No d'ordre : 2012-ISAL-0094 Année 2012

THÈSE

présentée devant

L'Institut National des Sciences Appliquées de Lyon

pour obtenir

Le Grade de Docteur

Spécialité

Informatique

École Doctorale : Informatique et Mathématiques

par

Thi Kim Ngan Nguyen

Generalizing Association Rules in N-ary Relations: Application to Dynamic Graph Analysis

Soutenue publiquement le 23 octobre 2012 devant le jury :

Dr. Francesco Bonchi	Yahoo! Research Barcelona, ES	Examinateur
Pr. Jean-François Boulicaut	INSA de Lyon, F	Directeur
Pr. Tu Bao Ho	JAIST, JP	Rapporteur
Pr. Stéphane Lallich	Université Lyon 2, F	Examinateur
Pr. Dominique Laurent	Université de Cergy Pontoise, F	Rapporteur
Dr. Marc Plantevit	Université Lyon 1, F	Examinateur

Thi Kim Ngan NGUYEN: *Generalizing Association Rules in N-ary Relations: Application to Dynamic Graph Analysis,* PhD Thesis, © 2008-2012

SUPERVISOR:
Jean-François Boulicaut

SUPERVISOR:
2008-2012

Résumé

Le calcul de motifs dans de grandes relations binaires a été très étudié. Un succès emblématique concerne la découverte d'ensembles fréquents et leurs post-traitements pour en dériver des règles d'association. Il s'agit de calculer des motifs dans des relations *Objets* $\times$ *Propriétés* qui enregistrent quelles sont les propriétés satisfaites par des objets. En fait, de nombreux jeux de données se présentent naturellement comme des relations n-aires (avec $n > 2$). Par exemple, avec l'ajout de dimensions spatiales et/ou temporelles (lieux et/ou temps où les propriétés sont enregistrées), on peut vouloir travailler sur une relation 4-aire *Objets* $\times$ *Propriétés* $\times$ *Lieux* $\times$ *Temps*. Nous avons généralisé le concept de règle d'association dans un tel contexte multi-dimensionnel, en travaillant non plus sur des matrices booléennes mais sur des tenseurs booléens d'arité arbitraire. Contrairement aux règles usuelles qui n'impliquent que des sous-ensembles d'un seul domaine de la relation, les prémisses et les conclusions de nos règles peuvent impliquer des sous-ensembles arbitraires des domaines retenus. Nous avons conçu des mesures de fréquence et de confiance pour définir la sémantique de telles règles et c'est une contribution significative de cette thèse. Le calcul exhaustif de toutes les règles qui ont des fréquences et confiances sufisantes et l'élimination des règles redondantes ont été étudiés. Nous proposons ensuite d'introduire des disjonctions dans les conclusions des règles. Ceci nécessite de retravailler les définitions des mesures d'intérêt et les questions de redondance. Pour ouvrir un champ d'application original, nous considérons la découverte de règles dans des graphes relationnels dynamiques codés dans des relations n-aires ($n \geq 3$). Une application à l'analyse des usages de vélos dans le système Vélo'v (système de Vélos en libre-service du Grand Lyon) montre quelques usages possibles des règles que nous savons calculer avec nos prototypes logiciels.

Mots clés

Motifs, Règle descriptive, Non-redondance, Données multidimensionnelles, Fouille sous contraintes, Tenseur booléen, Graphes dynamiques.

Abstract

Pattern discovery in large binary relations has been extensively studied. Typically, it needs to compute patterns that hold in relations $Objects \times Properties$ that denote whether given properties are satisfied or not by given objects. An emblematic success in this area concerns frequent itemset mining and its post-processing that derives association rules. It is however clear that many datasets correspond to n-ary relations where $n > 2$. For example, adding spatial and/or temporal dimensions (location and/or time when the properties are satisfied by the objects) leads to the 4-ary relation $Objects \times Properties \times Places \times Times$. Therefore, we study the generalization of association rule mining within arbitrary n-ary relations: the datasets are now Boolean tensors and not only Boolean matrices. Unlike standard rules that involve subsets of only one domain of the relation, in our setting, the head and the body of a rule can include arbitrary subsets of some selected domains. A significant contribution of this thesis concerns the design of interestingness measures for such generalized rules: besides a frequency measures, two different views on rule confidence are considered. The concept of non-redundant rules and the efficient extraction of the non-redundant rules satisfying the minimal frequency and minimal confidence constraints are also studied. To increase the subjective interestingness of rules, we then introduce disjunctions in their heads. It requires to redefine the interestingness measures again and to revisit the redundancy issues. Finally, we apply our new rule discovery techniques to dynamic relational graph analysis. Such graphs can be encoded into n-ary relations ($n \geq 3$). Our use case concerns bicycle renting in the Vélo'v system (self-service bicycle renting in Lyon). It illustrates the added-value of some rules that can be computed thanks to our software prototypes.

Keywords

Pattern, Descriptive rule, Non redundancy, Mutidimensional data, Constraint-based mining, Boolean tensor, Dynamic graph.

Publications

International journals

[CBNB12] Loïc Cerf, Jérémy Besson, Kim-Ngan T. Nguyen, and Jean-François Boulicaut. Closed and noise-tolerant patterns in n-ary relations. *Data Min. Knowl. Discov.*, page 42 p., August 2012. DOI 10.1007/s10618-012-0284-8.

[NCPB11] Kim-Ngan T. Nguyen, Loïc Cerf, Marc Plantevit, and Jean-François Boulicaut. Mining descriptive rules in dynamic graphs. *Intelligent Data Analysis*, 2011. In Press. To appear in Vol. 17.

International conferences

[NCPB10] Kim-Ngan T. Nguyen, Loïc Cerf, Marc Plantevit, and Jean-François Boulicaut. Discovering inter-dimensional rules in dynamic graphs. In *Workshop on Dynamic Networks and Knowledge Discovery co-located with ECML PKDD 2010*, DyNaK '10, pages 5–16. CEUR Workshop Proceedings, 2010.

[NCPB11] Kim-Ngan T. Nguyen, Loïc Cerf, Marc Plantevit, and Jean-François Boulicaut. Multidimensional association rules in boolean tensors. In *Proceedings of the 11th SIAM International Conference on Data Mining*, SDM '11, pages 570–581. SIAM / Omnipress, 2011.

[NPB12] Kim-Ngan T. Nguyen, Marc Plantevit, and Jean-François Boulicaut. Mining disjunctive rules in dynamic graphs. In *Proceedings of the 9th IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future*, RIVF '12, pages 1–6. IEEE, 2012.

National conferences

[NCB10] Kim-Ngan T. Nguyen, Loïc Cerf, and Jean-François Boulicaut. Sémantiques et calculs de règles descriptives dans une relation n-aire. In *Actes des 26èmes journées Bases de Données Avancées*, BDA '10, pages 1–20, 2010.

Acknowledgments

This thesis was carried out in the LIRIS lab (Laboratoire d'InfoRmatique en Image et Systèmes d'information UMR CNRS 5205). This work was funded by a Vietnam government scholarship and by the ANR Bingo2 project. I want to thank all those who have contributed directly or indirectly to the completion of this work.

I want to thank my teachers in Hanoi, Cam Ha Ho, Tho Hoan Pham, The Loc Nguyen and Thi Tinh Nguyen, and Thanh Thuy Nguyen, for encouraging me to pursue my studies at INSA de Lyon (National Institute of Applied Sciences in Lyon).

I express my deep gratitude to my advisor Jean-François Boulicaut who welcomed me in 2008 for starting my studies in his research group. He has raised my interest in the subject area of this thesis, guided and supported me in every step of this thesis along these years. He has supervised my progress with patient guidance even in the worst periods. I thank Jean-François for invaluable lessons that he taught me about research. It is my great fortune to have pursued my Ph.D. studies under his guidance.

I would like to say a special thank to Loïc Cerf and Marc Plantevit who coauthored the publications. Thanks to many suggestions and stimulating discussions, they contributed to train me to computer science and rigourous approaches in data mining.

I would like to express my gratefulness to Prof. Dominique Laurent and Prof. Tu Bao Ho for agreeing to be the thesis reviewers. I want to say a sincere thank to them for their time and thoughtful comments. Also, I would like to express my gratefulness to Prof. Stéphane Lallich and Dr. Francesco Bonchi for agreeing to be the other members of the jury.

I woul like to say a sincere thank to my French teachers in Lyon, Dominique Marie Sert and Ferret Christiane, for their support and great enthusiasm not only in exercises but also in life.

I want to cheerfully thank every member of the group for his/her presence, sharing and help such that I arrived where I am now. Since my arrival in Lyon in 2008, the Turing/Combining/DM2L group became my "second home" during the time

I studied at LIRIS. I would like to offer a special thank to Pierre-Nicolas Mougel and Élise Desmier whom I shared the office for a joyful and supportive daily presence.

My life as a PhD student would not have been enjoyed without my friends in Lyon. I thank Anh Phuong Ta and Viet Hung Nguyen for their help when I arrived in Lyon. I think of Thi Loan Bui, Thi Thuy Nga Duong and Sim Dang, thank to them for daily sharing and support.

Most of all, my gratitude goes to my family. My husband, his love and support is priceless. My parents and my parents-in-law for their boundless love and tremendous sacrifices to ensure that I have a good life. My sisters took care of my parents when I was away from home. I am in memory of my maternal grandparents who always believed in me and gave me words of encouragement. This work would not have been possible without them.

Contents

List of Notations

$\mathcal{O}$	A set of objects
$\mathcal{P}$	A set of items or a set of properties
$\mathcal{B}$	A binary relation over $\mathcal{O} \times \mathcal{P}$
$\mathcal{D}$	A set of n domains, $\mathcal{D} = \{D^1, D^2, ..., D^n\}$
$\mathcal{R}$	A n-ary relation over $D^1 \times D^2 \times ... \times D^n$
2^X	The set of all subsets of set X
$\|X\|$	The cardinality of set X
$\mathbb{R}_+$	Set of all positive real numbers
$\mathbb{N}$	Set of all integers
FIM	frequent itemset mining
w.r.t.	with regard to
resp.	respectively

List of Figures

Introduction

Data mining research has been motivated by the need for new computational methods that support Knowledge Discovery from large Datasets (KDD). At first, early in the 90's, methods from statistics, machine learning, and databases have been used before being revisited (for instance with respect to scalability issues). Today, data mining appears as a mature scientific domain with well-established series of conferences (e.g., ACM SIGKDD, IEEE ICDM, SIAM DM, ECML/PKDD) and quite good textbooks (see, e.g., [102]). The data mining researchers often address the so-called unsupervized methods whose goals are to describe, to summarize, and to suggest relevant hypothesis thanks to data analysis. Such methods enable to explicit relationships and properties which are hidden in the data and can be used afterhand to enhance knowledge discovery and decision support in many application domains.

Many popular data mining tasks can be formalized according to the simple model proposed by Mannila and Toivonen [73]. They assume that we often look for all potentially interesting patterns from a pattern language $\mathcal{L}$ in a given database Db and this is expressed as the computation of $\mathcal{TH}(Db, \mathcal{L}, \mathcal{C}) = \{\rho \in \mathcal{L} \mid \mathcal{C}(\rho, Db) \text{ is true}\}$ where the constraint $\mathcal{C}$ specifies pattern interestingness.

Once declaratively specified, one challenge concerns the correct and complete computation of such collections. When it is not possible, heuristics may be used that enable to look for good approximations of the solution (e.g., this is the case when performing clustering tasks). Considering a KDD process as a sequence of queries over the data combined with such computations that are also called *inductive query* evaluations is one promising direction of research for supporting typical interactive and iterative real-life KDD processes. Constraints play an important role here. Not only user-defined constraints enable to specify both objective and subjective interestingness, but also the constraints can be exploited to achieve computational feasability. Several books on inductive databases and constraint-based mining started to explore this long-term perspective on the KDD field [74, 20, 39].

This thesis concerns *pattern languages* that are set patterns and descriptive rules

that hold within arbitrary n-ary relations (Db denotes a relation over n dimensions). It means that we had to design pattern languages (e.g., the languages of n-sets or that of multidimensional rules) and primitive constraints (e.g., constraints that enforce thresholds on interestingness measures like frequency and confidence). Before introducing the contribution with more details, let us first discuss our context.

Context

The "Data Mining" research group in LIRIS UMR 5205 has an expertise on many instances of such a simple though generic constraint-based data mining setting. Among others and like many other data mining groups in the last two decades, its members have contributed to pattern discovery from large binary relations. This is also known as transactional or 0/1 data mining. Let us first emphasize a couple of milestones about binary relation mining expertise in this group.

- Studying the so-called frequent itemset mining problem [2] in dense datasets instead of sparse ones has given rise to nice results concerning various condensed representations of frequent sets like, among others, closed and δ-free itemsets (Ph. D. thesis A. Bykowski [28], 2002) ;
- On the same pattern language but also on the so-called standard association rules [2], optimizing data mining algorithms when considering that $\mathcal{C}$ is a conjunction of monotone and anti-monotone primitive constraints has been studied (Ph. D. thesis B. Jeudy [60], 2002) ;
- Considering Formal Concept Analysis [41] and closed pattern mining in binary relations, efficient though generic algorithms that compute complete collections of formal concepts that satisfy user-defined constraints have been designed. For instance, DMiner enables to compute formal concepts that have both a large intent and a large extent in large relations. Fault-tolerance can be expressed by means of primitive constraints as well (Ph. D. thesis J. Besson [14], 2005) ;
- Looking for fault-tolerant patterns by generalizing formal concepts and thus closed sets has been studied further: heuristic methods have been considered (for instance, clustering of formal concepts) but also exhaustive ones. Moreover, it has been shown that co-clustering can be implemented as a post-processing over collections of formal concepts (Ph. D. thesis R. Pensa [85], 2006).

In 2008, the group started to investigate the systematic extension of its methods towards arbitrary n-ary relation mining ($n \geq 2$). L. Cerf has defended his Ph. D thesis on closed pattern discovery from such relations in 2010 [31]. Thanks to the Data-Peeler algorithm [32, 33] and its fault-tolerant extension Fenster [CBNB12], given a n-ary relation, we know how to compute complete collections of (possibly error-tolerant) closed patterns that satisfy given piecewise (anti)-monotone constraints. This new class of constraints generalizes both monotone and anti-monotone constraints.

Our research topic the last few years and thus the core contribution of this thesis has been to generalize descriptive rule mining (more precisely the popular association rule mining methods that have been extensively studied in binary relations) within a n-ary relation setting. While standard association rule mining took the most from the research on closed set computation in binary relations (e.g., to tackle redundancy issues), our idea was that a clever generalization of such rules in n-ary relations may be based on closed pattern discovery as well. We now provide some details and we introduce the basic terminology before discussing the contribution.

From binary relations ...

Association rule mining was first introduced in [2] for basket data analysis, i. e., $Customers \times Products$ binary relation mining (each couple records that a given product has been bought by a given customer). Its goal is to explicit a priori interesting co-occurrences of purchases. For example, assume that in some basket data we have an association rule $\{wine, cheese\} \rightarrow \{grape, bread\}$ with a 2% frequency and 80% confidence. First, it means that 2% of the customers buy *wine*, *cheese*, *grape*, and *bread* together. Then, it tells that 80% of the customers who have been buying *wine* and *cheese* have bought *grape* and *bread* as well. Discovering the association rules that satisfy a minimal frequency constraint and a minimal confidence constraint thanks to user-defined thresholds enable to identify sets of products that tend to be bought together. As a result, computed rules may be used to plan marketing or advertising strategies, to support the design of catalogs or store layouts. Association rules have been widely used for basket data analysis but also for mining large binary relations that record whether some objects satisfy or not some boolean properties: in basket data, a property expresses that a given product (also called item) belongs or not to the transaction by a given customer (i.e., an object).

In Chapter 1, we formalize association rule mining in binary relations and we discuss the main directions of research that have been considered since the definition of the task in 1993 [2]. Given a binary relation $\mathcal{B}$ on two domains $\mathcal{O}$ (set of objects) and $\mathcal{P}$ (set of properties), i. e., $\mathcal{B} \subseteq \mathcal{O} \times \mathcal{P}$, the goal of association rule mining is to find out patterns of the form $X \rightarrow Y$ where X, Y are subsets of $\mathcal{P}$, i.e., one of the two domains. X is called the body of the association rule and Y is called its head. X and Y are sets that denotes conjunctions of properties: for instance, when *wine* and *cheese* are seen then *grape* and *bread* are seen as well.

The rule semantics are defined by means of interestingness measures like frequency and confidence. In the standard setting, these measures tell something about the strength of the hypothesis that objects that satisfy the properties in X tend to satisfy the properties in Y. In other terms, we use a *support domain*, here the domain $\mathcal{O}$, to assess pattern objective interestingness.

The standard association rule mining task is computationally hard. Only thresholds values for the interestingness measures are given, defining a priori interesting-

ness as a conjunction of a minimal frequency and a minimal confidence. To compute rules, designing complete methods is obviously hard but this has been quite well understood thanks to about 15 years of intensive research worldwide. Beside the computational feasibility, the major issue of interestingness has been addressed as well: designing better interestingness measures, exploiting new user-defined constraints that support subjective interestingness specification, tackling redundancy issues, etc. Some extensions of standard association rule mining have been studied, for instance, when considering taxonomies over properties or when adding disjunctions and/or negations within the rule bodies and/or heads. However, when limited to binary relations, rules clearly express relationships between elements from one domain only.

... to n-ary relations

Many datasets of interest correspond to relations whose number of dimensions is greater or equal to 3. For example, let us add a time dimension to a relation $Customers \times Products$ such that it becomes a ternary relation $Customers \times Products \times Seasons$: each tuple records that a given customer has bought a given product at a given season. From such a relation, we would like to discover rules like $\{orange\} \rightarrow \{winter\}$. An expected meaning would be that customers often buy oranges during winter. This rule is an implication of elements in two different domains and it cannot be extracted by means of traditional association rule mining algorithms that process binary relations.

Several researchers have considered association rule mining in a multidimensional model. According to the number of dimensions appearing in a rule and their multiple occurrences, the rules can be classified into three types: intra-dimensional, inter-dimensional, and hybrid rules. A rule whose elements belong to only one dimension is called an intra-dimensional rule. The standard association rules in binary relations are a special case of intra-dimensional rules. Schmitz et al. proposed in [92] an intra-dimensional association rule mining technique in ternary relations. Instead of describing co-occurrences of elements of only one domain, inter-dimensional association rules have been proposed (see, e.g., [62, 75]). An inter-dimensional association rule is an implication between elements of a few distinct domains and no dimension is repeated in the rule (i. e., a rule does not have two elements that belong to the same domain). This absence of repetition is a limitation on the expressiveness of the rules. Other researchers have been designing some more or less ad-hoc types of rules, namely hybrid rules, in which the repetition of few dimensions is enabled [53, 38, 104]. All these proposals are discussed in Chapter 2 where methods for descriptive rule mining in n-ary relations and their limits are considered.

Until now, the proposed solutions for generalizing association rule mining in n-ary relations have been always enforcing more or less severe restrictions on the dimensions that can appear or not in their bodies or heads. As a result, it is

not yet possible to discover rules which include arbitrary subsets of some domains. For example, in the 3-ary relation $Customers \times Products \times Seasons$, it is not yet possible to discover rules like $\{melon, orange\} \rightarrow \{summer, autumn\}$ or $\{cherry\} \times \{summer\} \rightarrow (\{apple, pear\}) \vee (\{grape\} \times \{autumn\})$. The expected meaning of the first rule could be that *melon* and *orange* are bought together in both *summer* and *autumn* seasons. The second rule may suggest that, if a customer buys *cherry* in *summer* then he/she can also buy *apple* and *pear*, or he/she tends to buy *cherry* and *grape* in both *summer* and *autumn*. In fact, computing such rules would help to describe and to analyse relationships of elements in the relation $Customers \times Products \times Seasons$.

Therefore, our objective has been to study more expressive generalizations of association rules in arbitrary n-relations. We had to work on declarative aspects like defining the pattern languages and the semantics of rules thanks to primitive constraints and thus new interestingness measures. We also had to design correct and complete algorithms that make the computation of a priori interesting rules feasible in practical situations.

Contributions

Generalizing association rule mining within a n-ary relation ($n \geq 2$) when associations (bodies and heads of rules) can be arbitrary subsets of some domains is surprisingly difficult. The two main subproblems to address are (a) how to define the semantics of rules thanks to constraints, and (b) how to efficiently compute such rules.

Point (a) is about defining the pattern language and objective measures of pattern interestingness. When generalized to n-ary relations, association rules may involve subsets of some domains. In this context, what does it means for a rule to be frequent or to have a high enough confidence? Is it possible to have measures that correspond to the special case of standard measures when $n = 2$ and that are as intuive as possible for analysts? How to generalize other relevancy concepts such as, for example, non redundancy?

Once these declarative issues are understood, Point (b) concerns the design of scalable methods to extract the patterns that satisfy a given conjunction of primitive constraints. When possible, correct and complete algorithms remain preferable: such methods list all solution rules and only them. Performance issues are important: a good algorithm must scale in the number of dimensions, in the size (number of values) of each of these dimensions, and in the number of tuples in the relation that is also the number of true values in the Boolean tensor that represents the relation.

In this dissertation, we investigate two types of descriptive rules that have been called multidimensional association rules and multidimensional disjunctive rules.

Multidimensional association rules

A *multidimensional association rule* is a generalization of an association rule as defined by Agrawal et al. [2] for binary relations. Given an arbitrary n-ary relation, a multidimensional association rule is an implication between two associations where each association can contain subsets of some arbitrary domains. In this context, we propose three objective interestingness measure. Generalizing the *frequency* measure is straightforward: it tells how often a rule is applicable thanks to its support domain. Designing confidence measures is much harder. We propose two generalizations. The *natural confidence* evaluates the probability to observe the head of a rule when its body holds. The *exclusive confidence* evaluates whether the association in the body of a rule "prefers" conjoining with the association in the head to conjoining with other elements. The concept of non-redundant rule having a minimal body and a maximal head (see, e.g., [82]) must be revisited as well.

For example, considering the relation $Customers \times Products \times Seasons$, let us discuss about the rule $\{melon, orange\} \rightarrow \{summer, autumn\}$ when its frequency is 0.01, its natural confidence is 0.5, and its exclusive confidence is 0.7. The values of the measures tell that 1% among the customers (this domain is the support domain) buy *melon* and *orange* together in both *summer* and *autumn* seasons, a half of the customers who are buying *melon* and *orange* together during a same season do so during both *summer* and *autumn*. Finally, the high exclusive confidence indicates that customers rarely buy *melon* and *orange* together in other seasons.

Our implemented multidimensional association rule mining method has been designed as a post-processing of the closed patterns that hold in n-ary relations. It proceeds in three successive steps:

- (1) We prepare the multidimensional relation (i.e., a Boolean tensor) to mine ;
- (2) We compute frequent closed sets thanks to the state-of-the-art algorithm Data-Peeler [32] ;
- (3) We process these frequent closed patterns to derive from them the non-redundant rules whose natural and exclusive confidences exceed the user-defined thresholds.

Multidimensional disjunctive rules

The multidimensional disjunctive rule mining task addresses the following question: "Which cases can occur when we observe a frequent association?" A *multidimensional disjunctive association rule* is an implication between associations: its body is an association and its head is a disjunction of associations. Enabling disjunction provides more expressive rules. Again, we had to design relevant interestingness measures. First, the association measures of a disjunctive rule evaluate the probability of the conjunction between the body and each association in the head. Next, the disjunctive measures evaluate the probability to observe at

least one association in the head when the body holds. Here again, we have been looking further at the the concept of non-redundancy. For example, let us consider the rule $\{cherry\} \times \{summer\} \rightarrow (\{apple, pear\}) \vee (\{grape\} \times \{autumn\})$, where $\{apple, pear\}$ co-occurs with $\{cherry\} \times \{summer\}$ by an association frequency (denoted f_{a}) of 0.02 and an association confidence (denoted c_{a}) of 0.5, $\{grape\} \times \{autumn\}$ co-occurs with $\{cherry\} \times \{summer\}$ by an association frequency 0.05 and an association confidence 0.6. Finally, let us say that the disjunctive frequency and the disjunctive confidence (resp. denoted as f_{d} and c_{d}) of the rule are respectively 0.8 and 0.9. Such a rule means that that when a customer buys *cherry* in *summer*, he/she tends to buy *apple*, *pear* or *grape* ($f_{\mathrm{d}} = 0.8$, $c_{\mathrm{d}} = 0.9$). If he/she prefers the products *grape* then he/she tends to also buy it in *autumn* ($f_{\mathrm{a}} = 0.05$, $c_{\mathrm{a}} = 0.6$).

Multidimensional disjunctive rule mining needs four successive steps:

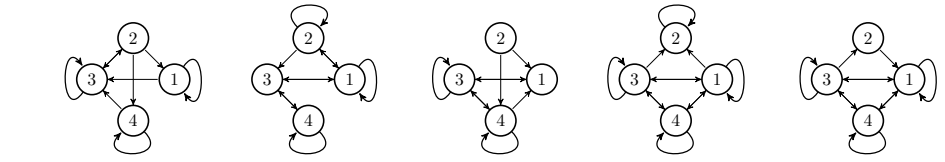
- (1) We prepare the multidimensional relation (i.e., a Boolean tensor) to mine ;
- (2) We compute frequent closed sets using Data-Peeler ;
- (3) We derive the *key association rules* satisfying the user-defined minimal association confidence constraint from the frequent closed sets ;
- (4) We compute the non-redundant disjunctive rules whose disjunctive frequency and disjunctive confidence are high enough given user-defined thresholds.

Applications to dynamic graph analysis

Graphs are a popular data structure to model the relationships between sets of entities. More and more graph data are available that denote, for instance, interactions between individuals in a social network. Graph data is ubiquitous and graph mining has recently received a lot of attention. Specially, many researchers are now interested in dynamic graphs that describe the evolution of a graph over time. However, there are only a few works concerning descriptive rule mining from dynamic graphs, to describe, for instance, local evolution trends over time (e. g., [111, 12]).

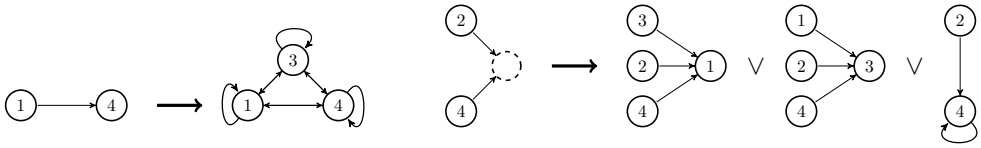
"What patterns can co-occur during the evolution of a graph?" is an important question that has not been really studied. For example, in a dynamic graph where we observe some periodical behavior, at what time does a bottleneck (i.e., many incoming edges) occur at a vertex? What vertices do outer edges tend to converge to? We seek to address this kind of questions thanks to multidimensional association (or disjunctive) rule mining.

Indeed, we focus on dynamic relational graphs whose vertices are all uniquely identified. We have a straightforward way to encode such a dynamic relational graph into a n-ary relation that is at least ternary (two dimensions are used to encode the adjacency matrices and at least one other dimension denotes time-stamps). To detect co-occurrences of patterns in the dynamic graph, we can express each pattern as an association in the associated n-ary relation, co-occurrences of patterns in the



	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4
d_1	1		1		1	1	1				1		1	1	1	1	1		1	1
d_2	1		1	1	1	1	1		1			1		1			1			
d_3		1	1		1			1	1		1	1	1	1	1	1	1	1	1	1
d_4			1	1			1	1	1		1	1	1		1	1	1		1	1
	t_1				t_2				t_3				t_4				t_5			

(a) A toy dynamic graph.



(b) Examples of rules mined on the toy dynamic graph.

Figure 1: Mining rules in dynamic graphs

dynamic graph as co-occurrences of associations in multidimensional association (or disjunctive) rules in the n-ary relation. For example, Figure 1a illustrates a toy dynamic relational graph represented by a ternary relation. Figure 1b shows some rules that we can mined thanks to our methods. The first rule means that the sub-network at its body can be enlarged to a clique with a high enough confidence. The second rule means that if the edges from Vertex 1 and Vertex 2 converge, they tend to converge to Vertex 1, Vertex 3 or Vertex 4.

We report experiments on real data that concern the `Vélov'v` network. `Vélov'v` is a bicycle rental service run by the urban community of Lyon in France, with 327 bicycle stations when the data was collected. A customer rents a bicycle at a station and returns it to any other station. We decided to build a dynamic graph that tells whether stations exchange a significant amount of bicycle at different time-stamps (the data is aggregated per day and per hour in a day). The goal of rule mining is to detect preferred time periods of departures and arrivals at stations, time periods of the exchange of bicycles between stations, stations that are blocked (impossible to rent or to return a bicycle) and their blocked time. We see that rule mining may support a better understanding of the `Vélov'v` network usage and thus it can help to improve the quality of service with respect to customers.

Thesis organization

The dissertation is structured in three parts.

Part 1 concerns the state-of-the-art. In Chapter 1, we introduce many useful concepts related to set pattern mining from binary relations. As a result, we discuss the frequent itemset mining task but also association rule mining. Chapter 2 presents existing approaches for mining patterns (n-sets, rules) in n-ary relations. Among others, it introduces the DATA-PEELER algorithm that computes closed n-sets under constraints.

Part 2 concerns our conceptual contribution. Chapter 3 focuses on multidimensional association rules in n-ary relations, it contains the definition of the pattern language and the design of relevant measures. An algorithm is proposed and it is empirically studied on real data. On this mining task, our preliminary results and first proposals were published in [NCB10, NCPB10] before the proposal in the conference paper [NCPB11] (Algorithm PINARD [1]) and its enhancement in the journal paper [NCPB11] (Algorithm PINARD++). Chapter 4 is about mining multidimensional disjunctive rules and it basically follows the same organization than Chapter 3. This mining task and the CIDRE [2] algorithm have been introduced in [NPB12].

Part 3 is dedicated to an application of our rule discovery methods to the analysis of dynamic relational graphs. It is made only of Chapter 5. The case study on the analysis of the `Vélov'v` network is detailed. A few computed rules are interpreted to have some qualitative counterpart to the performance studies in Part 2. [NCPB10] and [NCPB11] already addressed network analysis but this is detailed in the journal paper [NCPB11] and in the conference paper [NPB12].

Finally, we summarize the dissertation and we discuss directions for future research.

1. PINARD Is N-ary Association Rule Discovery.
2. CIDRE Is a Disjunctive Rule Extractor.

Part I

State of the art and theoretical basis

Outline

The simple abstraction of many data mining tasks proposed by Mannila and Toivonen [73] can be used here to introduce some relevant material about previous work. They assume that, in many data mining tasks, we look for all potentially interesting patterns from a pattern language $\mathcal{L}$ in a given database Db and that this can be expressed as the computation of $\{\rho \in \mathcal{L} \mid \mathcal{C}(\rho, Db) \text{ is true}\}$. Generally speaking, the constraint $\mathcal{C}$ specifies pattern interestingness in Db and it is a Boolean combination of several primitive constraints. In this thesis we only consider conjunctions of constraints. While many of them refer to the data (and checking whether they are satisfied needs to scan Db), it is also possible to have syntactic constraints that only work on ρ itself. In fact, for the sake of clarity and because it is always clear in the context, we generally omit the explicit reference to Db when defining/using constraints or operators.

In Chapter 1, we consider pattern domains that have been defined on binary relations, i.e., Db denotes a binary relation that will be often depicted by means of a Boolean matrix. Three pattern languages (three different instances of $\mathcal{L}$) will be considered. First, the language of itemsets is concerned: this is basically the set of attribute values that can be built on one of the two dimensions of the relation. Then, we can work with couples of sets of attribute values from both dimensions. Finally, we will consider the language of standard association rules that is built on attribute values from one dimension only. Some primitive constraints that have been well-studied are discussed. Among others, it includes constraints on the values of interestingness measures given user-defined thresholds (e.g., minimal frequency for itemsets and rules, minimal area for couples of sets, minimal confidence for rules). For itemsets and couples of sets, it concerns also maximality constraints and more generally constraints related to closedness.

In Chapter 2, we consider known extensions of these pattern languages when the data is an n-ary relation, often depicted by means of a Boolean tensor when $n > 2$.

Once declaratively specified, one challenge concerns the correct and complete computation of a priori interesting patterns. Major issues for the scalable computation

of patterns in both binary and n-ary relations are to be discussed as well. In Chapter 1, key concepts that have given rise to efficient algorithms for exploiting large binary relations are considered. In Chapter 2, we survey previous work in n-ary relations especially when $n > 2$. Among others, it introduces the Data-Peeler algorithm which has been both a motivation for this research and also a key component of the multidimensional rule mining software prototypes that are presented in Part 2.

Chapter 1

Association analysis in binary relations

This chapter is organized as follows. Section 1.1 recalls some terminology and basic concepts about binary relations and popular set pattern domains. Section 1.2 is dedicated to frequent itemset mining. Association rule mining is discussed in Section 1.3.

1.1 Binary relations and set pattern domains

1.1.1 Binary relations

A binary relation describes the relationship between the elements of two arbitrary sets, namely its *domains*. Given two finite disjoint sets $\mathcal{O} = \{o_1, \ldots, o_n\}$ and $\mathcal{P} = \{p_1, \ldots, p_m\}$, a binary relation on these sets, namely $\mathcal{B}$, is a collection of elements of the form (o_i, p_j) where $o_i \in \mathcal{O}$ and $p_j \in \mathcal{P}$. When $(o_i, p_j) \in \mathcal{B}$, it means that the relation holds between o_i and p_j. In other words, a relation on the domains $\mathcal{O}$ and $\mathcal{P}$ is a subset of their Cartesian product $\mathcal{O} \times \mathcal{P} = \{(o_1, p_1), ..., (o_1, p_m), ..., (o_n, p_1), ..., (o_n, p_m)\}$.

Definition 1 (Binary relation)**.** *Given two finite disjoint sets $\mathcal{O}$ and $\mathcal{P}$, a binary relation $\mathcal{B}$ on these domains is a subset of $\mathcal{O} \times \mathcal{P}$.*

Example 1. *Figure 2a presents an example of the binary relation $\mathcal{B}_E$ inspired by a basket data analysis setting. It concerns customers in $\mathcal{O}_E = \{o_1, o_2, o_3, o_4, o_5\}$ and products in $\mathcal{P}_E = \{p_1, p_2, p_3, p_4\}$. A couple $(o_1, p_1) \in \mathcal{B}_E$ means that the customer o_1 has been bying the product p_1.*

A binary relation like $\mathcal{B}_E$ can be described by a set of couples (Figure 2a), a set of transactions or sets (Figure 2b), or a matrix (Figure 2c).

(o_1, p_1), (o_1, p_2),
(o_2, p_1), (o_2, p_2), (o_2, p_4),
(o_3, p_1), (o_3, p_2), (o_3, p_3), (o_3, p_4),
(o_4, p_1), (o_4, p_3),
(o_5, p_2), (o_5, p_3), (o_5, p_4).

(a) $\mathcal{B}_E$ as a set of couples

Customers	*Products*
o_1	$\{p_1, p_2\}$
o_2	$\{p_1, p_2, p_4\}$
o_3	$\{p_1, p_2, p_3, p_4\}$
o_4	$\{p_1, p_3\}$
o_5	$\{p_2, p_3, p_4\}$

(b) $\mathcal{B}_E$ as a set of transactions

	p_1	p_2	p_3	p_4
o_1	1	1		
o_2	1	1		1
o_3	1	1	1	1
o_4	1		1	
o_5		1	1	1

(c) $\mathcal{B}_E$ as a matrix

Figure 2: A binary relation $\mathcal{B}_E$

Example 2. *Figure 2b describes a database of transactions (customer's purchases) over $\mathcal{P}_E$. Each transaction is a pair including a customer and a set of products he/she bought. For instance, the transaction $(o_1, \{p_1, p_2\})$ means that o_1 bought the products p_1 and p_2. It is represented by the couples (o_1, p_1) and (o_1, p_2) in Figure 2a. Figure 2c represents $\mathcal{B}_E$ by means of a Boolean matrix. A value 1 at the intersection of a row o_i and a column p_j means that $(o_i, p_j) \in \mathcal{B}_E$.*

Let us introduce some useful functions and concepts. We call $\mathcal{O}$ the set of objects and $\mathcal{P}$ the set of items. We write $2^{\mathcal{P}}$ (respectively $2^{\mathcal{O}}$) to denote the set of all subsets of $\mathcal{P}$ (respectively the set of all subsets of $\mathcal{O}$). For $P \subseteq \mathcal{P}$ and $O \subseteq \mathcal{O}$, we define the two following functions: $\psi(P)$ associates with P all the objects that share every item $p \in P$, i.e., it is the supporting set of P. $\phi(O)$ associates with O all the items that are shared by every $o \in O$, i.e., it is the supporting set of O.

Definition 2 (A Galois connection [109])**.** *Given a binary relation $\mathcal{B} \subseteq \mathcal{O} \times \mathcal{P}$, $P \in 2^{\mathcal{P}}$ and $O \in 2^{\mathcal{O}}$:*

$$\psi(P) : 2^{\mathcal{P}} \to 2^{\mathcal{O}},\ \psi(P) = \{o \in \mathcal{O} \mid \forall p \in P, (o, p) \in \mathcal{B}\}$$
$$\phi(O) : 2^{\mathcal{O}} \to 2^{\mathcal{P}},\ \phi(O) = \{p \in \mathcal{P} \mid \forall o \in O, (o, p) \in \mathcal{B}\}$$

The couple of applications (ψ, ϕ) is a Galois connection between the partial orders $(2^{\mathcal{P}}, \subseteq)$ and $(2^{\mathcal{O}}, \subseteq)$.

Example 3. *In the binary relation $\mathcal{B}_E$ presented in Figure 2, $\psi(\{p_1, p_4\}) = \{o_2, o_3\}$, $\phi(\{o_2, o_3\}) = \{p_1, p_2, p_4\}$, $\psi(\{p_1, p_2, p_4\}) = \{o_2, o_3\}$.*

Definition 3 (Galois closure operators and closed sets [109])**.** *Given $\mathcal{B} \subseteq \mathcal{O} \times \mathcal{P}$, $P \in 2^{\mathcal{P}}$ and $O \in 2^{\mathcal{O}}$, the operators $h(P) = \phi(\psi(P)) = \phi \circ \psi(P)$ and $h'(O) = \psi(\phi(O)) = \psi \circ \phi(O)$ are Galois closure operators. When $\mathcal{C}_{closed}(P) \equiv (h(P) = P)$ (resp. $\mathcal{C}_{closed}(O) \equiv (h'(O) = O)$) is satisfied, we say that P (resp. O) is a closed set.*

Example 4. *In $\mathcal{B}_E$, $h(\{p_1, p_4\}) = \{p_1, p_2, p_4\}$, $h(\{p_1, p_2, p_4\}) = \{p_1, p_2, p_4\}$. Therefore, $\{p_1, p_4\}$ is not a closed set, $\{p_1, p_2, p_4\}$ is a closed set. Notice that $\{o_2, o_3\} = \psi(\{p_1, p_2, p_4\})$ is a closed set as well.*

1.1.2 Pattern languages on binary relations

In this chapter, the data is a binary relation $\mathcal{B} \subseteq \mathcal{O} \times \mathcal{P}$ (for instance $\mathcal{B}_E$ presented in Figure 2). Let us now consider the different pattern languages of interest. Three pattern languages are considered:

- The so-called *language of itemsets* $2^{\mathcal{P}}$ is quite popular. We may also consider the other dimension and thus $2^{\mathcal{O}}$ instead. Later on, patterns from these languages are to be considered as special cases of *associations* that involve only one of the (two) domains of a binary relation.
- The language of bi-sets $2^{\mathcal{O}} \times 2^{\mathcal{P}} = \{(O, P) \mid O \subseteq \mathcal{O} \text{ and } P \subseteq \mathcal{P}\}$ is interesting as well. For instance, assuming $P \in 2^{\mathcal{P}}$, it makes sense to consider a pattern like $(\psi(P), P)$ which is basically an itemset and its supporting set of objects. Alternatively, one may be interested in $(O, \phi(O))$ for $O \in 2^{\mathcal{O}}$.
- The language $2^{\mathcal{P}} \times 2^{\mathcal{P}}$ can be used to denote standard association rules in binary relations. Indeed, such rules are couples of itemsets (X,Y) and, generally, we prefer to write $X \rightarrow Y$ to emphasize its body and its head.

Example 5. *Examples of itemsets in* $\mathcal{B}_E$ *are* $\emptyset$, $\{p_4\}$ *or* $\{p_1, p_2\}$. *Examples of bi-sets in* $\mathcal{B}_E$ *are* $(\{o_1, o_2\}, \{p_1, p_2\})$, $(\{o_2, o_3\}, \{p_2, p_4\})$ *or* $(\{o_2, o_3\}, \{p_1, p_2, p_4\})$. *Examples of association rules in* $\mathcal{B}_E$ *are* $\{p_1\} \rightarrow \{p_2\}$, $\{p_4\} \rightarrow \{p_1, p_2\}$ *or* $\{p_2, p_4\} \rightarrow \{p_3\}$.

Definition 4 (Formal concepts and closed 2-sets [41])**.** *Given* $\mathcal{B} \subseteq \mathcal{O} \times \mathcal{P}$, $O \in 2^{\mathcal{O}}$, *and* $P \in 2^{\mathcal{P}}$, (O, P) *is a formal concept or a closed 2-sets iff* $(O = \psi(P)) \wedge (P = \phi(O))$. *By construction,* O *and* P *are closed sets.*

Example 6. *Examples of bi-sets that are closed 2-sets in* $\mathcal{B}_E$ *are* $(\{o_2, o_3, o_5\}, \{p_4\})$, $(\{o_2, o_3\}, \{p_1, p_2, p_4\})$, *or* $(\{o_1, o_2, o_3\}, \{p_1, p_2\})$.

The term "formal concept" is used by the Formal Concept Analysis research community [41]. If (O, P) is a formal concept, its set of objects O is called its extent while its set of items P is called its intent. The term "closed 2-set" is used because formal concepts are a special case ($n = 2$) of closed n-sets in n-ary relations [32] (See Sections 1.2.2 and 2.2.1).

1.1.3 Constraint-based mining

In the so-called inductive queries that formalize data mining tasks, we use primitive constraints to specify a priori interestingness. This includes objective interestingness thanks to, for instance, statistical measures, but also subjective interestingness that is related to the goals of the analyst. For us, the constraint $\mathcal{C}$ is a conjunction of primitive constraints. Primitive constraints on the pattern languages we have considered in $\mathcal{O} \times \mathcal{P}$ have been extensively studied (see, e.g., [95, 18] for comprehensive surveys). Let us illustrate some typical ones with comments that assume a basket data analysis setting.

– **Minimal frequency constraint:** It is possible to measure the "strength" of an itemset $P \in 2^{\mathcal{P}}$ by considering how many objects are in its supporting set $\psi(P)$: $|\psi(P)|$ is the so-called *frequency* of P. Assuming that the analyst provides a relative frequency threshold $\mu \in [0, 1]$, the minimal frequency constraint is defined as follows:

$$\mathcal{C}_{\mu-\text{frequent}}(P) \equiv \frac{|\psi(P)|}{|\mathcal{O}|} \geq \mu.$$

A frequent itemset that satisfies $\mathcal{C}_{\mu-\text{frequent}}$ is thus a set of products that are purchased together by a large enough proportion of customers since $Customers$ is the support domain for the associations on $Products$.

– **Minimal size constraint:** Minimal size constraints on set patterns are often useful. It may be used on itemsets: if α is a user-defined threshold and if $P \in 2^{\mathcal{P}}$, we can define a minimal size constraint like $\mathcal{C}_{\alpha-\text{min-size}}(P) \equiv |P| \geq \alpha$. It can also be interesting to look at large enough bi-sets in the sense that both of their sets satisfy some minimal size constraints, possibly with respect to different thresholds. For instance, if a bi-set $(\psi(P), P)$ is large enough in $\mathcal{B}_E$, it means that we have a set of products that is large enough and such that these products are bought together by a large enough number of customers, i.e., not only P satisfies a minimal frequency constraint but also it implies a minimal number of items.

– **Average gross profit constraint:** It is possible to have other informations about objects and items and to characterize a priori interesting patterns by means of various aggregates on these values. For instance, we may know the gross profit made when any customer $o \in \mathcal{O}$ buys a product $p \in \mathcal{P}$ (assume this is returned by the function $gp : \mathcal{B}_E \to \mathbb{R}_+$). A constraint enforcing that average gross profit is greater than $v \in \mathbb{R}_+$ for a bi-set (O, P) where $O \in 2^{\mathcal{O}}$ and $P \in 2^{\mathcal{P}}$ is defined as follows:

$$\mathcal{C}_{\text{avg-gp}}((O, P)) \equiv \frac{\sum_{(o,p)\in O\times P} gp(o, p)}{|O \times P|} \geq v.$$

– **Closedness:** A closed set is a set that is maximal w.r.t. set inclusion and some other criterion. Indeed, when we say that an itemset $P \in 2^{\mathcal{P}}$ is closed, it means that this is the maximal itemset with the supporting set of objects $\psi(P)$: it is not possible to add an item to P without loosing at least one object in the set $\psi(P)$. We already introduced the following primitive constraints on an itemset P or a set of objects $O \in 2^{\mathcal{O}}$: $\mathcal{C}_{\text{closed}}(P) \equiv (P = \phi(\psi(P)))$ or $\mathcal{C}_{\text{closed}}(O) \equiv (O = \psi(\phi(O)))$.

Combining primitive constraints enables to define a priori interestingness. For instance, assuming that $P \in 2^{\mathcal{P}}$, the constraint $\mathcal{C}_{\mu-\text{frequent}}(P) \wedge \mathcal{C}_{\text{closed}}(P)$ declaratively specifies the so-called *frequent closed set mining* task.

When looking for bi-sets $(O, P) \in \mathcal{O} \times \mathcal{P}$ that are large enough formal concepts w.r.t. user-defined thresholds (natural numbers γ and α), we may check for the constraint $(|O| > \gamma) \wedge (|P| > \alpha) \wedge \mathcal{C}_{\text{closed}}(P) \wedge (O = \psi(P))$. Alternatively, a "minimal area" constraint may be preferred to formalize the "large enough" property, i.e., using something like $(|O| \times |P| > \alpha)$ instead of $(|O| > \gamma) \wedge (|P| > \alpha)$.

We provide examples of popular constraints on association rules in Section 1.3.

1.1.4 About constraint-based pattern mining feasability

Let us now discuss the complexity of the task of computing $\mathcal{TH}(Db, \mathcal{L}, \mathcal{C})$, i.e., solving the inductive query on selection criterion $\mathcal{C}$. In this section and in this thesis, we consider correct and complete algorithms that have to compute exactly $\mathcal{TH}(Db, \mathcal{L}, \mathcal{C})$.

In a practical data mining setting, we expect that Db can be extremely large (up to millions of objects, up to tens of thousands of items). Among other things, it means that accessing the data can cost a lot and that the satisfiability test of a constraint may be quite expensive. It also tells that the language of patterns, even though finite, will be so large that it is impossible to try any naive enumeration of all the sentences (i.e., checking constraint $\mathcal{C}$ in a post-processing way). Even though computable, a huge solution may also be irrelevant because one cannot expect that the analyst can interpret them and thus find the true interesting patterns among the computed a priori interesting ones.

Obviously, the size of $\mathcal{TH}(Db, \mathcal{L}, \mathcal{C})$ is important for computational feasability. Since $\mathcal{C}$ specifies a priori interestingness of patterns, its careful definition has a major impact on pattern relevancy. However, it can be so that this size is too large and that the computation of $\mathcal{TH}(Db, \mathcal{L}, \mathcal{C})$ turns to be intractable. A pragmatic behavior in that case is to consider more selective/stringent constraints, i.e., to design $\mathcal{C}'$ such that one expects $|\mathcal{TH}(Db, \mathcal{L}, \mathcal{C}')| << |\mathcal{TH}(Db, \mathcal{L}, \mathcal{C})|$. For instance, we see in the next section that the idea of condensed representations of frequent patterns consists in rewriting a minimal frequency constraint on itemsets to compute a much smaller solution while preserving the information about every frequent itemset and its frequency.

Once we expect that the size of the solution is not too large, clever search space strategies are needed. Indeed, we may use the constraint properties to perform search space safe pruning, i.e., being able to ignore part of the search space without missing solutions. Many constraint properties have been proposed: monotonicity, anti-monotonicity, loose anti-monotonicity[16], succinctness[78], convertibility[83], flexibility[96], piecewise (anti)-monotonicity[32], etc. The surveys in [95, 31] provide comprehensive studies of such properties and we only recall here some of them that are used hereafter. Notice that the fundamental paper [73] has been discussing the complexity of computing $\mathcal{TH}(Db, \mathcal{L}, \mathcal{C})$ in the generic setting where it exists a specialization relation on $\mathcal{L}$ (e.g., $\subseteq$ in $2^{\mathcal{P}}$) and when constraint $\mathcal{C}$ is anti-monotone

(e.g., $\mathcal{C}_{\mu-\text{frequent}}$). In that case, borders are useful concepts to discuss various aspects of the task complexity (number of data scans, number of evaluated candidate patterns, etc). Later, nice complexity results have been obtained when considering the arbitray Boolean combination of monotone and anti-monotone primitive constraints [86].

Definition 5 (Monotonicity)**.** *Let $(\mathcal{L}, \preceq_{\mathcal{L}})$ be a partial order set, a constraint $\mathcal{C}$ is said monotone iff $\forall X, Y \in \mathcal{L}$ such that $X \preceq_{\mathcal{L}} Y$ then $\mathcal{C}(X) \Rightarrow \mathcal{C}(Y)$.*

Definition 6 (Anti-monotonicity)**.** *Let $(\mathcal{L}, \preceq_{\mathcal{L}})$ be a partial order set, a constraint $\mathcal{C}$ is said anti-monotone iff $\forall X, Y \in \mathcal{L}$ such that $X \preceq_{\mathcal{L}} Y$ then $\mathcal{C}(Y) \Rightarrow \mathcal{C}(X)$.*

An extension of anti-monotonicity is the loose anti-monotonicity [16]. A loose anti-monotone constraint is such that if it is satisfied by a pattern of cardinality k then it is satisfied by at least one of its sub-patterns of cardinality $k-1$.

Definition 7 (Piecewise (anti)-monotonicity [32])**.** *A constraint $\mathcal{C}$ is piecewise (anti)-monotone iff the rewritten constraint $\mathcal{C}'$, attributing a separate argument to every occurrence of every variable in the expression of $\mathcal{C}$, is (anti)-monotone w.r.t. each of its arguments.*

In [31], the author shows that the flexible constraints only are a subset of the piecewise (anti)-monotone constraints.

Example 7. *Let us assume $(2^{\mathcal{P}}, \subseteq)$ as a partial order. The constraint $\mathcal{C}_{\mu-frequent}$ is anti-monotone, constraint $\mathcal{C}_{\alpha-\text{-}min\text{-}size}$ is monotone, and constraint $\mathcal{C}_{avg\text{-}gp}$ is piecewise (anti)-monotone. Indeed, by attributing a separate argument to every occurrence of O and P, $\mathcal{C}_{avg\text{-}gp}((O,P))$ can be rewritten as follows:*

$$\mathcal{C}'_{avg\text{-}gp\geq 1}((O_1, O_2, P_1, P_2)) \equiv \frac{\sum_{(o,p)\in O_1\times P_1} gp(o,p)}{|O_2 \times P_2|} \geq 1.$$

$\mathcal{C}'_{avg\text{-}gp\geq 1}$ is monotone on O_1 and P_1. It is anti-monotone on O_2 and P_2. As a consequence, $\mathcal{C}_{avg\text{-}gp\geq 1}$ is, by definition, piecewise (anti)-monotone.

Notice that a conjunction of monotone (resp. anti-monotone) constraints is monotone (resp. anti-monotone).

1.2 Frequent Itemset Mining

The goal of frequent itemset mining (FIM) has been first to find interesting associations of items that often occur together in a collection of transactions [2]. The frequent itemset mining has become one sub-problem of association rule mining, correlation analysis, associative classification, categorical data clustering, etc. The application of FIM goes far beyond basket data analysis and we have now about 20

years of research on solving efficiently this popular task. Interestingly, when solving FIM, we have to face with most of the fundamental issues of pattern discovery. It obviously motivates that researchers continue to consider FIM as a nice setting for studying many data mining open problems.

1.2.1 The original setting

Originally, the task of frequent itemset mining was introduced by IBM researchers in 1993 for basket data analysis [2]. It aims at finding all the subsets of items that frequently occur in a collection of transactions and we already illustrated that such a dataset corresponds to a large binary relation $\mathcal{O} \times \mathcal{P}$ where $\mathcal{O}$ (resp. $\mathcal{P}$) correspond to transactions (resp. to items). Given our notations, FIM can be formalized as follows:

Definition 8 (Frequent itemset mining)**.** *Given $\mathcal{B} \subseteq \mathcal{O} \times \mathcal{P}$ and a minimum frequency threshold $\mu \in [0, 1]$, the Frequent Mining Itemset task concerns the computation of $\mathcal{TH}(\mathcal{B}, 2^{\mathcal{P}}, \mathcal{C}_{\mu-frequent}) = \mathcal{S}_{freq} = \{\rho \in 2^{\mathcal{P}} \mid \mathcal{C}_{\mu-frequent}(\rho)$ is true$\}$.*

A key issue is that the search space for FIM, i.e., the pattern language, is structured as a lattice and that set inclusion is a specialization relation w.r.t. the minimal frequency constraint.

