this parameterization from regular 2-graphs [BHB07] to uniform n-graphs in equations 5.4 and 5.5.

The results showed in the quasi-clique test set (section 5.3.3) demonstrate that the seed stage of PresQ provides results close to the original cliques on uniform n-hypergraphs. The growing stage can recover them even for a high number of missing edges (up to 30%) at the expense of a higher run-time.

For real datasets, the ratio of missing edges can be intuitive to configure (simply $\gamma = 1 - \alpha$, where α is the test significance level), but λ can be harder to interpret. We propose instead an intuitive and statistically interpretable method to dynamically adapt the threshold to the degree which is expected to follow a hypergeometric distribution and can be adjusted based on the quasi-clique itself, as shown in equation 5.8. Our results also prove that the degree threshold based on the hypergeometric distribution offers comparable performance to a hand-picked ratio λ while being more stable and predictable.

While our tests on artificial hypergraphs seem to point to the redundancy of the parameter Λ, the results shown in the real-world test set (section 5.3.3) prove that for real noisy graphs, the combination of both performs consistently better than either of them separately. The γ parameter enables recovery from missing edges and, simultaneously, Λ avoids too many false positives due to spurious edges. Thanks to them, the efficacy can be kept even while maintaining or increasing the significance level of the tests. This reduces the risk of decreased performance since the density of the graphs can be kept under control.

If a more exhaustive listing of maximal quasi-cliques is required, the initial set of quasi-cliques can be used as *seeds* to grow other quasi-cliques by adding suitable vertices. The results shown in section 5.3 demonstrate that this method is capable of finding considerably more maximal quasi-cliques (not contained in any other found quasi-cliques) at the expense of a higher run-time. This is due to the traversal of the search space and the validation of the EDDs represented by the quasi-cliques.

The loss of accuracy introduced by this growing stage is minor when starting at $n = 3$, which means that the statistical test could not reject most candidates. However, most candidates were rejected for an initial $n = 2$. We consider this is mostly due to the lack of power of the kNN test for low dimensions, which introduces many spurious edges.

The overall run-time of the EDD finding algorithms is heavily influenced by the chosen parameter values. A strict parametrization will reject most seed candidates, and the quasi-clique search will fall into exponential complexity. Conversely, a flexible one will be faster at finding quasi-clique candidates. Yet, the statistical test will likely reject them, causing their decomposition into an exponential number of newer, smaller candidates. Generally, a balanced parametrization based on γ and Λ is more predictable.

As a final remark, the set of accepted EDDs may contain several false positives depending on the power of the multivariate statistical test. A second, more detailed pass can refine this original set. For instance, we can envision a ranking based on the previously discussed *randomness* [Zha10] to decide which set of EDDs is more suitable for cross-matching the datasets.

In conclusion, PresQ successfully identifies shared sets of attributes between multiple data files containing raw, numerical data solely based on their distribution. This can guide users to cross-match datasets with unknown schemas, but also to label attributes for which the metadata is lost as long as there is available one dataset with properly labeled attributes.

Furthermore, PresQ can also provide insights that drive serendipitous discoveries. The AFDS result shown in figure 5.7 is an example of such a case: even with a set of files of unknown schema and unknown content, we could infer a relation between samples, which we later confirmed looking at the paper that published the data.

7.2 Two-sample test based on Self-Organizing Map

The results shown in section 6.5 prove that the non-parametric, two-sample SOM test described in section 6.3 generally outperforms, in terms of power, other classifier-based two-sample tests, being in some cases comparable to kernel-based methods. It has the added advantage of generating an interpretable and usable model: for instance, the resulting SOM could be used as the basis for a SOM-kNN combined classification model [SD11].

Like other machine-learning or kernel-based approaches, our method requires some initial parameters, such as the SOM size, to be set by the user. From our tests, networks of size $O(100)$ neurons generally work well enough, but for more precise control, the SOM size can be set to the lengths of the two largest principal components [Koh13].

Rosenblatt *et al.* [Ros19] argue that provable, *proper* test statistics should be, in general, preferred over heuristic alternatives. However, our proposal is more oriented toward exploring abundant structured data. In this case, developing a tailored statistic for all possible combinations is not viable, and a pragmatic approach is more suitable [Kim21].

Finally, if the null hypothesis — that both samples come from the same underlying distribution — is rejected, the generated SOM model can be easily visualized and examined in more detail. In sections 6.5.3 and 6.5.4, we demonstrated that we can reject H_0 and obtain valuable information after exploring the learned model. This exercise would not be possible with a black-box method such as a neural network classifier [Fri03].

As a final note, SOM maps can be generalizable to non-vectorial data — i.e., strings — as long as more than one ordering relation is defined [Koh82; Koh13]. Therefore, our proposed statistical test could also be used to verify whether two sets of sequences — i.e., proteins or DNA — share their origin. Unfortunately, since the SOM implementation on which we based our implementation only works with real-valued dimensions [Wit17], we were not able to test this scenario.

7.3 Threats to validity

We now identify the internal and external threats to the validity of our research, and the measures we took to counteract their effect.

7.3.1 Internal validity

The PresQ results shown in the experiments described in section 5.3 could risk being just a fluctuation, not due to an underlying algorithmic improvement. However, the experimental design described in section 5.3.1 significantly reduces this possibility, thanks to the randomization of the initial conditions and the number of measurements. In any event, we made explicit the uncertainties of our results — using 95% confidence intervals or reporting distribution quartiles.

The results summarized in table 5.6 show that, on average, the quasi-clique-based searching algorithm consistently performs better both in terms of run-time and ability to find the maximal EDD. It has enough runs to make the difference significant. It is worth mentioning that there are proposed heuristics [KR03] to find higher arity EDDs, even when edges are missing, by merging found lower-arity EDDs and testing the resulting EDD candidate. Nonetheless, we consider that the run-time differences are significant enough to make the quasi-clique-based search a better approach in those cases. Even so, that heuristic can also be applied to the output of our proposed algorithm.

We implemented Find2 and PresQ from scratch, sharing many parts of the code — i.e., data structures, statistical tests, etc. While there is room for optimizations, both would benefit. Since the relative differences would remain similar, we are confident that the gains come from the algorithm rather than its implementation.

For the proposed two-sample statistical test, the results shown in section 6.5 originate from running the tests between 200 and 2000 times, with independent randomized samples. Again, the comparisons took into account the error in our measurements, making it easier to assess their significance.

7.3.2 External validity

The PresQ experiments were run over four different datasets of diverse nature and from three separate sources. The chosen statistical tests for uni- and multi-dimensional distributions were not customized to any of them. However, a better statistical test can be used if the underlying data distribution is more or less known (or suspected), which may reduce, or even remove, the advantage of the quasi-clique approach. Although it is also unlikely that the performance would be any worse: since an entire clique is still a quasi-clique, our algorithm can identify all of them, similar to the original Find2 algorithm.

One significant caveat of our approach is that it may not find any dependencies if prior filtering has been applied to only one of the two relations (i.e., signal-to-noise filtering). This is a limitation of the validation step. This issue was also recognized on the original Find2 proposal [KR03].

Similarly, the experiments for the two-sample statistical test based on SOM were run over five different datasets and compared with kernel-based and machine-learning-based alternatives. Given the performance shown by this test, we are confident in its capabilities.

Chapter 8

Conclusions and Future Work

In this chapter, we first summarize our research in section 8.1. Then, in section 8.2, we enumerate the main contributions of this thesis. Finally, in section 8.3, we suggest possible future lines of work to either improve or build upon the contributions from the present work.

8.1 Summary

In-situ Data Exploration is an active research area that requires a multidisciplinary approach: algorithms, data structures, machine learning, statistics, data visualization, information sciences, and the domain knowledge — or business understanding — provided by an expert.

Going back to the CRISP-DM model described in chapter 1 and shown in figure 1.2, our initial objective was to identify gaps in the tooling available for experts to understand data coming as a set of unprocessed and perhaps inconsistent sets of files. These files are not optimized for access, and any early decision on how to ingest them into a database may be counterproductive until the dataset is better understood.

The literature survey from chapter 3 showed that solutions for visualization, optimizations, indexing, and physical layout abound. Still, there is little to no mention of assisting users on *understanding* data schema, especially when it comes to multiple files from diverse origins or when meta-data is incomplete or inconsistent. In chapter 4, we confirmed that users spend a non-negligible amount of resources just examining the data structure and layout.

The following question was: can we leverage the data *distribution* to assist users in understanding the schema, on seeing how different datasets may come together? This is particularly important when the data is numerical and uncertain since one can not just compare tuples individually but needs to inspect distributions. In the relational world, the IND concept comes close to the objective: parts of a relation that are contained (included) within another. However, this modeling relies on the attributes' discrete nature, such as name, date of birth, etc.

In chapter 5 we propose a generalization, the EDD, that relaxes the strict containment relation required by INDs. We then introduce PresQ, a novel algorithm for EDD finding that incorporates uncertainty into its world modeling, proving that relying on data distribution alone is feasible. Therefore, PresQ is applicable in situations where most existing IND solutions are not: when the data is intrinsically uncertain — measures of physical properties —, or when the validation strategy for the inclusion is an approximate heuristic. The only requirement is that the expectation of false negatives — i.e., significance level for statistical tests — can be estimated.

For the experiments used to evaluate PresQ, described in 5.3, the statistical test of choice was based on kNN. While this test performs well for many datasets, the resulting trained model can not be easily reused. In chapter 6, we introduce a statistical test based on SOM, which provides, in addition to a p value, a trained projection that we can later use for binning and cross-matching both datasets using the matching set of features. The resulting SOM is also interpretable in case of rejection, which is also helpful in tentative data exploration.

8.2 Contributions

We summarize here the main contributions of this thesis:

8.2.1 Survey of state of the art

Idreos *et al.* surveyed the state of the art of IDE up to the first half of 2015 [IPC15]. Being an active research area, we performed an exhaustive systematic literature mapping up to mid-2017. Almost 60% of the classified works were published in 2015 and later.

By the end of 2022, we updated our survey to check on the evolution of the research. This second round showed that sampling techniques for AQP remain a popular and evolving area of research. Additionally, Deep Learning is becoming more popular in this domain thanks to its pattern recognition and data summarization capabilities.

The results of this survey, presented in chapter 3, are a valuable update on the state-of-the-art and contribute to the first sub-objective: finding existing techniques that help users to explore the data *in-situ*.

8.2.2 New category for Interactive Data Exploration

The state of the literature at the time of our survey indicated that within the IDE research community, querying datasets split over multiple files is often neglected [Sil16a]. One could argue that the logical file containing a given subset of the data is a detail that can be integrated into an existing index. This is true as long as the data layout is similar, but this is not always the case. We proposed a new category within the *Middleware Layer*: *Schema Homogenization*. A new set of articles has been classified under this category in table 4.3. This new categorization contributes to the first and second sub-objectives: identify gaps in the coverage.

8.2.3 Definition of Equally-Distributed Dependency

In section 5.1.1, we introduced the concept of Equally-Distributed Dependency, expanding the family of data dependencies to four types [AGN15]. We summarize the classification in table 8.1. Note that EDD are always, by definition, approximate.

Uniqueness	Key Discovery
	Conditional
	(Approximate)
Inclusion Dependency	Foreign key discovery
	Conditional
	(Approximate)
Functional Dependencies	Conditional
	(Approximate)
Equally-Distributed Dependency	(Approximate)

Table 8.1: Classification of dependency detection tasks.

The concept of EDD uncovers possible new future research areas. Hence, it contributes to the second sub-objective.

8.2.4 Algorithms

Querying data split across multiple files with similar schemas is another active area of research within the *Relational Algebra* domain. However, existing solutions do not work over attributes defined in the real domain.

In chapter 5 we presented PresQ, an algorithm for finding quasi-cliques on uniform hypergraphs that can be used to find sets of attributes EDD between datasets. PresQ is not limited to EDD finding and can potentially be of use to other areas that have a strong combinatorial aspect, such as chemistry or biology [Bre13].

In chapter 6, we introduced a two-sample multivariate statistical test based on SOMs with a statistical performance comparable to tests based on machine learning models. An additional advantage of this technique is that it outputs an interpretable projection of the two samples. It can be used by itself or in conjunction with PresQ.

These two contributions contribute to the third sub-objective: design new algorithms tailored to numerical and uncertain data.

8.2.5 Publications

The contributions from this thesis have been published in the following papers:

[APD19] Alejandro Alvarez-Ayllon, Manuel Palomo-Duarte, and Juan Manuel Dodero. "Interactive Data Exploration of Distributed Raw Files: A Systematic Mapping Study". In: *IEEE Access* 7.1 (2019), pp. 10691–10717. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2882244.

[APD21] Alejandro Alvarez-Ayllon, Manuel Palomo-Duarte, and Juan-Manuel Dodero. "Inference of common multidimensional equally-distributed attributes". arXiv: 2104.09809 [cs.DB]. 2021. URL: https://arxiv.org/abs/2104.09809.

[APD22a] Alejandro Alvarez-Ayllon, Manuel Palomo-Duarte, and Juan Manuel Dodero. "PresQ: Discovery of Multidimensional Equally-Distributed Dependencies via Quasi-Cliques on Hypergraphs". In: *IEEE Transactions on Emerging Topics in Computing* Special Section on Emerging Trends and Advances in Graph-based Methods and Applications (2022), pp. 1–16. ISSN: 2168-6750. DOI: 10.1109/TETC.2022.3198252.

[APD22b] Alejandro Alvarez-Ayllon, Manuel Palomo-Duarte, and Juan-Manuel Dodero. "Two-sample test based on Self-Organizing Maps". arXiv: 2212.08960 [cs.LG]. 2022. URL: https://arxiv.org/abs/2212.08960.

8.2.6 Software

The software generated and used in our tests are archived and publicly available in Zenodo:

[Alv22a] Alejandro Alvarez-Ayllon. *PresQ: Discovery of Multidimensional Equally- Distributed Dependencies Via Quasi-Cliques on Hypergraph (Source)*. July 2022. DOI: 10.5281/zenodo.6865856.

[Alv22b] Alejandro Alvarez-Ayllon. *SOMA: Self-Organizing Map Analysis (Source)*. Dec. 2022. DOI: 10.5281/zenodo.7452720.

8.3 Future work

Finally, we enumerate some interesting research paths that can further improve the current state of the art, adding valuable algorithms to the tool-sets available for data scientists:

- In chapter 5, we prove that finding quasi-cliques in hypergraphs is a successful technique to find EDD or IND based on approximate heuristics between relations. Therefore, a place for further research is to **improve the quasi-clique finding** algorithm, either via novel algorithms or by generalizing some of the many existing techniques [WH15].

- On the other hand, the existing algorithms based on clique and quasi-clique search decouple the data — only used during validation – from the candidate generation. An interesting research strategy can be making clique and quasi-clique finding algorithms **data-aware**, leveraging data features during the generation of candidates.

- Sometimes, the algorithms should not assume equality of distribution. For instance, one of the datasets may have been filtered beforehand — i.e., signal-to-noise, value clipping, etc. This issue affects both EDD and IND algorithms [KR03]. **Partial inclusion/equality-of-distribution** remains an open problem.

- PresQ is based on frequentist probability, which does not allow incorporating **prior beliefs** into the algorithm. i.e., a **domain-expert** has no way of influencing the result based on her knowledge of the area. A Bayesian framework could prove helpful in this respect. Furthermore, it can also be used to perform local null hypothesis testing [Sor15], which could help identify partial EDDs — as when a filter has been applied to one of the datasets.

- Searching for quasi-cliques involves exponential time complexity on the number of nodes. Thus, applying a **dimensionality reduction** would reduce the total run-time and decrease the noise. Nonetheless, a complication arises because we do not know which attributes are shared.

