Optimizing the Post-Mining Process in Association Rule Mining

Data Structures, Substitution Item Mining, and Visualization

Mikhail Kudriavtsev, MSc

Supervised by

Dr. Andrew McCarren and Dr. Marija Bezbradica

A thesis presented for the degree of Doctor of Philosophy

School of Computing

Dublin City University

May 2025

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Mikhail Kudriavtsev

ID No.: 19215129

Date: May 15th, 2025

Dedication

To my father,

You inspired my love for science and set me on this path that became my life's journey. Your unwavering support and belief in me have been my guiding light. Though you left us just months ago, before I could share this achievement with you, your spirit is with me in every step I take. I am so grateful for all you've done for me. I know you would be proud.

I love you.

Acknowledgements

First and foremost, I want to thank the people of Ireland for welcoming me with great kindness. The warmth and generosity I've experienced have given me a deep love for Ireland and its culture. Every person I've met along this journey has contributed to my understanding of how kind and beautiful the world can be.

I am especially grateful to my supervisors, Dr. Andrew McCarren and Dr. Marija Bezbradica. They have been my "science parents," teaching me not only in academia but in life too. Their influence will always be with me, guiding me in my future endeavors. Marija's love and passion have warmed my heart, while Andrew's drive has motivated me to strive for excellence. Thank you both; I truly love you.

I also want to thank Mr. Brendan Gallagher, my dear friend and landlord, for his endless positivity. He taught me the Irish language and culture. To me, he embodies the spirit of Ireland, deepening my love for this country each day.

To my dearest friend Denis, thank you for keeping life colorful and being someone I can always rely on. I am grateful to have you by my side on this long journey of life.

I am also thankful to our CRT Program Manager, Janet Choi, and CRT Officer, Angela Lally, for their support throughout this process.

This work would not have been possible without the financial support from Science Foundation Ireland (SFI), whose funding made this research possible. To everyone who played a part in this endeavor, I offer my sincerest thanks.

Lastly, I want to thank my mother. Your endless love and support have been my guiding light.

Contents

List of Figures

List of Tables

Optimizing the Post-Mining Processes in Association Rule Mining

Mikhail Kudriavtsev

Abstract

In Association Rule Mining (ARM), the generation of large volumes of association rules from complex datasets often presents challenges in terms of scalability, efficiency, and interpretability. This thesis addresses these challenges by developing novel methodologies and data structures tailored to improve the post-mining phase of ARM. Our approach begins with the creation of a specialized data structure to efficiently store and retrieve association rules, enhancing memory efficiency and processing speed. This data structure is then leveraged to design a robust methodology for substitute item mining, an emerging area that enables the identification of alternative items based on observed patterns, with potential applications in areas such as inventory management and consumer behavior analysis. Furthermore, to improve the interpretability of ARM results, we propose advanced visualization techniques that utilize the developed data structure, allowing users to effectively explore and understand complex relationships within large rulesets. The effectiveness of these methodologies was evaluated through surveys and case studies, demonstrating significant improvements in both cognitive load for visualization and alignment with consumer preferences in substitute item identification. This research contributes to the broader field of ARM by providing tools that enhance scalability, interpretability, and practical applicability, paving the way for more efficient knowledge discovery and decision-making in data-rich environments.

Chapter 1

Introduction

This thesis presents a novel approach to improving the post-mining process in Association Rule Mining (ARM), focusing on efficient identification and visualization of patterns within association rules. While the application to substitute items serves as a motivating example, the proposed methodologies are designed to be general-purpose and applicable across diverse ARM scenarios.

ARM is a key technique in data mining that uncovers relationships between items in large datasets, initially popularized by applications in market basket analysis (Agrawal, Imieliński, *et al.* 1993). While ARM is a powerful tool, the post-mining phase often struggles with managing and analyzing the extensive number of generated rules, limiting its practical utility and comprehensibility (Y. C. Chen *et al.* 2015; Hahsler 2016; Sethi *et al.* 2018; Jentner *et al.* 2019; Danh Bui-Thi *et al.* 2020; Ruiz *et al.* 2020; Amit Pande *et al.* 2022). To address these challenges, our research introduces a new data structure designed to significantly improve the speed and efficiency of processing ARM results. Building on this foundation, we address a crucial yet underexplored area in ARM: identifying substitute items. As interest grows in applications that go beyond discovering frequent item associations to understanding product alternatives and competitive relationships, we develop a robust methodology for substitute item mining within these rulesets. Additionally, we propose advanced visualization techniques to better reveal hidden patterns and support decision-making. By addressing these critical aspects, this thesis aims to fill

the gap in the field of processing association rules, offering a comprehensive solution for more effective and scalable post-mining analysis.

1.1 Background and Motivation

Association Rule Mining has become an essential technique in data mining, widely used for discovering interesting relationships between variables in large datasets. Originating from market basket analysis, where it was initially applied to identify associations between purchased items, ARM has since been employed in various domains, including healthcare, finance, and web usage mining (Shaukat Dar *et al.* 2015; Xiaotong Liu *et al.* 2016; Yazgana *et al.* 2016; Ghafari *et al.* 2019; Máša *et al.* 2024). The primary goal of ARM is to identify sets of frequently co-occurring items and generate association rules that can provide valuable insights into data-driven decision-making processes.

Despite its widespread application, the effective use of ARM is often limited by the sheer volume of rules it produces, especially when applied to large datasets. This challenge becomes particularly evident in the post-mining phase, where the task shifts from generating rules to efficiently processing, managing, and interpreting them. Traditional methods often fall short in handling the scalability required for such tasks, leading to inefficiencies that can obscure valuable insights and hinder decision-making (Hahsler and Karpienko 2017; Fister *et al.* 2023).

In recent years, research has gradually shifted from focusing solely on mining rules to understanding and interpreting them. Our work aims to contribute to this evolving trend by improving rule interpretability. However, before we can address interpretability, this work focuses on enhancing the underlying data itself. Rather than working directly with raw rules, we begin with a robust data preparation phase. This allows us to explore how we can better structure the data to facilitate more meaningful and efficient rule analysis, laying the groundwork for our further research. By improving the base data, we ensure that the subsequent steps in developing methods for rule interpretation are built on a solid foundation.

One of the key tasks in rule processing is uncovering underlying patterns, such as identifying substitute items. Although crucial, substitute item mining has received limited attention due to the complexity of evaluating and identifying these items, which depend on contextual factors and subtle shifts in item associations. Nevertheless, their identification is vital for understanding competitive product relationships, improving recommendation systems, and predicting market cannibalization (Guide *et al.* 2010; De Giovanni *et al.* 2018), where one product negatively impacts the sales of another one within the same company – an issue that can be anticipated and mitigated through effective substitute item analysis.

Furthermore, effective visualization of association rules is crucial for translating complex patterns into actionable insights (Jentner *et al.* 2019; Fister *et al.* 2023). Traditional visualization techniques often struggle with clarity and scalability, particularly when dealing with large datasets. This research proposes advanced visualization methods that not only enhance the interpretability of ARM results but also facilitate more informed decision-making by highlighting key patterns, such as substitution relationships or clusters, that might otherwise remain hidden (W. Huang *et al.* 2009; Jentner *et al.* 2019).

In summary, this thesis addresses critical gaps in the current state of ARM by proposing solutions that improve the efficiency, scalability, and usability of the post-mining process. The proposed methodologies are expected to significantly enhance the practical application of ARM across various domains, offering tools that are both powerful and user-friendly.

1.2 Research Questions

The primary issue this thesis addresses is the inefficiency and lack of scalability in the post-mining phase of Association Rule Mining. As datasets grow, the expanding volume of generated rules makes it difficult to extract meaningful insights, particularly in time-sensitive areas such as retail and e-commerce, where the understanding of consumer behavior can influence inventory management and customer satisfac-

tion. Existing methodologies struggle to process large rulesets efficiently, creating a gap in the practical application of ARM (Wu *et al.* 2020). This thesis proposes a novel data structure and methodology to improve the speed, manipulation, and visualization of ARM results, with a focus on substitute item mining.

This leads to the following research questions:

RQ1: Can a data structure be designed to efficiently process and manage the results of Association Rule Mining, enabling faster retrieval and manipulation of large rulesets?

The research question addresses the foundational challenge of designing a data structure that can significantly enhance the *efficiency* of processing association rules from large datasets. The goal is to explore and develop a data structure that optimizes both the storage and retrieval processes, thereby facilitating more effective knowledge discovery within ARM.

RQ2: Can a robust methodology be developed for identifying substitute items within the processed ruleset, leveraging the optimized data structure developed in RQ1?

This question focuses on the development of a methodology for substitution item mining, building upon the data structure introduced in RQ1. The aim is to develop a *methodology* that leverages the optimized data structure to accurately identify substitute items, incorporating complex queries and contextual analysis to provide precise and reliable substitution insights within large and complex rulesets.

RQ3: How can the results of substitution item mining and other discovered patterns in Association Rule Mining be effectively visualized to reveal implicitly hidden knowledge and facilitate decision-making?

The final research question focuses on the visualization aspect, which is crucial for interpreting the results of ARM and substitution item mining. This question aims to explore how the developed data structure can be adapted for *visual repre-*

sentation, making it easier for decision-makers to understand and utilize the mined rules and identified substitutes. The visualization techniques developed are designed to improve the interpretability and usability of association rule mining results.

Through these research questions, this thesis seeks to contribute to the field by developing a comprehensive approach that starts with data structure optimization, progresses through the creation of a substitution mining methodology, and finalizes with the visualization of ARM results. The central focus on substitution item mining ensures that the work addresses a significant challenge in the application of ARM in real-world scenarios.

The answers to these research questions will be provided in the following chapters. In Chapter 3, we introduce the data structure optimization for efficient rule processing. In Chapter 4, we present the methodology for substitution item mining, detailing the approach and its applications. Finally, in Chapter 5, we propose advanced visualization techniques that enhance the interpretability and usability of ARM results.

1.3 Contributions

This thesis makes several key contributions to the field of data mining and ARM. First, it introduces a novel data structure that significantly enhances the efficiency of processing and managing large association rulesets. This data structure not only improves retrieval and manipulation speed but also provides a foundation for more advanced analysis techniques. Second, the thesis develops a robust methodology for substitute item mining that leverages the efficiency gains from the data structure to provide more accurate and context-aware substitution analysis. Third, the research proposes advanced visualization techniques that enhance the interpretability and usability of ARM results. Although demonstrated in the context of substitution mining, these visualization techniques are applicable to general rule analysis and exploration. These contributions collectively advance the state of the art in ARM, offering practical tools and methods that can be applied across a range of industries

and applications.

To contextualize the structure of this thesis, Figure 1.1 compares the traditional pipeline for post-mining tasks in ARM with the integrated approach proposed in this work. Rather than treating tasks such as rule processing, substitution analysis, and visualization in isolation, the proposed framework unifies them to improve efficiency, scalability, and usability.

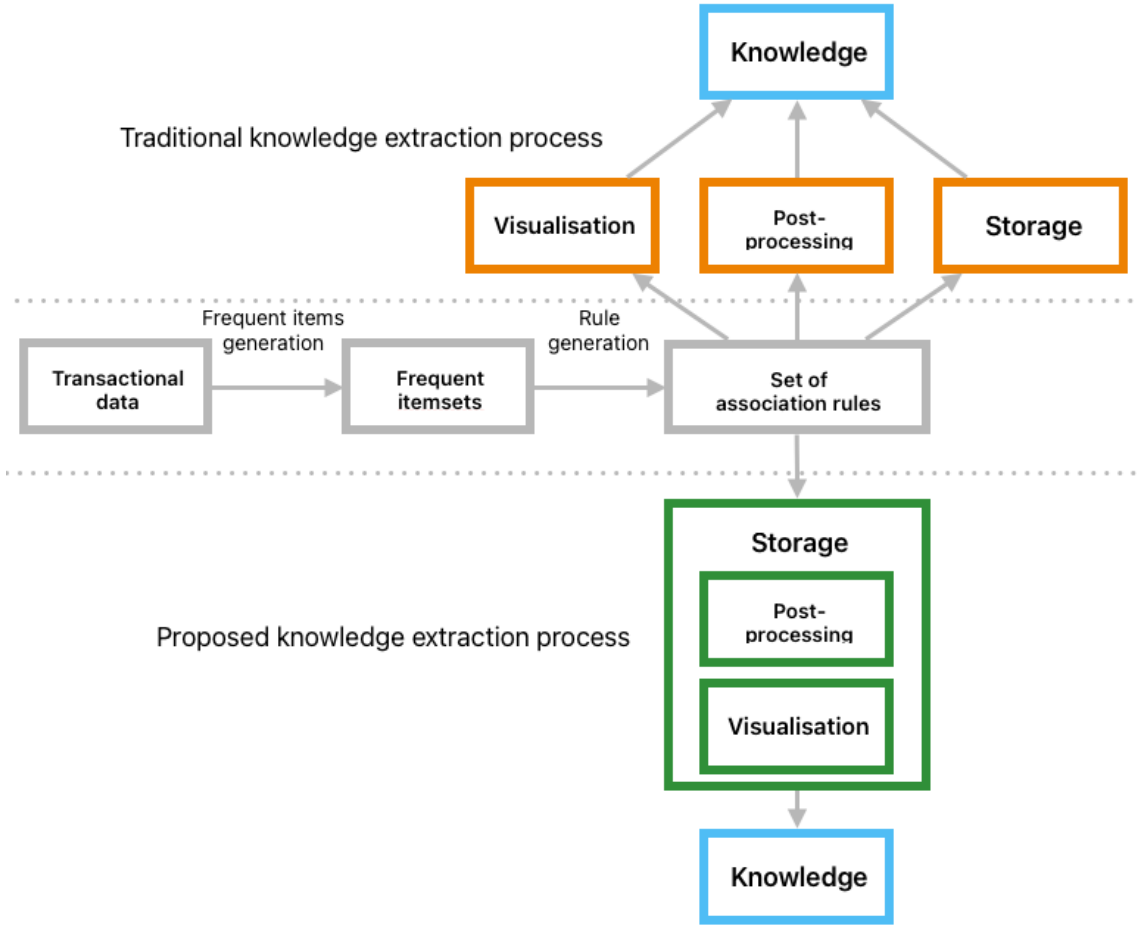


Figure 1.1: Comparison of traditional and proposed post-mining frameworks. The proposed approach integrates rule processing, knowledge discovery in the form of substitution mining, and visualization into a cohesive pipeline, providing improved scalability and interpretability.

1.4 Thesis Outline

The remainder of this thesis is structured as follows: Chapter 2 provides a comprehensive review of the related literature, focusing on ARM, substitute item mining, and data structures used in rule processing. Chapter 3 introduces the proposed data structure and details its design and implementation. Chapter 4 presents the methodology developed for substitute item mining and discusses its integration with the data structure. Chapter 5 explores advanced visualization techniques for ARM results and discusses their practical applications. Chapter 6 presents a case study that demonstrates how all research questions are interconnected. Chapter 7 concludes the thesis by summarizing the key findings and contributions and discussing directions for future research. Finally, an Appendix contains the algorithms discussed in this study for reference and implementation details.

Publications

During the course of this research, two publications were produced, each representing significant milestones corresponding to specific research questions addressed in this thesis.

- **Research Question 1 (RQ1):** Mikhail Kudriavtsev, Vuong M Ngo, Mark Roantree, Marija Bezbradica, and Andrew McCarren (2024). "Exploring the trie of rules: a fast data structure for the representation of association rules". In: *Journal of Intelligent Information Systems*, pp. 1–21

  This publication presents the development of the Trie of Rules data structure, addressing the efficiency and scalability challenges in storing and processing association rules, as discussed in Chapter 3 of this thesis.

- **Research Question 3 (RQ3):** Mikhail Kudriavtsev, Andrew McCarren, H. Lee, and Marija Bezbradica (2024). "Efficient Visualization of Association Rule Mining Using the Trie of Rules". In: *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1: KDIR*, pp. 72–80. ISBN: 978-989-758-716-0

  This work focuses on the application of the Trie of Rules data structure for enhancing the visualization of association rule mining results, corresponding to the methodologies and findings presented in Chapter 5.

Chapter 2

Background and Literature Review

Chapter 2 provides a comprehensive review of the existing literature relevant to this research. It begins by introducing the foundations and applications of Association Rule Mining, followed by a discussion of the challenges encountered in processing ARM results. The chapter then examines data structures commonly used in ARM, including those for storing transactions and association rules. Subsequently, it reviews current methodologies for substitution item mining and visual representation of ARM outcomes. By identifying key limitations in scalability, efficiency, and interpretability, this chapter establishes the motivation and context for the contributions developed in later chapters.

2.1 Introduction to Association Rule Mining

Association Rule Mining is a fundamental technique in data mining and is predominantly used to uncover relationships between items in large datasets (Agrawal, Imieliński, *et al.* 1993). The concept of ARM can be traced back to the pioneering work of Hájek *et al.* (1966), who introduced the GUHA method for automatic hypotheses determination. Their approach was one of the earliest attempts to systematically identify associations between properties in a dataset, laying the groundwork

for what would later become known as association rule mining.

Although Hajek et al.'s work provided a theoretical foundation, the practical application of ARM gained significant attention with the introduction of the Apriori algorithm by Agrawal, Imieliński, *et al.* (1993). The Apriori algorithm efficiently generates frequent itemsets by leveraging the downward closure property, marking a major advancement in the field. This development popularized ARM, particularly in the context of market basket analysis, where the goal is to discover co-occurrence relationships among items purchased together in transaction databases.

Over the years, ARM has evolved beyond its initial scope, expanding into diverse areas such as web usage mining, bioinformatics, and more (Tan, Steinbach, *et al.* 2019; Misman *et al.* 2020). The development of algorithms tailored for specific applications and the introduction of various measures such as Confidence, Lift, and Conviction have further enhanced the relevance and utility of ARM across different domains. Recent advances continue to focus on improving efficiency, scalability, and applicability to complex and large-scale datasets (Yazgana *et al.* 2016; Kaushik *et al.* 2021).

Following Agrawal's contribution, the field of ARM expanded rapidly, with significant milestones including the development of the FP-Growth algorithm by Han, Pei, and Yin (2000). FP-Growth addressed some of the computational inefficiencies of Apriori by using a divide-and-conquer strategy to compress the dataset into a compact data structure called the FP-tree, enabling faster pattern mining without candidate generation.

Through these developments, ARM has solidified its position as a critical tool in data mining, providing insights that drive decision-making in a wide range of domains. This subsection provides a brief overview of the key milestones in the evolution of ARM, setting the stage for a deeper exploration of its current challenges and advancements.

2.1.1 Applications of ARM

Association Rule Mining has proven to be a versatile and powerful tool across a wide range of industries, enabling the extraction of valuable insights from large datasets. One of the most well-known applications of ARM is in the retail industry, particularly in market basket analysis. ARM is used to uncover associations between products that are frequently purchased together, enabling retailers to optimize inventory, enhance cross-selling strategies, and design effective promotional campaigns (Agrawal, Imieliński, *et al.* 1993).

Beyond retail, ARM has been successfully applied in the healthcare industry, where it aids in the discovery of relationships between various medical conditions, treatments, and patient outcomes. For example, by analyzing patient records, ARM can identify patterns that suggest correlations between symptoms and diseases, or between treatments and recovery rates. Information about this relationship is crucial for developing treatment guidelines, enhancing patient care, and supporting clinical decision-making (Rajak *et al.* 2008; Misman *et al.* 2020).

In the financial sector, ARM is employed to detect patterns in transaction data that may indicate fraudulent activities. By identifying associations that deviate from normal transaction patterns, financial institutions can flag potentially fraudulent behavior more effectively, thereby enhancing security measures and reducing financial losses (Albashrawi 2016).

ARM is also utilized in web usage mining, where it helps to understand user behavior by analyzing the patterns of web page visits. This application is particularly valuable for enhancing user experience, personalizing content, and improving website navigation (B. Kumar *et al.* 2022).

Furthermore, ARM has applications in bioinformatics, where it is used to identify relationships between genetic markers and phenotypic traits. This can lead to significant advancements in understanding genetic predispositions to certain diseases, thereby contributing to the development of personalized medicine (Naulaerts *et al.* 2015; Misman *et al.* 2020).

The diverse applications of ARM across these industries highlight its importance as a tool for knowledge discovery, enabling organizations to make data-driven decisions that enhance efficiency, profitability, and innovation.

2.1.2 Mathematical Foundations of ARM

A solid understanding of the mathematical foundations of ARM is essential for exploring its applications and advancing its methodologies. ARM relies on a structured series of stages. Below, the primary elements and stages of ARM are introduced to provide a foundational framework for further discussion.

Key Components in ARM

The following concepts are central to ARM, forming the basis of association rule generation and analysis:

Transactional Data: Transactional data refers to collections of transactions, where each transaction is a set of items purchased or observed together. This data serves as the foundational input for ARM, enabling the identification of patterns and relationships between items (Han, Pei, Yin, and Mao 2004).

Frequent Itemsets: Frequent itemsets, as defined by Agrawal, Imieliński, *et al.* (1993), refer to sets of items that appear together in transactions with a frequency above a specified threshold. These itemsets consist of items that commonly co-occur without any particular order. Identifying frequent itemsets is fundamental in ARM as it serves as the basis for generating association rules that reveal significant patterns in datasets.

Association Rules: An association rule, originally formalized by Agrawal, Imieliński, *et al.* (1993), is an implication of the form $X \rightarrow Y$, suggesting that the presence of itemset X (referred to as the **antecedent**) implies the presence of itemset Y (referred to as the **consequent**) in a transaction. The significance of an association rule is measured by various metrics that help to assess its relevance and strength within a domain-specific context (Hahsler 2024).

Rulesets: A ruleset, as commonly used in ARM studies (Padua *et al.* 2014; Tan, Steinbach, *et al.* 2019; Máša *et al.* 2024), refers to the collection of all association rules generated from the frequent itemsets. Due to their complexity and size, rulesets often require further filtering or pruning to retain only the most relevant rules for decision-making. Advanced ARM techniques also focus on optimizing rulesets to reduce redundancy and improve interpretability (Y. C. Chen *et al.* 2015; Hahsler 2016; De Padua *et al.* 2018; Ruiz *et al.* 2020).

Core Stages in the ARM Process

The process flow of ARM, as outlined in foundational and recent literature (Agrawal, Imieliński, *et al.* 1993; Han, Pei, Yin, and Mao 2004; S. Zhang *et al.* 2008), includes the following steps:

1. **Data Preparation:** This initial step involves preparing the dataset for analysis, a critical process that includes data cleaning, transformation into an appropriate format, and selecting relevant attributes. This step ensures that the data is both accurate and suitable for identifying associations.

2. **Frequent Itemset Generation:** At the core of ARM is the task of identifying frequent itemsets, which are groups of items that appear together in transactions with a frequency that meets or exceeds a user-specified minimum Support threshold. Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items, and let a transaction T be a subset of I. The Support of an itemset $X \subseteq I$ is defined as:

$$\text{Support}(X) = \frac{|\{T \in D \mid X \subseteq T\}|}{|D|} \tag{2.1}$$

   where D represents the set of all transactions. Itemsets that satisfy the minimum Support threshold are deemed frequent itemsets.

3. **Rule Generation:** After identifying frequent itemsets, association rules are generated from these itemsets. An association rule takes the form $X \rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \emptyset$. The rule implies that the presence

of X in a transaction suggests the presence of Y. Rule strength is typically evaluated using Confidence and Lift:

$$\text{Confidence}(X \rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)} \tag{2.2}$$

$$\text{Lift}(X \rightarrow Y) = \frac{\text{Confidence}(X \rightarrow Y)}{\text{Support}(Y)} \tag{2.3}$$

Confidence indicates the reliability of the rule, while lift assesses the rule's significance relative to random co-occurrence (Hahsler 2024).

4. **Ruleset Formation:** The complete set of rules generated from frequent itemsets is termed a ruleset. This ruleset serves as the foundation for further analysis and decision-making. It can be refined using additional metrics such as conviction or leverage (Hahsler 2024), which provide deeper insights depending on the analysis requirements.

5. **Evaluation and Interpretation:** The final stage involves evaluating and interpreting the generated rules. Leveraging both statistical measures and domain knowledge, this stage is essential for extracting actionable insights and ensuring the rules have real-world applicability.

The diagram in Figure 2.1 provides a simplified representation of the process.

Figure 2.1: Process flow of Association Rule Mining.

2.1.3 Traditional ARM Methodologies and Limitations

ARM involves identifying relationships between items in transactional datasets. Several foundational algorithms have been developed for mining association rules, with

Apriori and FP-Growth being among the most well-known. Each of these algorithms offers unique solutions to the challenges posed by the high dimensionality and large size of real-world datasets. Below, we expand on these key algorithms and their operational principles.

Apriori Algorithm

The Apriori algorithm, introduced by Agrawal, Imieliński, *et al.* (1993), is one of the most widely used methods for frequent itemset mining. The core idea behind Apriori is to utilize the *downward closure property*, which states that if an itemset is frequent, all of its subsets must also be frequent. This property allows the algorithm to prune the search space and avoid examining itemsets that are unlikely to be frequent.

The algorithm works in a multi-phase process:

1. **Frequent Itemset Generation:** The algorithm starts by identifying frequent 1-itemsets (i.e., individual items that meet the minimum support threshold). Next, it iteratively generates candidate itemsets of size 2, 3, and so on, by joining frequent itemsets from the previous iteration.

2. **Pruning:** At each step, candidate itemsets that do not meet the minimum support threshold are discarded.

3. **Rule Generation:** Once all frequent itemsets are identified, the algorithm generates association rules in the form $A \rightarrow B$, where A and B are itemsets. A rule is considered strong if its Confidence or Support exceeds a predefined threshold.

While Apriori is conceptually simple and widely applicable, it is computationally expensive due to the need to scan the entire dataset multiple times and generate a large number of candidate itemsets, especially for datasets with many unique items (Han, Pei, and Yin 2000).

FP-Growth Algorithm

FP-Growth (Frequent Pattern Growth) (Han, Pei, and Yin 2000) is an efficient alternative to Apriori that avoids the need for candidate generation and reduces the number of passes over the dataset. The key idea behind FP-Growth is to represent the dataset in a compressed prefix tree structure called the **FP-tree**, which stores itemset frequency information in a compact form.

The FP-Growth algorithm operates as follows:

1. **Building the FP-tree:** The algorithm begins by scanning the dataset once to identify the frequent 1-itemsets. These itemsets are then arranged in a *sorted order* (according to frequency). A compact tree structure is constructed where each node represents an item and edges represent itemset co-occurrences across transactions.

2. **Recursive Mining:** After the FP-tree is constructed, the algorithm recursively mines frequent itemsets by projecting the database based on conditional pattern bases, which are smaller databases created from the FP-tree. This process continues recursively for each frequent itemset.

FP-Growth significantly improves over Apriori by eliminating candidate generation and reducing the number of database scans. It is more efficient for large and dense datasets but can still face memory issues when the FP-tree becomes very large.

Beyond the Apriori and FP-Growth algorithms, other methodologies like Eclat (Zaki *et al.* 1997) and H-Mine (Pei *et al.* 2001) have been developed to further optimize ARM processes. Eclat utilizes a vertical data format to speed up frequent itemset mining, while H-Mine leverages a hyper-structure mining algorithm for scalability. Despite their innovations, these methods also face challenges related to scalability and memory usage, particularly with high-dimensional data.

The limitations of these traditional methodologies highlight the ongoing need for more efficient and scalable ARM techniques. As datasets become increasingly large

and complex, developing new methodologies that can handle these challenges without compromising performance remains a critical area of research. The importance of advancing ARM algorithms is underscored by a continuous stream of recent research dedicated to optimizing performance, such as by improving execution speed, memory efficiency, and scalability (Lajus *et al.* 2020; Moslehi *et al.* 2020; Zheng *et al.* 2023).

Variants of Association Rule Mining

In addition to the traditional algorithms focused on discovering frequent itemsets, a wide range of ARM variants has been developed to address domain-specific requirements and data characteristics. These variants extend the classical ARM framework by modifying either the structure of the rules, the nature of the data, or the constraints on the mining process.

One important extension is **Class Association Rule Mining (CAR)**, which focuses on generating rules where the consequent is a class label, making it particularly useful for classification tasks (W. Li *et al.* 2001). **Multi-support ARM** allows different minimum support thresholds for different items (B. Liu *et al.* 1999), addressing the problem that rare but important items may be ignored under a single global threshold. **Quantitative ARM** (also known as **numeric ARM**) deals with numerical attributes by discretizing them or applying techniques like fuzzy logic or clustering to uncover meaningful associations involving continuous data (Srikant *et al.* 1996). **Sequential ARM** captures rules over ordered itemsets – essential for applications such as clickstream analysis or medical treatment planning (Agrawal and Srikant 1995). Other variants include **Weighted ARM**, which accounts for item importance; **High Utility Itemset Mining**, which incorporates item profit or utility instead of frequency (Y. Liu *et al.* 2012); and **Constraint-based ARM**, where rule generation is guided by user-defined syntactic or semantic constraints (Ng *et al.* 1998).