Example 8. *Figure 3b shows the complete itemset lattice of the example relation $\mathcal{B}_E$. It contains 16 itemsets and its height is 4. Assuming $\mu = 0.4$, we have 13 frequent itemsets in $\mathcal{B}_E$ (see Figure 3).*

Theorem 1 (Minimal frequency anti-monotonicity)**.** *$\forall X, Y \subseteq \mathcal{P}$, if $X \subseteq Y$ then $\mathcal{C}_{\mu-frequent}(Y) \Rightarrow \mathcal{C}_{\mu-frequent}(X)$ and thus $\neg\, \mathcal{C}_{\mu-frequent}(X) \Rightarrow \neg\, \mathcal{C}_{\mu-frequent}(Y)$.*

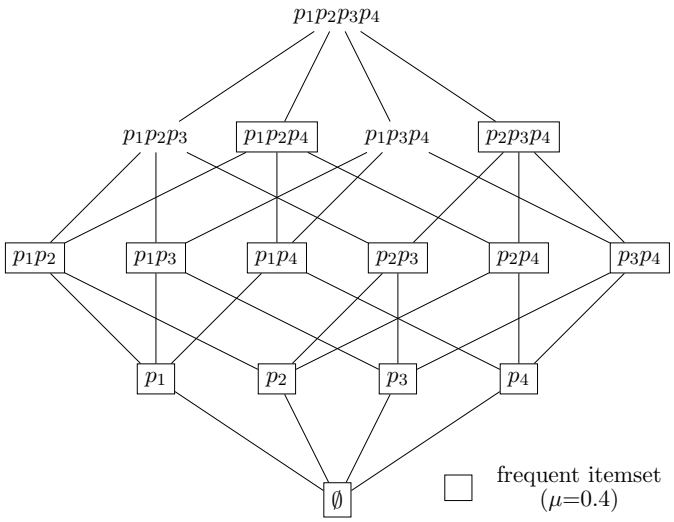
The minimal frequency anti-monotonicity has inspired many algorithms that efficiently prune the search space of itemsets (see Section 1.2.3) and achieve FIM tractability in sparse datasets like basket data ones. However, the size of $\mathcal{S}_{\text{freq}}$ can be huge, especially in dense and highly-correlated Boolean data. The so-called *condensed representations of frequent itemsets* have been studied for that purpose. The idea is to compute much smaller collections from which $\mathcal{S}_{\text{freq}}$ can be derived without having to access the data anymore. [21, 30] are survey papers on some of the condensed representations that can be used to solve FIM. We introduce some of them, namely the maximal frequent itemsets, the frequent closed itemsets, and the frequent free itemsets.

1.2.2 Constraint-based definition of condensed representations

It is useful to identify a small representative set of frequent itemsets from which all other frequent itemsets can be derived.

	p_1	p_2	p_3	p_4
o_1	1	1		
o_2	1	1		1
o_3	1	1	1	1
o_4	1		1	
o_5		1	1	1

(a) Binary relation $\mathcal{B}_E$



(b) Complete itemset lattice of $\mathcal{B}_E$.

Figure 3: Frequent itemsets in $\mathcal{B}_E$

The first idea could be to use here the concept of positive border of frequent itemsets [73] and thus to look at $\mathcal{B}d^+(\mathcal{S}_{freq})$, i.e., the collection of all the maximal frequent itemsets. Indeed, Theorem 1 says that if an itemset is frequent then all its subsets are frequent. If we know the frequent itemsets that are maximal (i.e., such that none of their supersets is frequent), then it is trivial to build the whole collection of the frequent itemsets.

Let us first recall the concept of border. Considering a set S of patterns from $\mathcal{L}$ such that S is closed downwards under a generalization relation $\preceq$ i. e., if $\rho \in S$ and $\theta \preceq \rho$, then $\theta \in S$. For frequent itemsets, $\subseteq$ is such a relation. The border $\mathcal{B}d(S)$ of S consists of those patterns ρ such that all generalizations of ρ are in S and none of the specializations of ρ is in S. Those patterns ρ in $\mathcal{B}d(S)$ that are in S are called the positive border $\mathcal{B}d^+(S)$, and those patterns ρ in $\mathcal{B}d(S)$ that are not in S are the negative border $\mathcal{B}d^-(S)$. In other words, the positive border consists of the most specific patterns in S (i.e., considering $\mathcal{S}_{freq}$, it means that we look for the maximal frequent itemsets) and the negative border consists of the most general patterns that are not in S (i.e., considering $\mathcal{S}_{freq}$, it means that we look for the minimal non-frequent sets, i.e., sets that are not frequent but whose all subsets are frequent).

Example 9. *Let* $\mathcal{S}_E = \mathcal{TH}(\mathcal{B}_E, 2^{\mathcal{P}_E}, \mathcal{C}_{0.4-frequent}) = \{\emptyset, \{p_1\}, \{p_2\}, \{p_3\}, \{p_4\}, \{p_1, p_2\}, \{p_1, p_3\}, \{p_1, p_4\}, \{p_2, p_3\}, \{p_2, p_4\}, \{p_3, p_4\}, \{p_1, p_2, p_4\}, \{p_2, p_3, p_4\}\}$.
$\mathcal{B}d^+(\mathcal{S}_E)$=$\{\{p_1, p_3\}, \{p_1, p_2, p_4\}, \{p_2, p_3, p_4\}\}$. $\mathcal{B}d^-(\mathcal{S}_E)$=$\{\{p_1, p_2, p_3\}, \{p_1, p_3, p_4\}\}$.

A positive border (or the negative one) can be used to characterize the solution of

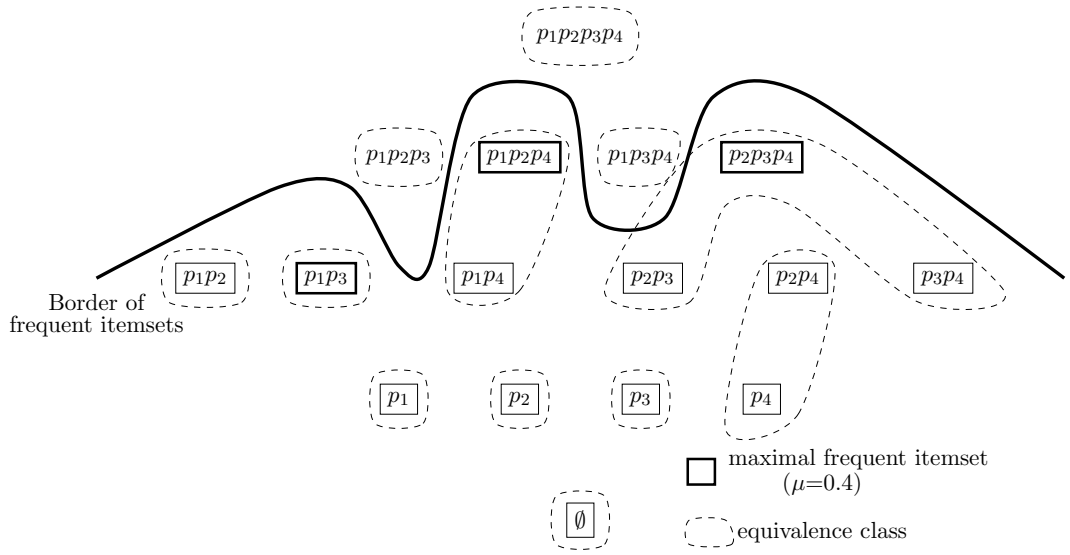


Figure 4: Positive border and equivalence classes of frequent itemsets in $\mathcal{B}_E$.

$\mathcal{TH}(\mathcal{B}, 2^{\mathcal{P}}, \mathcal{C})$ when $\mathcal{C}$ is anti-monotone, thus generalizing the case of just the mininmal frequency constraint. Dually, one can consider that when we have a monotone constraint, the solution is also characterized by a positive border. In fact, when we have to consider conjunctions of a monotone and an anti-monotone part, the collection of patterns is characterized by a so-called *Version Space* which is a couple of borders [86].

Maximal frequent itemsets

Definition 9 (Maximal frequent itemset)**.** *Given a minimum frequency threshold* $\mu \in [0, 1]$*,* $X \in 2^{\mathcal{P}}$ *is a maximal frequent itemset iff* $\mathcal{C}_{\mu-frequent}(X) \wedge \mathcal{C}_{maxi}(X)$ *where* $\mathcal{C}_{maxi}(X) \equiv (\forall p \in \mathcal{P} \setminus X, \neg\, \mathcal{C}_{\mu-frequent}(X \cup \{p\})$*.*

The set of maximal frequent itemsets corresponds to $\mathcal{TH}(\mathcal{B}, 2^{\mathcal{P}}, \mathcal{C}_{\mu-\text{frequent}} \wedge \mathcal{C}_{\text{maxi}}) = \mathcal{S}_{\text{freq-maxi}}$. It is the positive border of $\mathcal{TH}(\mathcal{B}, 2^{\mathcal{P}}, \mathcal{C}_{\mu-\text{frequent}})$, i. e., $\mathcal{B}d^+(\mathcal{S}_{\text{freq}})$.

The maximal frequent itemsets form the smallest set of itemsets from which all frequent itemsets can be derived. Despite providing a condensed representation, maximal frequent itemsets do not contain information about the supporting set (and thus the frequency) of their subsets. An expensive additional scan over the data is therefore needed to compute this information which is generally needed for the many applications of FIM, e.g., when deriving association rules (see Section 1.3). Therefore, it is relevant to look for condensed representations that preserve the information on supporting sets and thus frequencies of frequent itemsets.

Frequent closed itemsets

Computing $\mathcal{S}_{\text{freq-closed}} = \mathcal{TH}(\mathcal{B}, 2^{\mathcal{P}}, \mathcal{C}_{\mu-\text{frequent}} \wedge \mathcal{C}_{\text{closed}})$ provides the frequent closed itemsets. The nice formalization from [10] explains why this is a condensed representation of frequent itemsets. We can exploit equivalence classes of itemsets w.r.t. the relation `same-support`: given $X, Y \in 2^{\mathcal{P}}$, X `same-support` Y holds when $\psi(Y) = \psi(X)$. Each equivalence class of `same-support` is a group of itemsets with the same frequency. Each equivalence class has one maximal itemset which is a closed set. Therefore, mining frequent closed itemsets needs for computing the maximal itemsets of each equivalence class whose supporting sets of objects are large enough.

Definition 10 (Equivalence classes of `same-support`)**.** *Given $X \in 2^{\mathcal{P}}$, the equivalence class of X is $[X] = \{Y \in 2^{\mathcal{P}} \mid \psi(Y) = \psi(X)\}$.*

Example 10. *The equivalence classes of itemsets in $\mathcal{B}_E$ are show with dashed curves in Figure 4. Given $\mu = 0.4$, $\{p_1\}$, $\{p_1, p_2, p_4\}$ and $\{p_2, p_3, p_4\}$ are frequent closed itemsets in $\mathcal{B}_E$. Notice that $\{p_1\}$ is closed because it does not exist one superset with the same supporting set of objects. In contrast, $\{p_4\}$ is not closed because the supporting set of its superset $\{p_2, p_4\}$ is the same.*

Theorem 2. *For $X \in 2^{\mathcal{P}}$, $\psi(X) = \psi(h(X))$ (see, e.g., [115]).*

Thanks to this theorem, we can use the frequent closed itemsets to determine the frequency of the frequent itemsets that are not closed without accessing the data. If an itemset is not closed, its support must be identical to one closed itemset that is its superset, more precisely, the superset that is closed and has the largest supporting set of objects (i.e., the smaller one in terms of set cardinality). In the case of dense or correlated data, there are much fewer frequent closed itemsets than frequent itemsets and the whole information about the frequencies is preserved.

We have seen that formal concepts are built on closed sets: if $P \subseteq \mathcal{P}$, $O \subseteq \mathcal{O}$, (O, P) is a formal concept iff $(\psi(P) = O) \wedge (\phi(O) = P)$ and this turns to be equivalent to $(\mathcal{C}_{\text{closed}}(P) \wedge (\psi(P) = O))$ or $(\mathcal{C}_{\text{closed}}(O) \wedge (\phi(O) = P))$. Notice that formal concepts that correspond to frequent closed itemsets tends to have a large enough O and thus a rather small P. Another view on formal concepts that will be generalized later on (See Section 2.2.1) is now given. Formal concepts in a binary relation $\mathcal{B} \subseteq \mathcal{O} \times \mathcal{P}$ are bi-sets $(O, P) \in \mathcal{O} \times \mathcal{P}$ such that the conjunction of the two following constraints is satisfied:

- $\mathcal{C}_{\text{connected}}(O, P) \equiv O \times P \subseteq \mathcal{B}$,
- $\mathcal{C}_{\max}(O, P) \equiv (\forall o \in \mathcal{O} \backslash O, \neg\, \mathcal{C}_{connected}(\{o\}, P)) \wedge (\forall p \in \mathcal{P} \backslash P, \neg\, \mathcal{C}_{connected}(O, \{p\}))$.

The first constraint says that all the couples mades from one element of O and one element of P belong to the binary relation $\mathcal{B}$. In other terms, if bi-sets are considered as combinatorial rectangles (i.e., modulo arbitrary permutations of rows

and columns) in the Boolean matrices that represent the data, $\mathcal{C}_{\text{connected}}(O, P)$ says that (O, P) is a rectangle of true values. The second constraint $\mathcal{C}_{\max}$ says that (O, P) cannot be extended by an element from any dimension without a violation of $\mathcal{C}_{\text{connected}}$ and thus without introducing false values.

Example 11. *In $\mathcal{B}_E$, we see that $(\{o_2, o_3, o_4\}, \{p_1, p_4\})$ is not a closed 2-set because $(o_4, p_4) \notin \mathcal{B}_E$ (it does not satisfy $\mathcal{C}_{connected}$). $(\{o_2, o_3\}, \{p_1, p_4\})$ is not a closed 2-set because it does not satisfy $\mathcal{C}_{max}$. Indeed, it can be extended into $(\{o_2, o_3\}, \{p_1, p_2, p_4\})$ that satisfies $\mathcal{C}_{connected}$. Bi-set $(\{o_2, o_3\}, \{p_1, p_2, p_4\})$ is a closed 2-set.*

Free itemsets

To mine the frequent itemsets in $\mathcal{B}$, the approach of frequent closed itemset mining is based on maximal (w.r.t. set cardinality) itemsets of equivalence classes. On the contrary, frequent free itemset mining is based on their minimal itemsets. An itemset is a free itemset if it has no subset with the same supporting set of objects.

Definition 11 (Free Itemset)**.** *$X \in 2^{\mathcal{P}}$ is a free itemset iff X satisfies the freeness constraint $\mathcal{C}_{free}(X) \equiv (\psi(X) \subset \psi(Y),\ \forall Y \subset X)$.*

The terminology of free itemset has been introduced in [23] where this is a special case of the so-called δ-free itemsets (when $\delta = 0$). Free itemsets correspond to the minimal generators in [81] but also the key patterns in [10].

Property 1 (Freeness anti-monotonicity [24])**.** *Let $X \in 2^{\mathcal{P}}$, $\forall Y \subseteq X$, $\mathcal{C}_{free}(X) \Rightarrow \mathcal{C}_{free}(Y)$.*

Definition 12 (Frequent free itemset)**.** *Given a minimum frequency threshold $\mu \in [0, 1]$, $X \in 2^{\mathcal{P}}$ is a frequent free itemset iff $\mathcal{C}_{\mu-frequent}(X) \wedge \mathcal{C}_{free}(X)$ is true.*

Example 12. *Given $\mu = 0.4$, in $\mathcal{B}_E$, $\{p_1\}$, $\{p_4\}$ and $\{p_1, p_4\}$ are frequent free itemsets. $\{p_2, p_4\}$ is not a frequent free itemset, because $\{p_4\} \subset \{p_2, p_4\}$ and $\psi(\{p_4\}) = \psi(\{p_2, p_4\})$.*

The collection of frequent free itemsets is $\mathcal{TH}(\mathcal{B}, 2^{\mathcal{P}}, \mathcal{C}_{\mu-\text{frequent}} \wedge \mathcal{C}_{\text{free}}) = \mathcal{S}_{\text{freq-free}}$.

Property 2 (Free and closed itemsets)**.** *The closure of a free itemset gives a closed itemset and all the closed itemsets can be obtained by computing the closures of all free itemsets.*

The above property gives a generation process to get the frequency of all itemsets from free itemsets. Indeed, if an itemset is not free then it must exist a free itemset with the same supporting set. The support of every non-free itemset X is the support of the largest free itemset included in X. Therefore, it is possible to find the exact support of any frequent itemset in database. Even though $\mathcal{TH}(\mathcal{B}, 2^{\mathcal{P}}, \mathcal{C}_{\mu-\text{frequent}} \wedge \mathcal{C}_{\text{free}}) = \mathcal{S}_{\text{freq-free}}$ enables to compute exactly the frequency of all frequent non-free

sets, this is not enough to decide whether an itemset is frequent or not. For this purpose, we have to add the collection of infrequent free itemsets [22]. Now, given any itemset Y, if there exists $Z \subseteq Y$ such that Z is an infrequent free itemset, then we know that Y is not frequent. In the other case, the support of Y can be derived.

An equivalence class has only one maximal itemset (closed itemset) but it can have several minimal itemsets (free itemsets). It means that we have $|\mathcal{S}_{\text{freq-closed}}| \leq |\mathcal{S}_{\text{freq-free}}|$. However, freeness is anti-monotonic and this can be exploited efficiently for pruning. If a bounded number of errors on the frequency of itemsets is acceptable, the condensed representation of δ-free itemsets is more concise and can even be mined more efficiently [23, 22, 51]

Definition 13 (δ-free itemset)**.** *Given $\delta \in [0, |\mathcal{O}|]$, $X \in 2^{\mathcal{P}}$ is a δ-free itemset iff it satisfies the δ-freeness constraint $\mathcal{C}_{\delta-free}(X) \equiv (\psi(X) + \delta < \psi(Y),\ \forall Y \subset X)$.*

Theorem 3 (δ-freeness anti-monotonicity [22])**.** *Let $X \in 2^{\mathcal{P}}$, $\forall Y \subseteq X$, $\mathcal{C}_{\delta-free}(X) \Rightarrow \mathcal{C}_{\delta-free}(Y)$.*

Definition 14 (Frequent δ-free itemset)**.** *Given a minimum frequency threshold $\mu \in [0, 1]$, $\delta \in [0, |\mathcal{O}|]$, $\forall X \in \mathcal{P}$, X is a frequent δ-free itemset iff it satisfies $\mathcal{C}_{\mu-frequent}(X) \wedge \mathcal{C}_{\delta-free}(X)$.*

If $\delta = 0$ then a δ-free itemset is a free itemset.

Example 13. *Given $\delta = 1$ and $\mu = 0.4$, the frequent 1-free itemsets in $\mathcal{B}_E$ are $\{p_3\}$ et $\{p_4\}$. We see that the number of frequent 1-free itemsets is 2 that is less than the number of closed itemsets (8) or the number of free itemsets (9).*

$\mathcal{TH}(\mathcal{B}, 2^{\mathcal{P}}, \mathcal{C}_{\mu-\text{frequent}} \wedge \mathcal{C}_{\delta-\text{free}}) = \mathcal{S}_{\text{freq}-\delta-\text{free}}$ denotes the frequent δ-free itemsets. Using the collection of frequent δ-free itemsets and the set of infrequent δ-free itemsets, the frequency of any itemset X can be approximated. If X has a subset Y which is δ-free but not frequent then X is infrequent and the support of X is considered to be 0. Otherwise the frequency of X is approximated by the smallest supporting set among the supporting sets of the frequent δ-free itemsets that are its subsets. On dense data, δ-free itemset mining remains tractable once other methods fail [22].

The inclusion of frequent itemsets ($\mathcal{S}_{\text{freq}}$), maximal frequent itemsets ($\mathcal{S}_{\text{freq-max}}$), frequent closed itemsets ($\mathcal{S}_{\text{freq-closed}}$), and frequent free itemsets ($\mathcal{S}_{\text{freq-free}}$) are mentioned in Figure5. In the survey paper [30], other condensed representations have been discussed and their multiple uses have been introduced. Indeed, condensed representations have been designed not only to enhance FIM tractability in dense and correlated Boolean data but also because of their interesting semantics. For instance, properties like closedness, freeness or δ-freeness have been used in various context like non redundant association rule mining (see Section 1.3.2) or application-independant feature construction and associative classification (see, e.g., [42, 43]). Furthermore, it is clear that the maximality property on closed itemsets and closed 2-sets is extremely interesting within many pattern discovery processes.

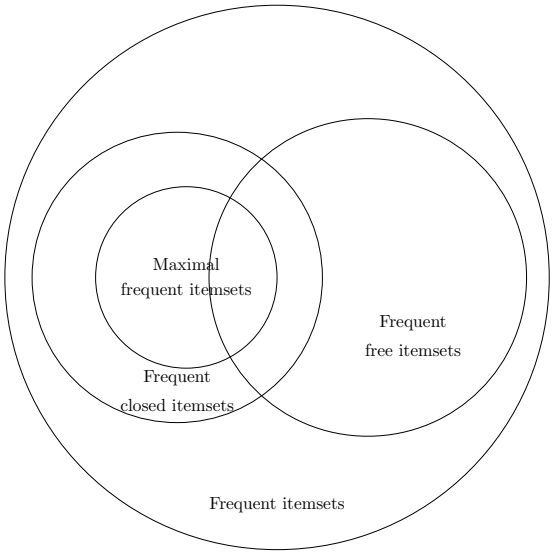


Figure 5: Classes of itemsets w.r.t $\mathcal{C}_{\mu-\text{frequent}}$, $\mathcal{C}_{\text{maxi}}$, $\mathcal{C}_{\text{closed}}$, $\mathcal{C}_{\text{free}}$.

1.2.3 Algorithmic issues

Computing frequent itemsets

Just after the problem setting and a first algorithm dedicated to FIM [2], the most influential ideas of the first efficient FIM algorithm have been published by Agrawal et al. [4, 3] and it has given the famous Apriori algorithm (around 13,000 citations recorded in the Google Scholar system). Actually Apriori solves both FIM and the association rule mining problem (see Section 1.3). It is a breadth-first (level-wise) complete search algorithm over the lattice associated to $(2^{\mathcal{P}}, \subseteq)$. Starting from singletons, it exploits Theorem 1 for safe pruning: it states that "When an itemset is infrequent, none of its superset can be frequent". Let us call a k-itemset. any itemset X whose cardinality $|X|$ is k. First, Apriori scans the data to find the frequent 1-itemsets (singletons that satisfy $\mathcal{C}_{\mu-\text{frequent}}$). Then it uses them to generate candidate 2-itemsets, and look at the data to obtain the frequent 2-itemsets. This process iterates until no more candidate k-itemsets can be generated for some k. When it stops, the maximal frequent itemsets have been found and, before them, all the frequent itemsets.

We use C_k (resp. F_k) to denote the set of k-itemsets that are candidate (resp. frequent) itemsets w.r.t. the frequency threshold μ. A high-level pseudo-code for the FIM part of Apriori is given as Algorithm 1.

It iterates on the two following steps:

– It checks whether $\mathcal{C}_{\mu-\text{frequent}}$ is satisfied for the candidates in C_k. For this, the data is scanned, one transaction at a time, and the frequency of every candidate

k-itemset that is supported by that transaction is incremented. All k-itemsets that satisfy the minimal frequency constraint are inserted into $\mathcal{F}_k$ (Line 1).

– It generates $\mathcal{C}_{k+1}$ by exploiting the frequent itemsets of size k (Line 2). This is performed in two sub-steps. First, in the so-called join step, the union $X \cup Y$ of sets $X, Y \in \mathcal{F}_k$ is generated if they have the same $k-1$ prefix (we assume that items in $\mathcal{P}$ are sorted in lexicographic order). Notice that this generation trick already performs some pruning "on the fly" (some candidates that would have to be pruned are even not generated). Second, in the pruning step, $X \cup Y$ is inserted into $\mathcal{C}_{k+1}$ only if all its k-subsets are frequent and thus belong to $\mathcal{F}_k$.

Input: $\mathcal{B} \subseteq \mathcal{P} \times \mathcal{O}$, $\mu \in [0, 1]$
Output: $\mathcal{TH}(\mathcal{B}, 2^{\mathcal{P}}, \mathcal{C}_{\mu-\text{frequent}})$

$C_1 \leftarrow \{\{p\} \mid p \in \mathcal{P}\}$
$k \leftarrow 1$
while $C_k \neq \emptyset$ **do**
  /* *Find frequent k-itemsets from the set of candidates C_k* */
1 $F_k \leftarrow \{X \in C_k \mid \mathcal{C}_{\mu-\text{frequent}}(X) \text{ is true}\}$
  /* *Generate candidates for level $k+1$* */
2 $C_{k+1} \leftarrow \text{Generate}(F_k)$
  $k \leftarrow k + 1$
Output $\cup F_k$

Algorithm 1: Frequent Itemset Mining with Apriori

Apriori is an efficient algorithm on typical basket analysis data where the largest frequent itemsets are not too large for the considered thresholds. It has however two fundamental limitations: (1) It requires multiple database scans and it has to check for a large set of candidates by computing many subset occurrences, (2) It requires a large amount of memory to handle the candidate itemsets. These can cause an immense amount of time spent and a memory overload when the collection of candidate itemsets is large. Let us observe that Apriori counts for the frequency of all the frequent itemsets plus those of the negative border of the frequent itemsets and this may be far too much in some datasets.

Many researchers have been studied these problems. First, the design of the condensed representations for frequent itemsets is clearly an answer. For instance, a straightforward modification of Apriori can be used to perform the level-wise search to compute frequent free itemsets and generate all the frequent sets and their frequencies thanks to them [10]. Other directions of work have been about data structures to support candidate pruning or candidate evaluation. Important proposals have concerned the use of a horizontal data format when keeping a breadth-first search, the use of a depth-first strategy with a vertical data format, building prefix-trees from the data and extracting frequent itemsets from such trees, or transforming

the original into transaction vectors used to immediately find frequent itemsets.

In APRIORI-like algorithms, generating a set of candidates and then evaluating each candidate to check for minimal frequency, it is possible to improve frequency counting (see, e.g., [25, 79]). If APRIORI and APRIORI-like algorithms use a horizontal data format during a breadth-first search, the ECLAT algorithm proposed by Zaki et al. [112] generates candidates in a depth-first strategy while the data are stored using a vertical format: object identifiers (OIDs) are associated with each itemset. With this format, mining can be performed by computing intersections of OIDs and the frequency count is simply the length of the OIDs for the itemset. There is no need to scan the data because the set of OIDs contains the needed information. To save memory, [114] proposes to avoid storing the sets of objects that support a k-itemset X and it only stores the difference between the supporting set of X and the supporting sets of its $k-1$-prefixes.

A rather different solution to FIM is the FP-GROWTH algorithm by Han et al. [50]. Instead of generating and testing candidates, FP-GROWTH encodes the data set using a compact data structure called an FP-tree and it extracts frequent itemsets directly from this structure. It scans the database only twice. In the first scan, all the frequent items and their frequencies are derived and they are sorted in the order of decreasing frequency in each transaction. In the second scan, items in each transaction are merged into a prefix-tree and items (nodes) that appear in common in different transactions are counted. FP-GROWTH works on FP-trees by choosing an item in the order of increasing frequency and extracting frequent itemsets that contain the chosen item by recursively calling itself on the conditional FP-tree. The main advantage of this technique is that it can exploit the so-called single prefix path case. That is, when it seems that all transactions in the currently observed conditional database share the same prefix, the prefix can be removed, and all subsets of that prefix can afterwards be added to all frequent itemsets that can still be found. This provides a significant performance improvement. Alternative data structures have been designed like the CP-tree in [103] that enables to have one database scan only.

Instead of using prefix-trees, [110] has introduced the transformation of each transaction into a $2^{|\mathcal{P}|}$-bit vector that corresponds to itemsets, called *transaction vectors*, one accumulates the frequency of occurrence of the itemsets such that. After scanning all the transaction vectors, one can immediately provide the frequent itemsets.

The above approaches suffer from massive memory requirements for any data that may contain too many frequent itemsets for the chosen threshold. Savasere et al. proposed the PARTITION algorithm [91] where the entire database is divided into n disjoint partitions such that each partition fits into main memory and can be mined separately. Since any itemset that is possibly frequent w.r.t. the entire data must occur as a frequent itemset in at least one of the partitions, all the found frequent itemsets become candidates which can be checked by accessing the entire dataset

only once. Another approach has been proposed by Toivonen [105]. The SAMPLING algorithm picks a random sample from the data and it looks for the frequent itemsets in that sample before checking the result within the whole database. In the cases where the sampling method does not produce all frequent itemsets, the missing sets can be found by generating all remaining potentially frequent itemsets and verifying their frequencies during a second scan over the data. The probability of such a failure can be kept small by decreasing the minimal support threshold during sample processing. However, for a reasonably small probability of failure, the threshold must be drastically decreased, which can cause a combinatorial explosion of the number of candidates. Nevertheless, in practice, finding all frequent patterns within a small sample of the database can be done very fast using any efficient frequent itemset mining algorithm. It has been shown that SAMPLING usually needs only one more scan resulting in a significant performance improvement [105].

An other approach is to design parallel mining algorithms for solving the problem of immense amount of time spent and memory overload when the collection of candidate itemsets is large. Such a algorithm can be executed a piece at a time on many different processing devices, and then put back together again at the end to get the correct result. The distributed dichotomous algorithm (DDA) proposed by Jen et al. [59] is an example. Its essential idea of is to partition the sets of candidate itemsets. First, the set of all 1-itemsets ($C_1 = \{p_1, ..., p_m\}$) is partitioned into two or three subsets (according to the parity of the cardinality of C_1). Those subsets are then used to partition the set of all k-itemsets (for a given $k > 1$) accordingly. To balance the workload of the machines involved in the computation, assuming that we have two machines M_1 and M_2 for computing the large itemsets, the sets of k-itemsets defined earlier are assigned to each machine M_1 and M_2 so as they both have the same number of candidates to process. The advantage of DDA is to work without data replication and redundant calculations, and moreover, the required degree of synchronization is low. Additionally, the flexibility of DDA allows to partition recursively the tasks and the data set until they fit the limited resources of computers.

Several efficient mechanisms have been designed to process other user-defined constraints. How to push different types of constraints together at mining time in order to reduce the computation as much as possible has been extensively studied. For example, Boulicaut et al. [24] studied the combining anti-monotone constraints and monotone constraints in order to get effective levelwise algorithms for mining frequent closed itemsets. Let us also recall that the relationship to *Version Spaces* has been studied in this context [86]. Bonchi et al. [17] showed how to combine anti-monotone constraints and monotone constraints for mining frequent closed itemsets in a depth-first computation. Bonchi et al. [16] introduced a class of tough constraints, namely *loose anti-monotone* constraints. Then they show how such constraints and anti-monotone constraints can be exploited in a level-wise Apriori-like

computation of frequent patterns by means of a data-reduction technique. The presentation in [18] reviewed and extended the state-of-the-art of the constraints that can be pushed in a frequent pattern computation. Many different kinds of constraints are pushed within a general level-wise APRIORI-like computation by means of data reduction techniques. For a comprehensive study on constraint-based mining for itemsets, we refer to [95].

Extracting maximal frequent itemsets

The implementation strategies for mining maximal frequent itemsets are based on the improvements and extensions of classical FIM algorithms like APRIORI, ECLAT, or FP-GROWTH. The main additions are the use of several lookahead techniques and efficient subset checking methods to efficiently prune the search space and be more efficient than during a classical FIM task. Notice that here, we are not looking for the frequency of every frequent itemset.

The PINCER-SEARCH algorithm [68] uses horizontal data format. It not only builds candidates in a bottom-up manner like APRIORI, but it also starts a top-down search at the same time, maintaining a candidate set of maximal itemsets. This can help in reducing the number of database scans, by removing earlier non-maximal itemsets. The maximal candidate set is a superset of the maximal itemsets, and in general, the overhead of maintaining it can be very high.

MAXMINER [89] employs a breadth-first traversal of the search space which is similar that of APRIORI. However, it uses efficient pruning techniques to quickly shrink the search. MAX-MINER uses pruning based on subset infrequency, as does APRIORI, but it also uses pruning based on superset frequency. To support pruning, MAX-MINER represents each node in the set enumeration tree by a two itemsets: the itemset enumerated by the node (called the head), and an ordered set of all items not in the head that can potentially appear in any sub-node (called the tail). If at a given node, the union of its head and its tail is frequent, then any itemset enumerated by a sub-node will also be frequent but not maximal. Superset-frequency pruning can therefore be implemented by stopping sub-node expansion at any node for which the union of its head and its tail is frequent. Next, considering the itemset made of the head and an item p in the tail, if this itemset is infrequent then any head of a sub-node that contains item p will also be infrequent. Subset infrequency pruning can therefore be implemented by simply removing any such tail item from a node before expanding its sub-nodes.

The DEPTHPROJECT algorithm was proposed by Agrawal et al. [1]. It represents the data as a bitmap. Each row in the bitmap is a bitvector corresponding to a transaction (an object), each column corresponds to an item. The number of rows is equal to the number of transactions, and the number of columns is equal to the number of items. A row has a 1 in the i^{th} position if the corresponding transaction contains the item p_i, and a 0 otherwise. This algorithm searches the itemset lattice in

a depth-first manner to find maximal frequent itemsets and it uses a counting method based on transaction projections along its branches. This projection is equivalent to a horizontal version of the TID-sets at a given node in the search tree. To reduce search space, DepthProject uses the look-ahead pruning method with item reordering. It returns a superset of the maximal frequent itemsets and it requires post-pruning to eliminate non-maximal itemsets. In [27], Burdick, Calimlim, and Gehrke extend the idea and propose the algorithm Mafia. It uses vertical bit-vector data format, and compressions and projections of bitmaps to improve performance. Mafia is a depth-first algorithm that uses three pruning strategies to remove non-maximal itemsets. The first is the look-ahead pruning first used in MaxMiner. The second is to check if a new itemset is subsumed by an existing maximal set. The last technique checks if $\psi(X) \subseteq \psi(Y)$. If so X is considered together with Y for extension. Mafia mines a superset of the maximal frequent itemsets and it requires a post-pruning step to eliminate non-maximal itemsets.

Gouda and Zaki have proposed the algorithm Genmax [47]. They use a novel technique called progressive focusing for maximality testing. Instead of comparing a newly found frequent itemsets with all maximal frequent itemsets found so far, it maintains a set of local maximal frequent itemsets. The newly found frequent itemset is firstly compared with itemsets in local maximal frequent itemset. Most non-maximal frequent itemsets can be detected by this step, thus reducing the number of subset tests. Genmax also uses a vertical representation of the data. However, for each itemset, Genmax stores a transaction identifier set, or TIS, rather than a bitvector. The cardinality of an itemset's TIS equals its frequency. The TIS of itemset $X \cup Y$ can be computed from the intersection of the TIS's of X and Y.

Finally, Fpmax [48] is an extension of the FP-Growth algorithm that exploits yet another Maximal Frequent Itemset tree structure to keep track of all maximal frequent itemsets.

Extracting frequent free itemsets and frequent closed itemsets

Several methods for extracting both frequent closed itemsets and frequent free itemsets (frequent generators) have been published, e.g. A-Close [80] or Titanic [99]. A-Close has two main steps. First, like Apriori-like algorithms, it browses level-wise the itemset lattice to mine the generators of all the closed itemsets (i.e., the free sets that are the minimal itemsets of all equivalence classes). These minimal elements can be discovered with intensive subset checking. After finding the frequent sets at level k, A-Close compares the support of each set with its subsets at the previous level. If the support of an itemset matches the support of any of its subsets, the itemset cannot be a free set and is thus pruned. Second, A-Close computes the closures of all the free sets found in the first step, which is done via intersection of all transactions where it occurs as a subset. This can be done in one pass over the data, provided all free sets fit in memory. Nevertheless computing closures this

way is an expensive operation. Moreover, since a single equivalence class may have more than one minimal itemsets, redundant closures may be computed. TITANIC is a descendent of A-CLOSE and its improved version PASCAL [10]. It relies on advanced features to avoid redundant computation, e.g., cardinality reasoning for closure computation and minimality checks for the filtering of non-free sets. Notice that the Formal Concept Analysis has also designed various algorithms for closed set mining without being interested in the use of other user-defined constraints but generally looking for the Galois lattice explicit building [41].

CHARM [115] and CLOSET [84], CLOSET+ [108] mine frequent closed itemsets without a candidate generation phase. Zaki et al. introduced the CHARM algorithm. It performs a bottom-up depth-first to generate frequent closed itemsets in a tree organized by inclusion. CHARM simultaneously explores both the itemset space and object space. It prunes candidates based on subset infrequency (i.e., no extensions of an infrequent are tested), and it also prunes candidates based on non-closure property, i.e., any non-closed itemset is pruned. To speed-up closure computation, it uses diffsets, the set difference on the TID-list of a given node (set of objects which support for the itemset of this node) and of its unique parent node in the tree. When a frequent itemset is generated, its TID-list is compared with those of the other itemsets having the same parent. The nodes whose TID-lists are the same, i. e., the itemsets of the nodes belong to the same equivalence, are merged. CHARM stores in the main memory the closed itemsets indexed by single level hash. It makes fewer database scans than the longest closed frequent set found. It scales linearly in the number of transactions and it is also linear in the number of found closed itemsets.

The CLOSET and CLOSET+ algorithms inherit from FP-Growth the compact FP-Tree data structure and the exploration technique based on recursive conditional projections of the FP-Tree. With a depth first browsing of the FP-Tree and recursive conditional FP-Tree projections, CLOSET mines closed itemsets by closure climbing, and by incrementally growing up frequent closed itemsets with items having the same support in the conditional data set. Duplicates are detected with subset checking by exploiting the property: given $X \subset \mathcal{P}$ and $p \in \mathcal{P}$, if $\psi(X) \subseteq \psi(p)$ then $p \in h(X)$. Thus, all closed sets previously discovered are kept in a two level hash table stored in main memory. CLOSET+ is an extension of CLOSET which is optimized for the case of sparse data sets whose transactions are quite short. In the case of dense data set, where the transactions are usually longer, closed itemsets equivalence classes are large and the number of duplicates is high, such a technique cannot be used because of its inefficiency [70].

The DCI-Closed algorithm [70] tackles the above problem. It tries to extract the set of the frequent closed itemsets without duplicate generation (hence in a linear time) and without maintaining it in main memory, using astute duplicate detection strategies.

While there are many works studying the frequent itemset (sets of columns) extraction technique enables to process cases large data that has millions of objects (respectively lines of binary matrix), hundreds of items (respectively columns of binary matrix) and the data is dense. Rioult et al. in [87] address extracting frequent itemsets in the case that data is dense and has few lines with regard to the number of columns. In this case, previous algorithms can fail. Thanks to the properties of Galois connections, if we compute the closed sets from the item space, the Galois connection allows to infer the closed sets of objects. Reciprocally, the extraction on the transposed matrix provides the closed sets on the objects and we can infer the closed sets of items. Thus, the same collection of closed sets can be extracted from a matrix or its transposed. So, in the case the matrix has few lines with regard to the number of columns, their idea is to use a transposed matrix to compute frequent closed itemsets for a original matrix. First, they extract free sets of objects which satisfy the frequency constraint. Second, they can compute the closures of these free sets. Then, they infer the frequent closed sets of items.

In [15], Besson et al. consided formal concept mining in difficult case, when the data is dense and when none of the dimensions is quite small.The proposed algorithm, namely DMiner, works top-down. It starts from the bi-set with all objects and all items. It performs a depth-first search of formal concepts by recursively splitting into bi-sets that do not contain false values. This algorithm is designed to exploit a large class of user-defined monotonic and anti-monotonic constraints.

In the case data is dense where the extraction of a complete and exact collection of frequent itemset becomes intractable, Boulicaut et al. [23, 22] proposed to compute δ-free itemsets. Because the δ-freeness is an anti-monotonic constraint (see Theorem 3) and the higher δ, the more we have pruning possibilities. So, the condensed representation of δ-free itemsets is more concise and can be mined more efficiently. The MinEx algorithm can be seen as an instance of the levelwise search algorithm. It explores the itemset lattice (w.r.t. set inclusion) levelwise, starting from the empty set and stopping at the level of the largest frequent free-sets. More precisely, the collection of candidates is initialized with the empty set as single member (the only set of size 0) and then the algorithm iterates on candidate evaluation and larger candidate generation. At each iteration of this loop, it scans the database to find out which candidates of size k are frequent free-itemsets. Then, it generates candidates for the next iteration, taking every itemset of size $k+1$ such that all proper subsets are frequent free-itemsets. The algorithm finishes when there is no more candidate. The search space is pruned by both the frequency constraint and the δ-free itemset constraint.

For mining δ-free itemsets, if MinEx performs a breadth-first search then FTminer proposed by Hébert et al. [51] is a depth-first search. From the observation that, with large data, there are only few objects which support for a set of items. The idea of FTminer is to check the δ-freeness constraint by using the corresponding

sets of objects. The key of the algorithm is to exploit a pruning criterion stemmed from the conjunction of the μ-frequency and δ-freeness. This criterion is checked by using the extensions of itemsets (extension of an itemset is the set of objects supporting it).

1.3 Association rule mining

1.3.1 Standard association rules

Pattern language and basic interestingness measures

Association rule mining was first introduced in [2] to support basket data analysis but has been now used in the many application domains where large binary relations that record Boolean properties of objects are available. We now survey some of the important aspects of this popular data mining task.

Given a binary relation $\mathcal{B} \subseteq \mathcal{O} \times \mathcal{P}$, standard association rules are built on itemsets (i.e., $2^{\mathcal{P}}$) and their objective interestingness are measured thanks to their supporting sets of objects (i.e., elements from $2^{\mathcal{O}}$).

Definition 15 (Association rule)**.** *Given a binary relation $\mathcal{B} \subseteq \mathcal{O} \times \mathcal{P}$, an association rule is an implication of the form $X \rightarrow Y$ where $X, Y \subseteq \mathcal{P}$ and $X \cap Y = \emptyset$. The itemsets X and Y are respectively called the body and the head of the rule.*

The condition that bodies and heads should be disjoint itemsets is motivated by the semantics of the rules once objective interestingness measures are used.

Definition 16 (Association rule frequency and confidence measures)**.** *Let $X \rightarrow Y$ be an association rule in $\mathcal{B} \subseteq \mathcal{O} \times \mathcal{P}$. Its frequency is $f(X \rightarrow Y)$ and its confidence is $c(X \rightarrow Y)$ with:*

$$f(X \rightarrow Y) = \frac{|\psi(X \cup Y)|}{|\mathcal{O}|} \text{ and } c(X \rightarrow Y) = \frac{|\psi(X \cup Y)|}{|\psi(X)|}.$$

Let us recall the toy example data $\mathcal{B}_E \subseteq \mathcal{O}_E \times \mathcal{P}_E$ from Figure 2c:

	p_1	p_2	p_3	p_4
o_1	1	1		
o_2	1	1		1
o_3	1	1	1	1
o_4	1		1	
o_5		1	1	1

Example 14. *In $\mathcal{B}_E$, consider $\{p_1\} \rightarrow \{p_2\}$ and $\{p_4\} \rightarrow \{p_1, p_2\}$:*

$$f(\{p_1\} \rightarrow \{p_2\}) = \frac{|\psi(\{p_1,p_2\})|}{|\mathcal{O}_E|} = \frac{|\{o_1,o_2,o_3\}|}{|\{o_1,o_2,o_3,o_4,o_5\}|} = \frac{3}{5}$$

$$c(\{p_1\} \rightarrow \{p_2\}) = \frac{|\psi(\{p_1,p_2\})|}{|\psi(\{p_1\})|} = \frac{|\{o_1,o_2,o_3\}|}{|\{o_1,o_2,o_3,o_4\}|} = \frac{3}{4}.$$

$$f(\{p_4\} \rightarrow \{p_1, p_2\}) = \frac{|\psi(\{p_1,p_2,p_4\})|}{|\mathcal{O}_E|} = \frac{|\{o_2,o_3\}|}{|\{o_1,o_2,o_3,o_4,o_5\}|} = \frac{2}{5}$$

$$c(\{p_4\} \rightarrow \{p_1, p_2\}) = \frac{|\psi(\{p_1,p_2,p_4\})|}{|\psi(\{p_4\})|} = \frac{|\{o_2,o_3\}|}{|\{o_2,o_3,o_5\}|} = \frac{2}{3}$$

Definition 17 (Association rule mining task)**.** *Given $\mu \in [0,1]$ and $\beta \in [0,1]$ the user-defined thresholds for frequency and confidence, an association rule is said frequent and valid if its frequency and its confidence are respectively greater than or equal to μ and β. In other words, an association rule $X \rightarrow Y$ is frequent and valid iff it satisfies the minimal frequency constraint $\mathcal{C}_{\mu-frequent}(X \rightarrow Y) \equiv (f(X \rightarrow Y) \geq \mu)$ and the minimal confidence constraint $\mathcal{C}_{\beta-valid}(X \rightarrow Y) \equiv (c(X \rightarrow Y) \geq \beta)$.*

Assume the collection of all possible association rules in $\mathcal{B}$ is denoted $\mathcal{L}_\mathcal{B}$, i. e., $\mathcal{L}_\mathcal{B} = \{X \rightarrow Y \mid X, Y \subseteq \mathcal{P}\}$, μ and β are the minimal frequency and the minimal confidence respectively. Association rule mining computes $\mathcal{TH}(\mathcal{B}, \mathcal{L}_\mathcal{B}, \mathcal{C}_{\mu-\text{frequent}} \wedge \mathcal{C}_{\beta-\text{valid}})$.

Computing association rules

If a rule $X \rightarrow Y$ satisfies the constraint $\mathcal{C}_{\mu-\text{frequent}}$, it means that $(X \cup Y)$ is a frequent itemset w.r.t. μ. Therefore, association rule mining can be decomposed into two subtasks: FIM and then valid rule generation from each frequent itemset. FIM was presented in Section 1.2.

When considering the rules that can be built from a given (frequent) itemset, we can use a pruning criterion to avoid testing rules whose confidence are for sure lower than the threshold β.

Theorem 4 (Confidence-based pruning w.r.t $\subseteq$)**.** *Given X, X', and Y in $2^\mathcal{P}$, let $X \subseteq X' \subseteq Y$, we have $c(X \rightarrow Y \setminus X) \leq c(X' \rightarrow Y \setminus X')$.*

Once frequent itemsets are available, rules can be extracted from them. The objective is to create, for every frequent itemset Y and its subsets X, a rule $Y \setminus X \rightarrow X$ such that $\mathcal{C}_{\beta-\text{valid}}(Y \setminus X \rightarrow X)$ is true. Theorem 4 shows that if $X \subset Z \subset Y$, we have $c(Y \setminus Z \rightarrow Z) \leq c(Y \setminus X \rightarrow X)$. Therefore, the largest confidence value will be obtained for body $Y \setminus X$ being as large as possible (or head X as small as possible). To generate association rules from a frequent itemset, the Apriori algorithm [4, 3] still uses a level-wise approach. It starts, at 1-level, with all rules including a single item in the head. Then, at 2-level, the candidate rule which have two items in the

head is generated by merging the heads of two rules from the 1-level. The process is performed until the kth-level until no more rule can be generated. Thanks to Theorem 4, if a rule does not satisfy $\mathcal{C}_{\beta-\text{valid}}$ then we do not need to evaluate a rule with a larger head.

A pseudocode for the rule generation step in the APRIORI algorithm is described in Algorithm 2. Mining association rules from a binary relation can be summarised in Algorithm 3.

Input: $Y \subseteq \mathcal{P}$, $\beta \in [0,1]$
Output: Every rule $Y \setminus X \rightarrow X$ satisfying the minimum confidence constraint
$C_1 \leftarrow \{\{p\} \mid p \in Y\}$
$k \leftarrow 1$
while $C_k \neq \emptyset$ **do**
  /* *Find frequent k-itemsets for the heads of rules* */
  $F_k \leftarrow \emptyset$
  forall $X \in C_k$ *such that* $\mathcal{C}_{\beta-valid}(Y \setminus X \rightarrow X)$ *is true* **do**
    $F_k \leftarrow F_k \cup X$
    Output $Y \setminus X \rightarrow X$
  /* *Generate candidates for level* $k+1$ */
  $C_{k+1} \leftarrow \text{Generate}(F_k)$
  $k \leftarrow k+1$

Algorithm 2: APRIORI rule generation

Input: $\mathcal{B} \subseteq \mathcal{O} \times \mathcal{P}$, $\mu \in [0,1]$, $\beta \in [0,1]$
Output: Every rule satisfying the minimal frequency and minimal confidence constraints
$\mathcal{S}_{\text{freq}} \leftarrow$ Frequent_Itemset_Mining_with_Apriori$(\mathcal{B}, \mu)$;
forall $Y \in \mathcal{S}_{freq}$ **do**
  Apriori_rule_generation(Y, β);

Algorithm 3: Mining association rules in a binary relation.

Among the well-known problems w.r.t. this standard version of association rule mining, the redundancy of computed rules has been identified and studied.

Example 15. *Given* $\mathcal{B}_E$, $\mu = 0.4$, *and* $\beta = 0.5$. *Let us consider the three following rules:*

- r_1: $\{p_4\} \rightarrow \{p_1, p_2\}$ *(*$f = \frac{2}{5}$, $c = \frac{2}{3}$*),*
- r_2: $\{p_4\} \rightarrow \{p_1\}$ *(*$f = \frac{2}{5}$, $c = \frac{2}{3}$*),*
- r_3: $\{p_4, p_2\} \rightarrow \{p_1\}$ *(*$f = \frac{2}{5}$, $c = \frac{2}{3}$*).*

The frequencies and the confidences of these rules exceed the given thresholds. The bodies of r_1 and r_2 are the same, but the head of r_2 is less informative than the head of r_1. When looking at the co-occurrences with item p_1, r_3 assume more information (the co-occurrence of p_2 with p_4) than r_1 and r_2. In other terms, we would like to consider r_2 and r_3 as more specific rules than r_1 and redundant ones in the sense that their frequency and confidence values are the same than those of r_1.

Another problem that is well-identified concerns the properties of the two objective measures that we have used so far.

Example 16. *In $\mathcal{B}_E$, with $\mu = 0.4$ and $\beta = 0.5$, let us consider the rule:*
– *r_4: $\{p_1\} \rightarrow \{p_3\}$ ($f = \frac{2}{5}$, $c = \frac{2}{4}$).*
At first glance, we might argue that if a customer buys p_1 also tends to buy p_3. Because the frequency and confidence of r_4 exceed the minimum thresholds. But, the fact that, the fraction of customers who buy p_3, regardless of whether they buy p_1, is $\frac{3}{5} = 0.6$, while the faction of customers buying p_1 tend to buy p_3 is only $\frac{2}{4} = 0.5$. Thus, a customer buying p_1 actually decreases her/his probability of buying p_3 from 0.6 to 0.5. The rule $\{p_1\} \rightarrow \{p_3\}$ is therefore misleading despite its high enough frequency and confidence.

Looking for solutions to these problems, let us now discuss some of the proposed solutions to improve the quality of association rules: the discovery of non-redundant rules, user-defined templates, the generalization to multilevel association rules, the design of alternative measures of interestingness.

1.3.2 Looking for relevant association rules

Non Redundancy

A problem in mining association rules is the number of extracted rules that is often very large. It can be even dramatic if the support and confidence thresholds are small and when data are dense or correlated. Indeed, in such cases, the number of frequent itemsets increases and the number of rules presented to the user typically increases exponentially. There are many approaches to define a condensed representation for association rules [63, 8]. We focus on the approach of the definition of "redundancy" for association rules, a rule is redundant if it can be inferred from other rules. As consequently, the condensed representation of rules retains only non-redundant rules. In this approach, the number of extracted rules can be reduced without losing any information. There are some different approaches for defining a non-redundant association rule: minimal body and minimal head [113], maximal body and maximal head [80, 98], minimal body and maximal head [82].

The non-redundant association rules proposed by Zaki [113] are based on free itemsets. A rule is non-redundant if and only if it does exist another rule whose

body and head are smaller but whose support and confidence are the same Such a rule is called the simplest association rule.

Definition 18 (Non redundant rule with minimal body and head)**.** *$\forall X, Y \subset \mathcal{P}$ and $X \cap Y = \emptyset$, the rule $X \rightarrow Y$ is non redundant iff it does not exist another rule $X' \rightarrow Y'$ which is different from $X \rightarrow Y$ such that $X' \subseteq X$, $Y' \subseteq Y$, $f(X \rightarrow Y) = f(X' \rightarrow Y')$ and $c(X \rightarrow Y) = c(X' \rightarrow Y')$.*

The set of all simplest association rules corresponds to the union of two subsets: the exact smallest association rules ($SimplestExact$) and the approximate smallest association rules ($SimplestApprox$):

$$SimplestExact = \{X \rightarrow Y \mid \mathcal{C}_{\text{free}}(X) \wedge \mathcal{C}_{\text{free}}(X \cup Y) \wedge h(X) = h(X \cup Y)\},$$
$$SimplestApprox = \{X \rightarrow Y \mid \mathcal{C}_{\text{free}}(X) \wedge \mathcal{C}_{\text{free}}(X \cup Y) \wedge h(X) \subset h(X \cup Y)\}.$$

In [80, 98], the non redundancy of association rules is based on the idea of the largest rules which are characterized by closed itemsets. A rule is non-redundant if and only if it does exist another rule whose body and head are larger but whose support and confidence are the same.

Definition 19 (Non redundant rule with maximal body and head)**.** *$\forall X, Y \subset \mathcal{P}$ and $X \cap Y = \emptyset$, the rule $X \rightarrow Y$ is non redundant iff it does not exist another rule $X' \rightarrow Y'$ which is different from $X \rightarrow Y$ such that $X' \supseteq X$, $Y' \supseteq Y$, $f(X \rightarrow Y) = f(X' \rightarrow Y')$ and $c(X \rightarrow Y) = c(X' \rightarrow Y')$.*

The set of the largest rules corresponds to the union of two subsets: the set of exact largest association rules ($LargestExact$) and the set of approximate largest association rules ($LargestApprox$):

$$LargestExact = \{X \rightarrow Y \mid \mathcal{C}_{\text{pseudo-closed}}(X) \wedge \mathcal{C}_{\text{closed}}(X \cup Y) \wedge h(X) = X \cup Y\},$$
$$LargestApprox = \{X \rightarrow Y \mid \mathcal{C}_{\text{closed}}(X) \wedge \mathcal{C}_{\text{closed}}(X \cup Y) \wedge X \subset (X \cup Y)\},$$

with $\mathcal{C}_{\text{pseudo-closed}}(X) \equiv (X \neq h(X)) \wedge (\forall X' \subset X \text{ s.t. } \mathcal{C}_{\text{pseudo-closed}}(X'), h(X') \subseteq X)$.

The non-redundant association rules studied by Pasquier et al. [82] are characterized by both closed itemsets and free itemsets. An association rule is considered redundant if it brings the same information or less information than is brought by another rules of the same support and confidence. Thus, such a non-redundant rule has minimal head and maximal body, called min-max association rule.

Definition 20 (Non redundant rule with minimal body and maximal head)**.** *$\forall X, Y \subset \mathcal{P}$ and $X \cap Y = \emptyset$, the rule $X \rightarrow Y$ is non redundant iff it does not exist another rule $X' \rightarrow Y'$ which is different from $X \rightarrow Y$ such that $X' \subseteq X$, $Y' \supseteq Y$, $f(X \rightarrow Y) = f(X' \rightarrow Y')$ and $c(X \rightarrow Y) = c(X' \rightarrow Y')$.*

The set of the min-max association rules corresponds to the union of two subsets: exact min-max association rules ($MinMaxExact$) and approximate min-max association rules ($MinMaxApprox$):

$$MinMaxExact = \{X \to Y \mid \mathcal{C}_{\text{closed}}(X \cup Y) \wedge \mathcal{C}_{\text{free}}(X) \wedge h(X) = X \cup Y \wedge X \neq Y\},$$
$$MinMaxApprox = \{X \to Y \mid \mathcal{C}_{\text{closed}}(X \cup Y) \wedge \mathcal{C}_{\text{free}}(X) \wedge h(X) \subset X \cup Y\}.$$

In addition, the above approaches can use the result of the following lemma to condense further the set of extracted rules.