- Finally, **multidimensional *complementarity*** is another interesting area for further research. For instance, a single dataset may be split between multiple files based on the values from a given set of attributes (i.e., coordinates), which may or may not overlap. Combining complementary datasets would be a powerful addition to EDD finding.

Future research can also be directed toward novel applications of the algorithms presented in this work, particularly in privacy-preserving data mining. For instance, PRESQ is designed to *link* multiple datasets containing noise, errors, and uncertainty. It can be interesting to evaluate its capabilities to perform *linking* (re-identification) attacks [Che09] on anonymized datasets. Additionally, the results of our tests on the two-sample test based on SOM, particularly the example shown in section 6.5.3, hint at the possibility of using this technique to achieve *attribute disclosure*: the attackers improve their knowledge of a set of attributes for a given individual [Cyb16]. Should these algorithms prove useful in performing this kind of attack, researchers could use them to guide their efforts toward new defense models.

Acknowledgments

This research was conducted in collaboration with the projects CRÊPES (ref. PID2020-115844RB-I00): *Creation and Evaluation in Privacy-preserving Extended Learning Scenarios* and PHADAS (ref. TED2021-132073B-I00): *Human-centered Privacy for Sustainable Open Data Analytics*.

Bibliography

[Abo18] Bela Abolfathi et al. “The Fourteenth Data Release of the Sloan Digital Sky Survey”. In: *The Astrophysical Journal Supplement Series* 235.2 (2018), p. 42. DOI: 10.3847/1538-4365/aa9e8a.

[Aga13] Sameer Agarwal et al. “BlinkDB: queries with bounded errors and bounded response times on very large data”. In: *Proceedings of the 8th ACM European Conference on Computer Systems*. New York, NY, USA: ACM, 2013, p. 29. ISBN: 978-1-4503-1994-2. DOI: 10.1145/2465351.2465355.

[AGN15] Ziawasch Abedjan, Lukasz Golab, and Felix Naumann. “Profiling relational data: a survey”. In: *The VLDB Journal* 24.4 (2015). publisher: Springer, pp. 557–581. DOI: 10.1007/s00778-015-0389-y.

[Ala12] Ioannis Alagiannis et al. “NoDB in action: adaptive query processing on raw data”. In: *Proceedings of the VLDB Endowment* 5.12 (Aug. 2012), pp. 1942–1945. ISSN: 2150-8097. DOI: 10.14778/2367502.2367543.

[Ala14] Abdussalam Alawini et al. “Helping scientists reconnect their datasets”. In: *Proceedings of the 26th international conference on scientific and statistical database management*. 2014, pp. 1–12. DOI: 10.1145/2618243.2618263.

[Ala16] Abdussalam Alawini. “Identifying relationships between scientific datasets”. PhD thesis. Portland State University, 2016. DOI: 10.15760/etd.2918.

[Alc11] Jesús Alcalá-Fernández et al. “Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework.” In: *Journal of Multiple-Valued Logic & Soft Comput-*

ing 17 (2011). URL: https://sci2s.ugr.es/sites/default/files/files/ScientificImpact/255-287%20pp%20MVLSC_169i.pdf.

[APD19] Alejandro Alvarez-Ayllon, Manuel Palomo-Duarte, and Juan Manuel Dodero. “Interactive Data Exploration of Distributed Raw Files: A Systematic Mapping Study”. In: *IEEE Access* 7.1 (2019), pp. 10691–10717. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2882244.

[APD22] Alejandro Alvarez-Ayllon, Manuel Palomo-Duarte, and Juan Manuel Dodero. “PresQ: Discovery of Multidimensional Equally-Distributed Dependencies via Quasi-Cliques on Hypergraphs”. In: *IEEE Transactions on Emerging Topics in Computing* Special Section on Emerging Trends and Advances in Graph-based Methods and Applications (2022), pp. 1–16. ISSN: 2168-6750. DOI: 10.1109/TETC.2022.3198252.

[Bai07] John Bailey et al. “Search engine overlaps: Do they agree or disagree?” In: *Proceedings - ICSE 2007 Workshops: Second International Workshop on Realising Evidence-Based Software Engineering, REBSE’07*. 2007. ISBN: 0-7695-2962-3. DOI: 10.1109/REBSE.2007.4.

[Bau12] J P Baud et al. “The LHCb Data Management System”. In: *Journal of Physics: Conference Series* 396.3 (2012), p. 32023. DOI: 10.1088/1742-6596/396/3/032023.

[BB10] Nicholas M. Ball and Robert J. Brunner. “Data Mining And Machine Learning In Astronomy”. In: *International Journal of Modern Physics D* 19.07 (2010), pp. 1049–1106. DOI: 10.1142/S0218271810017160.

[BHB07] Mauro Brunato, Holger H Hoos, and Roberto Battiti. “On effectively finding maximal quasi-cliques in graphs”. In: *International conference on learning and intelligent optimization*. 2007, pp. 41–55. DOI: 10.1007/978-3-540-92695-5_4.

[BHS09] Gordon Bell, Tony Hey, and Alex Szalay. “Beyond the data deluge”. In: *Science (New York, N.Y.)* 323.5919 (2009). Publisher: American Association for the Advancement of Science, pp. 1297–1298. DOI: 10.1126/science.1170411.

[Bi22] Wenyuan Bi et al. "Learning-Based Optimization for Online Approximate Query Processing". In: *Database Systems for Advanced Applications*. Ed. by Arnab Bhattacharya et al. Cham: Springer International Publishing, 2022, pp. 96–103. ISBN: 978-3-031-00123-9. DOI: 10.1007/978-3-031-00123-9_7.

[Bor00] Kirk D. Borne. "Data Mining in Astronomical Databases". In: *Mining the Sky*. arXiv: astro-ph/0010583. Springer, 2000, pp. 671–673. DOI: 10.1007/10849171_88.

[Bre07] Pearl Brereton et al. "Lessons from applying the systematic literature review process within the software engineering domain". In: *Journal of Systems and Software* 80.4 (2007). ISBN: 0164-1212, pp. 571–583. ISSN: 01641212. DOI: 10.1016/j.jss.2006.07.009.

[Bre13] Alain Bretto. "Applications of hypergraph theory: A brief overview". In: *Hypergraph theory: An introduction*. Heidelberg: Springer International Publishing, 2013, pp. 111–116. ISBN: 978-3-319-00080-0. DOI: 10.1007/978-3-319-00080-0_7.

[Bre84] Leo Breiman et al. *Classification And Regression Trees*. 1984. DOI: 10.1201/9781315139470.

[BSC13] Leilani Battle, Michael Stonebraker, and Remco Chang. "Dynamic reduction of query result sets for interactive visualizaton". In: *2013 IEEE International Conference on Big Data*. 2013, pp. 1–8. ISBN: 978-1-4799-1292-6. DOI: 10.1109/BigData.2013.6691708.

[Bud08] David Budgen et al. "Using Mapping Studies in Software Engineering". In: *Proceedings of Psychology of Programming Interest Group*. Vol. 2. 2008, pp. 195–204. URL: https://www.ppig.org/files/2008-PPIG-20th-budgen.pdf.

[CFP84] Marco A. Casanova, Ronald Fagin, and Christos H. Papadimitriou. "Inclusion dependencies and their interaction with functional dependencies". In: *Journal of Computer and System Sciences* 28.1 (1984), pp. 29–59. ISSN: 0022-0000. DOI: https://doi.org/10.1016/0022-0000(84)90075-8.

[Che09] Bee-Chung Chen et al. “Privacy-preserving data publishing”. In: *Foundations and Trends in Databases* 2 (Jan. 2009), pp. 1–167. DOI: 10.1561/1900000008.

[Chu86] Wai Fong Chua. “Radical Developments in Accounting Thought”. In: *The Accounting Review* (1986). ISBN: 00014826. ISSN: 0001-4826. DOI: 10.2307/247360.

[CKJ17] Javad Chamanara, Birgitta König-Ries, and H. V. Jagadish. “QUIS: In-situ heterogeneous data source querying”. In: *Proc. VLDB Endow.* 10.12 (Aug. 2017). Number of pages: 4 Publisher: VLDB Endowment, pp. 1877–1880. ISSN: 2150-8097. DOI: 10.14778/3137765.3137798.

[CT19] Felipe Campelo and Fernanda Takahashi. “Sample size estimation for power and accuracy in the experimental comparison of algorithms”. In: *Journal of Heuristics* 25.2 (Apr. 2019), pp. 305–338. ISSN: 1572-9397. DOI: 10.1007/s10732-018-9396-7.

[Cyb16] European Union Agency for Cybersecurity et al. *Privacy by design in big data : an overview of privacy enhancing technologies in the era of big data analytics*. European Network and Information Security Agency, 2016. DOI: doi/10.2824/641480.

[DB12] Clive L Dym and David C Brown. *Engineering Design: Representation and Reasoning*. 2nd. Issue: 1990. Cambridge University Press, 2012. ISBN: 978-0-521-51429-3. DOI: 10.1017/CBO9781139031813.

[Des20] G. Desprez et al. “Euclid preparation. X. The euclid photometric-redshift challenge”. In: *Astronomy & Astrophysics* 644 (Dec. 2020), A31. DOI: 10.1051/0004-6361/202039403.

[DG17] Dheeru Dua and Casey Graff. *UCI machine learning repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[DLP02] Fabien De Marchi, Stéphane Lopes, and Jean-Marc Petit. “Efficient algorithms for mining inclusion dependencies”. In: *International conference on extending database technology*. 2002, pp. 464–476. DOI: 10.1007/3-540-45876-X_30.

[Don17] Bin Dong et al. “ArrayUDF: User-defined scientific data analysis on arrays”. In: *Proceedings of the 26th international symposium on high-performance parallel and distributed computing*. HPDC ’17. Number of pages: 12 Place: Washington, DC, USA. New York, NY, USA: Association for Computing Machinery, 2017, pp. 53–64. ISBN: 978-1-4503-4699-3. DOI: 10.1145/3078597.3078599.

[DP03] Fabien De Marchi and J-M Petit. “Zigzag: a new algorithm for mining large inclusion dependencies in databases”. In: *Third IEEE international conference on data mining*. 2003, pp. 27–34. DOI: 10.1109/ICDM.2003.1250899.

[DP79] Gordon Bitter Davis and Clyde Alvin Parker. *Writing the doctoral dissertation: A systematic approach*. Barron’s Educational Series, 1979.

[Dür19] Falco Dürsch et al. “Inclusion dependency discovery: An experimental evaluation of thirteen algorithms”. In: *Proceedings of the 28th ACM international Conference on Information and Knowledge Management*. CIKM ’19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 219–228. ISBN: 978-1-4503-6976-3. DOI: 10.1145/3357384.3357916.

[Edw27] Edwin B. Wilson. “Probable inference, the law of succession, and statistical inference”. In: *Journal of the American Statistical Association* 22.158 (1927), pp. 209–212. DOI: 10.1080/01621459.1927.10502953.

[EO13] European Organization For Nuclear Research and OpenAIRE. *Zenodo*. en. 2013. DOI: 10.25495/7GXK-RD71.

[FPS96] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. “Knowledge Discovery and Data Mining: Towards a Unifying Framework”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. Series Title: KDD’96. AAAI Press, 1996, pp. 82–88. URL: http://dl.acm.org/citation.cfm?id=3001460.3001477.

[Fri03] Jerome Friedman. “On multivariate goodness-of-fit and two-sample testing”. In: *Proceedings of the Conference on Statistical Problems in Particle Physics, Astrophysics, and Cosmology* 1 (2003), pp. 311–313. URL: https://www.slac.stanford.edu/econf/C030908/papers/SLAC-R-703.pdf#page=321.

[Gau16] Anna Gaulton et al. “The ChEMBL database in 2017”. In: *Nucleic Acids Research* 45.D1 (Nov. 2016), pp. D945–D954. ISSN: 0305-1048. DOI: 10.1093/nar/gkw1074.

[Ghe19] Youcef Gheraibia et al. “Safety + AI: A novel approach to update safety models using artificial intelligence”. In: *IEEE Access* 7 (2019), pp. 135855–135869. DOI: 10.1109/ACCESS.2019.2941566.

[GL94] E. G. Guba and Y. S Lincoln. “Competing Paradigms in Qualitative Research”. In: *Handbook of qualitative research*. CitationKey: Guba1994. 1994.

[Gra02] Jim Gray et al. “Data Mining the SDSS SkyServer Database”. In: *Distributed Data and Structures 4: Records of the 4th International Meeting* (2002), pp. 189–210. DOI: 10.48550/arXiv.cs/0202014.

[Gra97] Jim Gray et al. “Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals”. In: *Data Mining and Knowledge Discovery* (1997). ISSN: 13845810. DOI: 10.1023/A:1009726021843.

[Gre12] Arthur Gretton et al. “A Kernel Two-Sample Test”. In: *Journal of Machine Learning Research* 13.25 (2012), pp. 723–773. URL: http://jmlr.org/papers/v13/gretton12a.html.

[Gub90] Egon G. Guba. “The alternative paradigm dialog”. In: *The paradigm dialog* (1990). ISSN: 03186431.

[HA93] Press William H and Teukolsky Saul A. *Numerical recipes in fortran: the art of scientific computing*. 1993.

[Han16] Rui Han et al. “AccuracyTrader: Accuracy-Aware Approximate Processing for Low Tail Latency and High Result Accuracy in Cloud Online Services”. In: *The 45th International Conference on Parallel Processing* 8 (Aug. 2016), pp. 278–287. ISSN: 01903918. DOI: 10.1109/ICPP.2016.39.

[HDF] HDF. *HDF Group*. URL: https://www.hdfgroup.org/.

[Hen88] Norbert Henze. "A multivariate two-sample test based on the number of nearest neighbor type coincidences". In: *The Annals of Statistics* 16.2 (1988), pp. 772–783. ISSN: 00905364. URL: http://www.jstor.org/stable/2241756.

[HMS21] Marc Hallin, Gilles Mordant, and Johan Segers. "Multivariate goodness-of-fit tests based on Wasserstein distance". In: *Electronic Journal of Statistics* 15.1 (2021), pp. 1328–1371. DOI: 10.1214/21-EJS1816.

[HNK17] Donghyoung Han, Yoon Min Nam, and Min Soo Kim. "A distributed in-situ analysis method for large-scale scientific data". In: *2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017*. Feb. 2017, pp. 69–75. ISBN: 978-1-5090-3015-6. DOI: 10.1109/BIGCOMP.2017.7881718.

[Hod58] John L Hodges. "The significance probability of the Smirnov two-sample test". In: *Arkiv för Matematik* 3.5 (1958), pp. 469–486.

[HTT09] Tony Hey, Stewart Tansley, and Kristin Tolle. "Jim Gray on eScience: A Transformed Scientific Method". In: *The fourth paradigm: Data-intensive scientific discovery*. Microsoft Research, Oct. 2009, pp. xvii–xxxi. ISBN: 978-0-9825442-0-4. URL: https://www.microsoft.com/en-us/research/publication/fourth-paradigm-data-intensive-scientific-discovery/.

[Idr11] Stratos Idreos et al. "Here are my Data Files. Here are my Queries. Where are my Results?" In: *CIDR '11: Fifth Biennial Conference on Innovative Data Systems Research* (2011), pp. 57–68. URL: https://infoscience.epfl.ch/record/161489.

[IKM07] Stratos Idreos, Martin L. Kersten, and Stefan Manegold. "Updating a cracked database". In: *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*. ISSN: 07308078. 2007, p. 413. ISBN: 978-1-59593-686-8. DOI: 10.1145/1247480.1247527.

[IPC15] Stratos Idreos, Olga Papaemmanouil, and Surajit Chaudhuri. "Overview of Data Exploration Techniques". In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data - SIGMOD '15*. ISSN: 07308078. 2015, pp. 277–281. ISBN: 978-1-4503-2758-9. DOI: 10.1145/2723372.2731084.