These extensions reflect the versatility of ARM and its adaptation to a wide

variety of applications beyond traditional market basket analysis. While each variant introduces algorithmic changes to suit its goals, they all share the fundamental objective of uncovering interpretable, useful patterns within large datasets.

2.2 Processing ARM Results and Associated Challenges

Generating association rules through ARM is not the end of the analytical process. The next critical step involves using these rules to derive meaningful insights, interpret them effectively, and potentially uncover actionable knowledge. This step, known as *post-mining* (Zhao *et al.* 2009), is essential for translating raw rules into valuable information that can inform decision-making.

Over the years, various methods have been developed to process ARM results, including:

- **Metrics:** Numerous metrics have been introduced to evaluate the strength and significance of association rules. While Support and Confidence are the foundational metrics, others like Lift, Leverage, Conviction, and χ^2 (the chi-squared test) provide different perspectives on the data. Each metric serves a specific purpose, capturing various aspects of the relationships between items (Tan, V. Kumar, *et al.* 2004).

- **Filtering:** To manage the vast number of generated rules, filtering techniques are employed based on thresholds for Support, Confidence, Lift, or other metrics. This helps in focusing on the most significant rules and reduces the complexity of the ruleset (Brin *et al.* 1997).

- **Clustering:** Clustering methods group similar rules together, enhancing interpretability and allowing analysts to identify broader patterns within the data. Techniques such as hierarchical clustering, k-means, and density-based

clustering have been successfully applied to association rules (Geng *et al.* 2006; Padua *et al.* 2014; Danh Bui-Thi *et al.* 2020).

- **Substitute Item Mining:** The process of identifying items that can serve as replacements for others, fulfilling similar needs, plays a critical role in retail and recommendation systems. By uncovering potential substitutes, businesses can better understand product competition and make informed decisions regarding inventory management and marketing strategies (McAuley *et al.* 2015; Achanuparp *et al.* 2016; Sethi *et al.* 2018; Ruiz *et al.* 2020; Akkoyunlu *et al.* 2020; Tkachuk *et al.* 2022).

- **Generalization and Hierarchies:** Utilizing taxonomies or hierarchies allows for the generalization of rules to broader categories, which can simplify analysis and reveal higher-level patterns (Sethi *et al.* 2018).

Despite these methods, several common challenges persist in processing ARM results.

2.2.1 Complexity of ARM Results

Association rule mining often results in the generation of a vast number of rules, especially with large and complex datasets (Cornelis *et al.* 2006; Y. C. Chen *et al.* 2015; Ruiz *et al.* 2020). This complexity is attributed to the combinatorial nature of itemset generation and the low thresholds that might be set for Support and Confidence to capture rare but significant associations. The sheer volume of rules can be overwhelming, making it difficult for analysts to sift through and identify the most meaningful ones.

Moreover, many of these rules may be redundant or represent trivial associations (Berrado *et al.* 2007; Y. C. Chen *et al.* 2015). The presence of such noise in the ruleset further complicates the task of extracting valuable insights. Additionally, the relationships captured by the rules may be non-intuitive or context-dependent, requiring domain expertise for proper interpretation.

2.2.2 Scalability and Efficiency Issues

As datasets grow in size and complexity, scalability becomes a significant concern in ARM (Agrawal, Imieliński, *et al.* 1993; Savasere *et al.* 1998; Lajus *et al.* 2020). Traditional algorithms may not scale well with the increasing volume of rules, leading to lengthy processing times and high computational resource requirements. This issue is exacerbated when dealing with high-dimensional data, where the number of possible itemsets grows exponentially.

Efficiency issues also arise in the post-mining phase, where processing and interpreting the large rulesets demand significant time and computational power (Y. C. Chen *et al.* 2015; Moahmmed *et al.* 2021; Amit Pande *et al.* 2022). The need for real-time or near-real-time analysis in certain applications adds to the urgency of addressing these scalability and efficiency challenges.

2.2.3 Existing Approaches to Handling ARM Results

Various methods have been proposed to tackle the challenges associated with processing large and complex ARM results:

Rule Pruning and Summarization: Techniques such as pruning redundant or insignificant rules help reduce the size of the ruleset (Toivonen *et al.* 1995; Alasow *et al.* 2020). Summarization methods aim to represent the ruleset concisely without significant loss of information.

Interestingness Measures: Introducing additional metrics to assess the interestingness or unexpectedness of rules can help focus on the most valuable associations (Kontonasios *et al.* 2012; Danh Bui-Thi *et al.* 2020). These measures consider statistical significance, novelty, and relevance to the domain.

Visualization Techniques: Visual representations of rules can aid in identifying patterns and relationships that are not immediately apparent from textual data, as reviewed by Jentner *et al.* (2019). Visualization helps simplify complex data and supports exploratory analysis.

Clustering and Grouping: Grouping similar rules using clustering algorithms

can make the ruleset more manageable (Hahsler 2016; De Padua *et al.* 2018; Danh Bui-Thi *et al.* 2020). These approaches help identify common themes or patterns within the data.

Constraint-Based Mining: Applying constraints during the mining process, such as focusing on specific itemsets or rules that meet certain criteria, can reduce the number of generated rules (Roberto J Bayardo *et al.* 2000; C. K.-S. Leung 2009).

Use of Advanced Data Structures: Implementing efficient data structures can enhance the speed and scalability of ARM algorithms (Han, Pei, and Yin 2000). Structures like FP-trees and hash tables optimize data storage and retrieval operations.

While these approaches offer partial solutions, there is still a need for comprehensive methods that address efficiency, scalability, and interpretability simultaneously. This gap underscores the importance of developing new techniques and data structures that can effectively manage and utilize large ARM results.

2.3 Data Structures in ARM

Efficient storage and retrieval of transactional data and association rules are critical for knowledge extraction in Association Rule Mining. Various data structures have been proposed to optimize the management of transactions and association rules, each with specific strengths and limitations. The following discussion provides an overview of key data structures used for storing transactions and association rules, highlighting their distinct characteristics and applicability.

2.3.1 Data Structures for Storing Transactions

The storage and retrieval of *transactional data* play a crucial role in ARM, as they form the foundation for identifying frequent itemsets, which are essential for generating association rules. Various data structures have been developed for transaction storage, each designed to address specific requirements in ARM. This section ex-

amines several prominent data structures and evaluates their suitability for ARM tasks.

Linked Lists or Vertical Database

Zaki *et al.* (1997) proposed parallel algorithms for discovering association rules using a vertical database layout. This layout organizes transactions by item, followed by its transaction ID list (TID-List), which lists transactions containing the item. While effective for support-based frequent itemset mining, this structure is not optimized for managing association rules or incorporating metrics beyond Support.

In another approach, Xiaobing Liu *et al.* (2012) introduced a trifurcate linked list storage structure for directed itemsets,where itemsets are treated as ordered pairs reflecting the direction of association between items, thereby enhancing ARM efficiency. While this method focuses on frequent itemset storage rather than association rule management, it optimizes data related to itemsets, including TID-Lists and Support values. Their structure demonstrates the potential of specialized data layouts to streamline itemset retrieval, yet highlights a need for further innovations specifically targeting association rule storage and retrieval. While effective for frequent itemset mining, it does not address broader needs such as managing additional metrics or relationships between antecedents and consequents, limiting its utility in comprehensive ARM processes.

Trees

Coenen *et al.* (2004) proposed the use of T-Trees and P-Trees as data structures for storing itemsets in ARM. Although these structures focus on itemset storage rather than association rule management, they are highly efficient for frequent itemset mining. T-Trees and P-Trees, optimized for support-based tasks, are limited to the mining phase and do not handle actual rules or metrics such as Confidence.

Vu *et al.* (2011) introduced the FEM algorithm, which employs the FP-tree data structure and TID-Lists to efficiently mine frequent patterns from transactional

databases. While effective for itemset mining, the FEM algorithm does not explicitly store association rules or manage additional metrics like lift.

Shabtay *et al.* (2021) proposed the guided FP-Growth algorithm, which combines FP-tree and TIS-Tree (Targeted Item-Set Tree) structures to optimize itemset mining. This approach efficiently mines itemsets of interest, particularly in imbalanced datasets, yet does not manage association rule metrics beyond Support.

Item Graphs

The Items Graph, presented by Koh *et al.* (2010), represents interactions between items in a dataset through nodes (items) and edges (strength of interactions). Similarly, Yen *et al.* (2001) discussed graph-based methods for discovering association rules, demonstrating how graphs can effectively represent item relationships. While advantageous for itemset analysis, graph-based structures are limited to support-based tasks and do not directly store association rules or additional metrics.

Summary

Data structures such as linked lists, trees, and item graphs are optimized for storing transactional data and supporting frequent itemset mining. However, these structures primarily serve the mining phase of ARM and often lack capabilities for handling association rules directly or incorporating metrics beyond Support. This work aims to address these limitations by focusing on efficient storage and manipulation of *association rules*, rather than solely on transactions or itemsets.

2.3.2 Data Structures for Storing Association Rules

Efficient storage and retrieval of association rules are essential for knowledge extraction in ARM. Various data structures have been developed specifically for managing association rules, each with unique advantages and limitations. The following sections explore some key structures for storing association rules, focusing on their applicability to ARM tasks. A more detailed discussion, including a comparison with

the novel data structure proposed in this thesis, is presented in Chapter 3, where design rationale, structural properties, and efficiency considerations are examined in depth. This background lays the foundation for understanding how different structures support or limit post-mining operations, and highlights the need for more integrated and scalable representations.

Data Structures in R and Python

Hahsler, Grun, *et al.* (2005) discussed the *Arules* package in R, which employs data frame structures to manage transactions, itemsets, and rules in ARM. The package provides an interface for processing association rules efficiently, utilizing classes like TID-Lists and rulesets.

Stancin *et al.* (2019) reviewed free Python libraries for data mining, emphasizing *Pandas* (McKinney 2010) as a popular tool for handling tabular data, including association rules. Similarly, Hahsler (2023) introduced *Arules.py*, a Python package that integrates *Pandas* data frames to manage and analyze association rules, providing a structured approach to rule storage and processing.

Rules Graphs

De Padua *et al.* (2018) discussed using similarity matrices to facilitate community detection and clustering in association rules. This graph-based method, where each rule is a node, supports clustering and community detection tasks. However, it is limited to clustering applications and does not prioritize efficient storage of association rules, which is crucial for scalability and quick retrieval, especially when dealing with large datasets and real-time analytical tasks.

Jin *et al.* (2020) proposed the Bundle Graph Convolutional Network (BGCN) model, which employs a heterogeneous graph for multi-behavior recommendation. This approach captures intricate item relationships, though it requires extensive domain knowledge, which can limit its applicability in ARM.

Rule Lists and Trees

Berrado *et al.* (2007) proposed a method for organizing discovered association rules using metarules, which are higher-level abstractions of rules, and storing them in a graph structure. This approach facilitates the grouping and organization of rules by capturing relationships and patterns within the dataset. However, it does not explicitly address the efficiency of storage or retrieval.

D. Bui-Thi *et al.* (2022) introduced MoMAC, an optimization-based approach for combining association rules in a rule list data structure. This ordered rule list efficiently optimizes classifier size and prediction accuracy, although it focuses on rule combination rather than on individual rule storage.

Alternative Data Structure Approaches

Several alternative approaches to data structures focus on specific tasks such as rule interpretation, visualization, clustering, and distributed processing. Y. Li *et al.* (2014) introduced a multi-tier granule mining approach, utilizing granules at different tiers to interpret association rules. While effective for interpretation, this method focuses on granular representation rather than explicitly storing association rules.

Visualization techniques have also been explored by Jentner *et al.* (2019), who examined matrix and graph-based representations. However, these approaches prioritize visual clarity and usability over the analysis of underlying storage structures.

In the domain of clustering, Danh Bui-Thi *et al.* (2020) proposed using feature vectors representing semantic and lexical relationships to group association rules. Although this method improves clustering, it does not address the storage of ARM metrics or manage rule relationships beyond the clustering application.

For big data environments, Moahmmed *et al.* (2021) utilized Hadoop MapReduce to cluster association rules, employing key-value pairs for distributed processing. While suitable for large-scale data, this approach does not emphasize the details of data structures used for managing the rules themselves.

While these methodologies address clustering or multi-tier structures, efficient data structures tailored specifically for storing and representing association rules remain underexplored as noted above. Most existing approaches utilize generic data frames or focus on rule combination without optimizing for rule-specific characteristics. Developing specialized data structures for association rules is essential for enhancing speed, memory efficiency, and knowledge extraction capabilities, ultimately expanding ARM's potential applications.

2.4 Substitution Item Mining in ARM

Understanding substitution patterns is crucial in various domains, particularly in retail and marketing, where it informs inventory management, pricing strategies, and recommendation systems. This section explores the concept of substitution from both consumer behavior and machine learning perspectives.

2.4.1 Concept of Substitution

From the perspective of consumer behavior and psychology, substitution can be defined as the act of choosing one product over another that fulfills a similar need or desire (Kotler *et al.* 2016). Substitute items are products or services that a consumer perceives as sufficiently similar to another product or service and which meet the same needs or desires (Lewin 1936; Hamilton *et al.* 2014). For example, tea and coffee can be considered substitutes for each other because they both serve the purpose of providing a caffeinated beverage to start the day.

However, in the field of economics, substitution is often defined in terms of price elasticity. Products are considered substitutes if a decrease in the price of one leads to a decrease in the demand for the other (M. Zhang *et al.* 2020). This economic perspective highlights the competitive relationship between products in the marketplace.

In the field of machine learning, substitution is often analyzed by examining

transactional data to determine how items co-occur or replace one another. Association Rule Mining plays a critical role in this analysis by identifying frequent patterns and uncovering the relationships between items. These patterns provide valuable insights into substitution behavior, enabling the discovery of items that are likely to serve as alternatives based on their observed associations (Y. C. Chen *et al.* 2015).

2.4.2 Substitution in Machine Learning

In machine learning, substitution is analyzed using statistical and computational methods to identify patterns of interchangeability between items. Association Rule Mining plays a critical role in this analysis by uncovering patterns and associations between items based on their co-occurrence.

For instance, Tian *et al.* (2021) define substitution as products that can replace each other in consumer choices by analyzing the connections between products in a bipartite product-purchase network. Substitutability is measured based on the similarity of shared complements, using cosine similarity between complementarity scores.

Y. Huang *et al.* (2021) define substitute items as products within the same category that can fulfill a similar need or function as the focal product. Their experiments demonstrated how different products within the same category could serve as substitutes by observing how recommendations influenced customer evaluations and purchase intentions.

Several studies employ advanced techniques to identify substitute items. For example, Yu et al. use similarity-based approaches, collaborative filtering, graph-based methods, and embedding space techniques to enhance recommendation models by improving their understanding of user preferences and item relationships (Yu *et al.* 2024). Similarly, Reddy *et al.* (2022) classify products into *Exact*, *Substitute*, *Complement*, or *Irrelevant* matches for queries using a multi-class classification problem

evaluated with an F1 score[1], thereby enhancing search result relevance and understanding user intent.

Another approach involves using topic models trained on product data, as demonstrated by McAuley *et al.* (2015). Their system, *Sceptre*, identifies substitute relationships by analyzing product features and generates ranked lists of potential substitutes based on the inferred relationships between items and features.

Similarly, Lin *et al.* (2020) focus on clothing outfit complementary item retrieval, defining substitute items as those that can replace or complement a missing item in an outfit. Their framework employs a category-based subspace attention network to find compatible substitutes based on multiple attribute dimensions.

Further contributions include the Cleora algorithm introduced by Tkachuk *et al.* (2022), which creates latent embeddings for products using transactional data to identify substitutes. This approach enables efficient training and accurate identification of substitute items by analyzing purchasing patterns. Additionally, Amit Pande *et al.* (2022) explore substitution techniques for grocery fulfillment and assortment optimization using product graphs. Their work focuses on providing relevant product recommendations and optimizing product assortments in stores.

While much of the literature on substitute item mining has focused on the relationships between items, the role of context in substitution decisions has received limited attention. Understanding how factors such as time, location, and individual preferences influence substitution can provide deeper insights that enhance existing models.

2.4.3 Contextual Analysis in Substitution

Contextual analysis adds depth to substitution by considering the circumstances under which consumer decisions are made (Shukla 2009; Hamilton *et al.* 2014; Wang *et al.* 2020). Previous research by Achanuparp *et al.* (2016) introduced the signifi-

[1]The F1 score is a metric used to evaluate the accuracy of a classification model. It is the harmonic mean of precision (the proportion of true positive predictions among all positive predictions) and recall (the proportion of true positives among all actual positives).

cant novelty of incorporating context into the analysis of substitution based on the consumption patterns of food pairs in similar contexts. However, their approach primarily treated context as an isolated feature and did not consider the dynamic interplay between items within that context.

Another study by Ruiz *et al.* (2020) analyzed contextual information to propose concepts of substitution and complementary products. While the method offered valuable insights, it required extensive additional information such as item attributes, product categories, brands, price variations, and other relevant features, along with detailed consumer preferences, including seasonal effects and price sensitivities. This reliance on comprehensive metadata can be a significant limitation, as not all datasets provide such detailed information.

In a slightly different domain, Pellegrini *et al.* (2021) explored substitution in the context of ingredient replacement in recipes using language models. Their approach involved generating contextualized embeddings for ingredients using models like FoodBERT, computing nearest neighbors for each ingredient embedding, and identifying potential substitutes based on the similarity of their embeddings. However, this method solely relies on Natural Language Processing (NLP)[2] techniques, overlooking important association patterns within the transactional dataset, such as item co-occurrence and frequency. These patterns, which form the basis of association rules, could provide additional insights into substitution decisions that NLP techniques alone may miss.

Therefore, there is a notable gap in the development of methods that can identify substitutions by considering the context in which substitution occurs, *without requiring extensive additional metadata or domain knowledge*. A methodology that captures the interplay of items within a specific context based solely on the frequency of their co-occurrence is needed. This would allow for a more universally applicable approach that is less dependent on detailed supplementary data.

[2]Natural Language Processing (NLP) is a field of artificial intelligence that focuses on enabling computers to understand, interpret, and generate human language. NLP techniques are widely used for tasks like text analysis, machine translation, and language modeling.

2.5 Visualization Techniques for ARM Results

2.5.1 Importance of Visualization in ARM

Visualizing ARM results is recognized as a challenging task, as indicated by surveys conducted by Hahsler and Chelluboina (2011), Fernandez-Basso *et al.* (2019), Jentner *et al.* (2019), Alyobi *et al.* (2020), Menin *et al.* (2021), and Fister *et al.* (2023). The complexity arises from the need to represent rules visually while considering the multitude of associated metrics and distinguishing between antecedents and consequents, leading to various proposed approaches.

Traditionally, rules are presented as plain tables or text-based methods due to their simplicity and familiarity. However, these methods often fail to effectively convey complex relationships.

Although various methods exist, they can be classified into three distinct groups: scatter plots, matrix-based methods, and graph-based methods.

The **scatter plot** approach, one of the more basic methods, was introduced by Roberto J. Bayardo *et al.* (1999). This method employs a two- or three-dimensional plot (Ong *et al.* 2002) to depict rules as dots. Although effective in handling a high number of rules, scatter plots lack insight into the structure of rules, requiring manual examination of the text-based representation of the original dataset. An example of scatter plot visuliastion is shown in Figure 2.2.

Matrix-based visualization, as presented by Hofmann *et al.* (2000), places antecedent and consequent sets on axes and displays metric values at their intersections. Despite its efficiency in revealing rule components, it suffers from scalability issues, particularly as the dataset size increases. A more modern implementation is provided by Varu *et al.* (2022).

An improvement to the matrix-based approach is the **grouped matrix-based visualization**, as proposed by Hahsler and Karpienko (2017), which alleviates size concerns by grouping similar rules. However, scalability remains a challenge. An example of grouped matrix based visuliastion is shown in Figure 2.3.

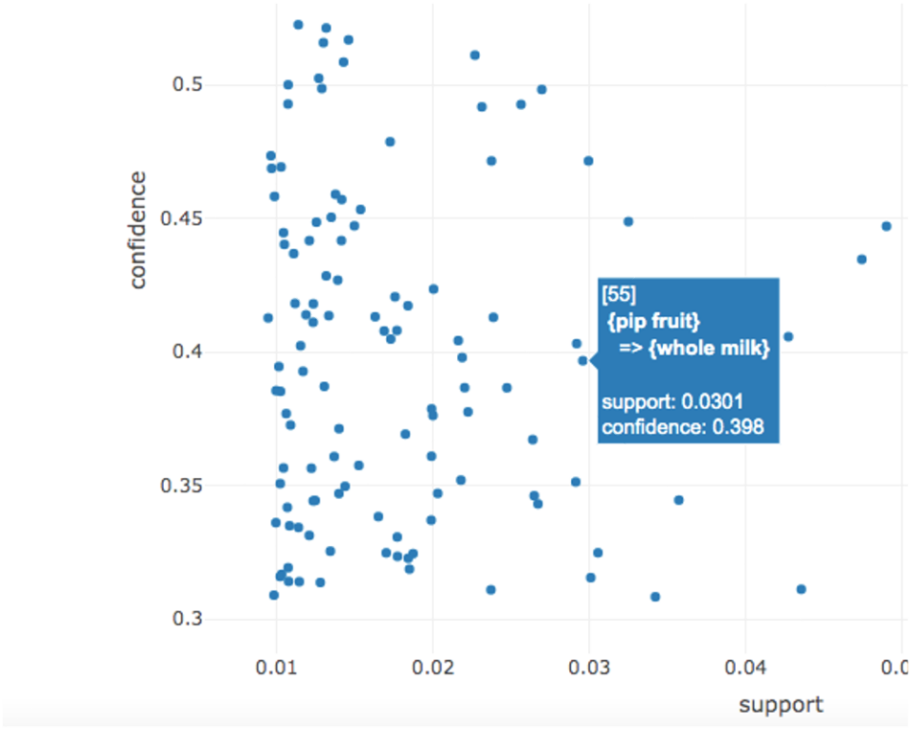


Figure 2.2: Example of a Scatter Plot visualization of association rules.

Figure 2.3: Example of a Matrix-Based visualization of association rules.

Graph-based visualization, widely employed in ARM (Klemettinen *et al.* 1994; Rainsford *et al.* 2000; Buono *et al.* 2005; Ertek *et al.* 2006; Fernandez-Basso *et al.* 2019; Alyobi *et al.* 2020; Menin *et al.* 2021), provides a clear representation of rule

structures. However, the main problem remains how to show all the items in a rule and distinguish between antecedents and consequents. This problem leads to either excessive size of the plot or low interpretability. Current methods rely on the idea that two types of nodes exist – items and rules. Items that go into (directed edge) the rule are antecedents, and edges that go out of a rule node are consequent. An example of such visualisation is shown in Figure 2.4.

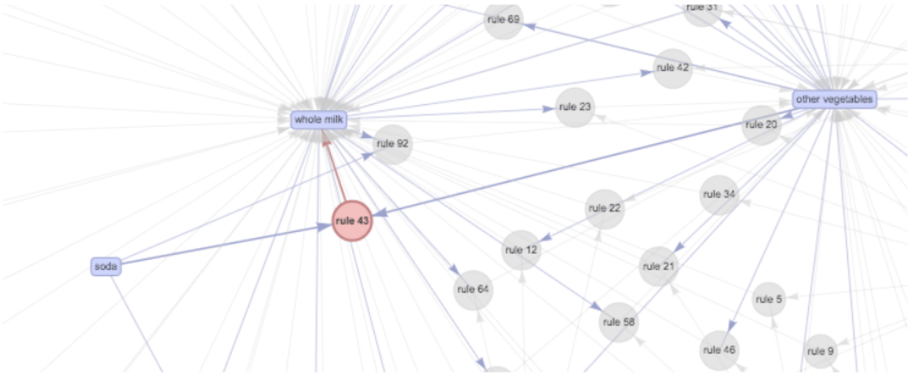


Figure 2.4: Example of a Graph-Based visualization of association rules.

These three main categories are implemented in popular libraries such as *arulesViz* for R (Hahsler and Karpienko 2017) and *arules.py* for Python (Hahsler 2023).

In conclusion, existing ARM visualization methods exhibit limitations in terms of scalability, interpretability, and representation of rule structures.

2.5.2 Visualization of Substitution Patterns

Visualizing substitution patterns presents specific challenges beyond those encountered in general ARM visualization. Substitution relationships often involve subtle and context-dependent interactions between items, which can be difficult to represent effectively.

Key challenges include:

- **Complex Relationships:** Substitution patterns may involve inverse relationships or negative correlations that are not easily captured in traditional visualization methods (Jentner *et al.* 2019).

- **Contextual Dependencies:** The substitutability of items can depend on various contextual factors such as time, location, or consumer preferences (Achanuparp *et al.* 2016). Visualizations need to accommodate these dimensions to provide meaningful insights.

- **Scalability:** Large datasets with numerous potential substitutions can result in cluttered and incomprehensible visual representations (Fernandez-Basso *et al.* 2019).

- **Interpretability:** Ensuring that the visualization is intuitive and actionable for decision-makers is critical. Overly complex or technical visualizations may hinder understanding (Menin *et al.* 2021).

The visualization of ARM results is vital yet challenging due to scalability, clarity, and interpretability constraints. Traditional methods like scatter plots, matrix-based, and graph-based visualizations offer unique strengths but face limitations.

Substitution patterns add complexity by introducing nuanced relationships and contextual dependencies that require adaptable, clear representations. Overall, these limitations highlight the need for advanced visualization techniques that balance scalability and interpretability, particularly in capturing complex substitution dynamics within ARM.

2.6 Gaps in the Current Literature and Summary

A review of the current literature in Association Rule Mining reveals critical gaps in addressing the challenges of post-mining processes comprehensively. Existing methods often target specific problems without considering how these solutions can be integrated into a holistic framework for knowledge discovery. Key gaps include:

- **Lack of Efficient Data Structures for Association Rules:** While data structures for transactional data are well-explored, specialized structures for managing association rules remain underdeveloped. Current approaches often

rely on generic data frames or focus primarily on transactions, which are not optimized for the complexity inherent in association rules.

- **Challenges in Substitution Mining Without Extensive Metadata:** Existing methodologies for substitution mining frequently depend on detailed metadata, such as product attributes and consumer preferences. This dependency restricts the applicability of available methods to datasets where extensive metadata is readily available, thus limiting their versatility.

- **Visualization Limitations in Handling Rulesets:** Existing visualization methods struggle with scalability and interpretability when dealing with large and complex rulesets, particularly for substitution patterns involving sophisticated relationships and contextual dependencies.

- **Limited Focus on the Post-Mining Process:** Current research often concentrates on discovery methodologies themselves, placing less emphasis on integrating preparation, storage, visualization, and presentation of association rules into a single framework. This siloed approach limits scalability, efficiency, and the ability to extract actionable insights.

Developing meaningful insights requires not only robust methodologies for knowledge discovery but also foundational data structures that enable efficient storage and retrieval. Optimizing data structures lays the groundwork for visualization and interpretation techniques that reveal patterns in an accessible and actionable manner. This layered approach – beginning with data structures, followed by methodology, and culminating in effective visualization and interpretation – forms the backbone of a holistic ARM post-mining process.

Summary and Research Direction: The identified gaps highlight the need for research aimed at creating:

- A specialized *data structure* optimized for association rules.

- A *substitution mining methodology* that does not require extensive metadata.

- Advanced *visualization techniques* capable of handling large rulesets and effectively displaying complex substitution patterns.
- *An integrated framework* combining these components to improve ARM's efficiency, scalability, and usability.

The subsequent chapters detail the methodology and contributions of this research, addressing these gaps by developing an integrated solution for ARM's post-mining processes.

Chapter 3

Data Structure for Efficient Processing of Association Rules

3.1 Introduction

In the context of Association Rule Mining, the efficiency of the post-mining process is critical, especially when dealing with large datasets that generate extensive rulesets. The computational complexity associated with storing, retrieving, and manipulating these rules is often a bottleneck in the practical application of ARM (Ait-Mlouk *et al.* 2017). Traditional data structures such as hash tables, lists, and trees, while effective for smaller datasets, struggle to scale efficiently when faced with the magnitude of modern data, as established in Chapter 2.

To address these challenges, this chapter introduces a novel data structure designed specifically to enhance the efficiency of processing association rules. The proposed data structure is tailored to optimize the storage, retrieval, and manipulation of large rulesets, significantly reducing the time and computational resources required for post-mining operations. This data structure serves as the foundation for the substitute item mining methodology discussed later in this thesis and aims to bridge the gap between ARM's powerful theoretical insights and its practical scalability issues.