Lemma 1 (Transitivity of confidence [71])**.** $\forall X \subseteq Y \subseteq Z \subseteq \mathcal{P}$, $c(X \to Y).c(Y \to Z) = c(X \to Z)$.

From this lemma and the observation that $\forall X \subseteq Y \subseteq Z \subseteq \mathcal{P}$, $f(X \to Z) = f(Y \to Z)$, we conclude that the frequency and the confidence of the rule $X \to Z$ can be inferred by those of the rules $X \to Y$ and $Y \to Z$. Such a rule $X \to Z$ is called a transitive rule. We can avoid to mine transitive rules. For example, Pasquier et al. [82] proposed a method for extracting association rules, namely non-transitive min-max association rules, which are both min-max association rules and non-transitive association rules.

User-defined templates

Syntactic constraints that enforce that some items appear or not, in bodies and/or heads of rules are quite often used. We can exploit further this idea by considering *rule templates* (see also the linguistic biases in different machine learning techniques). A rule template specifies what forms of rules are expected to be found from the data. Generally, such a rule template is used as a constraint during the data mining process. It can be represented as a structure language [9] or a meta-rule [40]. It provides a predefined format for the specification of rule extraction criteria that can even use variables that are instantiated during the rule extraction process.

Multilevel association rules

In many cases, most of the items have a low frequency. In such a case, if we mine association rules with a large enough frequency threshold then we can not find any interesting rule. But if we mine association rules with a low frequency threshold, extracted rules may be not interesting, and the number of computed rules explodes. To solve this problem, when a taxonomy on the items is available or can be designed, the user can be interested in finding association rules that span levels of the taxonomy instead of extracting rules on the set of items.

Example 17. *Figure 6 is a taxonomy saying that Jacket* is-a *Outerwear, Ski Pants* is-a *Outerwear, Outerwear* is-a *Clothes, etc. Rules which are found at the primitive*

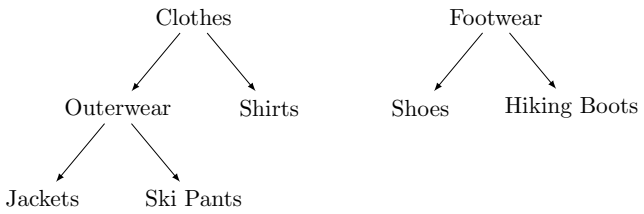


Figure 6: An example of a taxonomy [97].

concept level may be uninteresting. For instance, if we have only few customers buying Jackets, the frequency of the rule "customers who buy Jackets tend to buy Shoes" can be very low. But, we can be interested a rule like "customers who buy Outerwear tend to buy Shoes".

There are some possible directions to mine multiple-level association rules. In [97], Srikant et al. study association rules which can span different levels of the taxonomy. Such a rule can include items which belong to different levels. An obvious solution to the problem is to replace the dataset $\mathcal{B}$ by an extended dataset. Indeed, we can replace the set of items of an object o of $\mathcal{B}$ by another set which contains all the items in o but also all the ancestors of each item in o. Then, they can run any of the algorithms for association rule mining on the extended dataset. Notice that, obviously, the data now contain many built-in dependencies and some ad-hoc algorithms can be used.

While Srikant et al. [97] use the same minimum frequency and minimum confidence thresholds for all the levels to discover multiple-level association rules, Han et al. [49] adopt different minimum support thresholds for different levels. However, in this study, each multiple-level association rule consist of only items of a same level. For each level of the taxonomy, they scan the data to find frequent itemsets on this level. They then generate association rules satisfying the minimum confidence from each extracted frequent itemset.

Measures of interestingness

Interestingness measures play an important role in data mining, regardless of the kind of patterns being mined. These measures are intended for selecting and ranking patterns according to their potential interest to the user. Some of them are called objective because they rely on statistical observations while some others are called subjective because they take into account the analyst goal.

Objective measures Objective measures are numerical measures, they use concepts from probability, statistics, or information theory to estimate whether a pattern is interesting. The objective measures depend only on the data. Computing or using them do not require domain knowledge.

	Y	$\bar{Y}$	
X	$n(XY)$	$n(X\bar{Y})$	$n(X)$
$\bar{X}$	$n(\bar{X}Y)$	$n(\bar{X}\bar{Y})$	$n(\bar{X})$
	$n(Y)$	$n(\bar{Y})$	$\lvert\mathcal{O}\rvert$

Figure 7: Contingency table for rule $X \rightarrow Y$

Probability-based objective measures that evaluate the generality and the reliability of association rules have been thoroughly studied by many researchers (see [44] for a survey). To estimate the quality of rule $X \rightarrow Y$ in $\mathcal{B}$, these measures usually exploit the functions of a 2×2 contingency table (see Figure 7) where we have:

- $n(XY)$: number of objects supporting both X and Y,
- $n(X\bar{Y})$: number of objects supporting X but not Y,
- $n(X)$: number of objects supporting X,
- $n(\bar{X}Y)$: number of objects supporting Y but not X,
- $n(\bar{X}\bar{Y})$: number of objects that are neither supporting X nor Y,
- $n(\bar{X})$: number of objects which do not support X,
- $n(Y)$: number of objects supporting Y,
- $n(\bar{Y})$: number of objects which do not support Y,
- $|\mathcal{O}|$: total number of objects in $\mathcal{B}$.

Furthermore, let $P(X) = \frac{n(X)}{|\mathcal{O}|}$ denote the probability of X and $P(Y|X) = \frac{P(XY)}{P(X)}$.

Figure 8 lists some of the probability-based objective measures. Here, we analyze only some of them (see [101, 44, 66] for detailed analysis).

Interest factor. In Example 16, the rule $\{p_1\} \rightarrow \{p_3\}$ is misleading because its confidence measure ignores the frequency of the itemset appearing in the head of the rule. One way to solve this problem is to compute the ratio between the confidence of the rule and the frequency of its head. This measure has been called *Lift*:

$$Lift = \frac{P(Y|X)}{P(Y)}.$$

It is equivalent to another objective measure that has been called **interest factor**: it compares the frequency of itemset $X \cup Y$ and the frequencies of X and Y. It was designed to estimate whether the probabilities of X and Y are independent.

$$I(X,Y) = \frac{P(XY)}{P(X)P(Y)}$$

We can interpret the measure as follows:

$$I(X,Y) = \begin{cases} = 1 & \text{if } X \text{ and } Y \text{are independent,} \\ > 1 & \text{if } X \text{ and } Y \text{are positively correlated,} \\ < 1 & \text{if } X \text{ and } Y \text{are negatively correlated.} \end{cases}$$

1	Support	$P(XY)$
2	Confidence	$P(Y\|X)$
3	Lift/Interest	$\frac{P(Y\|X)}{P(Y)}$ or $\frac{P(XY)}{P(X)P(Y)}$
4	Jaccard	$\frac{P(XY)}{P(X)+P(Y)-P(XY)}$
5	Certainty Factor	$\frac{P(Y\|X)-P(Y)}{1-P(Y)}$
6	Odds Ratio	$\frac{P(XY)P(\bar{X}\bar{Y})}{P(X\bar{Y})P(\bar{X}Y)}$
7	Yule's Q	$\frac{P(XY)P(\bar{X}\bar{Y})-P(X\bar{Y})P(\bar{X}Y)}{P(XY)P(\bar{X}\bar{Y})+P(X\bar{Y})P(\bar{X}Y)}$
8	Yule's Y	$\frac{\sqrt{P(XY)P(\bar{X}\bar{Y})}-\sqrt{P(X\bar{Y})P(\bar{X}Y)}}{\sqrt{P(XY)P(\bar{X}\bar{Y})}+\sqrt{P(X\bar{Y})P(\bar{X}Y)}}$
9	Klosgen	$\sqrt{P(XY)}max(P(Y\|X)-P(Y), P(X\|Y)-P(X))$
10	Conviction	$\frac{P(X)P(\bar{Y})}{P(X\bar{Y})}$
11	Collective Strength	$\frac{P(XY)+P(\bar{Y}\|\bar{X})}{P(X)P(Y)+P(\bar{X})P(\bar{Y})} \times \frac{1-P(X)P(Y)-P(\bar{X})P(\bar{Y})}{1-P(XY)-P(\bar{Y}\|\bar{X})}$
12	Gini Index	$P(X)(P(Y\|X)^2+P(\bar{Y}\|X)^2)+P(\bar{X})(P(Y\|\bar{X})^2+P(\bar{Y}\|\bar{X})^2)-P(X)^2-P(\bar{X})^2$
13	Goodman and Kruskal	$\frac{\sum_i max_j P(X_iY_j)+\sum_j max_i P(X_iY_j)-max_i P(X_i)-max_j P(Y_j)}{2-max_i P(X_i)-max_j P(Y_j)}$
14	J-Measure	$P(XY)log(\frac{P(Y\|X)}{P(Y)})+P(X\bar{Y})log(\frac{P(\bar{Y}\|X)}{P(\bar{Y})})$
15	ϕ-Coefficient	$\frac{P(XY)-P(X)P(Y)}{\sqrt{P(X)P(Y)P(\bar{X})P(\bar{Y})}}$
16	Piatetsky-Shapiro	$P(XY)-P(X)P(Y)$
17	Cosine(IS)	$\frac{P(XY)}{\sqrt{P(X)P(Y)}}$
18	Sebag-Schoenauer	$\frac{P(XY)}{P(X\bar{Y})}$
19	Least Contradiction	$\frac{P(XY)-P(X\bar{Y})}{P(Y)}$
20	Odd Multiplier	$\frac{P(XY)P(\bar{Y})}{P(Y)P(X\bar{Y})}$

Figure 8: A sample of objective interestingness measures for rule $X \rightarrow Y$

	Y	$\bar{Y}$	
X	83	7	90
$\bar{X}$	7	3	10
	90	10	100

	W	$\bar{W}$	
Z	3	7	10
$\bar{Z}$	7	83	90
	10	90	100

Figure 9: Example of contingency tables of pairs of itemsets.

Example 18. *Consider* $\{p_1\} \rightarrow \{p_3\}$ *in* $\mathcal{B}_E$, $I(\{p_1\},\{p_3\}) = \frac{P(\{p_1,p_3\})}{P(\{p_1\})P(\{p_3\})} = \frac{5}{6} < 1$. *Therefore, the correlation between* p_1 *and* p_3 *is negative and it explains why the previous interpretation based on the confidence measure was misleading.*

Is Interest factor sufficient to evaluate the correlation between two itemsets? We illustrate its limitation with the following example.

Example 19. *Figure 9 provides contingency tables for two pairs of itemsets* (X, Y) *and* (Z, W) *where we assume that the dataset has 100 objects. We have* $I(X, Y) = 1.02$ *and* $I(Z, W) = 3$. *Because* $I(Z, W) > I(X, Y)$, *the correlation of* Z *and* W *seems much stronger than that of* X *and* Y. *But,* $P(Z|W) = 0.3$ *and* $P(W|Z) = 0.3$*: this means that* Z *and* W *seldom appear together in the data.* $I(X, Y) = 1,04$ *is close to 1 and thus it says that* X *and* Y *are independent. However,* X *and* Y *appear together in* 83% *of the objects and the rule* $X \rightarrow Y$ *turns to be interesting. From the examples in Figure 9, we see that the Interest factor is somewhat misleading.*

Correlation analysis. Correlation analysis uses a statistical-based technique to analyse the relationship between a pair of itemsets. The correlation of two itemsets is measured using the ϕ-coefficient, which is defined as

$$\phi(X, Y) = \frac{P(XY) - P(X)P(Y)}{\sqrt{P(X)P(Y)P(\bar{X})P(\bar{Y})}}.$$

The value of correlation ranges from -1 (perfect negative correlation) to $+1$ (perfect positive correlation). If $\phi = 0$, then X and Y is independent.

Example 20. *In* $\mathcal{B}_E$, *the* ϕ*-coefficient between* $\{p_1\}$ *and* $\{p_3\}$ *is* -0.41. *Therefore the correlation between* $\{p_1\}$ *and* $\{p_3\}$ *is negative. Moreover, in Figure 9, we have* $\phi(X, Y) = \phi(Z, W) = 0.22$.

The ϕ-coefficient measure has limitations. Consider two pairs of itemsets in Figure 9, although X and Y appear together more often than Z and W, the ϕ-coefficients are identical. This is because the ϕ-coefficient gives equal importance to both co-presence and co-absence of items in the data. It is therefore more suitable

for analysing symmetric binary variables (e. g., symmetric itemsets in a binary relation). Another limitation of this measure is that it does not remain invariant when there are proportional changes to the sample size.

IS measure. IS for a pair of itemsets (X, Y) is an alternative measure which includes both the Interest factor and the frequency of $X \cup Y$. This meausre is defined as follows:

$$IS(X,Y) = \sqrt{I(X,Y)f(X,Y)} = \frac{P(XY)}{\sqrt{P(X)P(Y)}},$$

where $f(X,Y) = \frac{|\psi(X \cup Y)|}{|\mathcal{O}|}$. It is possible to show that IS is mathematically equivalent to the cosine measure for two bit vectors [102].

Example 21. *Once again, consider the pairs of itemsets in Figure 9, we have* $IS(X,Y) = 0.92$, $IS(Z,W) = 0.3$. *Contrary to the results given by Interest factor and* ϕ*-coefficient, the* IS *measure suggests that the association between* X *and* Y *is stronger than the association between* Z *and* W. *It is consistent with that we expect from the given data.*

Assume two itemsets X and Y are independent, i. e., $P(AB) = P(X)P(Y)$, then $IS(X,Y) = \sqrt{P(X)P(Y)}$. In this case, the value $IS(X,Y)$ can be quite large, even for uncorrelated and negatively correlated itemsets.

Selecting appropriate measures. When looking at the quality of rules, we have to choose among an overwhelming number of interestingness measures (e. g., see the sample in Figure 8). However, not all measures are equally good and there is no measure that is consistently better than others in all application domains. This is because each measure has its own selection bias that justifies the rationale for fitting better to a dataset over another. Therefore, selecting appropriate measures for a given application is an important issue. Based on the properties of measures and empirical evaluations on datasets, ranking ans clustering methods have been proposed for comparing and analysing measures.

For example, Tan et al. [101] proposed a method to rank measures based on a specific dataset. First, the user selects a small set of datasets to mine patterns. Then, the user ranks a set of mined patterns, and the measure that has the most similar ranking results for these patterns is selected for further use. The selected patterns to rank have the greatest standard deviations in their rankings by the measures. Since these patterns cause the greatest "conflict" among the measures, they should be presented to the user for ranking.

Lenca et al. [66] presented another method to select the appropriate measures. It is based on the multiple criteria decision aid. In this approach, the user is not required to rank the mined patterns. But he/she must identify the desired properties and specify their significance for a particular application. Then, he/she assigns marks and weights to each property. Next, he/she sets up a table with each row

representing a measure, each column representing a property, and each cell (corresponding to the intersection of a row and a column) representing a weight. This table is called a decision matrix. Finally, by applying the multiple criteria decision process on this table, he/she can obtain a ranking of results.

An additional method for analyzing measures is to cluster the interestingness measures into groups [107]. This clustering method can be based on either the properties of the measures or the rule sets generated by experiments on datasets. *Property-based clustering* groups measures based on the similarity of their properties. *Experiment-based clustering* calculates the similarity between measures thanks to the rankings of their measures on a ruleset.

Subjective interestingness measures Patterns satisfying objective measures may not be interesting for the analyst. In such cases, the user's background knowledge and his/her objectives can help to select the appropriate patterns. Subjective interestingness aspects can be partly taken into account thanks to user-defined constraints. Also, subjective measures have been proposed: such a measure evaluates the interestingness of a pattern from the user point of view. Unlike the objective measures depending only on the data, a subjective interestingness measure takes into account both the data and the user's knowledge or goals. Because the user's knowledge may be represented in various forms, the subjective measures may not be representable by simple mathematical formulas. Therefore, the subjective measures are usually incorporated into the mining process. Let us mention three types of subjective measures of interestingness that are unexpectedness, novelty, and actionability.

- ***Unexpectedness.*** A pattern is unexpectedness if it is "surprising" to the user. It means that it contradicts a person's existing knowledge or expectations [93, 94] or it is an exception to a more general pattern which has already been discovered [11]. Such patterns are interesting because they identify failings in previous knowledge and may suggest an aspect of the data that needs further study.
- ***Novelty.*** A pattern is novel to a person if he or she did not know it before and is not able to infer it from other known patterns. Novelty is detected by having the user either explicitly identify a pattern as novel [90] or notice that a pattern cannot be deduced from and does not contradict previously discovered patterns. In the latter case, the discovered patterns are being used as an approximation to the user's knowledge.
  The difference between surprisingness and novelty is that a novel pattern is new and not contradicted by any pattern already known to the user, while a surprising pattern contradicts the user's previous knowledge or expectations.
- ***Actionability.*** A pattern is actionable if the user can do something with it to his or her advantage [93, 69]. Actionability is an important subjective measure

of interestingness because users always interested in patterns permitting them to improve their performance and establishing better work.

1.3.3 Disjunctive association rules

We present generalized disjunctive association rules such as "customers who buy shoes also buy jackets or shirts", "customers who buy either raincoats or umbrellas also buy flashlights", and "customers who buy jackets also buy bow ties or neckties and tiepins". Such rules can include disjunctions of itemsets.

A very few items are included in a large number of objects and most of the items are included in very few objects. Such a distribution is called *heavy-tailed* [77]. Since association rules in Definition 15 (conjunctive association rule) are based on the simultaneous occurrence of items in an object, they are ineffective in finding relationships when the items are included only in a few objects. One approach for this problem is to take a taxonomy on the items to mine multilevel association rules (as presented above). But, extracted multilevel association rules depend upon a predefined taxonomy and suffer from the problem of over-generalization. For example, it can not extract a rule as "customers who buy shoes also by jackets or shirts". To mine a rule in which an itemset with large frequency implies itemsets with small frequencies, Nanavati et al.[77] introduce generalized disjunctive association rules.

Definition 21 (Generalized disjunctive association rule)**.** *Let $\mathcal{X}$, $\mathcal{Y}$ be two sets of itemsets on $\mathcal{P}$. Let us denote by $\vee\mathcal{X}$ (respectively $\vee\mathcal{Y}$) a disjunction of the itemsets in $\mathcal{X}$ (respectively $\mathcal{Y}$). The implication $\vee\mathcal{X} \rightarrow \vee\mathcal{Y}$ is called a generalised disjunctive association rule.*

Definition 22 (Frequency of a disjunctive association rule)**.** *Let $\vee\mathcal{X} \rightarrow \vee\mathcal{Y}$ be a generalized disjunctive association rule. Its frequency is:*

$$f(\vee\mathcal{X} \rightarrow \vee\mathcal{Y}) = \frac{|(\cup_{X\in\mathcal{X}}\psi(X)) \cap (\cup_{Y\in\mathcal{Y}}\psi(Y))|}{|\mathcal{O}|}.$$

Definition 23 (Confidence of a disjunctive association rule)**.** *Let $\vee\mathcal{X} \rightarrow \vee\mathcal{Y}$ be a generalized disjunctive association rule. Its confidence is:*

$$c(\vee\mathcal{X} \rightarrow \vee\mathcal{Y}) = \frac{|(\cup_{X\in\mathcal{X}}\psi(X)) \cap (\cup_{Y\in\mathcal{Y}}\psi(Y))|}{|\cup_{X\in\mathcal{X}}\psi(X)|}.$$

Example 22. *In $\mathcal{B}_E$, $\{p_2\} \rightarrow \{p_1\} \vee \{p_3, p_4\}$ is a generalized disjunctive association rule whose frequency and confidence are:*

$$\begin{aligned} f(\{p_2\} \rightarrow \{p_1\} \vee \{p_3, p_4\}) &= \frac{|\psi(\{p_2\}) \cap (\psi(\{p_1\}) \cup \psi(\{p_3,p_4\}))|}{|\mathcal{O}_E|} \\ &= \frac{|\{o_1,o_2,o_3,o_5\} \cap \{o_1,o_2,o_3,o_4,o_5\}|}{|\{o_1,o_2,o_3,o_4,o_5\}|} = \frac{4}{5}, \end{aligned}$$

$$\begin{aligned} c(\{p_2\} \rightarrow \{p_1\} \vee \{p_3, p_4\}) &= \frac{|\psi(\{p_2\}) \cap (\psi(\{p_1\}) \cup \psi(\{p_3,p_4\}))|}{|\psi(\{p_2\})|} \\ &= \frac{|\{o_1,o_2,o_3,o_5\} \cap \{o_1,o_2,o_3,o_4,o_5\}|}{|\{o_1,o_2,o_3,o_5\}|} = \frac{4}{4}. \end{aligned}$$

1.4 Conclusion

Mining association rules in binary relations has a lot of applications and it has been extensively studied. However, most algorithms and techniques discussed above only concern patterns within a single domain (the domain of items). In the following chapter, we consider pattern mining methods in a multidimensional setting where patterns can involve elements of several domains.

Chapter 2

Association analysis in n-ary relations

Although, mining binary relations has a lot of applications, it is clear that many datasets of interest correspond to relations whose number of dimensions is greater than or equal to 3. For example, we can add spatial and temporal dimensions to a relation $Customers \times Products$ such that it becomes a relation $Customers \times Products \times Times \times Places$. In such a relation, we record that customers buy products in a given place at a given time. Another typical example concerns dynamic relational graph encoding for which we need to use at least three dimensions: two dimensions are to encode the graph adjacency matrices and at least one to denote time (see Section 5.1.1).

When the data has more than two dimensions, either premature projections are needed to use the binary relation mining algorithms or new Boolean attributes have to be designed that somehow combine information from the different dimensions. What are patterns in n-ary relations? Once pattern languages can be identified, what are the relevant primitive constraints that would support the discovery of interesting patterns? How can we compute them?

In this chapter, we consider some studies which seek to address these challenging questions. Particularly, we focus on two types of patterns that have been studied earlier in an n-ary relation mining setting, namely closed n-sets and rules. It is organized as follows. Section 2.1 defines n-ary relations and briefly considers relationships to multi-relational data mining. Section 2.2 discusses closed n-set mining. Finally, previous work that deals with rule discovery in n-ary relations is surveyed in Section 2.3.

	p_1	p_2	p_3	p_4	p_1	p_2	p_3	p_4	p_1	p_2	p_3	p_4	p_1	p_2	p_3	p_4
o_1	1	1			1	1		1		1	1			1	1	1
o_2	1	1		1	1	1					1					
o_3	1	1	1	1						1	1	1		1	1	
o_4	1		1					1							1	1
o_5		1	1	1			1		1	1	1	1	1		1	1
	s_1				s_2				s_3				s_4			

Figure 10: The n-ary relation $\mathcal{R}_E$

2.1 N-ary relations

A binary relation describes the relationship between the elements of only two domains, a n-ary relation enables to describe the relationship between the elements of n domains. Given an arbitrary number $n \in \mathbb{N}$, the set of n domains is denoted $\mathcal{D} = \{D^1, D^2, \ldots, D^n\}$ where each domain is a finite set of elements of a dimension. Without loss of generality, we assume the domains to be pairwise disjoints.

Definition 24 (N-ary relation)**.** *A n-ary relation $\mathcal{R}$ over $\{D^1, D^2, \ldots, D^n\}$ is a subset of the Cartesian product of these n domains, i. e., $\mathcal{R} \subseteq D^1 \times D^2 \times \cdots \times D^n$.*

Example 23. *Figure 10 is an example of a 3-ary relation, namely $\mathcal{R}_E$. It relates products in $D^1 = \{p_1, p_2, p_3, p_4\}$ bought along seasons in $D^2 = \{s_1, s_2, s_3, s_4\}$ by customers in $D^3 = \{o_1, o_2, o_3, o_4, o_5\}$. A 3-tuple $(p_i, s_j, c_k) \in \mathcal{R}_E$ corresponds to value '1' in Figure 10 at the intersection of three elements p_i, s_j and o_k. This means that the customer o_k buys the product p_i in the season s_j. For instance, the customer o_1 bought the product p_1 in the season s_1, but the customer o_4 did not bought the product p_2 in the season s_1.*

Such a n-ary relation can be also expressed as a relational database table that includes n attributes. Each attribute corresponds to one of the dimensions of the n-ary relation, its domain being the domain of its associated dimension. A record (say a tuple) in the relational database table corresponds to a n-tuple of the n-ary relation. However, when looking for database normalization, the table may be divided into smaller (and less redundant) tables, leading to a true (multi-)relational database.

Example 24. *The ternary relation $\mathcal{R}_E$ (see Figure 10) can be expressed as the relational database table in Figure 11.*

Pattern mining from relational databases has attracted some attentions in the past. For instance, pattern discovery techniques that look for dependencies (functional and inclusion dependencies) has been studied for a while. Also, some rule discovery techniques have been designed by the researchers from the Inductive Logic

product	season	customer
p_1	s_1	o_1
p_1	s_1	o_2
p_1	s_1	o_3
p_1	s_1	o_4
p_2	s_1	o_1
p_2	s_1	o_2
p_2	s_1	o_2
p_2	s_1	o_3
p_2	s_1	o_5
...	...	...

Figure 11: A relational table definition of $\mathcal{R}_E$

Programming community, i.e., researchers that focuse mainly on rich pattern languages that have to be discovered from multi-relational databases. This includes, for instance, the proposal for 1st order association rules [36].

Rcently, pattern discovery from relational database has attracted a significant attention. For example, Goethals et al. [46] address the issue of mining frequent conjunctive queries and association rule mining on arbitrary relational databases. In their work, each query is a relational algebra expression (including the projection and selection operators) on the Cartesian product of all tables in the mined relational database. In that setting, each association rule is a pair of queries, and the query in the head of the rule has to be more specific than the query in its body. The support of a query is the set of distinct tuples which are in the answer of the query. A query is said to be *frequent* if the cardinality of its answer is above a given threshold. While [46] concerns frequent conjunctive query mining without exploiting data dependencies, Jen et al. [57] propose to mine frequent queries (projection-selection queries) in a given relational table where functional and inclusion dependencies are known. They introduce a pre-ordering for comparing queries, and then show that the frequency (or support) measure is anti-monotonic w.r.t. this pre-ordering. They also define equivalent queries and they compute all frequent queries by exploiting the fact that equivalent queries have the same support. The authors then extend their study to mine frequent conjunctive queries (projection-selection-join queries) in a relational database [58]. In these queries, joins are performed along keys and foreign keys of tables in the database. However, the computation of frequent conjunctive queries is expensive for large fact tables (the number of scans of the database is quadratic). So, Dieng et al. [37] propose an efficient and scalable algorithm that overcome these limitations using appropriate auxiliary tables. To reduce the number of frequent conjunctive queries, Goethals et al. [45] present non redundant conjunctive query mining thanks to the use of functional dependencies.

Another approach presented by Hilali-Jaghdam et al. [52] is dedicated to frequent disjunctive query mining. They investigate the computation of frequent disjunctive selection queries in a given relational table. Such a query is a selection query on a relational table where the selection condition is a disjunction of equality comparison operators. They also address the design of a condensed representation which includes only the minimal frequent disjunctive selection queries.

The advantage of the mining frequent queries in a relational database is that each frequent query shows the correlation of elements belonging to different attribute domains. However, its limitation is that a frequent query can not show the correlation between elements in the same attribute domain. This limitation is due to the semantics of tuples in a table of a relational database. In a tuple, each attribute can not have more than one value. In the case of a binary relation like $Transactions \times Products$, [46] proposes a solution for this problem where they represent a product (an item) by a single unary relation in which each tuple is the transaction identifier of the transaction in which the product occurs. Then, the FIM task in a binary relation is replaced by the Frequent Query Mining task in the relational database that includes all such single unary relations. Notice however that this approach cannot be applied for n-ary relations with $n > 2$ because the Galois connection is lost in this setting (See Section 2.2.1).

Example 25. *In $\mathcal{R}_E$, we can mine patterns like $(\{p_1, p_2\}, \{s_1, s_2\}, \{o_1, o_2\})$ or $\{p_1, p_2\} \rightarrow \{s_1, s_2\}$. The first pattern means that the customers o_1 and o_2 buy the products p_1 and p_2 together in both seasons s_1 and s_2. The second pattern means that the customer often buys p_1 and p_2 together in the season s_1 and s_2. Such patterns can not be found with available data mining techbniques from relational databases.*

Techniques that compute interesting patterns in datasets defined as n-ary relations can overcome the above limitations.

2.2 Closed n-sets

We defined closed sets in binary relations and even the so-called closed 2-sets (see Section 1.1). We now consider the straightforward generalization to arbitrary relations. The goal is to look for maximal associations between elements of all the domains for a n-ary relation $\mathcal{R}$.

2.2.1 Definitions

A n-set is an association of n subsets of n domains of a relation $\mathcal{R} \subseteq D^1 \times \cdots \times D^n$.

Definition 25 (N-set)**.** *A pattern $X = (X^1, X^2, ..., X^n)$ such that $\forall i = 1..n, X^i \subseteq D^i$ is called a n-set. In other words, a n-set is a tuple of the Cartesian product $\times_{i=1..n} 2^{D^i}$.*

A n-set $X = (X^1, X^2, ..., X^n)$, $X^i \subseteq D^i$ for all $i = 1..n$, is a closed n-set in $\mathcal{R}$ if and only if (1) All elements of each set X^i are in relation with all the other elements of the other sets X^j ($j \neq i$) in $\mathcal{R}$ (we say that the constraint $\mathcal{C}_{\text{connected}}$ is satisfied), and (2) X^i sets cannot be enlarged by an element from any dimension without violating $\mathcal{C}_{\text{connected}}$ (we say that the constraint $\mathcal{C}_{\max}$ is satisfied).

Definition 26 (Closed n-set [33])**.** *$\forall X = (X^1, X^2, ..., X^n) \in \times_{i=1..n} 2^{D^i}$, X is a closed n-set iff it satisfies the conjunction of the two following constraints:*

- $\mathcal{C}_{connected}(X) \equiv \times_{i=1..n} X^i \subseteq \mathcal{R}$,
- $\mathcal{C}_{max}(X) \equiv \forall i = 1..n, \forall e \in D^i \backslash X^i, \neg\mathcal{C}_{connected}(X^1, X^2, ..., X^{i-1}, \{e\}, X^{i+1}, ..., X^n)$, *i. e.,* $X^1 \times X^2 \times ... \times X^{i-1} \times \{e\} \times X^{i+1} \times ... \times X^n \not\subseteq \mathcal{R}$.

Example 26. *In $\mathcal{R}_E$ (see Figure 10), $(\{p_1, p_2\}, \{s_1, s_2\}, \{o_1, o_2\})$ is a closed 3-set. $(\{p_1, p_2\}, \{s_1, s_2\}, \{o_1, o_2, o_3\})$ is not a closed 3-set because, among other things, we have $(p_1, s_2, o_3) \notin \mathcal{R}_E$. $(\{p_2\}, \{s_1, s_2\}, \{o_1, o_2\})$ is not a closed 3-set because it can be "extended" with p_1 without violating $\mathcal{C}_{connected}$.*

The Galois connection that exists in binary relations is a key property to enable the efficient computation of closed patterns like (frequent) closed itemsets or formal concepts (i.e., closed 2-sets). Indeed, it implies that the enumeration on one of the two dimensions enables to prune on the other one. However, we loose such a mechanism within n-ary relations with $n > 2$. Indeed, several closed n-sets can share a same subset of elements from one domain. For instance, $(\{p_2, p_3, p_4\}, \{s_1, s_3\}, \{o_3, o_5\})$ and $(\{p_3\}, \{s_1, s_3, s_4\}, \{o_3, o_5\})$ are closed 3-sets in $\mathcal{R}_E$. They both involve the subset $\{o_3, o_5\}$ of the third domain. In other terms, the subset $\{o_3, o_5\}$ does not uniquely determine $n-1$ set components which can connect with it to become a closed n-set. Nevertheless, $n-1$ set components of a closed n-set uniquely determines the last one [31]. Thus, unless $\mathcal{R}$ is a binary relation (i. e., $n = 2$), the related functions on closed n-sets are not injective and thus are not a part of Galois connections.

It makes sense to look for other constraints to express the a priori relevancy of (closed) n-sets. For instance, we can look at the closed n-sets $X = (X^1, \ldots, X^n)$ that are frequent in the sense where their X^i ($i = 1..n$) components are large enough thanks to set size constraints.

Definition 27 (Frequent closed n-set)**.** *Given $(\alpha^i)_{i=1..n} \in \mathbb{N}^n$, a closed n-set $X = (X^1, \ldots, X^n)$ is frequent in $\mathcal{R}$ iff it satisfies the $(\alpha^i)_{i=1..n}$-min-sizes constraint:*

$$\mathcal{C}_{(\alpha^i)_{i=1..n}\text{-}min\text{-}sizes}(X) \equiv \bigwedge_{i=1..n} (|X^i| \geq \alpha^i)$$

Such user-defined constraints have been used in the previous work on closed pattern discovery in n-ary relations [61, 56, 33]. More generally, the typical mining task concerns the computation of the following theory:

$$\mathcal{TH}(\mathcal{R}, \times_{i=1..n} 2^{D^i}, \mathcal{C}_{\text{connected}} \wedge \mathcal{C}_{\max} \wedge \mathcal{C}_{\text{relevancy}})$$
$$= \{X \in \times_{i=1..n} 2^{D^i} \mid \mathcal{C}_{\text{connected}}(X) \wedge \mathcal{C}_{\max}(X) \wedge \mathcal{C}_{\text{relevancy}}(X) \text{ is true}\}.$$

where $\mathcal{C}_{\text{relevancy}}$ specifies other aspects of objective and subjective interestingness that are beyond closedness which is already captured by $\mathcal{C}_{\text{connected}} \wedge \mathcal{C}_{\max}$.

2.2.2 Algorithms

Extracting frequent 3-sets from ternary relations

Ji et al. [61] propose two algorithms to extract closed 3-sets from ternary relations: *representative slice mining* and CubeMiner. The *representative slice mining* exploits frequent closed 2-set mining algorithms to mine frequent closed 3-sets. The basic idea is to transform a ternary relation into a set of binary relations, to mine the binary relation using an existing frequent closed 2-set mining algorithm, and then to prune away any frequent 3-set that is not closed.

While *representative slice mining* has the advantage that it can reuse existing frequent closed 2-set mining algorithms, the number of binary relations generated from the original ternary can be large. CubeMiner directly operates on the ternary relation. It generalizes the notion of cutter introduced in DMiner by Besson et al. [15] for closed 2-set mining. Cutters are used to split 3-sets to search for frequent closed 3-sets. A 3-set (X, Y, Z) is called a cutter if $\forall (x, y, z) \in X \times Y \times Z$, (x, y, z) is not in the ternary relation. CubeMiner first considers a 3-set which consists of the three whole domains of the ternary relation as a candidate. Then it splits this candidate recursively using the cutters until all cutters are used. Along a depth-first enumeration, the cutters are recursively applied to generate 3 candidate children containing less tuples absent from the relation than the parent: a first one without the elements of X, a second one without the elements of Y, and a third one without the elements of Z. For each candidate, several checks are required to ensure its closeness and unicity. For a child candidate to be unique, its newly removed elements must not be included in a cutter previously applied on this branch. To verify this, every formerly applied cutter is intersected with the current one. For a child candidate to be closed, the elements of these formerly applied cutters should not extend it. The authors indicate that the *representative slice mining* is efficient when the dimensions are small while CubeMiner performs better otherwise.

The same year, Jaschke et al. [56] have proposed the Trias to solve the same task, i.e., frequent closed 3-set mining in ternary relations. It relies on closed 2-set extractions from two binary relations. Given a ternary relation on three domains: D^1, D^2, D^3, Trias first constructs a new binary relation as $D^2 \times (D^2 \times D^3)$. Then it extracts every closed 2-set (X, Y) from this binary relation, X is a subset of D^1 and Y is subset of $D^2 \times D^3$. In the second step, Trias extracts every closed 2-set from the relation generated from Y and checks its closeness w.r.t. D^1. This can be performed easily by checking whether its closure is equal to X.

Extracting frequent n-sets from n-ary relations

Cerf et al. [33] propose the DATA-PEELER algorithm that can compute every closed n-set in arbitrary n-ary relations ($n \geq 2$). They indicate that despite the DATA-PEELER's broader scope, it is orders of magnitude faster than both TRIAS and CUBEMINER on ternary relations. Furthermore, DATA-PEELER can efficiently handle the expressive class of piecewise (anti)-monotonicity constraints. To simplify the brief introduction to DATA-PEELER, given n-sets $X = (X^1, X^2, ..., X^n)$ and $Y = (Y^1, Y^2, ..., Y^n)$, and $e \in \cup_{1..n} D^i$, we write:

- $X \sqsubseteq Y$ instead of $\forall i = 1, .., n, X^i \subseteq Y^i$,
- $X \sqcup Y$ instead of $(X^1 \cup Y^1, X^2 \cup Y^2, \ldots, X^n \cup Y^n)$,
- $X \sqcup e$ instead of $\begin{cases} (X^1 \cup \{e\}, X^2, ..., X^n) \text{ if } e \in D^1 \\ (X^1, X^2 \cup \{e\}, ..., X^n) \text{ if } e \in D^2 \\ ... \\ (X^1, X^2, ..., X^n \cup \{e\}) \text{ if } e \in D^n \end{cases}$,
- $X \setminus e$ instead of $\begin{cases} (X^1 \setminus \{e\}, X^2, ..., X^n) \text{ if } e \in D^1 \\ (X^1, X^2 \setminus \{e\}, ..., X^n) \text{ if } e \in D^2 \\ ... \\ (X^1, X^2, ..., X^n \setminus \{e\}) \text{ if } e \in D^n \end{cases}$.

DATA-PEELER recursively partitions the search space into two complementary parts following a popular "divide and conquer" strategy. In this way, a binary tree can represent the search space traversal. At every node of this tree, two n-sets, namely U and V, are updated that enable to bound the search space. From each node, we can derive all the n-sets containing all the elements of $\cup_{i=1..n} U^i$ and a subset of the elements of $\cup_{i=1..n} V^i$. In other words, each node is the search space of n-sets $(X^1, \ldots, X^n)$ such that $\forall i = 1, \ldots, n, U^i \subseteq X^i \subseteq (U^i \cup V^i)$. U is the smallest n-set that may be discovered from the node (according to $\sqsubseteq$), whereas $U \sqcup V$ is the largest. DATA-PEELER is initially called with $U = (\emptyset, \ldots, \emptyset)$ and $V = (D^1, \ldots, D^n)$ because, from this root node, all possible n-set are represented. In an enumeration sub-tree rooted by a left child, an arbitrary element $e \in \cup_{i=1,..,n} V^i$ is absent from every U n-set (e is "removed" from V). In the enumeration sub-tree rooted by its sibling node (right child), the same element e is present in every U n-set (e is "moved" from V to U).

Checking $\mathcal{C}_{connected}$. Right after an element e is "moved" to U (right child), the constraint $\mathcal{C}_{\text{connected}}$ is enforced. It removes from V every element $v \in \cup_{i=1,..,n} V^i$ that would violate $\mathcal{C}_{\text{connected}}$ if added to $(U \sqcup e)$, i. e., $\neg\mathcal{C}_{\text{connected}}(U \sqcup e \sqcup v)$. Figure 12 sums up this enumeration process.

Checking $\mathcal{C}_{max}$. For a node (U, V), if there exists an element $e \in D^i \setminus (U^i \cup V^i)$ such that $\mathcal{C}_{\text{connected}}(U^1 \cup V^1, ..., U^{i-1} \cup V^{i-1}, \{e\}, U^{i+1} \cup V^{i+1}, ..., U^n \cup V^n)$ is true, then every n-set discovered from the node (U, V) can be extended with e to form a larger

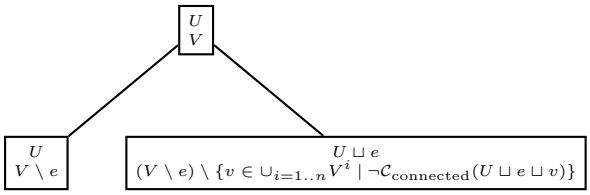


Figure 12: DATA-PEELER enumeration step for an element e.

n-set satisfying $\mathcal{C}_{\text{connected}}$. Thus, the n-set discovered from this node is not closed. In that case, such a node can be safely pruned. Indeed, we do not have to check every element of $\cup_{i=1..n} D^i \setminus \cup_{i=1..n}(U^i \cup V^i)$. Elements v which are removed from V when applying $\mathcal{C}_{connected}$ does not need to be checked because such a element v does not connect with any n-set discovered from the node (U, V). Only the elements e selected and removed during the enumeration, that is to say when a left child is built, have to be checked. Such elements e are put in a stack S.

Checking piecewise (anti)-monotonic constraints. According to the definition, each piecewise (anti)-monotonic constraint can be rewritten to form a new constraint which is (anti)-monotone w.r.t. each of its arguments. At each node (U, V), if an argument of the rewritten constraint is monotone, we check it with n-set $(U^1 \cup V^1, \ldots, U^n \cup V^n)$. If the constraint is not satisfied for $(U^1 \cup V^1, \ldots, U^n \cup V^n)$ then this will be the same for all the n-sets discovered from the node (U, V) and this node (U, V) can be pruned. If an argument of the rewritten constraint is anti-monotone, we check it with n-set U. If the constraint is not satisfied for U then the node (U, V) is pruned as well.

The DATA-PEELER pseudo-code for the computation of every closed n-set satisfying a conjunction of piecewise (anti)-monotonic constraints $\mathcal{C}_{\text{P(A)M}}$ is presented in Algorithm 4.

Input: A node (U, V) and a stack S
Output: Closed n-sets satisfying $\mathcal{C}_{\text{P(A)M}}$
if $\mathcal{C}_{max}(U, V, S) \wedge \mathcal{C}_{P(A)M}(U, V)$ **then**
  if $V = \emptyset$ **then**
    output U
  Choose $e \in \cup_{i=1..n} V^i$
  DATA-PEELER($U \sqcup e$, $(V \setminus e) \setminus \{v \in \cup_{i=1..n} V^i \mid \neg \mathcal{C}_{\text{connected}}(U \sqcup e \sqcup v)\}$, S)
  DATA-PEELER(U, $V \setminus e$, $S \cup \{e\}$)

Algorithm 4: DATA-PEELER.

The space complexity of DATA-PEELER is $\mathcal{O}((|D^1| + |D^2|)^2)$ if $n = 2$ and it is $\mathcal{O}(\times_{i=1..n}|D^i|)$ when $n > 2$ [33]. Notice that an extension of closed n-sets towards fault-tolerance has been designed as well. An absolute fault-tolerance has been

specified thanks to a straightforward generalization of the constraint $\mathcal{C}_{\text{connected}}$: some errors (say false values in the hyper-rectangle specified by the n-set) are accepted. From a computational perspective, this is much harder than computing closed n-sets. The so-called algorithm Fenster exploits the same enumeration strategy than Data-Peeler even though original counting mechansisms had to be designed to achieve enough efficiency [CBNB12].

2.3 Mining rules in n-ary relations

Traditional association rules describe relationships between elements on only a single dimension of a binary relation. To cope with higher arity relations, new rule pattern domains have to be studied. According to the number of dimensions appearing in a rule and the repetitions of dimensions in a rule, the rules mined from n-ary relations can be classified into three types: intra-dimensional, inter-dimensional and hybrid rules.

2.3.1 Intra-dimensional association rules

A rule whose elements belong to only one domain is called an intra-dimensional association rule. The association rule on itemsets in binary relations (see Definition 15) is a particular case of intra-dimensional association rule. Schmitz et al. [92] proposed the computation of intra-dimensional association rules in ternary relations. They are looking for association rules on one domain of the relation and they consider all tuples that belong to the Cartesian product of all other domains as a set of transactions. Formally, in the context of n-ary relation, the intra-dimensional association rules are defined as follows:

Definition 28 (Intra-dimensional association rule)**.** *$\forall D^i \in \mathcal{D}$, let $X, Y \subseteq D^i$, a rule $X \rightarrow Y$ is called an intra-dimensional association rule on D^i.*

Using the concatenation denoted as '.' (e. g., $(p_1) \cdot (s_3, o_5) = (p_1, s_3, o_5)$), the frequency and the confidence of an intra-dimensional association rule are defined as follows.

Definition 29 (Intra-dimensional association rule frequency)**.** *$\forall D^i \in \mathcal{D}$, let $X \rightarrow Y$ be an intra-dimensional association rule on D^i. Its frequency in $\mathcal{R}$ is:*

$$f(X \rightarrow Y) = \frac{|\{t \in \times_{D^j \in \mathcal{D} \setminus D^i} D^j \mid \forall e_i \in (X \cup Y), e_i \cdot t \in \mathcal{R}\}|}{|\times_{D^j \in \mathcal{D} \setminus D^i} D^j|}.$$

Definition 30 (Intra-dimensional association rule confidence)**.** *$\forall D^i \in \mathcal{D}$, let $X \rightarrow Y$ be an intra-dimensional association rule on D^i. Its confidence in $\mathcal{R}$ is:*

$$c(X \rightarrow Y) = \frac{|\{t \in \times_{D^j \in \mathcal{D} \setminus D^i} D^j \mid \forall e_i \in (X \cup Y), e_i \cdot t \in \mathcal{R}\}|}{|\{t \in \times_{D^j \in \mathcal{D} \setminus D^i} D^j | \forall e_i \in X, e_i \cdot t \in \mathcal{R}\}|}.$$

Example 27. *In $\mathcal{R}_E$, $\{p_2, p_3\} \rightarrow \{p_4\}$ is an intra-dimensional association rule.*

- *Its frequency is* $\frac{|\{(s_1,o_3),(s_1,o_5),(s_3,o_3),(s_3,o_5),(s_4,o_1)\}|}{|D^2 \times D^3|} = \frac{5}{20}$.
- *Its confidence is* $\frac{|\{(s_1,o_3),(s_1,o_5),(s_3,o_3),(s_3,o_5),(s_4,o_1)\}|}{|\{(s_1,o_3),(s_1,o_5),(s_3,o_1),(s_3,o_3),(s_3,o_5),(s_4,o_1),(s_4,o_3)\}|} = \frac{5}{7}$.

Assume the collection of all possible intra-dimensional association rules on D^i in $\mathcal{R}$ is denoted $\mathcal{L}_{\text{intra-dimensional}}$, i. e., $\mathcal{L}_{\text{intra-dimensional}} = \{X \rightarrow Y \mid X, Y \subseteq D^i\}$. The mining interesting intra-dimensional association rules on D^i in $\mathcal{R}$ corresponds to the finding of the following theory:

$$\begin{aligned}&\mathcal{TH}(\mathcal{R}, \mathcal{L}_{\text{intra-dimensional}}, \mathcal{C}_{\text{frequent}} \wedge \mathcal{C}_{\text{valid}})\\ &= \{r \in \mathcal{L}_{\text{intra-dimensional}} | \mathcal{C}_{\text{frequent}}(r, \mathcal{R}) \wedge \mathcal{C}_{\text{valid}}(r, \mathcal{R}) \text{ is true}\}.\end{aligned}$$

To compute interesting intra-dimensional association rules on D^i in $\mathcal{R}$, first we construct a new binary relation $\mathcal{R}'$ from $\mathcal{R}$ by "flattening" the dimensions in $\mathcal{D} \setminus D^i$ into a unique support dimension $D^{\text{supp}} = \times_{D^j \in \mathcal{D} \setminus D^i} D^j$. Assuming that for all $k = 1..n$, e_k is an element of the k^{th} domain, i. e., $e_k \in D^k$, $\mathcal{R}'$ is built as follows:

$$\mathcal{R}' = \{(e_i, (e_1, \cdots, e_{i-1}, e_{i+1}, \cdots, e_n)) | (e_1, \cdots, e_{i-1}, e_i, e_{i+1}, \cdots, e_n) \in \mathcal{R}\}.$$

Second, we use an algorithm which extracts association rules from a binary relation (see Section 1.3.1) to discover intra-dimensional association rules on D^i from $\mathcal{R}'$.

2.3.2 Inter-dimensional association rules

While intra-dimensional association rules describe co-occurrences of elements in only one domain, the inter-dimensional association rules are proposed to find associations or co-occurrences between elements in several domains of a n-ary relation. An inter-dimensional association rule is an implication between elements of a few distinct domains and no dimension is repeated in the rule (i. e., in a rule, there are no two elements that belong to the same domain) [62, 75].

Definition 31 (Inter-dimensional association rule)**.** *$\forall e_i \in D^i$ with $i = 1..n$, $\forall \mathcal{D}', \mathcal{D}'' \subseteq \mathcal{D}$ and $\mathcal{D}' \cap \mathcal{D}'' = \emptyset$, a rule of the form $\wedge_{D^i \in \mathcal{D}'} e_i \rightarrow \wedge_{D^i \in \mathcal{D}''} e_i$ is called an inter-dimensional association rule.*

The extraction of inter-dimensional association rules is often guided by user-defined meta-rules or templates. It means that we find inter-dimensional association rules which match the defined meta-rules and satisfy the given frequency and confidence constraints. Kamber et al. [62] introduced the concept of *metarule-guide* with distinct predicates for mining inter-dimensional association rules from single levels of dimensions. In this study, a metarule is a rule template of the following form

$$P_1 \wedge P_2 \wedge ... \wedge P_m \rightarrow Q_1 \wedge Q_2 \wedge ... \wedge Q_l$$

where P_i $(i = 1, .., m)$ and Q_j $(j = 1, .., l)$ are either instantiated predicates or predicate variables, $p = m + l$ is the number of predicates in the metarule $(p < n)$. All the predicates have distinct predicate names, i. e., no predicate is repeated in the metarule. A rule complies with the metarule iff it can be unified with this metarule.

Example 28. *$\forall x \in person, P_1(x, y) \wedge P_2(x, w) \rightarrow buys(x, \text{“}pentium\text{”})$ is a meta-rule, where P_1 and P_2 are predicate variables, x is a variable representing a person, y and w are object variables. The rule $\forall x \in person, owns(x, \text{“}laptop\text{”}) \wedge income(x, \text{“}high\text{”}) \rightarrow buys(x, \text{“}pentium\text{”})$ is an inter-dimensional association rule complying with the meta-rule. This rule means that if a person owns a laptop and if his/her income is high then he/she tends to buy a pentium computer.*

Frequency and confidence are computed according to the COUNT measure. Given $X \rightarrow Y$ is an inter-dimensional association rule, its frequency (or support) in $\mathcal{R}$ is the probability that the tuples in $\mathcal{R}$ contain both X and Y, its confidence is the probability that a tuple contains Y given that it contains X.

Messaoud et al [75] proposed a general framework for mining inter-dimensional association rules at multiple levels of abstraction. They use the concept of *inter-dimensional meta-rule* which allows users to guide the mining process and focus on a specific context from which rules can be extracted. Given $\mathcal{D}_\mathcal{C} \subset \mathcal{D}$ a subset of s *context dimensions*, a sub-cube on $\mathcal{R}$ according to $\mathcal{D}_\mathcal{C}$ defines the mining context. $\mathcal{D}_\mathcal{A}$ is a subset of *analysis dimensions* from which the predicates of an inter-dimensional association rule are selected. According to these authors, an inter-dimensional meta-rule is a template of the following form:

$$\begin{array}{|l} \text{In the context}(\Theta_1, .., \Theta_s) \\ (\alpha_1 \wedge ... \wedge \alpha_m) \rightarrow (\beta_1 \wedge ... \wedge \beta_l) \end{array}$$

where $(\Theta_1, ..., \Theta_s)$ is a sub-cube on $\mathcal{R}$ according to $\mathcal{D}_\mathcal{C}$. It defines the portion of $\mathcal{R}$ to be mined. When $\mathcal{D}_\mathcal{C} = \emptyset$ the mining process covers the whole $\mathcal{R}$. For all $k = 1, ..., m$ (respectively for all $k = 1, ..., l$), α_k (respectively β_k) is a dimension predicate in a distinct dimension from $\mathcal{D}_\mathcal{A}$, and the number of predicates $p = m+l$ in the meta-rule is equal to the number of dimensions in $\mathcal{D}_\mathcal{A}$.

Example 29. *Assume that a Sales relation (cube) contains the following dimensions: Shop (D^1), Product (D^2), Time (D^3), Profile (D^4), Profession (D^5), Gender (D^6) and Promotion (D^7). Dimension Shop has three levels: All, Continent, and Country. Dimension Product has three levels: All, Family, Article. Dimension Time has two levels: All, Year. Let $\mathcal{D}_\mathcal{C} = \{D^5, D^6\} = \{Profession, Gender\}$ and $\mathcal{D}_\mathcal{A} = \{D^1, D^2, D^3\} = \{Shop, Product, Time\}$. One possible inter-dimensional meta-rule scheme is:*

$$\begin{array}{|l} \textit{In the context } (Student, Female) \\ < e_1 \in Continent > \wedge < e_3 \in Year > \rightarrow < e_2 \in Article > \end{array}$$

According to this inter-dimensional meta-rule, association rules are mined in the sub-cube $(Student, Female)$ *which covers the sales concerning the female students. The dimensions Profile and Promotion do not appear in the rules. For such a rule, its body includes an element in the Continent level of* D^1 *and an element in the Year level of* D^3*, its head includes an element in the Article level of* D^2*. According to this meta-rule, one discovered association rule example can be* $America \wedge 2004 \rightarrow Laptop$.