[JN20] Lan Jiang and Felix Naumann. "Holistic primary key and foreign key detection". In: *Journal of Intelligent Information Systems* 54.3 (June 2020), pp. 439–461. ISSN: 1573-7675. DOI: 10.1007/s10844-019-00562-z.

[JS07] Magne Jorgensen and Martin Shepperd. "A Systematic Review of Software Development Cost Estimation Studies". In: *IEEE Transactions on Software Engineering* 33.1 (2007), pp. 33–53. ISSN: 0098-5589. DOI: 10.1109/TSE.2007.256943.

[JW12] Samireh Jalali and Claes Wohlin. "Systematic Literature Studies: Database Searches vs. Backward Snowballing". In: *ESEM'12: Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. ISSN: 1938-6451. 2012, pp. 29–38. ISBN: 978-1-4503-1056-7. DOI: 10.1145/2372251.2372257.

[Kam14] Niranjan Kamat et al. "Distributed and interactive cube exploration". In: *2014 IEEE 30th International Conference on Data Engineering*. ISSN: 10844627. 2014, pp. 472–483. ISBN: 978-1-4799-2554-4. DOI: 10.1109/ICDE.2014.6816674.

[Kan21] Daniel Kang et al. "Accelerating approximate aggregation queries with expensive predicates". In: *Proc. VLDB Endow.* 14.11 (Oct. 2021). Number of pages: 14 Publisher: VLDB Endowment, pp. 2341–2354. ISSN: 2150-8097. DOI: 10.14778/3476249.3476285.

[Kan92] Martti Kantola et al. "Discovering functional and inclusion dependencies in relational databases". In: *International Journal of Intelligent Systems* 7.7 (1992), pp. 591–607. DOI: 10.1002/int.4550070703.

[Kar14] Manos Karpathiotakis et al. “Adaptive query processing on RAW data”. In: *Proceedings of the VLDB Endowment* 7.12 (Aug. 2014), pp. 1119–1130. ISSN: 21508097. DOI: 10.14778/2732977.2732986.

[KB13] Barbara Kitchenham and Pearl Brereton. “A systematic review of systematic review process research in software engineering”. In: *Information and Software Technology* 55.12 (2013). ISBN: 09505849, pp. 2049–2075. ISSN: 09505849. DOI: 10.1016/j.infsof.2013.07.010.

[KC07] Barbara Kitchenham and S Charters. “Guidelines for performing Systematic Literature Reviews in Software Engineering”. In: *Engineering* 2 (2007), p. 1051.

[Ker11] M Kersten et al. “The Researcher’s Guide to the Data Deluge: Querying a Scientific Database in Just a Few Seconds”. In: *Proceedings of the VLDB Endowment*. ISSN: 21508097. 2011, p. 1474. DOI: 10.1145/1409360.1409380.

[Kho10] Nodira Khoussainova et al. “SnipSuggest: Context-aware autocompletion for SQL”. In: *Proceedings of the VLDB Endowment* 4.1 (Oct. 2010), pp. 22–33. ISSN: 2150-8097. DOI: 10.14778/1880172.1880175.

[KHZ17] Jakub Kuzilek, Martin Hlosta, and Zdenek Zdrahal. “Open University Learning Analytics dataset”. In: *Scientific Data* 4.1 (Nov. 2017), p. 170171. ISSN: 2052-4463. DOI: 10.1038/sdata.2017.171.

[Kim15] Sung-Soo Kim et al. “Flying KIWI: Design of Approximate Query Processing Engine for Interactive Data Analytics at Scale”. In: *Proceedings of the 2015 International Conference on Big Data Applications and Services*. Series Title: BigDAS ’15. New York, NY, USA: ACM, 2015, pp. 206–207. ISBN: 978-1-4503-3846-2. DOI: 10.1145/2837060.2837096.

[Kim21] Ilmun Kim et al. “Classification accuracy as a proxy for two-sample testing”. In: *The Annals of Statistics* 49.1 (2021). Publisher: Institute of Mathematical Statistics, pp. 411–434.

[Kir20] Matthias Kirchler et al. “Two-sample testing using deep learning”. In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. 2020, pp. 1387–1398. URL: https://proceedings.mlr.press/v108/kirchler20a.html.

[KKK98] Samuel Kaski, Jari Kangas, and Teuvo Kohonen. “Bibliography of self-organizing map (SOM) papers: 1981–1997”. In: *Neural computing surveys* 1.3&4 (1998), pp. 1–176. URL: http://cis.legacy.ics.tkk.fi/research/som-bibl/vol1_4.pdf.

[Koe02] Andreas Koeller. “Integration of heterogeneous databases: Discovery of meta-information and maintenance of schema-restructuring views”. PhD thesis. Worcester Polytechnic Institute, 2002. URL: https://digital.wpi.edu/concern/etds/00000005c?locale=en.

[Koh13] Teuvo Kohonen. “Essentials of the self-organizing map”. In: *Neural Networks* 37 (2013), pp. 52–65. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2012.09.018.

[Koh82] Teuvo Kohonen. “Self-organized formation of topologically correct feature maps”. In: *Biological cybernetics* 43.1 (1982), pp. 59–69. DOI: 10.1007/BF00337288.

[KPN22] Jan Kossmann, Thorsten Papenbrock, and Felix Naumann. “Data dependencies for query optimization: a survey”. In: *The VLDB Journal* 31.1 (Jan. 2022), pp. 1–22. ISSN: 0949-877X. DOI: 10.1007/s00778-021-00676-3.

[KR03] Andreas Koeller and Elke A Rundensteiner. “Discovery of high-dimensional inclusion dependencies”. In: *Proceedings 19th international conference on data engineering (cat. No. 03CH37405)*. 2003, pp. 683–685. DOI: 10.1109/ICDE.2003.1260834.

[KR06] Andreas Koeller and Elke A Rundensteiner. “Heuristic strategies for the discovery of inclusion dependencies and other patterns”. In: *Journal on Data Semantics V*. Springer, 2006, pp. 185–210. ISBN: 978-3-540-31427-1. DOI: 10.1007/11617808_7.

[Kru17] Sebastian Kruse et al. “Fast approximate discovery of inclusion dependencies”. In: *Datenbanksysteme für business, technologie und web (BTW 2017)*. Ed. by Bernhard Mitschang et al. Gesellschaft für Informatik, Bonn, 2017, pp. 207–226. URL: https://dl.gi.de/handle/20.500.12116/629.

[Lak18] Sriram Lakshminarasimhan et al. “Scalable in situ scientific data encoding for analytical query processing”. In: *Proceedings of the 22nd international symposium on high-performance parallel and distributed computing*. HPDC ’13. Number of pages: 12 Place: New York, New York, USA. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–12. ISBN: 978-1-4503-1910-2. DOI: 10.1145/2462902.2465527.

[Law15] Amanda Lawrence et al. “Collecting the Evidence: Improving Access to Grey Literature and Data for Public Policy and Practice”. In: *Australian Academic & Research Libraries* 46.4 (2015). Publisher: Routledge, pp. 229–249. DOI: 10.1080/00048623.2015.1081712.

[Liu20] Feng Liu et al. “Learning deep kernels for non-parametric two-sample tests”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of machine learning research. PMLR, July 2020, pp. 6316–6326. URL: https://proceedings.mlr.press/v119/liu20m.html.

[LO17] David Lopez-Paz and Maxime Oquab. “Revisiting classifier two-sample tests”. In: *International Conference on Learning Representations*. 2017. URL: https://hal.science/hal-01862834/.

[Loh09] Sharon Lohr. *Sampling: design and analysis*. Nelson Education, 2009.

[Mac05] Maggie MacLure. “’Clarity bordering on stupidity’: where’s the quality in systematic review?” In: *Journal of Education Policy* 20.4 (2005), pp. 393–416. ISSN: 0268-0939. DOI: 10.1080/02680930500131801.

[Mar23] Stavros Maroulis et al. “Resource-aware adaptive indexing for in situ visual exploration and analytics”. In: *The VLDB Journal* 32.1 (Jan. 2023), pp. 199–227. ISSN: 0949-877X. DOI: 10.1007/s00778-022-00739-z.

[MCS21] Venkata Vamsikrishna Meduri, Kanchan Chowdhury, and Mohamed Sarwat. “Evaluation of machine learning algorithms in predicting the next SQL query from the future”. In: *ACM Transactions on Database Systems* 46.1 (Mar. 2021). Number of pages: 46 Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0362-5915. DOI: 10.1145/3442338.

[Mil68] Robert B Miller. “Response Time in Man-computer Conversational Transactions”. In: *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*. Series Title: AFIPS ’68 (Fall, part I). New York, NY, USA: ACM, 1968, pp. 267–277. DOI: 10.1145/1476589.1476628.

[Moz17] Barzan Mozafari. “Approximate Query Engines : Commercial Challenges and Research Opportunities”. In: *Proceedings of the 2017 ACM International Conference on Management of Data*. Series Title: SIGMOD ’17 ISSN: 07308078. New York, NY, USA: ACM, 2017, pp. 5–8. ISBN: 978-1-4503-4197-4. DOI: 10.1145/3035918.3056098.

[MT17] Antonio Maccioni and Riccardo Torlone. “Crossing the finish line faster when paddling the data lake with KAYAK”. In: *Proc. VLDB Endow.* 10.12 (Aug. 2017). Number of pages: 4 Publisher: VLDB Endowment, pp. 1853–1856. ISSN: 2150-8097. DOI: 10.14778/3137765.3137792.

[Oak03] Ann Oakley. “Research Evidence, Knowledge Management and Educational Practice: early lessons from a systematic approach”. In: *London Review of Education* 1.1 (2003). ISBN: 1474-8460, pp. 21–33. ISSN: 1474-8460. DOI: 10.1080/14748460306693.

[Oat06] Briony J Oates. “Researching Information Systems and Computing”. In: 37 (2006). ISBN: 3175723993, p. 341. ISSN: 1520510X. DOI: 10.1016/j.ijinfomgt.2006.07.009.

[Oga11] Eduardo Ogasawara et al. “An algebraic approach for data-centric scientific workflows”. In: *Proceedings of the VLDB Endowment* 4.11 (2011), pp. 1328–1339. URL: https://hal.inria.fr/hal-00640431.

[OKK03] Merja Oja, Samuel Kaski, and Teuvo Kohonen. “Bibliography of self-organizing map (SOM) papers: 1998-2001 addendum”. In: *Neural Computing Surveys* 3 (Feb. 2003), pp. 1–156. URL: http://cis.legacy.ics.tkk.fi/research/som-bibl/NCS_vol3_1.pdf.

[Pal15] T Palpanas. “Data series management: The road to big sequence analytics”. In: *ACM SIGMOD Record* 44.2 (2015), pp. 47–52. ISSN: 01635808 (ISSN). DOI: 10.1145/2814710.2814719.

[Pan17] Zhifei Pang et al. “FlashView: An interactive visual explorer for raw data”. In: *Proc. VLDB Endow.* 10.12 (Aug. 2017). Number of pages: 4 Publisher: VLDB Endowment, pp. 1869–1872. ISSN: 2150-8097. DOI: 10.14778/3137765.3137796.

[Pap15] Thorsten Papenbrock et al. “Data profiling with metanome”. In: *Proceedings of the VLDB Endowment* 8.12 (2015). Publisher: VLDB Endowment, pp. 1860–1863.

[PBW84] G Pahl, W Beitz, and K Wallace. *Engineering design*. Design Council, 1984. ISBN: 978-0-85072-124-9. URL: https://books.google.ch/books?id=DYpRAAAAMAAJ.

[Ped11] F. Pedregosa et al. “Scikit-learn: Machine learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830. URL: https://dl.acm.org/doi/abs/10.5555/1953048.2078195.

[Pep14] Alberto Pepe et al. “How do astronomers share data? Reliability and persistence of datasets linked in AAS publications and a qualitative study of data practices among US astronomers”. In: *PLOS ONE* 9.8 (Aug. 2014). Publisher: Public Library of Science, pp. 1–11. DOI: 10.1371/journal.pone.0104798.

[Pér08] Fernando Pérez-Cruz. "Kullback-Leibler divergence estimation of continuous distributions". In: *2008 IEEE international symposium on information theory*. 2008, pp. 1666–1670. DOI: 10.1109/ISIT.2008.4595271.

[Pet07] Kai Petersen et al. "Systematic Mapping Studies in Software Engineering". In: *12th International Conference on Evaluation and Assessment in Software Engineering* 17 (2007). ISBN: 0-7695-2555-5, p. 10. ISSN: 02181940. DOI: 10.1142/S0218194007003112.

[PHK06] Matti Polla, Timo Honkela, and Teuvo Kohonen. "Bibliography of self-organizing map (SOM) papers: 2002–2005". In: *TKK Reports in Information and Computer Science* (Jan. 2006). URL: http://users.ics.aalto.fi/tho/online-papers/TKK-ICS-R23.pdf.

[Pia91] Gregory Piatetsky-Shapiro. "Knowledge Discovery in Real Databases: A Report on the International Joint Conference on Artificial Intelligence 89 Workshop". In: *AI Magazine* 11.5 (Jan. 1991), pp. 68–70. ISSN: 0738-4602. URL: http://dl.acm.org/citation.cfm?id=124898.124915.

[Pon15] Nancy Pontika et al. "Fostering open science to research using a taxonomy and an eLearning portal". In: *iKnow: 15th international conference on knowledge technologies and data driven business*. 2015. DOI: 10.1145/2809563.2809571.

[Ram15] Aaditya Ramdas et al. "On the decreasing power of kernel and distance based nonparametric hypothesis tests in high dimensions". In: *Proceedings of the AAAI conference on Artificial Intelligence*. Vol. 29. Number: 1. 2015. DOI: 10.1609/aaai.v29i1.9692.

[Rei99] Thomas Reinartz. *Focusing Solutions for Data Mining Analytical Studies and Experimental Results in Real-World Domains*. Publication Title: Case Analysis. Berlin, Heidelberg: Springer-Verlag, 1999. ISBN: 3-540-66429-7.

[Ron20] Kexin Rong et al. "Approximate partition selection for big-data workloads using summary statistics". In: *Proc. VLDB Endow.* 13.12 (Sept. 2020). Number of pages: 14 Publisher: VLDB Endowment, pp. 2606–2619. ISSN: 2150-8097. DOI: 10.14778/3407790.3407848.

[Ros09] Alexandra Rostin et al. “A machine learning approach to foreign key discovery.” In: *12th International Workshop on the Web and Databases*. 2009. URL: https://hpi.de/fileadmin/user_upload/fachgebiete/naumann/publications/2009/WebDB09_crc.pdf.

[Ros19] Jonathan D Rosenblatt et al. “Better-than-chance classification for signal detection”. In: *Biostatistics (Oxford, England)* 22.2 (2019), pp. 365–380. DOI: 10.1093/biostatistics/kxz035.

[RRS21] Nir Regev, Lior Rokach, and Asaf Shabtai. “Approximating aggregated SQL queries with LSTM networks”. In: *2021 international joint conference on neural networks (IJCNN)*. ISSN: 2161-4407. July 2021, pp. 1–8. DOI: 10.1109/IJCNN52387.2021.9533974.

[RTV16] Rasmus Rothe, Radu Timofte, and Luc Van Gool. “Deep expectation of real and apparent age from a single image without facial landmarks”. In: *International Journal of Computer Vision* 126.2 (2016), pp. 144–157. DOI: 10.1007/s11263-016-0940-3.

[RW79] Ronald H Randles and Douglas A Wolfe. *Introduction to the theory of nonparametric statistics*. Report. John Wiley, 1979.

[Sal05] Jarkko Salojärvi et al. “Inferring relevance from eye movements: Feature extraction”. In: *Neural Information Processing Systems*. Whistler, BC, Canada, 2005, p. 45. URL: http://research.ics.aalto.fi/events/inips2005/inips2005proceedings.pdf#page=45.