The following sections outline the design principles of the proposed data structure, its implementation, and the specific advantages it offers over existing approaches. The theoretical underpinnings that guided its development are explored, emphasizing improvements in time complexity and space efficiency for managing association rules. A comparative analysis highlights the performance enhancements achieved through integration with existing ARM processes.

3.2 Methodology

3.2.1 Trie of Rules

Prefix-trees, such as the FP-tree data structure, are widely used in Association Rule Mining for storing frequent sequences (Bodon *et al.* 2003; Grahne *et al.* 2003; Han, Pei, Yin, and Mao 2004). However, prefix-trees have not been explored as a data structure for *storing association rules* along with their metrics, which can serve as a valuable alternative to popular "plain" data structures. While prefix-trees have been extensively applied for frequent pattern storage, their application for managing association rules introduces additional challenges, such as avoiding redundancy and maintaining the logical consistency of antecedents and consequents – particularly when the consequent is lengthy.

This study introduces a novel application of prefix-trees, also known as tries (as defined by Crochemore *et al.* (2009)), specifically adapted for storing and managing association rules. The proposed "Trie of Rules" data structure builds upon the foundational concept of prefix-trees but extends it by incorporating rule metrics and ensuring the consistent representation of logically equivalent rules. This innovation is particularly valuable for graph-based knowledge extraction methods and visualizations, as well as for efficient storage and retrieval of rules. While prefix-trees themselves are not new, their adaptation for storing association rules, including metric values and logical consistency checks, represents a novel contribution of this study.

The proposed data structure takes the form of a graph that contains all rules and their associated metric values. This structure avoids redundancy by ensuring that shared components of rules utilize the same path in the Trie, thereby increasing traversal speed and efficiency.

To ensure consistent storage of logically equivalent rules in the Trie of Rules, the items within the antecedent and consequent are preprocessed by sorting them according to their frequency in the frequent set before being inserted into the Trie. This approach guarantees that rules such as $a \rightarrow b$ and $ab \rightarrow c$, as well as $a \rightarrow b$ and $ba \rightarrow c$, are treated identically and stored along the same path in the Trie. By aligning logically equivalent rules, the Trie minimizes redundancy and enhances the efficiency of both rule storage and retrieval.

The methodology for using the Trie of Rules can be explained through a three-step process, with detailed pseudocode for the underlying algorithms provided in Appendix A:

1. **Step 1**: Apply an ARM algorithm to a transactional dataset to generate a list of frequent sequences (a dictionary of the form {list of items : Support}) and a ruleset (a list of Rule objects, see Appendix A, Algorithm 1).

2. **Step 2**: Preprocess the ruleset by sorting the items within the antecedent and consequent of each rule based on their frequency in the set of frequent sequences. Insert the rules into the Trie one by one (see Appendix A, Algorithm 2), using the ruleset and frequent sequences from Step 1 to construct a Trie object.

   This step builds a Trie object (Appendix A, Algorithm 3) using the list of frequent sequences. Each node (Appendix A, Algorithm 4) in the Trie represents a rule. A rule is encoded as a path from the root to a specific node, where the consequent corresponds to the final node in the path, and the antecedent comprises all nodes encountered along the path prior to the final node (see Fig. 3.1).

3. **Step 3**: Label every node in the Trie with metrics such as Support, Confidence, and other relevant parameters for the corresponding rule. These metrics are calculated as described in Appendix A, Algorithm 6 (see Fig. 3.1).

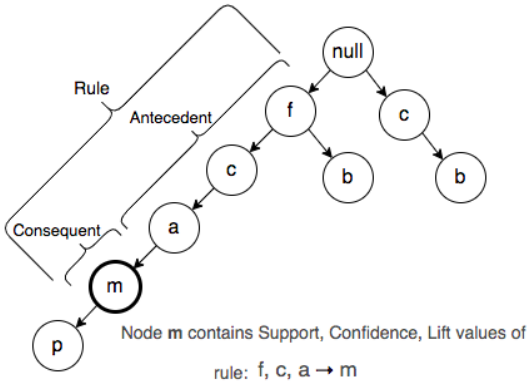


Figure 3.1: The structure of a rule in a Trie of Rules. A rule is represented as a path from the root to a chosen node, with the last node corresponding to the consequent and the preceding nodes representing the antecedents.

A simple breadth-first algorithm is used to retrieve rules. To retrieve a specific rule, refer to Appendix A, Algorithm 7, "Searching a Rule in the Trie."

3.2.2 An Illustrative Example

Let us consider an example from a simple dataset to illustrate the advantages of using the Trie of Rules approach.

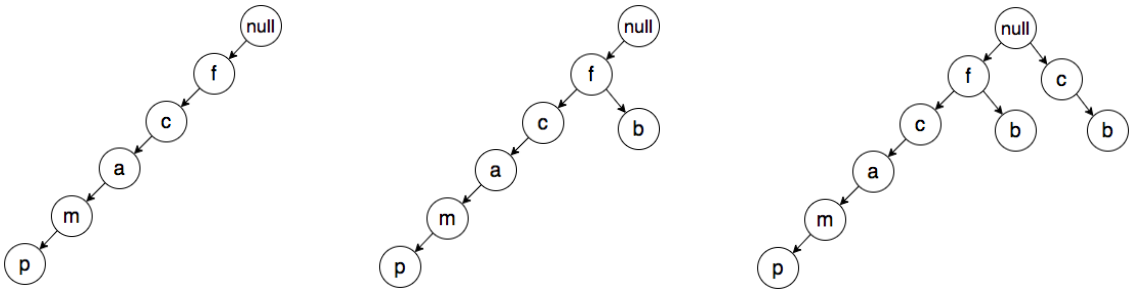
1. **Step 1**: In the first step, a ruleset is obtained through an ARM algorithm applied to the dataset. This can be done using any standard ARM algorithm, such as Apriori or FP-Max, as the choice of algorithm does not affect the process of creating the Trie of Rules. The ARM algorithm yields a list of frequent sequences and a set of rules (see Table 3.1).

2. **Step 2**: The list of rules serves as the source dataset for creating a Trie of Rules (see Appendix A Algorithm 2). The trie is initialized with a root node (Null), followed by the insertion of rules (see Appendix A Algorithm 5) from Table 3.1 into the trie, one by one. In this example, the first rule inserted

Table 3.1: Ruleset and Frequent Sequences

Ruleset		Frequent Sequences		
№	Rule	№	Frequent sequence	Support
1	f, c, a → m, p	1	f, c, a, m, p	0.1
2	f → b	2	f, c, a, m	0.15
3	c → b	3	f, c, a	0.2
		4	f, c	0.25
		5	f, b	0.2
		7	c, b	0.2
		6	f	0.5
		8	c	0.3
		9	b	0.25
		10	a	0.2
		11	m	0.15
		12	p	0.1

is $(f, c, a \rightarrow m, p)$. The items of the rule are inserted into the trie as nodes. Fig. 3.2a shows the trie after the first rule has been fully traversed. Subsequently, the second rule $(f \rightarrow b)$ is inserted into the trie. Note that the element f has occurred before, as seen in the trie created thus far. Therefore, instead of creating a new branch, the second rule overlays the existing trie and creates an additional branch only when the b item occurs. After traversing the second rule, the trie appears as shown in Fig. 3.2b. The last rule to be inserted is $(c \rightarrow b)$. Since this rule differs from others in terms of its first item, a new branch is created from the root. Finally, after traversing all the rules, the trie appears as shown in Fig. 3.2c.



(a) (b) (c)

Figure 3.2: Step 2: Insertion of rules $(f, c, a \rightarrow m, p)$, $(f \rightarrow b)$, and $(c \rightarrow b)$. The Trie of Rules after inserting the first (a), second (b), and third (c) rules is illustrated.

It is important to note that while a rule, for example, $(f, c, a \rightarrow m, p)$ is inserted into the trie, it is treated similarly to a frequent sequence for the purpose of traversal. This means that the antecedent (f, c, a) and the consequent (m, p) are sorted individually based on their frequency and then concatenated for insertion. Therefore, it appears in the trie as a sequence but maintains the distinction of being a rule due to the individual sorting and concatenation process. This approach ensures that the rules are correctly represented and stored within the Trie structure.

3. **Step 3**: Every node in the Trie of Rules is extended with metrics such as Support, Confidence, and any others corresponding to the rule that the node represents (see Appendix A Algorithm 6). Refer to Fig. 3.3 for illustration.

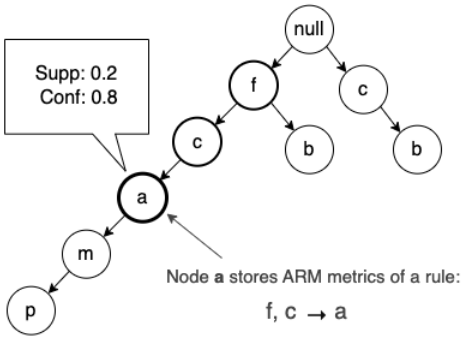


Figure 3.3: Step 3. ARM metrics of node *a*. Each node in the Trie of Rules contains metrics associated with the rule it represents, such as Support, Confidence, and Lift.

3.2.3 Confidence For Compound Consequents

The Confidence Transitivity Property was initially discussed by Luxenburger (1991) and Kryszkiewicz (2002), where the authors introduced a method for calculating the Confidence of compound consequents. This property allows the Confidence of a compound-consequent rule to be determined as the product of the Confidence values of the nodes in the consequent. The following equations illustrate this property:

$$Conf(a, b \rightarrow c, d) = \frac{Sup(a, b, c, d)}{Sup(a, b)} \tag{3.1}$$

$$Conf(a, b \rightarrow c) = \frac{Sup(a, b, c)}{Sup(a, b)} \tag{3.2}$$

$$Conf(a, b, c \rightarrow d) = \frac{Sup(a, b, c, d)}{Sup(a, b, c)} \tag{3.3}$$

To prove the point, Equation 3.1 can be derived by combining Equation 3.2 and Equation 3.3.

$$\begin{aligned} \text{Conf}(a, b \rightarrow c) \times \text{Conf}(a, b, c \rightarrow d) &= \frac{\text{Sup}(a, b, c)}{\text{Sup}(a, b)} \times \frac{\text{Sup}(a, b, c, d)}{\text{Sup}(a, b, c)} \\ &= \frac{\text{Sup}(a, b, c, d)}{\text{Sup}(a, b)} \\ &= \text{Conf}(a, b \rightarrow c, d) \end{aligned} \tag{3.4}$$

The Trie of Rules structure naturally employs this property for calculating the Confidence of compound consequents, allowing faster retrieval through the dataset. In a Trie of Rules, each node shows Confidence only for a rule with a single-item consequent; however, the proposed representation model can be used to derive the value of Confidence for more complex rules directly from the graph. This calculation is shown in Appendix A Algorithm 7, specifically in line 17. Figure 3.4 illustrates this concept.

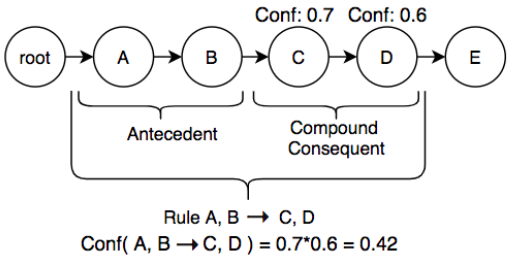


Figure 3.4: A Rule with a compound consequent. This figure illustrates a rule with a compound consequent and demonstrates that the Confidence for such a rule can be calculated by simply multiplying the Confidence values of the individual nodes in the consequent.

This feature is possible because of the specifics of a Trie of Rules. In order to

calculate Confidence of a rule, two values are used: Support of the antecedent and Support of the entire rule. As mentioned, the Trie of Rules is based on the prefix-tree structure. Hence, every path starting from the root is unique because identical sequences will overlay each other in one path. Consequently, when picking a node and observing a Support value in it, one can be sure that this value represents true Support for the sequence equal to the path to this node. Therefore, the calculation of Confidence does not require any information from other branches. All this allows us to multiply Confidence values for a sequence of nodes, which further allows us to evaluate rules with compound consequents in a Trie of Rules.

The next section presents an evaluation of the proposed data structure.

3.2.4 Limitations, Requirements, and Practical Considerations

While the Trie of Rules structure offers significant advantages in terms of efficient rule traversal, contextual querying, and integration with post-mining tasks, its implementation and maintenance come with several practical considerations.

Creation Overhead: Constructing the ToR involves preprocessing the rule set, parsing antecedent-consequent relationships, and inserting them into a hierarchical structure. This operation is more computationally intensive than simply loading rules from a CSV file into memory. However, this cost is a one-time expense and is offset by the substantial gains in retrieval performance and contextual analysis in downstream tasks.

Memory and Storage Requirements: The hierarchical and pointer-based nature of ToR results in higher memory usage compared to flat representations (e.g., ‘pandas‘ DataFrames). However, storage overhead remains tractable for real-world rule sets and can be mitigated through pruning strategies and selective node expansion.

Updating and Versioning: In dynamic environments where rules may be added or removed incrementally, ToR requires careful update procedures to maintain

structural integrity and avoid duplication. While this adds complexity compared to flat file formats, it also enables controlled versioning and modular updates – features that are essential for iterative knowledge discovery pipelines.

Comparison with Flat Structures: Simple formats such as CSV files or in-memory DataFrames offer ease of use and minimal setup costs but provide poor support for structural queries (e.g., prefix-based rule searches, subtree pruning, or clustering). These operations are critical for tasks such as substitution mining, where relational structure among rules encodes valuable semantic information. ToR enables such operations natively.

Handling Conflicting Rules: The current implementation of the Trie of Rules does not explicitly resolve or prioritize conflicting rules – cases where multiple rules share the same antecedent but lead to differing or contradictory consequents. While the data structure can accommodate such cases by storing all relevant consequents under the same node, it does not apply conflict resolution strategies such as rule weighting, ranking, or pruning. Addressing rule conflicts may require additional post-processing logic or integration with domain-specific criteria, which are considered out of scope for the current implementation but present a promising direction for future work.

Use of Graph Databases: Graph DBMSs such as Neo4j are theoretically well-suited for storing hierarchical structures like the Trie of Rules, as they support indexed traversal, parent-child relationships, and expressive querying through languages such as Cypher. Nevertheless, relying on an external DBMS introduces additional complexity, including setup overhead, dependency management, and potential performance trade-offs for high-frequency batch updates. For these reasons, the ToR in this work is implemented as a standalone, in-memory or lightweight persistent structure. This choice ensures portability, low-latency access, and direct control over traversal logic – qualities that are essential for the specific post-mining tasks addressed in this thesis.

In summary, while ToR introduces some overhead in creation and maintenance,

it offers considerable advantages for scalable and interpretable rule processing. Its benefits justify the added complexity, particularly in scenarios requiring structured rule exploration, substitution mining, and visualization.

3.3 Evaluation

Evaluating the proposed data structure is essential for assessing its effectiveness in advancing knowledge discovery methods. The following analyses are conducted:

- Analysis of search time for finding a rule in a ruleset.
- Assessment of how search time varies with different ruleset sizes.
- Evaluation of the memory efficiency of the data structure.

As many knowledge extraction algorithms involve traversing a dataset and searching for certain rules, improving the speed of these operations can substantially enhance the efficiency of downstream analytical tasks (Ait-Mlouk *et al.* 2017). To benchmark the performance of the Trie of Rules data structure, we compare it with widely used data structures implemented within `pandas` and `numpy`. As discussed in Section 2.3.2, these structures are representative of state-of-the-practice tools in association rule mining. `Pandas` DataFrames are commonly used for storing and filtering large rulesets in applied workflows, while `numpy` nd-arrays offer performance advantages in contexts involving matrix operations and numerical computations. Although neither is designed for hierarchical or contextual rule navigation, their popularity and efficiency in general-purpose data processing make them appropriate baselines. By comparing ToR against these standard tools, we aim to demonstrate the value of domain-specific structural design in supporting scalable and interpretable post-mining tasks.

All experiments were conducted on the same machine within the same environment. The machine used was a MacBook Air (2019) with a 1.6 GHz dual-core Intel

Core i5 processor, 8 GB of LPDDR3 RAM, a 256 GB SSD, and macOS Sonoma (version 14.5). The experiments were conducted within the JupyterLab environment, using Python 3.11, pandas version 2.1.4, and NumPy version 1.26.2.

3.3.1 Hash Table in Pandas DataFrame

The pandas library provides the DataFrame data structure (McKinney 2010; The Pandas development team 2024), designed for two-dimensional tabular data handling, akin to spreadsheets or SQL tables. Internally, pandas uses hash tables for indexing, enabling fast lookups and efficient data operations. With hash table indexing, pandas ensures swift data retrieval based on labels or position, facilitating operations such as filtering, aggregation, and transformation. This makes it a popular choice for storing association rules, as modern Python libraries for Association Rule Mining often default to pandas DataFrame for rule storage.

3.3.2 n-Dimensional Array in NumPy

NumPy (Harris *et al.* 2020) provides Support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

NumPy's ndarray (n-dimensional array) allows for efficient storage and manipulation of large datasets with its Support for multiple dimensions and various data types. It provides a versatile and efficient foundation for numerical computation and data manipulation tasks in Python.

The ndarray can serves as a data structure for storing association rules in a 2D matrix format with columns representing antecedent, consequent, Support, and Confidence.

3.3.3 Experiment on Grocery Dataset

To assess the proposed data structure, we utilized a grocery dataset sourced from the "arules" package within the R Project for Statistical Computing (Hahsler, Hornik,

et al. 2006). This dataset, originally introduced in the context of association rule mining, comprises transactional data collected from a local grocery outlet over a one-month period. With a total of 9,834 transactions and 169 unique items, this dataset provides a rich source of information for studying market basket analysis and deriving meaningful insights into consumer buying patterns.

Experiments were conducted to examine the average search time for finding a rule in a ruleset constructed with different pairs of Support and Confidence thresholds. All generated rules were used to construct the data structures. Using Algorithm 7 in Appendix A, which finds a rule and returns its metrics, all the rules were searched. This operation is commonly needed in knowledge discovery methods and for typical user queries to retrieve rule metrics. As shown in Figure 3.5, the average search time was compared across various Confidence and Support values, demonstrating how the size of the ruleset affects the search time. To cover a broad spectrum of ruleset sizes, ensuring both sparse and dense rulesets were included in the analysis, a range of 100 different minimum Support thresholds between 0.003 and 0.0135, along with four different Confidence values: 0.01, 0.1, 0.2, and 0.3, was chosen.

The results in Figure 3.5 clearly show that the Trie of Rules consistently outperforms the other data structures in terms of search time, particularly for large rulesets generated with lower Support thresholds. This demonstrates the scalability of the Trie of Rules, as it maintains efficient search times even as the ruleset size increases. In contrast, the Pandas DataFrame shows significant performance degradation as the ruleset grows, highlighting its inefficiency for handling large datasets. The NumPy-based approach performs better than the Pandas DataFrame but is still less efficient than the Trie of Rules. These findings underscore the suitability of the Trie of Rules for applications requiring fast rule retrieval across varying ruleset densities and sizes.

Based on this analysis, a minimum Support threshold of 0.005 and a Confidence threshold of 0.1 were selected for a more detailed evaluation. These values provided a balance between having a manageable number of rules and sufficient Confidence for

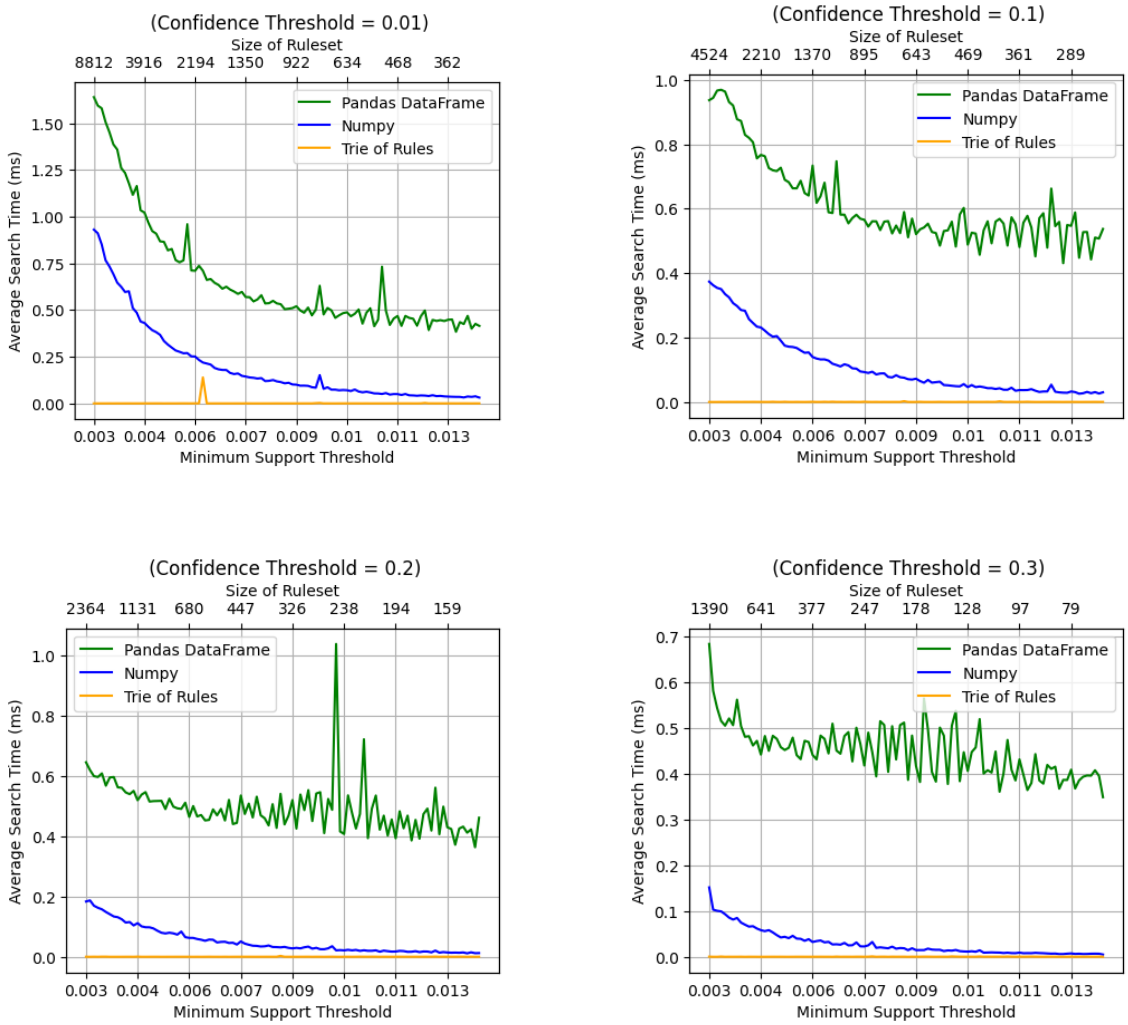


Figure 3.5: Comparison of average search time for rules in different data structures across varying Support and Confidence thresholds (the thresholds refer to those used during the rule generation phase to produce the initial ruleset from the source data). The Trie of Rules demonstrates superior performance, particularly for larger datasets.

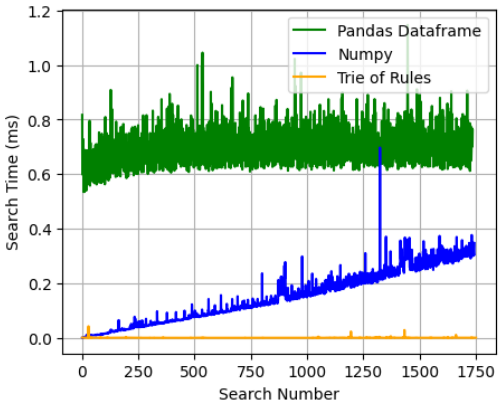
meaningful insights. The specific values allowed for generating a comprehensive set of rules while avoiding the extreme cases of very sparse or excessively dense rulesets. The ARM algorithm generated 1,001 frequent sequences and 1,752 association rules.

Figure 3.6a demonstrates the results of searching all rules in each data structure, with a summary of the results provided in Table 3.2. The average search time for the Trie of Rules was 1.63 microseconds, significantly lower than that of Pandas DataFrame, which was 770 microseconds, and NumPy, which was 183 microseconds. The total time taken to search all rules was 0.00287 seconds for the Trie of Rules,

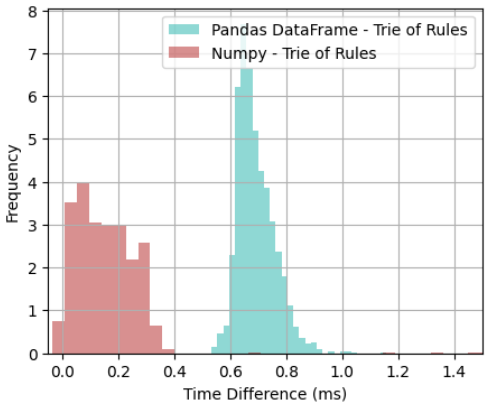
Table 3.2: Summary of Evaluation on Grocery Dataset
(1,752 rules, minimum Support = 0.005, minimum Confidence = 0.1)

	Trie of Rules	**Pandas DataFrame**	**NumPy**
Total traversal time (sec)	0.00287	1.361	0.320
Average search time (μs)	1.63	770	183
Size (megabytes)	0.917	1.996	0.0564

1.361 seconds for Pandas DataFrame, and 0.320 seconds for NumPy.



(a) Comparison of search time of rules in a ruleset between Trie of Rules, Pandas DataFrame, and NumPy data structures.



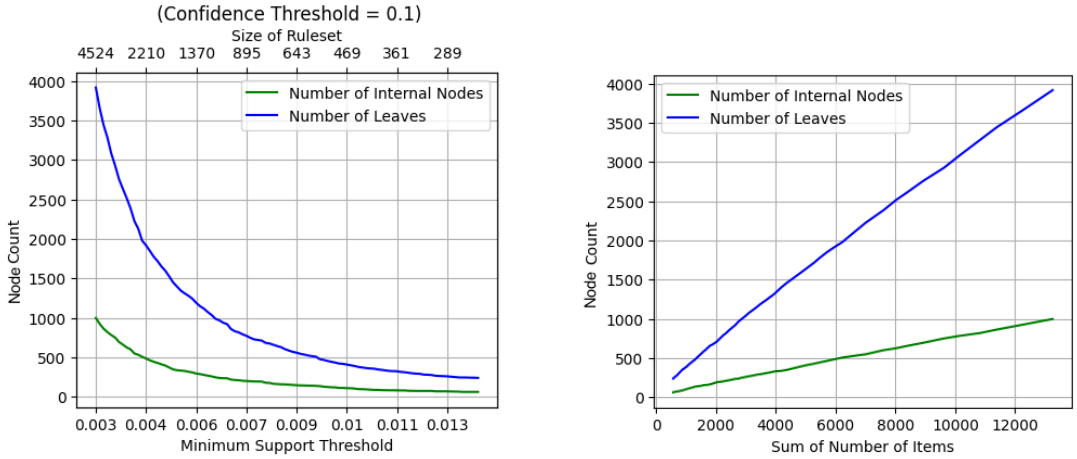
(b) Distribution of differences between search time in Trie of Rules and Pandas DataFrame and NumPy.

Figure 3.6: Search time analysis and comparative statistics for Trie of Rules, Pandas DataFrame, and NumPy.

Pairwise t-tests confirmed the significance of these differences (Fig. 3.6b). The t-test comparing the Trie of Rules and Pandas DataFrame yielded a t-statistic of $t = -14.5$, degrees of freedom $df = 1751$, and a p-value $p < 0.001$. Similarly, the comparison with NumPy showed $t = -12.07$, $df = 1751$, and $p < 0.001$. These results strongly suggest rejecting the null hypothesis that the mean search times of the Trie of Rules and the alternative data structures are equivalent, confirming the efficiency of the proposed approach.

The Trie of Rules utilizes a compact representation. The number of nodes is always approximately equal to the number of rules as indicated by Figure 3.7a. The number of internal nodes and leaves grows linearly with the total number of items in all rules, as shown in Figure 3.7b. The linear growth of internal nodes and leaves

relative to the total number of items in all rules highlights the scalability of the Trie of Rules. This property ensures that even as the size of the ruleset increases, the data structure maintains predictable and manageable growth, making it suitable for large-scale datasets. This efficiency in node representation contributes to reduced memory usage and faster traversal times.



(a) Comparison of the number of nodes with the size of the ruleset.

(b) Comparison of the number of nodes with the total number of items in rules.

Figure 3.7: Analysis of node count relative to ruleset size and total item count. The figure highlights the linear dependency of node growth on ruleset size, emphasizing the scalability of the Trie of Rules.