Let M be a user defined measure (M is an aggregate function, e. g., SUM or COUNT), r be a rule which complies with the defined inter-dimensional meta-rule.

$$r : \left| \begin{array}{l} \text{In the context}(\Theta_1, .., \Theta_s) \\ (x_1 \wedge ... \wedge x_m) \rightarrow (y_1 \wedge ... \wedge y_l) \end{array} \right.$$

Its frequency (or support) and its confidence are defined as follows:

$$f(r) = \frac{M(x_1, \cdots, x_m, y_1, \cdots, y_l, \Theta_1, \cdots, \Theta_s, All, \cdots, All)}{M(All, \cdots, All, \Theta_1, \cdots, \Theta_s, All, \cdots, All)},$$

$$c(r) = \frac{M(x_1, \cdots, x_m, y_1, \cdots, y_l, \Theta_1, \cdots, \Theta_s, All, \cdots, All)}{M(x_1, \cdots, x_s, All, \cdots, All, \Theta_1, \cdots, \Theta_s, All, \cdots, All)}.$$

In this approach, in addition to frequency and confidence, the authors add two descriptive criteria to evaluate the interestingness of mined association rules: Lift criterion and Loevinger criterion (Loev) under the independence hypothesis between the body $X = x_1 \wedge ... \wedge x_m$ and the head $Y = y_1 \wedge ... \wedge y_r$. P_X (respectively P_Y) denotes the relative measure M matching X (respectively Y) in the defined sub-cube $(\Theta_1, .., \Theta_s)$. We denote by $P_{\bar{X}} = 1 - P_X$ (respectively $P_{\bar{Y}} = 1 - P_Y$) the relative measure M not matching X (respectively Y).

$$P_X = \frac{M(x_1, \cdots, x_m, All, \cdots, All, \Theta_1, \cdots, \Theta_s, All, \cdots, All)}{M(All, \cdots, All, \Theta_1, \cdots, \Theta_s, All, \cdots, All)}$$

$$P_Y = \frac{M(All, \cdots, All, y_1, \cdots, y_l, \Theta_1, \cdots, \Theta_s, All, \cdots, All)}{M(All, \cdots, All, \Theta_1, \cdots, \Theta_s, All, \cdots, All)}$$

The Lift and Loev measures of the rule r are defined as follows:

$$Lift(r) = \frac{f(r)}{P_X P_Y}$$

$$Loev(r) = \frac{c(r) - P_Y}{P_{\bar{Y}}}$$

Let $\mathcal{L}_{\text{inter-dimensional}}$ be the set of all inter-dimensional association rules, the collection of interesting inter-dimensional association rules complying a given meta-rule in $\mathcal{R}$ corresponds to the following theory:

$$\mathcal{TH}(\mathcal{R}, \mathcal{L}_{\text{inter-dimensional}}, \mathcal{C}_{\text{meta-rule}} \wedge \mathcal{C}_{\text{frequent}} \wedge \mathcal{C}_{\text{valid}})$$
$$= \{r \in \mathcal{L}_{\text{inter-dimensional}} | \mathcal{C}_{\text{meta-rule}}(r, \mathcal{R}) \wedge \mathcal{C}_{\text{frequent}}(r, \mathcal{R}) \wedge \mathcal{C}_{\text{valid}}(r, \mathcal{R}) \text{ is true}\}.$$

The idea of the algorithms proposed by Kamber [62] and Messaoud [75] for discovering inter-dimensional association rules which comply a given meta-rule and satisfy the minimal frequency and confidence constraints contains two basic steps. First, they use a level-wise search approach as Apriori for finding large predicate sets. It means that they start searching the 1-predicate sets in each dimension and then use the $(k-1)$-predicate sets to enlarge k-predicate sets until p-predicate sets are found. If a k-predicate set is not frequent then all its super predicates are not frequent, then it is pruned and thus it is not used to compute $(k+1)$-predicate sets. Second, they extract inter-dimensional association rules from frequent p-predicate sets with respect to two conditions: (i) An inter-dimensional association rule r must comply with the given meta-rule, and (ii) The rule r must have a confidence greater than the confidence threshold.

2.3.3 Hybrid rules

The absence of repeats is a limitation on the expressiveness of inter-dimensional association rules. Thus, other studies investigated the extracting hybrid rules in which the repetition of few dimensions is enabled.

Missaoui et al. [76] studied the mining triadic association rules from ternary relations. Their approach uses two domains of the ternary relation for analyzing rules, and the remain domain for computing the frequency and the confidence of the rules. Rule computing is based on closed 3-sets and t-generators. Given a ternary relation $\mathcal{R} \subseteq D^1 \times D^2 \times D^3$, where D^1, D^2 and D^3 are respectively object, attribute and condition domains, they propose three types of rules: *attribute* $\times$ *condition association rules*, *conditional attribute association rules* and *attributional condition association rules*.

- *attribute* $\times$ *condition association rule*: An *attribute* $\times$ *condition* association rule is of the form $X \rightarrow Y$ (f, c), where $X, Y \subseteq D^2 \times D^1$, and f (resp. c) is the frequency (resp. the confidence) of the rule.
- *conditional attribute association rule*: A conditional attribute association rule is of the form $(X \rightarrow Y)_C$ (f, c, cov), where $X, Y \subseteq D^2$, $C \subseteq D^3$, f and c are the frequency and the confidence of the rule, $cov = \frac{|C|}{|D^3|}$. The rule means that whenever X occurs under all conditions in C then Y also occurs under the same conditions with a frequency s, a confidence c and a coverage cov.
- *attributional condition association rule*: An attributional condition association rule is of the form $(X \rightarrow Y)_A$ (f, c, cov), where $X, Y \subseteq D^3$, $A \subseteq D^2$, f and c

are the frequency and the confidence of the rule respectively, $cov = \frac{|A|}{|D^2|}$. The rule means that whenever the conditions in X occur for all attributes in A then the conditions in Y also occur for the same attributes with a frequency s, a confidence c and a coverage cov.

In [53], Imieliński et al. propose to compute association rules from data cubes by introducing the concept of *Cubegrade*. They focus on significant changes that affect measures when a cube is modified through specialization, generalization or mutation. Cubegrades are statements which can be interpreted as "what if" formulate about how selected aggregates are affected by various cube modifications. In this study, a pair of the form $dimension = value$ is called a descriptor. A cube depicts a multidimensional view of the data, it is expressed by a set of descriptors, it is the set of tuples in $\mathcal{R}$ satisfying this set of descriptors. A cubegrade is represented in the form:

SourceCube → TargetCube [Measures, Values, Delta-Values]

where SourceCube and TargetCube are cubes, Measures is the set of measures which are evaluated in SourceCube and TargetCube, Value is a function which evaluates measure $m \in Measures$ in the SourceCube, and Delta-Value is a function which computes the ratio of the value of $m \in Measures$ in the TargetCube versus the SourceCube. The considered measures are standard aggregate measures in data warehouses as COUNT, MIN, MAX, SUM, AVG.

They investigate three types of cubegrades:

- *Specializations*: A cubegrade is a *specialization* if the set of descriptors of the target cube is a superset of the set of descriptors of the set of descriptors of the source cube.
- *Generalizations*: A cubegrade is a *generalization* if the set of descriptors of the target cube is a subset of the set of descriptors of the set of descriptors of the source cube.
- *Mutations*: A cubegrade is a *mutation* if the target cube and source cube have the same set of dimensions but differ on the descriptor values.

Example 30. *Consider basket data including the dimensions* $Customer(D^1)$, $Area(D^2)$, $Age(D^3)$, $Income(D^4)$ *and* $Item(D^5)$, $Amount(D^6)$. *Here, Area: where the customer lives (e. g., suburban, urban, rural), Age: the age of the customer, Income: the income of the customer, Item: the product in the supermarket (e. g., milk, cereal, butter, etc), Amount: the amount spent monthly on individual items. In this database,* $(Area = \text{"urban"}, Age = 18\text{-}30)$ *is a cube, it consists of the tuples whose areas are urban and whose ages are between* 18 *to* 30.

- $(Area = \text{“urban“}) \rightarrow (Area = \text{“urban“}, Age = 18\text{-}30)$ $[AVG(salesMilk), AVG(salesMilk) = \$12.40, DeltaAVG(salesMilk) = 80\%]$ *is a specialization cubegrade. It means that the average amount spent on milk by urban buyers drops by* 20% *for buyers whose ages are from 18 to 30.*

- $(Area =$ *"urban"*$, Income = [50k - 70k]) \rightarrow (Area =$ *"urban"*$)$
  $[AVG(salesMilk), AVG(salesMilk) = \$13.78, DeltaAVG(salesMilk) = 90\%]$
  is a generalization cubegrade. It indicates that urban buyers with incomes between \$50000 *to* \$70000 *spend* 10% *more on milk than general urban buyers.*
- $(Area =$ *"suburban"*$) \rightarrow (Area =$ *"urban"*$)$
  $[AVG(salesMilk), AVG(salesMilk) = \$12.40, DeltaAVG(salesMilk) = 70\%]$
  is a mutation cubergrade. The rule means that the average amount spent on milk drops by 30% *when moving from suburban buyers to urban buyers.*

Since for each analysis task, a user is often interested in examining only a small subset of cubes in the given database, Dong et al. [38] propose to extract only cubegrades which satisfy *probe constraints*. A probe constraint enables to select a set of user-desired cubes. They aim at finding all interesting cubegrades (pairs of the form (SourceCube, TargetCube)) which satisfy a significance constraint, a probe constraint and a gradient constraint. The significance constraint is usually defined as conditions on measure attributes, the gradient constraint defines as $\frac{m(\text{TargetCube})}{m(\text{SourceCube})}$.

Tjioe et al. [104] proposed a method for mining only hybrid association rules which satisfy a given template. Based on the multidimensional data organization, their method is able to extract associations from multiple dimensions at multiple levels of abstraction by focusing on summarized data according to the COUNT measure. Assume the user wants to extract rules on a subset of the domains $\mathcal{D}' = \{D^1, D^2, ..., D^m\} \subseteq \mathcal{D}$, and, for each domain $D^i \in \mathcal{D}'$, he/she is interested in an interval value (or a classification value) $d_i(Val)$. All extracted hybrid association rules are in the form:

$$d_1(Val), x_2, ..., x_m \rightarrow y_2, ..., y_m$$

where $\forall i = 2..m$, $x_i, y_i \in d_i(Val)$, the dimension D^1 works as the grouping dimension.

Example 31. *Consider a database having seven dimensions:* $Product(D^1)$, $Time(D^2)$, $Customer(D^3)$, $Channel(D^4)$, $Promotion(D^5)$, $Quantiy(D^6)$ *and* $Dollar\_sold(D^7)$. *Suppose we want to discover interesting hybrid association rules limited to the three dimensions* $Time$, $Customer$ *and* $Product$ *while using the following selection criteria:* $Time(1998-200)$, $Customer(Australia)$ *and* $Product(Cars)$. *Then one hybrid association rule example can be:*

$$\begin{array}{l} Time(1998..2000), Customer(Melb), Product(Car)(\geq 500) \\ \rightarrow Customer(Syd), Product(Car)(\geq 250) \\ (supp = 10\%, conf = 20\%) \end{array} .$$

This rule means that in the years between 1998 and 2000, customers in Melbourne buy 500 *car units then customers in Sydney also buy* 250 *car units with a support of* 10% *of total sales across those dimensions. In those years those customers in*

Melbourne buying 500 *car units, then customers in Sydney also buying* 250 *car units have a confidence* 20%.

The extracting process of interesting hybrid association rules includes two steps: finding all frequent (or significant) cubes and generating hybrid association rules from those frequent cubes. The second step is trivial. For the first step, the algorithms proposed by Imieliński et al. [53] and by Tjioe et al. [104] are based on the idea of a level wise search approach as Apriori and the algorithm proposed by Dong [38] uses a depth-first search.

2.4 Conclusion

We see that the proposed solutions for association rule mining in n-ary relations have some limitations. The intra-dimensional association rule describes co-occurrences of elements of only one dimension. The inter-dimensional association rule allows us to find associations or co-occurrences between elements of several domains of a n-ary relation. However, in an inter-dimension rule, no dimension is repeated (i. e. there are no two elements that belong to the same domain). Therefore, the inter-dimension rule can not discover the correlation between elements in the same domain. The hybrid rule allows to repeat some dimensions. However, in existing studies, the repetition of dimensions in hybrid rules is limited. The first limitation is that, in a rule, the dimensions which appear in its head depend on those in its body. The second limitation, except in [76], is that one dimension can appear in both body and head of a rule, but no dimension can be repeated in each part.

In addition, many of the mined rules are redundant because such a rule suggests the information which is included in another more general rule. So, the user is not interested in the redundant rules.

This dissertation seeks to address these drawbacks. Such a mined rule can contain arbitrary subsets of some domains and the mined rule is non-redundant.

Part II

Contributions

Outline

Mining rules in n-ary relations has attracted some attention in the last few years. However, we have discussed the limitations of the previous work in Section 2.4. As a result, it is not yet possible to discover rules which include arbitrary subsets of some domains. For example, in the ternary relation $\mathcal{R}_E$ (see Figure 10), discovering the following rules is not yet possible:

- $\{p_1, p_2\} \rightarrow \{s_1, s_2\}$,
- $\{p_4\} \times \{s_1\} \rightarrow (\{p_1, p_2\}) \vee (\{p_2, p_3\} \times \{s_3\}) \vee (\{p_3\})$.

The first rule suggests that the products p_1 and p_2 are bought together in both seasons s_1 and s_2. The second one suggests that, if a customer buys the product p_4 in the season s_1 then he/she tends to buy the products p_1, p_2 or p_3 in this same season. If we manage to have a framework for mining such rules, then we offer to practionners better tools to describe and to analyze the more or less hidden relationships within n-ary relations.

Our goal is to generalize association rule mining within n-ary relations ($n > 2$), yet getting the standard semantics of the standard association rules when $n = 2$. We propose that our rules can include arbitrary subsets of some domains from the targeted n-ary relations. Thanks to this generalization, to the best of our knowledge, the previous rule models that have been studied earlier become special cases of our current proposal.

This generalization is however surprisingly difficult. The two main subproblems to address are (a) How to define the semantics of the rules thanks to primitive constraints, and (b) Their efficient computation. Point (a) is about defining the pattern language and objective interestingness measures for rules. When generalized to n-ary relations, rules may involve arbitrary subsets of some domains. In this context, what does it means for a rule to be frequent or to have enough confidence? Is it possible to have measures that correspond to the special case of standard measures when $n = 2$ and that are as intuitive as possible for analysts? How to generalize other relevancy concepts such as, for example, non redundancy? Once these declarative

issues are understood, Point (b) concerns the design of scalable methods to extract the patterns that satisfy a given conjunction of primitive constraints. When possible, correct and complete algorithms remain preferable: such methods list all solution patterns and only them. Performance issues are important: a good algorithm must scale in the number of dimensions, in the domain size (of each dimension), and in the number of tuples in the relation. In Part 2, we investigate two types of rules that have been called multidimensional association rules (Chapter 3) and multidimensional disjunctive rules (Chapter 4).

Chapter 3

Generalizing association rules in n-ary relations

In this chapter, we generalize the association rule mining task [2] to arbitrary n-ary relations. Our contribution is twofold. The first contribution is the design of the pattern domain of multidimensional association rules. Such a rule is an implication between two associations where each association can contain subsets of some arbitrary domains. In this context, we provide three objective interestingness measures for such rules: frequency, exclusive confidence and natural confidence. We also revisit the concept of non-redundant rules having a minimal body and a maximal head in our extended setting. The second contribution is the design of the first complete algorithm, namely PINARD++ [1], which lists the collection of *a priori* interesting rules.

The chapter is organized as follows. The next section provides the basic concepts to build the new pattern domain of multidimensional association rules. In Section 3.2, we define the language for such rules and we design interestingness measures for them. Section 3.3 describes our algorithm that computes a priori interesting rules on a n-ary relation. Section 3.4 reports experimental results on a real-life ternary relation. Section 3.5 summarizes the chapter.

3.1 Basic concepts

We generalize the concept of *itemsets* in a binary relation to the concept of *associations* in a n-ary relation because we build the new pattern domain of multidimensional association rules on the relationships between associations.

In a binary relation that describes the relationship between two domains only, an itemset is a subset of one domain, and its frequency (say its "strength") is computed on the other domain (see Section 1.2). Let us propose a generalization when defining

1. PINARD Is N-ary Association Rule Discovery.

associations within a n-ary relation. We consider that an association can involve subsets of the different domains and that it strenght should be defined in terms of the remaining ones, i.e., the domains that are not involved. For example, in a 3-ary relation $Products \times Seasons \times Customers$, an association can be a set of products, or a set of seasons, but it can also consist of both products and seasons, etc. In the context of arbitrary n-ary relations, *how do we express such associations?*, *How do we specify the objectif interestingness of such associations?*. We address these questions and we provide the operators on associations that will be used from now.

Given an arbitrary number $n \in \mathbb{N}$, we recall that the set of n domains is denoted by $\mathcal{D} = \{D^1, \ldots, D^n\}$, and that the n-ary relation in which patterns are to be discovered is $\mathcal{R} \subseteq D^1 \times \cdots \times D^n$. To emphasize the relevancy of the proposed patterns, the definitions are illustrated on the toy ternary relation $\mathcal{R}_E$ (see Figure 10 in Chapter 2) whose tabular representation is recalled here :

	p_1	p_2	p_3	p_4	p_1	p_2	p_3	p_4	p_1	p_2	p_3	p_4	p_1	p_2	p_3	p_4
o_1	1	1			1	1		1		1	1			1	1	1
o_2	1	1		1	1	1					1					
o_3	1	1	1	1						1	1	1		1	1	
o_4	1		1					1							1	1
o_5		1	1	1			1		1	1	1	1	1		1	1
	s_1				s_2				s_3				s_4			

It relates products in $D^1 = \{p_1, p_2, p_3, p_4\}$, seasons in $D^2 = \{s_1, s_2, s_3, s_4\}$, and customers in $D^3 = \{o_1, o_2, o_3, o_4, o_5\}$.

In an n-ary relation, an *association* on $\mathcal{D}' \subseteq \mathcal{D}$ is the Cartesian product of subsets of the domains in $\mathcal{D}'$. Without loss of generality, the dimensions are assumed to be ordered such that $\mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\}$.

Definition 32 (Association)**.** *$\forall \mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\} \subseteq \mathcal{D}$, $\times_{i=1..|\mathcal{D}'|} X^i$ is an association on $\mathcal{D}'$ iff $\forall i = 1..|\mathcal{D}'|$, $X^i \neq \emptyset \wedge X^i \subseteq D^i$. By convention, the only association on $\mathcal{D}' = \emptyset$ is denoted by $\emptyset$.*

Example 32. *In $\mathcal{R}_E$, $\{p_1, p_2\} \times \{s_1\}$ and $\{p_1, p_2\} \times \{s_1, s_2\}$ are associations on $\{D^1, D^2\}$, $\{p_1, p_2\}$ is an association on $\{D^1\}$, and $\{s_1, s_3\}$ is an association on $\{D^2\}$.*

In binary relations (like for instance the relation $\mathcal{B}_E$ in Chapter 1), the support domain of any itemset (e.g., a set of products) is a set of objects (e.g., a set of customers), i.e., the remaining dimension. We introduced the function ψ that associate to each itemset its supporting set and that can be used, for instance, to evaluate the itemset frequency. Let us generalize this to associations.

Definition 33 (Support domain of an association)**.** *Given an arbitrary association $\times_{i=1..|\mathcal{D}'|} X^i$ on $\mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\} \subseteq \mathcal{D}$, its* support domain *in the n-ary relation $\mathcal{R}$ is $\times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$.*

Example 33. *In $\mathcal{R}_E$,, the support domain of $\{p_1, p_2\} \times \{s_1\}$ and $\{p_1, p_2\} \times \{s_1, s_2\}$ is D^3, that of $\{p_1, p_2\}$ is $D^2 \times D^3$ and that of $\{s_1, s_3\}$ is $D^1 \times D^3$.*

The *support* of an association is a subset of its support domain. Its definition uses concatenation denoted by '$\cdot$'. For instance, $(p_2, s_1) \cdot (o_1) = (p_2, s_1, o_1)$.

Definition 34 (Support of an association)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, let X be an association on $\mathcal{D}'$. Its support is*

$$s(X) = \{u \in \times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i \mid \forall x \in X,\ x \cdot u \in \mathcal{R}\}.$$

Let us mention some special cases. An association involving the n domains ($\mathcal{D}' = \mathcal{D}$) is either false (at least one n-tuple it contains is absent from $\mathcal{R}$) or true (every n-tuple it contains is in $\mathcal{R}$). By using the convention $\times_{D^i \in \emptyset} D^i = \{\epsilon\}$ (where ϵ is the empty word), Definition 34 reflects that every possible association on $\mathcal{D}$ has either zero or one element, ϵ, in its support. The opposite extreme case is the support of the empty association, $s(\emptyset)$, which is $\mathcal{R}$.

The support of an association generalizes that of an *itemset* (see the operator ψ in Chapter 1). Indeed, when considering associations in a relation $\mathcal{O} \times \mathcal{P}$, if we choose $\mathcal{D}' = \{\mathcal{P}\}$, the support domain is $\mathcal{O}$ and ψ gives the support of any association on $\mathcal{P}$.

Example 34. *Let us give examples of supports for three associations in $\mathcal{R}_E$.*

- $s(\{p_1, p_2\} \times \{s_1\}) = \{o_1, o_2, o_3\}$,
- $s(\{p_1, p_2\} \times \{s_1, s_2\}) = \{o_1, o_2\}$,
- $s(\{p_1, p_2\}) = \{(s_1, o_1), (s_1, o_2), (s_1, o_3), (s_2, o_1), (s_2, o_2), (s_3, o_5)\}$.

Let us now introduce some operators to manipulate associations. Their definitions are illustrated for $X_e = \{p_2, p_3\}$ and $Y_e = \{p_1, p_2\} \times \{s_1, s_2\}$.

Definition 35 (Projection π)**.** *$\forall \mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\} \subseteq \mathcal{D}$, let $X = X^1 \times \cdots \times X^{|\mathcal{D}'|}$ be an association on $\mathcal{D}'$. $\forall D^i \in \mathcal{D}$, $\pi_{D^i}(X) = X^i$ if $D^i \in \mathcal{D}'$, $\emptyset$ otherwise.*

Example 35. *$\pi_{D^1}(X_e) = \{p_2, p_3\}$, $\pi_{D^2}(X_e) = \emptyset$, $\pi_{D^3}(X_e) = \emptyset$, $\pi_{D^1}(Y_e) = \{p_1, p_2\}$, $\pi_{D^2}(Y_e) = \{s_1, s_2\}$, and $\pi_{D^3}(Y_e) = \emptyset$.*

Definition 36 (Union $\sqcup$)**.** *$\forall \mathcal{D}_X \subseteq \mathcal{D}$ and $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, let X (resp. Y) be an association on $\mathcal{D}_X$ (resp. on $\mathcal{D}_Y$). $X \sqcup Y$ is an association on $\mathcal{D}_X \cup \mathcal{D}_Y$ for which $\forall D^i \in \mathcal{D}$, $\pi_{D^i}(X \sqcup Y) = \pi_{D^i}(X) \cup \pi_{D^i}(Y)$.*

Example 36. *$X_e \sqcup Y_e$ is an association on $\{D^1, D^2\}$ ($= \{D^2\} \cup \{D^1, D^2\}$), $X_e \sqcup Y_e = (\pi_{D^1}(X_e) \cup \pi_{D^1}(Y_e)) \times (\pi_{D^2}(X_e) \cup \pi_{D^2}(Y_e)) = (\{p_2, p_3\} \cup \{p_1, p_2\}) \times (\emptyset \cup \{s_1, s_2\}) = \{p_1, p_2, p_3\} \times \{s_1, s_2\}$.*

Definition 37 (Complement $\setminus$)**.** *$\forall \mathcal{D}_X \subseteq \mathcal{D}$ and $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, let X (resp. Y) be an association on $\mathcal{D}_X$ (resp. on $\mathcal{D}_Y$). $Y \setminus X$ is an association on $\{D^i \in \mathcal{D}_Y \mid \pi_{D^i}(Y) \not\subseteq \pi_{D^i}(X)\}$ for which $\forall D^i \in \mathcal{D}$, $\pi_{D^i}(Y \setminus X) = \pi_{D^i}(Y) \setminus \pi_{D^i}(X)$.*

Example 37. *$Y_e \setminus X_e$ is an association on $\{D^1, D^2\}$, $Y_e \setminus X_e = (\pi_{D^1}(Y_e) \setminus \pi_{D^1}(X_e)) \times (\pi_{D^2}(Y_e) \setminus \pi_{D^2}(X_e)) = (\{p_1, p_2\} \setminus \{p_2, p_3\}) \times (\{s_1, s_2\} \setminus \emptyset) = \{p_1\} \times \{s_1, s_2\}$. In contrast, $X_e \setminus Y_e$ is an association on $\{D^1\}$ only and $X_e \setminus Y_e = \pi_{D^1}(X_e) \setminus \pi_{D^1}(Y_e) = \{p_2, p_3\} \setminus \{p_1, p_2\} = \{p_3\}$.*

Definition 38 (Inclusion $\sqsubseteq$)**.** *$\forall \mathcal{D}_X \subseteq \mathcal{D}$ and $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, let X (resp. Y) be an association on $\mathcal{D}_X$ (resp. on $\mathcal{D}_Y$). X is included in Y, denoted $X \sqsubseteq Y$, iff $\forall D^i \in \mathcal{D}$, $\pi_{D^i}(X) \subseteq \pi_{D^i}(Y)$.*

Example 38. *There are inclusions between three of the four associations illustrating Definition 32: $\{p_1, p_2\} \sqsubseteq \{p_1, p_2\} \times \{s_1\} \sqsubseteq \{p_1, p_2\} \times \{s_1, s_2\}$.*

With this generalized inclusion, the *anti-monotonicity* of the support cardinality, which is well known in itemset mining, still holds. The proof is given in the Appendix A at the end of the thesis.

Theorem 5 (Support anti-monotonicity)**.** *$\forall \mathcal{D}_X \subseteq \mathcal{D}$ and $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, let X (resp. Y) be an association on $\mathcal{D}_X$ (resp. on $\mathcal{D}_Y$), $X \sqsubseteq Y \Rightarrow |s(X)| \geq |s(Y)|$.*

Example 39. *Considering the double inclusion illustrating Definition 38, one can verify that $|s(\{d_1, d_2\})| \geq |s(\{d_1, d_2\} \times \{a_1\})| \geq |s(\{d_1, d_2\} \times \{a_1, a_3\})|$, i. e., Theorem 5 holds.*

3.2 Multidimensional association rules

We now define the pattern domain of multidimensional association rules in n-ary relations. Examples of such rules, therefore introducing the pattern language, have been intuitively discussed. For instance, we may discover $\{p_1, p_2\} \rightarrow \{s_1, s_2\}$ or $\{p_3\} \times \{s_3, s_4\} \rightarrow \{p_2\}$ in the ternary relation $\mathcal{R}_E$. The first rule tells that the products p_1 and p_2 are bought together both seasons s_1 and s_2. The second rule would mean that the customers who buy the product p_3 in the seasons s_3 and s_4 also tend to buy p_2 in these seasons. To provide a semantics, we design measures that evaluate how significant are the relations between the associations in both sides of the implication sign. In fact, we have to look for the n-dimensional counterpart of the fequency and confidence measures that are so popular in the context of binary relation mining. Other aspects related to relevancy and/or computational feasability are considered as well (e.g., non redundancy, scalability).

3.2.1 Definitions

Given an n-ary relation $\mathcal{R}$ on $\mathcal{D} = \{D^1, \ldots, D^n\}$, a *multidimensional association rule* on $\mathcal{D}' \subseteq \mathcal{D}$ specifies a relationship between two associations whose union is an association on $\mathcal{D}'$. It is simply called a rule when it is clear from the context.

Definition 39 (Multidimensional association rule)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, a multidimensional association rule on $\mathcal{D}'$ is a pattern of the form $X \to Y$, where X and Y are associations on subsets of $\mathcal{D}'$ and $X \sqcup Y$ is an association on $\mathcal{D}'$. X is called the body and Y is called the head.*

Example 40. *In $\mathcal{R}_E$, $\{p_1, p_2\} \to \{s_1, s_2\}$ and $\{p_3\} \times \{s_3, s_4\} \to \{p_2\}$ are two rules on $\{D^1, D^2\}$. $\{p_1\} \to \{p_2\}$ is not a rule on $\{D^1, D^2\}$ because no element in D^2 appears in its body nor in its head. It is however a rule on $\{D^1\}$.*

In the binary case, we know many measures that assess the strength and the type of relationships between the itemset in the body and that in the head [101, 44]. Many measures are however based on frequency. In the context of n-ary relations, it turns out that a natural definition of rule frequency exists.

3.2.2 A frequency measure

The frequency of a rule enable to tell how often a rule is applicable to a given data set. For this, we consider the support domain of a rule, i.e., the Cartesian product of the domains which do not appear in the rule. For a rule $X \to Y$ on $\mathcal{D}' \subseteq \mathcal{D}$, its support domain is $\times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$. In other terms, the support domain of the rule is that of the association $(X \sqcup Y)$ and we can define the (relative) frequency of the rule as, in the support domain, the proportion of elements supporting the union of its body and its head.

Definition 40 (Frequency)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, let $X \to Y$ be a rule on $\mathcal{D}'$. Its frequency is:*

$$f(X \to Y) = \frac{|s(X \sqcup Y)|}{|\times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i|}.$$

Example 41. *Let us consider the rules $r_1 : \{p_1, p_2\} \to \{s_1, s_2\}$ and $r_2 : \{p_3\} \times \{s_3, s_4\} \to \{p_2\}$ in $\mathcal{R}_E$.*

- $f(r_1) = \frac{|s(\{p_1,p_2\} \sqcup \{s_1,s_2\})|}{|D^3|} = \frac{|s(\{p_1,p_2\} \times \{s_1,s_2\})|}{|D^3|} = \frac{|\{o_1,o_2\}|}{|\{o_1,o_2,o_3,o_4,o_5\}|} = \frac{2}{5}$;
- $f(r_2) = \frac{|s(\{p_3\} \times \{s_3,s_4\} \sqcup \{p_2\})|}{|D^3|} = \frac{|s(\{p_2,p_3\} \times \{s_3,s_4\})|}{|D^3|} = \frac{|\{o_1,o_3\}|}{|\{o_1,o_2,o_3,o_4,o_5\}|} = \frac{2}{5}$.

The frequency of r_1 is the proportion of customers who buy the products p_1 and p_2 together in the both seasons s_1 and s_2. Similarly, the frequency of r_2 is the proportion of customers who buy the products p_2 and p_3 together in both seasons s_3 and s_4.

Frequency is an important measure because a rule that has very low frequency may occur simply by chance. For example, suppose the frequency of r_1 in R_E is low, then the rule may be uninteresting from a business perspective because it may not be profitable to promote the products p_1 and p_2 in the seasons s_1 and s_1 when customers seldom buy them together. For this reasons, frequency is often used to eliminate uninteresting rules. However, to be honest, enforcing a minimal frequency is also a key issue to achieve the scalability of correct and complete computation of frequent patterns in real data. It was true for itemsets and association rules in binary relations, it is true for our multidimensional rules as well.

We may now look for futher objective interestingness measures like a generalization of the confidence measure. Looking for other measures based on frequencies is clearly out of the scope of this thesis. Indeed, the fundamental counting problems already arise with the confidence computation and have to be understood beforehand.

3.2.3 Confidence measures

The Problem

Is it possible and useful to directly generalize the confidence measure of association rules in binary relations to n-ary relations? In other terms, can we say that the confidence of a rule $X \to Y$ is defined as:

$$\frac{|s(X \sqcup Y)|}{|s(X)|}.$$

If X and $X \sqcup Y$ are associations on the same domains (so they have the same support domain), this definition is intuitive: the confidence is a proportion of elements in a same support domain. For instance, in $\mathcal{R}_E$, the confidence of $\{p_3\} \times \{s_3, s_4\} \to \{p_2\}$ would be:

$$\frac{|s(\{p_3\} \times \{s_3, s_4\} \sqcup \{p_2\})|}{|s(\{p_3\} \times \{s_3, s_4\})|} = \frac{|s(\{p_2, p_3\} \times \{s_3, s_4\})|}{|s(\{p_3\} \times \{s_3, s_4\})|} = \frac{|\{o_1, o_3\}|}{|\{o_1, o_3, o_5\}|} = \frac{2}{3}.$$

It is a proportion of customers and it means that the customers who buy p_3 during both s_3 and s_4 also tend to buy p_2 during these seasons.

Nevertheless, this semantics is not satisfactory for any rule whose head involves some dimension that is not in its body. Indeed, in this case, $s(X \sqcup Y)$ and $s(X)$ are disjoint sets and the ratio of their cardinalities does not make any sense. For instance, in $\mathcal{R}_E$, consider the rule $\{p_1, p_2\} \to \{s_1, s_2\}$, $s(\{p_1, p_2\} \times \{s_1, s_2\})$ is a set of customers (it is $\{o_1, o_2\}$) whereas $s(\{p_1, p_2\})$ is not (it contains couples such as (s_1, o_1), or (s_2, o_1)). As a result, there is a need for a new confidence measure that would make sense for any multidimensional association rule $X \to Y$. When X and

$X \sqcup Y$ are defined on the same domain(s), we would like to measure its confidence by means of $|s(X \sqcup Y)|/|s(X)|$.

We propose two solutions to this problem. The first solution is to compute the confidence of $X \to Y$ on the support domain of X. The proposed confidence measure is called an *exclusive confidence*. The idea is to introduce a new factor that is multiplied with $|s(X \sqcup Y)|$ such that this multiplication and $|s(X)|$ become comparable. The second solution is to compute the confidence of $X \to Y$ on the support domain of $(X \sqcup Y)$. In this case, the confidence measure is called *natural confidence*. The idea here is introduce a new definition of the support of X when considering the support domain of $(X \sqcup Y)$.

Exclusive confidence

Computing the confidence of a rule $X \to Y$ on $\mathcal{D}'$ is problematic if X is defined on a set $\mathcal{D}_X$ *strictly* included in $\mathcal{D}'$. The idea to solve this problem is to multiply $|s(X \sqcup Y)|$ by the cardinality of the projection of Y on the domains that are absent from $\mathcal{D}_X$, i. e., $| \times_{D^i \in \mathcal{D}' \setminus \mathcal{D}_X} \pi_{D^i}(Y)|$.

Let us observe that $s(X)$ and $s(X \sqcup Y) \times (\times_{D^i \in \mathcal{D}' \setminus \mathcal{D}_X} \pi_{D^i}(Y))$ are the same domains. Therefore, $|s(X)|$ and $|s(X \sqcup Y)| \times | \times_{D^i \in \mathcal{D}' \setminus \mathcal{D}_X} \pi_{D^i}(Y)|$ are comparable and the exclusive confidence of $X \to Y$ is the proportion of these two values.

When the exclusive confidence of $X \to Y$ is high, it means that X "prefers" to be "connected" with Y than being "connected" with other elements.

Definition 41 (Exclusive confidence)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, let $X \to Y$ be a rule on $\mathcal{D}'$ and $\mathcal{D}_X$ be the domains on which X is defined. The exclusive confidence of this rule is:*

$$c_{exclusive}(X \to Y) = \frac{|s(X \sqcup Y)| \times | \times_{D^i \in \mathcal{D}' \setminus \mathcal{D}_X} \pi_{D^i}(Y)|}{|s(X)|} \ .$$

Roughly speaking, the remedial factor $| \times_{D^i \in \mathcal{D}' \setminus \mathcal{D}_X} \pi_{D^i}(Y)|$, applied to $|s(X \sqcup Y)|$, allows to count the elements at the numerator of the fraction "in the same way" as those at the denominator. As desired above, if X is an association on $\mathcal{D}'$, the exclusive confidence of $X \to Y$ is $|s(X \sqcup Y)|/|s(X)|$ under the convention $\times_{D^i \in \emptyset} \pi_{D^i}(Y) = \{\epsilon\}$.

Example 42. *Consider the rule $\{p_1, p_2\} \to \{s_1, s_2\}$ in $\mathcal{R}_E$ and let us name a* transaction *the purchase of a customer during a specific season. There are two customers, o_1 and o_2, who buy both products p_1 and p_2 during both seasons s_1 and s_2, i. e., we have $|\{o_1, o_2\}| \times |\{s_1, s_2\}| = 4$ transactions. Consider now the body of the rule, i. e., $\{p_1, p_2\}$. Six transactions, (s_1, o_1), (s_1, o_2), (s_1, o_3), (s_2, o_1), (s_2, o_2) and (s_3, o_5), involve both p_1 and p_2. Thus,*

$$c_{exclusive}(\{p_1, p_2\} \to \{s_1, s_2\}) = \frac{|s(\{p_1, p_2\} \times \{s_1, s_2\})| \times |\{s_1, s_2\}|}{|s(\{p_1, p_2\})|} = \frac{4}{6} \ .$$

Customer o_3 buys both products p_1 and p_2 during season s_1, whereas he/she does not buy them together during season s_2. This actually lowers the confidence in the fact that customers like buying both products during both seasons s_1 and s_2. Notice also that customer o_5 buying these two products during season s_3 lowers the confidence as well. In fact, the exclusive confidence $c_{exclusive}(\{p_1, p_2\} \to \{s_1, s_2\})$ indicates to what extent products p_1 and p_2 are bought together during both seasons s_1 and s_2 only*. This* exclusivity *explains the chosen name. If $c_{exclusive}(\{p_1, p_2\} \to \{s_1, s_2\})$ was* 1*, every customer who buys p_1 and p_2 together would* always *do so during both seasons s_1 and s_2 (and* never *during another season).*

The exclusive confidence measure actually penalizes a rule whose elements in its support domain individually allow to conclude on other elements than those at its head. In this way, a minimal exclusive confidence threshold favors the discovery of multidimensional association rules with "maximal" heads. Unfortunately, this exclusivity also makes the function $X \mapsto c_{\text{exclusive}}(X \to Y \setminus X)$ (with $X \sqsubseteq Y$) does not increase w.r.t. $\sqsubseteq$.

Example 43. *Consider the rules $\{s_1, s_3\} \to \{p_2, p_3, p_4\}$ and $\{p_2\} \times \{s_1, s_3\} \to \{p_3, p_4\}$ in $\mathcal{R}_E$, $c_{exclusive}(\{s_1, s_3\} \to \{p_2, p_3, p_4\}) = \frac{6}{7}$ and $c_{exclusive}(\{p_2\} \times \{s_1, s_3\} \to \{p_3, p_4\}) = \frac{2}{3}$. We observe that $\{s_1, s_3\} \sqsubseteq \{p_2\} \times \{s_1, s_3\} \sqsubseteq \{p_2, p_3, p_4\} \times \{s_1, s_3\}$. However $c_{exclusive}(\{s_1, s_3\} \to \{p_2, p_3, p_4\})$ is greater than $c_{exclusive}(\{p_2\} \times \{s_1, s_3\} \to \{p_3, p_4\})$.*

This prevents to efficiently list every rule whose exclusive confidence is greater than a user-defined threshold. Let us now consider an alternative definition for the confidence measure.

Natural confidence

To define the confidence of $X \to Y$, a straightforward generalization of the binary case is not possible when the support domains of X and $X \sqcup Y$ are different. Enforcing the support of X to be a subset of the support domain $\times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$ of $X \sqcup Y$ allows to define a confidence measure that is a *natural* proportion, i. e., a proportion of elements in a same support domain. The *natural confidence* of $X \to Y$ is the probability to observe Y when X holds on the support domain of $X \sqcup Y$. The cost of such a natural confidence is the need for a new definition of the support when applied to rule bodies.

Definition 42 (Natural support of bodies)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, let $X \to Y$ be a rule on $\mathcal{D}'$. The natural support of X is*

$$s_{\mathcal{D} \setminus \mathcal{D}'}(X) = \{u \in \times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i \mid \exists w \in \times_{D^i \in \mathcal{D}' \setminus \mathcal{D}_X} D^i \text{ s.t. } \forall x \in X,\ x \cdot w \cdot u \in \mathcal{R}\}\ ,$$

where $\mathcal{D}_X$ is the set of domains on which X is defined. For $x \cdot w \cdot u$ to possibly be in $\mathcal{R}$, the domains in $\mathcal{D}_X$ must appear first, i. e., the domain index may have to be changed.

Definition 43 (Natural confidence)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, let $X \to Y$ be a rule on $\mathcal{D}'$. Its natural confidence is*

$$c_{natural}(X \to Y) = \frac{|s(X \sqcup Y)|}{|s_{\mathcal{D}\setminus\mathcal{D}'}(X)|} \ .$$

Notice that if X is an association on $\mathcal{D}'$, the natural confidence of $X \to Y$ is $|s(X \sqcup Y)|/|s(X)|$ under the convention $\times_{D^i \in \emptyset} D^i = \{\epsilon\}$.

Example 44. *Once again, consider the rule $\{p_1, p_2\} \to \{s_1, s_2\}$ in $\mathcal{R}_E$. The customers who buy the products p_1 and p_2 together (during at least one season) are o_1, o_2, o_3, and o_5. Among them, only o_1 and o_2 buy p_1 and p_2 during both seasons s_1 and s_2. Thus,*

$$c_{natural}(\{p_1, p_2\} \to \{s_1, s_2\}) = \frac{|s(\{p_1, p_2\} \times \{s_1, s_2\})|}{|s_{\{D^3\}}(\{p_1, p_2\})|} = \frac{|\{o_1, o_2\}|}{|\{o_1, o_2, o_3, o_5\}|} = \frac{2}{4} \ .$$

It means that a half of the customers buying both p_1 and p_2 during a same season do so during both seasons s_1 and s_2. Now, the customers who support the rule can buy both p_1 and p_2 during another season and that does not "lower" the natural confidence, whereas it does lower the exclusive one.

The natural confidence has a monotonicity property which the exclusive confidence misses. It can give rise to the efficient discovery of multidimensional association rules in large datasets.

Theorem 6 (Pruning criterion)**.** *Let $X \to Y \setminus X$ and $X' \to Y \setminus X'$ be two rules on $\mathcal{D}'$. We have $X' \sqsubseteq X \sqsubseteq Y \Rightarrow c_{natural}(X' \to Y \setminus X') \leq c_{natural}(X \to Y \setminus X)$.*

The proof is given in the Appendix A.

Example 45. *In $\mathcal{R}_E$, $\{p_1, p_2\} \to \{s_1, s_2\}$ and $\{p_1, p_2\} \times \{s_1\} \to \{s_2\}$ are two rules on $\{D^1, D^2\}$. The natural confidence of the first rule is $\frac{2}{4}$ (see above). The natural confidence of the second one is:*

$$\frac{|s(\{p_1, p_2\} \times \{s_1, s_2\})|}{|s_{D^3}(\{p_1, p_2\} \times \{s_1\})|} = \frac{|\{o_1, o_2\}|}{|\{o_1, o_2, o_3\}|} = \frac{2}{3}.$$

It illustrates Theorem 6. Indeed, $\{p_1, p_2\} \sqsubseteq \{p_1, p_2\} \times \{s_1\} \sqsubseteq \{p_1, p_2\} \times \{s_1, s_2\}$ and $c_{natural}(\{p_1, p_2\} \to \{s_1, s_2\}) \leq c_{natural}(\{p_1, p_2\} \times \{s_1\} \to \{s_2\})$.

In Section 3.3, this is used to prune the search space where no rule can satisfy a minimal natural confidence constraint.

3.2.4 Non-redundancy

Definition 44 (Syntactic equivalence of rules)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, the rules $X \to Y$ and $X \to Z$ on $\mathcal{D}'$ are syntactically equivalent iff $X \sqcup Y = X \sqcup Z$.*

Proving the following lemma is straightforward.

Lemma 2. *Syntactically equivalent rules have the same frequency, the same exclusive confidence and the same natural confidence.*

Definition 45 (Canonical rule)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, a rule $X \to Y$ on $\mathcal{D}'$ is canonical iff $\forall D^i \in \mathcal{D}$, $\pi_{D^i}(X) \cap \pi_{D^i}(Y) = \emptyset$.*

Any complete collection of rules satisfying constraints on frequency and/or confidences can be condensed, without any loss of information, into its canonical rules only. Indeed, given a canonical rule $X \to Y$ in the collection, Lemma 2 entails that all syntactically equivalent rules necessary are in the collection as well. Moreover constructing them is easy: they are the rules $X \to Y \sqcup Z$ with $Z \sqsubseteq X$.

Not all rules satisfying the minimum thresholds of frequency and confidences are interesting. We do not investigate here the use of other objective interestingness measures but the crucial issue of rule redundancy.

Example 46. *In $\mathcal{R}_E$, let us consider the following rules:*

- *r_3: $\{s_1, s_3\} \to \{p_2, p_3, p_4\}$ ($f : 0.4$, $c_{natural} : 0.67$, $c_{exclusive} : 0.86$),*
- *r_4: $\{p_2\} \times \{s_1, s_3\} \to \{p_3\}$ ($f : 0.4$, $c_{natural} : 0.67$, $c_{exclusive} : 0.67$),*
- *r_5: $\{p_1\} \times \{s_2\} \to \{p_2\} \times \{s_1\}$ ($f : 0.4$, $c_{natural} : 1$, $c_{exclusive} : 1$),*
- *r_6: $\{p_1\} \times \{s_1, s_2\} \to \{p_2\}$ ($f : 0.4$, $c_{natural} : 1$, $c_{exclusive} : 1$),*

They all are canonical and their frequencies, their exclusive confidences and their natural confidences respectively exceed 0.4, 0.6, *and* 0.6. *In this regard, they* individually *satisfy this aspect of interestingness. Nevertheless,* altogether*, they provide redundant information. For instance, r_4 is more specific than r_3 because it requires more condition to apply (the purchases must involve p_2) and its conclusion is less informative (it does not tell anything on p_4). However this specialization does not grant r_4 a greater frequency or greater confidences than r_3. Therefore r_4 is said to be redundant. Similarly, by the existence of r_5, r_6 is redundant. Since the analyst would not find any added-value in the rules r_4 and r_6, they should not be returned.*

We generalize the concept of non-redundant rule having minimal body and maximal head [82] within our multidimensional setting.

Definition 46 (Non-redundant rule)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, a rule $X \to Y$ on $\mathcal{D}'$ is non-redundant iff it is canonical and no other canonical rule $X' \to Y'$ is such that:*

$$\begin{cases} ((X' \sqcup Y' = X \sqcup Y) \wedge (X' \sqsubset X)) \vee ((X' \sqcup Y' \sqsupset X \sqcup Y) \wedge (X' \sqsubseteq X)) \\ f(X' \to Y') \geq f(X \to Y) \\ c_{exclusive}(X' \to Y') \geq c_{exclusive}(X \to Y) \\ c_{natural}(X' \to Y') \geq c_{natural}(X \to Y) \end{cases}.$$

The first condition defines the form of the rules that may be redundant. Obviously, there exists other more general rules (with less elements) that are not matched. Nevertheless, this definition allows to remove many redundant rules that are worse that the selected ones in term of frequency (second condition), exclusive confidence (third condition) and natural confidence (fourth condition). For instance, the rules r_4 and r_6 are not presented to the analyst. The choice of the first condition was partly based on procedural considerations: the non-redundant rules, as defined above, can be efficiently derived from closed sets.

Let us introduce the relation in which these patterns are extracted. It is obtained from $\mathcal{R}$ by "flattening" the dimensions which are absent from $\mathcal{D}'$ into a unique support dimension $D^{\text{supp}} = \times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$. Denoted $\mathcal{R}_A$, this relation is defined on the domains $\mathcal{D}_A = \mathcal{D}' \cup \{D^{\text{supp}}\}$. Assuming that for all $i = 1..n$, e_i is an element of the i^{th} domain, i. e., $e_i \in D^i$, we have to build:

$$\mathcal{R}_A = \{(e_1, e_2, \ldots, e_{|\mathcal{D}'|}, (e_{|\mathcal{D}'|+1}, \ldots, e_n)) | (e_1, e_2, \ldots, e_{|\mathcal{D}'|}, e_{|\mathcal{D}'|+1}, \ldots, e_n) \in \mathcal{R}\}.$$

To adapt notations of association, we rewrite the definition of closed sets (Definition 26). Indeed, it does not affect the properties of closed sets if we write a closed set $(X^1 \times X^2 \times ... \times X^n)$ instead of $(X^1, X^2, ..., X^n)$. Therefore, the definition of a closed set (Definition 26) is equivalent to the following one.

Definition 47 (Closed set)**.** *Given a relation $\mathcal{R}_A$ on $\mathcal{D}_A$, X is a closed set in $\mathcal{R}_A$ iff* $\begin{cases} X \subseteq \mathcal{R}_A \\ \forall D^i \in \mathcal{D}_A, \forall e \in D^i \setminus \pi_{D^i}(X), X \sqcup \{e\} \not\subseteq \mathcal{R}_A \end{cases}$.

Example 47. *Considering $\mathcal{R}_E$, if $\mathcal{D}'$ contains two domains, then $\mathcal{R}_A = \mathcal{R}_E$ and $\{p_1, p_2\} \times \{s_1, s_2\} \times \{o_1, o_2\}$ is a closed set. $\{p_1, p_2\} \times \{s_1, s_2\} \times \{o_1, o_2, o_3\}$ is not a closed set because it covers $(p_1, s_2, o_3) \notin \mathcal{R}_A$. $\{p_1, p_2\} \times \{s_2\} \times \{o_1, o_2\}$ is not a closed set either because it can be extended with s_1.*

The following theorem, its proof is in the Appendix A, states that the non-redundant rules on $\mathcal{D}'$ are exactly those derivable from the closed sets in $\mathcal{R}_A$ (their elements in $\cup_{D^i \in \mathcal{D}'} D^i$ being split between bodies and heads) and satisfying a second condition pertaining to the confidences of the more general rules sharing the same elements.

Theorem 7 (Closed sets and non-redundant rules)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, let $X \to Y$ be a canonical rule on $\mathcal{D}'$. $X \to Y$ is a non-redundant rule iff $(X \sqcup Y \sqcup s(X \sqcup Y))$ is a closed set in $\mathcal{R}_A$ and $\forall X' \sqsubset X$, $c_{exclusive}(X' \to (Y \sqcup X) \setminus X') < c_{exclusive}(X \to Y)$ or $c_{natural}(X' \to (Y \sqcup X) \setminus X') < c_{natural}(X \to Y)$.*

In this section, we have presented what are multidimensional association rules in n-ary relations, and have proposed measures to evaluate the significances of rules.

We have also discussed the redundancy of rules. Indeed, in many contexts, rules of interest only involve some of the attribute domains $\mathcal{D}' \subseteq \mathcal{D}$. For example, in $\mathcal{R}_E$, the analyst want to focus on rules including products and seasons, i. e.., $\mathcal{D}' = \{D^1, D^2\}$. For this reason, in the next section, we will propose an algorithm which finds non-redundant and interesting rules on the user-defined domains of interest ($\mathcal{D}' \subsetneq \mathcal{D}$).

3.3 Discovering multidimensional association rules

Given an n-ary relation $\mathcal{R} \subseteq \times_{D^i \in \mathcal{D}} D^i$ and the user-defined domains of interest $\mathcal{D}' \subsetneq \mathcal{D}$, the objective of the PINARD++ algorithm is to enumerate very interesting and non-redundant association rule on $\mathcal{D}'$. Such rules have the frequency beyond $\mu \in [0; 1]$, the exclusive confidence beyond $\beta_{\text{exclusive}} \in [0; 1]$, and the natural confidence beyond $\beta_{\text{natural}} \in [0; 1]$. In other terms, our algorithm PINARD++ computes:

$$\{X \to Y \text{ on } \mathcal{D}' \mid \begin{cases} X \to Y \text{ is non-redundant} \\ f(X \to Y) \geq \mu \\ c_{\text{exclusive}}(X \to Y) \geq \beta_{\text{exclusive}} \\ c_{\text{natural}}(X \to Y) \geq \beta_{\text{natural}} \end{cases} \} \ .$$

PINARD++ proceeds in three successive steps: (1) It constructs the relation $\mathcal{R}_A$ defined at the end of the previous section; (2) It extracts the *frequent* closed sets in $\mathcal{R}_A$; (3) It derives from these patterns the non-redundant rules whose exclusive and natural confidences exceed the user-defined thresholds. The first step is trivial. The second step relies on the state-of-the art algorithm DATA-PEELER for extracting closed sets from which frequent enough rules are obtained. The derivation of the non-redundant rules from the closed sets is presented in Sec. 3.3.2.

3.3.1 Computing closed n-sets

Theorem 7 states the link between the non-redundant association rules and the closed sets in $\mathcal{R}_A$ but, to be *a priori* interesting, the rules must satisfy constraints. Some approaches have been proposed to exhaustively list the closed sets in *ternary* relations, for example, CUBEMINER [61] and TRIAS [56]. An other algorithm, DATA-PEELER [33], can compute every closed set in arbitrary n-ary relations ($n \geq 2$). Despite its broader scope, it is orders of magnitude faster than both TRIAS and CUBEMINER on ternary relations. Furthermore, DATA-PEELER can efficiently handle an expressive class of constraints. This is particularly appealing in our context (in Chapter 5). To guarantee that all rules exceed the user-defined frequency threshold, in $\mathcal{R}_A$, we only discover the frequent closed sets which gather at least a proportion μ of the elements in D^{supp}. It means that every extracted closed set C must satisfy the constraint $\mathcal{C}_{freq}(C) \equiv \frac{|\pi_{D^{\text{supp}}}(C)|}{|D^{\text{supp}}|} \geq \mu$. DATA-PEELER can handle it directly on the closed sets.

From a closed set C, PINARD++ derives interesting and non-redundant multidimensional association rules on $\mathcal{D}'$ that involve all the elements in $\cup_{D^i \in \mathcal{D}'} \pi_{D^i}(C)$.

3.3.2 Deriving non-redundant rules

Data: (B, H), i. e., a body and a head
forall $e \succ \max_{\prec}(H)$ **do**
  if $c_{natural}(B \setminus \{e\} \rightarrow H \sqcup \{e\}) \geq \beta_{natural}$ **then**
    $c_{\mathrm{e}} \leftarrow c_{\mathrm{exclusive}}(B \setminus \{e\} \rightarrow H \sqcup \{e\})$
    if $c_e \geq \beta_{exclusive} \wedge \neg$REDUNDANT$(B \setminus \{e\}, H \sqcup \{e\}, \epsilon, c_e)$ **then** `/* ` ϵ `is smaller (w.r.t. ` $\prec$`) than any element */`
      output $B \setminus \{e\} \rightarrow H \sqcup \{e\}$
    RULES$(B \setminus \{e\}, H \sqcup \{e\})$

Algorithm 5: RULES.

Data: $(B', H', e', c_{\mathrm{e}})$, i. e., a body, a head, the last enumerated element and the exclusive confidence of the tested rule
forall $f' \in \{f' \in \cup_{D^i \in \mathcal{D}'} \pi_{D^i}(B') \mid f' \succ e'\}$ **do**
  if
  $c_{natural}(B' \setminus \{f'\} \rightarrow H' \sqcup \{f'\}) = c_{natural}(B' \rightarrow H') \wedge \Big(c_{exclusive}(B' \setminus \{f'\} \rightarrow H' \sqcup \{f'\}) \geq c_e \vee$ REDUNDANT$(B' \setminus \{f'\}, H' \sqcup \{f'\}, f', c_e)\Big)$ **then**
    return true
return false

Algorithm 6: REDUNDANT.