[SC05] M Stonebraker and U Cetintemel. “One size fits all: an idea whose time has come and gone”. In: *21st International Conference on Data Engineering (ICDE’05)*. ISSN: 1063-6382. 2005, pp. 2–11. DOI: 10.1109/ICDE.2005.1.

[SC21] Hoseung Song and Hao Chen. “A fast and effective large-scale two-sample test based on kernels”. In: (2021). DOI: 10.48550/arXiv.2110.03118.

[Sch86] Mark F Schilling. “Multivariate two-sample tests based on nearest neighbors”. In: *Journal of the American Statistical Association* 81.395 (1986), pp. 799–806. URL: https://www.jstor.org/stable/2241756.

[SD11] Leandro A Silva and Emilio Del-Moral-Hernandez. "A SOM combined with KNN for classification task". In: *The 2011 International Joint Conference on Neural Networks*. 2011, pp. 2368–2373. DOI: 10.1109/IJCNN.2011.6033525.

[SDSa] SDSS. *Sample SQL Queries*. URL: http://cas.sdss.org/dr15/en/help/docs/realquery.aspx.

[SDSb] SDSS. *SQL Logs*. URL: http://skyserver.sdss.org/log/en/traffic/sql.asp.

[SDT18] Seyed-Vahid Sanei-Mehri, Apurba Das, and Srikanta Tirthapura. "Enumerating top-k quasi-cliques". In: *2018 IEEE International Conference on Big Data*. 2018, pp. 1107–1112. DOI: 10.1109/BigData.2018.8622352.

[SH72] WB Smith and RR Hocking. "Algorithm as 53: Wishart variate generator". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 21.3 (1972). Publisher: JSTOR, pp. 341–345.

[She00] Colin Shearer et al. "The CRISP-DM model: The New Blueprint for Data Mining". In: *Journal of Data Warehousing* 5.4 (2000), pp. 13–22. ISSN: 1092-6208.

[Sid17] Aisha Siddiqa et al. "On the analysis of big data indexing execution strategies". In: *Journal of Intelligent and Fuzzy Systems* 32.5 (2017), pp. 3259–3271. ISSN: 18758967. DOI: 10.3233/JIFS-169269.

[Sil16a] Vítor Silva et al. "Analyzing related raw data files through dataflows". In: *Concurrency and Computation: Practice and Experience* 28.8 (2016), pp. 2528–2545. ISSN: 1532-0634. DOI: 10.1002/cpe.3616.

[Sil16b] Vítor Silva et al. "Raw data queries during data-intensive parallel workflow execution". In: *Future Generation Computer Systems* (2016). ISSN: 0167739X. DOI: 10.1016/j.future.2017.01.016.

[Sjø05] Dag I.K. Sjøberg et al. "A survey of controlled experiments in software engineering". In: *IEEE Transactions on Software Engineering* 31.9 (2005). ISBN: 0-7803-9507-7, pp. 733–753. ISSN: 00985589. DOI: 10.1109/TSE.2005.97.

[Sor15] Jacopo Soriano. “Bayesian methods for two-sample comparison”. PhD thesis. Duke University, 2015. URL: https://dukespace.lib.duke.edu/dspace/handle/10161/9859.

[SPF22] Hamid Shahrivari, Odysseas Papapetrou, and George Fletcher. “Workload prediction for adaptive approximate query processing”. In: *2022 IEEE international conference on big data (big data)*. Dec. 2022, pp. 217–222. DOI: 10.1109/BigData55660.2022.10020614.

[SSS08] Forrest Shull, Janice Singer, and Dag I.K. Sjøberg. *Guide to advanced empirical software engineering*. ISSN: 1098-6596. 2008. ISBN: 978-1-84800-043-8. DOI: 10.1007/978-1-84800-044-5.

[Sto09] Michael Stonebraker. “Requirements for Science Data Bases and SciDB”. In: *4th Biennial Conference on Innovative Data Systems Research CIDR’09* (2009), p. 173184. DOI: 10.1.1.145.1567.

[Sto11] Michael Stonebraker et al. “The architecture of SciDB”. In: *Lecture Notes in Computer Science* 6809 LNCS (2011). Publisher: Springer Berlin Heidelberg ISBN: 9783642223501, pp. 1–16. ISSN: 03029743. DOI: 10.1007/978-3-642-22351-8_1.

[Thi20] Saravanan Thirumuruganathan et al. “Approximate query processing for data exploration using deep generative models”. In: *2020 IEEE 36th international conference on data engineering (ICDE)*. ISSN: 2375-026X. Apr. 2020, pp. 1309–1320. DOI: 10.1109/ICDE48307.2020.00117.

[Tia14] Yongchao Tian et al. “DiNoDB: Efficient Large-Scale Raw Data Analytics”. In: *Proceedings of the First International Workshop on Bringing the Value of "Big Data" to Users (Data4U 2014)*. Series Title: Data4U ’14. New York, NY, USA: ACM, 2014, pp. 1–6. ISBN: 978-1-4503-3186-9. DOI: 10.1145/2658840.2658841.

[Tic95] Walter F Tichy et al. “Experimental evaluation in computer science: A quantitative study”. In: *Journal of Systems and Software* 28.1 (1995), pp. 9–18. ISSN: 0164-1212. DOI: https://doi.org/10.1016/0164-1212(94)00111-Y.

[Tom81] Ioan Tomescu. “Le nombre maximum de cliques et de recouvrements par cliques des hypergraphes chromatiques complets”. In: *Discrete Mathematics* 37.2 (1981), pp. 263–277. ISSN: 0012-365X. DOI: 10.1016/0012-365X(81)90225-9.

[Ult07] Alfred Ultsch. “Emergence in self organizing feature maps”. In: *International workshop on self-organizing maps: Proceedings (2007)*. 2007. DOI: 10.2390/biecoll-wsom2007-114.

[UM05] Alfred Ultsch and Fabian Mörchen. *ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM*. Report. University of Marburg, 2005. URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=916f0fdb30a54fc5d9e5e9c69324ae2a8691

[Uno10] Takeaki Uno. “An efficient algorithm for solving pseudo clique enumeration problem”. In: *Algorithmica* 56.1 (2010). Publisher: Springer, pp. 3–16. DOI: 10.1007/s00453-008-9238-3.

[Vil99] Th. Villmann. “Topology preservation in self-organizing maps”. In: *Kohonen maps*. Ed. by Erkki Oja and Samuel Kaski. Amsterdam: Elsevier Science B.V., 1999, pp. 279–292. ISBN: 978-0-444-50270-4. DOI: 10.1016/B978-044450270-4/50022-X.

[WCA15] Yi Wang, Linchuan Chen, and Gagan Agrawal. “Supporting online analytics with user-defined estimation and early termination in a MapReduce-like framework”. In: *Proceedings of the 2015 International Workshop on Data-Intensive Scalable Computing Systems*. Series Title: DISCS ’15. New York, NY, USA: ACM, 2015, pp. 1–8. ISBN: 978-1-4503-3993-3. DOI: 10.1145/2831244.2831247.

[WH15] Qinghua Wu and Jin-Kao Hao. “A review on algorithms for maximum clique problems”. In: *European Journal of Operational Research* 242.3 (2015), pp. 693–709. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2014.09.064.

[Wie06] Roel Wieringa et al. “Requirements engineering paper classification and evaluation criteria: A proposal and a discussion”. In: *Requirements Engineering* 11.1 (2006). ISBN: 0947-3602, pp. 102–107. ISSN: 09473602. DOI: 10.1007/s00766-005-0021-6.

[Wil45] Frank Wilcoxon. “Individual comparisons by ranking methods”. In: *Biometrics Bulletin* 1.6 (1945), pp. 80–83. ISSN: 00994987. DOI: 10.2307/3001968.

[Wit17] Peter Wittek et al. “Somoclu: An efficient parallel library for self-organizing maps”. In: *Journal of Statistical Software* 78.9 (2017). DOI: 10.18637/jss.v078.i09.

[Wu08] Wensheng Wu et al. “Discovering topical structures of databases”. In: *Proceedings of the 2008 ACM SIGMOD international conference on management of data*. SIGMOD ’08. New York, NY, USA: Association for Computing Machinery, 2008, pp. 1019–1030. ISBN: 978-1-60558-102-6. DOI: 10.1145/1376616.1376717.

[Wu09] K. Wu et al. “FastBit: Interactively searching massive data”. In: *Journal of Physics: Conference Series*. ISSN: 17426596. 2009. DOI: 10.1088/1742-6596/180/1/012053.

[WW02] Jane Webster and Richard T Watson. “Analyzing the Past to Prepare for the Future: Writing a Literature Review.” In: *MIS Quarterly* 26.2 (2002), pp. xiii–xxiii. URL: https://www.jstor.org/stable/4132319.

[ZGB13] Wojciech Zaremba, Arthur Gretton, and Matthew Blaschko. “B-test: A Non-parametric, Low Variance Kernel Two-sample Test”. In: *Advances in Neural Information Processing Systems* 26 (2013). URL: https://proceedings.neurips.cc/paper/2013/file/a49e9411d64ff53eccfdd09ad10a15b3-Paper.pdf.

[Zha10] Meihui Zhang et al. “On multi-column foreign key discovery”. In: *Proceedings of the VLDB Endowment* 3.1–2 (Sept. 2010), pp. 805–814. ISSN: 2150-8097. DOI: 10.14778/1920841.1920944.

[Zha18] Weijie Zhao et al. “Distributed caching for processing raw arrays”. In: *Proceedings of the 30th international conference on scientific and statistical database management*. SSDBM ’18. Number of pages: 12 Place: Bozen-Bolzano, Italy. New York, NY, USA: Association for Computing Machinery, 2018. ISBN: 978-1-4503-6505-5. DOI: 10.1145/3221269.3221295.

[Zou15] Kostas Zoumpatianos. "Query Workloads for Data Series Indexes". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015). ISBN: 9781450336642, pp. 1603–1612. DOI: 10.1145/2783258.2783382.

[ZW97] Marvin V Zelkowitz and Dolores Wallace. "Experimental validation in software engineering". In: *Information and Software Technology* 39.11 (1997), pp. 735–743. ISSN: 0950-5849. DOI: 10.1016/S0950-5849(97)00025-6.

[ZZ15] Yanxia Zhang and Yongheng Zhao. "Astronomy in the big data era". In: *Data Science Journal* 14 (2015). Publisher: Ubiquity Press. DOI: 10.5334/dsj-2015-011.

Appendix A

PresQ Benchmarking Tools

The repository `MatchBox`[1] contains the Python implementation of PresQ and Find2 used as a reference for the performance comparison shown in chapter 5, together with instructions to reproduce the results.

For facilitating the replicability of the results, the repository contains an `environment.yml` file that allows re-creating a working Python environment with all the requirements installed using `Conda`:

```
conda env create
conda activate matchbox
```

A.1 Benchmarking quasi-clique finding

`benchmark-quasiclique.py` compares the performances of Find2 (baseline) and PresQ when searching for a known quasi-clique generated randomly based on an initial parameterization: rank, cardinality, the ratio of missing edges, the number of nodes not belonging to the quasi-clique, and the ratio of spurious edges. The results are written to a CSV file.

The script snippet 2 shows the relevant extract used to evaluate the capacity of PresQ to recover from missing edges, as reported in figure 5.4.

[1] `https://github.com/ayllon/MatchBox/`

Code 2: Benchmark quasi-clique search with a set of missing ratios. The comments need to be removed.

```
for alpha in 0.05 0.10 0.15 0.20 0.25 0.30; do
  ./bin/benchmark-quasiclique.py \
    --out "results/quasi3.csv" \ # Output CSV
    --rank 3 \                   # 3-hypergraph
    --cardinality 10 20 30 \     # Number of nodes on the quasi-clique
    --additional 0.5 \           # |V| * 0.5 additional nodes
    --repeat 15 \                # Generate 15 different hypergraphs
    --missing-edges ${alpha} \   # Remove $alpha edges from the quasi-clique
    --extra-edges 0 \            # Do not add any spurious edge
    --timeout 300                # Limit execution to 5 minutes
done
```

On the other hand, snippet 3 shows the call used to evaluate the capacity of PresQ to recover from both missing and spurious edges, as reported in figure 5.5. The set of spurious edges S contains *all* possible edges on the hypergraph *not* belonging to the quasi-clique.

Code 3: Benchmark quasi-clique search with a set of additional edges. The comments need to be removed.

```
for beta in 0.2 0.4 0.6 0.8; do
  ./bin/benchmark-quasiclique.py \
    --out "results/quasi3.csv" \ # Output CSV
    --rank 3 \                   # 3-hypergraph
    --cardinality 10 20 30 \     # Number of nodes on the quasi-clique
    --additional 0.5 \           # |V| * 0.5 additional nodes
    --repeat 15 \                # Generate 15 different hypergraphs
    --missing-edges 0.1 \        # Remove 10% of edges from the quasi-clique
    --extra-edges ${beta} \      # Add $beta * |S| spurious edges
    --timeout 1200               # Limit execution to 20 minutes
done
```

A.2 Benchmarking Equally-Distributed Dependency finding

`benchmark.py` is a script that can be used to measure the performance of PresQ and the custom implementation of Find2 over datasets available in any of the supported formats: Flexible Image Transport System (FITS), KEEL `.dat` files, CSV, and SQLite. The script needs a minimum of two relations and supports running multiple parameterizations of PresQ with a single invocation.

The measurements are written to a set of CSV files:

Name	Description
`sampling.csv`	Sampling time
`uind.csv`	Statistics about unary EDD
`bootstrap.csv`	Statistics about initial edges tested and accepted
`find2.csv`	Find2 runs
`findg_{`Λ`}_{`θ`}_{grow}.csv`	PresQ runs with different parameterizations
`{uuid[0:2]}/{uuid}/histogram.txt`	Histogram of the EDD arity found for run `uuid`
`{uuid[0:2]}/{uuid}/nind.txt`	List of EDD found for run `uuid`

Note that the effective value of $\gamma = 1 - \alpha \times \theta$. i.e, `findg_0.05_1.00_1.csv` contains the results for a run of PresQ with $\Lambda = 0.05$, $\gamma = 1 - 0.1 \times 1 = 0.9$ (for $\alpha = 0.1$), and the growing stage enabled.

A.2.1 Find2 vs. PresQ

The snippet 4 shows, as an illustration, how the comparison for the datasets `Ailerons vs Elevators` was executed for table 5.6. For completeness, it also shows the parameters that kept their default values.

Code 4: Benchmark Find2 vs. PresQ over the `Ailerons vs. Elevators` datasets. The comments need to be removed.

```
ID="ailerons_$(date +%Y%m%d)"

./bin/benchmark.py \
    --output-dir "./results/" \ # Write the results to this directory
    --id "${ID}"              \ # Under the given folder name
    --repeat 1000             \ # 1000 randomized runs
    --timeout 3000            \ # With a timeout of 50 minutes for one run
    --sample-size 200         \ # Sample size (DEFAULT)
    -k 3                      \ # Neighbors for the kNN test (DEFAULT)
    --permutations 500        \ # Permutations for the kNN test (DEFAULT)
    --nind-alpha 0.05         \ # Significance level for EDDs (DEFAULT)
    --bootstrap-arity 2       \ # Rank for the initial hypergraph (DEFAULT)
    --lambdas 0.05 0.1        \ # PresQ parameter Lambda (DEFAULT)
    --gammas 1.               \ # gamma = 1 - alpha * 1 (DEFAULT)
     # Significance levels for initial edges/2-EDD (DEFAULT)
    --bootstrap-alpha 0.05 0.1 0.15 \
    "./data/keel/ailerons/ailerons.dat" \
    "./data/keel/elevators/elevators.dat"
```

A.2.2 Scalability

The snippet 5 corresponds to the performance evaluation shown in figure 5.8, where new relations are incrementally added to measure the scalability of PresQ with respect to the number of attributes. One instance of ChemBL contains 80 relations. Note that the script originally expected one file per relation; thus, the `SQLite` back-ends abuses nomenclature for the parameter `-files`.