3.3.4 Experiment on Retail Dataset

To further evaluate the scalability of the proposed data structure, a larger dataset was used. The Retail dataset (D. Chen 2015) contains information related to online retail customers, capturing their transactional behavior over a given period. It consists of approximately 18,000 transactions and 3,600 different items. To produce a larger set of rules, suitable for thoroughly examining the effectiveness and efficiency of the data structures, a minimum Support threshold of 0.002 and a minimum Confidence threshold of 0.2 were chosen, resulting in 45,362 frequent sequences and 381,912 association rules. The summary of the evaluation is presented in Table 3.3.

Traversing through all rules in the Trie of Rules took 9.4 seconds with an average time of 24 microseconds. For NumPy, the total time was 5 hours with an average time of 48.9 milliseconds, while for Pandas DataFrame, it took more than 7 hours

Table 3.3: Summary of evaluation on Retail Dataset
(381,912 rules, minimum Support = 0.02, minimum Confidence = 0.2)

	Trie of Rules	**Pandas DataFrame**	**NumPy**
Total traversal time (sec)	9.4	27,766	18,675
Average search time (ms)	0.024	72	48.9
Size (megabytes)	188	452	12

to traverse through the whole dataset with an average time of 72 milliseconds.

The Trie of Rules boasted impressive space efficiency, as evidenced by its compact representation. The tree has a height of 9, comprising 248,864 leaf nodes and 141,053 internal nodes, occupying 188 megabytes of memory. Meanwhile, NumPy and DataFrame both contained 381,912 rows, with NumPy consuming only 12 megabytes of memory and DataFrame 452 megabytes. These results underscore the Trie of Rules' capacity for space optimization, further emphasizing its suitability for handling large-scale datasets.

On this larger dataset, the time to create a Trie of Rules was also measured. As shown in Figure 3.8, the creation time is linearly dependent on the size of the ruleset. The plot indicates that the slope is approximately 0.4 seconds per 100,000 rules. This linear relationship suggests that the Trie of Rules scales efficiently with increasing ruleset sizes, maintaining a predictable and manageable increase in creation time as the number of rules grows.

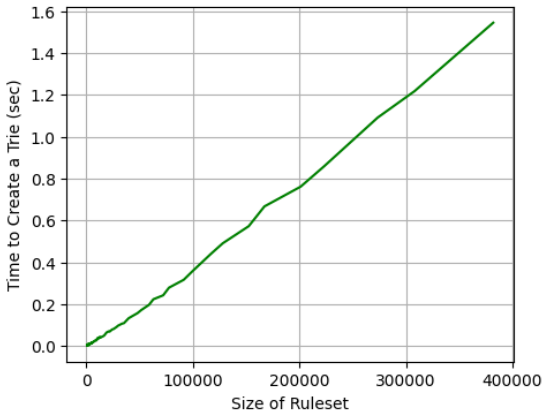


Figure 3.8: Time required to create a Trie of Rules (in seconds) as a function of the ruleset size. The creation time grows linearly with the size of the ruleset, demonstrating the efficiency of the proposed method.

In conclusion, the Trie of Rules outperforms both Pandas DataFrame and NumPy in search time and outperforms Pandas in space efficiency across different datasets. This faster search time can be attributed to the Trie of Rules naturally clustering rules, narrowing the search space with each additional item as it traverses deeper into the tree. Its structural efficiency and fast traversal make it a promising candidate for efficiently handling large-scale datasets. Moreover, its unique properties, such as a breadth-first search strategy and optimized space usage, contribute to its effectiveness in knowledge extraction tasks.

3.3.5 Discussion on Neo4J and Trie of Rules

Neo4J is a graph database management system optimized for managing relationships and patterns through graph-based storage (Neo4j 2012). In contrast to Neo4J, which functions as a comprehensive database system with diverse capabilities, the Trie of Rules represents a methodology specifically designed to enhance the storage and retrieval of association rules within the context of ARM. This methodology offers a lightweight and specialized approach that can be integrated into existing tools and libraries, such as Pandas and NumPy, to evaluate and benchmark its efficiency.

Although the Trie of Rules was not implemented within Neo4J for this research, such an implementation could potentially exploit Neo4J's graph-based architecture, enabling enhanced scalability and support for more complex queries. Demonstrating the performance of the Trie of Rules in a Neo4J environment is a promising avenue for future exploration. This study focused on assessing the methodology within a consistent experimental framework using widely adopted tools to ensure the comparability of results. The aim was to isolate and emphasize the inherent advantages of the Trie of Rules as a data structure for ARM.

3.4 Summary

Efficient data structures are fundamental for scaling Association Rule Mining to larger datasets and facilitating advanced analysis. This chapter introduced the "Trie of Rules," a novel data structure specifically designed to address the challenges of rule storage, retrieval, and traversal in ARM. By leveraging a prefix-tree graph structure, the Trie of Rules offers a scalable and systematic approach to managing complex rulesets, addressing the limitations of traditional storage methods such as plain tables or hash-table-supported data frames.

The Trie of Rules organizes rules into a prefix-tree graph structure, enabling faster traversal through the ruleset and reducing the time complexity of knowledge discovery methods. Comparative analysis with existing methods demonstrated that the Trie of Rules provides superior efficiency in terms of traversal time, particularly for large and complex rulesets. This innovative data structure serves as a valuable tool for knowledge extraction, with significant implications for enhancing rule exploration, post-mining processes, and visualization techniques in ARM.

The practical application of the Trie of Rules for real-world substitution mining and visualization is demonstrated in Chapter 6, where the effectiveness of the structure is evaluated in an integrated case study.

Chapter 4

Substitute Item Mining

4.1 Introduction

Association Rule Mining has been widely used to discover interesting patterns and relationships among items in large datasets. Traditional ARM techniques focus on identifying items that frequently occur together, providing valuable insights for applications such as market basket analysis, recommender systems, and inventory management. However, these techniques often overlook the identification of *substitute items* – items that can replace each other in fulfilling similar needs or functions.

Substitution item mining is crucial in domains such as retail, e-commerce, and supply chain management. Understanding substitution patterns enables businesses to optimize inventory levels, improve product recommendations, and enhance customer satisfaction by offering suitable alternatives when preferred items are unavailable. It also aids in competitive analysis by revealing how products compete within the same market segment.

Existing methods for substitution mining frequently rely on extensive metadata (Sethi *et al.* 2018), domain-specific knowledge (Pellegrini *et al.* 2021), or complex modeling approaches that may not be practical for all datasets (Ruiz *et al.* 2020). Moreover, a substantial body of research infers substitutability based on global similarity measures, such as lift, confidence, Jaccard similarity (Sarwar *et al.* 2001a), collaborative filtering (Sarwar *et al.* 2001b; Koren *et al.* 2009), or la-

tent space embeddings (Grbovic *et al.* 2015; Vasile *et al.* 2016). These techniques typically assess substitution relationships independently of the specific transaction context, treating substitutability as a global property of items across the entire dataset.

While these context-free approaches have proven effective in various recommendation and retrieval tasks, they often fail to capture the situational nuances that arise when the presence of particular items influences the substitutability of others. In contrast, the methodology proposed in this thesis introduces a context-sensitive framework for substitution mining, where substitutability is evaluated relative to specific antecedent conditions derived from association rules. By incorporating context into the analysis, this approach enables a more behaviorally grounded and flexible identification of substitution patterns, without requiring additional metadata or domain expertise.

To address the limitations of existing methodologies, this chapter presents a substitution mining method built upon the structure and traversal capabilities of the Trie of Rules developed in Chapter 3. The proposed approach integrates contextual analysis, similarity measures, and co-appearance patterns, enabling efficient computation of exclusion-based substitution relationships directly from transactional data. Details on how the Trie of Rules facilitates this process and a practical application of the methodology is demonstrated through a real-world case study in Chapter 6.

4.2 Methodology

To effectively quantify item substitution, we introduce a novel approach by defining substitution through three key parameters: context, similarity, and co-appearance. The relationship is captured by the following equation 4.1:

$$\text{Substitution} = f(\text{context}, \text{similarity}, \text{co-appearance}) \tag{4.1}$$

where f represents a function that integrates these three parameters to quantify the likelihood that one item is a substitute for another.

The **context** (C) refers to the environment or scenario where two potentially substitutable groups of items are compared. It represents the specific conditions under which substitution dynamics are observed. In this methodology, the context is expressed as the *antecedent* in association rules. That is, C consists of items that commonly co-occur with the items being compared, providing a situational framework for analyzing their potential to act as substitute items.

Dispersion measures the variability or spread between values and is often used to quantify how much a set of numbers differs from one another (Newbold *et al.* 2013). In this work, dispersion is used to assess the difference in Support values of items within a given context, providing a foundation for evaluating their similarity.

The **similarity** (Sim) between items within a given context is calculated using the dispersion of their Support values. Dispersion between two values, x and y, is defined as:

$$\text{Dispersion}(x, y) = \frac{|x - y|}{x + y} \tag{4.2}$$

This formula, as outlined in Newbold *et al.* (2013), provides a normalized measure of the difference between two values, ranging from 0 (no difference) to 1 (maximum difference).

Building on this concept, the notion of similarity between items A and B within a specific context C is introduced. Applying the dispersion measure to the Support values of A and B within the context C, the similarity is defined as:

$$\text{Similarity}(C, A, B) = 1 - \text{Dispersion}\left(\sup(C \to A \,|\, \neg B), \sup(C \to B \,|\, \neg A)\right) \tag{4.3}$$

where:

- A and B represent the sets of items being compared.
- $\sup(C \to A \,|\, \neg B)$ is the Support value for the rule $C \to A$, excluding B.

- $\sup(C \to B \,|\, \neg A)$ is the Support value for the rule $C \to B$, excluding A.
- The exclusion condition can be formally defined as:

$$\sup(C \to A \,|\, \neg B) = \sup(C \to A) - \sup(C \to AB) \tag{4.4}$$

This expression reflects the number of transactions where C and A co-occur without B. In the context of substitution mining, it captures the weakening of the $C \to A$ association when B is present, and thus contributes to identifying potential substitutes or competing relationships.

Expanding the similarity equation gives:

$$\text{Similarity}(C, A, B) = 1 - \frac{|\sup(C \to A \,|\, \neg B) - \sup(C \to B \,|\, \neg A)|}{\sup(C \to A \,|\, \neg B) + \sup(C \to B \,|\, \neg A)} \tag{4.5}$$

This formulation expresses similarity as one minus the normalized difference (dispersion) between the Support values for A and B, conditioned on their shared context C. A similarity value close to 1 indicates high alignment between the Support values of A and B, suggesting that they are strong substitutes within the given context. Conversely, a lower similarity value indicates a reduced substitution potential.

To calculate similarity, this methodology uses Support values for each item within the given context while excluding the co-occurring item. Support is chosen as it represents the frequency or prevalence of certain items within transactions, making it an essential measure in Association Rule Mining. High Support values indicate that an item or a set of items frequently appears within the dataset, which is crucial for identifying potential substitutes.

In this context, excluding B when calculating $\sup(C \to A \,|\, \neg B)$ (and vice versa) serves to isolate the individual presence of each item in the context without the influence of the other. This separation has two primary benefits:

1. **Isolating Individual Contribution:** Excluding B when measuring Support

for A (and vice versa) allows the distinct presence of each item within the context to be captured independently. This allows us to evaluate whether each item can effectively stand alone in that context, which is essential for assessing substitutability. If both items display high Support values independently of each other, it suggests they could fulfill similar roles within that context, supporting the idea that they are substitutes.

2. **Avoiding Confounding Effects of Co-Occurrence:** Including both A and B when calculating Support might make it difficult to distinguish whether their Support values are due to individual popularity or their tendency to appear together. Excluding B when examining A's Support ensures that A's prevalence is not influenced by associations with B, providing a clearer measure of its independent presence. This helps prevent a confounding effect where frequent co-occurrence could otherwise artificially inflate similarity, leading to a potential misinterpretation of association as substitutability.

Using Support and excluding the other item when calculating similarity therefore isolates each item's behavior, offering a clearer view of how independently prevalent each is within the context. This approach aligns with the objective of identifying true substitutes by emphasizing individual behavior rather than joint patterns, leading to a more reliable similarity measure that accurately reflects potential substitutability.

The similarity value ranges between 0 and 1, where a value close to 1 indicates high similarity (i.e., the items behave similarly within the given context), and a value close to 0 indicates low similarity.

Similarity highlights how similar the behavior of two potential substitutes is within a given context. If both items have similar Support within that context, they are considered similar. Conversely, if one item appears frequently while the other is significantly less frequent, it suggests differing behavior of those items in that context.

The **co-appearance** (Coapp) of items measures how often two sets of items, A and B, appear together within the context C. This concept is rooted in joint

probability and association rule mining principles (Agrawal, Imieliński, *et al.* 1993), where joint probability quantifies the likelihood of two events occurring simultaneously. Co-appearance, in the context of itemsets, captures whether items tend to be used or purchased together frequently or rarely, giving insights into their associative strength within a specific context.

In probabilistic terms, the joint probability $P(A \cap B)$ represents the probability that both events A and B occur simultaneously. Similarly, co-appearance in association rule mining quantifies how often two itemsets co-occur relative to their individual occurrences within a specified context.

Co-appearance can be expressed as follows:

$$\text{Co-appearance}(A, B) = \frac{P(A \cap B)}{P(A) + P(B) - P(A \cap B)} \tag{4.6}$$

This ratio compares the joint occurrence $P(A \cap B)$ of A and B to the total occurrences of A and B, adjusting for overlap to prevent double-counting. In association rule mining, this can be interpreted as the frequency with which A and B co-occur relative to their total occurrences.

Adapting this concept to the co-appearance of itemsets A and B within a specific context C, the co-appearance is calculated as:

$$\text{Co-appearance}(C, A, B) = \frac{\sup(C \to AB)}{\sup(C \to A \,|\, \neg B) + \sup(C \to B \,|\, \neg A) + \sup(C \to AB)} \tag{4.7}$$

where:

- $\sup(C \to AB)$ is the support of items A and B appearing together within the context C,
- $\sup(C \to A \,|\, \neg B)$ is the support of A in the context of C, excluding B, and is formally defined as:

$$\sup(C \to A \mid \neg B) = \sup(C \to A) - \sup(C \to AB) \tag{4.8}$$

- $\sup(C \to B \,|\, \neg A)$ is the support of B in the context of C, excluding A, and is formally defined as:

$$\sup(C \to B \mid \neg A) = \sup(C \to B) - \sup(C \to AB) \tag{4.9}$$

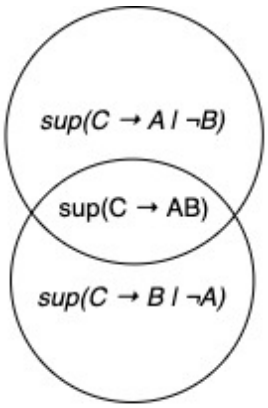


Figure 4.1: Venn diagram for co-appearance calculation.

Including $\sup(C \to AB)$ in the denominator ensures that co-appearance reflects the proportion of times A and B appear together out of all possible occurrences in the context. This formulation provides a balanced measure of co-occurrence, emphasizing that frequent co-occurrence is necessary to achieve a high co-appearance score, even if individual occurrences are high.

The co-appearance value ranges between 0 and 1:

- A value close to 1 indicates that the items frequently appear together within the context.

- A value close to 0 indicates that they rarely co-occur within the context.

Calculating Substitution

Previous studies have explored substitution using diverse methods, including contextualized embedding vectors in natural language processing (NLP) (Pellegrini *et al.* 2021) and distributional similarity based on the cosine similarity of context

vectors (Achanuparp *et al.* 2016). These approaches, while effective in identifying substitution patterns to some extent, often focus narrowly on co-consumption, potentially overlooking nuances such as variations in item relevance across different contexts or the influence of indirect relationships between items. Additionally, many methods rely heavily on domain-specific knowledge or require extensive metadata, limiting their applicability to broader or less-structured datasets.

To address these challenges, the proposed methodology expands the analysis by incorporating both the similarity of items within a given context and their co-appearance patterns. This dual focus enables a more robust assessment of substitution relationships, capturing not only the direct compatibility of items but also their behavior across diverse transactional scenarios. By leveraging intrinsic data patterns as a *context* rather than external metadata or pre-trained models, the approach ensures applicability across various domains and datasets.

The substitution score can be calculated as follows:

$$\text{Substitution} = \text{Similarity} \times (1 - \text{Co-appearance}) \tag{4.10}$$

This equation is based on principles of multi-criteria evaluation discussed by Belton *et al.* (2002), where each parameter (similarity and co-appearance) serves as a distinct criterion that influences the overall score. Specifically, this approach aligns with probabilistic adjustment methods and penalization schemes commonly used in recommendation systems (Ricci *et al.* 2011). The reasons behind this approach can be listed as follows:

1. **Independent Criteria Adjustment:** In multi-criteria evaluation methods, criteria are often combined multiplicatively to ensure that each factor independently contributes to the final score. By multiplying similarity with $(1 - \text{Co-appearance})$, the calculation reflects how closely items behave (similarity) and adjusts for how often they co-occur (co-appearance) without assuming that the two factors are directly proportional. This approach allows

similarity and co-appearance to influence the score independently: high similarity increases the score, while high co-appearance reduces it. This interaction models real-world substitutability, where items that behave similarly but rarely co-occur are more likely to be substitutes.

2. **Penalization for Co-occurrence:** The term $(1 - \text{Co-appearance})$ functions as a penalty for high co-appearance. Penalization is a common technique in probabilistic frameworks and recommendation algorithms (Ricci *et al.* 2011). By subtracting co-appearance from 1, the formula reduces the substitution score for items that frequently appear together, thereby filtering out complementary items that do not serve as substitutes.

In essence, this formula effectively uses a penalty-based approach to adjust similarity for contextual co-occurrence, ensuring that the resulting substitution score represents true substitutability rather than a simple association.

This formulation ensures that:

- If two items are similar but rarely appear together (high similarity, low co-appearance), the substitution value is high, indicating they are good substitutes.
- If two items are similar and frequently appear together (high similarity, high co-appearance), the substitution value is low, suggesting they are more likely complements or associated items.
- If two items are dissimilar, the substitution value is low regardless of co-appearance, as they do not behave similarly in the context.

The substitution value ranges between 0 and 1, where a higher value indicates a higher likelihood of substitution.

4.2.1 2D Space Representation

To visualize the substitution dynamics, a 2D space is used with one axis representing similarity and the other representing co-appearance (Figure 4.2). This representa-

tion provides deeper insights into the relationships between items, allowing for the identification of different categories based on their positions in this space.

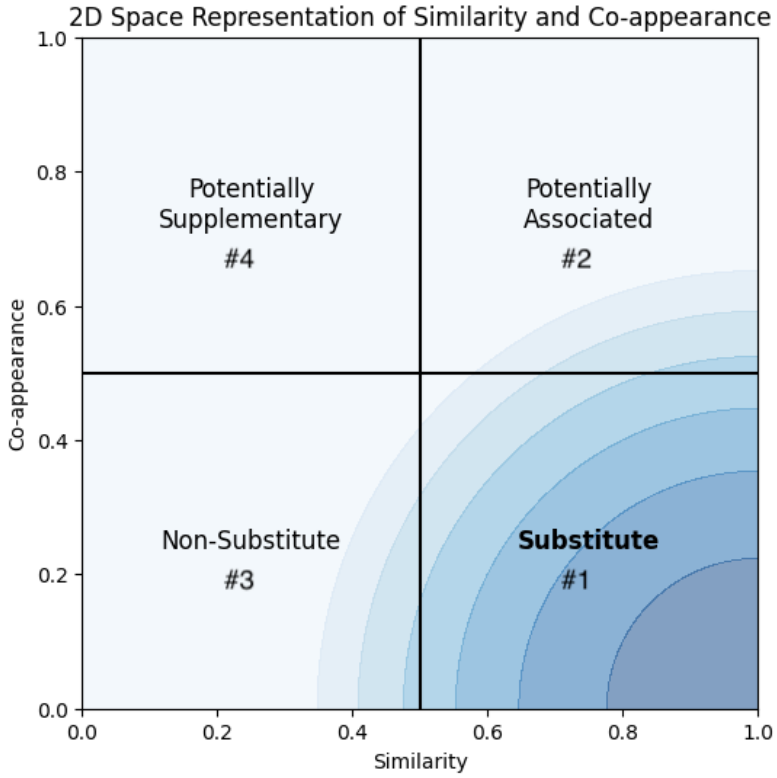


Figure 4.2: 2D Space representation of similarity and co-appearance. The gradient in the lower right quadrant represents the strength of substitution, with darker shades indicating a higher likelihood that the items are substitutes. This visual demonstrates how items with different combinations of similarity and co-appearance values fall into various categories.

The four quadrants represent:

1. **#1 – High similarity and low co-appearance (Bottom Right Quadrant):** Items here are likely substitutes. They behave similarly within the context but rarely appear together, indicating that consumers may use them interchangeably.

2. **#2 – High similarity and high co-appearance (Top Right Quadrant):** Items in this quadrant are associated and may form strong association rules. They frequently appear together and have similar frequencies, suggesting complementary or related usage. For example, bread and butter are often purchased together.

3. **#3 – Low similarity and low co-appearance (Bottom Left Quadrant):** Items here are neither similar nor frequently appear together, indicating little to no relationship.

4. **#4 – Low similarity and high co-appearance (Top Left Quadrant):** Items in this quadrant might be supplementary. One item may be less frequent but often purchased with a more popular item, serving as an accessory or enhancement. For instance, a laptop and a laptop cooling pad.

4.3 Evaluation

Evaluating substitution methodologies presents unique challenges, primarily due to the lack of ground truth data that accurately reflects consumer choices and context. To address this, different approaches are employed depending on the evaluation goal. Subjective methods, such as expert opinions (Amit Pande *et al.* 2022; Tkachuk *et al.* 2022), provide qualitative insights but are inherently limited by bias and lack of scalability. In contrast, quantitative metrics, such as confusion matrices or F1 scores (Peng *et al.* 2002; Pellegrini *et al.* 2021), offer an objective way to evaluate model performance by comparing predicted outcomes to observed data. Statistical tests can further validate these results by analyzing the significance of differences between predictions and observations (Draper *et al.* 1998). Surveys capturing human perspectives on substitutability (McAuley *et al.* 2015; Achanuparp *et al.* 2016; M. Zhang *et al.* 2020) provide a complementary method, bridging the gap between subjective and quantitative evaluations.

Most existing methods for substitute item mining are based on collaborative filtering, global similarity metrics, or metadata-driven approaches (Sarwar *et al.* 2001a; Koren *et al.* 2009; Grbovic *et al.* 2015), and typically do not incorporate transaction-specific context. The method proposed in this thesis differs fundamentally by identifying substitution relationships in a context-aware manner, using association rule interactions without relying on external metadata. As a result, direct

quantitative comparison to these methods would not be meaningful, since the evaluation objectives and assumptions differ substantially. Instead, the effectiveness of the proposed methodology is assessed through internal validation and alignment with human judgment, focusing on whether the discovered patterns are interpretable and consistent with real-world substitutability perceptions.

To address this challenge, the proposed substitution mining methodology was applied to real-world retail data to generate predicted substitution values for various item pairs. Subsequently, a survey was conducted among a diverse group of participants to capture their perceptions of substitutability for the same item pairs. By comparing the model's predictions with the participants' responses, it is possible to assess how well the methodology aligns with human intuition and real-world substitutability perceptions.

4.3.1 Data Collection and Survey Design

The evaluation utilized the Groceries dataset (Hahsler, Hornik, *et al.* 2006), a widely used dataset in Association Rule Mining research. This dataset contains transactional data from a retail grocery store, providing a suitable basis for deriving substitution patterns.

From the generated substitution values using the proposed methodology, a selection of item pairs was made to include both high and low predicted substitution scores. This ensured that the evaluation would cover a range of substitutability scenarios.

An online survey was designed and approved by the Dublin City University Research Ethics Committee. Participants were recruited through various channels, including surveyswap.io and student communities at Dublin City University, to ensure diversity in the sample.

Participants were provided with a clear definition of substitution, along with examples, to ensure consistent understanding. The survey consisted of 23 questions, each presenting an item pair within a specific context, formatted as:

*"In the context of item(s) **C**, how substitutable is item **A** for item **B**?"*

A total of 31 participants were asked to rate the substitutability on a Likert scale from 1 (*Not substitutable at all*) to 5 (*Perfectly substitutable*).

The survey data were organized in a matrix format, with participants as rows and questions as columns, as shown in Table 4.1. The data elements are described as follows:

- r_i: Represents respondent i in the set of respondents R.
- a_{ij}: Denotes the answer by respondent i to question j, with all answers in the range $[1, 5]$.
- q_j: Refers to question j in the set of questions Q.
- p_j: Indicates the unscaled predicted value for question j, in the range $[0, 1]$, generated by the substitution model.
- $\tilde{p}_j$: Represents the scaled predicted value for question j, transformed to match the survey's $[1, 5]$ rating scale using the formula:

$$\tilde{p}_j = 1 + 4 \times p_j \tag{4.11}$$

Here, $\tilde{p}_j$ allows for a direct comparison with the survey responses.

Table 4.1: Organization of survey responses and predicted substitution values

Participant (i)	**Question 1**	**Question 2**	...	**Question** j
1	a_{11}	a_{12}	...	$a_{1,j}$
2	a_{21}	a_{22}	...	$a_{2,j}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
i	$a_{i,1}$	$a_{i,2}$	...	$a_{i,j}$
Predicted (unscaled)	p_1	p_2	...	p_j
Predicted (scaled)	$\tilde{p}_1$	$\tilde{p}_2$	...	$\tilde{p}_j$

4.3.2 Statistical Analysis Methods

To evaluate the alignment between the model's predicted substitution values and participants' responses, several statistical methods were employed:

- **Mixed-Effects Models**: Mixed-effects models are well-suited for handling hierarchical data structures, as they account for variability appearing from differences between participants and questions, enabling more precise inference (Bates 2007; Demidenko 2013).

- **Logistic Regression**: Applied to assess the ability of the model to classify item pairs as substitutes or non-substitutes based on predicted substitution values. This method evaluates the predictive power of the model in a binary classification context.

4.3.3 Mixed-Effects Models

Mixed-effects models are statistical techniques used to analyze data that involve both fixed effects and random effects, particularly when the data have a hierarchical or grouped structure (Demidenko 2013). They are especially suitable for datasets where observations are grouped at more than one level, such as participants and questions in this study. Mixed-effects models allow for the modeling of the influence of predictor variables that are consistent across all observations (fixed effects) while accounting for variability introduced by grouping factors (random effects).

A **fixed effect** represents the average effect of a predictor variable that is assumed to be the same across all levels of the grouping factors. In this analysis, the fixed effect is the predicted substitution value, p_j, which is consistent across all participants for a given question.

A **random effect** captures the variability in the response variable that is associated with the specific levels of the grouping factors, such as individual differences among participants or unique characteristics of questions. Random effects are assumed to be randomly sampled from a population, allowing the model to account for

correlations within groups and variability that is not explained by the fixed effects.

In this analysis, the participant ratings, a_{ij}, are modeled as a function of the predicted substitution values, p_j, with random effects for participants and questions:

$$a_{ij} = \beta_0 + \beta_1 p_j + u_i + v_j + \epsilon_{ij} \tag{4.12}$$

where:

- a_{ij}: Rating given by participant i to question j.
- p_j: Predicted substitution value for question j.
- β_0: Intercept term representing the overall average rating when $p_j = 0$.
- β_1: Fixed-effect coefficient representing the average effect of the predicted substitution value on participant ratings.
- $u_i \sim N(0, \sigma_u^2)$: Random effect for participant i, capturing individual deviations from the overall mean rating.
- $v_j \sim N(0, \sigma_v^2)$: Random effect for question j, capturing question-specific deviations from the overall mean rating.
- $\epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$: Residual error term representing unexplained variability not accounted for by the fixed or random effects.

This model effectively captures the hierarchical structure of the data, where ratings are nested within participants and questions. The fixed effect β_1 allows for the assessment of the overall relationship between the predicted substitution values and participant ratings across all participants and questions.

This approach acknowledges that observations are not independent but may be correlated within participants or questions, and it adjusts for this in the estimation process. It quantifies how well the model's predictions align with human judgments and identifies the extent to which variability in ratings is due to differences among participants, questions, or unexplained factors.

Model Results The fixed-effect coefficient for the predicted substitution value was found to be statistically significant:

- Fixed effect (β_1): 1.25, $SE = 0.5$, $t = 2.26$, $p < 0.05$.
- Intercept estimate (β_0): 2.2, $SE = 0.36$, $t = 6.05$, $p < 0.05$.

This indicates a strong positive relationship between the model's predicted substitution values and the participants' ratings, suggesting that **the model effectively captures human perceptions of substitutability**.