RULES (Algorithm 5) derives *a priori* interesting and non-redundant rules, of the form $B \rightarrow H$, from every frequent closed association A $(= C \setminus \pi_{D^{\mathrm{supp}}}(C))$. It splits *all* elements in $\cup_{D^i \in \mathcal{D}'} \pi_{D^i}(A)$ between the body B and the head H, i. e., $B \sqcup H = A$. The candidate rules are structured in a tree. By only looking at the heads, H, of the rules (A and H being given, the body B is $A \setminus H$), this tree actually is that of APRIORI [3]. Nevertheless, RULES traverses the tree by a depth-first search. The root of the tree is $A \rightarrow \emptyset$. At every level, H grows by one element which is removed from B. An arbitrary total order $\prec$ is chosen for the elements in $\cup_{D^i \in \mathcal{D}'} \pi_{D^i}(A)$. At every node, the singletons that are allowed to augment (via $\sqcup$) the head are greater than any element in the current head (i. e., greater than $\max_{\prec}(H)$ and under the convention that $\max_{\prec}(\emptyset)$ is smaller than any other element). The pruning criterion is the minimal natural confidence constraint. According to Theorem 6, this pruning is safe, i. e., no rule, with a natural confidence higher than β_{natural}, is missed. On the opposite, the minimal exclusive confidence and the non-redundancy constraints cannot give rise to search space pruning. That

is why they are checked after the constraint on the minimal natural confidence. If both are satisfied then the rule is output. Checking whether the exclusive confidence exceeds $\beta_{exclusive}$ is straightforward. To enforce the non-redundancy, Theorem 7 indicates that, beside the necessity of processing a closed set, RULES must check the confidences of the more general rules sharing the same elements. If such a rule has the same natural confidence and a greater or equal exclusive confidence, then the current rule is redundant. That is why the REDUNDANT function (Algorithm 6) browses these more general rules and compare their confidences with that of the current rule. Like RULES, REDUNDANT exploits Theorem 6 such that it does not traverse rules with strictly smaller natural confidence. Finally, PINARD++ is described in Algorithm 7).

Input: A relation $\mathcal{R}$ on $\mathcal{D}$, $\mathcal{D}' \subsetneq \mathcal{D}$, and $(\mu, \beta_{natural}, \beta_{exclusive}) \in [0,1]^3$
Output: Every non-redundant and *a priori* interesting rule on $\mathcal{D}'$
$D^{\text{supp}} \leftarrow \times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$
$(\mathcal{D}_A, \mathcal{R}_A) \leftarrow (\mathcal{D}' \cup D^{\text{supp}}, \emptyset)$
forall $(e_1, e_2, \ldots, e_{|\mathcal{D}'|}, e_{|\mathcal{D}'|+1}, \ldots, e_n) \in \mathcal{R}$ **do**
  $\mathcal{R}_A \leftarrow \mathcal{R}_A \cup (e_1, e_2, \ldots, e_{|\mathcal{D}'|}, (e_{|\mathcal{D}'|+1}, \ldots, e_n))$
$\mathcal{C} \leftarrow$ DATA-PEELER$(\emptyset, \times_{D^i \in \mathcal{D}_A} D^i)$
forall $C \in \mathcal{C}$ **do**
  RULES$(C \setminus \pi_{D^{supp}}(C), \emptyset)$

Algorithm 7: PINARD++.

3.4 Empirical study

We empirically study multidimensional association rule mining and the efficiency of PINARD++. We begin by describing the real data used for our experiments. Then, we analyze the results w.r.t. the following questions: (a) Do the discover relevant rules? (b) What do the different confidence definitions capture?, and (c) How does the algorithm PINARD++ behave w.r.t. parameter settings?

All our experiments have been performed on a GNU/Linux™ system equipped with two Intel(R) Core(TM)2 Duo CPU E7300 at 2.66 GHz and 2.9 GB of RAM. The prototypes are implemented in C++ and compiled with GCC 4.2.4.

3.4.1 Dataset: DistroWatch

`DistroWatch`[2] is a website gathering information about GNU/Linux, BSD and Solaris operating systems. Every distribution is described on a separate page. When a visitor loads a page, his/her country is known from his/her IP address. The logs of

2. `http://www.distrowatch.com`

the Web server are easily converted into a three dimensional tensor that gives for any time period (13 semesters from early 2004 to early 2010) the number of visits from any country on any distribution (655 distributions). We decided to keep countries that are associated with at least 2,000 consultations of a distribution at a semester (i.e., the 96 "most active" countries). These numerical data are normalized so that every couple $(semester, country)$ has the same weight. Then, a procedure, inspired by the computation of a p value, *locally* chooses the relevant 3-tuples: for every distribution (hence, "locally"), the 3-tuples associated with the greatest normalized values are kept until their sum reaches 20% of the sum of all normalized values involving the distribution. If a 3-tuple (d, s, c) belongs to the relation, it means that a significant amount of users from country c have been visiting the description of software distribution d during semester s. The resulting ternary relation, namely $\mathcal{R}_{\texttt{DistroWatch}}$, contains 21,033 3-tuples, hence a $\frac{21{,}033}{13\times655\times96} = 2.6\%$ density. This relation is used to discover associations between distributions and countries.

3.4.2 Experimental results

Let us first discuss a qualitative study where we look for rules in $\mathcal{R}_{\texttt{DistroWatch}}$ that involve countries and distributions. These two dimensions form the set $\mathcal{D}'$ and we used the thresholds $\mu = 0.75$, $\beta_{\text{exclusive}} = 0.6$, and $\beta_{\text{natural}} = 0.8$. PINARD++ computes 39 canonical and non-redundant rules. Here as some of them:

- $\{\text{Taiwan}\} \times \{\text{fedora}\} \rightarrow \{\text{b2d}\}$
  $(f : 0.846,\ c_{\text{natural}} : 0.917,\ c_{\text{exclusive}} : 0.917)$;
- $\{\text{Japan}\} \times \{\text{centOS}\} \rightarrow \{\text{Ecuador}\}$
  $(f : 0.769,\ c_{\text{natural}} : 0.909,\ c_{\text{exclusive}} : 0.909)$;
- $\{\text{berry,plamo}\} \rightarrow \{\text{Japan}\}$
  $(f : 0.923,\ c_{\text{natural}} : 1,\ c_{\text{exclusive}} : 0.75)$;
- $\{\text{berry,momonga,plamo}\} \rightarrow \{\text{Japan}\}$
  $(f : 0.769,\ c_{\text{natural}} : 1,\ c_{\text{exclusive}} : 1)$;
- $\{\text{caixamagica}\} \rightarrow \{\text{Portugal}\}$
  $(f : 0.846,\ c_{\text{natural}} : 1,\ c_{\text{exclusive}} : 1)$.

The first rule listed above indicates that if the Taiwaneses show interest in *fedora* then they also show interest in *b2d* at the same semester, its confidence is larger than 0.9 ($c_{\text{natural}} = c_{\text{exclusive}} = 0.917$). The probability that Ecuadorian people consult *centOS* during the semesters Japanese do so, is greater than 90% (the second rule having 0.909 for confidences). Japan is the origin country of the distributions *berry*, *plamo*, *momonga*, i. e., these distributions are developed by Japanese. That is why the visits on these related Web pages almost exclusively come from this country. The natural confidence of the third rule is 1, it means that Japanese visits *berry* et *plamo* at all semester when these distributions are visited together. This rule also indicates that people from other countries rarely consult these distributions at the same semester ($1 - c_{\text{exclusive}} = 0.25$). Since the fourth rule involves the three

distributions *berry*, *plamo*, *momonga* at the same semester, the exclusive confidence is higher. It is 1, i. e., outside Japan, no other country frequently loads the three related Web pages of these distributions at the same semester. The same interpretation holds for the last rule. The distribution *caixamagica* is developed by and for people in Portugal. It is visited exclusively by them ($c_{\text{natural}} = c_{\text{exclusive}} = 1$).

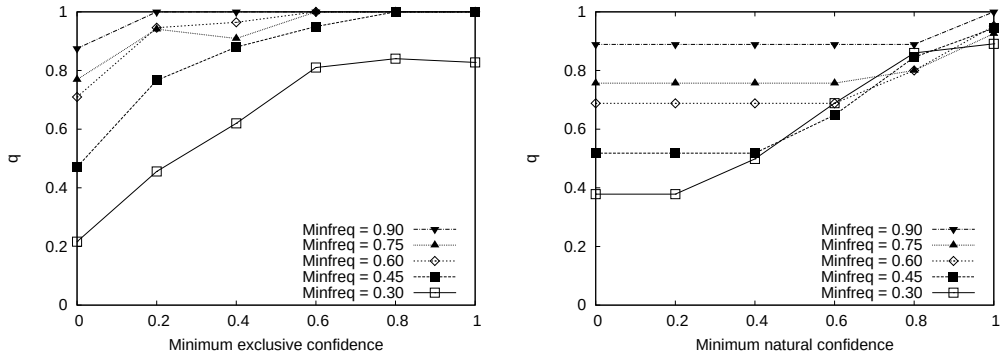
In fact, in most of the discovered rules of the form *distributions* $\rightarrow$ *countries*, we observe that their heads often involve only the countries where at least one of the distributions has been developed. This proportion is

$$q = \frac{|\{X \rightarrow Y \mid \begin{cases} X \subseteq D^{\text{distributions}} \wedge Y \subseteq D^{\text{countries}} \\ \forall y \in Y, \exists x \in X \mid \text{origin}(x) = y \end{cases} \}|}{|\{X \rightarrow Y \mid X \subseteq D^{\text{distributions}} \wedge Y \subseteq D^{\text{countries}}\}|}$$

where *origin*(x) is the origin country of the distribution x. Indeed, distributions that are specifically developed by and for a country mainly attract users from this country. Therefore, we expect that higher minimal thresholds on the designed measures (frequency and confidences) actually capture higher q value. Figure 13 plots q in function of these thresholds. We see that q actually increases w.r.t. every minimal threshold. This empirically corroborates the relevance of our semantic measures that higher values of the measures actually capture more relevant patterns. The measure q increases more quickly with $\beta_{\text{exclusive}}$ than with β_{natural}. This makes sense: a conjunction of distributions that *exclusively* interests visitors from a given country usually means that at least one of these distributions is developed by people in this country and for this country (with, often, language specifics taken into account). Finally, it is interesting to understand that, under a given minimal frequency constraint μ, the collections of rules computed with $\beta_{\text{natural}} \leq \mu$ ($\beta_{\text{exclusive}}$ remaining constant) are the same, this explains the horizontal segments in Figure 13b. Indeed, the natural confidence is a proportion of elements in the support domain of the rule and the frequency constraint forces the rule to match at least a proportion μ of elements in this domain. As a consequence, no rule can have a natural confidence beneath μ.

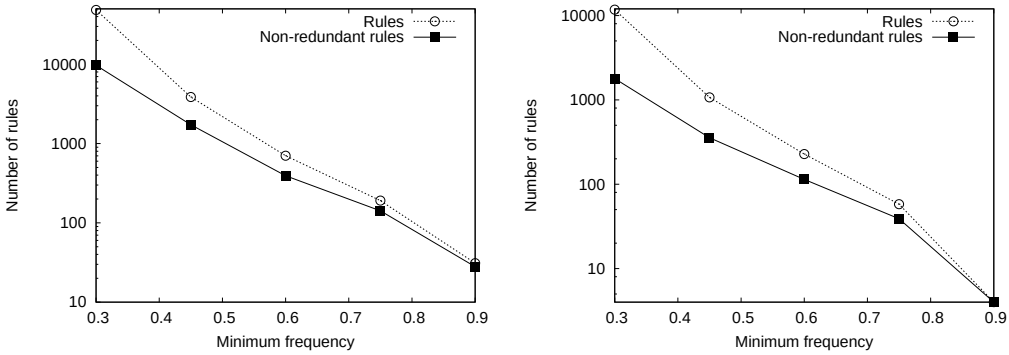
When mining rules that only satisfy the minimum frequency and minimum confidence constraints, many redundant rules are returned although they do not provide new insights. Figure 14 illustrates the proportion of rules that are avoided thanks to our non-redundancy approach (see Sec. 3.2.4). Obviously, with low minimum frequency constraints, this significantly limits pattern flooding.

We now report a performance study in $\mathcal{R}_{\texttt{DistroWatch}}$ with rules involving countries and distributions (i. e. $\mathcal{D}' = \{\text{Countries, Distributions}\}$). Indeed, PINARD++ prunes large areas of the search space where every closed set violates the minimal frequency constraint. As a consequence, when the minimal frequency threshold increases, the number of frequent closed sets decreases, so the number of frequent rules decreases



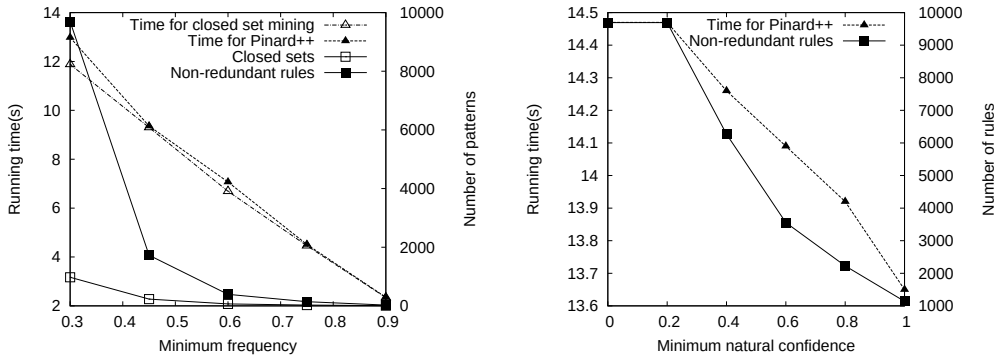
(a) q with varying $\beta_{\text{exclusive}}$ ($\beta_{\text{natural}} = 0$). (b) q with varying β_{natural} ($\beta_{\text{exclusive}} = 0$).

Figure 13: Confidence qualitative assessment



(a) $\beta_{\text{natural}} = \beta_{\text{exclusive}} = 0$. (b) $\beta_{\text{natural}} = 0.8$, $\beta_{\text{exclusive}} = 0.6$.

Figure 14: Impact of non-redudancy.



(a) Pruning w.r.t. μ.

(b) Pruning w.r.t. β_{natural}.

Figure 15: Effectiveness of PINARD++

and the running time decreases (Figure 15a obtained with $\beta_{\text{natural}} = \beta_{\text{exclusive}} = 0$). By exploiting Theorem 6, the RULES algorithm does not traverse the enumeration sub-trees which do not include any confident rule (w.r.t. natural confidence). Thereby, the number of rules and the running time of the extraction decrease when the minimum natural confidence threshold increases (Figure 15b). This experiment was performed with $\beta_{exclusive} = 0$ and $\mu = 0.3$.

PINARD++'s scalability was tested w.r.t. the size and the density of the data. Starting with the size, rules on $\mathcal{D}' = \{\text{Countries, Distributions}\}$ are mined with $\mu = 0.75$ and $\beta_{\text{natural}} = \beta_{\text{exclusive}} = 0$. $\mathcal{R}_{\texttt{DistroWatch}}$ was replicated up to 10 times with the timestamps. It turns out that the algorithm scales linearly. More precisely, a linear regression of $R \mapsto \frac{T_R}{T_1}$ (where R is the replication factor, T_R is the running time on this replicated dataset) gives $y = 2.57x - 2.66$ with 0.97 as a determination coefficient.

To test the PINARD++'s scalability w.r.t. the density of the dataset, synthetic 3-ary relations have been generated. The sizes of the domains are constant: $10 \times 50 \times 100$. Here, the only variable is the density, i. e., the ratio between the number of 3-tuples present in the relation and $10 \times 50 \times 100 = 50,000$. In our test, it increases, 0.02 by 0.02, from 0.1 (for the first dataset) to 0.5 (for the last dataset). The experiment was performend with $\mu = 0.1$, $\beta_{exclusive} = 0.4$ and $\beta_{natrual} = 0.7$. The PINARD++'s running times are in Figure 16. As predicted, when the density is high, the extraction is much harder. However, let us note that 40% density is already extremely high to be met in practice.

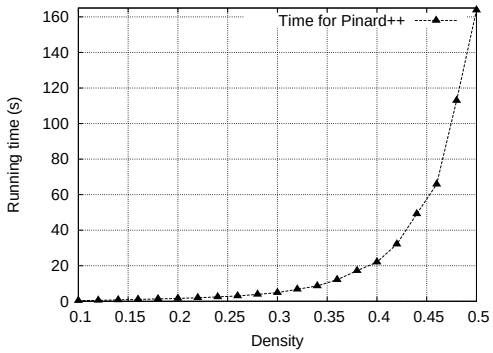


Figure 16: PINARD++'s scalability w.r.t. the density.

3.5 Conclusion

We have presented a generalization of association rules in n-ary relation. A multidimensional association rule is an implication between two associations where each association can contain subsets of some arbitrary domains. Three proposed objective measures for evaluating the interestingness of such rules are frequency, exclusive confidence and natural confidence. We also considered the redundancy of multidimensional association rules. A rule is redundant if its information is included in that of another general rule. We propose the concept of non-redundant multidimensional rules having minimal body and maximal head. Theorem 7 shows that a rule is non-redundant if it is derived from a closed set. To compute non-redundant interesting rules, we proposed the PINARD++ algorithm which is a post-processing of the patterns extracted with the state-of-the art algorithm DATA-PEELER. The performance of PINARD++ was tested on real and synthetic datasets. The experiments show that the performance of PINARD scales linearly with the dataset size, but it does not scale linearly with the dataset density. We also presented a multidimensional association rule mining for the analysis of the `DistroWatch` data. The output rules may be used to understand communities (generally related to countries and spoken languages) that prefer to look at some groups of distributions. The relevancy of the patterns has been quantified thanks to some simple domain knowledge.

Multidimensional association rules enable to describe and evaluate co-occurrence of associations. However, another natural goal could be to look for associations that can be the consequent of a frequent association even though all of them do not co-occur. We address this question in the next chapter.

Chapter 4

Generalizing disjunctive rules in n-ary relations

Given an n-ary relation, a multidimensional association rule (see Chapter 3) conveys information about the co-occurrence between elements of some domains. The designed measures aim to evaluate the probability of the co-occurrence between the head and the body of a rule and to evaluate whether the association in the rule body "*prefers*" to co-occur with the association in the head than with other associations. We now look for distinct relationships among associations. Our goal is to answer the following question: "*Which cases can occur when we observe a frequent association?*"

Our contribution is twofold. First, we design the pattern domain of multidimensional disjunctive rules. Such a rule is an implication between associations: its body is an association and its head is a disjunction of associations. We provide two types of objective interestingness measures: association measures and disjunctive measures. We also look further at the concept of non-redundant rule. Second we propose the CIDRE [1] algorithm which lists the complete collection of interesting multidimensional disjunctive rules.

The next section introduces the motivation and objective interestingness for multidimensional disjunctive rules. Section 4.2 defines the pattern domain of multidimensional disjunctive rules. Section 4.3 presents our algorithm which extracts interesting multidimensional disjunctive rules within a n-ary relation. Experiments on real-life data are reported in Section 4.4. Section 4.5 summarizes the chapter.

4.1 Motivation and objective interestingness

An association can frequently co-occurs with some other associations, but all these associations do not necessarily co-occur together. For example, observe the

1. CIDRE Is a Disjunctive Rule Extractor.

relation $\mathcal{R}_E$ (see Figure 10), the products p_1, p_2 and p_4 are frequently bought in season s_2. However, customers rarely buy all these products together in the same transaction. Thus, the multidimensional association rule mining cannot provide a rule like $\{s_2\} \rightarrow (\{p_1, p_2\} \times \{s_1\}) \vee (\{p_4\} \times \{s_4\})$. Such a rule means that when a customer goes shopping in the season s_2, he/she tends to buy p_1, p_2 or p_4. If he/she prefers the products p_1 and p_2 then he/she also buys them in the season s_1. If he/she prefers p_4 then he/she tends to also buy it in the season s_4. Indeed, such rules are more informative than conjunctive rules.

In addition, it is ineffective to find multidimensional association rules on datasets having a very few frequent associations and a very large number of infrequent associations because those rules are based on the co-occurrence relation of associations with a large enough frequency and confidences.

We address the above problems via the study of multidimensional disjunctive rule mining in n-ary relations. Our goals are to answer the question "*Which cases can occur when we observe a frequent association?*" and to mine rules in which an association with large frequency implies associations with small frequency.

4.2 Multidimensional disjunctive rules

4.2.1 Definitions

Given a relation $\mathcal{R} \subseteq D^1 \times \cdots \times D^n$ and the user-defined domains of interest $\mathcal{D}' \subseteq \mathcal{D} = \{D^1 \times \cdots \times D^n\}$, a *multidimensional disjunctive rule* on $\mathcal{D}'$ is of the form $X \rightarrow \vee\mathcal{Y}$ such that the union of its body and each association in the disjunctions of its head is an association on $\mathcal{D}'$. It is simply called a rule when it is clear from the context. Without loss of generality, the dimensions are assumed ordered such that $\mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\}$.

Definition 48 (Multidimensional disjunctive rule)**.** *$\forall\mathcal{D}' \subseteq \mathcal{D}$, $X \rightarrow \vee\mathcal{Y}$ is a multidimensional disjunctive rule on $\mathcal{D}'$ iff X is an association on a subset of $\mathcal{D}'$ and $\mathcal{Y}$ is a set of associations on subsets of $\mathcal{D}'$ such that $\forall Y \in \mathcal{Y}$, $X \sqcup Y$ is an association on $\mathcal{D}'$.*

The *support domain* of a multidimensional disjunctive rule on $\mathcal{D}'$ is the *Cartesian product* of all domains that are not in $\mathcal{D}'$, i. e.., $\times_{D \in \mathcal{D} \setminus \mathcal{D}'} D$.

Example 48. *In $\mathcal{R}_E$, $\{s_2\} \rightarrow (\{p_1, p_2\} \times \{s_1\}) \vee (\{p_4\} \times \{s_4\})$, $\{p_3, p_4\} \rightarrow (\{p_2\} \times \{s_1, s_3\}) \vee \{s_4\}$ and $\{p_4\} \times \{s_1\} \rightarrow (\{p_1, p_2\}) \vee (\{p_2\}) \vee (\{p_2, p_3\} \times \{s_3\})$ are three multidimensional disjunctive rules on $\{D^1, D^2\}$. Their support domain is D^3.*

In the binary case (i. e., $n = 2$), the semantics of association rules, even when generalized to disjunctive or negative association rules [77, 6], is based on the frequency

and the confidence measures. In the context of n-ary relations, we introduced generalizations of these measures for multidimensional association rules (see Section 3.2). We now adapt such measures to our disjunctive rule mining setting.

Given a multidimensional disjunctive rule, we want first to evaluate the strength of the co-occurrences between its body and each association in its head. We want also to measure how often we observe a occurrence of at least one association in its head when its body holds. Therefore, two types of interestingness measures are proposed, namely the association measures and the disjunctive measures.

We use some of the definitions presented in Chapter 3: it concerns the support of an association $s(X)$ (Definition 34) and the natural support of an association $s_{\mathcal{D}\setminus\mathcal{D}'}(X)$ (Definition 42).

4.2.2 Association measures

The objective of the association measures of a rule is to evaluate the probability of the conjunction between the body and each association in the head. The association measures are association frequency and association confidence. The association frequency is defined as the ratio of elements (in the support domain, $\times_{D^i\in\mathcal{D}\setminus\mathcal{D}'} D^i$) that support both the body and the considered association to the total number of elements in the support domain. The association confidence is defined as the ration of the number of elements (in the support domain) supporting both the body and the considered association to the total number of elements (in the support domain) supporting the body.

Definition 49 (Association frequency)**.** *$\forall\mathcal{D}' \subseteq \mathcal{D}$, let $X \to \vee\mathcal{Y}$ be a multidimensional disjunctive rule on $\mathcal{D}'$. $\forall Y \in \mathcal{Y}$, the association frequency of $X \to Y$ is*

$$f_a(X \to Y) = \frac{|s(X \sqcup Y)|}{|\times_{D^i\in\mathcal{D}\setminus\mathcal{D}'} D^i|}.$$

Example 49. *Considering $r_1 : \{s_2\} \to (\{p_1, p_2\} \times \{s_1\}) \vee (\{p_4\} \times \{s_4\})$ in $\mathcal{R}_E$, we have:*

- $f_a(\{s_2\} \to \{p_1, p_2\} \times \{s_1\}) = \frac{|s(\{s_2\}\sqcup\{p_1,p_2\}\times\{s_1\})|}{|D^3|} = \frac{|\{o_1,o_2\}|}{|\{o_1,o_2,o_3,o_4,o_5\}|} = \frac{2}{5}$,
- $f_a(\{s_2\} \to \{p_4\} \times \{s_4\}) = \frac{|s(\{s_2\}\sqcup\{p_4\}\times\{s_4\})|}{|D^3|} = \frac{|\{o_1,o_4\}|}{\{o_1,o_2,o_3,o_4,o_5\}|} = \frac{2}{5}$.

Definition 50 (Association confidence)**.** *$\forall\mathcal{D}' \subseteq \mathcal{D}$, let $X \to \vee\mathcal{Y}$ be a multidimensional disjunctive rule on $\mathcal{D}'$. $\forall Y \in \mathcal{Y}$, the association confidence of $X \to Y$ is*

$$c_a(X \to Y) = \frac{|s(X \sqcup Y)|}{|s_{\mathcal{D}\setminus\mathcal{D}'}(X)|}.$$

Example 50. *Considering again r_1 in $\mathcal{R}_E$, we have:*

– $c_a(\{s_2\} \rightarrow \{p_1, p_2\} \times \{s_1\}) = \frac{|s(\{s_2\} \sqcup \{p_1,p_2\} \times \{s_1\})|}{|s_{D^3}(\{s_2\})|} = \frac{|\{o_1,o_2\}|}{|\{o_1,o_2,o_4,o_5\}|} = \frac{2}{4}$,

– $c_a(\{s_2\} \rightarrow \{p_4\} \times \{s_4\}) = \frac{|s(\{s_2\} \sqcup \{p_4\} \times \{s_4\})|}{|s_{D^3}(\{s_2\})|} = \frac{|\{o_1,o_4\}|}{|\{o_1,o_2,o_4,o_5\}|} = \frac{2}{4}$.

It means that, in the rule r_1, the confidence of the conjunction between the body and any association in the head is 0.5.

4.2.3 Disjunctive measures

The objective of the disjunctive measures of a rule is to evaluate the probability to observe at least one association in the head when the body holds. The disjunctive measures are disjunctive frequency and disjunctive confidence. The disjunctive frequency is defined as the ratio of elements (in the support domain, $\times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$) which simultaneously support the body and at least one association in the head to the total number of elements in the support domain. The disjunctive confidence is defined as the ratio of the number of elements (in the support domain) which simultaneously support the body and at least one association in the head to the total number of elements (in the support domain) which support the body.

Definition 51 (Disjunctive frequency)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, let $X \rightarrow \vee\mathcal{Y}$ be a multidimensional disjunctive rule on $\mathcal{D}'$. The disjunctive frequency of $X \rightarrow \vee\mathcal{Y}$ is*

$$f_d(X \rightarrow \vee\mathcal{Y}) = \frac{|\cup_{Y \in \mathcal{Y}} s(X \sqcup Y)|}{|\times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i|}.$$

Example 51. *Consider $r_1 : \{s_2\} \rightarrow (\{p_1, p_2\} \times \{s_1\}) \vee (\{p_4\} \times \{s_4\})$ and $r_2 : \{p_3, p_4\} \rightarrow (\{p_2\} \times \{s_1, s_3\}) \vee \{s_4\}$ in $\mathcal{R}_E$, we have:*

– $f_d(r_1) = \frac{|s(\{s_2\} \sqcup \{p_1,p_2\} \times \{s_1\}) \cup s(\{s_2\} \sqcup \{p_4\} \times \{s_4\})|}{|D^3|} = \frac{|\{o_1,o_2,o_4\}|}{|\{o_1,o_2,o_3,o_4,o_5\}|} = \frac{3}{5}$,

– $f_d(r_2) = \frac{|s(\{p_3,p_4\} \sqcup \{p_2\} \times \{s_1,s_3\}) \cup s(\{p_3,p_4\} \sqcup \{s_4\})|}{|D^3|} = \frac{|\{o_1,o_3,o_4,o_5\}|}{|\{o_1,o_2,o_3,o_4,o_5\}|} = \frac{4}{5}$.

Definition 52 (Disjunctive confidence)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, let $X \rightarrow \vee\mathcal{Y}$ be a multidimensional disjunctive rule on $\mathcal{D}'$. The disjunctive confidence of $X \rightarrow \vee\mathcal{Y}$ is*

$$c_d(X \rightarrow \vee\mathcal{Y}) = \frac{|\cup_{Y \in \mathcal{Y}} s(X \sqcup Y)|}{|s_{\mathcal{D} \setminus \mathcal{D}'}(X)|}.$$

Example 52. *Consider again r_1 and r_2 in $\mathcal{R}_E$, we have:*

– $c_d(r_1) = \frac{|s(\{s_2\} \sqcup \{p_1,p_2\} \times \{s_1\}) \cup s(\{s_2\} \sqcup \{p_4\} \times \{s_4\})|}{|s_{D^3}(\{s_2\})|} = \frac{|\{o_1,o_2,o_4\}|}{|\{o_1,o_2,o_4,o_5\}|} = \frac{3}{4}$,

– $c_d(r_2) = \frac{|s(\{p_3,p_4\} \sqcup \{p_2\} \times \{s_1,s_3\}) \cup s(\{p_3,p_4\} \sqcup \{s_4\})|}{|s_{D^3}(\{p_3,p_4\})|} = \frac{|\{o_1,o_3,o_4,o_5\}|}{|\{o_1,o_3,o_4,o_5\}|} = \frac{4}{4}$.

Rule r_1 indicates that when a customer goes shopping in the season s_2, he tends to buy p_1, p_2 or p_4 ($f_d = 0.6$, $c_d = 0.75$). If he/she prefers the products p_1 and p_2 then he/she also buys them in the season s_1 ($f_a = 0.4$, $c_a = 0.5$). If he/she prefers

p_4 then he/she tends to also buy it in the season s_4 ($f_a = 0.4$, $c_a = 0.5$). Rule r_2 means that if a customer likes the products p_3 and p_4 then he/she tends to buy them in the seasons s_1, s_3 or s_4 ($f_d = 0.8$, $c_d = 1$). The confidence that the products p_3 and p_4 are bought together in the season s_4 is 0.75 ($c_a = 0.75$). The confidence that these products are taken in the seasons s_1 and s_3 is 0.5 ($c_a = 0.5$), in this case, the customer may also buy p_2. Enabling disjunctions within the heads of the rules provides rules that convey more information than conjunctive rules.

4.2.4 Non-redundancy

Given a minimal association frequency, a minimal association confidence and a frequent association (rule body), there may be a large number of associations that co-occurs with it. Suppose the number of associations that co-occur with the given frequent association is k, the number of disjunctions which can be generated from the subsets of these associations is 2^k. So, given a frequent association, there are a huge number of generated multidimensional disjunctive rules that satisfy the minimality constraints on the interestingness measures. It is computationally expensive to find all such rules and, here again, we have to face with redundant rules. We consider the concept of non-redundant multidimensional disjunctive rule: a rule is non redundant if its information content is not included in another more general rule. It means a non-redundant rule has a minimal body and maximal head.

Example 53. *In $\mathcal{R}_E$, let us consider the following rules:*

- *r_3: $\{p_4\} \times \{s_1\} \rightarrow (\{p_1, p_2\}) \vee (\{p_1\}) \vee (\{p_2, p_3\} \times \{s_3\})$ ($f_d : 0.6, c_d : 1$),*
- *r_4: $\{p_2, p_4\} \times \{s_1\} \rightarrow \{p_1\}$ ($f_d : 0.4, c_d : 0.67$),*
- *r_5: $\{p_4\} \times \{s_1\} \rightarrow (\{p_1, p_2\}) \vee (\{p_1\}) \vee (\{p_2, p_3\} \times \{s_3\}) \vee (\{p_3\})$ ($f_d : 0.6, c_d : 1$).*

They have their association frequencies, their association confidences, their disjunctive frequencies, their disjunctive confidences respectively exceeding 0.4, 0.5, 0.4 and 0.65 that are the user-defined thresholds. Therefore, they may "individually" satisfy this aspect of interestingness. Nevertheless, "all together", they provide redundant information. For instance, the premise of r_4 is more informative than that of r_3 (to match the body of r_4, a customer must additionally take p_2), but the conclusion of r_4 is less informative (it does not tell anything about p_3 and s_3). In addition, this does not provide r_4 a greater frequency or a greater confidence than r_3. Rule r_4 is therefore said redundant. The conclusion of r_5 has more elements than that in the conclusion of r_3. However, in r_5, $\{p_3\} \sqsubset \{p_2, p_3\} \times \{s_3\}$, $f_a(\{p_4\} \times \{s_1\} \rightarrow \{p_3\}) = f_a(\{p_4\} \times \{s_1\} \rightarrow \{p_2, p_3\} \times \{s_3\}) = 0.4$ and $c_a(\{p_4\} \times \{s_1\} \rightarrow \{p_3\}) = c_a(\{p_4\} \times \{s_1\} \rightarrow \{p_2, p_3\} \times \{s_3\}) = 0.67$. Therefore, the appearance of $\{p_3\}$ in the conclusion of r_5 does not provide new insight. $\{p_3\}$ is thus redundant in r_5. In r_3, although $\{p_2\} \sqsubseteq \{p_1, p_2\}$, $\{p_2\}$ is not redundant since $f_a(\{p_4\} \times \{s_1\} \rightarrow \{p_2\}) = 0.6 > f_a(\{p_4\} \times \{s_1\} \rightarrow \{p_1, p_2\}) = 0.4$ and

$f_a(\{p_4\}\times\{s_1\} \to \{p_2\}) = 1 > f_a(\{p_4\}\times\{s_1\} \to \{p_1, p_2\}) = 0.67$. *Since the end-user would not find any added-value in rules* r_4 *and* r_5*, these rules must not be returned.*

By the meaning of minimal body and maximal head, a rule is non-redundant if its body is a minimal association and its head includes the maximal number of associations which can conjoin with its body such that the union of the body and each association in the head is a closed set.

Definition 53 (Non-redundant multidimensional disjunctive rule)**.** $\forall \mathcal{D}' \subseteq \mathcal{D}$*, a multidimensional disjunctive rule* $X \to \vee\mathcal{Y}$ *on* $\mathcal{D}'$ *is non-redundant iff it satisfies the following constraints:*
(1) $\forall Y \in \mathcal{Y}$*,* $X \to Y$ *is a key association rule on* $\mathcal{D}'$*. It means that, it is canonical and there is no other canonical association rule* $X' \to Y'$*, where* $X' \sqcup Y'$ *is an association on* $\mathcal{D}'$ *such that*

$$\begin{cases} (X' \sqcup Y' = X \sqcup Y \wedge X' \sqsubset X) \vee (X' \sqcup Y' \sqsupset X \sqcup Y \wedge X' \sqsubseteq X) \\ f_a(X' \to Y') \geq f_a(X \to Y) \\ c_a(X' \to Y') \geq c_a(X \to Y) \end{cases}.$$

(2) There is no rule which is more general than $X \to \vee\mathcal{Y}$*. It means that it does not exist any multidimensional disjunctive rule* $X \to \vee\mathcal{Z}$*, where* $\mathcal{Y} \subset \mathcal{Z}$*, such that*

$$\begin{cases} X \to \vee\mathcal{Z} \text{ satisfies the constraint (1)} \\ f_d(X \to \vee\mathcal{Z}) \geq f_d(X \to \vee\mathcal{Y}) \\ c_d(X \to \vee\mathcal{Z}) \geq c_d(X \to \vee\mathcal{Y}) \end{cases}.$$

The first condition defines the form of the key association rule having minimal body and maximal head. So, if a disjunctive association satisfies the first condition then its body is always minimal. If it also satisfies the second condition then its head has the most associations. As a consequence, when both conditions are satisfied, the disjunctive association rule has a minimal body and a maximal head without the redundancy.

The two following theorems indicate that the non-redundant rules on $\mathcal{D}'$, as defined above, can be efficiently derived from the closed sets extracted from a relation $\mathcal{R}_A$. It is obtained from $\mathcal{R}$ by "flattening" the dimensions absent from $\mathcal{D}'$ into a unique support dimension $D^{\text{supp}} = \times_{D^i \in \mathcal{D}\setminus\mathcal{D}'} D^i$. It is defined on the domains $\mathcal{D}_A = \mathcal{D}' \cup \{D^{\text{supp}}\}$. $\mathcal{R}_A$ was introduced in Chapter 3: assuming that for all $i = 1..n$, e_i is an element of the i^{th} domain, i. e., $e_i \in D^i$, we build

$$\mathcal{R}_A = \{(e_1, e_2, \ldots, e_{|\mathcal{D}'|}, (e_{|\mathcal{D}'|+1}, \ldots, e_n)) | (e_1, e_2, \ldots, e_{|\mathcal{D}'|}, e_{|\mathcal{D}'|+1}, \ldots, e_n) \in \mathcal{R}\}.$$

Theorem 8 (Closed set and key association rule)**.** $\forall \mathcal{D}' \subseteq \mathcal{D}$*, let* $X \to Y$ *be a canonical association rule such that* $X \sqcup Y$ *is an association on* $\mathcal{D}'$*.* $X \to Y$ *is a key association rule on* $\mathcal{D}'$ *iff* $(X \sqcup Y \sqcup s(X \sqcup Y))$ *is a closed set in* $\mathcal{R}_A$ *and* $\forall X' \sqsubset X$*,* $c_a(X' \to (Y \sqcup X) \setminus X') < c_a(X \to Y)$.

Theorem 9 (Key association rule and non-redundant multidimensional disjunctive rule)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, let $\mathcal{P}$ be the set of all key association rules on $\mathcal{D}'$, a multidimensional disjunctive rule $X \rightarrow \vee\mathcal{Y}$ on $\mathcal{D}'$ is a non redundant iff $\forall Y \in \mathcal{Y}$, $X \rightarrow Y$ is a key association rule and $\mathcal{Y} = \cup_{X \rightarrow Y \in \mathcal{P}} Y$.*

4.3 Discovering multidimensional disjunctive rules

Given a n-ary relation $\mathcal{R} \subseteq \times_{D^i \in \mathcal{D}} D^i$, $\mathcal{D}' \subset \mathcal{D}$, we look for every *a priori* interesting and non-redundant multidimensional disjunctive rule. Such a rule satisfies user-defined constraints based on measure thresholds: the minimal association frequency μ_a, the minimal association confidence β_a, the minimal disjunctive frequency μ_d and the minimal disjunctive confidence β_d. In other terms, the algorithm CIDRE computes:

$$\{X \rightarrow \vee\mathcal{Y} \text{ on } \mathcal{D}' \mid \begin{cases} X \rightarrow \vee\mathcal{Y} \text{ is non-redundant} \\ \forall Y \in \mathcal{Y}, f_a(X \rightarrow Y) \geq \mu_a \\ \forall Y \in \mathcal{Y}, c_a(X \rightarrow Y) \geq \beta_a \\ f_d(X \rightarrow \vee\mathcal{Y}) \geq \mu_d \\ c_d(X \rightarrow \vee\mathcal{Y}) \geq \beta_d \end{cases} .$$

CIDRE is divided into four successive steps: (1) It constructs the relation $\mathcal{R}_A$ (2) It extracts the *frequent* closed sets in $\mathcal{R}_A$; (3) It derives the key association rules satisfying the minimal association measures from these closed sets; (4) It computes the non-redundant disjunctive rules whose disjunctive frequency and disjunctive confidence hold for the user-defined thresholds μ_d and β_d.

4.3.1 Computing closed n-sets

Theorem 8 indicates that the key association rules are efficiently derived from closed sets. However, to guarantee all key association rules on $\mathcal{D}'$ exceed the user-defined association frequency threshold, in $\mathcal{R}_A$, we only discover the frequent closed sets which gather at least a proportion μ_a of the elements in $D^{supp} = \times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$. It means that every extracted closed set C must satisfy the constraint $\mathcal{C}_{freq}(C) \equiv \frac{|\pi_{D^{\text{supp}}}(C)|}{|D^{\text{supp}}|} \geq \mu_a$. We use DATA-PEELER to extract such closed sets.

4.3.2 Deriving key association rules

From a closed set C, KEY_ASSOCIATION_RULES (Algorithm 8) derives key association rules on $\mathcal{D}'$ that involve all the elements in $\cup_{D^i \in \mathcal{D}'} \pi_{D^i}(C)$.

KEY_ASSOCIATION_RULES derives *a priori* interesting key association rules, of the form $B \rightarrow H$, from every frequent closed association A $(= C \setminus \pi_{D^{\text{supp}}}(C))$. Its

idea is similar the deriving association rules in Section 3.3.2. Particularly, to split *all* elements in $\cup_{D^i \in \mathcal{D}'} \pi_{D^i}(A)$ between the body B and the head H, i. e., $B \sqcup H = A$, it generates a tree of candidate rules and traverses this tree as the same way of the RULES algorithm in Section 3.3.2. Thanks to Theorem 6, KEY_ASSOCIATION_RULES prunes the enumeration sub-trees where every rule violates the minimal association confidence constraint. According to Theorem 8, KEY_ASSOCIATION_RULES must check the association confidences of the more general association rules sharing the same elements. If such a rule has the same association confidence then the current rule is not key. This check cannot give rise to search space pruning. Therefore, it is done after checking the minimal association confidence constraint. If the rule is a key association rule then it is output.

Data: (B, H), i. e., a body and a head
forall $e \succ \max_{\prec}(H)$ **do**
  if $c_a(B \setminus \{e\} \rightarrow H \sqcup \{e\}) \geq \beta_a$ **then**
    forall $f \in \cup_{D^i \in \mathcal{D} \setminus \mathcal{D}_S} \pi_{D^i}(B \setminus \{e\})$ **do**
      if $c_a((B \setminus \{e\}) \setminus \{f\} \rightarrow H \sqcup \{e\} \sqcup \{f\}) = c_a(B \setminus \{e\} \rightarrow H \sqcup \{e\})$ **then**
        goto skip
    output $B \setminus \{e\} \rightarrow H \sqcup \{e\}$
    skip: KEY_ASSOCIATION_RULES$(B \setminus \{e\}, H \sqcup \{e\})$

Algorithm 8: KEY_ASSOCIATION_RULES.

4.3.3 Computing non-redundant rules

Let $\mathcal{P}$ denote the set of all key association rules on $\mathcal{D}'$ which are extracted thanks to KEY_ASSOCIATION_RULES (see Section 4.3.2). According to Theorem 9, we construct non-redundant multidimensional disjunctive rules of the form $X \rightarrow \vee\mathcal{Y}$ where $\mathcal{Y} = \cup_{X \rightarrow Y \in \mathcal{P}} Y$. Algorithm 9 only outputs the non-redundant multidimensional disjunctive rules whose disjunctive frequencies and disjunctive confidences exceed the user-defined thresholds. The whole process is presented in Algorithm 9).

4.4 Empirical study

We now evaluate our multidimensional disjunctive rule mining method thanks to experiments on the `Distrowatch` data. First, we interpret some rules. Second, we want to evaluate the performance of the CIDRE algorithm.

To emphasize an interesting relationship between distributions and countries, we look for multidimensional disjunctives rules on the domains *Distribution* and *Country* with the thresholds $\mu_a = 0.45$, $\mu_d = 0.6$, $\beta_a = 0.6$, $\beta_d = 0.8$. CIDRE extracts 81

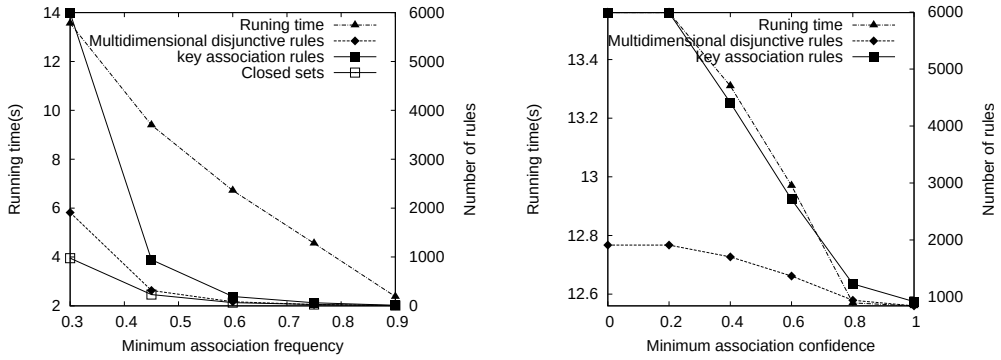
Input: A relation $\mathcal{R}$ on $\mathcal{D}$, $\mathcal{D}' \subsetneq \mathcal{D}$, and $(\mu_a, \beta_a, \mu_d, \beta_d) \in [0,1]^4$
Output: Every interesting and non-redundant disjunctive rule on $\mathcal{D}'$
$D^{\text{supp}} \leftarrow \times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$
$(\mathcal{D}_A, \mathcal{R}_A) \leftarrow (\mathcal{D}' \cup D^{\text{supp}}, \emptyset)$
forall $(e_1, \ldots, e_{|\mathcal{D}'|}, e_{|\mathcal{D}'|+1}, \ldots, e_n) \in \mathcal{R}$ **do**
  $\mathcal{R}_A \leftarrow \mathcal{R}_A \cup (e_1, \ldots, e_{|\mathcal{D}'|}, (e_{|\mathcal{D}'|+1}, \ldots, e_n))$
$\mathcal{C} \leftarrow$ Data-Peeler$(\emptyset, \times_{D^i \in \mathcal{D}_A} D^i)$
$\mathcal{P} \leftarrow \emptyset$
forall $C \in \mathcal{C}$ **do**
  $\mathcal{P} \leftarrow \mathcal{P} \cup$ Key_Association_Rules$(C \setminus \pi_{D^{supp}}(C), \emptyset)$
forall $X \rightarrow Y \in \mathcal{P}$ **do**
  $\mathcal{Y} \leftarrow Y$
  forall $X \rightarrow Y' \in \mathcal{P}$ *such that* $Y' \neq Y$ **do**
    $\mathcal{Y} \leftarrow \mathcal{Y} \cup Y'$
    delete $X \rightarrow Y'$ from $\mathcal{P}$
  if $(f_d(X \rightarrow \vee\mathcal{Y}) \geq \mu_d) \wedge (c_d(X \rightarrow \vee\mathcal{Y}) \geq \beta_d)$ **then**
    output $X \rightarrow \vee\mathcal{Y}$
  delete $X \rightarrow Y$ from $\mathcal{P}$

Algorithm 9: Cidre.

non-redundant rules. Here are some of them:

- $\{Biglinux\} \rightarrow (\{Brazil\})_{f_a:77, c_a:1}$
  $\vee(\{Brazil\} \times \{Goblinx\})_{f_a:62, c_a:0.8}$
  $\vee(\{Brazil\} \times \{Goblinx, Litrix\})_{f_a:0.54, c_a:0.7}$
  $\vee(\{Brazil\} \times \{Litrix\})_{f_a:0.69, c_a:0.9}$
  $(f_d : 0.77, c_d : 1)$,
- $\{Centos, Fedora\} \rightarrow (\{Taiwan\} \times \{B2D\})_{f_a:0.54, c_a:0.78}$
  $\vee(\{Taiwan, Japan\})_{f_a:0.46, c_a:0.67}$
  $\vee(\{Japan\} \times \{Berry\})_{f_a:0.62, c_a:0.89}$
  $\vee(\{Japan\} \times \{Berry, Momonga\})_{f_a:0.54, c_a:0.78}$
  $\vee(\{Japan\} \times \{Berry, Momonga, Plamo\})_{f_a:0.46, c_a:0.67}$
  $\vee(\{Japan\} \times \{Berry, Plamo\})_{f_a:0.54, c_a:0.78}$
  $(f_d : 0.69, c_d : 1)$,
- $\{Poland\} \times \{Kate\} \rightarrow (\{Linuxeducd\})_{f_a:0.46, c_a:0.75}$
  $\vee(\{PLD\})_{f_a:0.54, c_a:0.88}$
  $(f_d : 0.62, c_d : 1)$.

The first rule implies that *Biglinux* is especially interesting for Brazilians, and when a Brazilian consults it, then he/she usually shows interest in *Goblinx* or *Litrix* too. The second rule indicates that the people who consult *Centos* and *Fedora* at the same semester are Taiwanese and Japanese. When Taiwanese visitors look at them, they also visit *B2D*. In the case, the visitors are Japanese, they are also



(a) Pruning w.r.t. μ_a. (b) Pruning w.r.t. β_a.

Figure 17: Effectiveness of CIDRE.

interested in the distributions *Berry*, *Momonga* or *Plamo*. The third rule tells us that when Polish visitors look at *Kate* then it is sure that, at this time, they consult *Linuxeducd* or *PLD* as well.

We see that these rules suggest us more information than multidimensional association rules. We can now observe the relationship between countries and distributions even if visitors from these countries do not visit these distributions at the same time.

Let us finally provide a performance study when mining multidimensional disjunctive rules in $\mathcal{R}_{\texttt{DistroWatch}}$ with $\mathcal{D}' = \{Distribution, Country\}$. When the minimal association frequency threshold increases, CIDRE prunes large areas of the search space where no closed set is frequent. Consequently, both the number of frequent rules and the running time decrease. The experiments in Figure 17a were performed with the minimal association frequency (μ_a) varying from 0.3 to 0.9, $\mu_d = \mu_a$, $\beta_a = 0$ and $\beta_d = 0$. This figure indicates that when μ_a is small the number of multidimensional disjunctive rules compared with that of key association rules decreases significantly.

Theorem 6 also enables to deeply prune the search space. Indeed, Algorithm KEY_ASSOCIATION_RULES does not traverse the enumeration sub-trees empty of valid rules (w.r.t the minimal association confidence threshold). That is why both the number of rules and the time it takes to extract them decrease when the minimum association confidence threshold increases. The experiments in Figure 17b are performed with the minimal association confidence varying from 0 to 1, $\mu_a = \mu_d = 0.3$ and $\beta_d = \beta_a$. Here, we also see that the number of multidimensional disjunctive rules is much less than that of key association rules.

On the contrary, the search space cannot be pruned thanks to the thresholds on disjunctive frequency and disjunctive confidence. Indeed, CIDRE must consider every association rule when computing disjunctive ones.

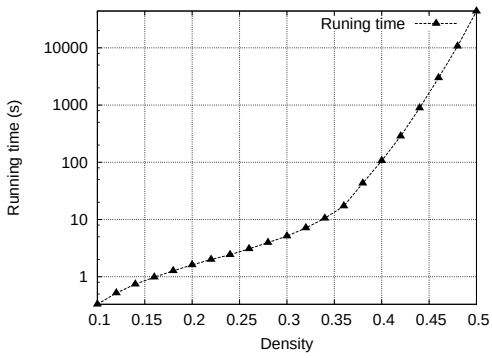


Figure 18: CIDRE's scalability w.r.t. the density.

CIDRE's scalability was tested w.r.t. the size and the density of the data. Starting with the size, rules on $\mathcal{D}' = \{\text{Countries, Distributions}\}$ are mined with $\mu_a = \mu_d = 0.3$ and $\beta_a = \beta_d = 0$. $\mathcal{R}_{\texttt{DistroWatch}}$ was replicated up to 10 times with the timestamps. It turns out that the algorithm scales linearly. More precisely, a linear regression of $R \mapsto \frac{T_R}{T_1}$ (where R is the replication factor, T_R is the running time on this replicated dataset) gives $y = 3.23x - 3.78$ with 0.96 as a determination coefficient.

To test the CIDRE's scalability w.r.t. the density of the dataset, synthetic 3-ary relations have been generated. The sizes of the domains were kept constant and equal to $10 \times 50 \times 100$. Here, the only variable is the density, i. e., the ratio between the number of 3-tuples of the relation and $10 \times 50 \times 100 = 50,000$. In our test, it increases 0.02 by 0.02, from 0.1 (for the first dataset) to 0.5 (for the last dataset). The experiments were performed with $\mu_a = \mu_d = 0.1$, $\beta_a = \beta_d = 0.5$. The CIDRE's running times are in Figure 18. As predicted, when the density is high, the extraction is much harder. However, let us note that 40% density is already extremely high in practice.

4.5 Conclusion

We considered the problem of mining multidimensional disjunctive rules in n-ary relations. Such a rule is an implication between associations: its body is an association and its head is a disjunction of associations. Enabling disjunctions within the heads of the rules provides rules that convey more information than conjunctive rules. We proposed two types of interestingness measures for evaluating the conjunction between its body and each association in its head (association measures), and the occurrence of at least one association in its head when its body holds (disjunctive measures). We considered the concept of a non-redundant multidimensional disjunctive rule having a minimal body and a maximal head. Theorem 8 and Theorem 9 show how such non-redundant rules are related to closed sets. The CIDRE algorithm

which discovers non-redundant and interesting multidimensional disjunctive rules in a n-ary relation is a post-processing of the state-of-the art algorithm DATA-PEELER. CIDRE prunes the search space by taking the minimal thresholds of association measures. On the contrary, the search space cannot be pruned thanks to the minimal thresholds of disjunctive measures because it must consider every key association rule when computing multidimensional disjunctive rule ones. Its performance was tested on real and synthetic datasets w.r.t the varieties of the dataset size and the dataset density. The experiment results show that CIDRE scales linearly with the dataset size, but it does not scale linearly with the the dataset density. The analysis of some multidimensional disjunctive rules extracted on the `DistroWatch` dataset has given a qualitative feedback on rule relevancy.

Part III

Application

Outline

Graphs are an universal data structure to model and to analyse the relationships between a set of entities. e.g., interactions between individuals in a social network. Applications of graphs arise in not only computer science, but also in physics, biology, economics, history, and finally in almost all fields of science. Therefore, graph mining has recently received a lot of attention in the data mining community.

We are here interested on specific graphs that are called relational and we want to investigate dynamic graphs. More and more researchers focus on dynamic graphs that describe the evolution of a graph over time. However, there are only a few works concerning descriptive rule mining from dynamic graphs. Their goals are to describe local evolution trends of the graph over time (e. g.,[111, 12]). For instance, such a rule means that if the sub-graph in the body appears at time t then the sub-graph in the head may appear in time $t + k$. Although this is valuable to describe the evolution of the graph, it does not explicit the simultaneousness of patterns in this evolution process.

In fact, "what are the patterns that can co-occur in the evolution of graph?" is also an important question. For example, in a dynamic graph whose time periods are cyclical, at what time does a bottleneck (i.e., many incoming edges) occur at a vertex? What vertices do outer edges tend to converge to?

We address this question based on multidimensional (association/disjunctive) rule mining proposed in the chapters 3 and 4. The experiments on a real-world dynamic graph illustrates the significance of our proposal.