Code 5: Benchmark performance with respect to the number of columns. The comments need to be removed.

```
ID="chembl_$(date +%Y%m%d)"

for i in {82..160..6}; do       # From 82 to 160 relations
  ./bin/benchmark.py \
    --output-dir "./results/" \ # Write the results to this directory
     --id "${ID}"             \ # Under the given folder name
    --repeat 10               \ # 10 randomized runs
    --timeout 3000            \ # With a timeout of 50 minutes for one run
    --lambdas 0.1             \ # PresQ parameter Lambda
    --bootstrap-alpha 0.05    \ # Significance levels for initial edges/2-EDD
    --no-find2                 \ # Do not run Find2 in this case
    --files $i                 \ # Number of relations
    --sample-size 200          \ # Sample size (DEFAULT)
    -k 3                       \ # Neighbors for the kNN test (DEFAULT)
    --permutations 500         \ # Permutations for the kNN test (DEFAULT)
    --nind-alpha 0.05          \ # Significance level for EDDs (DEFAULT)
    --bootstrap-arity 2        \ # Rank for the initial hypergraph (DEFAULT)
    --gammas 1.                \ # gamma = 1 - alpha * 1 (DEFAULT)
    ./data/chembl/chembl_??.db
done
```

Appendix B

Prototypes Overview

Iterating over a set of ideas is an intrinsic part of both the *Engineering Design Process* and research, and some end being discarded. Nevertheless, knowing about the discarded paths can be helpful [Con20] since they can point to either future lines of work or dead-ends not worth pursuing.

In this appendix, we briefly describe one insight and two prototypes generated during the development of this thesis that did not prove successful.

B.1 Types of Data Correlation

In 2017, [Kra18] proposed a novel idea: an index over a dataset is just a model. This model takes as input a value, or a range of values, and "predicts" the physical location of the corresponding tuples. In this work, Kraska *et al.* propose *learned* alternatives to well-known data structures often used on databases for different access patterns: value range (typically modeled with B-trees), unique values (hash tables), and existence queries (Bloom filters). However, Kraska *et al.* trained models over a single attribute. Extending these types of models might also be possible when multiple files exist.

Thus, based only on the data, we can learn models to predict the value of attributes related to the physical layout, such as the file where a tuple is contained and its offset. This is possible under the assumption that there are three types of *correlations* or *dependencies*:

- **File Correlation** Dependence between values and containing file.
- **Offset Correlation** Dependence between values and offset within the file [Kra18].
- **Tuple Correlation** Dependence between the attribute values of a tuple.

Note that the first two correlations depend on the acquisition method and the third on the measured properties. PresQ builds on the third assumption in order to find EDDs: same set of attributes from the same population follow the same distribution.

We now describe an attempt to leverage the first type of correlation in section B.2, a naive indexing approach on section B.3, and an early prototype for schema matching in section B.4.

B.2 File Correlation

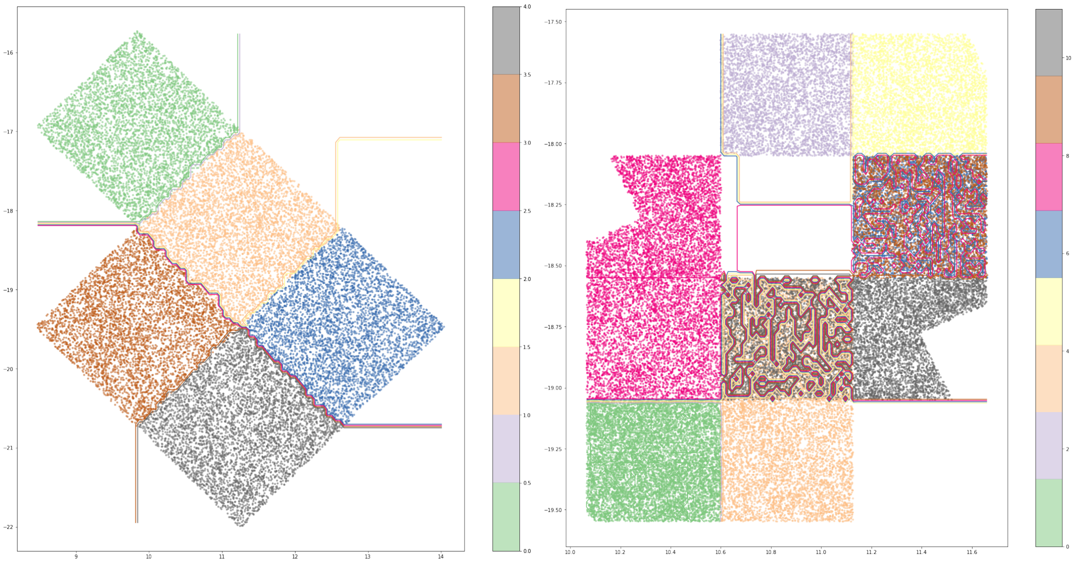
Large datasets may be partitioned into multiple files to facilitate their use. This partitioning can be done as a function of some data features: spatial coordinates, alphanumerical order, temporal order, etc. Therefore, machine learning techniques should be able to model (index) this partitioning without user intervention.

As a proof-of-concept, we took two astronomical catalogs partitioned into multiple files generated by two different processes: simulation and measurement. We can see the indexing as a "classification" task: we want to model the file corresponding to a given data set. For the feature selection, we use Random Forest with 100 trees. Features with higher weight are those used to partition the data. Code 6 shows a snippet of the feature selection. We executed it over a random sample of 5000 tuples per file. We obtained that the sky coordinates — right ascension and declination — were the best features for both the simulation and the measurement catalogs.

```python
# feat_cols contains all features except the file index
classifier = RandomForestClassifier(n_estimators = 100)
classifier.fit(train[feat_cols], train['FILE'])
# Contains the weight for each feature
classifier.feature_importances_
```

Code 6: Feature selection for file correlation.

Since trees are a data structure commonly used for spatial partitioning — KD-Tree, Binary Partition Tree, R-Tree — we trained two decision trees over both catalogs using the sky coordinates as features. We display the resulting “learned” partitioning over the actual data distribution in figure B.1. We can see that in the simulated catalog, the prediction is very accurate. However, the results are suboptimal over the real-world catalog due to overlapping catalogs. This issue could be fixed by training a binary classifier by file.



(a) Without overlap.

(b) With overlap.

Figure B.1: Data sets distributed multiple files based on two spatial coordinates. Each color corresponds to a single file.

B.3 Offset Correlation

We work with the assumption that the data acquisition method influences the data distribution within a file containing raw data. To validate the idea, we obtain random samples from three astronomical catalogs (Cosmic Evolution Survey (Cosmos)[Lai16], SDSS[Abo18], and Kilo Degree Survey (KiDS)[Jon13]), all extracted from the same sky region.

As for the file correlation case, the best features are sky coordinates. The target variable is the page offset within the file, which we obtained via code 7. Rather than training a simple regression model and predicting the exact page, we trained models capable of predicting a page range: *Quantile Regression Forests* [Mei06] and *Gradient Boosting* [Mas99]. The training set comprises 10% of the tuples (1% for Cosmos, given its size) and the test set 20% of the remaining tuples.

The same test were run over a non astronomical data-set[1], for which the best features are `Start_Time`, `End_time` and `Weather_Timestamp`[2].

```
block_size = os.statvfs("/path/to/files").f_bsize
row_size = np.array(table[0:1]).nbytes
page = (np.arange(len(table)) * row_size) // block_size
```

Code 7: Computation of the page offset.

Table B.1 summarizes the results of both models for the four datasets. *Quantile Regression Forests* can be accurate — it predicts the correct range — but the overhead is not negligible — it needs to read a considerable portion of the file per prediction.

B.4 Attribute Correlation

Within a relation, we expect some attributes to be closely correlated. This correlation is inherent to the attributes and the sampled population. Thus,

[1] https://www.kaggle.com/sobhanmoosavi/us-accidents

[2] `ID` is a better feature, but trivial.

Catalog	N.Pages	Method	Time (s)	Accuracy	Precision
Cosmos	78,499	QRF	16.69	94.30%	99.61% (306)
		GB	128.07	63.30%	98.43% (1,236)
SDSS	1,825	QRF	6.86	95.74%	93.47% (119)
		GB	21.28	60.73%	96.85% (57)
KiDS	7,773	QRF	8.62	97.37%	99.64% (28)
		GB	41.22	62.33%	98.37% (127)
Accidents	261,193	QRF	127.56	84.86%	80.89% (49,926)
		GB	9.13	61.00%	85.90% (36,820)

Table B.1: QR vs GBR over different datasets. *Time*: Training time *Accuracy*: Ratio of predicted ranges that contain the queried tuple. *Precision*: 1 - (Average predicted range size divided by the total file size).

two files containing samples for the same attributes from the same underlying population should have a similar distribution.

This correlation is intrinsic to the data and independent of the physical schema of the data — i.e., file layout or attribute names. Alternatively, two datasets with different schemas but containing a similar subset of measures should show the same dependencies between attributes. We can expect to use these dependencies to match schemas[3].

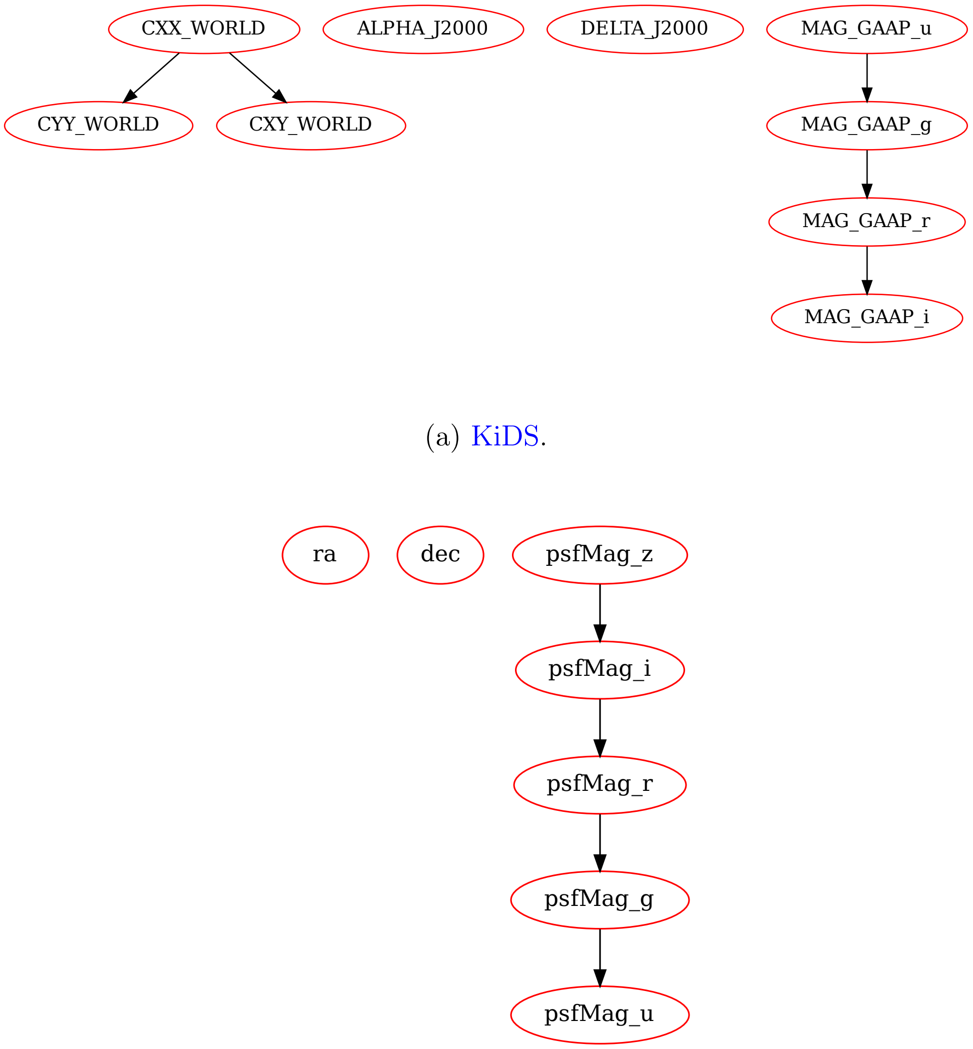
Since Bayesian networks [Pea88] model the data dependencies as a directed acyclic graph, we can expect that graphs trained over different datasets containing the same information should contain isomorphic sub-graphs. For validating the idea, we trained three Bayesian networks using Pomegranate [Sch17] over the catalogs SDSS, KiDS and Cosmos. Figure B.2 shows the resulting graphs for the first two, and figure B.3 for the last.

These graphs are significantly similar for SDSS and KiDS, even on the order of the bands. This match is less evident for Cosmos, given the different bands in the catalog. However, the correspondence is remarkable if we consider the physical order of the bands — shown in table B.2. However, there remain two problems:

[3] This insight directed the research toward the Equally-Distributed-Dependencies.

Firstly, most existing algorithms do not support learning Bayesian networks over continuous features, and it is necessary to discretize them first. If this approach is not enough, there are some proposals to train Bayesian networks using a combination of continuous and discrete variables [LH15; CWK17].

Secondly, there needs to be more than the correspondence between nodes in the graphs to compute the correspondence between attributes. The Bayesian networks may not be stable, with attributes changing relative positions or relations being missed.



(a) KiDS.

(b) SDSS.

Figure B.2: Two PGNs trained over two different astronomical catalogs.

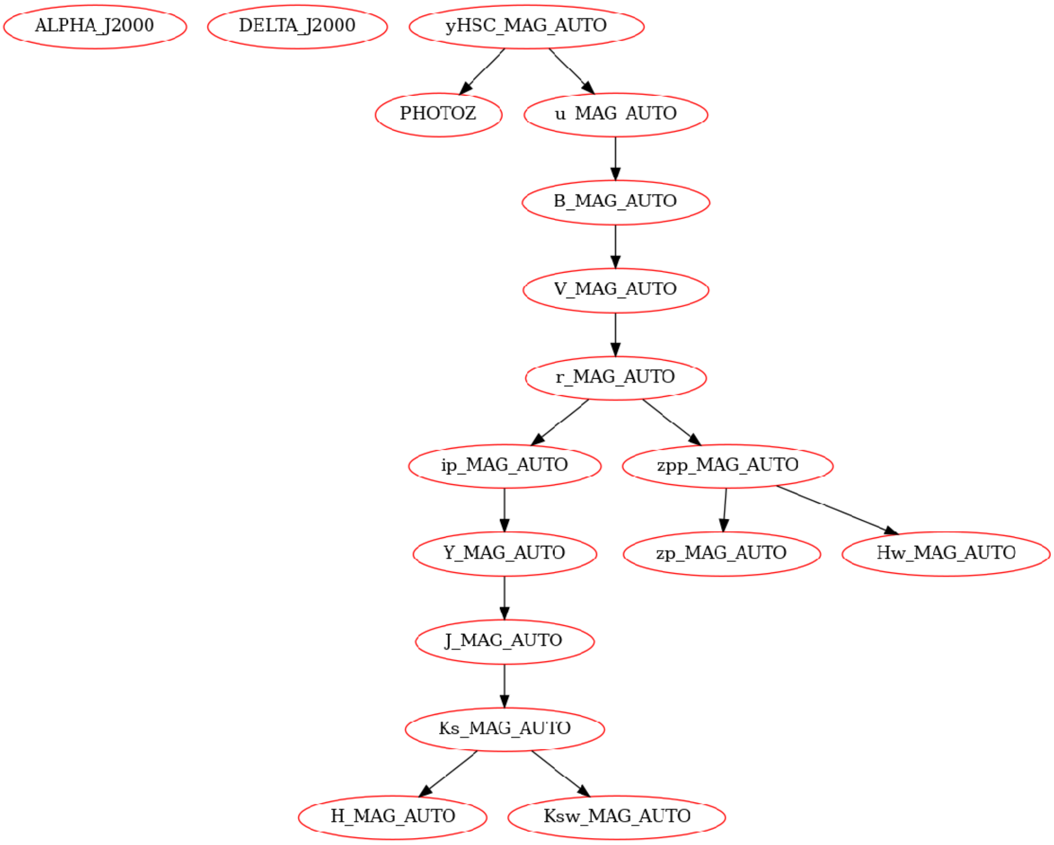


Figure B.3: Probabilistic Graphical Network trained over Cosmos.