The variance components for the random effects were estimated as:

- σ_u^2 (Participants) = 0.1
- σ_v^2 (Questions) = 0.78
- σ_ϵ^2 (Residual) = 1.27

The Intraclass Correlation Coefficient (ICC) was calculated to assess the proportion of variance attributable to the random effects. The ICC provides a measure of how much of the total variability in the data can be explained by the grouping structure (Shrout *et al.* 1979) – in this case, differences between participants and differences between questions.

The ICC is calculated by dividing the variance component of interest by the total variance, which includes all variance components (random effects and residual error). This gives the proportion of the total variance that is attributable to that specific random effect.

For participants, the ICC is calculated as:

$$\text{ICCParticipants} = \frac{\sigma_u^2}{\sigma_u^2 + \sigma_v^2 + \sigma\epsilon^2} = \frac{0.1}{0.1 + 0.78 + 1.27} = \frac{0.1}{2.15} \approx 0.049 \tag{4.13}$$

For questions, the ICC is:

$$\text{ICCQuestions} = \frac{\sigma_v^2}{\sigma_u^2 + \sigma_v^2 + \sigma\epsilon^2} = \frac{0.78}{0.1 + 0.78 + 1.27} = \frac{0.78}{2.15} \approx 0.363 \tag{4.14}$$

These ICC values indicate that approximately 5% of the variance is due to differences between participants, and 36% is due to differences between questions. The residual variance accounts for the remaining proportion:

$$\text{Residual Variance Proportion} = \frac{\sigma_\epsilon^2}{\sigma_u^2 + \sigma_v^2 + \sigma_\epsilon^2} = \frac{1.27}{2.15} \approx 0.59 \tag{4.15}$$

This means that around 59% of the total variance is attributed to residual error, representing unexplained variability at the individual response level.

An ICC of 5% for participants suggests that individual differences among participants contribute only minimally to the variability in ratings. This indicates that the model performs consistently across participants and is not strongly influenced by specific individual biases or tendencies to rate substitutions higher or lower.

An ICC of 36% for questions highlights that about one-third of the variability can be explained by the specific item pairs and contexts presented in the questions. This suggests that some questions inherently stimulate higher or lower ratings, likely due to the nature of their content or the perceived substitutability of the items involved.

These results suggest that the model generalizes well across participants. However, the relatively high variance attributed to questions (36%) indicates the need for further investigation into how question content and context impact substitutability ratings.

Random Effects Interpretation The random effects for questions (v_j) represent how much each question's average response deviates from the overall average, after accounting for the effect of the predicted substitution value. These random effects capture question-specific variations not explained by the fixed effect of the predicted substitution values.

Interpreting the random effects for questions:

- **Positive Random Effect**: Participants rated the substitution *higher* than predicted by the model. This suggests an **underestimation** by the model for that question.

- **Negative Random Effect**: Participants rated the substitution *lower* than predicted by the model. This indicates an **overestimation** by the model.
- **Zero Random Effect**: Participants' ratings align closely with the model's predictions. This suggests the model accurately captures substitutability for that question.

To assess whether the distribution of random effects for questions deviates from zero, a statistical test was performed with the null hypothesis that the mean of the random effects is zero. The random effects analysis yielded a mean of -2.66×10^{-14}, with a standard error of 0.1763, $t = -1.51 \times 10^{-13}$, df = 22, and $p = 1.0$.

The extremely small t-statistic and a p-value of 1 indicate no evidence against the null hypothesis. Therefore, the mean of the random effects for questions is not significantly different from zero. This suggests that, on average, the model does not systematically overestimate or underestimate substitutability across questions.

A plot of the random effects for each question is shown in Figure 4.3. The plot indicates that the random effects for 12 questions are not significant (their confidence intervals include zero). The remaining questions show significant deviations, with some questions exhibiting positive random effects and others negative.

Only a few questions display random effects with absolute values greater than 1, meaning the model's predictions for these questions missed participant ratings by more than 1 point on the Likert scale. These instances highlight where the model's performance was notably less accurate. A closer examination of these questions is provided in the next section.

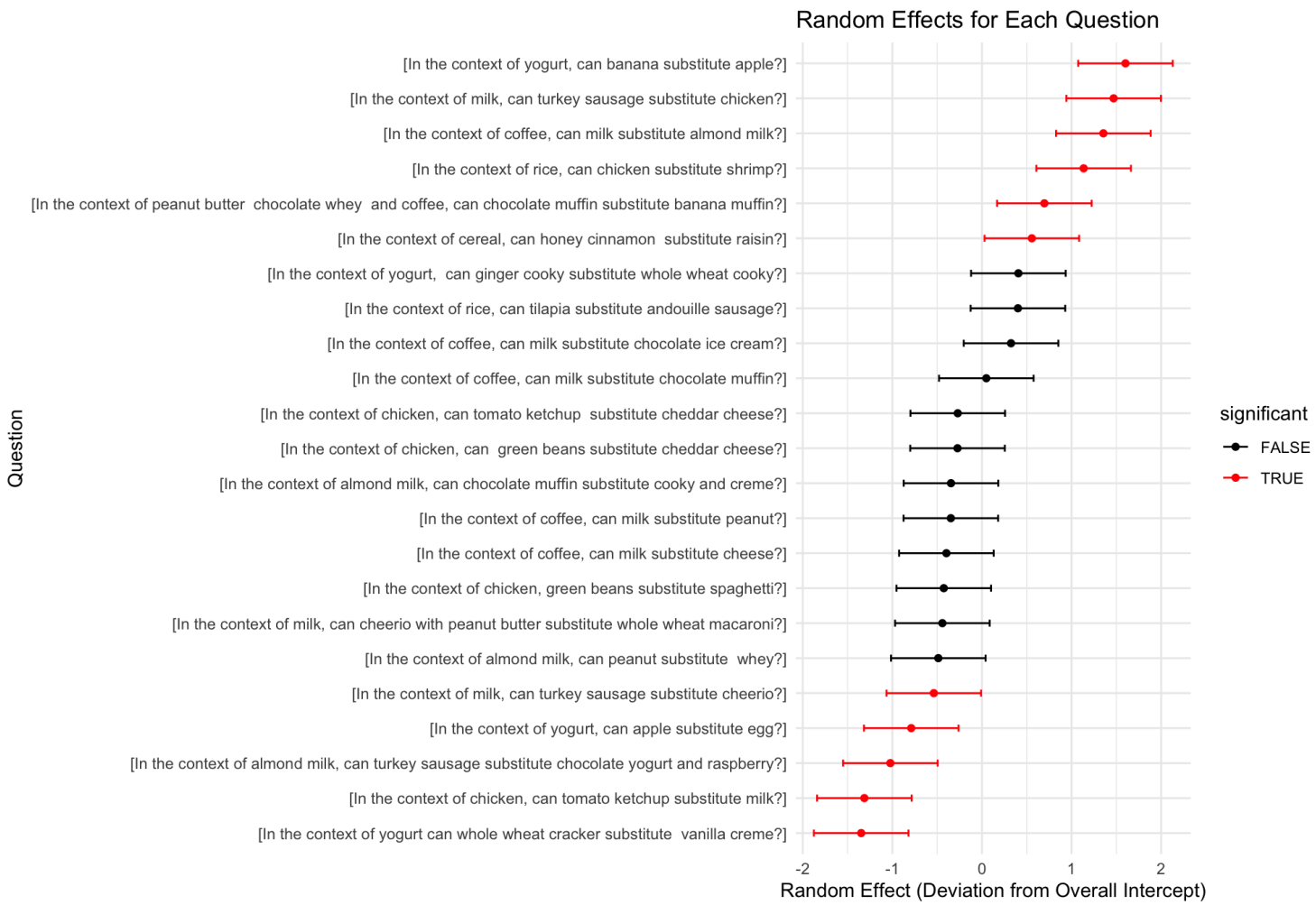


Figure 4.3: Random effects for questions. The bars represent the estimated random effect v_j for each question, with error bars indicating the 95% confidence intervals. Positive values suggest underestimation by the model, negative values indicate overestimation, and values not significantly different from zero imply accurate model predictions.

Identification of Underperforming Questions

To investigate questions where the model's predictions did not align well with participant responses, a second mixed-effects model was fitted. This model aimed to assess the differences between questions in terms of participant ratings and the predicted substitution values:

$$\text{Diff}_{ij} = a_{ij} - \tilde{p}_j = \gamma_0 + \gamma_j + u_i + \epsilon_{ij} \tag{4.16}$$

where:

- Diff_{ij}: Difference between participant i's rating and the scaled predicted substitution value for question j.
- γ_j: Fixed effect for question j, capturing systematic deviations in the differences for each question.
- $u_i \sim \mathcal{N}(0, \sigma_u^2)$: Random effect for participant i, accounting for individual-level variability.
- $\epsilon_{ij} \sim \mathcal{N}(0, \sigma_\epsilon^2)$: Residual error, representing unexplained variability.

This model evaluates how well the predicted substitution values align with participant responses by focusing on the individual questions. Specifically, the fixed effect γ_j provides insights into whether the model systematically overestimates or underestimates substitutability for specific questions.

Estimated Marginal Means: Estimated Marginal Means (EMMs) (Xue *et al.* 2010) were used to analyze the systematic effects of the model across different questions, focusing on fixed effects to identify whether the model consistently overestimates or underestimates substitutability for specific questions. Unlike the Intraclass Correlation Coefficient, which quantifies the proportion of variance attributable to random effects (differences among participants or questions), EMMs directly compare adjusted group means, revealing systematic biases in the model's predictions.

This approach allows for the identification of underperforming questions where the model fails to align with participant responses.

A plot of the Estimated Marginal Means with confidence intervals (Figure 4.4) highlights clusters of questions based on their deviation from zero. These clusters reflect how the model's predictions align with participant responses for each question. Letters assigned to each question indicate groups where questions are not significantly different from one another, while different letters represent statistically significant differences between groups.

The EMM analysis identifies distinct clusters of questions, reflecting varying levels of alignment between participant responses and model predictions.

- **Group l:** This group, positioned at the top of the plot (Figure 4.4), highlights a single question that significantly deviates from the others. Referring to both the random effects plot of questions (Figure 4.3) and the distribution of responses per question with predicted values (Figure 4.5), it becomes clear that the model underestimated the substitution value for this question. This discrepancy likely arises due to insufficient evidence in the training data to justify the hypothesis that the corresponding item pair could function as substitutes within this specific context. However, survey participants identified this substitution as valuable, potentially influenced by cultural or contextual factors absent in the training dataset. Similar behavior is observable in cluster k, but the deviations are less evident.

- **Groups a and b:** These groups, at the bottom of the plot (Figure 4.4), represent questions with significant deviations. Response distributions indicate *inconsistent participant agreement* on the substitutions. While the data suggest potential substitutes, the lack of strong consensus leads to false positives. These errors, though not ideal, are less critical in retail applications, as they highlight substitutes that may appeal to varying customer preferences or contexts.

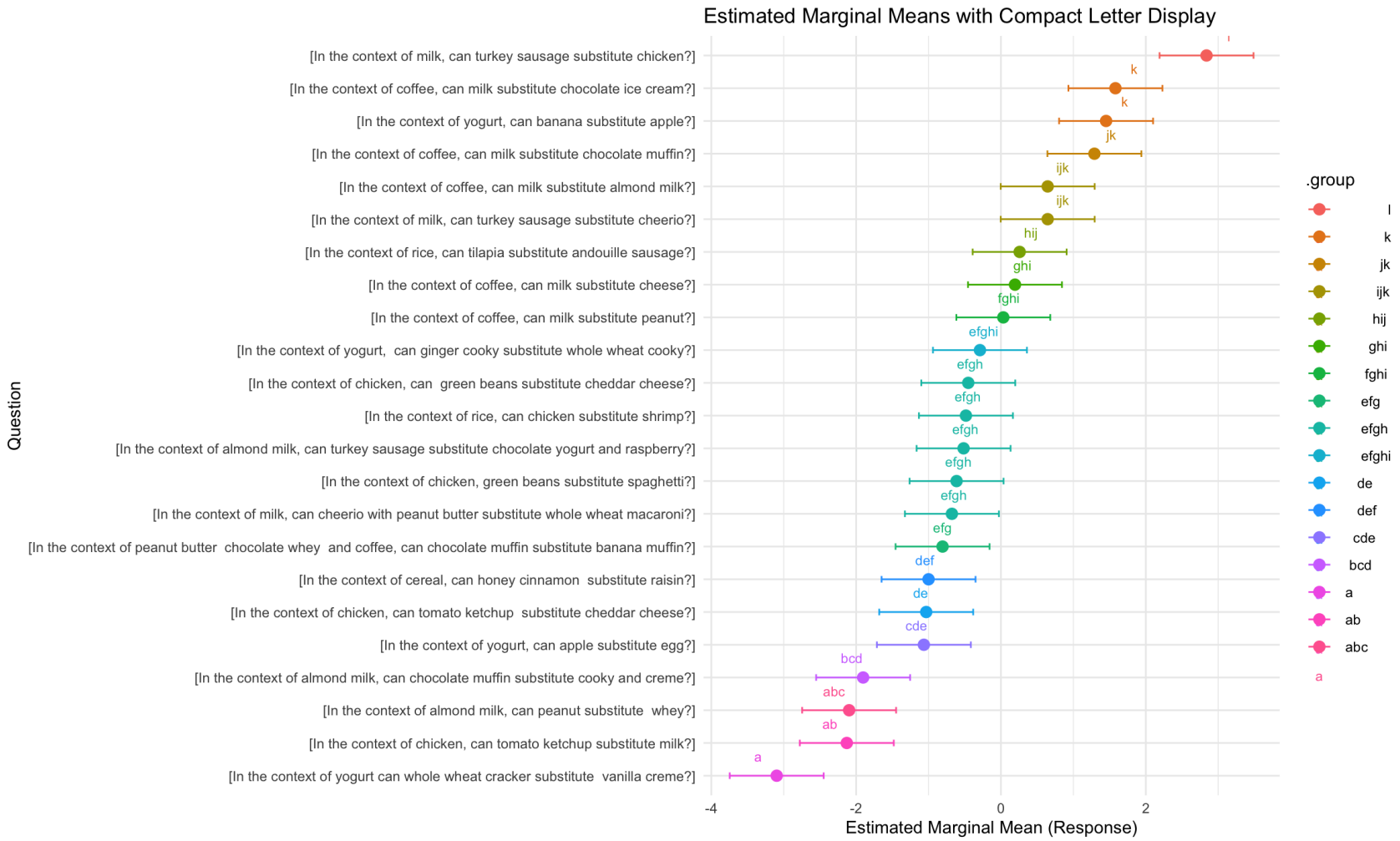


Figure 4.4: Estimated Marginal Means (EMMs) for questions. The EMMs represent the average difference between participant ratings and model predictions for each question, with error bars indicating 95% confidence intervals. Letters indicate statistical groupings, where identical letters imply no significant difference between questions.

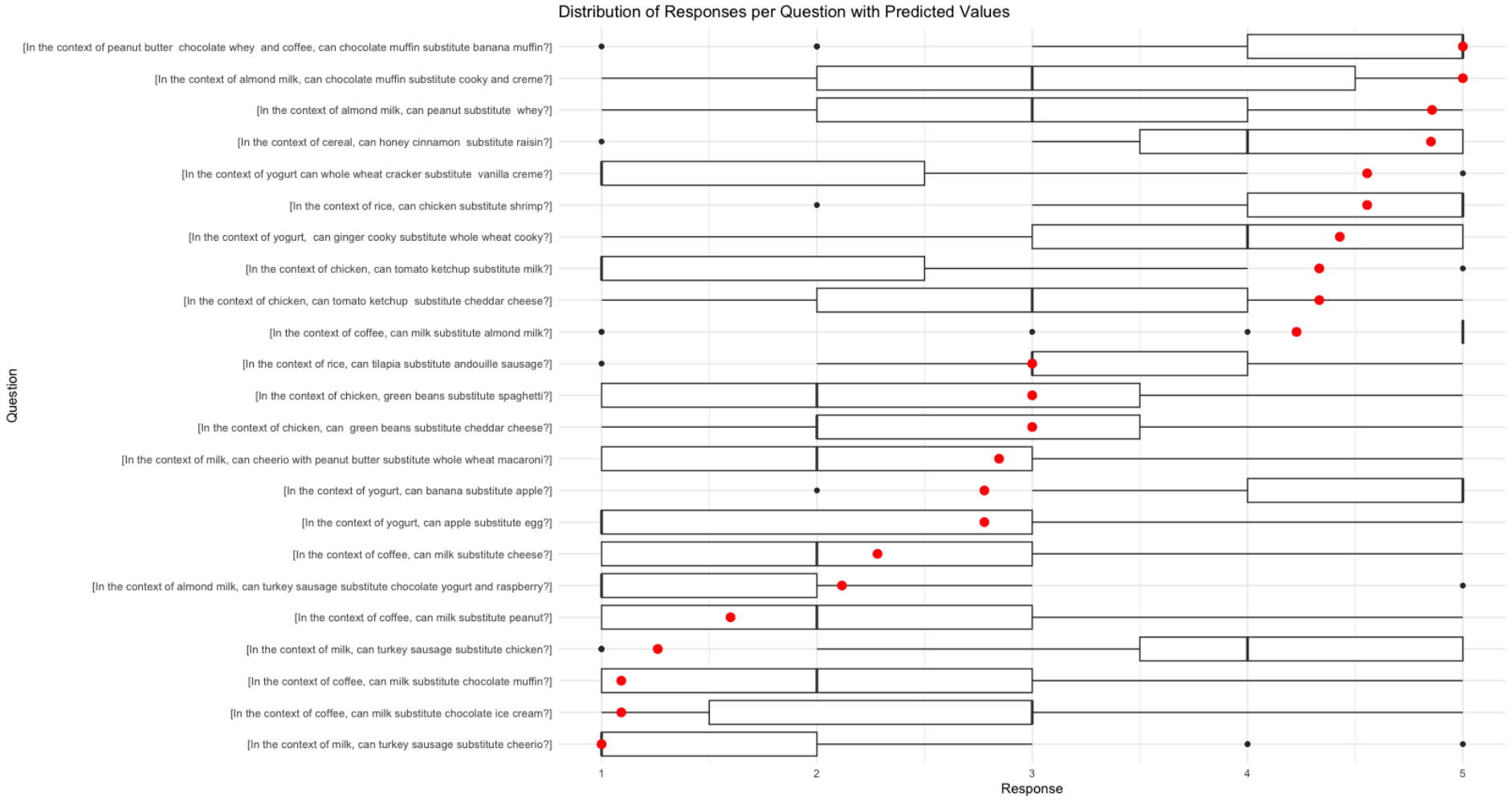


Figure 4.5: Distribution of responses per question with predicted Values. The boxplots represent the participant response distributions for each question, while the red points indicate the predicted substitution values using the proposed methodology. This visual comparison helps identify alignment or misalignment between participant ratings and model predictions.

Overall, the EMM analysis demonstrates that the majority of questions align well with the model's predictions, as most lie near the center of the plot. However, the presence of outliers in Groups l and a/b highlights areas where further refinement of the substitution model or additional data may improve alignment.

In summary, the overall performance of the substitution model is robust, with the majority of questions aligning well with participant responses. Cases of underestimation and overestimation are limited.

4.3.4 Logistic Regression Analysis

To further evaluate from a question perspective, the survey responses were averaged for each question across all users to determine a general substitutability rating for each item pair. The approach of averaging across users is justified by the fact that the variance attributed to differences between participants was found to be less than 5%, as established in the previous section. Item pairs with an average rating above 3 were classified as *true substitutes*, while those with an average rating of 3 or below were classified as *non-substitutes*. The threshold of 3 was chosen as it represents the midpoint of the scale, where respondents indicated uncertainty (i.e., a rating of 3 implied "maybe yes, maybe no" regarding substitutability). By using this midpoint, the classification effectively captures general user sentiment, aligning with the non-personalized nature of the substitute item predictions generated by the proposed model, which is based on overall behavioral patterns rather than individual preferences.

To evaluate the performance of the substitution model, a logistic regression analysis was conducted (Peng *et al.* 2002). In this model, the observed classification from survey responses (substitute/non-substitute) served as the dependent variable, while the substitution score generated by the proposed methodology served as the independent variable. The logistic regression aimed to evaluate the accuracy of the model's binary predictions and determine an optimal threshold within the model's output scale (0 to 1) for classifying items as substitutes or non-substitutes. Analysis

identified an optimal substitution threshold of 0.43, which effectively separates item pairs into substitutable and non-substitutable categories.

The logistic regression model is:

$$\log\left(\frac{P(\text{Substitute})}{1-P(\text{Substitute})}\right) = \beta_0 + \beta_1 \tilde{p}_j \tag{4.17}$$

where $P(\text{Substitute})$ is the probability that item pair j is a substitute, and $\tilde{p}_j$ is the scaled predicted substitution value.

Table 4.2: Confusion matrix of logistic regression classification

	Predicted Non-Substitute	**Predicted Substitute**
Actual Non-Substitute	5	5
Actual Substitute	2	12

Performance metrics:

- **Accuracy**: $\frac{5+12}{5+5+2+12} = \frac{17}{24} \approx 70.8\%$
- **Precision**: $\frac{12}{12+5} = \frac{12}{17} \approx 70.6\%$
- **Recall (Sensitivity)**: $\frac{12}{12+2} = \frac{12}{14} \approx 85.7\%$
- **Specificity**: $\frac{5}{5+5} = \frac{5}{10} = 50\%$
- **F1 Score**: $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \approx 77.4\%$

The model demonstrates a good balance between precision and recall, with a higher recall indicating that the model is effective at identifying true substitutes. The lower specificity suggests that the model is less effective at correctly identifying non-substitutes, leading to some false positives (Type I errors). However, in applications such as recommendation systems, false positives are generally less harmful than false negatives, as suggesting an item that is not a perfect substitute may still be acceptable to users. On the contrary, in critical fields such as healthcare, where recommending the wrong drug could have severe consequences, false positives become a significant issue. In such cases, a more nuanced adjustment of the developed model is required to account for the higher cost of errors.

4.4 Summary

Substitute item mining plays a crucial role in understanding consumer behavior and enhancing decision-making processes in retail and recommendation systems. This chapter explored the development and evaluation of a novel substitution model designed to predict substitutability between items. By leveraging association rule mining, the model offers a structured and scalable approach to identifying substitution relationships, addressing limitations in existing methodologies that rely heavily on extensive metadata or subjective assessments.

The evaluation demonstrated that the developed substitution model performs well, as evidenced by the mixed-effects analysis. The predicted substitution values significantly align with human responses, confirming the validity of the model's methodology. While variability in responses is partially attributed to differences among participants and questions, the model itself exhibits strong predictive and generalization capability.

A small subset of questions showed misalignment, particularly those where the predicted substitution value was overestimated. The data suggest that these substitutions were plausible within the given context, but participant ratings reflected differing perspectives. This leads to Type I errors (false positives), which, while not ideal, are less critical in retail applications. In fact, false positives can be beneficial in these scenarios as they suggest potential substitutes that may appeal to different customer segments, thus expanding the range of products that could be relevant to diverse consumer preferences. In many cases, suggesting a broader variety of substitutes can enhance the user experience by offering more choices, even if some of those suggestions might not be perfect for every customer.

Further evaluation through a classification model supported the findings, confirming that the model effectively identifies true substitutes. This demonstrates the robustness and reliability of the proposed substitution model in practical scenarios, making it a valuable tool for identifying substitution relationships.

Chapter 5

Visualization Technique for Enhanced Interpretation of ARM Results

5.1 Introduction

Association Rule Mining generates an extensive number of rules that can uncover significant patterns and relationships within large datasets. However, the substantial volume and complexity of these rules present considerable challenges for interpretation and analysis. Effective visualization techniques are crucial for transforming raw ARM results into actionable insights that support decision-making processes. Traditional visualization methods often face limitations in scalability, interpretability, and their ability to reveal hidden knowledge, particularly when handling large and complex rulesets.

Previous chapters addressed the challenges associated with processing and managing ARM results through the development of the specialized Trie of Rules data structure and methodologies for substitution item mining without requiring extensive metadata. Building on these foundations, this chapter emphasizes the visualization of ARM results to enhance interpretability and uncover implicitly hidden

knowledge, such as clusters. While the visualization method introduced here facilitates the identification of substitution patterns, the practical discovery and analysis of substitution patterns using this visualization approach are demonstrated in Chapter 6.

The primary focus of this chapter is to introduce a visualization approach that utilizes the FP-tree structure, adapted into the Trie of Rules, to represent association rules effectively. This method aims to address the limitations of existing visualization techniques by offering a scalable and interpretable representation of ARM results. By overlapping rules with common items and highlighting hierarchical relationships, the proposed visualization enhances understanding of the data, facilitating the identification of patterns, clusters, and potential substitute items.

This chapter provides a detailed explanation of the visualization methodology, focusing on its role in improving the representation of association rules. A case study illustrates the practical application of the approach, while its effectiveness is assessed by evaluating user cognitive load in comparison to traditional visualization methods. The chapter concludes with a discussion of the findings' implications and suggestions for future work and potential enhancements to the visualization technique.

5.2 Methodology

To utilize the **Trie of Rules structure for visualizing association rules**, the concept of rules (discussed in Chapter 3) is adapted for visualization purposes. The Trie of Rules method facilitates understanding of hierarchical relationships between items and rule formation, while also minimizing the plot size by overlapping rules with shared items.

Concept of Rules. In the Trie of Rules, each path from the root (Null node) to a node represents an association rule, where the nodes along the path form the antecedent, and the final node is the consequent (Fig. 5.1). This structure allows users to trace the hierarchical relationships between items and understand

the formation of rules, thereby enhancing the interpretability and manageability of the visualization.

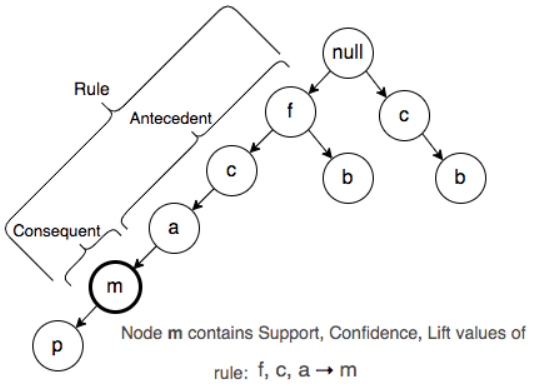


Figure 5.1: The structure of a rule in a Trie of Rules.

Metrics Display. Metrics are displayed through the color and size of nodes, and optionally, through the size of the caption near nodes. For instance, in Figure 5.3a, node size captures Confidence while node color represents Lift, although various other configurations are possible.

The approach also facilitates the discovery of additional insights, such as clusters and substitute items:

- **Clusters:** Groups of items that frequently occur together can be easily identified through their shared paths in the FP-tree structure, revealing natural clusters within the data.

- **Substitute Items:** Items that can replace each other in transactions are revealed through the overlapping paths in the tree, providing insights into alternative itemsets.

5.2.1 Confidence for Compound Consequent

A unique feature of the proposed approach is the ability to calculate Confidence for rules with compound consequents directly from the plot. The Confidence of a compound-consequent rule can be calculated as the multiplication of Confidence

values of the nodes in the consequent, as discussed in Chapter 3 Section 3.2.3 and illustrated in Figure 5.2.

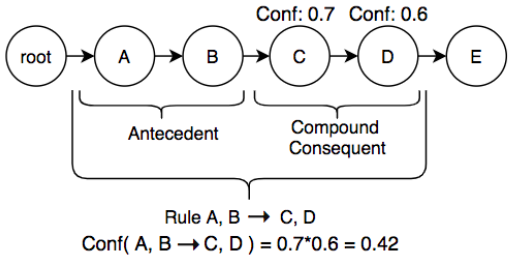


Figure 5.2: A rule with a compound consequent.

Although this method specifically applies to Confidence, the Support value for items with a compound consequent does not require additional calculation. The Support of a rule $A, B, C \rightarrow D$ is equal to the Support of the rule $A, B \rightarrow C, D$. This approach not only enables the derivation of metrics for complex rules but also resolves the issue of visualizing rules with a consequent consisting of multiple items. The behavior of other metrics in the context of a Trie of Rules is beyond the scope of this thesis and is left for future investigation.

It is important to emphasize that the novel contribution of this visualization approach lies in the structured representation of association rules through the Trie of Rules format. By organizing rules into a hierarchical trie structure, the method addresses challenges related to scalability, interpretability, and the overlap of shared itemsets. Translating the ToR into a graph-style view requires careful adaptation to ensure that hierarchical relationships are preserved and visually comprehensible.

The actual rendering of the visualization, including layout generation, node positioning, and graphical encodings (such as node size, color, and label placement), is handled by standard graph visualization tools such as Gephi. These visual encoding choices are flexible and can be adjusted independently of the underlying data structure. Thus, the focus of the contribution is on the preparation and structuring of ARM results into an efficient and interpretable format suitable for graph-based visualization, rather than on the development of new rendering algorithms.