Chapter 5

Rule discovery in dynamic relational graphs

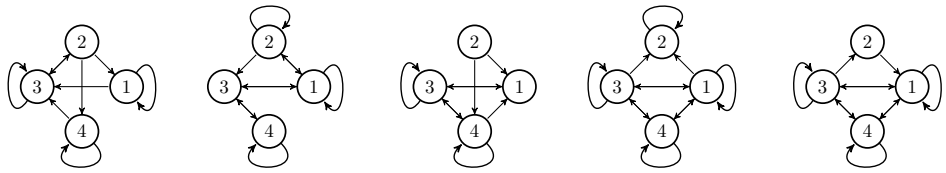
This chapter presents an approach for detecting patterns that can co-occur in dynamic graphs via multidimensional association/disjunctive rule mining. We report the experiments on the `Vélov'v` network that is a bicycle rental service run by the urban community of Lyon in France. Our goal is to report on the renting logs and to discover patterns that may help to improve the service.

This chapter begins by introducing rule mining in dynamic graphs. Next we discuss related work. We then report experimental results on the `Vélov'v` network (Section 5.3). Finally, we briefly conclude on this use case.

5.1 Mining multidimensional rules in dynamic graphs

5.1.1 Dynamic relational graphs

We investigate rule discovery from dynamic directed relational graphs, i. e., such a dynamic graph is a collection of static graphs that all share the same set of uniquely identified vertices. In our setting, given a set of vertices, directed edges can change (i. e., appear or disappear) through time. Thus, the considered dynamic graph can be modelled by a sequence of adjacency matrices. For example, Figure 19 depicts a dynamic directed graph involving four nodes through five time-stamps. The sequence of its adjacency matrices corresponds to a ternary relation $\mathcal{R}_G$ that describes the relationship between the departure vertices in $D^1 = \{d_1, d_2, d_3, d_4\}$ and the arrival vertices in $D^2 = \{a_1, a_2, a_3, a_4\}$ at the time-stamps in $D^3 = \{t_1, t_2, t_3, t_4, t_5\}$. Every '1' in $\mathcal{R}_G$, the intersection of three elements $(d_i, a_j, t_k) \in D^1 \times D^2 \times D^3$, indicates a directed edge from d_i to a_j at time t_k. Therefore, we need at least three dimensions to encode a dynamic relational graph as a n-ary relation. Two dimensions are used to encode the graph adjacency matrices and at least one other denotes



	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4
d_1	1		1		1	1	1				1		1	1	1	1	1		1	1
d_2	1		1	1	1	1	1		1			1		1			1			
d_3		1	1		1			1	1		1	1	1	1	1	1	1	1	1	1
d_4			1	1			1	1	1		1	1	1		1	1	1		1	1
	t_1				t_2				t_3				t_4				t_5			

Figure 19: $\mathcal{R}_G \subseteq \{d_1, d_2, d_3, d_4\} \times \{a_1, a_2, a_3, a_4\} \times \{t_1, t_2, t_3, t_4, t_5\}$.

time. However, more dimensions may be used to encode additional information on edges and/or time aspects.

5.1.2 Multidimensional rules in dynamic graphs

Given a dynamic graph, patterns can be sub-graphs, or can include subsets of arbitrary domains. In our approach where a dynamic graph is represented as a n-ary relation, a pattern can be expressed as an association in the n-ary relation (see Definition 32). *A multidimensional rule in the dynamic graph* is a multidimensional association rule (Definition 39) or a multidimensional disjunctive rule (Definition 48) in the n-ary relation. Such a multidimensional rule describes the simultaneousness of patterns in its body and its head. In particular, the temporal dimensions can either explicitly appear in the rules or be used to measure their relevancy (i. e., these dimensions "support" the rules).

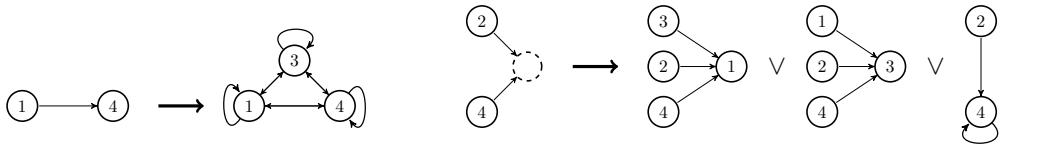
Some examples of rules are given in Figure 20 and Figure 21. The rules in Figure 20 show the relationship between departure vertices and time-stamps. For instance, the rule in Figure 20a indicates that the event in which the outer edges from Vertex 1 and Vertex 2 go to the same nodes usually happens at times t_1 and t_2. The rule in Figure 20b says that most of the arrival vertices of the edges from Vertex 3 at Time t_2 are also the arrival vertices of edges leaving this vertex at the times t_3, t_4 and t_5. In Figure 21, we provide examples of the rules related to both departure vertices and arrival vertices. Figure 21a describes the dependency between sub-networks. More precisely, it tells that the sub-network at its body can be enlarged to a clique with a high enough confidence. The rule in Figure 21b shows that if the edges from Vertex 2 and Vertex 4 converge, they tend to converge to Vertex 1, Vertex 3 or Vertex 4.

5.1.3 Discovering multidimensional rules in dynamic graphs

To discover multidimensional rules in dynamic graphs, we first represent the dynamic graph that we want to analyse as a n-ary relation. We then use the algorithms

(a) $\{d_1, d_2\} \to \{t_1, t_2\}$. (b) $\{d_3\} \times \{t_2\} \to \{t_3, t_4, t_5\}$.

Figure 20: Example of rules on $\{D^1, D^3\}$ in $\mathcal{R}_G$.



(a) $\{d_1\} \times \{a_4\} \to \{d_3, d_4\} \times \{a_1, a_3\}$. (b) $\{d_2, d_4\} \to (\{d_3\} \times \{a_1\}) \vee (\{d_1\} \times \{a_3\}) \vee (\{a_4\})$.

Figure 21: Example of rules on $\{D^1, D^2\}$ in $\mathcal{R}_G$.

PINARD++ and CIDRE to extract rules from this n-ary relation. The set $\mathcal{D}'$ that we choose depends on the analysis goal. For example, in $\mathcal{R}_G$, for discovering preferring departure time at vertices, the dimensions that we consider are departure vertices and time-stamps ($\mathcal{D}' = \{D^1, D^3\}$). Some of the rules are in Figure 20. For mining rules that conclude on graphs, we extract rules consisting of departure vertices and arrival vertices ($\mathcal{D}' = \{D^1, D^2\}$). Some of them are in Figure 21.

Some constraints can be added to the process to focuse on specific properties of the graph. For instance, one can define minimal numbers of elements in the domains that appear in the rules (i. e., the domains in $\mathcal{D}'$). For example, from the dynamic graph in Figure 19, let us look for rules that conclude on cross-graph cliques and such that these cliques involves at least two nodes. Therefore, the closed sets from which the rules are derived must contain at least two departure vertices and at least two arrival vertices.

Definition 54 ($(\alpha^i)_{i=1..|\mathcal{D}'|}$-min-sizes)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, given $(\alpha^i)_{i=1..|\mathcal{D}'|} \in \mathbb{N}^{|\mathcal{D}'|}$, a closed set C satisfies the minimal size constraint on $\mathcal{D}'$, $\mathcal{C}_{(\alpha^i)_{i=1..|\mathcal{D}'|}\text{-min-sizes}}(C)$, iff*

$$\bigwedge_{i=1..|\mathcal{D}'|} (|\pi_{D^i}(C)| \geq \alpha^i).$$

Another constraint that we should consider is to enforce a cross-graph closed clique property to find rules that conclude on cliques. For this, an extracted closed set must satisfy the *symmetry* constraint between the set of departure vertices and the set of arrival vertices. In Figure 22, $\{d_1, d_3, d_4\} \times \{a_1, a_3, a_4\} \times \{t_4, t_5\}$ is such a closed set on $\mathcal{D}$. It is not only a closed 3-set but also a cross-graph closed clique

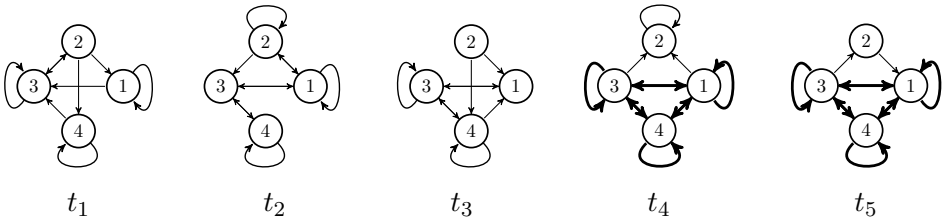


Figure 22: Maximal clique $\{1, 3, 4\}$ preserved along two timestamps.

(between the vertices 1, 3 and 4; at the times t_4 and t_5). Notice that the *closedness* ensures the maximality of the clique, i. e., it cannot be enlarged into another one that would still hold at both t_4 *and* t_5. It also ensures its "maximality on time", i. e., the clique does not appear in any other snapshot of the graph.

Definition 55 (Cross-graph closed clique)**.** *A closed set C is a cross-graph closed clique iff it satisfies the cross-graph clique constraint $\mathcal{C}_{cross\text{-}graph}(C) \equiv \pi_{D^{dep}}(C) = \pi_{D^{arr}}(C)$, where D^{dep} (resp. D^{arr}) is the set of departure (resp. arrival) vertices.*

It is easy to see that the first constraint is monotone and that the second constraint is piecewise (anti)-monotone constraint. Both classes of constraints can be enforced to provide faster extractions of closed sets with Data-Peeler [33]. Let us recall that Pinard++ and Cidre post-process the output of Data-Peeler. Therefore, these constraints also improve the efficiency of rule mining as well. This will be illustrated in our experiments on a real-life network (Section 5.3).

5.2 Related work

The study of graphs has attracted much attention in the last few years. Many papers study the evolution of graphs over time with a large variety of techniques. In these studies, we observe two complementary directions of research. First, several papers have focused on the evolution of macroscopic graph properties [100, 35, 106, 64, 5] where some have concerned the dynamical properties of real graphs such as densification laws, shrinking diameters [67], and the evolution of known communities over time [7, 64]. Second, some works have studied graph evolution at a local level thanks to local patterns. This section focuses on such methods.

In [19], Borgwardt et al. extract local patterns in labeled dynamic graphs. The approach aims at finding subgraphs that are topologically frequent and show an identical dynamic behavior over time, i. e., insertions and deletions of edges occur in the same order. Because this task is computationally hard, the algorithm is not complete. Indeed, computing the overlap-based support measure means solving a maximal independent set problem for which they propose a greedy algorithm. Inokuchi and Washio introduce a fast algorithm to mine frequent transformation

subsequences from a set of dynamic labeled graphs (the labels on vertices and edges can change over time). Assuming that the changes in a dynamic graph are gradual, they propose to succinctly represent the dynamics with a graph grammar: each change between two observed successive graph states is interpolated by axiomatic transformation rules. A significant improvement is proposed in [54]. Motivated by the intractability of their approach on long sequences of large graphs, the same authors define, in [55], induced subgraph subsequence. This novel class of subgraph subsequence enables to efficiently mine frequent patterns from graph sequences containing long sequences and large graphs. In [65], Lahiri et al. introduce the periodic subgraph mining problem, i. e., identifying every frequent closed periodic subgraph. They empirically study the efficiency and the interest of their proposal on several real-world dynamic social networks. By encoding dynamic graphs as ternary relations, [34] describes a constraint-based mining approach to discover maximal cliques that are preserved over almost-contiguous time-stamps. The constraints are pushed into the closed pattern mining algorithm Data-Peeler. In [88], Robardet proposes a constraint-based approach too. It studies the evolution of dense and isolated subgraphs defined by two user-parameterized constraints. Associating a temporal event type with each pattern captures the temporal evolution of the identified subgraph, i. e., the formation, dissolution, growth, diminution and stability of subgraphs between two consecutive time-stamps. The algorithm incrementally processes the time series of graphs.

In [111], You et al. study how a graph is structurally transformed through time. They compute graph rewriting rules that describe the evolution of two consecutive graphs. These rules are then abstracted into patterns representing the dynamics of a sequence of graphs. The main step concerns the computation of maximum common subgraphs between two consecutive graphs. This problem is NP-complete. In the case of relational graphs (graphs with unique vertex labels such as the ones tackled by us), it remains tractable [26, 29]. Indeed, the complexity is then quadratic and graph rewriting rules are efficiently discovered. In [13], the authors focus on detecting clusters of temporal snapshots of an evolving network. These clusters can be interpreted as evolution eras of the dynamic graph. This approach enables to detect periods in which sudden change of "behaviour" appears. Such high-level trends are expressed by sudden increases or decreases of the similarity between the structures of the consecutive graphs. In [12], the authors introduce the problem of extracting graph evolution rules satisfying minimal support and confidence constraints. It finds isomorphic subgraphs that match the time-stamps associated with each edge, and, if present, the properties of the vertices and edges of the dynamic graph. Graph evolution rules are then derived with two different confidence measures. Nevertheless, this work focuses on the dynamic changes in the graph whereas we provide a generic framework to discover multidimensional rules that implicate the simultaneousness of patterns, in which the time is either in the rule or in its support.

5.3 A case study

5.3.1 Dataset: Vélov'v

Vélov'v is a bicycle rental service run by the urban community of Lyon, France. Vélov'v stations are spread over Lyon and its nearby. One of them is depicted in Figure 23 [1]. At any of these stations, the users can take a bicycle and return it to any other station. Whenever a bicycle is rented or returned, this event is logged. Our research group obtained parts of these logs (e. g., no user identification to preserve privacy) recorded between May 27th 2005 (when the system was opened to the public) and December 17th 2007. They represent more than 13.1 million rides. Encoding this graph data consists of the two following steps that have been set up by Cerf in [31].

Figure 23: A Vélov'v station.

The first, we remove "abnormal" records from the dataset. The earliest records relate to the users discovering Vélov'v and how useful it may be in their daily mobility. To study the network usage in "normal" conditions, these earliest records were ignored. The chosen date, after which the considered dataset starts, is December 17th 2005. In this way, two full years are kept and aggregations do not favor any part of the year (along which the network usage evolves). Many records stand for rides from a station to itself. These rides usually last a few seconds. They can be mainly explained by users who are not satisfied by the quality of the bicycle they have just rent (e. g., a flat tire) or who have changed their mind (e. g., a bus arrives). Because, from a given station, the most frequent rides are to itself, keeping these records influence a lot any normalization procedure. That is why these records are removed but, after the post-processing steps, the related routes are all claimed frequent, i. e., appended to the relation. A few more records were removed. They relate to abnormal rides (incoherent dates) or rides implying stations that are not

1. © 2005 Frédéric Bonifas (from Wikimedia Commons)

opened to the public (e. g., where bicycles are repaired). About 10.2 million records remain after these first steps.

The second, we represent `Vélo'v` data as a dynamic directed graph evolving into two temporal dimensions: the 7 days of the week and the 24 one-hour periods in a day. The vertices correspond to the `Vélov'v` stations. The edges are labelled with the total number of rides from the head vertex (departure station) to the tail vertex (arrival station) during the considered period of hour and day of the week. To normalize the data, we use a local test inspired by the computation of a p-value. At each time period (a day, a hour period), it considers the vertices one by one, computes the sum S of the labels of both its incoming and outgoing edges, and claims frequent the routes related to the edges with the greatest values and whose sum is just beyond $0.1 \times S$. By definition, this procedure keeps at least one edge involving each station. In average, 191 edges per station are kept (still excluding the reflexive routes). In this way, each retained edge corresponds to a significant amount of rides from the (departure) station ds to the (arrival) station as on day d (e. g., Monday) at hour h (e. g., between 1pm and 2pm). When the data are binarized, the Boolean predicate decides whether routes are frequent w.r.t. time period (a day, a hour period). In other terms, (ds, as, d, h) belongs to the relation $\mathcal{R}_{\texttt{Vélov'v}} \subseteq$ *Departure* $\times$ *Arrival* $\times$ *Day* $\times$ *Hour*. This relation contains $117,411$ 4-tuples, hence a $\frac{117,411}{7\times24\times327\times327} = 0.7\%$ density.

In the following, from $\mathcal{R}_{\texttt{Vélov'v}}$, our goals are to detect:

- Preferred time periods (days and hours) of departures and arrivals at stations.
- Time periods of the exchange of bicycles between stations. It means that we want to answer the following questions: If the people take a bicycle at one station then what station do they tend to return it? And when do this occur?
- When can stations be blocked? A stations is blocked when it is empty (no bicycle can be rented from it) or when it is full (no bicycle can be returned to it).

5.3.2 Mining multidimensional association rules in `Vélov'v` graphs

We report on our experimental results when running our algorithm PINARD++ on dataset `Vélo'v`. We show that PINARD++ can mine several useful types of multidimensional association rules on the dataset when we use different values of input parameters that depend on our mining objectives. The experiments of this section aim at discovering preferred time periods (days and hours) of departures and arrivals at stations; preferred hours for the exchange of bicycles between stations, and frequent usage sub-networks that can confidently be enlarged into cliques. Finally, we also evaluate PINARD++'s efficiency w.r.t constraints.

We first describe our results on detecting preferred time periods of departures and arrivals at stations.

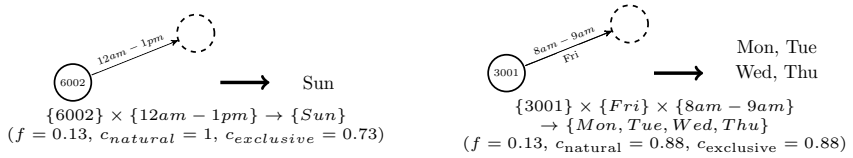


Figure 24: Example of rules on $\{Departure, Day, Hour\}$.

To detect preferred time periods of departures at stations, we mine rules that include departure stations, days and hours ($\mathcal{D}' = \{Departure, Day, Hour\}$), and their support domain is *Arrival* (327 arrival stations). To investigate preferred time periods of arrivals at stations, we mine rules that contain arrival stations, days and hours ($\mathcal{D}' = \{Arrival, Day, Hour\}$), and their support domain is *Departure*(327 departure stations). The experiments are done with the minimal frequency threshold μ varying from 0.06 to 0.2, the minimal exclusive confidence threshold $\beta_{\text{exclusive}}$ and the minimal natural confidence β_{natural} varying from 0.6 to 1. When observing the computed rules, we see that their meaning is consistent with the available background knowledge. For instance, the frequency of using bicycles at stations being near railway stations, shopping centers, entertainment centers and universities appear to be higher than that of the other stations. Concerning preferred time periods when the people take (or return) bicycles at a station, the rules tell us that the departure/arrival time periods of stations next to railway stations are often close to the arrival/departure hours of trains. Also, the preferred time for using bicycles from stations being near entertainment centers is usually the weekend.

Some of the extracted rules are reported below.

With $\mathcal{D}' = \{Departure, Day, Hour\}$, $\mu = 0.12$, $\beta_{\text{natural}} = 0.8$ and $\beta_{\text{exclusive}} = 0.6$, PINARD++ extracts 632 rules. Figure 24a and Figure 24b report two of them. The rule in Figure 24a shows that the departures from Station 6002 between 12am and 1pm almost exclusively occur on Sundays ($c_{\text{exclusive}} = 0.73$). The natural confidence is 1, i. e., whatever the arrival station, the frequent rides from Station 6002 between midday and 1pm all occur on Sundays. This is consistent with our knowledge of the city because Station 6002 is at the main entrance of the most popular park, where people like to take bicycles for coming back home, hence the high frequency in terms of number of arrival stations. The rule in Figure 24b indicates that the rides from Station 3001 between 8am and 9am usually occur during the working days. This is again consistent with our knowledge that many people living outside Lyon come to work by train and Station 3001 is the closest to the train station in the main working area of the city. It turns out that they then finish their daily trips to work by bicycle.

With $\mathcal{D}' = \{Arrival, Day, Hour\}$, $\mu = 0.06$, $\beta_{\text{natural}} = 0.8$ and $\beta_{\text{exclusive}} = 0.6$,

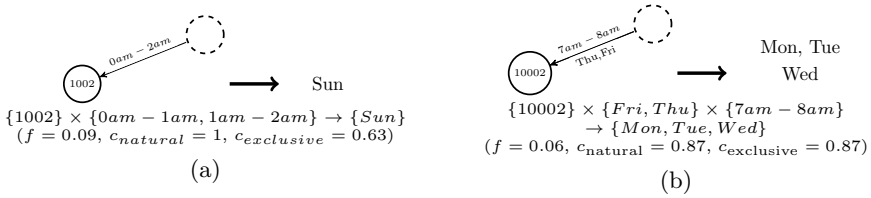
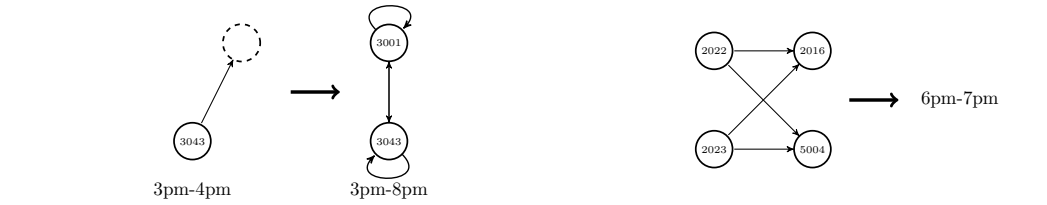


Figure 25: Example of rules on $\{Arrival, Day, Hour\}$.

PINARD++ extracts 2494 rules. Figure 25 presents two of the rules. The rule in Figure 25a shows that we rarely observe arrivals to Station 1002 between 0am and 2am except on Sundays ($c_{exclusive} = 0.63$). In fact, this station is located in a district with many pubs and the favoured time to go to pubs is after parties on Saturday evenings. At this time, the public transportation services stop, thus `Vélo'v` is a good alternative to go to pubs. Observe the Station 10002 appearing in the rule of Figure 25b, we see that it is in located on a campus called La Doua-LyonTech. This is a large campus which encompasses many other science-oriented schools and universities. Here, the school day begins at 7am, and students like taking bicycles to go to school. That explains why the frequency of arrivals to Station 10002 between 7am and 8am on weekdays is higher than the other time. This rule also indicates that most of departure stations of routes arriving Station 10002 between 7am and 8am on Thursdays and Fridays are also that of routes arriving Station 10002 between 7am and 8am on Mondays, Tuesdays and Wednesdays ($c_{natural} = c_{exclusive} = 0.87$).

Second, we present our results on rules related to preferred hours of the exchanges of bicycles between stations in every day. We extract rules on $\mathcal{D}' = \{Departure, Arrival, Hour\}$. To focus on rules that hold every day, the minimal frequency threshold is set to 1. Consequently, the natural confidence of the rules is always 1. The experiments are made with $\beta_{\text{exclusive}}$ varies from 0.5 to 0.9. The discovered rules show that these preferred hours are only from 1pm to 9pm, never at the other hour. For example, with $\beta_{\text{exclusive}} = 0.6$, PINARD++ returns 384 rules involving at least one hour, two departure stations and two arrival stations. Figure 26 depicts two of them. The rule in Figure 26a means that there is always the exchange of bicycles between Station 3001 and Station 3043 from 3pm to 8pm in every day ($f = 1$ and $c_{natural} = 1$). This rule also indicates that the arrival stations of departures from Station 3043 in 3pm-4pm are almost only Station 3001 and Station 3043 ($c_{exclusive} = 0.74$). The rule in Figure 26b shows that there is always the exchanges bicycles between stations 2022, 2023 and stations 5004, 2016 from 6pm to 7pm in every day ($f = 1$ and $c_{natural} = 1$). And these exchanges rarely occur in another hour ($c_{exclusive} = 0.78$).

We now consider patterns on graph evolution: we want to look at frequent usage sub-networks (i. e., sub-networks that are often observed) that can confidently be



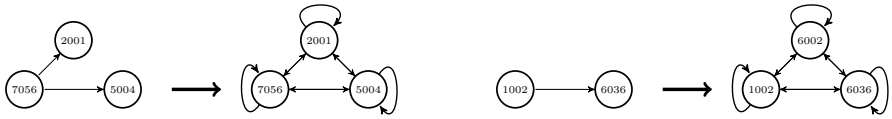
$\{3pm - 4pm\} \times \{3043\} \rightarrow \{4pm - 8pm\} \times \{3001\} \times \{3043, 3001\}$
$(f = 1,\ c_{\text{natural}} = 1,\ c_{\text{exclusive}} = 0.74)$

(a)

$\{2022, 2023\} \times \{5004, 2016\} \rightarrow \{6pm - 7pm\}$
$(f = 1,\ c_{\text{natural}} = 1,\ c_{\text{exclusive}} = 0.78)$

(b)

Figure 26: Example of rules on $\{Departure, Arrival, Hour\}$.



$\{7056\} \times \{5004, 2001\} \rightarrow \{5004, 2001\} \times \{7056\}$
$(f = 0.05,\ c_{\text{natural}} = 0.9,\ c_{\text{exclusive}} = 0.9)$

(a)

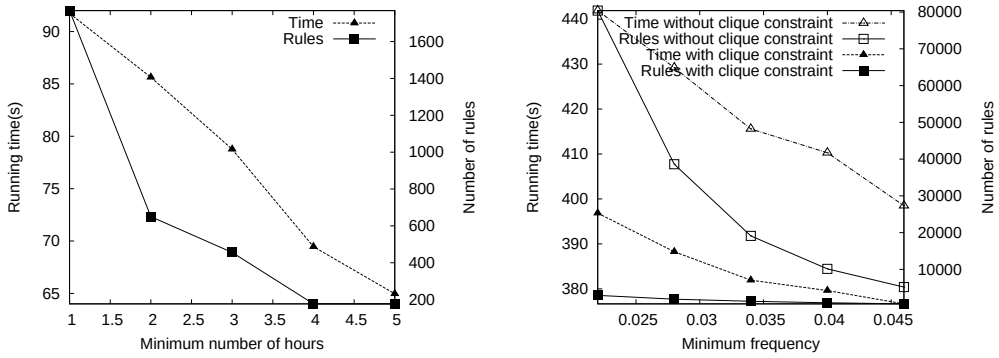
$\{1002\} \times \{6036\} \rightarrow \{6036, 6002\} \times \{1002, 6002\}$
$(f = 0.04,\ c_{\text{natural}} = 1,\ c_{\text{exclusive}} = 1)$

(b)

Figure 27: Example of rules of the form "sub-network" $\rightarrow$ "maximal clique".

enlarged into cliques? To study such patterns, a rule has to involve *Departure* and *Arrival* stations, i. e., $\mathcal{D}' = \{Departure, Arrival\}$. As a result, the support domain is the Cartesian product of the 7 days and the 24 hours. Additional constraints, defined in Sect. 5.1.3, are enforced so that PINARD++ processes $(3, 3)$-min-sizes cross-graph closed cliques into rules. Moreover we force the body of every rule to be a graph with at least one edge, i. e., it must involve at least one departure station and one arrival station. The non-redundancy of the extracted rules favours the discovery of minimal sub-networks (at the bodies of the rules) that can be confidently (i. e., with a high enough confidence) enlarged into maximal cliques (unions of the bodies and the heads). With $\mu = 0.02$ and $\beta_{\text{natural}} = \beta_{\text{exclusive}} = 0.7$, 165 rules are discovered. Some of them are reported in Figure 27. The enlarged sub-networks can contain only more edges (see Figure 27a) or more vertices (see Figure 27b). The extracted rules display the influence of the bicycle exchanges between stations to that between other stations.

We finally report the effectiveness of pruning search spaces of PINARD++ thanks to the min-sizes and cross-graph closed clique constrains. First, to test the performance of PINARD++ with the min-sizes constraint, we extract rules which describe the exchange of bicycles between stations at favour hours in every day, and such a rule has to include at least two departure stations and two arrival stations. Thus, $\mathcal{D}' = \{Depart, Arrival, Hour\}$, $D^{supp} = Day$, μ is set to 1 (consequently, $c_{natural}$



(a) Min-sizes constraint. (b) Cross-graph closed clique constraint.

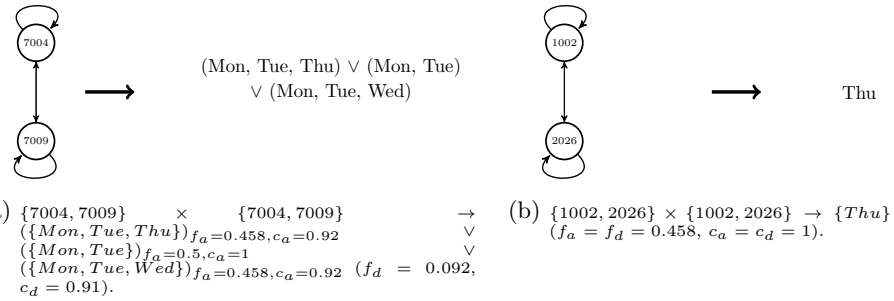
Figure 28: Efficiency of PINARD++ with constraints

is always 1). The experiments are done with $\beta_{exclusive} = 0$, and the minimum number of hours when stations have exchanging bicycles in all days varies from 1 to 5. Figure 28a shows that, when the minimum number of hours increases, the number of rules and the running time decrease. Next, to test the PINARD++'s performance with the cross-graph closed clique constraint, we mine rules of the form "sub-network → maximal clique" and of the form "sub-network → larger network". The experiments are done with the min-sizes constraint $\mathcal{C}_{(3,3)-min-sizes}$, $\beta_{natural} = \beta_{exclusive} = 0$ and μ varying from 0.022 to 0.046. As we see in Figure28b, the number of rules and the running time with the cross-graph closed clique constraint are always lower than that without this constraint.

5.3.3 Mining multidimensional disjunctive rules in `Vélov'v` graphs

We now present our experimental results by running our algorithm CIDRE on dataset `Vélo'v`. We illustrate the computation of several useful types of multidimensional association rules on this dataset, by using different values of input parameters that depend on our mining objectives. The experiments of this section aim at discovering preferred days of the exchange of bicycles between stations; time periods when stations can be blocked; convergence points of departures from different stations. Finally, it is used to further report on CIDRE's efficiency w.r.t constraints.

We first describe our results on discovering days when stations have exchanges of bicycles in many hours. For discovering days when stations have exchanges of bicycles in many hours, we mine rules on $\mathcal{D}' = \{Departure, Arrival, Day\}$, the supports of the rules are sets of hours. To focus on rules that hold many hours, the minimal association frequency threshold (μ_a) varies from 0.375 (respective 9 hours) to 0.5 (respective 12 hours). The experiments are done with $\mu_d = \mu_a$ and $\beta_a = \beta_d$ varies from 0 to 1. For example, with $\mu_a = \mu_d = 0.45$ and $\beta_a = \beta_d = 0.8$, CIDRE



(a) $\{7004, 7009\} \times \{7004, 7009\} \rightarrow (\{Mon, Tue, Thu\})_{f_a=0.458, c_a=0.92} \vee (\{Mon, Tue\})_{f_a=0.5, c_a=1} \vee (\{Mon, Tue, Wed\})_{f_a=0.458, c_a=0.92}$ ($f_d = 0.092$, $c_d = 0.91$).

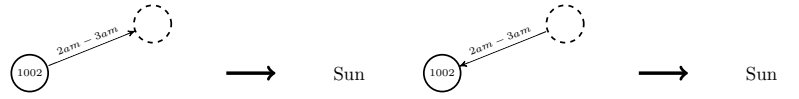
(b) $\{1002, 2026\} \times \{1002, 2026\} \rightarrow \{Thu\}$ ($f_a = f_d = 0.458$, $c_a = c_d = 1$).

Figure 29: Example of rules on $\{Departure, Arrival, Day\}$.

outputs 29 rules. Figure 29 presents two of them. The rule in Figure 29a shows that the days that Station 7004 and Station 7009 have exchanges of bicycles in many hours are from Monday to Thursday. The rule in Figure 29b indicates that the day that Station 1002 and Station 2026 have exchanges of bicycles in many hours is only Thursday.

Second, we present our results on rules which detect time periods when can stations be blocked. When a station is blocked, i. e.., it is empty (or full), the user can not rent (or return) a bicycle at this station. Therefore, detect when stations are blocked is important to improve the fulfilled service. Because the number of bike posts of each station is finite (< 40). So, a station can be empty when its departures are very more than its arrivals. On the contrary, a station can be full when its arrivals are a lot more than its departures. To know the number of departures of stations, we mine disjunctive association rules on $\mathcal{D}' = \{Departure, Day, Hour\}$, their support domain consists of 327 arrival stations. To know the number of arrivals of stations, we extract disjunctive association rules on $\mathcal{D}' = \{Arrival, Day, Hour\}$, their support domain containing 327 departure stations. These experiments are done with μ_a ($= \mu_d$) varying from 0.02 to 0.2 and β_a ($= \beta_d$) varying from 0 to 1. Two examples of blocked stations are given in Figure 30 and Figure 31. Station 1002 has 22 bike posts. As we see the rule in Figure30a, there is 38 departures from Station 1002 ($f_a = f_d = 0.116$, $\lceil 0.116 \times 327 \rceil = 38$) during from 2am to 3am on Sunday, with the confidence 0.90. But, the rule in Figure 30b indicates that Station 1002 only has 17 arrivals at this time ($f_a = f_d = 0.049$, $\lceil 0.049 \times 327 \rceil = 17$) with the confidence 0.84. Consequently, during from 2am to 3am on Sunday, Station 1002 has a lot of departures, but it has a few arrivals. Therefore, Station 1002 can be empty between 2am and 3am on Sunday.

Station 6002 has 29 bike posts. Figure31a shows that, during from 2pm to 3pm on Sunday, the number of departures from Station 6002 is 31 ($f_d = 0.092$, $\lceil 0.092 \times 327 \rceil = 31$) with the confidence 0.91. However, the rule in Figure 31b means that the number of arrivals of Station 6002 at this time is 53 ($f_d = 0.162$, $\lceil 0.162 \times 327 \rceil = 53$)



(a) $\{1002\} \times \{2am - 3am\} \rightarrow \{Sun\}$ $(f_a = f_d = 0.116,\ c_a = c_d = 0.90)$.

(b) $\{1002\} \times \{2am - 3am\} \rightarrow \{Sun\}$ $(f_a = f_d = 0.049,\ c_a = c_d = 0.84)$.

Figure 30: Example of empty stations.

(a) $\{6002\} \times \{2pm - 3pm\} \rightarrow (\{Sun\})_{f_a=0.092, c_a=0.91} \vee (\{Sun\} \times \{5pm - 6pm\})_{f_a=0.083, c_a=0.82} \vee (\{Sun\} \times \{6pm - 7pm\})_{f_a=0.089, c_a=0.88}$ $(f_d = 0.092,\ c_d = 0.91)$.

(b) $\{6002\} \times \{2pm - 3pm\} \rightarrow (\{Sun\})_{f_a=0.168, c_a=0.93} \vee (\{Sun\} \times \{3pm - 4pm\})_{f_a=0.147, c_a=0.81}$ $(f_d = 0.168,\ c_d = 0.93)$.

Figure 31: Example of full stations.

with the confidence 0.93. Because, during from 2pm to 3pm on Sunday, the number of arrivals of Station 6002 is very higher than the number of departures. So, Station 6002 can be full between 2pm and 3pm on Sunday.

We now consider the graph evolution, to know convergence points of departures from different station, we extract rules on $\mathcal{D}' = \{Departure, Arrival\}$ whose bodies are only departure stations (with at least 2 departure stations), whose heads conclude on arrival stations. Consequently, the support domain is the Cartesian product of the 7 days and the 24 hours. With $\mu_a = \mu_d = 0.25$ and $\beta_a = \beta_d = 0.8$, 249 rules are discovered. Figure 32 are two of these rules. The rule in Figure 32a shows that when the outer edges (departures) from Station 2008 and Station 7033 go to the same station then they tend to converge to Station 7033 or Station 2008. At time when they converge to Station 2008, the outer edge from Station 7035 also converges to Station 2008. With the constraint that the confidence of convergence is at least 0.8 ($\beta_a \geq 0.8$), the rule in Figure 32b indicates that the outer edges from the stations 6007, 6011 and 6031 only converge to Station 3003. And at every time when they converge the outer edge from Station 3032 tends to go to this convergent point.

We finally report the effectiveness of pruning search spaces of CIDRE thanks to the min-sizes and cross-graph closed clique constrains. First, to test the performance of CIDRE with the min-sizes constraint, we extract rules which describe convergences of outer edges (departures) from stations, i. e., rules on $\mathcal{D}' = \{Departure, Arrival\}$, their bodies are only departure stations and their heads conclude on arrival stations. The support domain of the rules is the Cartesian product of the 7 days and the 24 hours. The experiments are done with $\mu_a = \mu_d = 0.25$, $\beta_a = \beta_d = 0$ and the minimum number of departure stations in the body of each rule varies from 2 to 6.

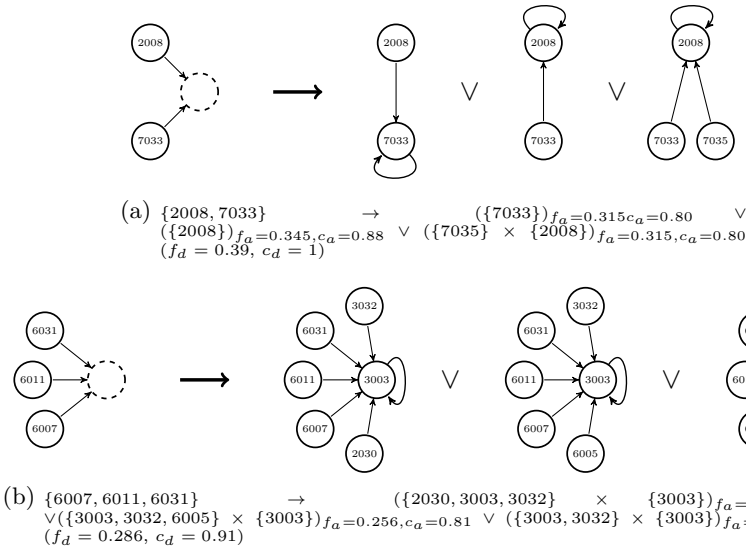
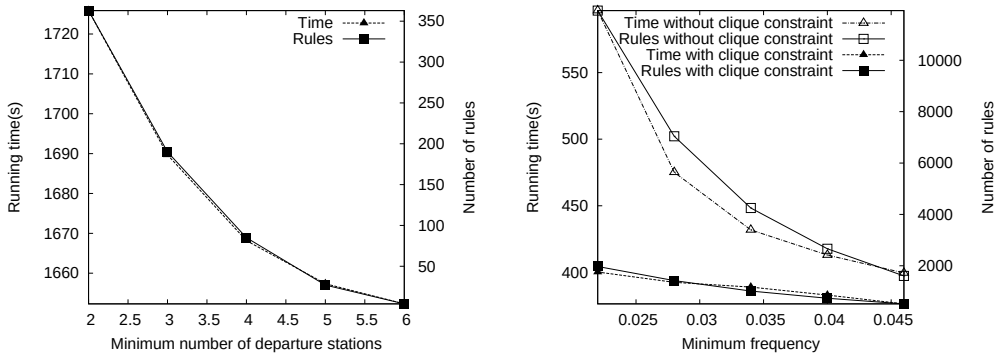


Figure 32: Example of rules that denote convergences.

Figure 33a shows that when the minimum number of departure stations increases the number of rules and the running time decrease. Second, to test the CIDRE's performance with the cross-graph closed clique constraint, we mine rules of the form "sub-network → maximal clique" and of form "sub-network → larger network". The experiments are done with the min-sizes constraint is $\mathcal{C}_{(3,3)-min-sizes}$, $\beta_{natural} = \beta_{exclusive} = 0$ and μ varies from 0.022 to 0.046. As we see in Figure33b, the number of rules and the running time of mining rules with the cross-graph closed clique constraint are always lower than that of mining rules without this constraint. We conclude that the perform of CIDRE is more effective when we add more piecewise (anti)-monotone constraints on input information because the algorithm prunes more branches on the search space thanks to these constraints.

5.4 Conclusion

We presented a solution to the problem of detecting the simultaneousness of patterns in dynamic relational graphs via rule mining. The approach represents a dynamic graph as a n-ary relation, and a pattern in the dynamic graph is expressed as an association in the relation that encodes this graph. Thus, the simultaneousness of patterns in the dynamic graph corresponds to that of associations in the body and the head of a multidimensional association (or disjunctive) rule that holds in the n-ary relation. We can apply the algorithms proposed in the chapters 3 and 4 to extract relevant rules. We presented some constraints that not only allow us to



(a) Min-sizes constraint. (b) Cross-graph closed clique constraint.

Figure 33: Effectiveness of CIDRE with constraints

extract specific rules (w.r.t. subjective interestingness), but also improve the overall efficiency of the extraction phase. The added-value of our rule mining techniques has been demonstrated on the `Vélo'v` dataset. Interpreting the discovered rules helps to better understand "How `Vélo'v` is used".

Conclusion

Mining descriptive rules which aims at discovering interesting associations among elements in databases is one of the most popular data mining task. For instance, association rule mining in large binary relations (2-ary relations) has been extensively studied. However, many datasets of interest correspond to relations whose number of dimensions is greater or equal to 3. Therefore, we have studied the generalization of association rule mining in n-ary relations when $n > 2$. We now summarize our key results and we discuss directions for future research.

Summary of contributions

Our main objective has been to study two descriptive rule types within n-ary relations, namely multidimensional association rules and the multidimensional disjunctive rules. The contribution includes the design of the declarative specification of their *a priori* interestingness in the context of arbitrary n-ary relations. Furthermore, we had to implement effective methods to extract relevant collections of rules. Like every research in data mining, empirical studies are useful to assess the performances of the algorithms but also the added-value of the computed paterns in real-life settings. Among others, we have presented a use case about dynamic graph analysis based on our types of rules.

We first generalized the concept of association rules introduced by Agrawal et al. [2] to the context of n-ary relations by formalizing multidimensional association rules. Such a multidimensional association rule is an implication between two associations where each association can include subsets of some arbitrary domains. It turns out that a natural generalization of rule frequency exists. On the contrary, it is fairly hard to define a confidence measure for multidimensional rules because, for a given rule, the support domain of its body and the support domain of the rule cannot be the same. We proposed two solutions for the above problem. The first solution is to compute the confidence of the rule on the support domain of the body,

the confidence measure is named "exclusive confidence". When the exclusive confidence is high, it means that the association in the body prefers connecting with the association in the head to connecting with other elements. The second solution is to compute the confidence of the rule on its support domain, the confidence measure is named "natural confidence". The natural confidence of the rule is the probability of observing its head when its body holds on the support domain of the rule. We revisited the concept of non redundancy for multidimensional association rules. A rule is non redundant if its information is not included in that of another general rule. Such a non-redundant rule has a minimal body and a maximal head w.r.t rule frequency and rule confidences. Therefore, we consider that a rule is non-redundant if it is derived from a closed set and if the confidences of every more general rule which is derived from the same closed set exceed its own measures.

Second, we have tackled the problem of mining multidimensional disjunctive rules in n-ary relations. Such a rule is an implication between associations: its body is an association and its head is a disjunction of associations. Enabling disjunctions within the heads of the rules provides rules that convey more information than conjunctive rules. The measures for evaluating the interestingness of such a rule are computed on the support domain of its body. We proposed two types of interestingness measures. First, the association measures are to evaluate the conjunction between its body and each association in its head. Second, the disjunctive measures are to evaluate the occurrence of at least one association in its head when its body holds. Because the number of associations that can frequently co-occur with the given frequent association may be large, the number of multidimensional disjunctive rules satisfying the minimal objective measure constraints can be huge. It is computationally expensive to find all such rules. We propose to extract only the rules that give us the most complete information: such a rule has a minimal body and a maximal head. Again, we provide to the analyst non redundant rules that are related to closed sets.

The algorithms for mining multidimensional association rules and multidimensional disjunctive rules in n-ary relations perform a post-processing over patterns computed with the state-of-the art algorithm Data-Peeler. Therefore, our algorithms take advantage of the pruning techniques of Data-Peeler. In addition, they can reduce the search space by using several minimal confidence constraints. The experiments on both synthetic and real datasets show that our algorithms scale linearly with the dataset size but not with the dataset density.

Finaly, we proposed an application for mining interesting rules in relational dynamic graphs that can be encoded into n-ary relations ($n \geq 3$) Our goal was to discover patterns which can co-occur in the evolution of a such dynamic graph. Patterns in the dynamic graph can be expressed as associations in the n-ary relation. Detecting the simultaneousness of patterns in the dynamic graph corresponds to mining descriptive rules in the n-ary relation. In particular, the temporal dimen-

sions can either explicitly appear in the rules or be used to measure their relevancy (i. e., these dimensions "support" the rules). The interest of discovering such rules is demonstrated on the `Vélo'v` network and a real use case.

Future Research Directions

We have made a significant contribution to the generalization of descriptive rule mining when considering arbitrary n-ary relations. We still have to face many interesting problems when considering the quality of the computed rules and their potential use in various application domains.

Future research can proceed along the following directions: (1) Mining multilevel rules and improving objective interestingness measures; (2) Mining rules in noisy datasets; (3) Extending the rule pattern domains towards different languages and semantics. Furthermore, we are convinced that dynamic graph analysis will be a major application domain for data mining in the near future. Therefore, we consider that a promising perspective of this thesis concerns the assesment of our rule mining methods for solving important problems in large graphs (e.g., online social networks) and probably the design of new pattern domain dedicated to dynamic graph analysis.

Mining multilevel rules and improving objective interestingness measures

In this thesis, we only mention the mining of multidimensional association rules and multidimensional disjunctive rules at a single level. Also, we designed only frequency and confidence measures. Therefore one first perspective would be to mine rules that may span levels of taxonomies on the different domains. The problem is to design a new mining method. Indeed, if we start from the approach of Srikant et al. in [97], then the size of the extended n-ary relation will explode as soon as many dimensions are associated to taxonomies. If we start from the approach of Chen et al. in [49], mined rule may include only elements belonging to a same level. A second important extension can be to design other measures of interestingness which allow us to remove non interesting rules and rank patterns for the needed interpretation phase.

Mining rules in noisy datasets

Real n-ary relations suffer from noise that can have several causes (i. e., intrinsic noise in the studied system, erroneous measures, mis-parameterized pre-processing steps, etc). For instance, the computation of error-tolerant closed sets in noisy n-ary relations has been recently studied [CBNB12]. In a noisy dataset, a rule can have a high confidence while it may cover only a very small subset of cases (i. e., its frequency is low). Instead of finding some kind of exact rules in the dataset, one may look for noise-tolerant rules. For example, we may want to find a rule like "In

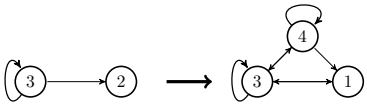


Figure 34: An interesting rule in the dynamic graph from Figure 19.

summer, a customer who buys at least two out of three products: *cherry*, *apple*, *pear* tends to buy *melon*". Such a noise-tolerant rule would cover more cases and its overall quality seems to much more interesting.

Improving the rule pattern domains

In a relation like $Customers \times Products \times Seasons$, we can be interested in a rule like "A customer who buys melon in summer and buys grape in autumn tends to buy chestnut in winter". Considering the semantics of such a rule, we would expect that a customer is not enforced to buy all three products (melon, grape and chestnut) together in all 3 seasons (summer, autumn and winter). The support of this rule could be the intersection of three sets: the set of customers buying melon in summer, the set of customers buying grape in autumn, and the set of customers buying chestnut in winter. This is different from the support of a multidimensional association/disjunctive rule which is based on the support of the union of its body and each association in its head. Therefore, the above rule cannot be found by means of available multidimensional association/disjunctive rule mining tools. Notice that such a rule is not an implication between associations that co-occur: it cannot be derived from a n-set.

Discovering patterns in dynamic graphs

Considering our case study about multidimensional association/disjunctive rule mining in dynamic graphs, we can discuss some of its limitations. For instance, frequent subgraphs whose structures are arbitrary cannot be mined. Indeed, in a subgraph of such a rule, each departure vertex must be connected to all arrival vertices, and each arrival vertex must be connected to all departure vertices. For example, it is not yet possible to discover the rule in Figure 34 because, in its head, Vertex 1 is a departure vertex and Vertex 4 is an arrival vertex but there is no edge from Vertex 1 towards Vertex 4. In addition, as mentioned in Chapter 5, our approach handles only dynamic graphs which can be encoded naturally by means of n-ary relations. So far, our encoded graphs can include properties on their edges but not on their vertices.

Therefore, our approach could be extended to mine arbitrary patterns about the graph evolution and to exploit properties on both edges and vertices.

Bibliography

[1] Ramesh C. Agarwal, Charu C. Aggarwal, and V. V. V. Prasad. Depth first generation of long patterns. In *Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 108–118. ACM, 2000.

[2] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, SIGMOD '93, pages 207–216. ACM, 1993.

[3] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI/MIT Press, 1996.

[4] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499. Morgan Kaufmann, 1994.

[5] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. RTM: Laws and a recursive generator for weighted time-evolving graphs. In *Proceedings of the 8th IEEE International Conference on Data Mining*, ICDM '08, pages 701–706. IEEE Computer Society, 2008.

[6] Maria-Luiza Antonie and Osmar R. Zaïane. Mining positive and negative association rules: an approach for confined rules. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, ECML PKDD '04, pages 27–38. Springer-Verlag New York, Inc., 2004.

[7] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In

Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '06, pages 44–54. ACM, 2006.

[8] José L. Balcázar. Redundancy, deduction schemes, and minimum-size bases for association rules. *Logical Methods in Computer Science*, 6(2), 2010.

[9] Elena Baralis and Giuseppe Psaila. Designing templates for mining association rules. *J. Intell. Inf. Syst.*, 9(1):7–32, 1997.

[10] Yves Bastide, Rafik Taouil, Nicolas Pasquier, Gerd Stumme, and Lotfi Lakhal. Mining frequent patterns with counting inference. *SIGKDD Explor. Newsl.*, 2:66–75, 2000.

[11] Stephen D. Bay and Michael J. Pazzani. Detecting change in categorical data: mining contrast sets. In *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, pages 302–306. ACM, 1999.

[12] M. Berlingerio, F. Bonchi, B. Bringmann, and A. Gionis. Mining graph evolution rules. In *ECML/PKDD*, volume 5781, pages 115–130. Springer, 2009.

[13] Michele Berlingerio, Michele Coscia, Fosca Giannotti, Anna Monreale, and Dino Pedreschi. As time goes by: discovering eras in evolving social networks. In *Proceedings of the 14th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '10, pages 81–90. Springer-Verlag, 2010.

[14] Jérémy Besson. *Découvertes de motifs pertinents pour l'analyse du transcriptome: application à l'insulino-résistance*. PhD thesis, INSA Lyon, November 2005.

[15] Jérémy Besson, Céline Robardet, Jean-François Boulicaut, and Sophie Rome. Constraint-based formal concept mining and its application to microarray data analysis. *Intelligent Data Analysis*, 9(1):59–82, 2005.

[16] Francesco Bonchi and Claudio Lucchese. Pushing tougher constraints in frequent pattern mining. In *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, PAKDD '05, pages 114–124. Springer-Verlag, 2005.

[17] Francesco Bonchi and Claudio Lucchese. On condensed representations of constrained frequent patterns. *Knowl. Inf. Syst.*, 9(2):180–201, 2006.

[18] Francesco Bonchi and Claudio Lucchese. Extending the state-of-the-art of constraint-based pattern discovery. *Data Knowl. Eng.*, 60(2):377–399, 2007.

[19] Karsten M. Borgwardt, Hans-Peter Kriegel, and Peter Wackersreuther. Pattern mining in frequent dynamic subgraphs. In *Proceedings of the 6th International Conference on Data Mining*, ICDM '06, pages 818–822. IEEE Computer Society, 2006.

[20] J-F. Boulicaut, Luc De Raedt, and Heikki Mannila, editors. *Constraint-Based Mining and Inductive Databases*, volume 3848 of *LNCS*. Springer, 2005.

[21] Jean-François Boulicaut. Inductive databases and multiple uses of frequent itemsets: The cInQ approach. In *Database Support for Data Mining Applications*, volume 2682 of *LNCS*, pages 1–23. Springer, 2004.

[22] Jean-François Boulicaut, Artur Bykowski, and Christophe Rigotti. Free-sets: A condensed representation of boolean data for the approximation of frequency queries. *Data Min. Knowl. Discov.*, 7(1):5–22, 2003.

[23] Jean-Francois Boulicaut, Artur Bykowski, and Christophe Rigotti. Approximation of frequency queris by means of free-sets. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '00, pages 75–85. Springer-Verlag, 2000.

[24] Jean-Francois Boulicaut and Baptiste Jeudy. Mining free itemsets under constraints. In *Proceedings of the International Database Engineering & Applications Symposium*, IDEAS '01, pages 322–329. IEEE Computer Society, 2001.

[25] Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur. Dynamic itemset counting and implication rules for market basket data. *SIGMOD Rec.*, 26:255–264, June 1997.

[26] Björn Bringmann and Siegfried Nijssen. What is frequent in a single graph? In *Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining*, PAKDD '08, pages 858–863. Springer-Verlag, 2008.

[27] Doug Burdick, Manuel Calimlim, and Johannes Gehrke. MAFIA: A maximal frequent itemset algorithm for transactional databases. In *Proceedings of the 17th International Conference on Data Engineering*, ICDE '01, pages 443–452. IEEE Computer Society, 2001.

[28] Artur Bykowski. *Condensed Representations of Frequent Itemsets: Application to Descriptive Pattern Discovery.* PhD thesis, INSA Lyon, November 2002.

[29] Toon Calders, Jan Ramon, and Dries Van Dyck. Anti-monotonic overlap-graph support measures. In *Proceedings of the 8th IEEE International Conference on Data Mining*, ICDM '08, pages 73–82. IEEE Computer Society, 2008.

[30] Toon Calders, Christophe Rigotti, and Jean françois Boulicaut. A survey on condensed representations for frequent sets. In *Constraint Based Mining and Inductive Databases, Springer-Verlag, LNAI*, pages 64–80. Springer, 2005.

[31] Loïc Cerf. *Constraint-based mining of closed patterns in noisy n-ary relations.* PhD thesis, L'Institut National des Sicenses Appliquées de Lyon, 2010.

[32] Loïc Cerf, Jérémy Besson, Céline Robardet, and Jean-François Boulicaut. Data-Peeler: Constraint-based closed pattern mining in n-ary relations. In *Proceedings of the Eighth SIAM International Conference on Data Mining*, SDM '08, pages 37–48. SIAM, 2008.

[33] Loïc Cerf, Jérémy Besson, Céline Robardet, and Jean-François Boulicaut. Closed patterns meet n-ary relations. *ACM Trans. Knowl. Discov. Data*, 3(1):1–36, 2009.

[34] Loïc Cerf, Tran Bao Nhan Nguyen, and Jean-François Boulicaut. Mining constrained cross-graph cliques in dynamic networks. In *Inductive Databases and Constraint-based Data Mining*, pages 199–228. Springer, 2010.