Bands	λ	**FWHM**	**Filters**	**Description**
		Ultraviolet		
U	365 nm	66 nm	u, u', u*	
		Visible		
G	464 nm	128 nm	g'	Green
R	658 nm	138 nm	r, r', R', Rc, Re, Rj	Red
		Near Infrared		
I	806 nm	149 nm	i, i', Ic, Ie, Ij	Infrared
Z	900 nm		z, z'	

Table B.2: Subset of electromagnetic bands. λ corresponds to the wavelength.

Bibliography

[Abo18] Bela Abolfathi et al. "The Fourteenth Data Release of the Sloan Digital Sky Survey". In: *The Astrophysical Journal Supplement Series* 235.2 (2018), p. 42. DOI: 10.3847/1538-4365/aa9e8a.

[Con20] Gemma Conroy. *Three reasons to share your research failures*. Sept. 2020. URL: https://www.nature.com/nature-index/news-blog/three-reasons-to-share-your-research-science-failures.

[CWK17] Yi-Chun Chen, Tim A Wheeler, and Mykel J Kochenderfer. "Learning discrete Bayesian networks from continuous data". In: *Journal of Artificial Intelligence Research* 59 (2017), pp. 103–132. DOI: 10.1613/jair.5371.

[Jon13] Jelte TA de Jong et al. "The Kilo-Degree Survey". In: *Experimental Astronomy* 35.1-2 (2013), pp. 25–44. DOI: 10.1007/s10686-012-9306-1.

[Kra18] Tim Kraska et al. "The Case for Learned Index Structures". In: *Proceedings of the 2018 International Conference on Management of Data* abs/1712.0 (2018), pp. 489–504. DOI: 10.1145/3183713.3196909.

[Lai16] Clotilde Laigle et al. "The COSMOS2015 catalog: Exploring the 1 $<$ z $<$ 6 universe with half a million galaxies". In: *The Astrophysical Journal Supplement Series* 224.2 (2016), p. 24. DOI: 10.3847/0067-0049/224/2/24.

[LH15] Peter JF Lucas and Arjen Hommersom. "Modeling the interactions between discrete and continuous causal factors in Bayesian networks". In: *International Journal of Intelligent Systems* 30.3 (2015), pp. 209–235. DOI: 10.1002/int.21698.

[Mas99] Llew Mason et al. "Boosting algorithms as gradient descent". In: *Advances in Neural Information Processing Systems*. Vol. 12. 1999. URL: https://proceedings.neurips.cc/paper/1999/file/96a93ba89a5b5c6c226e49b88973f46e-Paper.pdf.

[Mei06] Nicolai Meinshausen. "Quantile regression forests". In: *Journal of Machine Learning Research* 7.Jun (2006), pp. 983–999. URL: https://www.jmlr.org/papers/volume7/meinshausen06a/meinshausen06a.pdf.

[Pea88] Judea Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann, 1988.

[Sch17] Jacob Schreiber. "Pomegranate: Fast and Flexible Probabilistic Modeling in Python". In: *Journal of Machine Learning Research* 18.1 (2017), pp. 5992–5997. URL: https://dl.acm.org/doi/abs/10.5555/3122009.3242021.

Appendix C

Interactive Data Exploration Update

Title	Year	Cluster	Type	Ref.
A Formal Framework for Data Lakes Based on Category Theory	2022	Flexible Engines	Philosophical Paper	[Guy22]
A Radviz-Based Visualization for Understanding Fuzzy Clustering Results	2017	Visual Tools	Proposal of Solution	[Zho17]
A Sampling-Based System for Approximate Big Data Analysis on Computing Clusters	2019	Sampling	Proposal of Solution	[SWH19]
A Unified Correlation-Based Approach to Sampling over Joins	2017	Sampling	Proposal of Solution	[KN17]
AQP++: Connecting Approximate Query Processing with Aggregate Precomputation for Interactive Analytics	2018	Query Approximation	Proposal of Solution	[Pen18]
AQapprox: Aggregation Queries Approximation with Distribution-Aware Online Sampling	2020	Query Approximation	Proposal of Solution	[WWL20]
Accelerating Approximate Aggregation Queries with Expensive Predicates	2021	Query Approximation	Proposal of Solution	[Kan21]
Adaptive Partitioning and Indexing for in Situ Query Processing	2019	Adaptive Indexing	Proposal of Solution	[Olm19a]
Aggregate Queries on Knowledge Graphs: Fast Approximation with Semantic-Aware Sampling	2022	Query Approximation	Proposal of Solution	[Wan22]
Aggregate Query Prediction under Dynamic Workloads	2019	Query Approximation	Proposal of Solution	[SAT19]
Aperture: Fast Visualizations over Spatiotemporal Datasets	2019	Visual Optimizations	Proposal of Solution	[BP19]
ApproxJoin: Approximate Distributed Joins	2018	Query Approximation	Proposal of Solution	[Quo18]

Title	Year	Cluster	Type	Ref.
Approximate Partition Selection for Big-Data Workloads Using Summary Statistics	2020	Query Approximation	Proposal of Solution	[Ron20]
Approximate Query Processing for Big Data in Heterogeneous Databases	2020	Query Approximation	Proposal of Solution	[MAT20]
Approximate Query Processing for Data Exploration Using Deep Generative Models	2020	Query Approximation	Proposal of Solution	[Thi20]
Approximate Selection with Guarantees Using Proxies	2020	Query Approximation	Proposal of Solution	[Kan20]
Approximating Aggregated SQL Queries with LSTM Networks	2021	Query Approximation	Proposal of Solution	[RRS21]
Are We Ready for Learned Cardinality Estimation?	2021	Indexes	Validation Research	[Wan21]
ArrayUDF: User-defined Scientific Data Analysis on Arrays	2017	Flexible Engines	Proposal of Solution	[Don17]
Babelfish: Efficient Execution of Polyglot Queries	2022	Flexible Engines	Proposal of Solution	[GZM22]
Balancing Familiarity and Curiosity in Data Exploration with Deep Reinforcement Learning	2021	Automatic Exploration	Proposal of Solution	[Per21]
BigIN4: Instant, Interactive Insight Identification for Multi-Dimensional Big Data	2018	Query Approximation	Proposal of Solution	[Lin18]
BlinkML: Efficient Maximum Likelihood Estimation with Probabilistic Guarantees	2019	Query Approximation	Proposal of Solution	[Par19]
Bounded Approximate Query Processing	2019	Query Approximation	Proposal of Solution	[Li19]
CLAP: Component-level Approximate Processing for Low Tail Latency and High Result Accuracy in Cloud Online Services	2017	Query Approximation	Proposal of Solution	[Han17]

Title	Year	Cluster	Type	Ref.
Coconut: A Scalable Bottom-up Approach for Building Data Series Indexes	2019	Time Series	Proposal of Solution	[Kon19]
Combining Aggregation and Sampling (Nearly) Optimally for Approximate Query Processing	2021	Query Approximation	Proposal of Solution	[Lia21]
Continuous Prefetch for Interactive Data Applications	2020	Data Prefetching	Proposal of Solution	[Moh20]
CoopStore: Optimizing Precomputed Summaries for Aggregation	2020	Query Approximation	Proposal of Solution	[GBC20]
CrossIndex: Memory-Friendly and Session-Aware Index for Supporting Crossfilter in Interactive Data Exploration	2022	Adaptive Indexing	Proposal of Solution	[Xia22]
Crossing the Finish Line Faster When Paddling the Data Lake with KAYAK	2017	Query Approximation	Proposal of Solution	[MT17]
DBEst: Revisiting Approximate Query Processing Engines with Machine Learning Models	2019	Query Approximation	Proposal of Solution	[MT19]
Database and Caching Support for Adaptive Visualization of Large Sensor Data	2020	Visual Optimizations	Proposal of Solution	[Tan20]
DeepDB: Learn from Data, Not from Queries!	2020	Query Approximation	Proposal of Solution	[Hil20]
Demonstrating the Voice-Based Exploration of Large Data Sets with CiceroDB-Zero	2020	Novel Query Interfaces	Proposal of Solution	[Tru20]
Demonstration of ScroogeDB: Getting More Bang for the Buck with Deterministic Approximation in the Cloud	2020	Query Approximation	Validation Research	[JPT20]

Title	Year	Cluster	Type	Ref.
Distributed Caching for Processing Raw Arrays	2018	Data Prefetching	Proposal of Solution	[Zha18]
EDA4SUM: Guided Exploration of Data Summaries	2022	Assisted Query Formulation	Proposal of Solution	[PYA22]
Efficiently Processing Deterministic Approximate Aggregation Query on Massive Data	2018	Query Approximation	Proposal of Solution	[Han18]
Evaluation of Machine Learning Algorithms in Predicting the next SQL Query from the Future	2021	Data Prefetching	Proposal of Solution	[MCS21]
Exploiting Machine Learning Models for Approximate Query Processing	2022	Query Approximation	Validation Research	[Lee22]
Fast Data Series Indexing for In-Memory Data	2021	Time Series	Proposal of Solution	[PFP21]
Filter before You Parse: Faster Analytics on Raw Data with Sparser	2018	Flexible Engines	Proposal of Solution	[Pal18]
FishStore: Fast Ingestion and Indexing of Raw Data	2019	Flexible Engines	Proposal of Solution	[Cha19]
FlashView: An Interactive Visual Explorer for Raw Data	2017	Visual Tools	Proposal of Solution	[Pan17]
Geo-Gap Tree: A Progressive Query and Visualization Method for Massive Spatial Data	2019	Visual Optimizations	Proposal of Solution	[Xio19]
Hashedcubes: Simple, Low Memory, Real-Time Visual Exploration of Big Data	2017	Visual Optimizations	Proposal of Solution	[Pah17]
Hercules against Data Series Similarity Search	2022	Adaptive Indexing	Proposal of Solution	[Ech22]

Title	Year	Cluster	Type	Ref.
Improved Selectivity Estimation by Combining Knowledge from Sampling and Synopses	2018	Query Approximation	Proposal of Solution	[MMK18]
Incremental Approximate Computing	2019	Query Approximation	Proposal of Solution	[Quo19]
Informative Sample-Aware Proxy for Deep Metric Learning	2022	Sampling	Proposal of Solution	[Li22]
Interactive Visual Graph Mining and Learning	2018	Visual Tools	Proposal of Solution	[Ros18]
LAQP: Learning-based Approximate Query Processing	2021	Query Approximation	Proposal of Solution	[ZW21]
LHist: Towards Learning Multi-Dimensional Histogram for Massive Spatial Data	2021	Query Approximation	Proposal of Solution	[LSC21]
Learning to Sample: Counting with Complex Queries	2019	Sampling	Proposal of Solution	[Wal19]
Learning-Based Optimization for Online Approximate Query Processing	2022	Query Approximation	Proposal of Solution	[Bi22]
Moment-Based Quantile Sketches for Efficient High Cardinality Aggregation Queries	2018	Query Approximation	Proposal of Solution	[Gan18]
Multi-Objective Fuzzy-Swarm Optimizer for Data Partitioning	2022	Adaptive Storage	Proposal of Solution	[Goy22]
Navigating the Data Lake with DATAMARAN: Automatically Extracting Structure from Log Datasets	2018	Flexible Engines	Proposal of Solution	[GHP18]
Northstar: An Interactive Data Science System	2018	Visual Tools	Proposal of Solution	[Kra18]

Title	Year	Cluster	Type	Ref.
One Size Does Not Fit All: A Bandit-Based Sampler Combination Framework with Theoretical Guarantees	2022	Query Approximation	Proposal of Solution	[Pen22]
Optimally Leveraging Density and Locality for Exploratory Browsing and Sampling	2018	Indexes	Proposal of Solution	[Kim18]
Optimizing Performance of Aggregate Query Processing with Histogram Data Structure	2019	Query Approximation	Proposal of Solution	[YZ19]
Photon: A Fast Query Engine for Lakehouse Systems	2022	Flexible Engines	Proposal of Solution	[Beh22]
Plato: Approximate Analytics over Compressed Time Series with Tight Deterministic Error Guarantees	2020	Time Series	Proposal of Solution	[LBP20]
Probabilistic Database Summarization for Interactive Data Exploration	2017	Query Approximation	Proposal of Solution	[OBS17]
ProgressiveDB: Progressive Data Analytics as a Middleware	2019	Query Approximation	Proposal of Solution	[Ber19]
QUIS: In-situ Heterogeneous Data Source Querying	2017	Flexible Engines	Proposal of Solution	[CKJ17]
Qd-Tree: Learning Data Layouts for Big Data Analytics	2020	Adaptive Storage	Proposal of Solution	[Yan20]
Query Morphing: A Proximity-Based Data Exploration for Query Reformulation	2019	Assisted Query Formulation	Proposal of Solution	[PS19]
QueryVis: Logic-based Diagrams Help Users Understand Complicated SQL Queries Faster	2020	Visual Tools	Proposal of Solution	[Lev20]

Title	Year	Cluster	Type	Ref.
RC-Index: Diversifying Answers to Range Queries	2018	Indexes	Proposal of Solution	[WMM18]
RSATree: Distribution-aware Data Representation of Large-Scale Tabular Datasets for Flexible Visual Query	2020	Novel Query Interfaces	Proposal of Solution	[Mei20]
Resource-Aware Adaptive Indexing for in Situ Visual Exploration and Analytics	2023	Adaptive Indexing	Proposal of Solution	[Mar23]
STULL: Unbiased Online Sampling for Visual Exploration of Large Spatiotemporal Data	2020	Visual Optimizations	Proposal of Solution	[Wan20]
Salvaging Failing and Straggling Queries	2022	Query Approximation	Proposal of Solution	[Sun22]
Sampling for Scientific Data Analysis and Reduction	2022	Sampling	Proposal of Solution	[Bis22]
Scalable in Situ Scientific Data Encoding for Analytical Query Processing	2018	Adaptive Indexing	Proposal of Solution	[Lak18]
SmartCube: An Adaptive Data Management Architecture for the Real-Time Visualization of Spatiotemporal Datasets	2020	Visual Optimizations	Proposal of Solution	[Liu20]
Speculative Distributed CSV Data Parsing for Big Data Analytics	2019	Adaptive Loading	Proposal of Solution	[Ge19]
Tabula in Action: A Sampling Middleware for Interactive Geospatial Visualization Dashboards	2020	Query Approximation	Proposal of Solution	[YCS20]
Taster: Self-tuning, Elastic and Online Approximate Query Processing	2019	Query Approximation	Proposal of Solution	[Olm19b]
TopKube: A Rank-Aware Data Cube for Real-Time Exploration of Spatiotemporal Data	2018	Visual Optimizations	Proposal of Solution	[Mir18]

Title	Year	Cluster	Type	Ref.
VerdictDB: Universalizing Approximate Query Processing	2018	Query Approximation	Proposal of Solution	[Par18]
VisSnippets: A Web-Based System for Impromptu Collaborative Data Exploration on Large Displays	2020	Visual Tools	Proposal of Solution	[BRJ20]
Visualization and Analytics Tool for Multi-Dimensional Data	2018	Visual Tools	Proposal of Solution	[Jes18]
Visualization of Big Spatial Data Using Coresets for Kernel Density Estimates	2021	Visual Optimizations	Proposal of Solution	[Zhe21]
WFApprox: Approximate Window Functions Processing	2020	Query Approximation	Proposal of Solution	[Lin20]
Workload Prediction for Adaptive Approximate Query Processing	2022	Query Approximation	Proposal of Solution	[SPF22]

Bibliography

[Beh22] Alexander Behm et al. “Photon: A fast query engine for lakehouse systems”. In: *Proceedings of the 2022 international conference on management of data*. SIGMOD ’22. Number of pages: 14 Place: Philadelphia, PA, USA. New York, NY, USA: Association for Computing Machinery, 2022, pp. 2326–2339. ISBN: 978-1-4503-9249-5. DOI: 10.1145/3514221.3526054.