5.2.2 Case Study

The implementation and testing of the Trie of Rules methodology were conducted using the "Online Retail" dataset (D. Chen 2015). This dataset, known for its large size and sparsity, includes 3,663 unique items and 18,484 transactions. A minimum Support threshold of 0.015 was applied for the ARM algorithm, generating 234 association rules. The FP-growth algorithm (Han, Pei, and Yin 2000) was employed to process the dataset, and the developed library (an implementation of the Trie of Rules methodology[1]) was used to produce the graph file.

The resulting Trie of Rules was visualized as a graph structure using Gephi 0.9.2 (Bastian *et al.* 2009). An overlay method "Yifan Hu" (Hu 2006) in Gephi was applied to enhance the clarity of the visualization.

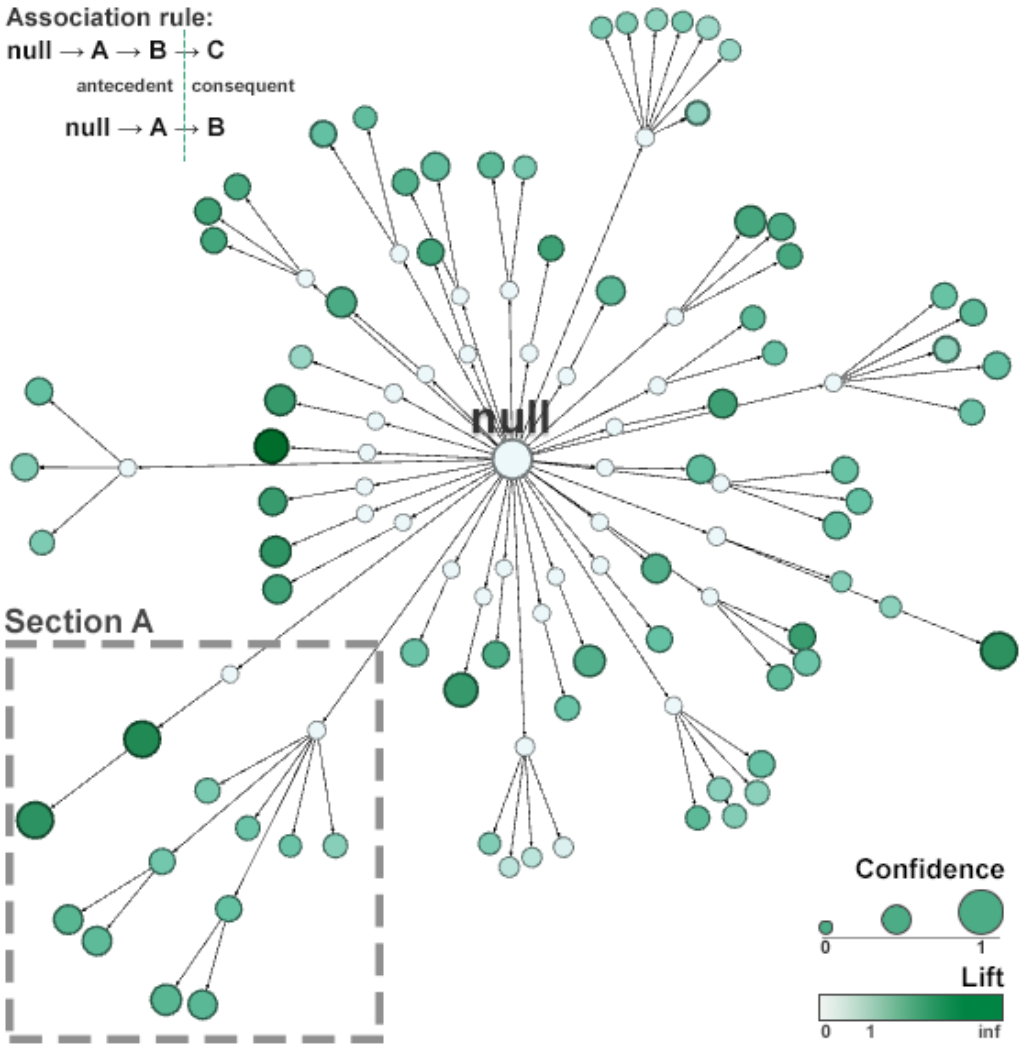
Figure 5.3a illustrates the Trie of Rules generated from the Online Retail dataset. The visualization highlights clusters, the hierarchical structure of association rules, and substitute items, providing valuable insights into the dataset.

There are several valuable implications that can be drawn from exploring Figure 5.3b:

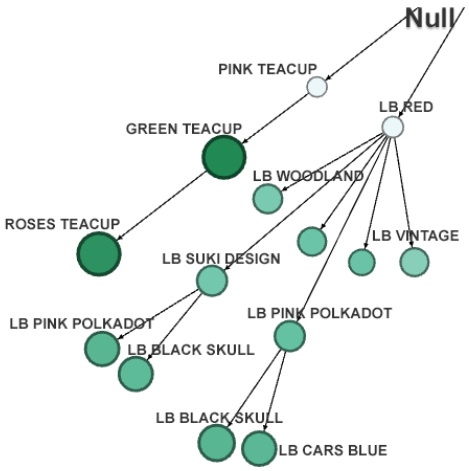
- The branch that starts with *LB RED* represents a cluster of rules that consist solely of Lunch Bag (LB) items in different designs, such as Vintage, Pink Polkadot, and Cars Blue. This indicates that these bags are frequently purchased together in various combinations of designs. Based on this cluster, it is possible to propose selling these items as sets. Additionally, sets of color palettes can be suggested based on the association rules observed in the Trie of Rules, for example, *(RED, VINTAGE)* or *(RED, SUKI DESIGN, PINK POLKADOT)*. Since *(LB RED)* is the starting point of this branch, it can be inferred that *(LB RED)* is the most popular item and could serve as the "default" item in these sets.

- The branch that starts with *(PINK TEACUP)* forms a hierarchy of rules, highlighting sub-rules that demonstrate relationships between items. The color

[1] [https://github.com/ARM-interpretation/Trie-of-rules]



(a)



(b) Section A

Figure 5.3: (a) Trie of Rules visualization of the ARM results for the online retail dataset without captions displayed. (b) Zoomed section A of Figure 5.3a. LB stands for Lunch Bag.

and size of the nodes indicate high Lift and Confidence values. This branch consists of two rules:

1. $PINK\ TEACUP \rightarrow GREEN\ TEACUP$
2. $(PINK\ TEACUP, GREEN\ TEACUP) \rightarrow ROSES\ TEACUP$

The first rule is a sub-rule of the second, illustrating a hierarchical structure where simpler rules are nested within more complex ones. This hierarchy indicates that these items are frequently purchased together with high probability, progressing from individual items to combinations of items. Based on this structure, these items could be sold as a cohesive set of designs. The hierarchical organization naturally suggests a single color palette: *(PINK, GREEN, ROSES)*.

5.3 Evaluation

Evaluating visualization approaches for Association Rule Mining is a complex task. Previous studies have employed various methods to assess the effectiveness of visualization techniques:

- **Expert-feedback:** some researchers simply invite one or two experts to provide subjective feedback on their method's effectiveness (Menin *et al.* 2021; Varu *et al.* 2022).
- **Example-based-validation:** others demonstrate the utility of their visualization techniques using "validation through awesome example" (Ong *et al.* 2002; C. K. S. Leung *et al.* 2009).
- **Comparative-analysis:** another common approach is to outline the advantages and disdavantages of the proposed methods without conducting rigorous user studies (Fernandez-Basso *et al.* 2019; Jentner *et al.* 2019; Hahsler and Chelluboina 2011; Fister *et al.* 2023).

However, those methods are not considered robust enough or objective; literature suggests using more comprehensive evaluation methodologies, such as those described by Elmqvist *et al.* (2012), emphasising the importance of assessing *cognitive load* and *user efficiency*, especially when dealing with complex visualization tasks. Cognitive load refers to the amount of cognitive resources required to perform a task. As highlighted by W. Huang *et al.* (2009), Henike *et al.* (2020), and Yoghourdjian *et al.* (2021), it provides a quantitative measure to compare the efficiency of different visualization methods, making cognitive load a suitable metric in this study. A conceptual construct of cognitive load in the context of visualization efficiency (W. Huang *et al.* 2009) is illustrated in Fig. 5.4.

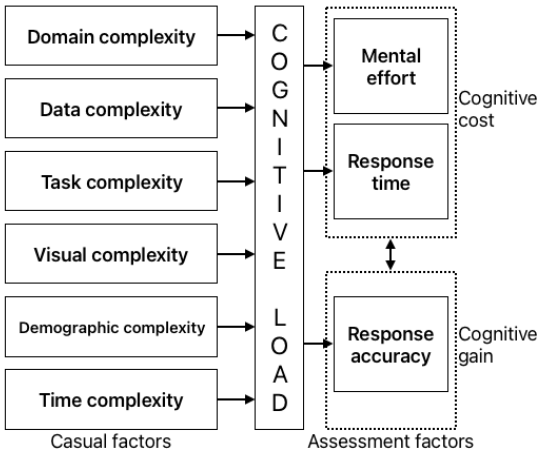


Figure 5.4: The construct of cognitive load for visualization understanding. The figure illustrates the components contributing to cognitive load, highlighting factors critical for effective interpretation of visualizations. Adapted from W. Huang *et al.* (2009).

The evaluation in this study focuses on measuring efficiency and learnability, similar to the approach used by W. Huang *et al.* (2009). The evaluation process involved a carefully designed survey and tasks, structured as follows.

5.3.1 Survey Construction

A survey, approved by the ethical committee of Dublin City University, was designed and conducted as part of this study. To the best of our knowledge, this is the first attempt to evaluate Association Rule Mining visualization methods using cogni-

tive load or any other statistical-based method. This evaluation approach provides unique insights into the usability and efficiency of ARM visualization techniques.

The survey involved 34 participants, all with higher education backgrounds, who completed it remotely on their personal computers. The LimeSurvey[2] platform was utilized to collect responses and record the time taken for each question, with participants informed in advance that their response times were being tracked. Although the survey was conducted anonymously, the participant pool included 14 second-year computer science students from Dublin City University. The remaining 20 participants were recruited through SurveySwap[3], with a restriction to individuals having higher education and a technical background. This selection aimed to reflect the perspectives of potential users of the proposed visualization.

The survey, which took approximately 50 minutes for each participant, included four sections – one for each type of visualization: scatter plot, matrix-based, graph-based (for more details on these visualisation techniques see Section 2.5), and the proposed Trie of Rules approach. The sections were presented in a random order for each participant. At the beginning of the survey, participants were introduced to Association Rule Mining to ensure they could perform the tasks effectively. Participants were not limited in time and were asked the same questions across different visualization methods, but with varying items to ensure consistency.

Each section contained 9 questions:

- One introductory question to assess the ease of understanding the visualization method on a scale from 1 to 10, measuring learnability.

- Four simple questions focusing on tasks such as finding the Support or Confidence of a rule and identifying the rule with the maximum Support or Confidence. For example, "What is the Support of the rule $A \rightarrow B$?" or "Which rule has the highest Support in this dataset?"

[2]LimeSurvey is an open-source survey tool for creating and managing online surveys. More details are available at `https://www.limesurvey.org`.

[3]SurveySwap is an online platform for exchanging survey participation among researchers. More details are available at `https://surveyswap.io`.

- Four complex questions requiring deeper analysis, such as determining relationships between rules, identifying substitute items, assessing clusters, counting rules with a specific item, or finding the longest rule. For instance, "Is item A as popular as item B when purchased with item C?" or "What is the maximum number of items in a rule in this dataset?"

The questions were divided into two groups – simple and complex – to evaluate how well each visualization method supports tasks of varying difficulty. Simple questions were designed to test the basic functionality and efficiency of the visualization methods, focusing on straightforward tasks. In contrast, complex questions required deeper cognitive processing, testing the ability of the visualization methods to facilitate advanced reasoning and analysis, such as exploring relationships between items or interpreting complex patterns. This division helps assess whether the methods are equally effective for both basic and advanced tasks, ensuring a comprehensive evaluation of their usability and effectiveness.

5.3.2 Measured Metrics

The following metrics were measured to evaluate the effectiveness of the visualization techniques:

- **Response Time (RT):** The time taken to complete each task. Shorter response times indicate more efficient visualizations.
- **Cognitive Gain (CG):** The correctness of the answers provided. Higher accuracy indicates higher cognitive gain and, consequently, more effective visualizations.
- **Mental Effort (ME):** Self-reported effort on a scale of 1 to 10. Lower mental effort suggests that the visualization is easier to understand and use.

To standardize the results and facilitate a fair comparison across different visualization methods, z-scores were calculated for these metrics following the methodology proposed by W. Huang *et al.* (2009). The z-score transformation normalizes

the data by subtracting the mean and dividing by the standard deviation of the respective metric, resulting in a standardized score with a mean of 0 and a standard deviation of 1. The formula for calculating the z-score is:

$$z = \frac{X - \mu}{\sigma} \tag{5.1}$$

where X is the raw score, μ is the mean of the scores, and σ is the standard deviation.

The following formula for visualization efficiency was utilized (W. Huang *et al.* 2009):

$$E = Z_{CG} - Z_{ME} - Z_{RT} \tag{5.2}$$

In this formula, E represents the efficiency via cognitive load, Z_{CG} is the z-score for cognitive gain, Z_{ME} is the z-score for mental effort, and Z_{RT} is the z-score for response time. This metric captures the trade-off between cognitive gain, effort, and time, providing a comprehensive measure of visualization efficiency. High efficiency is achieved when high cognitive gain is associated with low mental effort and short response time.

5.3.3 Survey Results and Analysis

The results of the evaluation are summarized in Table 5.1 and Table 5.2.

Table 5.1: Means and standard deviations of response time, cognitive gain, mental effort, and efficiency on simple questions

	Trie of Rules		Matrix		Graph		Scatter	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Time (sec.)	56	39	43	29	73	44	40	28
Cognitive Gain	0.44	0.18	0.34	0.23	0.20	0.22	0.44	0.15
Effort	3.11	1.10	3.32	1.26	3.03	1.08	2.57	1.10
Efficiency	0.23		-0.46		-2.76		2.99	

In terms of cognitive gain, the Trie of Rules method demonstrated better performance on complex questions (0.59) compared to the other methods (Matrix: 0.17, Graph: 0.29, Scatter: 0.23). This indicates that while the Trie of Rules may be novel

Table 5.2: Means and standard deviations of response time, cognitive gain, mental effort, and efficiency on complex questions

	Trie of Rules		Matrix		Graph		Scatter	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Time (sec.)	35	24	40	32	46	33	35	22
Cognitive Gain	0.59	0.13	0.17	0.19	0.29	0.05	0.23	0.10
Effort	3.11	1.10	3.32	1.26	3.03	1.08	2.57	1.10
Efficiency	1.89		-1.99		-1.56		1.66	

and less familiar to users, its structured representation of association rules enables more accurate analysis of complex relationships. However, for simple questions, the cognitive gain of the Trie of Rules (0.44) was on par with the Scatter plot (0.44) and better than the Matrix (0.34) and Graph (0.20) methods. This suggests that while the Trie of Rules is effective for both simple and complex tasks, its advantage becomes more pronounced with increased complexity. This aligns with the design of the Trie of Rules, which is specifically built to facilitate the analysis of intricate patterns and complex associations in large rulesets.

Regarding mental effort, an Analysis of Variance (ANOVA) test was conducted to assess differences between the methods. The results indicated a statistically significant difference in mental effort across the methods ($F = 2.83$, $p = 0.041$). To further examine these differences, pairwise comparisons were performed using Tukey's HSD test.[4]

The results of the Tukey HSD test, shown in Table 5.3, demonstrate that the Trie of Rules method showed *no statistically significant differences in mental effort* compared to the Graph-Based, Matrix-Based, and Scatter Plot methods. Specifically, all adjusted p-values for comparisons involving the Trie of Rules exceeded 0.05, indicating that mental effort required by this approach is comparable to other methods. This finding supports the conclusion that the Trie of Rules is not more cognitively demanding than traditional methods.

The only significant difference observed was between the Matrix-Based and Scat-

[4]Tukey's Honest Significant Difference (HSD) test is a post-hoc statistical test used to determine whether the means of different groups differ significantly. It controls the family-wise error rate and is typically used following an ANOVA to perform pairwise comparisons while accounting for multiple testing.

Table 5.3: Multiple Comparison of Means of Mental Effort – Tukey HSD

Group 1	Group 2	Mean Diff	p-adj	Lower	Upper	Reject
Trie of Rules	Graph-Based	0.0789	0.9903	-0.5992	0.7570	False
Trie of Rules	Matrix-Based	-0.2105	0.8512	-0.8886	0.4676	False
Trie of Rules	Scatter Plot	0.5263	0.1865	-0.1518	1.2044	False
Graph-Based	Matrix-Based	-0.2895	0.6844	-0.9676	0.3886	False
Graph-Based	Scatter Plot	-0.4474	0.3199	-1.1255	0.2307	False
Matrix-Based	Scatter Plot	-0.7368	0.0274	-1.4149	-0.0587	True

ter Plot methods ($p = 0.0274$), with the Scatter Plot requiring the least effort ($mean = 2.57$). This result aligns with the intuitive and widely recognized nature of scatter plots (Friendly 2008; Ware 2012).

The response time for simple questions was slightly higher for the Trie of Rules (56 seconds) compared to the other methods, with the Scatter plot being the fastest (40 seconds). This suggests that users may need more time to familiarize themselves with the Trie of Rules. However, for complex questions, the Trie of Rules (35 seconds) performed on par with the Scatter plot (35 seconds), indicating that once users become familiar with the method, they can analyze complex information just as quickly as with more traditional methods.

5.4 Summary

Existing data representation methods often struggle with scalability, interpretability, and the effective representation of rule structures, which limits their practical utility in real-world applications. This chapter introduced the "Trie of Rules" as a novel visualization technique that addresses these limitations by providing a structured and efficient way to represent association rules. The Trie of Rules captures a wealth of information, reveals implicit insights such as clusters and substitute items, and maintains a manageable visualization size by overlapping common items.

The evaluation results demonstrate that the Trie of Rules method significantly outperforms existing visualization methods in terms of efficiency and accuracy, particularly for complex queries. While the slightly higher response time for simple

questions suggests a learning curve due to its novel representation, the method remains comparable to traditional approaches, such as Scatter Plots, in terms of mental effort. The benefits of the Trie of Rules become more pronounced in scenarios involving larger datasets and more complex association rules, highlighting its potential for scaling effectively with data size.

These findings underscore the Trie of Rules' ability to enhance the interpretability and usability of ARM visualizations. By offering an accurate and efficient means to analyze intricate relationships within the data, the method facilitates better decision-making and knowledge discovery.

Chapter 6

Case Study: Evaluating the Trie of Rules for Substitution Mining and Visualization

This case study demonstrates the practical application of the methodologies developed throughout this thesis, focusing on the Trie of Rules data structure for substitution mining. By integrating key components such as theoretical insights, computational analysis, and visualization, it showcases how these methodologies work together to improve the efficiency, scalability, and usability of Association Rule Mining processes. The study highlights the effectiveness of the proposed approach compared to traditional methods, emphasizing its potential for real-world applications.

6.1 Substitution Analysis Using the Trie of Rules

In the Trie of Rules, association rules are stored in a hierarchical manner where common antecedents share the same paths before diverging into different consequents. This structure inherently exposes points of divergence, which are critical in the proposed substitution analysis. Specifically, when branches split within the trie, it indicates that the same context (antecedent) leads to different items (consequent),

suggesting potential substitution relationships.

For example, consider a branch in the trie representing the context C leading to item A, and another branch where the same context C leads to item B. The point at which the paths diverge, as illustrated in Figure 6.1, indicates the potential substitution between items A and B within the context C. By focusing on these divergence points, the analysis can be limited to relevant item pairs, significantly reducing the number of computations required.

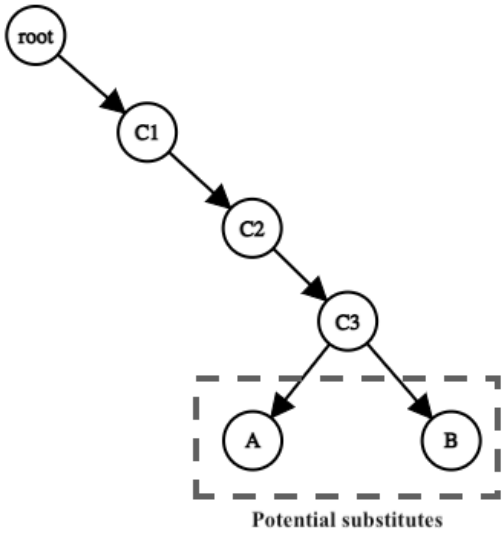


Figure 6.1: Illustration of the Trie of Rules highlighting divergence points for substitution analysis. Common antecedent or context (C1, C2, C3) leads to different consequents (A and B), indicating potential substitutes at the split nodes.

Traditional substitution mining approaches often require scanning all possible item pairs, leading to a computational complexity of $O(n^2)$ for n sets of items. This becomes impractical with large datasets, as the number of possible pairs grows exponentially.

By utilizing the Trie of Rules, the computational complexity is reduced. The trie allows to:

- **Prune Irrelevant Paths:** Only paths that share a common context up to a divergence point need to be considered for substitution analysis.
- **Focus on Split Nodes:** Potential substitutions are indicated by split nodes

where the trie branches into different items, reducing the number of item pairs to analyze.

This targeted approach reduces the number of necessary computations from $O(n^2)$ to $O(k)$, where k is the number of split nodes in the trie, and typically $k < n$. By analyzing different thresholds on the retail dataset (D. Chen 2015), it was observed that k is consistently several times smaller than n (Figure 6.2). This relationship demonstrates the efficiency of the Trie of Rules structure in focusing only on relevant nodes for computation.

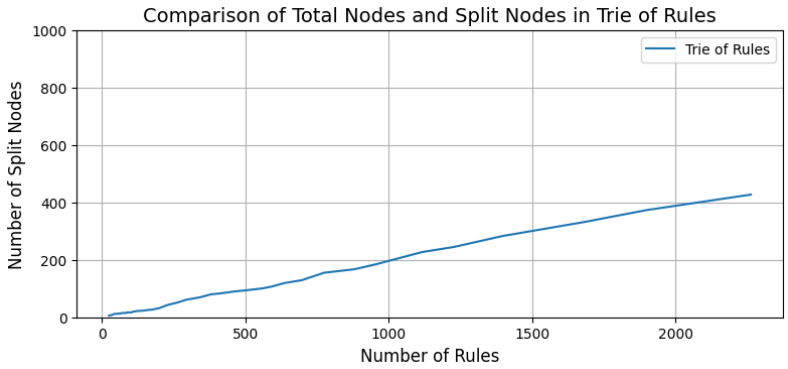


Figure 6.2: Comparison of total nodes versus split nodes across different thresholds. The figure demonstrates a linear dependency between the number of total nodes and split nodes as thresholds vary, highlighting the consistent scalability of the Trie of Rules structure.

To evaluate the efficiency of the Trie of Rules in substitute item mining (Algorithm 8, Appendix A), a comparative analysis was conducted against the same technique implemented using a traditional data structure, specifically a flat, table-like representation in the form of Pandas. The execution time for each method was measured on rulesets of varying sizes generated by applying different minimum Support thresholds. The results, shown in Figure 6.3, indicate that the Trie of Rules consistently outperformed traditional implementation in terms of execution time.

A paired t-test was performed to statistically validate the observed differences in execution times. The null hypothesis, stating that the difference in execution times between the Trie of Rules and the pandas-based approach is not significantly different from zero, was rejected. The test confirmed that the performance improvement

achieved by the Trie of Rules is statistically significant, yielding $t = -3.24$, $df = 49$, and $p = 0.002$.

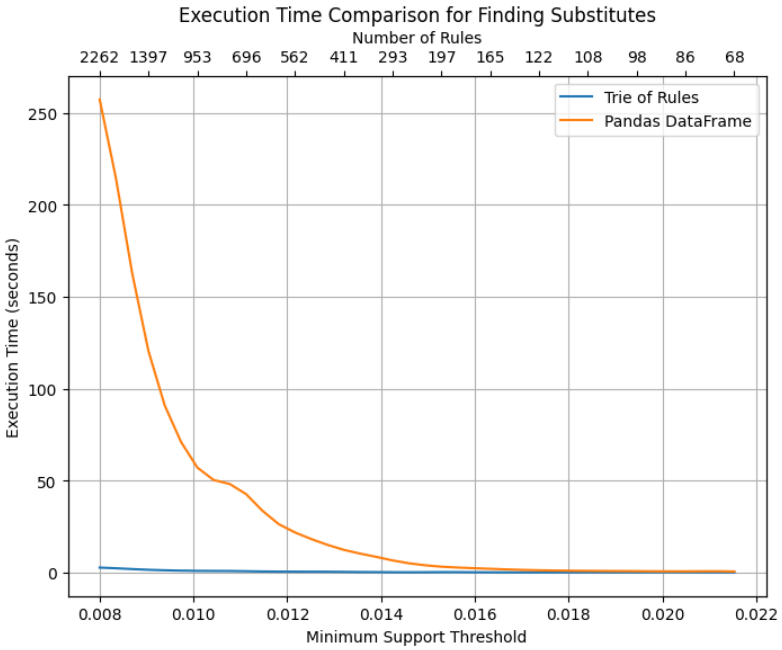


Figure 6.3: Comparison of substitute item mining times between a plain structure and the Trie of Rules. The Trie of Rules significantly reduces mining times, particularly for larger rulesetes.

6.2 Visualization and Calculation of Substitution Relationships

To demonstrate the application of the proposed visualization approach and substitution calculation methodology, the analysis was applied to the Sporting Goods Store dataset (Microsoft 2019), a transactional dataset from a bike and sports store. The dataset was processed with a minimum Support threshold of 0.004 and no minimum Confidence threshold, resulting in 129 association rules. The visualization of these rules using the Trie of Rules structure is presented in Figure 6.4. The plot highlights distinct clusters and potential substitution pairs within each cluster, providing insights into customer purchasing patterns.