[35] Yun Chi, Shenghuo Zhu, Xiaodan Song, Junichi Tatemura, and Belle L. Tseng. Structural and temporal analysis of the blogosphere through community factorization. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 163–172. ACM, 2007.

[36] Luc Dehaspe and Luc De Raedt. Mining association rules in multiple relations. In *Inductive Logic Programming, 7th International Workshop*, volume 1297 of *LNCS*, pages 125–132. Springer, 1997.

[37] Cheikh Tidiane Dieng, Tao-Yuan Jen, and Dominique Laurent. An efficient computation of frequent queries in a star schema. In *Proceedings of the 21st international conference on Database and expert systems applications: Part II*, DEXA'10, pages 225–239. Springer-Verlag, 2010.

[38] Guozhu Dong, Jiawei Han, Joyce M. W. Lam, Jian Pei, and Ke Wang. Mining multi-dimensional constrained gradients in data cubes. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 321–330. Morgan Kaufmann Publishers Inc., 2001.

[39] S. Dzeroski, B. Goethals, and P. Panov, editors. *Inductive Databases and Queries: Constraint-Based Data Mining*. Springer, 2010.

[40] Yongjian Fu and Jiawei Han. Meta-rule-guided mining of association rules in relational databases. In *Proceedings of the 1st of the 1st International Workshop on Integration of Knowledge Discovery with Deductive and Object-Oriented Databases*, pages 39–46, 1995.

[41] Bernhard Ganter, Gerd Stumme, and Rudolf Wille, editors. *Formal Concept Analysis, Foundations and Applications*, volume 3626 of *LNCS*. Springer, 2005.

[42] Gemma C. Garriga, Petra Kralj, and Nada Lavrac. Closed sets for labeled data. *Journal of Machine Learning Research*, 9:559–580, 2008.

[43] Dominique Gay, Nazha Selmaoui-Folcher, and Jean-François Boulicaut. Application-independent feature construction based on almost-closedness properties. *Knowl. Inf. Syst.*, 30(1):87–111, 2012.

[44] Liqiang Geng and Howard J. Hamilton. Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38(3), 2006.

[45] Bart Goethals, Dominique Laurent, and Wim Le Page. Discovery and application of functional dependencies in conjunctive query mining. In *Proceedings of the 12th international conference on Data warehousing and knowledge discovery*, DaWaK'10, pages 142–156. Springer-Verlag, 2010.

[46] Bart Goethals, Wim Le Page, and Heikki Mannila. Mining association rules of simple conjunctive queries. In *Proceedings of the 8th SIAM International Conference on Data Mining*, SDM '08, pages 96–107. SIAM, 2008.

[47] Karam Gouda and Mohammed J. Zaki. Genmax: An efficient algorithm for mining maximal frequent itemsets. *Data Min. Knowl. Discov.*, 11:223–242, 2005.

[48] Gösta Grahne and Jianfei Zhu. Efficiently using prefix-trees in mining frequent itemsets. In *Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations*, volume 90 of *FIMI '03*. CEUR-WS.org, 2003.

[49] Jiawei Han and Yongjian Fu. Discovery of multiple-level association rules from large databases. In *Proceedings of the 21th International Conference on Very Large Data Bases*, VLDB '95, pages 420–431. Morgan Kaufmann Publishers Inc., 1995.

[50] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, 2004.

[51] Céline Hébert and Bruno Crémilleux. Mining frequent *delta*-free patterns in large databases. In *Discovery Science*, volume 3735 of *Lecture Notes in Computer Science*, pages 124–136. Springer, 2005.

[52] Inès Hilali-Jaghdam, Tao-Yuan Jen, Dominique Laurent, and Sadok Ben Yahia. Mining frequent disjunctive selection queries. In *Proceedings of the 22nd international conference on Database and expert systems applications - Volume Part II*, DEXA '11, pages 90–96. Springer-Verlag, 2011.

[53] Tomasz Imieliński, Leonid Khachiyan, and Amin Abdulghani. Cubegrades: Generalizing association rules. *Data Min. Knowl. Discov.*, 6(3):219–257, 2002.

[54] Akihiro Inokuchi and Takashi Washio. GTRACE2: Improving performance using labeled union graphs. In *Proceedings of the 14th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '10, pages 178–188. Springer-Verlag, 2010.

[55] Akihiro Inokuchi and Takashi Washio. Mining frequent graph sequence patterns induced by vertices. In *Proceedings of the 10th SIAM International Conference on Data Mining*, SDM '10, pages 466–477. SIAM, 2010.

[56] Robert Jaschke, Andreas Hotho, Christoph Schmitz, Bernhard Ganter, and Gerd Stumme. TRIAS–an algorithm for mining iceberg tri-lattices. In *Proceedings of the 6th International Conference on Data Mining*, ICDM '06, pages 907–911. IEEE Computer Society, 2006.

[57] Tao-Yuan Jen, Dominique Laurent, and Nicolas Spyratos. Mining all frequent projection-selection queries from a relational table. In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*, EDBT '08, pages 368–379. ACM, 2008.

[58] Tao-Yuan Jen, Dominique Laurent, and Nicolas Spyratos. Mining frequent conjunctive queries in star schemas. In *Proceedings of the 2009 International Database Engineering & Applications Symposium*, IDEAS '09, pages 97–108. ACM, 2009.

[59] Tao-Yuan Jen, Rafik Taouil, and Dominique Laurent. A dichotomous algorithm for association rule mining. In *Proceedings of the 15th International Workshop on Database and Expert Systems Applications*, DEXA '04, pages 567–571. IEEE Computer Society, 2004.

[60] Baptiste Jeudy. *Optimisation de requêtes inductives: Application á l'extraction sous contraintes de règles d'association.* PhD thesis, INSA Lyon, December 2002.

[61] Liping Ji, Kian-Lee Tan, and Anthony K. H. Tung. Mining frequent closed cubes in 3d datasets. In *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pages 811–822. VLDB Endowment, 2006.

[62] Micheline Kamber, Jiawei Han, and Jenny Y. Chiang. Metarule-guided mining of multi-dimensional association rules using data cubes. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, KDD '97, pages 207–210. AAAI Press, 1997.

[63] Marzena Kryszkiewicz. Concise representations of association rules. In *Pattern Detection and Discovery*, pages 92–109. Springer, 2002.

[64] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. Structure and evolution of online social networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 611–617. ACM, 2006.

[65] Mayank Lahiri and Tanya Y. Berger-Wolf. Mining periodic behavior in dynamic social networks. In *Proceedings of the 8th IEEE International Conference on Data Mining*, ICDM '08, pages 373–382. IEEE Computer Society, 2008.

[66] Philippe Lenca, Patrick Meyer, Benoît Vaillant, and Stéphane Lallich. On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. *European Journal of Operational Research*, 184(2):610–626, 2008.

[67] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 177–187. ACM, 2005.

[68] Dao-I Lin and Zvi M. Kedem. Pincer search: A new algorithm for discovering the maximum frequent set. In *Proceedings of the 6th International Conference on Extending Database Technology*, EDBT '98, pages 105–119, 1998.

[69] Charles X. Ling, Tielin Chen, Qiang Yang, and Jie Cheng. Mining optimal actions for profitable crm. In *Proceedings of the 2nd IEEE International Conference on Data Mining*, ICDM '02, pages 767–770. IEEE Computer Society, 2002.

[70] Claudio Lucchese, Salvatore Orlando, and Raffaele Perego. Fast and memory efficient mining of frequent closed itemsets. *IEEE Trans. Knowl. Data Eng.*, 18(1):21–36, 2006.

[71] Michael Luxenburger. Implications partielles dans un contexte. *Mathématiques et Sciences Humaines*, 29(113):35–55, 1991.

[72] Heikki Mannila and Hannu Toivonen. Multiple uses of frequent sets and condensed representations (extended abstract). In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, KDD '96, pages 189–194, 1996.

[73] Heikki Mannila and Hannu Toivonen. Levelwise search and borders of theories in knowledgediscovery. *Data Min. Knowl. Discov.*, 1(3):241–258, 1997.

[74] R. Meo, P. L. Lanzi, and M. Klemettinen, editors. *Database Support for Data Mining Applications: Discovering Knowledge with Inductive Queries*, volume 2682 of *LNCS*. Springer, 2004.

[75] Riadh Ben Messaoud, Sabine Loudcher Rabaséda, Omar Boussaid, and Rokia Missaoui. Enhanced mining of association rules from data cubes. In *Proceedings of the 9th ACM international workshop on Data warehousing and OLAP*, DOLAP '06, pages 11–18. ACM, 2006.

[76] Rokia Missaoui and Léonard Kwuida. Mining triadic association rules from ternary relations. In *Proceedings of the 9th international conference on Formal concept analysis*, ICFCA '11, pages 204–218. Springer-Verlag, 2011.

[77] Amit A. Nanavati, Krishna P. Chitrapura, Sachindra Joshi, and Raghu Krishnapuram. Mining generalised disjunctive association rules. In *Proceedings of the 10th international conference on Information and knowledge management*, CIKM '01, pages 482–489. ACM, 2001.

[78] Raymond T. Ng, Laks V. S. Lakshmanan, Jiawei Han, and Alex Pang. Exploratory mining and pruning optimizations of constrained associations rules. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, SIGMOD '98, pages 13–24. ACM, 1998.

[79] Salvatore Orlando, Paolo Palmerini, Raffaele Perego, and Fabrizio Silvestri. Adaptive and resource-aware mining of frequent sets. In *Proceedings of the 2nd IEEE International Conference on Data Mining*, ICDM '02, pages 338–345. IEEE Computer Society, 2002.

[80] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th Inter-*

national Conference on Database Theory, ICDT '99, pages 398–416. Springer-Verlag, 1999.

[81] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Efficient mining of association rules using closed itemset lattices. *Inf. Syst.*, 24(1):25–46, 1999.

[82] Nicolas Pasquier, Rafik Taouil, Yves Bastide, Gerd Stumme, and Lotfi Lakhal. Generating a condensed representation for association rules. *J. Intell. Inf. Syst.*, 24(1):29–60, 2005.

[83] Jian Pei and Jiawei Han. Can we push more constraints into frequent pattern mining? In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 350–354. ACM, 2000.

[84] Jian Pei, Jiawei Han, and Runying Mao. Closet: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[85] Ruggero G. Pensa. *Un cadre générique pour la co-classification sous contraintes : application à l'analyse du transcriptome.* PhD thesis, INSA Lyon, November 2006.

[86] Luc De Raedt, Manfred Jaeger, Sau Dan Lee, and Heikki Mannila. A theory of inductive query answering. In P. Panov S. Dzeroski, B. Goethals, editor, *Inductive Databases and Cosntraint-Based Data Mining*, pages 79–103. Springer, 2010.

[87] François Rioult, Jean-François Boulicaut, Bruno Crémilleux, and Jérémy Besson. Using transposition for pattern discovery from microarray data. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, DMKD '03, pages 73–79. ACM, 2003.

[88] Céline Robardet. Constraint-based pattern mining in dynamic graphs. In *Proceedings of the 9th IEEE International Conference on Data Mining*, ICDM '09, pages 950–955. IEEE Computer Society, 2009.

[89] J Roberto and Jr Bayardo. Efficiently mining long patterns from databases. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, SIGMOD '98, pages 85–93. ACM, 1998.

[90] Sigal Sahar. Interestingness via what is not interesting. In *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, pages 332–336. ACM, 1999.

[91] Ashoka Savasere, Edward Omiecinski, and Shamkant B. Navathe. An efficient algorithm for mining association rules in large databases. In *Proceedings of the 21th International Conference on Very Large Data Bases*, VLDB '95, pages 432–444. Morgan Kaufmann Publishers Inc., 1995.

[92] C. Schmitz, A. Hotho, R. Jäschke, and G. Stumme. Mining association rules in folksonomies. In *Data Science and Classification*, pages 261–270. Springer, 2006.

[93] Avi Silberschatz and Alexander Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*, KDD '95, pages 275–281. AAAI Press, 1995.

[94] Avi Silberschatz and Alexander Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Trans. on Knowl. and Data Eng.*, 8(6):970–974, 1996.

[95] Arnaud Soulet. *Un cadre générique de découverte de motifs sous contraintes fondé sur des primitives*. PhD thesis, Université de Caen, July 2006.

[96] Arnaud Soulet and Bruno Crémilleux. An efficient framework for mining flexible constraints. In *Proceedings of the 9th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'05, pages 661–671. Springer-Verlag, 2005.

[97] Ramakrishnan Srikant and Rakesh Agrawal. Mining generalized association rules. In *Proceedings of the 21th International Conference on Very Large Data Bases*, VLDB '95, pages 407–419. Morgan Kaufmann Publishers Inc., 1995.

[98] Gerd Stumme, Rafik Taouil, Yves Bastide, Nicolas Pasquier, and Lotfi Lakhal. Intelligent structuring and reducing of association rules with formal concept analysis. In *Proceedings of the Joint German/Austrian Conference on AI: Advances in Artificial Intelligence*, KI '01, pages 335–350. Springer-Verlag, 2001.

[99] Gerd Stumme, Rafik Taouil, Yves Bastide, Nicolas Pasquier, and Lotfi Lakhal. Computing iceberg concept lattices with TITANIC. *Data Knowl. Eng.*, 42(2):189–222, 2002.

[100] Jimeng Sun, Spiros Papadimitriou, Philip S. Yu, and Christos Faloutsos. Graphscope: parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 687–696. ACM, 2007.

[101] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 32–41. ACM, 2002.

[102] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[103] Syed Khairuzzaman Tanbeer, Chowdhury Farhan Ahmed, Byeong-Soo Jeong, and Young-Koo Lee. Efficient single-pass frequent pattern mining using a prefix-tree. *Inf. Sci.*, 179:559–583, February 2009.

[104] Haorianto Cokrowijoyo Tjioe and David Taniar. Mining association rules in data warehouses. *International Journal of Data Warehousing and Mining*, 1(3):28–62, 2005.

[105] Hannu Toivonen. Sampling large databases for association rules. In *Proceedings of the 22th International Conference on Very Large Data Bases*, VLDB '96, pages 134–145. Morgan Kaufmann Publishers Inc., 1996.

[106] Hanghang Tong, Spiros Papadimitriou, Jimeng Sun, Philip S. Yu, and Christos Faloutsos. Colibri: fast mining of large static and dynamic graphs. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 686–694. ACM, 2008.

[107] Benoît Vaillant, Philippe Lenca, and Stéphane Lallich. A clustering of interestingness measures. In *Discovery Science*, pages 290–297. Springer, 2004.

[108] Jianyong Wang, Jiawei Han, and Jian Pei. Closet+: searching for the best strategies for mining frequent closed itemsets. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 236–245. ACM, 2003.

[109] Rudolf Wille. Concept lattices and conceptual knowledge systems. *Computers and Mathematics with Applications*, 23:493–515, 1992.

[110] Fan Wu, Shih-Wen Chiang, and Jiunn-Rong Lin. A new approach to mine frequent patterns using item-transformation methods. *Inf. Syst.*, 32:1056–1072, November 2007.

[111] Chang Hun You, Lawrence B. Holder, and Diane J. Cook. Learning patterns in the dynamics of biological networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 977–986. ACM, 2009.

[112] Mohammed J. Zaki. Scalable algorithms for association mining. *IEEE Trans. on Knowl. and Data Eng.*, 12:372–390, May 2000.

[113] Mohammed J. Zaki. Mining non-redundant association rules. *Data Min. Knowl. Discov.*, 9(3):223–248, 2004.

[114] Mohammed J. Zaki and Karam Gouda. Fast vertical mining using diffsets. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 326–335. ACM, 2003.

[115] Mohammed J. Zaki and Ching jui Hsiao. Charm: An efficient algorithm for closed itemset mining. In *Proceedings of the 2nd SIAM International Conference on Data Mining*, SDM '02, pages 457–473, 2002.

Appendix A

Proofs

A.1 Proof of Theorem 5

Proof. According to the definitions 38 and 34:

- $X \sqsubseteq Y \Rightarrow \begin{cases} \mathcal{D}_X \subseteq \mathcal{D}_Y \\ \forall D^i \in \mathcal{D},\ \pi_{D^i}(X) \subseteq \pi_{D^i}(Y) \end{cases}$;
- $s(Y) = \{\mathrm{u} \in \times_{D^i \in \mathcal{D} \setminus \mathcal{D}_Y} D^i \mid \forall y \in Y,\ y \cdot \mathrm{u} \in \mathcal{R}\}$;
- $s(X) = \{w \in \times_{D^i \in \mathcal{D} \setminus \mathcal{D}_X} D^i \mid \forall x \in X,\ x \cdot w \in \mathcal{R}\}$
  $= \{\mathrm{v} \cdot \mathrm{u} \mid \mathrm{v} \in \times_{D^i \in \mathcal{D}_Y \setminus \mathcal{D}_X} D^i, \mathrm{u} \in \times_{D^i \in \mathcal{D} \setminus \mathcal{D}_Y} D^i$
  and $\forall x \in X,\ x \cdot \mathrm{v} \cdot \mathrm{u} \in \mathcal{R}\}$.

Let $\pi_{\mathcal{D} \setminus \mathcal{D}_Y} s(X) = \{\mathrm{u} \in \times_{D^i \in \mathcal{D} \setminus \mathcal{D}_Y} D^i \mid \exists \mathrm{v} \in \times_{D^i \in \mathcal{D}_Y \setminus \mathcal{D}_X} D^i$ such that $\forall x \in X$, $x \cdot \mathrm{v} \cdot \mathrm{u} \in \mathcal{R}\}$.

Then, $\begin{cases} s(Y) \subseteq \pi_{\mathcal{D} \setminus \mathcal{D}_Y} s(X) \\ |\pi_{\mathcal{D} \setminus \mathcal{D}_Y} s(X)| \leq |s(X)| \end{cases}$ and $|s(Y)| \leq |\pi_{\mathcal{D} \setminus \mathcal{D}_Y} s(X)| \leq |s(X)|$. □

A.2 Proof of Theorem 6

Proof. Using Def. 42, we have $X' \sqsubseteq X \Rightarrow s_{\mathcal{D} \setminus \mathcal{D}'}(X) \subseteq s_{\mathcal{D} \setminus \mathcal{D}'}(X')$.

Because $X \sqsubseteq X' \sqsubseteq Y$ and according to Def. 43:

$$\begin{cases} c_{\text{natural}}(X \to Y \setminus X) = \frac{|s(Y)|}{|s_{\mathcal{D} \setminus \mathcal{D}'}(X)|} \\ c_{\text{natural}}(X' \to Y \setminus X') = \frac{|s(Y)|}{|s_{\mathcal{D} \setminus \mathcal{D}'}(X')|} \end{cases}$$

$\Rightarrow c_{\text{natural}}(X' \to Y \setminus X') \leq c_{\text{natural}}(X \to Y \setminus X)$. □

A.3 Proof of Theorem 7

Proof. The first, we proof that if $X \to Y$ is a non-redundant rule then $(X \sqcup Y \sqcup s(X \sqcup Y))$ is a closed n-set and for all $X' \sqsubset X$, $(c_{\text{exclusive}}(X' \to (Y \sqcup X) \setminus X') <$

$c_{\text{exclusive}}(X \to Y)) \vee (c_{\text{exclusive}}(X' \to (Y \sqcup X) \setminus X') < c_{\text{exclusive}}(X \to Y))$.

– Assume that $(X \sqcup Y \sqcup s(X \sqcup Y))$ is not closed, by $s(X \sqcup Y)$ is closed with $(X \sqcup Y)$, so it exists an element $e \in \cup_{D^i \in \mathcal{D}'}(D^i \setminus \pi_{D^i}(X \sqcup Y))$ such that $s(X \sqcup Y \sqcup \{e\}) = s(X \sqcup Y)$. Therefore, it exists a rule $X \to Y \sqcup \{e\}$ such that:

$$\begin{cases} f(X \to Y \sqcup \{e\}) = f(X \to Y) \\ c_{\text{exclusive}}(X \to Y \sqcup \{e\}) \geq c_{\text{exclusive}}(X \to Y) \\ c_{\text{natural}}(X \to Y \sqcup \{e\}) = c_{\text{natural}}(X \to Y) \end{cases} .$$

This is contrary to the assumption that $X \to Y$ is a non-redundant rule. Therefore $(X \sqcup Y \sqcup s(X \sqcup Y))$ is a closed n-set.

– By $X \to Y$ is a non-redundant rule, according to Definition 46, so it does not exist $X' \sqsubset X$ such that

$$\begin{cases} f(X' \to (Y \sqcup X) \setminus X') \geq f(X \to Y) \\ c_{\text{exclusive}}(X' \to (Y \sqcup X) \setminus X') \geq c_{\text{exclusive}}(X \to Y) \\ c_{\text{natural}}(X' \to (Y \sqcup X) \setminus X') \geq c_{\text{natural}}(X \to Y) \end{cases} .$$

This corresponds to for all $X' \sqsubset X$, one of three assertions above must be broken. According to Definition 40, $f(X' \to (Y \sqcup X) \setminus X') = f(X \to Y)$, i. e., the first assertion is never broken. As a consequence, the second assertion or third assertion is broken. It means that for all $X' \sqsubset X$, $(c_{\text{exclusive}}(X' \to (Y \sqcup X) \setminus X') < c_{\text{exclusive}}(X \to Y)) \vee (c_{\text{exclusive}}(X' \to (Y \sqcup X) \setminus X') < c_{\text{exclusive}}(X \to Y))$.

Now, we proof that if $(X \sqcup Y \sqcup s(X \sqcup Y))$ is a closed n-set on $\mathcal{R}_A$ and for all $X' \sqsubset X$, $(c_{\text{exclusive}}(X' \to (Y \sqcup X) \setminus X') < c_{\text{exclusive}}(X \to Y)) \vee (c_{\text{natural}}(X' \to (Y \sqcup X) \setminus X') < c_{\text{natural}}(X \to Y))$ then $X \to Y$ is a non-redundant rule. Assume that $X \to Y$ is a redundant rule, one of the two following cases will occur.

– Or it exists a rule $X' \to Y'$ such that:

$$\begin{cases} (X' \sqcup Y' = X \sqcup Y \wedge X' \sqsubset X) \\ f(X' \to Y') \geq f(X \to Y) \\ c_{\text{natural}}(X' \to Y') \geq c_{\text{natural}}(X \to Y) \\ c_{\text{exclusive}}(X' \to Y') \geq c_{\text{exclusive}}(X \to Y) \end{cases} .$$

This is contrary to the assumption that for all $X' \sqsubset X$, $(c_{\text{exclusive}}(X' \to (Y \sqcup X) \setminus X') < c_{\text{exclusive}}(X \to Y)) \vee (c_{\text{natural}}(X' \to (Y \sqcup X) \setminus X') < c_{\text{natural}}(X \to Y))$.

– Or it exists a rule $X' \to Y'$ such that:

$$\begin{cases} (X' \sqcup Y' \sqsupset X \sqcup Y \wedge X' \sqsubseteq X) \\ f(X' \to Y') \geq f(X \to Y) \\ c_{\text{natural}}(X' \to Y') \geq c_{\text{natural}}(X \to Y) \\ c_{\text{exclusive}}(X' \to Y') \geq c_{\text{exclusive}}(X \to Y) \end{cases} .$$

In this case, because $X' \sqcup Y' \sqsupset X \sqcup Y$ and $f(X' \to Y') \geq f(X \to Y)$, according to Theorem 5 and Definition 40, we have $s(X' \sqcup Y') = s(X \sqcup Y)$. This indicates that $(X \sqcup Y \sqcup s(X \sqcup Y)) \subset (X' \sqcup Y' \sqcup s(X' \sqcup Y')) \subset \mathcal{R}_A$. So $(X \sqcup Y \sqcup s(X \sqcup Y))$ is not closed.

Consequently, $X \to Y$ is a non-redundant rule. □

A.4 Proof of Theorem 8

Proof. The first, we proof that $X \to Y$ is a key association rule on $\mathcal{D}'$ then $(X \sqcup Y \sqcup s(X \sqcup Y))$ is a closed set in $\mathcal{R}_A$ and $\forall X' \sqsubset X$, $c_a(X' \to (Y \sqcup X) \setminus X') < c_a(X \to Y)$.

– Assume that $(X \sqcup Y \sqcup s(X \sqcup Y))$ is not a closed set,
by $s(X \sqcup Y)$ is closed with $(X \sqcup Y)$, so it exists an element $e \in \cup_{D^i \in \mathcal{D}'}(D^i \setminus \pi_{D^i}(X \sqcup Y))$ such that $s(X \sqcup Y \sqcup \{e\}) = s(X \sqcup Y)$. Therefore, it exists a rule $X \to Y \sqcup \{e\}$ such that:

$$\begin{cases} f_a(X \to Y \sqcup \{e\}) = f_a(X \to Y) \\ c_d(X \to Y \sqcup \{e\}) = c_a(X \to Y) \end{cases}.$$

This is contrary to the assumption that $X \to Y$ is a key association rule. Therefore $(X \sqcup Y \sqcup s(X \sqcup Y))$ is a closed n-set.

– Assume that it exists an association X' such that $X' \sqsubset X$ and $c_a(X' \to (Y \sqcup X) \setminus X') \geq c_a(X \to Y)$, according to Definition 49, we have $f_a(X \to Y) = f_a(X' \to (Y \sqcup X) \setminus X')$. So, it exits a rule $X' \to (Y \sqcup X) \setminus X'$ such that:

$$\begin{cases} X' \sqsubset X \wedge (X' \sqcup ((Y \sqcup X) \setminus X')) = (Y \sqcup X) \\ f_a(X' \to (Y \sqcup X) \setminus X') = f_a(X \to Y) \\ c_d(X' \to (Y \sqcup X) \setminus X') \geq c_a(X \to Y) \end{cases}.$$

This is contrary to the assumption that $X \to Y$ is a key association rule. Therefore $\forall X' \sqsubset X$, $c_a(X' \to (Y \sqcup X) \setminus X') < c_a(X \to Y)$.

Now, we proof that if $(X \sqcup Y \sqcup s(X \sqcup Y))$ is a closed set in $\mathcal{R}_A$ and $\forall X' \sqsubset X$, $c_a(X' \to (Y \sqcup X) \setminus X') < c_a(X \to Y)$ then $X \to Y$ is a key association rule. Assume that $X \to Y$ is a key association rule, one of the two following cases will occur.

– Or it exists a rule $X' \to Y'$ such that:

$$\begin{cases} (X' \sqcup Y' = X \sqcup Y \wedge X' \sqsubset X) \\ f_a(X' \to Y') \geq f_a(X \to Y) \\ c_a(X' \to Y') \geq c_a(X \to Y) \end{cases}.$$

This is contrary to the assumption that for all $X' \sqsubset X$, $c_a(X' \to (Y \sqcup X) \setminus X') < c_a(X \to Y)$.

– Or it exists a rule $X' \rightarrow Y'$ such that:

$$\begin{cases} (X' \sqcup Y' \sqsupset X \sqcup Y \wedge X' \sqsubseteq X) \\ f_a(X' \rightarrow Y') \geq f_a(X \rightarrow Y) \\ c_a(X' \rightarrow Y') \geq c_a(X \rightarrow Y) \end{cases}.$$

In this case, because $X' \sqcup Y' \sqsupset X \sqcup Y$ and $f_a(X' \rightarrow Y') \geq f_a(X \rightarrow Y)$, according to Theorem 5 and Definition 49, we have $s(X' \sqcup Y') = s(X \sqcup Y)$. This indicates that $(X \sqcup Y \sqcup s(X \sqcup Y)) \subset (X' \sqcup Y' \sqcup s(X' \sqcup Y')) \subset \mathcal{R}_A$. So $(X \sqcup Y \sqcup s(X \sqcup Y))$ is not closed.

Consequently, $X \rightarrow Y$ is a key association rule. □

A.5 Proof of Theorem 9

Proof. we first proof that, $\forall \mathcal{D}' \subseteq \mathcal{D}$, let $\mathcal{P}$ the set of all key association rules on $\mathcal{D}'$, if $X \rightarrow \vee\mathcal{Y}$ is a non-redundant multidimensional disjunctive rule on $\mathcal{D}'$ then $\forall Y \in \mathcal{Y}$, $X \rightarrow Y$ is a key association rule on $\mathcal{D}'$ and $\mathcal{Y} = \cup_{X \rightarrow Y \in \mathcal{P}} Y$.

– According to Definition 53, if $X \rightarrow \vee\mathcal{Y}$ is a non-redundant multidimensional disjunctive rule then $\forall Y \in \mathcal{Y}$, $X \rightarrow Y$ is a key association rule on $\mathcal{D}'$.
– Assume $\mathcal{Y} \neq \cup_{X \rightarrow Y \in \mathcal{P}} Y$. It occurs the one of two following cases:
  – Or $\mathcal{Y} \subset \cup_{X \rightarrow Y \in \mathcal{P}} Y$. Therefore, set $\mathcal{Z} = \cup_{X \rightarrow Y \in \mathcal{P}} Y$, according to Definition 51 and Definition 52, it exits another rule $X \rightarrow \vee\mathcal{Z}$ such that

$$\begin{cases} \mathcal{Z} \supset \mathcal{Y} \\ \forall Z \in \mathcal{Z}, X \rightarrow Z \text{ is a key association rule} \\ f_d(X \rightarrow \vee\mathcal{Z}) \geq f_d(X \rightarrow \vee\mathcal{Y}) \\ c_d(X' \rightarrow \vee\mathcal{Z}) \geq c_d(X \rightarrow \vee\mathcal{Y}) \end{cases}.$$

    This is contrary to the assumption that $X \rightarrow \vee\mathcal{Y}$ is a non-redundant multidimensional disjunctive rule.
  – Or $\begin{cases} \mathcal{Y} \supset \cup_{X \rightarrow Y \in \mathcal{P}} Y \\ \forall Y \in \mathcal{Y}, X \rightarrow Y \text{ is a key association rule} \end{cases}$.

    It means that it exits an association Y such that $X \rightarrow Y \notin \mathcal{P}$ and $X \rightarrow Y$ is a key association rule. This is contrary to the assumption that $\mathcal{P}$ is the set of all key association rules on $\mathcal{D}'$.

  Consequently, $\mathcal{Y} = \cup_{X \rightarrow Y \in \mathcal{P}} Y$.

Now, we proof that, $\forall \mathcal{D}' \subseteq \mathcal{D}$, let $\mathcal{P}$ the set of all key association rules on $\mathcal{D}'$, if $\forall Y \in \mathcal{Y}$, $X \rightarrow Y$ is a key association rule on $\mathcal{D}'$ and $\mathcal{Y} = \cup_{X \rightarrow Y \in \mathcal{P}} Y$ then $X \rightarrow \vee\mathcal{Y}$ is a non-redundant multidimensional disjunctive rule on $\mathcal{D}'$. Assume $X \rightarrow \vee\mathcal{Y}$ is not a non-redundant multidimensional disjunctive rule, one of the two following cases will occur:

- Or it exits an association $Y \in \mathcal{Y}$ such that $X \rightarrow Y$ is not a key association rule. This is contrary to the first condition of the assumption.
- Or it exits a set of association $\mathcal{Z} \supset \mathcal{Y}$ such that:

$$\begin{cases} \forall Z \in \mathcal{Z}, X \rightarrow Z \text{ is a key association rule} \\ f_d(X \rightarrow \vee\mathcal{Z}) \geq f_d(X \rightarrow \vee\mathcal{Y}) \\ c_d(X \rightarrow \vee\mathcal{Z}) \geq c_d(X \rightarrow \vee\mathcal{Y}) \end{cases} .$$

  It means that it exits an association $Z \notin \mathcal{Y}$ such that $X \rightarrow Z$ is a key association rule. This is contrary to the second condition of the assumption that $\mathcal{Y} = \cup_{X \rightarrow Y \in \mathcal{P}} Y$.

Consequently, $X \rightarrow \vee\mathcal{Y}$ is a non-redundant multidimensional disjunctive rule. □

Appendix B

Résumé en Français

B.1 Introduction

La fouille de grandes relations binaires a mobilisé énormément de chercheurs et de ressources. Il s'agit, par exemple, d'analyser des relations *Transactions* × *Produits* (on parle aussi de données transactionnelles) ou plus généralement des relations *Objets* × *Propriétés* où les deux dimensions peuvent être de grande taille. De nombreuses propositions permettent aujourd'hui d'alimenter des processus de découverte de connaissances à partir de telles données. Nous nous intéressons aux méthodes descriptives basées sur des calculs de régularités ou de motifs locaux. Il peut s'agir d'ensembles fréquents (voir, e.g., [2, 72]), d'ensembles fermés ou de concepts formels (voir, e.g., [41, 99]), de règles d'association (voir, e.g., [2, 3]) ou encore de leurs généralisations avec, par exemple, l'introduction de négations [72] ou la découverte de règles dans un contexte multi-relationnel [36, 58]. Il existe aujourd'hui un savoir-faire algorithmique pour calculer efficacement de nombreux types de motifs dans des grandes relations binaires. Ceci étant, de nombreux jeux de données se présentent naturellement comme des relations n-aires avec, par exemple, l'ajout de dimensions spatiales et/ou temporelles sur des relations *Transactions* × *Produits* qui peuvent devenir des relations *Transactions* × *Produits* × *Date* × *Lieu de vente* × Temps.

Étendre les méthodes de fouille de relations binaires au contexte des relation d'arité arbitraire paraît donc être une direction de recherche importante et encore peu étudiée. Le problème est que l'extension aux relations n-aires est plus ou moins difficile et que nous devons considérer trois problèmes majeurs dans la fouille de données non supervisée au moyen de motifs (ou des règles descriptives qui peuvent en être dérivées).

1. Quelle est la sémantique du domaine de motif ? Autrement dit, quelles sont les formes qui peuvent prendre les motifs dans des relations n-aires et quels sont les mesures qui vont permettre d'en déterminer l'intérêt a priori ? Si l'on veut spécifier ce qu'est une règle d'association [2] dans ce contexte des relations

n-aires, que deviennent les classiques mesures de fréquence et de confiance ?

2. Quels sont les mécanismes qui vont permettre de spécifier les attentes de l'analyste et donc l'intérêt subjectif ? Depuis quelques années, de nombreux chercheurs développent le cadre de la fouille de données sous contraintes pour lequel des combinaisons Booléennes de contraintes primitives peuvent spécifier déclarativement des propriétés souhaitées sur les motifs solutions (voir, e.g., [20]). Il faudrait donc idéalement identifier les "bonnes" contraintes primitives.
3. Quels sont les moyens de calculs qui vont permettre de calculer les motifs solutions c'est-à-dire satisfaisant les contraintes posées ? Si possible, on souhaite réaliser des calculs corrects et complets qui délivrent tous les motifs solutions et seulement ceux-là. Il faut pouvoir passer à l'échelle au regard du nombre de dimensions et de la taille (nombre de valeurs) de chacune d'entre elles.

Ainsi, étendre la sémantique des motifs ensemblistes comme des concepts formels (couples d'ensembles fermés sur chacune des deux dimensions) au contexte des relations n-aires est trivial d'un point de vue déclaratif (spécification a priori des critères d'intérêts objectifs et subjectifs) mais difficile sur un plan calculatoire [61, 56, 32]. Par contre, et c'est l'objet de cette thèse , définir la sémantique des règles d'association dans des relations n-aires s'est révélé délicat. En fait, depuis la proposition initiale de cette tâche prototypique en fouille de données [2], la sémantique des règles d'association a été assez peu étudiée et formalisée. Bien qu'il s'agisse d'un type de motif simple, on note que des notions importantes pour la sémantique des règles (e.g., les concepts de fréquence ou de contre-exemples) peuvent connaître des définitions différentes selon les auteurs.

Lorsque l'on travaille sur des relations n-aires, il va falloir redéfinir et le langage des motifs et ce que peuvent être de telles mesures lorsque les prémisses et les conclusions des règles peuvent porter sur des sous-ensembles de n'importe lesquelles des dimensions. Ainsi, notre première contribution consiste à concevoir la sémantique des règles via des mesures d'intérêt comme les notions de fréquence et de confiance.

Notre seconde contribution est algorithmique et concerne la conception d'un premier algorithme d'extraction efficace pour calculer les règles a priori intéressantes. Il s'appuie sur les principes qui viennent d'être proposés pour le calcul de motifs multidimensionnels fermés [32, 33]. Nous décrivons ici l'algorithme et nous établissons quelques unes de ses propriétés. Son comportement expérimental est également étudié.

Le résumé de la thèse est organisé comme suit. Dans la section B.2, nous présentons la construction du domaine de motif des règles d'association dans une relation n-aire. Sur cette tâche, nos résultats préliminaires et les premières propositions ont été publiés dans [NCB10, NCPB10] avant la présentation dans l'article de conférence [NCPB11] (Algorithme PINARD [1]) et son amélioration dans l'article de journal (Algorithme PINARD + +) [NCPB11]. Nous proposons ensuite d'introduire des disjonc-

1. PINARD Is N-ary Association Rule Discovery.

	p_1	p_2	p_3	p_4	p_1	p_2	p_3	p_4	p_1	p_2	p_3	p_4	p_1	p_2	p_3	p_4
o_1	1	1			1	1		1		1	1			1	1	1
o_2	1	1		1	1	1					1					
o_3	1	1	1	1						1	1	1		1	1	
o_4	1		1					1							1	1
o_5		1	1	1			1		1	1	1	1	1		1	1
	s_1				s_2				s_3				s_4			

Figure B.1: La relation n-aire $\mathcal{R}_E$

tions dans les conclusions des règles dans la section B.3. Cette étudie et l'algorithme CIDRE [2] ont été introduits dans [NPB12]. Enfin, la section B.4 est dédiée à une application de nos méthodes pour l'analyse des graphes relationnels dynamiques. Les articles de conférence [NCPB10] et [NCPB11] ont déjà abordé l'analyse des graphes, mais cela est détaillée dans l'article de journal [NCPB11] et généralisée avec des disjonctions dans [NPB12].

B.2 Généralisation des règles d'association au cas n-aire

B.2.1 Relation n-aire

Soit n ensembles finis supposés disjoints (sans perte de généralité) $\{D^1, \ldots, D^n\} = \mathcal{D}$. Nous notons $\mathcal{R} \subseteq D^1 \times \cdots \times D^n$ la relation n-aire à partir de laquelle on souhaite découvrir des associations. Considérons un exemple jouet de relation ternaire, $\mathcal{R}_E$, représentée dans Figure B.1. $\mathcal{R}_E$ relie des produits de $D^1 = \{p_1, p_2, p_3, p_4\}$ achetés au cours des saisons de $D^2 = \{s_1, s_2, s_3, s_4\}$ par des clients de $D^3 = \{o_1, o_2, o_3, o_4, o_5\}$. Chaque '1' dans Figure B.1 se trouve à l'intersection de trois éléments $(p_i, s_j, c_k) \in D^1 \times D^2 \times D^3$ formant un triplet présent dans $\mathcal{R}_E$. Ainsi le produit p_1 est acheté à la saison s_1 par le client o_1, mais le client o_4 n'achète pas le produit p_2 en saison s_1.

B.2.2 Définitions préliminaires

Nous généralisons d'abord la notion d'itemsets dans une relation binaire à la notion d'associations dans une relation n-aire, car le nouveau domaine de motifs des règles d'association multidimensionnelles est conduit sur des associations. Nous introduisons ensuite certains opérateurs pour manipuler des associations.

Dans une relation binaire qui décrit la relation entre deux domaines seulement, un itemset est un sous-ensemble d'un domaine, et sa fréquence est calculé sur l'autre

2. CIDRE Is a Disjunctive Rule Extractor.

domaine. Laissez-nous proposer une généralisation au moment de définir les associations dans une relation n-aire. Nous considérons qu'une association peut comporter de quelques sous-ensembles de certains domaines différents et que sa fréquence doit être définie en terme des autre domains, i. e., les domaines que elle ne comporte pas. Par exemple, dans une relation 3-aire $Produits \times Saisons \times Clients$, une association peut être un ensemble de produits, ou un ensemble de saisons, mais elle peut aussi concerner à la fois des produits et des saisons, etc. Dans le contexte de la relation n-aires, "Comment pouvons-nous exprimer de telles associations?", "Comment pouvons-nous préciser l'intérêt subjectif de telles associations?".

Dans une relation n-aire, une association sur $\mathcal{D}' \subseteq \mathcal{D}$ est le produit Cartésien de des sous-ensembles non vides des domaines dans $\mathcal{D}'$. Sans perte de généralité, nous supposons que $\mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\}$.

Définition B.1 (Association)**.** *$\forall \mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\} \subseteq \mathcal{D}$, $X^1 \times \cdots \times X^{|\mathcal{D}'|}$ est une association sur $\mathcal{D}'$ si et seulement si $\forall i = 1..|\mathcal{D}'|, X^i \subseteq D^i \wedge X^i \neq \emptyset$. Par convention, si $\mathcal{D}'$ est vide, la seule* association *sur $\mathcal{D}'$ est l'ensemble vide noté $\emptyset$.*

Exemple B.1. *Dans $\mathcal{R}_E$, représentée Figure B.1, $\{p_1, p_2\} \times \{s_1\}$ et $\{p_1, p_2\} \times \{s_1, s_2\}$ sont deux associations sur $\{D^1, D^2\}$. Par contre, $\{p_1, p_2\}$ est une association sur $\{D^1\}$.*

Le domaine support d'une association sur $\mathcal{D}' \subseteq \mathcal{D}$ est $\times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$. Par exemple, dans $\mathcal{R}_E$, le domaine support d'une association sur $\{D^1, D^2\}$ est D^3. Le support d'une association est un sous-ensemble of le domaine support. La définition suivante utilise l'opérateur de concaténation noté $\cdot$. On a, par exemple, $(p_2, s_1) \cdot (o_1) = (p_2, s_1, o_1)$.

Définition B.2 (Support d'une association)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, soit X une association sur $\mathcal{D}'$, son support noté $s(X)$ est :*

$$s(X) = \{u \in \times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i \mid \forall x \in X,\ x \cdot u \in \mathcal{R}\}.$$

On peut noter qu'une association impliquant tous les n domaines ($\mathcal{D}' = \mathcal{D}$) est soit vraie (tous les n-uplets qu'elle contient appartiennent à $\mathcal{R}$), soit fausse (au moins un des n-uplets qu'elle contient n'appartient *pas* à $\mathcal{R}$). Nous n'avons donc pas de graduation possible de sa qualité. Dans ce cas particulier, en utilisant la convention $\times_{D^i \in \emptyset} D^i = \{\epsilon\}$ (où ϵ est le mot vide), les associations possibles ont bien, respectivement, soit un support d'un élément soit un support vide. Un second cas extrême, et peu intéressant, correspond à $s(\emptyset) = \mathcal{R}$. Le support d'une association généralise celui d'un *itemset* dans une relation binaire (cas où $n = 2$ et $\mathcal{D}' = \{D^1\}$).

Exemple B.2. *Considons des exemples de supports des trois associations dans $\mathcal{R}_E$.*
- $s(\{p_1, p_2\} \times \{s_1\}) = \{o_1, o_2, o_3\}$,
- $s(\{p_1, p_2\} \times \{s_1, s_2\}) = \{o_1, o_2\}$,

– $s(\{p_1, p_2\}) = \{(s_1, o_1), (s_1, o_2), (s_1, o_3), (s_2, o_1), (s_2, o_2), (s_3, o_5)\}$.

Dans la suite, la notion de support d'une association est très utilisée. Donnons quelques définitions complémentaires pour exprimer la sémantique d'une règle d'association dans une relation n-aire.

Définition B.3 (Composante). *$\forall \mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\} \subseteq \mathcal{D}$, soit $X = X^1 \times \cdots \times X^{|\mathcal{D}'|}$ une association sur $\mathcal{D}'$. $\forall D^i \in \mathcal{D}$, la composante de X sur D^i, notée $\pi_{D^i}(X)$, est X^i si $D^i \in \mathcal{D}'$, $\emptyset$ sinon.*

Définition B.4 (Union d'associations). *$\forall \mathcal{D}_X \subseteq \mathcal{D}$ et $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, soit X une association sur $\mathcal{D}_X$ et Y une association sur $\mathcal{D}_Y$, l'union de X et Y notée $X \sqcup Y$ est l'association sur $\mathcal{D}_X \cup \mathcal{D}_Y$ pour laquelle $\forall D^i \in \mathcal{D}, \pi_{D^i}(X \sqcup Y) = \pi_{D^i}(X) \cup \pi_{D^i}(Y)$.*

Définition B.5 (Complément d'associations). *$\forall \mathcal{D}_X \subseteq \mathcal{D}$ et $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, soit X une association sur $\mathcal{D}_X$ et Y une association sur $\mathcal{D}_Y$, le complément de X dans Y noté $Y \setminus X$ est l'association sur $\{D^i \in \mathcal{D}_Y \mid \pi_{D^i}(Y) \not\subseteq \pi_{D^i}(X)\}$ telle que $\forall D^i \in \mathcal{D}, \pi_{D^i}(Y \setminus X) = \pi_{D^i}(Y) \setminus \pi_{D^i}(X)$.*

Définition B.6 (Inclusion d'associations). *$\forall \mathcal{D}_X \subseteq \mathcal{D}$ et $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, soit X une association sur $\mathcal{D}_X$ et Y une association sur $\mathcal{D}_Y$, l'inclusion des associations est notée $X \sqsubseteq Y$. On a $X \sqsubseteq Y \Leftrightarrow \forall D^i \in \mathcal{D}, \pi_{D^i}(X) \subseteq \pi_{D^i}(Y)$.*

L'anti-monotonie du support est préservée dans le cadre plus général des associations et en utilisant la notion d'inclusion que nous venons de définir.

Théorème B.1 (Anti-monotonie du support). *$\forall \mathcal{D}_X \subseteq \mathcal{D}$ et $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, soit X une association sur $\mathcal{D}_X$ et Y une association sur $\mathcal{D}_Y$, on a $X \sqsubseteq Y \Rightarrow |s(X)| \geq |s(Y)|$.*

Preuve dans l'Annexe A.1.

Exemple B.3. *Comme $\{p_1, p_2\} \sqsubseteq \{p_1, p_2\} \times \{s_1\} \sqsubseteq \{p_1, p_2\} \times \{s_1, s_2\}$, on observe bien $|s(\{p_1, p_2\})| \geq |s(\{p_1, p_2\} \times \{s_1\})| \geq |s(\{p_1, p_2\} \times \{s_1, s_2\})|$.*

B.2.3 Règle d'association multidimensionnelle

Définition

Étant donné une relation n-aire $\mathcal{R}$ sur l'ensemble de domaines $\mathcal{D} = \{D^1, ..., D^n\}$, une règle d'association sur $\mathcal{D}' \subseteq \mathcal{D}$ est un couple d'associations sur des ensembles de domaines qui peuvent être différents mais dont l'union doit être $\mathcal{D}'$. Le domaine support de la règle est $\times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$.

Définition B.7 (Règle d'association multi-dimensionnelle). *$\forall \mathcal{D}' \subseteq \mathcal{D}$, une règle d'association multi-dimensionnelle sur $\mathcal{D}'$ est un motifs de la forme $X \rightarrow Y$, où X et Y sont associations sur des sous-ensembles de $\mathcal{D}'$ et $X \sqcup Y$ est une association sur $\mathcal{D}'$. X est appelée la prémisse et Y est appelée la conclusion.*

Exemple B.4. *Dans $\mathcal{R}_E$, $\{p_1, p_2\} \to \{s_1, s_2\}$ et $\{p_3\} \times \{s_3, s_4\} \to \{p_2\}$ sont deux règles sur $\{D^1, D^2\}$. $\{p_1\} \to \{p_2\}$ n'est pas une règle sur $\{D^1, D^2\}$ car aucun élément de D^2 n'apparaît dans la prémisse ou la conclusion de la règle. Par contre, $\{p_1\} \to \{p_2\}$ est bien une règle d'association sur $\{D^1\}$.*

Dans le cas binaire, la sémantique classique d'une règle d'association repose sur les mesures de fréquence et de confiance et l'intérêt a priori d'une règle est spécifié au moyen de seuils : une règle a priori intéressante satisfait une conjonction de contraintes spécifiant que sa fréquence et sa confiance doivent être supérieures à des seuils fournis par les analystes [2]. Une règle est fréquente si elle se vérifie sur un grand nombre d'éléments du domaine support. Plus précisément, l'union de la prémisse et de la conclusion de la règle a pour support un ensemble contenant un nombre suffisant d'éléments. Une règle est valide au sens d'une confiance suffisante si la probabilité conditionnelle d'observer la conclusion lorsque l'on observe la prémisse est suffisamment grande. En fait, dans le contexte des règles multi-dimensionnelles, une définition de la fréquence d'une règle va s'imposer naturellement. Par contre, il va être difficile de définir la confiance d'une règle dans le cas où l'association en conclusion est définie sur un ensemble de domaines qui n'est pas inclus dans celui de la prémisse.

Définition de la fréquence

La fréquence (relative) d'une règle d'association est, dans le domaine support, la proportion d'éléments dans le support de l'union de la prémisse et de la conclusion.

Définition B.8 (Fréquence d'une règle)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, soit $X \to Y$ une règle d'association sur $\mathcal{D}'$. Sa fréquence, notée $f(X \to Y)$, est :*

$$f(X \to Y) = \frac{|s(X \sqcup Y)|}{|\times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i|} \ .$$

Exemple B.5. *Considérons deux règles $r_1 : \{p_1, p_2\} \to \{s_1, s_2\}$ and $r_2 : \{p_3\} \times \{s_3, s_4\} \to \{p_2\}$ dans $\mathcal{R}_E$.*

- $f(r_1) = \frac{|s(\{p_1,p_2\} \sqcup \{s_1,s_2\})|}{|D^3|} = \frac{|s(\{p_1,p_2\} \times \{s_1,s_2\})|}{|D^3|} = \frac{|\{o_1,o_2\}|}{|\{o_1,o_2,o_3,o_4,o_5\}|} = \frac{2}{5}$;
- $f(r_2) = \frac{|s(\{p_3\} \times \{s_3,s_4\} \sqcup \{p_2\})|}{|D^3|} = \frac{|s(\{p_2,p_3\} \times \{s_3,s_4\})|}{|D^3|} = \frac{|\{o_1,o_3\}|}{|\{o_1,o_2,o_3,o_4,o_5\}|} = \frac{2}{5}$.

La fréquence de r_1 est la proportion de clients qui achètent les produits p_1 et p_2 à la fois aux saisons s_1 et s_2. De même, la fréquence de r_2 est la proportion de clients qui achètent les produits p_2 et p_3 à la fois aux saisons s_3 et s_4.

Définition de la confiance

Difficulté à définir la confiance Est-il possible de généraliser facilement le concept de confiance d'une règle d'association dans une relation binaire à notre nouveau

contexte, et ainsi de vouloir attribuer à une règle $X \to Y$ la mesure de confiance $\frac{|s(X \sqcup Y)|}{|s(X)|}$?

Lorsque X et $X \sqcup Y$ sont des associations sur le même ensemble de domaines (leurs domaines support sont donc les mêmes), cette définition est souhaitable. Elle est une proportion d'éléments d'un même domaine support.

Exemple B.6. *Dans $\mathcal{R}_E$, la confiance de la règle $\{p_3\} \times \{s_3, s_4\} \to \{p_2\}$ serait*

$$\frac{|s(\{p_3\} \times \{s_3, s_4\} \sqcup \{p_2\})|}{|s(\{p_3\} \times \{s_3, s_4\})|} = \frac{|s(\{p_2, p_3\} \times \{s_3, s_4\})|}{|s(\{p_3\} \times \{s_3, s_4\})|} = \frac{|\{o_1, o_3\}|}{|\{o_1, o_3, o_5\}|} = \frac{2}{3},$$

ce qui correspond à une proportion de clients. *Cela signifie que, parmi ceux qui achètent le produit p_3 à la fois aux saisons s_3 et s_4, la plupart achète aussi le produit p_2 durant ces saisons.*

Cependant, cette sémantique n'est pas satisfaisante pour une règle où l'association en conclusion est définie sur un ensemble de domaines qui n'est pas inclus dans celui de l'association en prémisse. En effet, $s(X \sqcup Y)$ et $s(X)$ sont alors des ensembles disjoints et mettre leurs cardinaux en rapport n'a aucun sens.

Exemple B.7. *Considérons la règle $\{p_1, p_2\} \to \{s_1, s_2\}$ dans $\mathcal{R}_E$. On a $s(\{p_1, p_2\} \times \{s_1, s_2\}) = \{o_1, o_2\}$, qui est un ensemble de* clients, *et $s(\{p_1, p_2\}) = \{(s_1, o_1), (s_1, o_2), (s_1, o_3), (s_2, o_1), (s_2, o_2), (s_3, o_5)\}$, qui est un ensemble de couples $(saison, client)$.*

En conséquence, il est nécessaire pour definir une nouvelle mesure de confiance qui formerait le sens pour toute règle multidimensionnelle $X \to Y$. Lorsque X et $X \sqcup Y$ sont définies sur le (s) même (s) domaine (s), nous aimerions mesurer la confiance de la règle $X \to Y$ au moyen de $\frac{|s(X \sqcup Y)|}{|s(X)|}$. En particulier, les mesures proposées sont des généralisations de la mesure de confiance introduite dans [2].

Notre première solution est de calculer la confiance de $X \to Y$ sur le domaine support de X. La mesure de confiance proposé est appelé une *confiance exclusive*. L'idée est d'introduire un nouveau facteur qui est multiplié avec $|s(X \sqcup Y)|$ telle que cette multiplication et $|s(X)|$ deviennent comparables. La seconde solution est de calculer la confiance de $X \to Y$ sur le domaine support de $(X \sqcup Y)$. Dans ce cas, la mesure de confiance est appelée une *confiance naturelle*. L'idée ici est d'introduire une nouvelle définition de support de X sur le domaine support de $(X \sqcup Y)$.

Confiance exclusive Calculer la confiance de $X \to Y$ pose donc un problème lorsque X est définie sur un ensemble $\mathcal{D}_X$ inclus *strictement* dans celui $\mathcal{D}'$ de $X \sqcup Y$. L'idée pour résoudre ce problème consiste à multiplier $|s(X \sqcup Y)|$ avec la cardinalité de la projection de Y sur les domaines qui sont absents de $\mathcal{D}_X$.

Notons que $s(X)$ et $s(X \sqcup Y) \times (\times_{D^i \in \mathcal{D}' \setminus \mathcal{D}_X} \pi_{D^i}(Y))$ sont les mêmes domaines. Par conséquent, $|s(X)|$ et $|s(X \sqcup Y) \times (\times_{D^i \in \mathcal{D}' \setminus \mathcal{D}_X} \pi_{D^i}(Y))|$ sont comparables et la confiance exclusive de la règle $X \to Y$ est la proportion de ces deux valeurs.