[Ber19] Lukas Berg et al. “ProgressiveDB: Progressive data analytics as a middleware”. In: *Proc. VLDB Endow.* 12.12 (Aug. 2019). Number of pages: 4 Publisher: VLDB Endowment, pp. 1814–1817. ISSN: 2150-8097. DOI: 10.14778/3352063.3352073.

[Bi22] Wenyuan Bi et al. “Learning-Based Optimization for Online Approximate Query Processing”. In: *Database Systems for Advanced Applications*. Ed. by Arnab Bhattacharya et al. Cham: Springer International Publishing, 2022, pp. 96–103. ISBN: 978-3-031-00123-9. DOI: 10.1007/978-3-031-00123-9_7.

[Bis22] Ayan Biswas et al. “Sampling for scientific data analysis and reduction”. In: *In situ visualization for computational science*. Ed. by Hank Childs, Janine C. Bennett, and Christoph Garth. Cham: Springer International Publishing, 2022, pp. 11–36. ISBN: 978-3-030-81627-8. DOI: 10.1007/978-3-030-81627-8_2.

[BP19] Kevin Bruhwiler and Shrideep Pallickara. “Aperture: Fast visualizations over spatiotemporal datasets”. In: *Proceedings of the 12th IEEE/ACM international conference on utility and cloud computing*. UCC’19. Number of pages: 10 Place: Auckland, New Zealand.

New York, NY, USA: Association for Computing Machinery, 2019, pp. 31–40. ISBN: 978-1-4503-6894-0. DOI: 10.1145/3344341.3368817.

[BRJ20] Andrew Burks, Luc Renambot, and Andrew Johnson. "VisSnippets: A web-based system for impromptu collaborative data exploration on large displays". In: *Practice and experience in advanced research computing*. PEARC '20. Number of pages: 8 Place: Portland, OR, USA. New York, NY, USA: Association for Computing Machinery, 2020, pp. 144–151. ISBN: 978-1-4503-6689-2. DOI: 10.1145/3311790.3396666.

[Cha19] Badrish Chandramouli et al. "FishStore: Fast ingestion and indexing of raw data". In: *Proc. VLDB Endow.* 12.12 (Aug. 2019). Number of pages: 4 Publisher: VLDB Endowment, pp. 1922–1925. ISSN: 2150-8097. DOI: 10.14778/3352063.3352100.

[CKJ17] Javad Chamanara, Birgitta König-Ries, and H. V. Jagadish. "QUIS: In-situ heterogeneous data source querying". In: *Proc. VLDB Endow.* 10.12 (Aug. 2017). Number of pages: 4 Publisher: VLDB Endowment, pp. 1877–1880. ISSN: 2150-8097. DOI: 10.14778/3137765.3137798.

[Don17] Bin Dong et al. "ArrayUDF: User-defined scientific data analysis on arrays". In: *Proceedings of the 26th international symposium on high-performance parallel and distributed computing*. HPDC '17. Number of pages: 12 Place: Washington, DC, USA. New York, NY, USA: Association for Computing Machinery, 2017, pp. 53–64. ISBN: 978-1-4503-4699-3. DOI: 10.1145/3078597.3078599.

[Ech22] Karima Echihabi et al. "Hercules against data series similarity search". In: *Proc. VLDB Endow.* 15.10 (Sept. 2022). Number of pages: 14 Publisher: VLDB Endowment, pp. 2005–2018. ISSN: 2150-8097. DOI: 10.14778/3547305.3547308.

[Gan18] Edward Gan et al. "Moment-based quantile sketches for efficient high cardinality aggregation queries". In: *Proc. VLDB Endow.* 11.11 (July 2018). Number of pages: 14 Publisher: VLDB Endowment, pp. 1647–1660. ISSN: 2150-8097. DOI: 10.14778/3236187.3236212.

[GBC20] Edward Gan, Peter Bailis, and Moses Charikar. "CoopStore: Optimizing precomputed summaries for aggregation". In: *Proc. VLDB Endow.* 13.12 (Sept. 2020). Number of pages: 14 Publisher: VLDB Endowment, pp. 2174–2187. ISSN: 2150-8097. DOI: 10.14778/3407790.3407817.

[Ge19] Chang Ge et al. "Speculative distributed CSV data parsing for big data analytics". In: *Proceedings of the 2019 international conference on management of data.* SIGMOD '19. Number of pages: 17 Place: Amsterdam, Netherlands. New York, NY, USA: Association for Computing Machinery, 2019, pp. 883–899. ISBN: 978-1-4503-5643-5. DOI: 10.1145/3299869.3319898.

[GHP18] Yihan Gao, Silu Huang, and Aditya Parameswaran. "Navigating the data lake with DATAMARAN: Automatically extracting structure from log datasets". In: *Proceedings of the 2018 international conference on management of data.* SIGMOD '18. Number of pages: 16 Place: Houston, TX, USA. New York, NY, USA: Association for Computing Machinery, 2018, pp. 943–958. ISBN: 978-1-4503-4703-7. DOI: 10.1145/3183713.3183746.

[Goy22] S. B. Goyal et al. "Multi-objective Fuzzy-Swarm Optimizer for Data Partitioning". In: *Advanced Computing and Intelligent Technologies.* Ed. by Monica Bianchini et al. Singapore: Springer Singapore, 2022, pp. 307–318. ISBN: 978-981-16-2164-2. DOI: 10.1007/978-981-16-2164-2_25.

[Guy22] Alexis Guyot et al. "A formal framework for data lakes based on category theory". In: *Proceedings of the 26th international database engineered applications symposium.* IDEAS '22. Number of pages: 9 Place: Budapest, Hungary. New York, NY, USA: Association for Computing Machinery, 2022, pp. 75–83. ISBN: 978-1-4503-9709-4. DOI: 10.1145/3548785.3548797.

[GZM22] Philipp Marian Grulich, Steffen Zeuch, and Volker Markl. "Babelfish: Efficient execution of polyglot queries". In: *Proc. VLDB Endow.* 15.2 (Feb. 2022). Number of pages: 15 Publisher: VLDB Endowment, pp. 196–210. ISSN: 2150-8097. DOI: 10.14778/3489496.3489501.

[Han17] Rui Han et al. "CLAP: Component-level approximate processing for low tail latency and high result accuracy in cloud online services". In: *IEEE Transactions on Parallel and Distributed Systems* 28.8 (Aug. 2017), pp. 2190–2203. ISSN: 1558-2183. DOI: 10.1109/TPDS.2017.2650988.

[Han18] Xixian Han et al. "Efficiently processing deterministic approximate aggregation query on massive data". In: *Knowledge and Information Systems* 57.2 (Nov. 2018), pp. 437–473. ISSN: 0219-3116. DOI: 10.1007/s10115-017-1136-z.

[Hil20] Benjamin Hilprecht et al. "DeepDB: Learn from data, not from queries!" In: *Proc. VLDB Endow.* 13.7 (Mar. 2020). Number of pages: 14 Publisher: VLDB Endowment, pp. 992–1005. ISSN: 2150-8097. DOI: 10.14778/3384345.3384349.

[Jes18] David Jesenko et al. "Visualization and analytics tool for multi-dimensional data". In: *Proceedings of the 2018 international conference on big data and education.* ICBDE '18. Number of pages: 5 Place: Honolulu, HI, USA. New York, NY, USA: Association for Computing Machinery, 2018, pp. 11–15. ISBN: 978-1-4503-6358-7. DOI: 10.1145/3206157.3206159.

[JPT20] Saehan Jo, Jialing Pei, and Immanuel Trummer. "Demonstration of ScroogeDB: Getting more bang for the buck with deterministic approximation in the cloud". In: *Proc. VLDB Endow.* 13.12 (Sept. 2020). Number of pages: 4 Publisher: VLDB Endowment, pp. 2961–2964. ISSN: 2150-8097. DOI: 10.14778/3415478.3415519.

[Kan20] Daniel Kang et al. "Approximate selection with guarantees using proxies". In: *Proc. VLDB Endow.* 13.12 (Sept. 2020). Number of pages: 14 Publisher: VLDB Endowment, pp. 1990–2003. ISSN: 2150-8097. DOI: 10.14778/3407790.3407804.

[Kan21] Daniel Kang et al. "Accelerating approximate aggregation queries with expensive predicates". In: *Proc. VLDB Endow.* 14.11 (Oct. 2021). Number of pages: 14 Publisher: VLDB Endowment, pp. 2341–2354. ISSN: 2150-8097. DOI: 10.14778/3476249.3476285.

[Kim18] Albert Kim et al. "Optimally leveraging density and locality for exploratory browsing and sampling". In: *Proceedings of the workshop on human-in-the-loop data analytics*. HILDA'18. Number of pages: 7 Place: Houston, TX, USA. New York, NY, USA: Association for Computing Machinery, 2018. ISBN: 978-1-4503-5827-9. DOI: 10.1145/3209900.3209903.

[KN17] Niranjan Kamat and Arnab Nandi. "A unified correlation-based approach to sampling over joins". In: *Proceedings of the 29th international conference on scientific and statistical database management*. SSDBM '17. Number of pages: 12 Place: Chicago, IL, USA. New York, NY, USA: Association for Computing Machinery, 2017. ISBN: 978-1-4503-5282-6. DOI: 10.1145/3085504.3085524.

[Kon19] Haridimos Kondylakis et al. "Coconut: A scalable bottom-up approach for building data series indexes". In: *Proc. VLDB Endow.* 11.6 (Jan. 2019). Number of pages: 14 Publisher: VLDB Endowment, pp. 677–690. ISSN: 2150-8097. DOI: 10.14778/3199517.3199519.

[Kra18] Tim Kraska. "Northstar: An interactive data science system". In: *Proc. VLDB Endow.* 11.12 (Aug. 2018). Number of pages: 15 Publisher: VLDB Endowment, pp. 2150–2164. ISSN: 2150-8097. DOI: 10.14778/3229863.3240493.

[Lak18] Sriram Lakshminarasimhan et al. "Scalable in situ scientific data encoding for analytical query processing". In: *Proceedings of the 22nd international symposium on high-performance parallel and distributed computing*. HPDC '13. Number of pages: 12 Place: New York, New York, USA. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–12. ISBN: 978-1-4503-1910-2. DOI: 10.1145/2462902.2465527.

[LBP20] Chunbin Lin, Etienne Boursier, and Yannis Papakonstantinou. "Plato: Approximate analytics over compressed time series with tight deterministic error guarantees". In: *Proc. VLDB Endow.* 13.7 (Mar. 2020). Number of pages: 14 Publisher: VLDB Endowment, pp. 1105–1118. ISSN: 2150-8097. DOI: 10.14778/3384345.3384357.

[Lee22] Taewhi Lee et al. "Exploiting machine learning models for approximate query processing". In: *2022 IEEE international conference on big data (big data)*. Dec. 2022, pp. 6752–6754. DOI: 10.1109/BigData55660.2022.10020252.

[Lev20] Aristotelis Leventidis et al. "QueryVis: Logic-based diagrams help users understand complicated SQL queries faster". In: *Proceedings of the 2020 ACM SIGMOD international conference on management of data*. SIGMOD '20. Number of pages: 16 Place: Portland, OR, USA. New York, NY, USA: Association for Computing Machinery, 2020, pp. 2303–2318. ISBN: 978-1-4503-6735-6. DOI: 10.1145/3318464.3389767.

[Li19] Kaiyu Li et al. "Bounded approximate query processing". In: *IEEE Transactions on Knowledge and Data Engineering* 31.12 (Dec. 2019), pp. 2262–2276. ISSN: 1558-2191. DOI: 10.1109/TKDE.2018.2877362.

[Li22] Aoyu Li et al. "Informative sample-aware proxy for deep metric learning". In: *Proceedings of the 4th ACM international conference on multimedia in asia*. MMAsia '22. Number of pages: 11 Place: Tokyo, Japan. New York, NY, USA: Association for Computing Machinery, 2022. ISBN: 978-1-4503-9478-9. DOI: 10.1145/3551626.3564942.

[Lia21] Xi Liang et al. "Combining aggregation and sampling (nearly) optimally for approximate query processing". In: *Proceedings of the 2021 international conference on management of data*. SIGMOD '21. Number of pages: 13 Place: Virtual Event, China. New York, NY, USA: Association for Computing Machinery, 2021, pp. 1129–1141. ISBN: 978-1-4503-8343-1. DOI: 10.1145/3448016.3457277.

[Lin18] Qingwei Lin et al. "BigIN4: Instant, interactive insight identification for multi-dimensional big data". In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery &amp; data mining*. KDD '18. Number of pages: 9 Place: London, United Kingdom. New York, NY, USA: Association for Computing Machinery, 2018, pp. 547–555. ISBN: 978-1-4503-5552-0. DOI: 10.1145/3219819.3219867.

[Lin20] Chunbo Lin et al. “WFApprox: Approximate Window Functions Processing”. In: *Database Systems for Advanced Applications*. Ed. by Yunmook Nah et al. Cham: Springer International Publishing, 2020, pp. 72–87. ISBN: 978-3-030-59410-7. DOI: 10.1007/978-3-030-59410-7_5.

[Liu20] Can Liu et al. “SmartCube: An adaptive data management architecture for the real-time visualization of spatiotemporal datasets”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pp. 790–799. DOI: 10.1109/TVCG.2019.2934434.

[LSC21] Qiyu Liu, Yanyan Shen, and Lei Chen. “LHist: Towards learning multi-dimensional histogram for massive spatial data”. In: *2021 IEEE 37th international conference on data engineering (ICDE)*. ISSN: 2375-026X. Apr. 2021, pp. 1188–1199. DOI: 10.1109/ICDE51399.2021.00107.

[Mar23] Stavros Maroulis et al. “Resource-aware adaptive indexing for in situ visual exploration and analytics”. In: *The VLDB Journal* 32.1 (Jan. 2023), pp. 199–227. ISSN: 0949-877X. DOI: 10.1007/s00778-022-00739-z.

[MAT20] Manoj Muniswamaiah, Tilak Agerwala, and Charles C. Tappert. “Approximate query processing for big data in heterogeneous databases”. In: *2020 IEEE international conference on big data (big data)*. Dec. 2020, pp. 5765–5767. DOI: 10.1109/BigData50022.2020.9378310.

[MCS21] Venkata Vamsikrishna Meduri, Kanchan Chowdhury, and Mohamed Sarwat. “Evaluation of machine learning algorithms in predicting the next SQL query from the future”. In: *ACM Transactions on Database Systems* 46.1 (Mar. 2021). Number of pages: 46 Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0362-5915. DOI: 10.1145/3442338.

[Mei20] Honghui Mei et al. “RSATree: Distribution-aware data representation of large-scale tabular datasets for flexible visual query”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1

(Jan. 2020), pp. 1161–1171. ISSN: 1941-0506. DOI: 10.1109/TVCG.2019.2934800.

[Mir18] Fabio Miranda et al. "TopKube: A rank-aware data cube for real-time exploration of spatiotemporal data". In: *IEEE Transactions on Visualization and Computer Graphics* 24.3 (Mar. 2018), pp. 1394–1407. ISSN: 1941-0506. DOI: 10.1109/TVCG.2017.2671341.