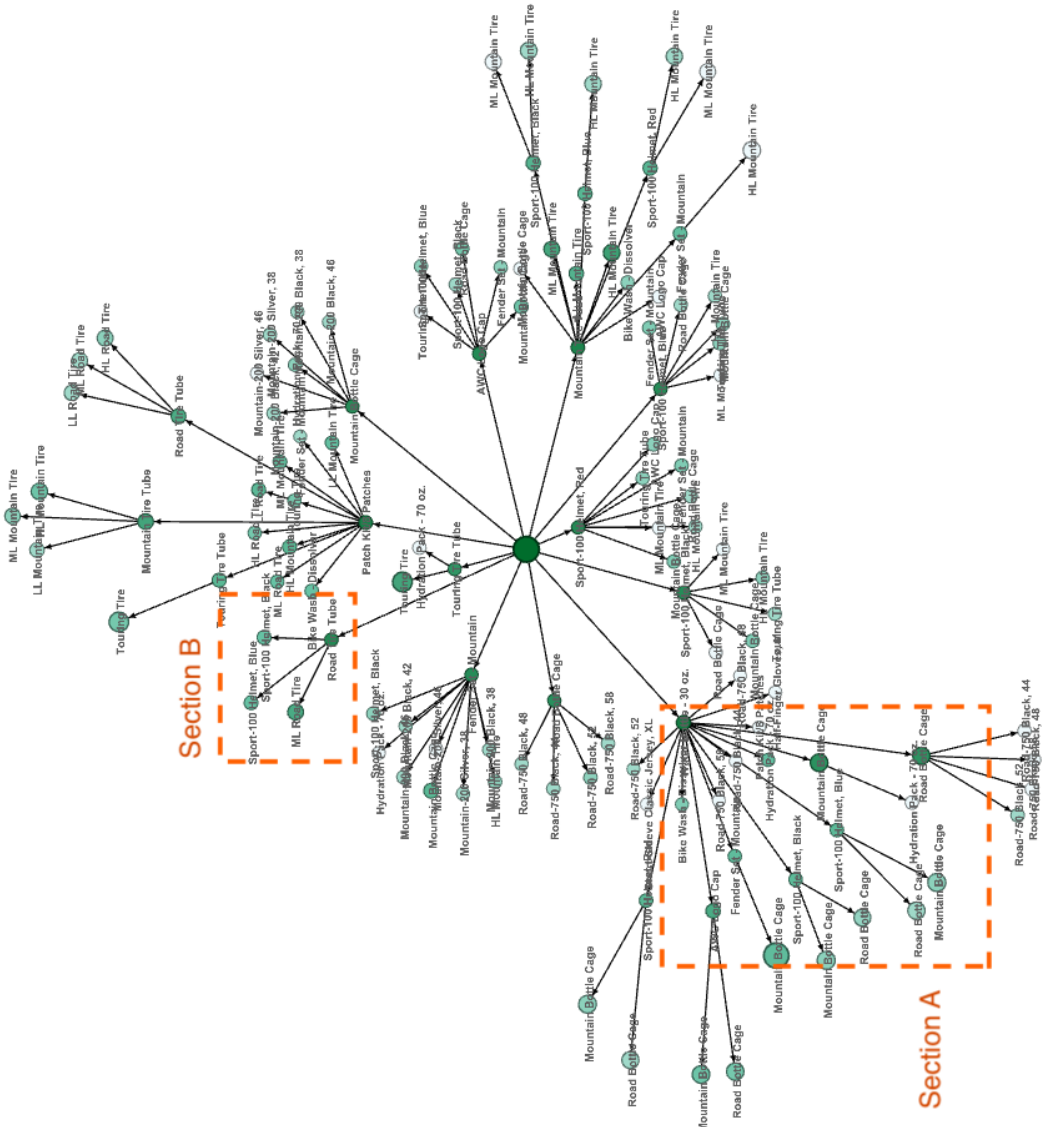
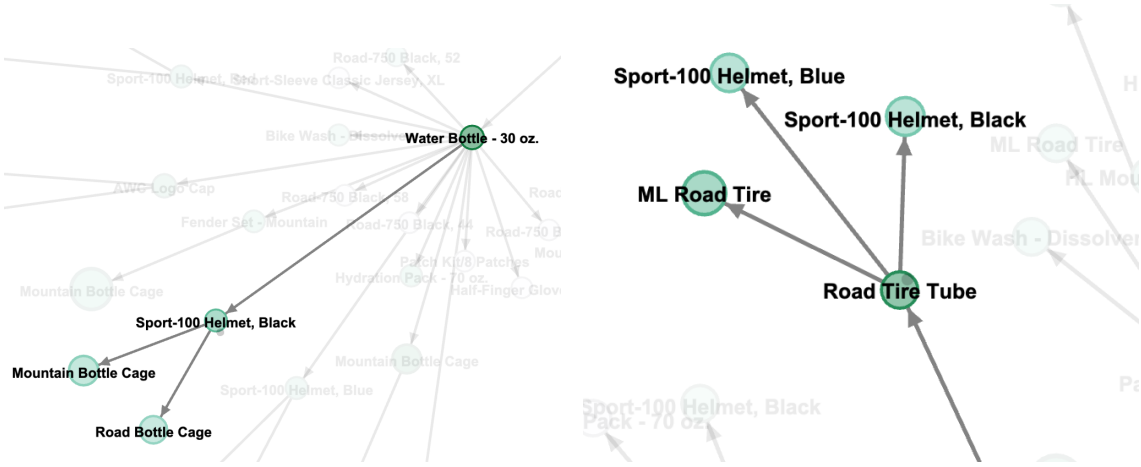


Figure 6.4: Visualization of the Trie of Rules gathered from Sporting Goods Store dataset, where node colour indicates Support and node size represents Confidence



(a) Section A of Figure 6.4 (b) Section B of Figure 6.4

Figure 6.5: Sections A and B of Figure 6.4

In Figure 6.5a (Section A of Figure 6.4), the context (C) consists of "Water Bottle 30 oz." and "Sport-100 Helmet, Black". Two possible consequents – "Mountain Bottle Cage" *(Support = 0.006)* and "Road Bottle Cage" *(Support = 0.004)* – are observed. This suggests that customers who purchase a helmet and a water bottle tend to choose bottle cages based on their bicycle type: mountain or road.

The substitution value between "Mountain Bottle Cage" and "Road Bottle Cage" within this context was calculated as follows:

$$\begin{aligned}\text{Similarity} &= 1 - \frac{\sup(C \to \text{Mountain}) - \sup(C \to \text{Road})}{\sup(C \to \text{Mountain}) + \sup(C \to \text{Road})} \\ &= 1 - \frac{0.006 - 0.004}{0.006 + 0.004} \\ &= 0.8.\end{aligned}$$

The co-appearance value, calculated based on the joint Support of both outcomes, is:

$$\begin{aligned}\text{Co-appearance} &= \frac{\sup(C \to \text{Mountain, Road})}{\sup(C \to \text{Mountain}) + \sup(C \to \text{Road}) + \sup(C \to \text{Mountain, Road})} \\ &= 0,\end{aligned}$$

since there are no transactions where both bottle cages were purchased together.

Finally, the substitution value is:

$$\begin{aligned} \text{Substitution} &= \text{Similarity} \times (1 - \text{Co-appearance}) \\ &= 0.8 \times 1 \\ &= 0.8. \end{aligned}$$

This result indicates a strong substitution relationship between the two items, aligning with the assumption that customers choose between these items based on their specific bicycle type.

Section B (Figure 6.5b) presents a context of "Road Tire Tube" (*Tire Tube, denoted as* TT) with three possible consequents:

1. "Sport-100 Helmet, Blue" (*Helmet Blue, denoted as* HB),

2. "Sport-100 Helmet, Black" (*Helmet Black, denoted as* HBl), and

3. "ML Road Tire" (*Tire, denoted as* T), where ML refers to Medium Load tires.

Within the context of TT, people might purchase either a helmet or tire. While two types of helmets (HB and HBl) may be substitutable for each other, a tire (T) does not appear to be a logical substitute for a helmet. Calculations support this observation and is based on the data from Table 6.1.

Table 6.1: Support values for items in Section B.

Rule	Support
$TT \rightarrow HB$	0.01
$TT \rightarrow HBl$	0.01
$TT \rightarrow T$	0.022
$TT \rightarrow HB, HBl$	0
$TT \rightarrow T, HBl$	0.00314

The substitution value between *HB* and *HBl* is calculated as follows:

$$\begin{aligned}
\text{Similarity} &= 1 - \frac{|\sup(TT \to HB) - \sup(TT \to HBl)|}{\sup(TT \to HB) + \sup(TT \to HBl)} \\
&= 1 - \frac{|0.01 - 0.01|}{0.01 + 0.01} \\
&= 1. \\
\text{Co-appearance} &= \frac{\sup(TT \to HB,\ HBl)}{\sup(TT \to HB) + \sup(TT \to HBl) + \sup(TT \to HB,\ HBl)} \\
&= \frac{0}{0.01 + 0.01 + 0} \\
&= 0. \\
\text{Substitution} &= \text{Similarity} \times (1 - \text{Co-appearance}) \\
&= 1 \times 1 \\
&= 1.
\end{aligned}$$

This calculation confirms that *HB* (Helmet Blue) and *HBl* (Helmet Black) are perfectly substitutable. In contrast, the substitution value between *HBl* (Helmet Black) and *T* (Tire) is significantly lower, as shown below:

$$\begin{aligned}
\text{Similarity} &= 1 - \frac{|\sup(TT \to HBl) - \sup(TT \to T)|}{\sup(TT \to HBl) + \sup(TT \to T)} \\
&= 1 - \frac{|0.01 - 0.022|}{0.01 + 0.022} \\
&= 1 - 0.375 \\
&= 0.625. \\
\text{Co-appearance} &= \frac{\sup(TT \to HBl,\ T)}{\sup(TT \to HBl) + \sup(TT \to T) + \sup(TT \to HBl,\ T)} \\
&= \frac{0.00314}{0.01 + 0.022 + 0.00314} \\
&= 0.11. \\
\text{Substitution} &= \text{Similarity} \times (1 - \text{Co-appearance}) \\
&= 0.625 \times 0.89 \\
&\approx 0.56.
\end{aligned}$$

The substitution value of 1 between two Helmets of different colours (*HB* and *HBl*) indicates perfect substitutability, while the lower substitution value of 0.56 between *HBl* (Helmet) and *T* (Tire) shows a lower likelihood of being a substitute and suggests a lack of meaningful substitution. This demonstrates how the proposed methodology effectively identifies true substitutes using only data-driven observations without relying on external metadata.

6.3 Summary

The case study presented in this chapter demonstrates the practical utility and versatility of the proposed Trie of Rules data structure and associated methodologies. These innovations, when applied to real-world datasets, have demonstrated their effectiveness in addressing challenges related to scalability, substitution analysis, and visualization in Association Rule Mining. The results validate the theoretical contributions of this thesis, showcasing significant improvements in efficiency and interpretability. These findings support the value of the proposed approach and underscore its potential for adoption in various domains.

Chapter 7

Conclusion

7.1 Summary of Contributions

This thesis has addressed critical challenges in the post-mining phase of Association Rule Mining, focusing on scalability, interpretability, and practical applicability. By proposing novel methodologies and data structures, this research contributes to the field in three significant ways:

- A **Trie of Rules** data structure was developed, enabling efficient storage, retrieval, and manipulation of large rulesets. The Trie of Rules organizes association rules into a hierarchical structure, where nodes represent items and paths encode rules, allowing common prefixes to be shared among related rules. This approach reduces redundancy and significantly optimizes memory usage compared to traditional "flat structures". Experimental evaluations demonstrated that the proposed method achieves substantial improvements in time efficiency, with query processing and rule retrieval times being up to 7 times faster on average compared to baseline methods. These enhancements establish the Trie of Rules as a scalable and effective solution for handling the large and complex datasets characteristic of modern ARM applications.

- A robust **methodology for substitute item mining** was introduced, leveraging the Trie of Rules to efficiently identify patterns of substitutability within

association rules. By encoding and traversing the hierarchical structure of the Trie of Rules, the methodology detects relationships between items based on co-occurrence and similarity of rules. This approach provides a systematic way to analyze substitutability across large datasets, ensuring accuracy and scalability. Evaluations using mixed models demonstrated strong alignment between predicted and observed substitutability patterns, confirming the methodology's reliability. Its potential applications span inventory management, consumer behavior analysis, and recommendation systems, where understanding item interchangeability is crucial for optimizing operations and enhancing decision-making processes.

- An advanced **visualization technique** was proposed to improve the interpretability of ARM results by transforming complex association rulesets into intuitive and interactive visual representations. This technique leverages the hierarchical structure of the Trie of Rules to generate interpretable plots that highlight key patterns and relationships, enabling users to explore and analyze data more effectively. Usability studies and user feedback indicated that these visualizations reduced cognitive load by simplifying the navigation of large datasets and improved decision-making efficiency. By bridging the gap between complex data and actionable insights, these tools empower both technical and non-technical users to make informed decisions based on ARM outputs.

These contributions collectively address the identified gaps in current literature and advance the usability of ARM in real-world scenarios.

7.2 Implications for Association Rule Mining

The methodologies presented in this thesis significantly extend the practical capabilities of Association Rule Mining by addressing its key limitations and providing robust, scalable, and versatile solutions. These advancements have implications for

both theoretical research and practical applications, as detailed below:

- **Scalability**: The Trie of Rules data structure represents a breakthrough in handling the constantly increasing size of modern datasets. By reducing redundancy and efficiently organizing rules in a hierarchical format, this approach minimizes storage requirements and processing times. This scalability ensures that ARM remains relevant and practical for industries dealing with massive transactional datasets, such as retail and finance, where quick and accurate rule retrieval is essential for timely decision-making.

- **Interpretability**: Enhanced visualization method improves the efficiency of knowledge discovery by allowing an easy transition from complex data outputs to actionable insights. By leveraging advanced graphical representations tailored to the Trie of Rules structure, domain experts can intuitively distinguish key patterns without requiring deep technical expertise. These tools also facilitate collaborative decision-making by enabling non-technical users to participate in interpreting and applying ARM results effectively, thus broadening access to data insights.

- **Applicability**: The introduction of a robust substitute item mining methodology opens new avenues for applying ARM beyond its traditional use cases. By relying solely on the inherent structure and co-occurrence patterns within the data, rather than requiring external meta-data, this methodology achieves a high degree of adaptability. This independence from meta-data broadens its applicability to diverse domains where such additional information might be unavailable or inconsistent. Businesses can utilize this approach to optimize inventory management and pricing strategies, respond dynamically to consumer preferences, and address supply chain disruptions. Furthermore, its versatility extends ARM's utility into areas such as personalized recommendations, market basket analysis, and sustainability initiatives, where identifying substitutability can support waste reduction and resource efficiency. This ver-

satility highlights the methodology's capability to improve decision-making across different industries and applications.

Overall, these advancements position ARM as a critical tool for data-driven decision-making, ensuring its adaptability and relevance in the world of big data. By addressing the foundational challenges of scalability, interpretability, and applicability, this research provides a pathway for future innovations in the field.

7.3 Future Research Directions

While this thesis provides significant advancements, it also opens new avenues for future research, which are outlined below:

1. **Dynamic Substitution Analysis**: Future work could explore real-time substitution analysis, incorporating temporal and spatial dynamics to better understand context-dependent relationships. This would involve studying how substitutability evolves over time or varies across different data features, such as geographical regions. Temporal dynamics could help identify seasonal or trend-based changes in substitution patterns, while spatial dynamics could uncover location-specific preferences or constraints. For example, integrating this methodology with real-time inventory data or consumer behavior analytics could enable businesses to respond proactively to market changes, thus enhancing the methodology's practical utility.

2. **Integration with Graph-Based Machine Learning**: Combining the Trie of Rules with graph-based machine learning techniques presents an exciting avenue for uncovering deeper and more complex patterns. By treating the Trie as a graph structure, a wide variety of machine learning methods designed for graphs can be utilized to analyze relationships within the data. Techniques such as clustering, community detection, and graph-based similarity measures can uncover inherent structures and dependencies in the data. Additionally, statistical methods like graph centrality metrics, spectral graph analysis, and

random walk models can provide insights into latent substitutability patterns and higher-order relationships. These approaches can reveal subtle interactions that may not be evident through traditional analysis, enabling more nuanced recommendations and enhanced decision-making processes.

3. **Visualization Enhancements**: Developing interactive and customizable visualization tools would further empower users to tailor insights to their specific needs. Future research could focus on creating dynamic dashboards that allow users to filter, group, and drill down into specific rule subsets based on contextual requirements. Incorporating features such as real-time updates, comparative visualizations, and augmented reality interfaces could make these tools more engaging and practical for stakeholders. Additionally, providing domain-specific visualization templates could ensure that these tools meet the diverse needs of industries like retail, healthcare, and finance.

4. **Expansion to Multi-Domain Applications**: Applying the proposed methodologies in domains such as bioinformatics, finance, and social network analysis offers opportunities to uncover unique challenges and extend the methodology's utility. This could be further enhanced by introducing multi-layer networks, which could integrate the Trie of Rules with knowledge graphs. These layered structures could combine transactional data with domain knowledge, enriching the rule-mining process. For instance, in bioinformatics, integrating genetic interaction networks with the Trie could facilitate the discovery of substitution patterns in gene functions or pathways. In finance, layering transaction data with market sentiment graphs could yield insights into consumer behavior shifts. In social network analysis, combining user activity logs with social connections could reveal nuanced substitution patterns in online interactions.

7.4 Closing Remarks

This thesis has made meaningful contributions to Association Rule Mining, addressing critical challenges in scalability, interpretability, and applicability. The Trie of Rules introduces a robust approach to efficiently storing and processing large rulesets, ensuring ARM remains scalable in the world of growing data demands. The substitute item mining methodology broadens ARM's applicability, providing an adaptable framework for uncovering substitutability patterns without reliance on external meta-data. The advanced visualization technique increases the efficiency of interpretability without introducing additional complexity in understanding the plot.

Through these rigorous empirical studies, we provide practical solutions to long-standing issues in ARM while opening avenues for future exploration. By combining innovation with applicability, this research ensures that ARM continues to evolve as a critical tool for navigating the complexities of big data. This thesis not only addresses current challenges but also sets the stage for transformative applications across diverse domains, supporting the importance of ARM in our data-driven world.

Data Availability The datasets used in this study are publicly available. The Groceries dataset can be downloaded at `https://github.com/mhahsler/arules/blob/master/data/Groceries.rda`. The Retail dataset can be downloaded at `https://archive.ics.uci.edu/dataset/352/online+retail`. The Sporting Goods Store dataset can be downloaded at `https://www.kaggle.com/datasets/cnezhmar/sporting-goods-store`

Bibliography

Achanuparp, Palakorn and Ingmar WEBER (2016). "Extracting Food Substitutes From Food Diary via Distributional Similarity". In: *10th ACM Conference on Recommender Systems*.

Agrawal, Rakesh, Tomasz Imieliński, and Arun Swami (June 1993). "Mining association rules between sets of items in large databases". In: *ACM SIGMOD Record* 22.2, pp. 207–216. ISSN: 0163-5808. DOI: 10.1145/170036.170072.

Agrawal, Rakesh and Ramakrishnan Srikant (1995). "Mining sequential patterns". In: *Proceedings of the Eleventh International Conference on Data Engineering*. IEEE, pp. 3–14.

Ait-Mlouk, Addi, Tarik Agouti, and Fatima Gharnati (2017). "Mining and prioritization of association rules for big data: multi-criteria decision analysis approach". In: *Journal of Big Data* 4, pp. 1–21.

Akkoyunlu, Sema, Cristina Manfredotti, Antoine Cornuéjols, Nicolas Darcel, Sema Akkoyunlu, Cristina Manfredotti, Antoine Cornuéjols, Nicolas Darcel, Fabien Delaere Investi-, Sema Akkoyunlu, Cristina Manfredotti, Antoine Cornuéjols, Nicolas Darcel, and Fabien Delaere (2020). "Investigating substitutability of food items in consumption data To cite this version : HAL Id : hal-02482142 Investigating substitutability of food items in consumption data fr". In.

Alasow, Mohamed A., Salahadin A. Mohammed, and El-Sayed M. El-Alfy (2020). "Parallel association rules pruning algorithm on Hadoop MapReduce". In: *Applied Soft Computing and Communication Networks: Proceedings of ACN 2019*. Ed. by Sabu M. Thampi, Elizabeth Sherly, Soura Dasgupta, Jaime Lloret Mauri,

Jemal H. Abawajy, Evgeny Khorov, and Jimson Mathew. Singapore: Springer Singapore, pp. 117–130. ISBN: 978-981-15-3852-0. DOI: 10.1007/978-981-15-3852-0_8.

Albashrawi, Mousa (July 2016). "Detecting Financial Fraud Using Data Mining Techniques: A Decade Review from 2004 to 2015". In: *Journal of Data Science* 14, pp. 553–570. DOI: 10.6339/JDS.201607_14(3).0010.

Alyobi, Manal A and Arwa A Jamjoom (2020). "A Visualization Framework for Post-Processing of Association Rule Mining". In: *International Journal Transaction on Machine Learning and Data Mining* 2020.2, pp. 83–99.

Amit Pande, Aparupa Das Gupta, Kai Ni, Rahul Biswas, and Sayon Majumdar (2022). "Substitution Techniques for Grocery Fulfillment and AssortmentOptimization Using Product Graphs". In: *ACM Computing Surveys*. ISSN: 0360-0300.

Bastian, Mathieu, Sebastien Heymann, and Mathieu Jacomy (2009). *Gephi: An Open Source Software for Exploring and Manipulating Networks*. URL: http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154.

Bates, Douglas (2007). "Linear mixed model implementation in lme4". In: *Manuscript, University of Wisconsin* 15, pp. 5–2.

Bayardo, Roberto J, Rakesh Agrawal, and Dimitrios Gunopulos (2000). "Constraint-based rule mining in large, dense databases". In: *Data mining and knowledge discovery* 4, pp. 217–240.

Bayardo, Roberto J. and Rakesh Agrawal (1999). "Mining the most interesting rules". In: *Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining - KDD '99*. New York, New York, USA: ACM Press, pp. 145–154. ISBN: 1581131437. DOI: 10.1145/312129.312219.

Belton, Valerie and Theodor J. Stewart (2002). *Multi-Criteria Decision Analysis: An Integrated Approach*. Springer Science & Business Media.

Berrado, Abdelaziz and George C. Runger (2007). "Using metarules to organize and group discovered association rules". In: *Data Mining and Knowledge Discovery* 14.3, pp. 409–431. ISSN: 13845810. DOI: 10.1007/s10618-006-0062-6.

Bodon, F. and L. Rónyai (2003). "Trie: an alternative data structure for data mining algorithms". In: *Mathematical and computer modelling*. Vol. 38. 7-9, pp. 739–751. DOI: 10.1016/0895-7177(03)90058-6.

Brin, Sergey, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur (1997). "Dynamic itemset counting and implication rules for market basket data". In: *SIGMOD Record (ACM Special Interest Group on Management of Data)* 26.2, pp. 255–264. ISSN: 01635808. DOI: 10.1145/253262.253325.

Bui-Thi, D., P. Meysman, and K. Laukens (2022). "Momac: multi-objective optimization to combine multiple association rules into an interpretable classification". In: *Applied Intelligence*. DOI: 10.1007/s10489-021-02595-w.

Bui-Thi, Danh, Pieter Meysman, and Kris Laukens (2020). "Clustering association rules to build beliefs and discover unexpected patterns". In: *Applied Intelligence* 50.6, pp. 1943–1954. ISSN: 15737497. DOI: 10.1007/s10489-020-01651-1.

Buono, Paolo and Maria Francesca Costabile (2005). "Visualizing Association Rules in a Framework for Visual Data Mining". In: *From Integrated Publication and Information Systems to Information and Knowledge Environments*. Springer Berlin Heidelberg, pp. 221–231. ISBN: 978-3-540-31842-2. DOI: 10.1007/978-3-540-31842-2_22.

Chen, Daqing (2015). *Online retail*. UCI Machine Learning Repository. URL: https://doi.org/10.24432/C5BW33.

Chen, Yi Chu and Guanling Lee (2015). *Mining non-redundant substitution rules between sets of items in large databases*. Tech. rep. 2, pp. 659–674. DOI: 10.6688/JISE.2015.31.2.16.

Coenen, Frans, Paul Leng, and Shakil Ahmed (2004). "Data structure for association rule mining: T-trees and P-trees". In: *IEEE Transactions on Knowledge and Data Engineering* 16.6, pp. 774–778. ISSN: 10414347. DOI: 10.1109/TKDE.2004.8.

Cornelis, Chris, Peng Yan, Xing Zhang, and Guoqing Chen (2006). "Mining positive and negative association rules from large databases". In: *2006 IEEE Conference*

on Cybernetics and Intelligent Systems, pp. 0–5. DOI: 10.1109/ICCIS.2006.252251.

Crochemore, Maxime and Thierry Lecroq (2009). "Trie". In: *Encyclopedia of Database Systems*. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, pp. 3179–3182. ISBN: 978-0-387-39940-9. DOI: 10.1007/978-0-387-39940-9_1143.

De Giovanni, Pietro and Vinay Ramani (2018). "Product cannibalization and the effect of a service strategy". In: *Journal of the operational research society* 69.3, pp. 340–357.

De Padua, Renan, Lais Pessine Do Carmo, Solange Oliveira Rezende, and Veronica Oliveira De Carvalho (2018). "An analysis on community detection and clustering algorithms on the post-processing of association rules". In: *Proceedings of the International Joint Conference on Neural Networks* 2018-July. DOI: 10.1109/IJCNN.2018.8489603.

Demidenko, Eugene (2013). *Mixed models: theory and applications with R*. John Wiley & Sons.

Draper, Norman R. and Harry Smith (1998). *Applied Regression Analysis*. 3rd. John Wiley & Sons. ISBN: 978-0-471-17082-2.

Elmqvist, Niklas and Ji Soo Yi (2012). "Patterns for visualization evaluation". In: *ACM International Conference Proceeding Series* October 2012. DOI: 10.1145/2442576.2442588.

Ertek, Gürdal and Ayhan Demiriz (2006). "A Framework for Visualizing Association Mining Results". In: *Computer and Information Sciences – ISCIS 2006*. Springer Berlin Heidelberg, pp. 593–602. ISBN: 978-3-540-47243-8. DOI: 10.1007/11902140_63.

Fernandez-Basso, Carlos, M. Dolores Ruiz, Miguel Delgado, and Maria J. Martin-Bautista (2019). "A comparative analysis of tools for visualizing association rules: A proposal for visualising fuzzy association rules". In: *Proceedings of the 11th conference of the european society for fuzzy logic and technology (EUSFLAT*

2019). Atlantis Press, pp. 520–527. ISBN: 978-94-6252-770-6. DOI: 10.2991/eusflat-19.2019.72.

Fister, Iztok, Iztok Fister, Dušan Fister, Vili Podgorelec, Iztok Fister, and Sancho Salcedo-Sanz (2023). "A comprehensive review of visualization methods for association rule mining: Taxonomy, challenges, open problems and future ideas". In: *Expert Systems with Applications* 233.June, p. 120901. ISSN: 09574174. DOI: 10.1016/j.eswa.2023.120901. arXiv: 2302.12594.

Friendly, Michael (2008). "A Brief History of Data Visualization". In: *Handbook of Data Visualization*. Ed. by Chun-houh Chen, Wolfgang Karl Härdle, and Antony Unwin. Berlin, Germany: Springer, Berlin, Heidelberg, pp. 15–56.

Geng, Liqiang and Howard J. Hamilton (Sept. 2006). "Interestingness measures for data mining: a survey". In: *ACM Comput. Surv.* 38.3, 9–es. ISSN: 0360-0300. DOI: 10.1145/1132960.1132963.

Ghafari, Seyed Mohssen and Christos Tjortjis (2019). "A survey on association rules mining using heuristics". In: *WIREs Data Mining and Knowledge Discovery* 9.4, e1307. DOI: 10.1002/widm.1307.

Grahne, Gosta and Jianfei Zhu (2003). "Efficiently using prefix-trees in mining frequent itemsets." In: *Proc. of the 1st IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, pp. 236–245.

Grbovic, Mihajlo, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp (2015). "E-commerce in Your Inbox: Product Recommendations at Scale". In: *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, pp. 1809–1818. ISBN: 978-1-4503-3664-2. DOI: 10.1145/2783258.2788627. URL: https://doi.org/10.1145/2783258.2788627.

Guide, V. Daniel R. and Jiayi Li (2010). "The potential for cannibalization of new products sales by remanufactured products". In: *Decision Sciences* 41.3, pp. 547–572. ISSN: 00117315. DOI: 10.1111/j.1540-5915.2010.00280.x.

Hahsler, Michael (2016). "Grouping association rules using Lift". In: *Proceedings of the 11th INFORMS Workshop on Data Mining and Decision Analytics*. URL: http://cran.r-project.org/.

— (2023). "ARULESPY: exploring association rules and frequent itemsets in Python". In: arXiv: 2305.15263. URL: http://arxiv.org/abs/2305.15263.

— (2024). "A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules". In: URL: https://mhahsler.github.io/arules/docs/measures.pdf..

Hahsler, Michael and S Chelluboina (2011). *Visualizing Association Rules: Introduction to the R-extension Package arulesViz*. Tech. rep. February, pp. 1–24. URL: http://www.comp.nus.edu.sg/~zhanghao/project/visualization/%5B2010%5DarulesViz.pdf.

Hahsler, Michael, B. Grun, and K. Hornik (2005). "Arules - A computational environment for mining association rules and frequent itemsets". In: *Journal of Statistical Software* 14.15. DOI: 10.18637/jss.v014.i15.

Hahsler, Michael, Kurt Hornik, and Thomas Reutterer (2006). "Implications of probabilistic data modeling for mining association rules". In: *From Data and Information Analysis to Knowledge Engineering*, pp. 598–605. DOI: 10.1007/3-540-31314-1_73.

Hahsler, Michael and Radoslaw Karpienko (Apr. 2017). "Visualizing association rules in hierarchical groups". In: *Journal of Business Economics* 87.3, pp. 317–335. ISSN: 18618928. DOI: 10.1007/s11573-016-0822-8.

Hájek, P., I. Havel, and M. Chytil (Dec. 1966). "The GUHA method of automatic hypotheses determination". In: *Computing 1966 1:4* 1.4, pp. 293–308. ISSN: 1436-5057. DOI: 10.1007/BF02345483. URL: https://link.springer.com/article/10.1007/BF02345483.

Hamilton, Rebecca W., Debora V. Thompson, Zachary G. Arens, Simon J. Blanchard, Gerald Häubl, P. K. Kannan, Uzma Khan, Donald R. Lehmann, Margaret G. Meloy, Neal J. Roese, and Manoj Thomas (2014). "Consumer substitution

decisions: an integrative framework". In: *Marketing Letters* 25.3, pp. 305–317. ISSN: 09230645, 1573059X. URL: http://www.jstor.org/stable/24571058 (visited on 07/08/2024).

Han, Jiawei, Jian Pei, and Yiwen Yin (2000). "Mining frequent patterns without candidate generation". In: *SIGMOD Record (ACM Special Interest Group on Management of Data)* 29.2, pp. 1–12. ISSN: 01635808. DOI: 10.1145/335191.335372.