Lorsque la confiance exclusive de $X \to Y$ est élevé, cela signifie que X préfère être "co-apparu" avec Y plutôt que d'être "co-apparu" avec les autres éléments.

Définition B.9 (Confiance exclusive)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, soit $X \to Y$ une règle d'association sur $\mathcal{D}'$ et notons $\mathcal{D}_X$ l'ensemble de domaines sur lequel X est défini, sa confiance exclusive notée $c_{exclusive}(X \to Y)$ est :*

$$c_{exclusive}(X \to Y) = \frac{|s(X \sqcup Y)| \times |\times_{D^i \in \mathcal{D}' \setminus \mathcal{D}_X} \pi_{D^i}(Y)|}{|s(X)|} \ .$$

Lorsque X est une association sur $\mathcal{D}'$, la confiance exclusive de $X \to Y$ vaut $\frac{|s(X \sqcup Y)|}{|s(X)|}$ sous la convention $\times_{D^i \in \emptyset} \pi_{D^i}(Y) = \{\epsilon\}$. Le facteur correctif, $|\times_{D^i \in \mathcal{D}' \setminus \mathcal{D}_X} \pi_{D^i}(Y)|$, appliqué à $|s(X \sqcup Y)|$ permet de comptabiliser les éléments de $s(X \sqcup Y)$ "de la même façon au numérateur et au dénominateur de la fraction".

Exemple B.8. *Considérons la règle $\{p_1, p_2\} \to \{s_1, s_2\}$ dans $\mathcal{R}_E$ et supposons que l'achat d'un client en une saison s'appelle une transaction. On trouve qu'il n'y a que deux clients $\{o_1, o_2\}$ qui achètent les deux produits p_1 et p_2 à la fois aux saisons s_1 et s_2. Dans ce cas, la somme des transactions pour lesquelles les produits p_1 et p_2 sont achetés par les clients o_1 et o_2 au moment des saisons s_1 et s_2 est $|\{o_1, o_2\}| \times |\{s_1, s_2\}| = 4$. Il y a 6 transactions pour lesquelles les produits p_1 et p_2 sont achetés ensemble en n'importe quelle saison: (s_1, o_1), (s_1, o_2), (s_1, o_3), (s_2, o_1), (s_2, o_2), (s_3, o_5). La confiance exclusive de la règle vaut donc :*

$$c_{exclusive}(\{p_1, p_2\} \to \{s_1, s_2\}) = \frac{|s(\{p_1, p_2\} \times \{s_1, s_2\})| \times |\{s_1, s_2\}|}{|s(\{p_1, p_2\})|} = \frac{4}{6} \ .$$

Le fait que le client o_3 achète les deux produits p_1 et p_2 à la saison s_1 mais qu'il ne les achète pas ensemble à la saison s_2 fait aussi "baisser" la confiance en ce que les clients aiment bien acheter ces produits à la fois aux saisons s_1 et s_2. Le fait que le client o_5 achète ces produits en saison s_3 fait "baisser" la confiance sur le fait que l'on n'aime les acheter qu'aux saisons s_1 et s_2. Si cette confiance valait 1 et donc la valeur maximale, cela voudrait dire que les clients appréciant les deux produits p_1 et p_2 achètent ces produits aux saisons s_1 et s_2 mais aussi qu'ils ne les achètent pas pendant les autres saisons. C'est pourquoi nous parlons de *confiance exclusive*.

La confiance exclusive favorise la découverte d'une règle d'association concluant sur un maximum d'éléments. Toutefois, cette exclusivité présente un défaut dommageable à une extraction efficace des règles d'association valides, c'est-à-dire présentant une confiance supérieure à un seuil fixé par l'analyste : $X \mapsto c_{\text{exclusive}}(X \to Y \setminus X)$ avec $X \sqsubseteq Y$ n'est pas une fonction croissante ordonné par $\sqsubseteq$.

Exemple B.9. *Considérons les règles $\{s_1, s_3\} \to \{p_2, p_3, p_4\}$ et $\{p_2\} \times \{s_1, s_3\} \to \{p_3, p_4\}$ dans $\mathcal{R}_E$, $c_{exclusive}(\{s_1, s_3\} \to \{p_2, p_3, p_4\}) = \frac{6}{7}$ et $c_{exclusive}(\{p_2\} \times \{s_1, s_3\} \to \{p_3, p_4\}) = \frac{2}{3}$. Nous observons que $\{s_1, s_3\} \sqsubseteq \{p_2\} \times \{s_1, s_3\} \sqsubseteq \{p_2, p_3, p_4\} \times \{s_1, s_3\}$. Cependant $c_{exclusive}(\{s_1, s_3\} \to \{p_2, p_3, p_4\})$ est plus grande que $c_{exclusive}(\{p_2\} \times \{s_1, s_3\} \to \{p_3, p_4\})$.*

Cela empêche un calcul efficace de toutes les règles dont la exclusive confiance est supérieure à un seuil défini par l'utilisateur. Nous considérons maintenant une autre définition de la mesure de confiance.

Confiance naturelle Rappelons que la définition de la confiance de $X \to Y$ est problématique lorsque le domaine support de X, $\times_{D^i \in \mathcal{D} \setminus \mathcal{D}_X} D^i$, est différent du domaine support de $(X \sqcup Y)$ qui est $\times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$. La confiance dite naturelle repose sur l'idée de ramener le support de X à un sous-ensemble du domaine support de $(X \sqcup Y)$. La confiance de $X \to Y$ est alors une proportion d'éléments de $\times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$ et se voit qualifiée de *naturelle*. Le prix à payer est la nécessité d'une nouvelle définition du support spécifique aux prémisses des règles et dépendant aussi de leurs conclusions.

Définition B.10 (Support naturel d'une prémisse)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, soit $X \to Y$ une règle d'association sur $\mathcal{D}'$, le support naturel de X noté $s_{\mathcal{D} \setminus \mathcal{D}'}(X)$ est :*

$$s_{\mathcal{D} \setminus \mathcal{D}'}(X) = \{u \in \times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i \mid \exists w \in \times_{D^i \in \mathcal{D}' \setminus \mathcal{D}_X} D^i \text{ tel que } \forall x \in X,\ x \cdot w \cdot u \in \mathcal{R}\} \ ,$$

où $\mathcal{D}_X$ est l'ensemble de domaines de définition de X et $x \cdot w \cdot u$ est la concaténation de x, w et u (quitte à changer l'indexation des domaines de sorte que ceux dans $\mathcal{D}_X$ soient les premiers).

Définition B.11 (Confiance naturelle)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, soit $X \to Y$ une règle d'association sur $\mathcal{D}'$, sa confiance naturelle notée $c_{naturelle}(X \to Y)$ est :*

$$c_{naturelle}(X \to Y) = \frac{|s(X \sqcup Y)|}{|s_{\mathcal{D} \setminus \mathcal{D}'}(X)|} \ .$$

Lorsque X est une association sur $\mathcal{D}'$, comme pour la confiance exclusive, la confiance naturelle de $X \to Y$ vaut $\frac{|s(X \sqcup Y)|}{|s(X)|}$ sous la convention $\times_{D^i \in \emptyset} D^i = \{\epsilon\}$ que nous avons déjà utilisée.

Exemple B.10. *Dans $\mathcal{R}_E$, considérons à nouveau la règle $\{p_1, p_2\} \to \{s_1, s_2\}$. Les clients qui achètent les produits p_1 et p_2 ensemble (lors d'au moins une saison) sont o_1, o_2, o_3 et o_5. Ceux qui les achètent ensemble à la fois en s_1 et en s_2 sont o_1 et o_2. La confiance naturelle la règle vaut donc :*

$$c_{natural}(\{p_1, p_2\} \to \{s_1, s_2\}) = \frac{|s(\{p_1, p_2\} \times \{s_1, s_2\})|}{|s_{\{D^3\}}(\{p_1, p_2\})|} = \frac{|\{o_1, o_2\}|}{|\{o_1, o_2, o_3, o_5\}|} = \frac{2}{4} \ .$$

La confiance naturelle mesure ainsi, parmi les clients ayant au moins une fois acheté p_1 et p_2 ensemble, la proportion des clients qui les achètent ensemble à la fois aux deux saisons s_1 et s_2. À la différence de la confiance exclusive, les clients vérifiant la règle pourraient, par ailleurs, acheter p_1 et p_2 au cours d'autres saisons sans que cela ne fasse "baisser" la confiance naturelle.

La confiance naturelle a une bonne propriété lui permettant, contrairement à la confiance exclusive, un élagage de l'espace de recherche lors du calcul complet des règles à forte confiance.

Théorème B.2 (Condition à élaguer l'espace de recherche)**.** *Soit $X \to Y \setminus X$ et $X' \to Y \setminus X'$ deux règles d'association sur $\mathcal{D}' \subseteq \mathcal{D}$, on a :*

$$X' \sqsubseteq X \sqsubseteq Y \Rightarrow c_{naturelle}(X' \to Y \setminus X') \leq c_{naturelle}(X \to Y \setminus X) \ .$$

Preuve dans l'Annexe A.2.

Exemple B.11. *Dans $\mathcal{R}_E$, $\{p_1, p_2\} \to \{s_1, s_2\}$ et $\{p_1, p_2\} \times \{s_1\} \to \{s_2\}$ sont deux règle sur $\{D^1, D^2\}$. La confiance naturelle de la première règle est $\frac{2}{4}$ (voir ci-dessus). La confiance naturelle de la seconde est:*

$$\frac{|s(\{p_1, p_2\} \times \{s_1, s_2\})|}{|s_{D^3}(\{p_1, p_2\} \times \{s_1\})|} = \frac{|\{o_1, o_2\}|}{|\{o_1, o_2, o_3\}|} = \frac{2}{3}.$$

Ces deux règles illustrent le Théorème B.2. En effet, on a $\{p_1, p_2\} \sqsubseteq \{p_1, p_2\} \times \{s_1\} \sqsubseteq \{p_1, p_2\} \times \{s_1, s_2\}$ et $c_{naturelle}(\{p_1, p_2\} \to \{s_1, s_2\}) \leq c_{naturelle}(\{p_1, p_2\} \times \{s_1\} \to \{s_2\})$.

Dans la découverte des règles, nous utilisons ce théorème pour élaguer des sous-espaces de recherche où nous sommes certain qu'aucune règle ne pourra satisfaire une contrainte de confiance naturelle minimale.

Règle d'association canonique

Nous définissons maintenant un principe d'équivalence entre règles d'association et le concept de canonicité.

Définition B.12 (Équivalence syntaxique)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, les règles d'association $X \to Y$ et $X \to Z$ sur $\mathcal{D}'$ sont syntaxiquement équivalentes si et seulement si $X \sqcup Y = X \sqcup Z$.*

À partir des Définitions B.8, B.9 et B.11, on démontre directement le lemme suivant.

Lemme B.1. *Deux règles d'association syntaxiquement équivalentes ont même fréquence, même confiance exclusive et même confiance naturelle.*

Chaque règle d'association canonique représente sa classe d'équivalence syntaxique.

Définition B.13 (Règle d'association canonique)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, une règle d'association $X \to Y$ sur $\mathcal{D}'$ est canonique si et seulement si $\forall D^i \in \mathcal{D}, \pi_{D^i}(X) \cap \pi_{D^i}(Y) = \emptyset$.*

Toute collection complète de règles d'association satisfaisant des contraintes sur leurs fréquences et/ou confiances peut être résumée, sans perte d'information, à celles qui, parmi elles, sont canoniques. En effet, étant donné une règle d'association canonique $X \to Y$ dans la collection, le Lemme 2 permet d'affirmer la présence, dans la collection, de toutes les règles qui lui sont syntaxiquement équivalentes. De plus, les construire est facile : ce sont les règles de la forme $X \to Y \sqcup Z$ avec $Z \sqsubseteq X$.

B.2.4 Calcul de règles a priori intéressantes

Face à une relation n-aire $\mathcal{R} \subseteq \times_{D^i \in \mathcal{D}} D^i$, nous voulons calculer des collections de règles a priori intéressantes, ce qui se traduit ici par le calcul de *toutes* les règles d'association canoniques :

- définies sur un sous-ensemble $\mathcal{D}' \subsetneq \mathcal{D}$;
- ayant une fréquence supérieure à un seuil $\mu \in [0; 1]$;
- ayant une confiance exclusive supérieure à un seuil $\beta_{\text{exclusive}} \in [0; 1]$;
- ayant une confiance naturelle supérieure à un seuil $\beta_{\text{naturelle}} \in [0; 1]$.

Plus formellement, une fois qu'un analyste a spécifié $\mathcal{D}' \subsetneq \mathcal{D}$ et les différents seuils (μ, $\beta_{\text{exclusive}}$ et $\beta_{\text{naturelle}}$), l'algorithme PINARD [3] doit calculer :

$$\{X \to Y \text{ canonique sur } \mathcal{D}' \mid \begin{cases} f(X \to Y) \geq \mu \\ c_{\text{exclusive}}(X \to Y) \geq \beta_{\text{exclusive}} \\ c_{\text{naturelle}}(X \to Y) \geq \beta_{\text{naturelle}} \end{cases} \}.$$

Cette tâche sera effectuée en trois étapes : la construction du domaine support, l'extraction de l'ensemble des associations qui satisfont la contrainte de fréquence minimale, puis l'extraction des règles dont les confiances exclusive et naturelle dépassent les seuils choisis par l'analyste.

Construction du domaine support

Le domaine support des règles d'association sur $\mathcal{D}'$ est $D^{\text{support}} = \times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$. Soit $\mathcal{D}_A = \mathcal{D}' \cup D^{\text{support}}$. La relation $\mathcal{R}_A$ sur $\mathcal{D}_A$ est construite de la façon suivante:

$$\mathcal{R}_A = \{(e_1, e_2, \ldots, e_{|\mathcal{D}'|}, (e_{|\mathcal{D}'|+1}, \ldots, e_n)) \mid (e_1, e_2, \ldots, e_{|\mathcal{D}'|}, e_{|\mathcal{D}'|+1}, \ldots, e_n) \in \mathcal{R}\} \ .$$

Extraction des associations fréquentes

La fréquence d'une règle d'association sur $\mathcal{D}'$ est supérieure ou égale à μ si et seulement si l'union de sa prémisse et de sa conclusion est une association dont le support contient au moins $\alpha = \lceil \mu \times |D^{\text{support}}| \rceil$ éléments. L'extraction complète de telles associations ressemble au problème de l'extraction des itemsets fréquents dans une relation binaire. Cependant, il est doit être généralisé au contexte des relations

3. PINARD Is N-ary Association Rule Discovery.

n-aires. Un algorithme comme DATA-PEELER [33] résout un problème assez proche : il impose la fermeture des associations alors que nous souhaitons ici lister toutes les associations fréquentes, qu'elles soient fermées ou non. Nous avons donc modifié DATA-PEELER et ne présentons ici qu'une vision très abstraite de cette phase (voir [33] pour des détails).

Extraire toutes les associations A sur $\mathcal{D}'$ avec au moins α éléments dans son support peut s'exprimer comme le calcul de chaque association $A \sqcup A^{\text{support}}$ sur $\mathcal{D}_A$ satisfaisant les quatre contraintes suivantes :

- $\mathcal{C}_{\text{sur-}\mathcal{D}'}(A \sqcup A^{\text{support}}) \equiv \forall D^i \in \mathcal{D}', \pi_{D^i}(A) \neq \emptyset$;
- $\mathcal{C}_{\text{connecté}}(A \sqcup A^{\text{support}}) \equiv A \sqcup A^{\text{support}} \subseteq \mathcal{R}_A$;
- $\mathcal{C}_{\text{support-entier}}(A \sqcup A^{\text{support}}) \equiv A^{\text{support}} = s(A)$;
- $\mathcal{C}_{\alpha\text{-fréquent}}(A \sqcup A^{\text{support}}) \equiv |A^{\text{support}}| \geq \alpha$.

La dernière contrainte traduit l'obligation, pour les règles utilisant tous les éléments de $\cup_{D^i \in \mathcal{D}'} \pi_{D^i}(A)$, d'excéder la fréquence minimale μ. En effet $\frac{|s(A)|}{|D^{\text{support}}|} \geq \mu$ équivaut à $|s(A)| \geq \alpha$ et, comme l'avant-dernière contrainte ($A^{\text{support}} = s(A)$) doit également être vérifiée, on trouve bien $|A^{\text{support}}| \geq \alpha$. L'avant-dernière contrainte, $\mathcal{C}_{\text{support-entier}}$, force un support "fermé". En effet, par définition du support d'une association, ajouter un élément à A^{support} ($= s(A)$) viole forcément $\mathcal{C}_{\text{connecté}}$. $\mathcal{C}_{\text{support-entier}}(A \sqcup A^{\text{support}})$ équivaut ainsi à $\forall t \in D^{\text{support}} \setminus A^{\text{support}}$, $A \sqcup \{t\} \not\subseteq \mathcal{R}_A$. C'est sous cette forme que nous l'utiliserons.

L'extracteur, que nous appelons ASSOCIATIONS, parcourt l'espace de recherche en le partitionnant en deux à chaque appel récursif. L'énumération suit donc un arbre binaire. À chaque nœud de l'arbre, deux associations, appelées U et V, sont telles que U est la plus petite association (au sens de $\sqsubseteq$) qui pourra être extraite depuis ce nœud, $U \sqcup V$ la plus grande (au sens de $\sqsubseteq$). Ainsi, l'appel initial de ASSOCIATIONS se fait avec $U = \emptyset$ et $V = \times_{D^i \in \mathcal{D}_A} D^i$ et toutes les associations dans $\mathcal{R}_A$ vérifiant les quatre contraintes listées précédemment sont extraites. Les nœuds qui ne sont pas des feuilles ont deux fils. Un premier fils où un élément $e \in \cup_{D^i \in \mathcal{D}_A} \pi_{D^i}(V)$ est choisi pour être présent dans les associations qui seront extraites dans le sous-arbre d'énumération dont il est racine (e est "déplacé" de V vers U). Un second fils où ce même élément est déclaré absent des associations dans le sous-arbre d'énumération dont il est racine (e est "supprimé" de V).

Deux raisons peuvent faire qu'un nœud est une feuille de l'arbre d'énumération. La première raison est l'assurance qu'au moins une des quatre contraintes n'est vérifiée par aucune association U dans le sous-arbre d'énumération qui dériverait du nœud. C'est le cas lorsque :

- $\exists D^i \in \mathcal{D}' \mid \pi_{D^i}(U \sqcup V) = \emptyset$ ($\mathcal{C}_{\text{sur-}\mathcal{D}'}$ est violée);
- $\times_{D^i \in \mathcal{D}_A} \pi_{D^i}(U) \not\subseteq \mathcal{R}_A$ ($\mathcal{C}_{\text{connecté}}$ est violée);
- $\exists t \in D^{\text{support}} \setminus \pi_{D^{\text{support}}}(U \sqcup V) \mid \Big(\times_{D^i \in \mathcal{D}'} \pi_{D^i}(U \sqcup V) \Big) \times \{t\} \subseteq \mathcal{R}_A$ ($\mathcal{C}_{\text{support-entier}}$ est violée);
- $|\pi_{D^{\text{support}}}(U \sqcup V)| < \alpha$ ($\mathcal{C}_{\alpha\text{-fréquent}}$ est violée).

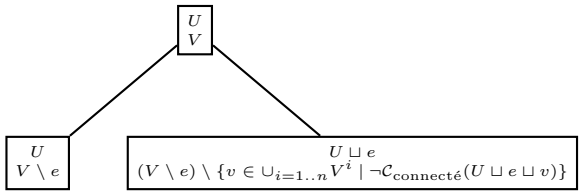


Figure B.2: Enumération de l'élément $e \in \cup_{D^i \in \mathcal{D}_A} \pi_{D^i}(V)$.

Les preuves de ces propriétés d'élagage reposent sur une généralisation des notions de monotonie et d'anti-monotonie qui sont vérifiées par les quatre contraintes. La contrainte $\mathcal{C}_{\text{connecté}}$, dont la variable a été remplacée par U, est monotone : lorsque U viole la contrainte, toutes les associations plus grandes (au sens de $\sqsubseteq$) la violent également. De façon duale, les autres contraintes, dont la variable a été remplacée par $U \sqcup V$, sont anti-monotones : lorsque $U \sqcup V$ viole la contrainte, toutes les associations plus petites (au sens de $\sqsubseteq$) la violent également. L'autre raison qui peut faire qu'un nœud est une feuille de l'arbre d'énumération est que $V = \emptyset$. Il n'y a alors plus d'élément à énumérer. Si les quatre contraintes sont vérifiées, U est alors une association à partir de laquelle des règles d'association seront construites.

Une stratégie d'énumération améliorée évite de générer des nœuds violant $\mathcal{C}_{\text{connecté}}$ puis d'élaguer l'espace de recherche. À la place, à chaque appel récursif, on supprime de $\cup_{D^i \in \mathcal{D}_A} \pi_{D^i}(V)$ les éléments qui, si ils étaient "déplacés" vers U, violeraient $\mathcal{C}_{\text{connecté}}$. Ainsi, après avoir choisi un élément e à énumérer, les nœuds fils sont tels que décrits par Figure B.2. L'algorithme d'extraction des associations fréquentes est donné sous forme de pseudo-code (Algorithme B.1).

Entrée : (U, V)
Sortie : Toutes les associations fréquentes qui sont plus grandes que U et plus petites que $U \sqcup V$ (au sens de $\sqsubseteq$)
si $\mathcal{C}_{\text{sur-}\mathcal{D}'}(U \sqcup V) \wedge \mathcal{C}_{\text{support-entier}}(U \sqcup V) \wedge \mathcal{C}_{\alpha\text{-fréquent}}(U \sqcup V)$ **alors**
  si $V = \emptyset$ **alors**
    Sortir $U \setminus \pi_{D^{\text{support}}}(U)$
  sinon
    Choisir $e \in \cup_{D^i \in \mathcal{D}_A} \pi_{D^i}(V)$
    Associations$(U \sqcup \{e\}, (V \setminus e) \setminus \{v \in \cup_{i=1..n} V^i \mid \neg \mathcal{C}_{\text{connecté}}(U \sqcup e \sqcup v)\})$
    Associations$(U, V \setminus \{e\})$
  fin si
fin si

Algorithme B.1: Associations.

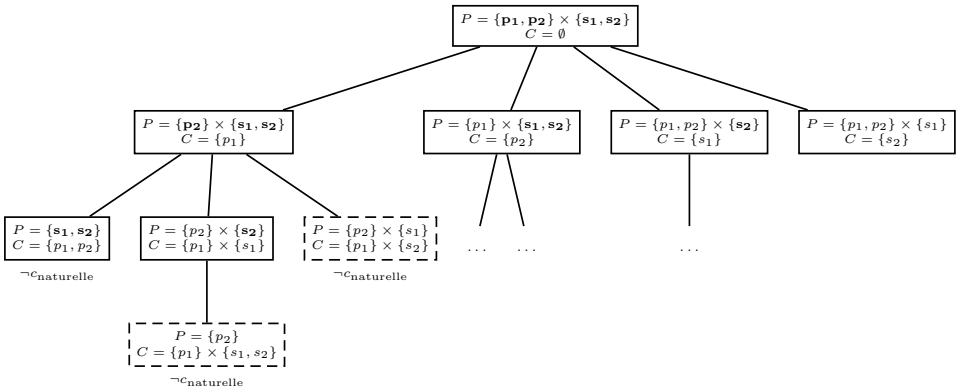


Figure B.3: Calcul des règles à partir d'une association.

Extraction des règles avec les confiances minimales

À partir d'une association fréquente A extraite par ASSOCIATIONS, il s'agit maintenant de construire des règles d'association canoniques utilisant *tous* les éléments de $\cup_{D^i \in \mathcal{D}'} \pi_{D^i}(A)$. Chacune de ces règles, $P \to C$, répartit ces éléments entre prémisse, P, et conclusion, C. En d'autres termes $P \sqcup C = A$. Pour énumérer ces règles, la stratégie d'énumération choisie construit un arbre. À chaque nœud de l'arbre est associée une règle d'association candidate. En d'autres termes, P et C sont instanciés et, si $P \to C$ vérifie les contraintes de confiances naturelle et exclusive minimales, alors elle est retenue.

En ne regardant que les conclusions (C) et étant donné A, la prémisse(P) est unique, $P = A \setminus C$. En particulier, sa racine est $A \to \emptyset$ et C grandit d'un élément (via $\sqcup$) à chaque niveau de l'arbre (en parallèle, P se voit retirer ce même élément). Néanmoins cet arbre est, dans notre cas, parcouru en profondeur et ce n'est pas une fréquence minimale qui l'élague mais la confiance naturelle minimale. Le théorème à l'œuvre est donc le Théorème B.2.

Par exemple, dans $\mathcal{R}_E$, considérons l'extraction de règles d'association canoniques ayant une confiance naturelle d'au moins $0,6$. Figure B.3 illustre le processus de production de ces règles à partir de l'association $A = \{p_1, p_2\} \times \{s_1, s_2\}$, extraite par ASSOCIATIONS. Les éléments de $\cup_{D^i \in \mathcal{D}'} \pi_{D^i}(A)$ sont ordonnés de façon arbitraire. Dans cet exemple, l'ordre $\prec$ choisi est $p_1 \prec p_2 \prec s_1 \prec s_2$. À chaque nœud, les éléments qui peuvent augmenter (via $\sqcup$) la conclusion sont ceux qui sont plus grands (selon $\prec$) que tous les éléments déjà en conclusion (autrement dit, plus grand que $\max_{\prec}(C)$, (pour l'appel initial à ASSOCIATIONS, l'élément $\max_{\prec}(\emptyset)$ est défini comme plus petit que tous les autres dans l'ordre $\prec$). Sur la figure, ces éléments sont en gras. Un nœud sans élément en gras n'a aucun fils. Un nœud qui ne satisfait pas la contrainte de confiance naturelle minimale (il est, sur la figure, encadré de pointillés), n'en a pas non plus. D'après le Théorème B.2, cet élagage est *sûr* : aucune règle avec une confiance suffisante n'est manquée.

Comme nous l'avons vu, et contrairement à la confiance naturelle, la confiance exclusive n'est pas anti-monotone. Vérifier la contrainte de confiance exclusive minimale est donc l'ultime condition à vérifier pour produire la règle mais elle ne donne jamais lieu à élagage. L'Algorithme B.2 résume l'extraction des règles canoniques de confiance suffisante depuis une association fréquente A. Pour obtenir de bonnes performances, précisons que les confiances (exclusive et naturelle) sont, autant que possible, calculées sans retour à $\mathcal{R}_A$. Déjà $|s(P \sqcup C)| = |s(A)|$, qui intervient dans les deux définitions, est constante et connue dès l'extraction de A par Associations. Ensuite $|s(P)|$ est connue si aucune de ses composantes n'est vide : en effet, puisque $P \sqsubseteq (P \sqcup C)$, le Théorème B.2 nous assure que P est une association fréquente sur $\mathcal{D}'$ et a donc été extraite par Associations. Enfin, à chaque calcul de $|s(P)|$ (P a alors une composante vide) ou de $|s_{\mathcal{D}'}(P)|$ depuis $\mathcal{R}_A$, la valeur est stockée pour éviter de la calculer à nouveau si cette même prémisse est considérée pour une autre règle.

Entrée : (P, C)
Sortie : Toutes les règles d'association canoniques qui utilisent tous les éléments de $\cup_{D^i \in \mathcal{D}'} \pi_{D^i}(P \sqcup C)$, avec une prémisse plus petite que P (selon $\sqsubseteq$), une conclusion plus grande que C (selon $\sqsubseteq$) et satisfaisant les contraintes de confiance minimale
pour tout $e \succ \max_{\prec}(C)$ **faire**
  $(P', C') \leftarrow (P \setminus \{e\}, C \sqcup \{e\})$
  si $c_{\text{naturelle}}(P' \rightarrow C') \geq \beta_{\text{naturelle}}$ **alors**
    si $c_{\text{exclusive}}(P' \rightarrow C') \geq \beta_{\text{exclusive}}$ **alors**
      Sortir $P' \rightarrow C'$
    fin si
    Règles(P', C')
  fin si
fin pour

Algorithme B.2: Règles.

Nous pouvons maintenant donner l'Algorithme B.3 qui répond au problème du calcul des règles a priori intéressantes sur $\mathcal{D}' \subsetneq \mathcal{D}$.

B.2.5 Validation empirique

Ensemble de données: `DistroWatch`

`DistroWatch` est un site Web qui rassemble une information complète sur les distributions GNU/Linux, BSD et Solaris. Chaque distribution est décrite sur une page séparée. Lorsque qu'un visiteur charge une page, on considère que la distribution qu'elle décrit l'intéresse. L'adresse IP du visiteur nous permet de connaître son pays. Les données produites pendant sont agrégées par semestre (13 semestres à partir de

Entrée : Relation $\mathcal{R}$ sur $\mathcal{D} = \{D^1, \ldots, D^n\}$, $\mathcal{D}' \subsetneq \mathcal{D}$, $(\mu, \beta_{\text{exclusive}}, \beta_{\text{naturelle}}) \in [0;1]^3$
Sortie : Toutes les règles d'association canoniques sur $\mathcal{D}'$ satisfaisant les contraintes de fréquence et de confiances minimales
$D^{\text{support}} \leftarrow \times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$
$(\mathcal{D}_A, \mathcal{R}_A) \leftarrow (\mathcal{D}' \cup D^{\text{support}}, \emptyset)$
pour tout $(e_1, e_2, \ldots, e_{|\mathcal{D}'|}, e_{|\mathcal{D}'|+1}, \ldots, e_n) \in \mathcal{R}$ **faire**
  $\mathcal{R}_A \leftarrow \mathcal{R}_A \cup (e_1, e_2, \ldots, e_{|\mathcal{D}'|}, (e_{|\mathcal{D}'|+1}, \ldots, e_n))$
fin pour
$\alpha \leftarrow \lceil \mu \times |D^{\text{support}}| \rceil$
$\mathcal{A} \leftarrow$ ASSOCIATIONS$(\emptyset, \times_{D^i \in \mathcal{D}_A} D^i)$
pour tout $A \in \mathcal{A}$ **faire**
  RÈGLES$(A, \emptyset)$
fin pour

Algorithme B.3: PINARD.

début 2004 à début 2010), par pays et par distribution (655 distributions). Seuls les pays associés à au moins 2000 consultations d'une distribution lors d'un semestre ont été gardés (96 pays gardés). Les données numériques sont ensuite normalisées de sorte que tous les pays (resp. tous les semestres) aient la même importance. Enfin, elles sont transformées en une relation ternaire listant les triplets les plus significatifs. Ces derniers sont choisis à l'aide d'une procédure locale (i.e., par distribution) inspirée du calcul d'une valeur p : pour chaque distribution, on garde ses triplets associés aux plus grandes valeurs numériques jusqu'à ce que leur somme atteigne 20% de la somme de toutes les valeurs impliquant la distribution. Si un 3-uplet (p, d, s) appartient à la relation, cela signifie qu'une quantité importante d'utilisateurs de pays p ont visité la description de la distribution d pendant le semestre s. Nous appelons la relation ainsi obtenue $\mathcal{R}_{\texttt{DistroWatch}}$. Elle contient 21,033 triplets, sa densité est $\frac{21{,}033}{13 \times 655 \times 96} = 2.6\%$.

Étude qualitative

Nous souhaitons découvrir des règles associant pays et distributions (ces deux dimensions forment l'ensemble que nous avons appelé $\mathcal{D}'$ jusqu'à maintenant). PINARD est utilisé avec pour seuils de fréquence et de confiances $\mu = 0,75$, $\beta_{\text{exclusive}} = 0,6$ et $\beta_{\text{naturelle}} = 0,8$. On extrait alors 58 règles d'association canoniques. Parmi elles :

- $\{\text{Taiwan}\} \times \{\text{fedora}\} \rightarrow \{\text{b2d}\}$
  (f : 0.846, c_{natural} : 0.917, $c_{\text{exclusive}}$: 0.917);
- $\{\text{Japan}\} \times \{\text{centos}\} \rightarrow \{\text{Ecuador}\}$
  (f : 0.769, c_{natural} : 0.909, $c_{\text{exclusive}}$: 0.909);
- $\{\text{berry,plamo}\} \rightarrow \{\text{Japan}\}$

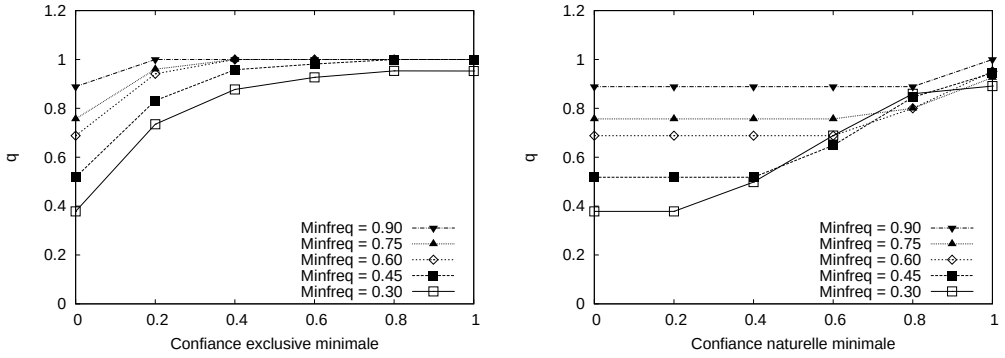
(f : 0.923, c_{natural} : 1, $c_{\text{exclusive}}$: 0.75);
– {berry,momonga,plamo} → {Japan}
(f : 0.769, c_{natural} : 1, $c_{\text{exclusive}}$: 1);
– {caixamagica} → {Portugal}
(f : 0.846, c_{natural} : 1, $c_{\text{exclusive}}$: 1).

La première règle ci-dessus indique que si les Taiwaneses s'intéressent à *fedora* puis ils s'intéressent aussi à *b2d* au même semestre, la confiance est plus grande que 0,9 ($c_{\text{naturel}} = c_{\text{exclusive}} = 0,917$). La probabilité pour *centor* est consulté par les Equatoriens, en même temps où les Japonais le consultent, est supérieur à 90% (à la seconde règle, $c_{\text{naturel}} = c_{\text{exclusive}} = 0,909$). Japon est le pays d'origine des distributions *berry*, *plamo* et *momonga*, c'est à dire, ces distributions sont développées par les Japonais. C'est pourquoi presque des visiteurs de ces distributions sont des Japonais. La confiance naturelle de la troisième règle est de 1, cela signifie que les Japonais visitent Berry et Plamo à tout le semestre, lorsque ces distributions sont visitées ensemble. Cette règle indique aussi que les gens d'autres pays consultent rarement ces distributions au même semestre ($1 - c_{\text{exclusive}} = 0,25$). Parce que la quatrième règle indique que les trois distributions *berry*, *plamo* et *momonga* sont visitées au même semestre, la confiance exclusive est plus élevé. Il est 1, autrement dit, en dehors du Japon, aucun autre pays charge souvent ces trois distributions au même semestre. La même interprétation vaut pour la dernière règle. La distribution *caixamagica* est développée par et pour les personnes aux Philippines. Elle est visitée exclusivement par eux ($c_{\text{naturel}} = c_{\text{exclusive}} = 1$).

Les règles que nous venons de détailler font sens puisque les distributions qui sont développées spécifiquement par et pour un pays intéressent particulièrement les internautes de ce pays. Il se trouve que les règles que nous n'avons pas discutées mais qui ont des distributions en prémisse et des pays en conclusion, sont majoritairement interprétables de cette façon. Pour en rendre compte et valider les mesures de confiances que nous avons définies, la Figure B.4 représente, pour différents paramétrages, la valeur suivante :

$$q = \frac{|\{X \to Y \mid \begin{cases} X \subseteq D^{\text{distributions}} \wedge Y \subseteq D^{\text{countries}} \\ \forall y \in Y, \exists x \in X \mid \text{origin}(x) = y \end{cases} \}|}{|\{X \to Y \mid X \subseteq D^{\text{distributions}} \wedge Y \subseteq D^{\text{countries}}\}|}$$

où origine(x) est le pays d'où provient la distribution x. Lorsque les seuils de confiances minimales augmentent, q augmente. Globalement, lorsque le seuil de fréquence minimale augmente, q augmente aussi. Cela corrobore donc empiriquement les choix des sémantiques associées à ces mesures. q augmente plus vite avec $c_{\text{exclusive}}$ qu'avec $c_{\text{naturelle}}$. Les paliers observés sur la Figure B.4b pour $\beta_{\text{naturelle}} \leq \mu$ sont des conséquences directes des Définitions B.8 et B.11 : les règles extraites sont les mêmes.



(a) q lorsque $\beta_{\text{exclusive}}$ varie ($\beta_{\text{naturelle}} = 0$). (b) q lorsque $\beta_{\text{naturelle}}$ varie ($\beta_{\text{exclusive}} = 0$).

Figure B.4: Validation qualitative des mesures

Élagage dans la génération des règles

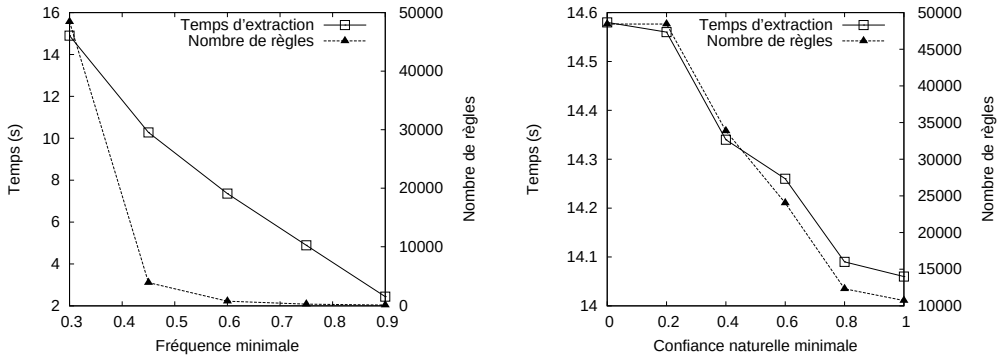
Si le seuil de fréquence augmente, le nombre d'associations fréquentes ne peut que diminuer. Dans l'algorithme ASSOCIATIONS, la vérification de la contrainte $\mathcal{C}_{\alpha\text{-fréquent}}$ va donc élaguer de plus grandes régions de l'espace de recherche où la contrainte est violée. Alors le nombere de règles et le temps d'extraction va décroître avec le seuil de fréquence minimale. Figure B.5a illustre l'efficacité de l'élagage de l'algorithme ASSOCIATIONS sur l'extraction des règles associant pays et distributions dans $\mathcal{R}_{\texttt{DistroWatch}}$ avec $\beta_{exclusive} = \beta_{naturelle} = 0$ et le seuil de fréquence minimale varie de $0,3$ à $0,9$.

En exploitant le Théorème B.2, l'algorithme RÈGLES élague les arbres dérivant des associations fréquentes. Ainsi, quand le seuil de confiance naturelle augmente, le nombre de règles et le temps de calcul de ces règles diminuent (Figure B.5b). Cette expérience est réalisée sur $\mathcal{R}_{\texttt{DistroWatch}}$ avec $\beta_{exclusive} = 0$, $\beta_{naturelle} = 0,3$ et le seuil de confiance naturelle minimale varie de 0 à 1.

L'évolutivité de PINARD a été testé sur l'extraction des règles d'assciation sur les deux domaines *Pays* et *Distributions* avec $\mu = 0,75$ et $\beta_{\text{naturelle}} = \beta_{\text{exclusive}} = 0$. $\mathcal{R}_{\texttt{DistroWatch}}$ a été reproduite jusqu'à 10 fois sur le domaine *Semestres*. Il montre que l'algorithme est linéaire. Plus précisément, une régression linéaire de $R \mapsto \frac{T_R}{T_1}$ (où R est le facteur de réplication; T_R est le temps d'exécution sur cet jeu de données répliquée) donne $y = 2.27x - 2.91$ avec $0,96$ comme un coefficient de détermination. Ainsi, PINARD se comporte linéairement selon le facteur de réplication.

B.2.6 Règle d'association multidimensionnelle non redondante

Si une règle implique l'information qui est incluse dans une autre règle plus générale, alors la règle n'est pas intéressante et elle n'est pas nécessaire de tourner.



(a) Élagage w.r.t. μ. (b) Élagage w.r.t. $\beta_{naturelle}$.

Figure B.5: Efficacité de PINARD

Exemple B.12. *Dans $\mathcal{R}_E$, considérons les règles suivantes:*
- *r_3: $\{s_1, s_3\} \rightarrow \{p_2, p_3, p_4\}$ ($f : 0.4$, $c_{natural} : 0.67$, $c_{exclusive} : 0.86$),*
- *r_4: $\{p_2\} \times \{s_1, s_3\} \rightarrow \{p_3\}$ ($f : 0.4$, $c_{natural} : 0.67$, $c_{exclusive} : 0.67$),*
- *r_5: $\{p_1\} \times \{s_2\} \rightarrow \{p_2\} \times \{s_1\}$ ($f : 0.4$, $c_{natural} : 1$, $c_{exclusive} : 1$),*
- *r_6: $\{p_1\} \times \{s_1, s_2\} \rightarrow \{p_2\}$ ($f : 0.4$, $c_{natural} : 1$, $c_{exclusive} : 1$),*

Elles sont toutes canoniques et leurs frequences, leurs confiances exclusive et leurs confiances naturelles dépassent respectivement 0.4, 0.6, *et* 0.6. *À cet égard, elles satisfont individuellement à cet aspect d'ntérêt. Néanmoins, tous ensemble, elles fournissent des redondances. Par exemple, r_4 est plus spécifique que r_3, car elle nécessite plus de condition pour appliquer (les achats doivent impliquer p_2) et sa conclusion est moins informative (elle ne dit rien sur p_4). Cependant, cette spécialisation n'attribue pas la fréquence et les confiances de r_4 plus grandes que ceux de r_3. Par conséquent, r_4 est dite d'être redondante. De même, par l'existence de r_5, r_6 est redondante. Puisque l'analyste ne trouverait aucune valeur ajoutée dans les règles r_4 et r_6, elles ne devraient pas être retournées.*

Nous généralisons la notion de règle non-redondante ayant la prémisse minimale et la conclusion maximale [82] pour notre établissement de la règle d'association multidimensionnelle.

Définition B.14 (Règle d'association multidimensionnelle non-redondante)**.** *$\forall \mathcal{D}' \subseteq \mathcal{D}$, une règle $X \rightarrow Y$ sur $\mathcal{D}'$ est non-redondante si et seulement si elle est canonique et aucune autre règle canonique $X' \rightarrow Y'$ est tel que:*

$$\begin{cases} ((X' \sqcup Y' = X \sqcup Y) \wedge (X' \sqsubset X)) \vee ((X' \sqcup Y' \sqsupset X \sqcup Y) \wedge (X' \sqsubseteq X)) \\ f(X' \rightarrow Y') \geq f(X \rightarrow Y) \\ c_{exclusive}(X' \rightarrow Y') \geq c_{exclusive}(X \rightarrow Y) \\ c_{natural}(X' \rightarrow Y') \geq c_{natural}(X \rightarrow Y) \end{cases}.$$

L'extraction des règles d'association non-redondantes est présentée dans la section 3.3 (l'algorithme PINARD++). Les expériences pour évaluer l'avantage de l'élimination des règles redondantes et pour évaluer l'efficacité de PINARD++ sont présentées dans la section 3.4.

B.3 Règles disjonctives dans une relation n-aire

Une association peut fréquemment co-appaître avec certains autres associations, alors que ces associations ne vont pas nécessairement co-appaître ensemble. Par exemple, observons la relation $\mathcal{R}_E$ (voir la Figure B.1), les produits p_1, p_2 et p_4 sont fréquentement achetés en saison s_2. Toutefois, un client achète rarement tous ces produits ensemble dans la même transaction. Ainsi, la fouille de règles d'association multidimensionnelles ne peut pas fournir une règle comme $\{s_2\} \rightarrow (\{p_1, p_2\} \times \{s_1\}) \vee (\{p_4\} \times \{s_4\})$. Une telle règle signifie que quand un client fait des courses en saison s_2, il/elle a tendance à acheter p_1, p_2 ou p_4. Si il/elle préfère les produits p_1, p_2, alors il/elle les achète aussi en saison s_1. Si il/elle préfère p_4, alors il/elle a tendance à également l'acheter en saison s_4. En effet, des telles règles ont plus information que des règles conjonctives.

En outre, il est inefficace de trouver des règles d'association multidimensionnelles sur un jeu de données ayant un très petit nombre d'associations fréquentes et un très grand nombre d'associations infréquentes, parce que les règles d'association sont basées sur la relation de co-occurrence des associations avec la fréquence et les confidences assez grandes.

Nous abordons les problèmes ci-dessus en introduisant des disjonctions dans les conclusions des règles qui s'appelent règles disjonctives multidimensionnelles. Nos objectifs sont de répondre à la question "*Quelles associations peuvent appaître lorsque l'on observe une association fréquente?*" et de fouiller des règles dans lesquelles une association avec une grandes fréquence implique des associations avec une fréquence faible.

Soit n ensembles finis supposés disjoints $\{D^1, \ldots, D^n\} = \mathcal{D}$, $\mathcal{R} \subseteq D^1 \times \cdots \times D^n$ la relation n-aire à partir de laquelle on souhaite découvrir des règles, $\mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\}$ l'ensemble de domaines d'intérêt défini par l'utilisateur, une *règle disjonctive multidimensionnelle* sur $\mathcal{D}'$ est de la forme $X \rightarrow \vee\mathcal{Y}$ telle que l'union de sa prémisse et chaque association dans les disjonctions de sa conclusion est une association sur $\mathcal{D}'$. Il s'appelle simplement une règle quand elle est claire dans le contexte.

Définition B.15 (Règle disjonctive multidimensionnelle)**.** *$\forall\mathcal{D}' \subseteq \mathcal{D}$, $X \rightarrow \vee\mathcal{Y}$ est une règle disjonctive multidimensionnelle sur $\mathcal{D}'$ si et seulement si X est une association sur un sous-ensemble de $\mathcal{D}'$ et $\mathcal{Y}$ est un ensemble d' associations sur des sous-ensembles de $\mathcal{D}'$ tels que $\forall Y \in \mathcal{Y}$, $X \sqcup Y$ est une association sur $\mathcal{D}'$.*

Le domaine support d'une règle disjonctive multidimensionnelle sur $\mathcal{D}'$ est le *produit Cartésien* de tous domaines qui ne sont pas dans $\mathcal{D}'$, i. e.., $\times_{D \in \mathcal{D} \setminus \mathcal{D}'} D$.

Exemple B.13. *Dans* $\mathcal{R}_E$, $\{s_2\} \rightarrow (\{p_1, p_2\} \times \{s_1\}) \vee (\{p_4\} \times \{s_4\})$, $\{p_3, p_4\} \rightarrow (\{p_2\} \times \{s_1, s_3\}) \vee (\{s_4\})$ *et* $\{p_4\} \times \{s_1\} \rightarrow (\{p_1, p_2\}) \vee (\{p_2\}) \vee (\{p_2, p_3\} \times \{s_3\})$ *sont trois règles disjonctives multidimensionnelles sur* $\{D^1, D^2\}$. *Leur domaine support est* D^3.

Étant donné une règle disjonctive multidimensionnelle, nous voulons d'abord d'évaluer la probabilité de la conjonction entre la prémisse et chaque association dans la conclusion, les measures s'appellent fréquence d'association (Definition 49) et confiance d'association (Definition 50). Nous voulons aussi mesurer la probabilité d'observer au moins une association dans sa conclusion où sa prémisse apparaît, et ces measures s'appellent fréquence disjonctive (Definition 51) et confiance disjonctive (Definition 52).

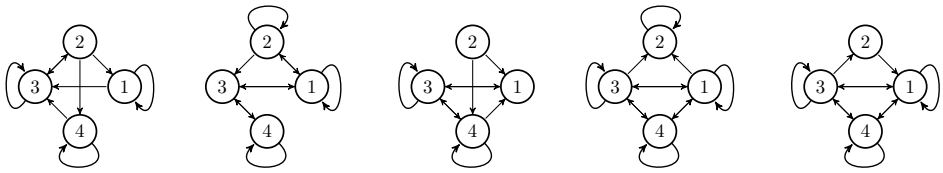
Il peut y avoir un grand nombre d'associations co-apparaissant fréquemment avec une association fréquente donnée. Supposons que le nombre d'associations co-apparaissant fréquemment avec l'association fréquente donnée est k, le nombre de disjonctions qui peuvent être générées à partir des sous-ensembles de ces associations est 2^k. Donc, étant donné une association fréquente, il y a un grand nombre de règles disjonctives multidimensionnelles générées qui satisfont aux contraintes sur les mesures d'intérêt minimales. Le calcul de la fouille de toutes ces règles est cher, ici encore, nous devons faire face à des règles redondantes. Nous considérons qu'une règle est non-redondante si son contenu d'information n'est pas impliqué dans une autre règle plus générale. Cela signifie qu'une règle non-redondante a une prémisse minimale et une conclusion maximale. Plus spécifiquement, une règle est non-redondante si sa prémisse est une association minimale et sa conclusion comprend le nombre maximal d'associations qui peuvent conjoindre avec sa prémisse telle que l'union la prémisse et chaque association dans la conclusion est un ensemble fermé.

L'extraction de règles disjonctives multidimensionnelles non-redondantes est présentée dans la section 4.3 (l'algorithme CIDRE [4]). Les expériences pour vérifier la signification de la fouille de règles disjonctives multidimensionnelles et pour évaluer l'efficacité de CIDRE sont présentées dans la section 4.4.

B.4 Application à l'analyse des graphes relationnels dynamiques

Nous présentons une approche pour la détection de motifs qui peuvent co-apparaître dans l'évolution d'un graphe dynamique grâce à la fouille de règles d'association (or disjonctives) multidimensionnelles.

4. CIDRE Is a Disjunctive Rule Extractor.



	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4
d_1	1		1		1	1	1				1		1	1	1	1	1		1	1
d_2	1		1	1	1	1	1		1			1		1			1			
d_3		1	1		1			1	1		1	1	1	1	1	1	1	1	1	1
d_4			1	1			1	1	1		1	1	1		1	1	1		1	1
	t_1				t_2				t_3				t_4				t_5			

Figure B.6: Exemple d'un graphe relationnel dynamique

En effet, nous nous concentrons sur des graphes relationnels dynamiques dont les sommets sont fixés, des arcs orientés peuvent changer (i. e.., apparaître ou disparaître) dans le temps. Nous codons un tel graphe relationnel dynamique en une relation n-aire qui est au moins ternaire (deux dimensions sont utilisées pour coder les matrices d'adjacence, une telle matrice décrit le graphe à un temps, et au moins une autre dimension indique les temps).

Exemple B.14. *Figure B.6 représente un tel graphe relationnel dynamique: il décrit la relation entre les sommets de départ dans* $D^1 = \{d_1, d_2, d_3, d_4\}$ *et les sommets d'arrivée dans* $D^2 = \{a_1, a_2, a_3, a_4\}$ *aux temps dans* $D^3 = \{t_1, t_2, t_3, t_4, t_5\}$*. Chaque '*1*' dans la relation* $\mathcal{R}_G \subseteq D^1 \times D^2 \times D^3$ *est à l'intersection de trois éléments* (d_i, a_j, t_k)*, ce qui indique un arc orienté de* a_j *à* d_i *au temps* t_k*.*

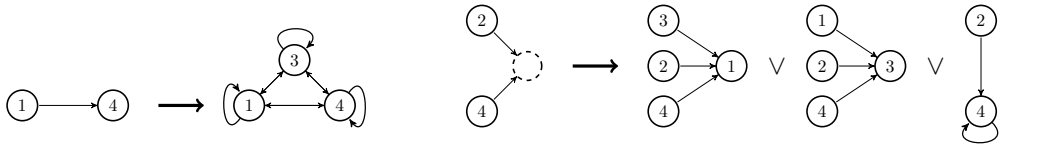
Pour détecter les co-occurrences de des motifs dans un graphe dynamique, nous d'abord codons le graphe dynamique en une relation n-aire, nous ensuite considérons chaque motif comme une association dans la relation n-aire, la co-occurrence de des motifs dans le graphe dynamique est considérée comme la co-occurrence de des associations dans une règle d'association (or disjonctive) multidimensionnelle dans la relation n-aire. Nous introduisons aussi certaines contraintes qui nous permettent non seulement d'extraire des règles spécifiques (par rapport le sujet d'intérêt), mais aussi d'améliorer l'efficacité de la phase d'extraction.

Exemple B.15. *Quelques exemples de règles pouvant être fouillées à partir du graphe dynamique dans Figure B.6 sont donnés dans Figure B.7 et dans Figure B.8.*

Les règles de la Figure B.7 montrent des relations entre des sommets de départ et des temps. Par exemple, la règle de la Figure B.7a indique que l'événement pour lequel des arcs de départ des sommets 1 *et* 2 *vont à un même sommet se produit dans les temps* t_1 *et* t_2*. La règle de la Figure B.7b dit que la plupart des sommets d'arrivée des arcs partant au sommet* 3 *au temps* t_2 *sont aussi les sommets d'arrivée des arcs partant à ce sommet aux temps* t_3*,* t_4 *et* t_5*.*

(a) $\{d_1, d_2\} \rightarrow \{t_1, t_2\}$. (b) $\{d_3\} \times \{t_2\} \rightarrow \{t_3, t_4, t_5\}$.

Figure B.7: Des règles sur $\{D^1, D^3\}$ dans $\mathcal{R}_G$.



(a) $\{d_1\} \times \{a_4\} \rightarrow \{d_3, d_4\} \times \{a_1, a_3\}$. (b) $\{d_2, d_4\} \rightarrow (\{d_3\} \times \{a_1\}) \vee (\{d_1\} \times \{a_3\}) \vee (\{a_4\})$.

Figure B.8: Des règles sur $\{D^1, D^2\}$ dans $\mathcal{R}_G$.

Dans la Figure B.8, nous donnons des exemples de règles comprenant à la fois des sommets de départ et des sommets d'arrivée. La Figure B.8a décrit la dépendance entre des sous-graphes. Plus précisément, elle indique que le sous-graphe en prémisse de la règle peut être élargi à la clique en conclusion avec une confiance assez élevée. La règle de la Figure B.8b montre que si les arcs de départ des sommets 2 *et* 4 *convergent, ils convergeront vers les sommets* 1*,* 3 *ou* 4*.*

Dans le cadre d'une application à l'analyse des usages de vélos dans le système `Vélo'v` (système de Vélos en libre-service dans Grand Lyon), les expériences sont présentées dans Section 5.3. Nous montrons que les règles obtenues aident à mieux comprendre " Comment le système `Vélov'v` est utilisé". Cette compréhension est bienfaisante pour améliorer le service accompli et développer le système `Vélov'v`.