[MMK18] Magnus Müller, Guido Moerkotte, and Oliver Kolb. "Improved selectivity estimation by combining knowledge from sampling and synopses". In: *Proc. VLDB Endow.* 11.9 (May 2018). Number of pages: 13 Publisher: VLDB Endowment, pp. 1016–1028. ISSN: 2150-8097. DOI: 10.14778/3213880.3213882.

[Moh20] Haneen Mohammed. "Continuous prefetch for interactive data applications". In: *Proceedings of the 2020 ACM SIGMOD international conference on management of data*. SIGMOD '20. Number of pages: 3 Place: Portland, OR, USA. New York, NY, USA: Association for Computing Machinery, 2020, pp. 2841–2843. ISBN: 978-1-4503-6735-6. DOI: 10.1145/3318464.3384405.

[MT17] Antonio Maccioni and Riccardo Torlone. "Crossing the finish line faster when paddling the data lake with KAYAK". In: *Proc. VLDB Endow.* 10.12 (Aug. 2017). Number of pages: 4 Publisher: VLDB Endowment, pp. 1853–1856. ISSN: 2150-8097. DOI: 10.14778/3137765.3137792.

[MT19] Qingzhi Ma and Peter Triantafillou. "DBEst: Revisiting approximate query processing engines with machine learning models". In: *Proceedings of the 2019 international conference on management of data*. SIGMOD '19. Number of pages: 18 Place: Amsterdam, Netherlands. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1553–1570. ISBN: 978-1-4503-5643-5. DOI: 10.1145/3299869.3324958.

[OBS17] Laurel Orr, Magdalena Balazinska, and Dan Suciu. "Probabilistic database summarization for interactive data exploration". In: *Proc. VLDB Endow.* 10.10 (June 2017). Number of pages: 12 Publisher: VLDB Endowment, pp. 1154–1165. ISSN: 2150-8097. DOI: 10.14778/3115404.3115419.

[Olm19a] Matthaios Olma et al. “Adaptive partitioning and indexing for in situ query processing”. In: *The VLDB Journal* 29.1 (Nov. 2019). Number of pages: 23 Place: Berlin, Heidelberg Publisher: Springer-Verlag, pp. 569–591. ISSN: 1066-8888. DOI: 10.1007/s00778-019-00580-x.

[Olm19b] Matthaios Olma et al. “Taster: Self-tuning, elastic and online approximate query processing”. In: *2019 IEEE 35th international conference on data engineering (ICDE)*. ISSN: 2375-026X. Apr. 2019, pp. 482–493. DOI: 10.1109/ICDE.2019.00050.

[Pah17] Cícero A. L. Pahins et al. “Hashedcubes: Simple, low memory, real-time visual exploration of big data”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (Jan. 2017), pp. 671–680. ISSN: 1941-0506. DOI: 10.1109/TVCG.2016.2598624.

[Pal18] Shoumik Palkar et al. “Filter before you parse: Faster analytics on raw data with sparser”. In: *Proc. VLDB Endow.* 11.11 (July 2018). Number of pages: 14 Publisher: VLDB Endowment, pp. 1576–1589. ISSN: 2150-8097. DOI: 10.14778/3236187.3236207.

[Pan17] Zhifei Pang et al. “FlashView: An interactive visual explorer for raw data”. In: *Proc. VLDB Endow.* 10.12 (Aug. 2017). Number of pages: 4 Publisher: VLDB Endowment, pp. 1869–1872. ISSN: 2150-8097. DOI: 10.14778/3137765.3137796.

[Par18] Yongjoo Park et al. “VerdictDB: Universalizing approximate query processing”. In: *Proceedings of the 2018 international conference on management of data*. SIGMOD ’18. Number of pages: 16 Place: Houston, TX, USA. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1461–1476. ISBN: 978-1-4503-4703-7. DOI: 10.1145/3183713.3196905.

[Par19] Yongjoo Park et al. “BlinkML: Efficient maximum likelihood estimation with probabilistic guarantees”. In: *Proceedings of the 2019 international conference on management of data*. SIGMOD ’19. Number of pages: 18 Place: Amsterdam, Netherlands. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1135–1152. ISBN: 978-1-4503-5643-5. DOI: 10.1145/3299869.3300077.

[Pen18] Jinglin Peng et al. "AQP++: Connecting approximate query processing with aggregate precomputation for interactive analytics". In: *Proceedings of the 2018 international conference on management of data*. SIGMOD '18. Number of pages: 16 Place: Houston, TX, USA. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1477–1492. ISBN: 978-1-4503-4703-7. DOI: 10.1145/3183713.3183747.

[Pen22] Jinglin Peng et al. "One size does not fit all: A bandit-based sampler combination framework with theoretical guarantees". In: *Proceedings of the 2022 international conference on management of data*. SIGMOD '22. Number of pages: 14 Place: Philadelphia, PA, USA. New York, NY, USA: Association for Computing Machinery, 2022, pp. 531–544. ISBN: 978-1-4503-9249-5. DOI: 10.1145/3514221.3517900.

[Per21] Aurélien Personnaz et al. "Balancing familiarity and curiosity in data exploration with deep reinforcement learning". In: *Fourth workshop in exploiting AI techniques for data management*. aiDM '21. Number of pages: 8 Place: Virtual Event, China. New York, NY, USA: Association for Computing Machinery, 2021, pp. 16–23. ISBN: 978-1-4503-8535-0. DOI: 10.1145/3464509.3464884.

[PFP21] Botao Peng, Panagiota Fatourou, and Themis Palpanas. "Fast data series indexing for in-memory data". In: *The VLDB Journal* 30.6 (June 2021). Number of pages: 27 Place: Berlin, Heidelberg Publisher: Springer-Verlag, pp. 1041–1067. ISSN: 1066-8888. DOI: 10.1007/s00778-021-00677-2.

[PS19] Jay Patel and Vikram Singh. "Query Morphing: A Proximity-Based Data Exploration for Query Reformulation". In: *Computational Intelligence: Theories, Applications and Future Directions - Volume I*. Ed. by Nishchal K. Verma and A. K. Ghosh. Singapore: Springer Singapore, 2019, pp. 247–259. ISBN: 978-981-13-1132-1. DOI: 10.1007/978-981-13-1132-1_20.

[PYA22] Aurélien Personnaz, Brit Youngmann, and Sihem Amer-Yahia. “EDA4SUM: Guided exploration of data summaries”. In: *Proc. VLDB Endow.* 15.12 (Sept. 2022). Number of pages: 4 Publisher: VLDB Endowment, pp. 3590–3593. ISSN: 2150-8097. DOI: 10.14778/3554821.3554851.

[Quo18] Do Le Quoc et al. “ApproxJoin: Approximate distributed joins”. In: *Proceedings of the ACM symposium on cloud computing.* SoCC ’18. Number of pages: 13 Place: Carlsbad, CA, USA. New York, NY, USA: Association for Computing Machinery, 2018, pp. 426–438. ISBN: 978-1-4503-6011-1. DOI: 10.1145/3267809.3267834.

[Quo19] Do Le Quoc et al. “Incremental Approximate Computing”. In: *Encyclopedia of Big Data Technologies.* Ed. by Sherif Sakr and Albert Y. Zomaya. Cham: Springer International Publishing, 2019, pp. 1000–1007. ISBN: 978-3-319-77525-8. DOI: 10.1007/978-3-319-77525-8_151.

[Ron20] Kexin Rong et al. “Approximate partition selection for big-data workloads using summary statistics”. In: *Proc. VLDB Endow.* 13.12 (Sept. 2020). Number of pages: 14 Publisher: VLDB Endowment, pp. 2606–2619. ISSN: 2150-8097. DOI: 10.14778/3407790.3407848.

[Ros18] Ryan A. Rossi et al. “Interactive visual graph mining and learning”. In: *ACM Transactions on Intelligent Systems and Technology* 9.5 (July 2018). Number of pages: 25 Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 2157-6904. DOI: 10.1145/3200764.

[RRS21] Nir Regev, Lior Rokach, and Asaf Shabtai. “Approximating aggregated SQL queries with LSTM networks”. In: *2021 international joint conference on neural networks (IJCNN).* ISSN: 2161-4407. July 2021, pp. 1–8. DOI: 10.1109/IJCNN52387.2021.9533974.

[SAT19] Fotis Savva, Christos Anagnostopoulos, and Peter Triantafillou. “Aggregate query prediction under dynamic workloads”. In: *2019 IEEE international conference on big data (big data).* Dec. 2019, pp. 671–676. DOI: 10.1109/BigData47090.2019.9006267.

[SPF22] Hamid Shahrivari, Odysseas Papapetrou, and George Fletcher. “Workload prediction for adaptive approximate query processing”. In: *2022 IEEE international conference on big data (big data)*. Dec. 2022, pp. 217–222. DOI: 10.1109/BigData55660.2022.10020614.

[Sun22] Bruhathi Sundarmurthy et al. “Salvaging failing and straggling queries”. In: *2022 IEEE 38th international conference on data engineering (ICDE)*. ISSN: 2375-026X. May 2022, pp. 1382–1395. DOI: 10.1109/ICDE53745.2022.00108.

[SWH19] Salman Salloum, Yinxu Wu, and Joshua Zhexue Huang. “A sampling-based system for approximate big data analysis on computing clusters”. In: *Proceedings of the 28th ACM international conference on information and knowledge management*. CIKM ’19. Number of pages: 4 Place: Beijing, China. New York, NY, USA: Association for Computing Machinery, 2019, pp. 2481–2484. ISBN: 978-1-4503-6976-3. DOI: 10.1145/3357384.3358124.

[Tan20] Sapan Tanted et al. “Database and caching support for adaptive visualization of large sensor data”. In: *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*. CoDS COMAD 2020. Number of pages: 9 Place: Hyderabad, India. New York, NY, USA: Association for Computing Machinery, 2020, pp. 98–106. ISBN: 978-1-4503-7738-6. DOI: 10.1145/3371158.3371170.

[Thi20] Saravanan Thirumuruganathan et al. “Approximate query processing for data exploration using deep generative models”. In: *2020 IEEE 36th international conference on data engineering (ICDE)*. ISSN: 2375-026X. Apr. 2020, pp. 1309–1320. DOI: 10.1109/ICDE48307.2020.00117.

[Tru20] Immanuel Trummer. “Demonstrating the voice-based exploration of large data sets with CiceroDB-Zero”. In: *Proc. VLDB Endow.* 13.12 (Sept. 2020). Number of pages: 4 Publisher: VLDB Endowment, pp. 2869–2872. ISSN: 2150-8097. DOI: 10.14778/3415478.3415496.

[Wal19] Brett Walenz et al. “Learning to sample: Counting with complex queries”. In: *Proc. VLDB Endow.* 13.3 (Nov. 2019). Number of pages: 13 Publisher: VLDB Endowment, pp. 390–402. ISSN: 2150-8097. DOI: 10.14778/3368289.3368302.

[Wan20] Guizhen Wang et al. "STULL: Unbiased online sampling for visual exploration of large spatiotemporal data". In: *2020 IEEE conference on visual analytics science and technology (VAST)*. Oct. 2020, pp. 72–83. DOI: 10.1109/VAST50239.2020.00012.

[Wan21] Xiaoying Wang et al. "Are we ready for learned cardinality estimation?" In: *Proc. VLDB Endow.* 14.9 (Oct. 2021). Number of pages: 15 Publisher: VLDB Endowment, pp. 1640–1654. ISSN: 2150-8097. DOI: 10.14778/3461535.3461552.

[Wan22] Yuxiang Wang et al. "Aggregate queries on knowledge graphs: Fast approximation with semantic-aware sampling". In: *2022 IEEE 38th international conference on data engineering (ICDE)*. ISSN: 2375-026X. May 2022, pp. 2914–2927. DOI: 10.1109/ICDE53745.2022.00263.

[WMM18] Yue Wang, Alexandra Meliou, and Gerome Miklau. "RC-Index: Diversifying answers to range queries". In: *Proc. VLDB Endow.* 11.7 (Mar. 2018). Number of pages: 14 Publisher: VLDB Endowment, pp. 773–786. ISSN: 2150-8097. DOI: 10.14778/3192965.3192969.

[WWL20] Han Wu, Xiaoling Wang, and Xingjian Lu. "AQapprox: Aggregation Queries Approximation with Distribution-Aware Online Sampling". In: *Web Information Systems Engineering – WISE 2020*. Ed. by Zhisheng Huang et al. Cham: Springer International Publishing, 2020, pp. 404–416. ISBN: 978-3-030-62008-0. DOI: 10.1007/978-3-030-62008-0_28.

[Xia22] Tianyu Xia et al. "CrossIndex: Memory-Friendly and Session-Aware Index for Supporting Crossfilter in Interactive Data Exploration". In: *Database Systems for Advanced Applications*. Ed. by Arnab Bhattacharya et al. Cham: Springer International Publishing, 2022, pp. 476–492. ISBN: 978-3-031-00123-9. DOI: 10.1007/978-3-031-00123-9_38.

[Xio19] Wei Xiong et al. “Geo-gap tree: A progressive query and visualization method for massive spatial data”. In: *IEEE access : practical innovations, open solutions* 7 (2019), pp. 99428–99440. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2929531.

[Yan20] Zongheng Yang et al. “Qd-tree: Learning data layouts for big data analytics”. In: *Proceedings of the 2020 ACM SIGMOD international conference on management of data*. SIGMOD ’20. Number of pages: 16 Place: Portland, OR, USA. New York, NY, USA: Association for Computing Machinery, 2020, pp. 193–208. ISBN: 978-1-4503-6735-6. DOI: 10.1145/3318464.3389770.

[YCS20] Jia Yu, Kanchan Chowdhury, and Mohamed Sarwat. “Tabula in action: A sampling middleware for interactive geospatial visualization dashboards”. In: *Proc. VLDB Endow.* 13.12 (Sept. 2020). Number of pages: 4 Publisher: VLDB Endowment, pp. 2925–2928. ISSN: 2150-8097. DOI: 10.14778/3415478.3415510.

[YZ19] Liang Yong and Mu Zhaonan. “Optimizing Performance of Aggregate Query Processing with Histogram Data Structure”. In: *Software Engineering Methods in Intelligent Algorithms*. Ed. by Radek Silhavy. Cham: Springer International Publishing, 2019, pp. 342–350. ISBN: 978-3-030-19807-7. DOI: 10.1007/978-3-030-19807-7_33.

[Zha18] Weijie Zhao et al. “Distributed caching for processing raw arrays”. In: *Proceedings of the 30th international conference on scientific and statistical database management*. SSDBM ’18. Number of pages: 12 Place: Bozen-Bolzano, Italy. New York, NY, USA: Association for Computing Machinery, 2018. ISBN: 978-1-4503-6505-5. DOI: 10.1145/3221269.3221295.

[Zhe21] Yan Zheng et al. “Visualization of big spatial data using coresets for kernel density estimates”. In: *IEEE Transactions on Big Data* 7.3 (July 2021), pp. 524–534. ISSN: 2332-7790. DOI: 10.1109/TBDATA.2019.2913655.

[Zho17] Fangfang Zhou et al. “A radviz-based visualization for understanding fuzzy clustering results”. In: *Proceedings of the 10th interna-*

tional symposium on visual information communication and interaction. VINCI '17. Number of pages: 7 Place: Bangkok, Thailand. New York, NY, USA: Association for Computing Machinery, 2017, pp. 9–15. ISBN: 978-1-4503-5292-5. DOI: 10.1145/3105971.3105980.

[ZW21] Meifan Zhang and Hongzhi Wang. "LAQP: Learning-based approximate query processing". In: *Information Sciences* 546 (2021), pp. 1113–1134. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2020.09.070.