Han, Jiawei, Jian Pei, Yiwen Yin, and Runying Mao (2004). "Mining frequent patterns without candidate generation: a frequent-pattern tree approach". In: *Data Mining and Knowledge Discovery* 8.1, pp. 53–87. ISSN: 13845810. DOI: 10.1023/B:DAMI.0000005258.31418.83.

Harris, Charles R., K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant (Sept. 2020). "Array programming with NumPy". In: *Nature* 585.7825, pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

Henike, Tassilo, Martin Kamprath, and Katharina Hölzle (2020). "Effecting, but effective? How business model visualisations unfold cognitive impacts". In: *Long Range Planning* 53.4. ISSN: 18731872. DOI: 10.1016/j.lrp.2019.101925.

Hofmann, Heike, Arno P.J.M. Siebes, and Adalbert F.X. Wilhelm (2000). "Visualizing association rules with interactive Mosaic plots". In: *Proceeding of the sixth ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 227–235. ISBN: 1581132336. DOI: 10.1145/347090.347133.

Hu, Y. (2006). "Efficient, High-Quality Force-Directed Graph Drawing". In: *Mathematica journal*. ISSN: 1047-5974.

Huang, Weidong, Peter Eades, and Seok Hee Hong (Sept. 2009). "Measuring effectiveness of graph visualizations: A cognitive load perspective". In: *Information Visualization* 8.3, pp. 139–152. ISSN: 14738716. DOI: 10.1057/ivs.2009.10. URL: https://dl.acm.org/doi/abs/10.1057/ivs.2009.10.

Huang, Yunhui, Zhijie Lin, and Lu Yang (2021). "Complements are warm and substitutes are competent: the effect of recommendation type on focal product evaluation". In: *Internet Research* 010414370113. ISSN: 10662243. DOI: 10.1108/INTR-09-2020-0510.

Jentner, Wolfgang and Daniel A. Keim (2019). "Visualization and visual analytic techniques for patterns". In: *High-Utility Pattern Mining: Theory, Algorithms and Applications*. Springer International Publishing, pp. 303–337. ISBN: 978-3-030-04921-8. DOI: 10.1007/978-3-030-04921-8_12.

Jin, Bowen, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li (2020). "Multi-behavior recommendation with graph convolutional networks". In: *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 659–668. DOI: 10.1145/3397271.3401072. arXiv: arXiv:2005.03475v1.

Kaushik, Minakshi, Rahul Sharma, Sijo Arakkal Peious, Mahtab Shahin, Sadok Ben Yahia, and Dirk Draheim (2021). "A systematic assessment of numerical association rule mining methods". In: *SN Computer Science* 2.5, p. 348.

Klemettinen, Mika, Heikki Mannila, Pirjo Ronkainen, Hannu Toivonen, and A. Inkeri Verkamo (1994). "Finding interesting rules from large sets of discovered association rules". In: pp. 401–407. DOI: 10.1145/191246.191314.

Koh, Yun Sing, Russel Pears, and Wai Yeap (2010). "Valency based weighted association rule mining". In: vol. 6118 LNAI. PART 1, pp. 274–285. ISBN: 3642136567. DOI: 10.1007/978-3-642-13657-3_31.

Kontonasios, Kleanthis Nikolaos, Eirini Spyropoulou, and Tijl De Bie (2012). "Knowledge discovery interestingness measures based on unexpectedness". In: *Wiley In-*

terdisciplinary Reviews: Data Mining and Knowledge Discovery 2.5, pp. 386–399. ISSN: 19424795. DOI: 10.1002/widm.1063.

Koren, Yehuda, Robert Bell, and Chris Volinsky (2009). "Matrix Factorization Techniques for Recommender Systems". In: *Computer* 42.8, pp. 30–37. DOI: 10.1109/MC.2009.263. URL: https://doi.org/10.1109/MC.2009.263.

Kotler, Philip and Kevin Lane Keller (2016). *Marketing Management*. 15th. Upper Saddle River, NJ: Pearson Education. ISBN: 9780134236933.

Kryszkiewicz, Marzena (2002). "Concise representations of association rules". In: *Pattern Detection and Discovery*. Ed. by David J. Hand, Niall M. Adams, and Richard J. Bolton. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 92–109. ISBN: 978-3-540-45728-2.

Kudriavtsev, Mikhail, Andrew McCarren, H. Lee, and Marija Bezbradica (2024). "Efficient Visualization of Association Rule Mining Using the Trie of Rules". In: *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1: KDIR*, pp. 72–80. ISBN: 978-989-758-716-0.

Kudriavtsev, Mikhail, Vuong M Ngo, Mark Roantree, Marija Bezbradica, and Andrew McCarren (2024). "Exploring the trie of rules: a fast data structure for the representation of association rules". In: *Journal of Intelligent Information Systems*, pp. 1–21.

Kumar, Biresh, Sharmistha Roy, Anurag Sinha, Celestine Iwendi, and L'ubomíra Strážovská (2022). "E-commerce website usability analysis using the association rule mining and machine learning algorithm". In: *Mathematics* 11.1, p. 25.

Lajus, Jonathan, Luis Galárraga, and Fabian Suchanek (2020). "Fast and exact rule mining with AMIE 3". In: *The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17*. Springer, pp. 36–52.

Leung, Carson Kai Sang and Christopher L. Carmichael (2009). "FpViz: A visualizer for frequent pattern mining". In: *Proceedings of the ACM SIGKDD Workshop on*

Visual Analytics and Knowledge Discovery, VAKD '09 January 2009, pp. 30–39. DOI: 10.1145/1562849.1562853.

Leung, Carson Kai-Sang (2009). "Constraint-based association rule mining". In: *Encyclopedia of Data Warehousing and Mining, Second Edition*. IGI Global, pp. 307–312.

Lewin, Kurt (1936). "A dynamic theory of personality: Selected papers". In: *The Journal of Nervous and Mental Disease* 84.5, pp. 612–613.

Li, Wenmin, Jiawei Han, and Jian Pei (2001). "CMAR: Accurate and efficient classification based on multiple class-association rules". In: *Proceedings 2001 IEEE International Conference on Data Mining*, pp. 369–376.

Li, Yuefeng and Jingtong Wu (2014). "Interpretation of association rules in multi-tier structures". In: *International Journal of Approximate Reasoning* 55.6, pp. 1439–1457. ISSN: 0888613X. DOI: 10.1016/j.ijar.2014.04.015.

Lin, Yen Liang, Son Tran, and Larry S. Davis (2020). "Fashion outfit complementary item retrieval". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3308–3316. ISSN: 10636919. DOI: 10.1109/CVPR42600.2020.00337. arXiv: 1912.08967.

Liu, Bing, Wynne Hsu, and Yiming Ma (1999). "Mining association rules with multiple minimum supports". In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 337–341.

Liu, Xiaobing, Kun Zhai, and Witold Pedrycz (2012). "An improved association rules mining method". In: *Expert Systems with Applications* 39.1, pp. 1362–1374. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2011.08.018.

Liu, Xiaotong and Han-Wei Shen (2016). "Association analysis for visual exploration of multivariate scientific data sets". In: *IEEE Transactions on Visualization and Computer Graphics* 22.1, pp. 955–964. DOI: 10.1109/TVCG.2015.2467431.

Liu, Yimin, Guozhu Liao, Subhankar Choudhury, and Qu Wu (2012). "Mining high utility itemsets without candidate generation". In: *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 55–64.

Luxenburger, Michael (1991). "Implications partielles dans un contexte". fr. In: *Mathématiques informatique et sciences humaines* 113, pp. 35–55. URL: http://www.numdam.org/item/MSH_1991__113__35_0/.

Máša, Petr and Jan Rauch (2024). "A novel algorithm for mining couples of enhanced association rules based on the number of output couples and its application". In: *Journal of Intelligent Information Systems* 62.2, pp. 431–458. ISSN: 15737675. DOI: 10.1007/s10844-023-00820-1.

McAuley, Julian, Rahul Pandey, and Jure Leskovec (2015). "Inferring networks of substitutable and complementary products". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 2015-Augus, pp. 785–794. DOI: 10.1145/2783258.2783381. arXiv: 1506.08839.

McKinney, Wes (2010). "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

Menin, Aline, Lucie Cadorel, Andrea Tettamanzi, Alain Giboin, Fabien Gandon, and Marco Winckler (July 2021). "ARViz: Interactive Visualization of Association Rules for RDF Data Exploration". In: *Proceedings of the international conference on information visualisation*. Vol. 2021-July. IEEE, pp. 13–20. ISBN: 9781665438278. DOI: 10.1109/IV53921.2021.00013.

Microsoft (2019). *AdventureWorks Sample Database*. https://www.kaggle.com/datasets/cnezhmar/sporting-goods-store. Accessed: 2024-12-02.

Misman, M. Azah and Nurhayati Kamaruddin (2020). "Association rule mining in bioinformatics data analysis: A survey". In: *Malaysian Journal of Fundamental and Applied Sciences* 16.2, pp. 147–157.

Moahmmed, Salahadin A., Mohamed A. Alasow, and El-Sayed M. El-Alfy (2021). "Clustering of Association Rules for Big Datasets using Hadoop MapReduce". In: *International Journal of Advanced Computer Science and Applications* 12.3. DOI: 10.14569/IJACSA.2021.0120364.

Moslehi, Fateme, Abdorrahman Haeri, and Francisco Martínez-Álvarez (2020). "A novel hybrid GA–PSO framework for mining quantitative association rules". In: *soft computing* 24.6, pp. 4645–4666.

Naulaerts, Stefan, Pieter Meysman, Wout Bittremieux, Trung Nghia Vu, Wim Vanden Berghe, Bart Goethals, and Kris Laukens (2015). "A primer to frequent itemset mining for bioinformatics". In: *Briefings in bioinformatics* 16.2, pp. 216–231.

Neo4j (2012). *Neo4j - The World's Leading Graph Database*. URL: `http://neo4j.org/`.

Newbold, Paul, William L. Carlson, and Betty Thorne (2013). *Statistics for Business and Economics*. 8th. Pearson Education.

Ng, Raymond T, Laks V S Lakshmanan, Jiawei Han, and Alex Pang (1998). "Exploratory mining and pruning optimizations of constrained association rules". In: *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pp. 13–24.

Ong, Kian-huat, Kok-leong Ong, Wee-keong Ng, and Ee-peng Lim (2002). "CrystalClear: Active Visualization of Association Rules". In: *ICDM'02 International Workshop on Active Mining AM2002* February, pp. 1–6.

Padua, Renan De, Solange Oliveira Rezende, and Veronica Oliveira De Carvalho (2014). "Post-processing association rules using networks and transductive learning". In: *Proceedings - 2014 13th International Conference on Machine Learning and Applications, ICMLA 2014*, pp. 318–323. DOI: `10.1109/ICMLA.2014.57`.

Pei, Jian, Jiawei Han, Hongjun Lu, Shojiro Nishio, Shiwei Tang, and Dongqing Yang (2001). "H-mine: Hyper-structure mining of frequent patterns in large databases". In: *proceedings 2001 IEEE international conference on data mining*. IEEE, pp. 441–448.

Pellegrini, Chantal, Ege Özsoy, Monika Wintergerst, and Georg Groh (2021). "Exploiting food embeddings for ingredient substitution". In: *HEALTHINF 2021 - 14th International Conference on Health Informatics; Part of the 14th Interna-*

tional Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2021 5.Biostec, pp. 67–77. DOI: 10.5220/0010202000670077.

Peng, Joanne, Kuk Lee, and Gary Ingersoll (Sept. 2002). "An Introduction to Logistic Regression Analysis and Reporting". In: *Journal of Educational Research - J EDUC RES* 96, pp. 3–14. DOI: 10.1080/00220670209598786.

Rainsford, Chris P. and John F. Roddick (2000). "Visualisation of temporal interval association rules". In: *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*. Vol. 1983, pp. 91–96. ISBN: 3540414509. DOI: 10.1007/3-540-44491-2{_}14.

Rajak, Akash and Mahendra Kumar Gupta (2008). "Association Rule Mining: Applications in Various Areas". In: *International Conference on Data Management* January 2008, pp. 3–7. URL: https://pdfs.semanticscholar.org/e5db/a964a791763d7cd9c9d8979f2c00604e7b9a.pdf.

Reddy, Chandan K., Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian (2022). *Shopping Queries Dataset: A Large-Scale ESCI Benchmark for Improving Product Search*. Vol. 1. 1. Association for Computing Machinery. arXiv: 2206.06588. URL: http://arxiv.org/abs/2206.06588.

Ricci, Francesco, Lior Rokach, Bracha Shapira, and Paul B. Kantor (2011). *Recommender Systems Handbook*. Springer.

Ruiz, Francisco J.R., Susan Athey, and David M. Blei (2020). "Shopper: A probabilistic model of consumer choice with substitutes and complements". In: *Annals of Applied Statistics* 14.1, pp. 1–27. ISSN: 19417330. DOI: 10.1214/19-AOAS1265. arXiv: 1711.03560.

Sarwar, Badrul, George Karypis, Joseph Konstan, and John Riedl (2001a). "Item-based collaborative filtering recommendation algorithms". In: *Proceedings of the 10th international conference on World Wide Web*. ACM, pp. 285–295.

— (2001b). “Item-based collaborative filtering recommendation algorithms”. In: *Proceedings of the 10th international conference on World Wide Web*. ACM, pp. 285–295.

Savasere, Ashok, Edward Omiecinski, and Shamkant Navathe (1998). “Mining for strong negative associations in a large database of customer transactions”. In: *Proceedings - International Conference on Data Engineering*, pp. 494–502. DOI: 10.1109/icde.1998.655812.

Sethi, Rupal and B. Shekar (2018). “Mining substitution rules: A knowledge-based approach using dynamic ontologies”. In: *ICAART 2018 - Proceedings of the 10th International Conference on Agents and Artificial Intelligence* 2.Icaart, pp. 73–84. DOI: 10.5220/0006577400730084.

Shabtay, Lior, Philippe Fournier-Viger, Rami Yaari, and Itai Dattner (2021). “A guided FP-Growth algorithm for mining multitude-targeted item-sets and class association rules in imbalanced data”. In: *Information Sciences* 553, pp. 353–375. ISSN: 0020-0255. DOI: 10.1016/j.ins.2020.10.020.

Shaukat Dar, Kamran and Sana Zaheer (Nov. 2015). “Association rule mining: an application perspective”. In: *International Journal of Computer Science and Innovation* 1, pp. 29–38.

Shrout, Patrick E. and Joseph L. Fleiss (1979). “Intraclass correlations: uses in assessing rater reliability”. In: *Psychological Bulletin* 86.2, pp. 420–428.

Shukla, Paurav (July 2009). “Impact of contextual factors, brand loyalty and brand switching on purchase decisions”. In: *Journal of Consumer Marketing* 26, pp. 348–357. DOI: 10.1108/07363760910976600.

Srikant, Ramakrishnan and Rakesh Agrawal (1996). “Mining quantitative association rules in large relational tables”. In: *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pp. 1–12.

Stancin, I. and A. Jovic (2019). “An overview and comparison of free Python libraries for data mining and big data analysis”. In: *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectron-*

ics, MIPRO 2019 - Proceedings, pp. 977–982. DOI: 10.23919/MIPRO.2019.8757088.

Tan, Pang-Ning, Vipin Kumar, and Jaideep Srivastava (2004). "Selecting the right objective measure for association analysis". In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM New York, NY, USA, pp. 32–41.

Tan, Pang-Ning, Michael Steinbach, Anuj Karpatne, and Vipin Kumar (2019). *Introduction to Data Mining*. Pearson.

The Pandas development team (Apr. 2024). *Pandas-dev/pandas: Pandas*. Version v2.2.2. DOI: 10.5281/zenodo.10957263.

Tian, Yu, Sebastian Lautz, Alisdair O.G. Wallis, and Renaud Lambiotte (2021). "Extracting complements and substitutes from sales data: a network perspective". In: *EPJ Data Science* 10.1. ISSN: 21931127. DOI: 10.1140/epjds/s13688-021-00297-4. arXiv: 2103.02042.

Tkachuk, Sergiy, Anna Wroblewska, Jacek Dabrowski, and Szymon Lukasik (2022). "Identifying Substitute and Complementary Products for Assortment Optimization with Cleora Embeddings". In: *Proceedings of the International Joint Conference on Neural Networks* 2022-July, pp. 1–10. DOI: 10.1109/IJCNN55064.2022.9892361. arXiv: 2208.06262.

Toivonen, Hannu., Mika. Klemettinen, Pirjo. Ronkainen, Kimmo. Hätönen, and Heikki. Mannila (1995). *Pruning and Grouping Discovered Association Rules*. Tech. rep., pp. 47–52.

Varu, Rakshit, Leonardo Christino, and Fernando V. Paulovich (2022). "ARMatrix: An Interactive Item-to-Rule Matrix for Association Rules Visual Analytics". In: *Electronics (Switzerland)* 11.9. ISSN: 20799292. DOI: 10.3390/electronics11091344.

Vasile, Flavian, Elena Smirnova, and Alexis Conneau (2016). "Meta-learning for context-aware recommender systems". In: *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 191–198.

Vu, L and G Alaghband (2011). "A fast algorithm combining FP-tree and TID-list for frequent pattern mining". In: *Proceedings of information and knowledge engineering*, pp. 472–477.

Wang, Jianling, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee (2020). "Next-item Recommendation with Sequential Hypergraphs". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '20. Virtual Event, China: Association for Computing Machinery, pp. 1101–1110. ISBN: 9781450380164. DOI: 10.1145/3397271.3401133. URL: https://doi.org/10.1145/3397271.3401133.

Ware, Colin (2012). *Information Visualization: Perception for Design*. 3rd. Burlington, MA, USA: Morgan Kaufmann.

Wu, Zhiang, Changsheng Li, Jie Cao, and Yong Ge (2020). "On scalability of association-rule-based recommendation: A unified distributed-computing framework". In: *ACM Transactions on the Web (TWEB)* 14.3, pp. 1–21.

Xue, Xiaonan, Stephen J Gange, Ye Zhong, Robert D Burk, Howard Minkoff, L Stewart Massad, D Heather Watts, Mark H Kuniholm, Kathryn Anastos, Alexandra M Levine, *et al.* (2010). "Marginal and mixed-effects models in the analysis of human papillomavirus natural history data". In: *Cancer epidemiology, biomarkers & prevention* 19.1, pp. 159–169.

Yazgana, Pinar and Ali Osman Kusakci (2016). "A literature survey on association rule mining algorithms". In: *Southeast Europe Journal of Soft Computing* 5.1, pp. 5–14. ISSN: 2233-1859. DOI: 10.21533/scjournal.v5i1.102.

Yen, Show-Jane and A.L.P. Chen (2001). "A graph-based approach for discovering various types of association rules". In: *IEEE Transactions on Knowledge and Data Engineering* 13.5, pp. 839–845. DOI: 10.1109/69.956106.

Yoghourdjian, Vahan, Yalong Yang, Tim Dwyer, Lee Lawrence, Michael Wybrow, and Kim Marriott (2021). "Scalability of Network Visualisation from a Cognitive Load Perspective". In: *IEEE Transactions on Visualization and Com-*

puter Graphics 27.2, pp. 1677–1687. ISSN: 19410506. DOI: 10.1109/TVCG.2020.3030459. arXiv: 2008.07944.

Yu, Junliang, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang (2024). "Self-Supervised Learning for Recommender Systems: A Survey". In: *IEEE Transactions on Knowledge and Data Engineering* 36.1, pp. 335–355. ISSN: 15582191. DOI: 10.1109/TKDE.2023.3282907. arXiv: 2203.15876.

Zaki, Mohammed J., Srinivasan Parthasarathy, Mitsunori Ogihara, and Wei Li (1997). "Parallel algorithms for discovery of association rules". In: *Data Mining and Knowledge Discovery* 1.4, pp. 343–373. ISSN: 13845810. DOI: 10.1023/A:1009773317876. URL: www.aaai.org.

Zhang, Mingyue and Jesse Bockstedt (2020). "Complements and substitutes in online product recommendations: The differential effects on consumers' willingness to pay". In: *Information and Management* 57.6, p. 103341. ISSN: 03787206. DOI: 10.1016/j.im.2020.103341. URL: https://doi.org/10.1016/j.im.2020.103341.

Zhang, Shichao, Xindong Wu, Chengqi Zhang, and Jingli Lu (2008). "Computing the minimum-support for mining frequent patterns". In: *Knowledge and Information Systems* 15.2, pp. 233–257. ISSN: 02193116. DOI: 10.1007/s10115-007-0081-7.

Zhao, Yuan Yuan and He Jiang (2009). "Research of mining positive and negative weighted association rules based on chi-squared analysis". In: *2009 2nd International Conference on Information and Computing Science, ICIC 2009* 1, pp. 344–347. DOI: 10.1109/ICIC.2009.94.

Zheng, Hui, Jing He, Qing Liu, Jianhua Li, Guangli Huang, and Peng Li (2023). "Multi-objective optimisation based fuzzy association rule mining method". In: *World Wide Web* 26.3, pp. 1055–1072.

Appendix A

Algorithms

Algorithm 1 Rule Object Definition with Constructor Procedure

```
1: class Rule:
2:     attributes:
3:     - antecedent: list[str]              ▷ Sequence of items in the rule antecedent
4:     - consequent: list[str]              ▷ Sequence of items in the rule consequent
5:     - metrics: dictionary[str, float]    ▷ Dictionary to store metrics
6:
7:     procedure Rule():
8:         self.antecedent ← []
9:         self.consequent ← []
10:        self.metrics ← { 'Support': None, 'Confidence': None }
```

Algorithm 2 Creating a Trie from Ruleset

Description: This function creates a Trie data structure from a given set of rules and their corresponding frequent sequences.

```
1: function create_trie(rule_set, frequent_sequences) → Trie   ▷ rule_set is a list of rule objects, frequent_sequences is a dictionary with sequences of items as keys and their associated Support values
2:     trie ← Trie()                                ▷ Initialize an empty trie
3:     for each rule in rule_set do
4:         trie.insert_rule(rule, frequent_sequences)   ▷ Insert the rule into the trie
5:     end for
6:     for each child in trie.root.children do
7:         trie.extend_node(child, frequent_sequences)   ▷ Calculate metrics starting the recursive traversal from the root node children, using the frequent sequences
8:     end for
9:     return trie
10: end function
```

Algorithm 3 Trie Structure with Constructor Procedure

```
1: class Trie:
2:     attributes:
3:     - root: TrieNode                          ▷ Represents the root node of the trie
4:     - item_frequency: dictionary[str, int]    ▷ Frequency of individual items
5:     methods:
6:     - insert_rule(self, rule): void           ▷ Inserts a new rule into the trie
7:     - search_rule(self, rule): Rule or None   ▷ Searches for a rule in the trie
8:     - extend_node(self, node, frequent_sequences): void   ▷ Extend nodes with
   ARM metrics
9:
10:    procedure Trie():
11:        self.root ← TrieNode(predecessor=None, value=None)
12:        self.root.metrics ← { 'Support': 1.0, 'Confidence': 1.0 }
13:        self.item_frequency ← {}
```

Algorithm 4 Trie Node Structure with Constructor Procedure

```
1: class TrieNode:
2:     attributes:
3:     - children: dictionary[str, TrieNode]   ▷ dictionary of child nodes (item
   name : node)
4:     - predecessor: TrieNode                 ▷ The parent node
5:     - value: str                            ▷ Name of the item associated with the node
6:     - metrics: dictionary[str, float]       ▷ Dictionary to store metrics
7:
8:     procedure TrieNode(predecessor, value):
9:         self.children ← {}                  ▷ Initially, no children; dictionary is empty
10:        self.predecessor ← predecessor
11:        self.value ← value
12:        self.metrics ← { 'Support': None, 'Confidence': None }
```

Algorithm 5 Inserting a Rule in the Trie - Trie Class Method

Description: This procedure inserts a rule into a Trie object after sorting the antecedent and consequent items based on their frequency.

```
1: procedure TRIE.INSERT_RULE(self, rule, frequent_sequences)
2:     current_node ← self.root
3:                                                  ▷ Update item frequencies
4:     for each item in rule.antecedent + rule.consequent do  ▷ Concatenate two lists
5:         if item not in self.item_frequency then
6:             self.item_frequency[item] ← frequent_sequences[item]
7:         end if
8:     end for
9:                         ▷ Sort the antecedent items by their frequency (descending)
10:    sorted_antecedent ← sort(rule.antecedent, λ(x, y) → self.item_frequency[x] > self.item_frequency[y])
11:                        ▷ Sort the consequent items by their frequency (descending)
12:    sorted_consequent ← sort(rule.consequent, λ(x, y) → self.item_frequency[x] > self.item_frequency[y])
13:    rule_itemlist ← sorted_antecedent + sorted_consequent      ▷ Concatenate two lists
14:    for each item in rule_itemlist do
15:        if item not in current_node.children then
16:            current_node.children[item] ← TRIENODE(predecessor = current_node, value = item)
17:        end if
18:        current_node ← current_node.children[item]
19:    end for
20: end procedure
```

Algorithm 6 Extend Trie with ARM Metrics - Trie Class Method

Description: This procedure adds Support and Confidence metrics to the node and its children recursively.

```
1: procedure TRIE.EXTEND_NODE(self, node, frequent_sequences)  ▷ frequent_sequences is a dictionary with sequences of items as keys and their associated Support values
2:     item_list ← []
3:     current_node ← node
4:     while current_node ≠ self.root do
5:         item_list ← item_list + current_node.value  ▷ Append items in the path to the root to the list
6:         current_node ← current_node.predecessor
7:     end while
8:     node.metrics['Support'] ← frequent_sequences[item_list]  ▷ Set Support metric from the frequent sequences
9:                                                  ▷ Calculate Confidence for the node:
10:    node.metrics['Confidence'] ← node.metrics['Support'] / node.predecessor.metrics['Support']
11:    for each child in node.children do
12:        self.extend_node(child, frequent_sequences)  ▷ Recursively traverse child nodes
13:    end for
14: end procedure
```

Algorithm 7 Searching a Rule in the Trie - Trie Class Method

Description: This function searches for a rule in the Trie and returns its metrics if found.

```
1: function TRIE.SEARCH_RULE(self, rule)
2:     current_node ← self.root
3:     sorted_antecedent ← sort(rule.antecedent, λ(x, y) → self.item_frequency[x] > self.item_frequency[y])
4:     sorted_consequent ← sort(rule.consequent, λ(x, y) → self.item_frequency[x] > self.item_frequency[y])
5:     for each item in sorted_antecedent do
6:         if item not in current_node.children then
7:             return None                                   ▷ Rule not found
8:         end if
9:         current_node ← current_node.children[item]
10:    end for
11:    rule_confidence ← 1.0
12:    for each item in sorted_consequent do
13:        if item not in current_node.children then
14:            return None                                   ▷ Rule not found
15:        end if
16:        current_node ← current_node.children[item]
17:        rule_confidence ← rule_confidence × current_node.metrics['Confidence']
18:    end for
19:    rule.metrics ← {'Support': current_node.metrics['Support'], 'Confidence': rule_confidence}
20:    return rule.metrics
21: end function
```

Algorithm 8 Substitute Item Mining with Context

Description: This procedure finds all substitute item pairs along with their respective contexts in the Trie of Rules.

```
1: procedure FIND_SUBSTITUTE_PAIRS_WITH_CONTEXT(node, context, substitutes)
2:     if context is None then context ← []                ▷ Initialize context if None
3:     if substitutes is None then substitutes ← [] ▷ Initialize substitutes if None
4:     if node.value is not None then
5:         context.append(node.value)    ▷ Update context with the current node's value
6:     children_items ← list of children items of node.children
7:     if len(children_items) >1 then
8:         branches ← collect_items_from_branch(node)
9:         for each pair of items item_a and item_b from different branches do
10:            substitutes.append((context, item_a, item_b))  ▷ Store substitute pair with context
11:        end for
12:    end if
13:    for each child in node.children do
14:        FIND_SUBSTITUTE_PAIRS_WITH_CONTEXT(child, context, substitutes)  ▷ Recursively explore child nodes
15:    end for
16:    return substitutes
17: end procedure
```

Algorithm 9 Collecting Items from a Branch

Description: This procedure collects all items in a branch from the current node down to its leaf nodes.

```
1: procedure COLLECT_ITEMS_FROM_BRANCH(node)
2:     items ← []
3:     procedure TRAVERSE(current_node, path)
4:         if current_node.value is not None then
5:             path.append(current_node.value)  ▷ Add current node's value to path
6:         end if
7:         if current_node.children is empty then
8:             items.append(path)                          ▷ Store the path at leaf nodes
9:         else
10:            for each child in current_node.children do
11:                TRAVERSE(child, path)                   ▷ Recursively traverse child nodes
12:            end for
13:        end if
14:    end procedure
15:    TRAVERSE(node, [])
16:    return list of tuples of items from the branch
17: end procedure
```