



Two decades of Pattern Mining eBISS2016, Tours

Outline



What is Pattern Mining?



What are the main principles?



What are the recent trends?

What is Pattern Mining?



20 years ago...

Mining Association Rules between Sets of Items in Large Databases

Rakesh Agrawal Tomasz Imielinski* Arun Swami

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95129

Abstract

We are given a large database of customer transactions. Each transaction consists of items purchased by a customer in a visit. We present an efficient algorithm that generates all significant association rules between items in the database. The algorithm incorporates buffer management and novel optimization and pruning techniques. We also present results of applying the algorithm to sales data obtained from a large retailing company, which shows the effectiveness of the algorithm.

1 Introduction

Consider a supermarket with a large collection of items. Typical business decisions that the management of the supermarket has to make include what to put on sale, how to design coupons, how to place merchandise on shelves in order to maximize the profit, etc. Analysis of past transaction data is a commonly used approach in order to improve the quality of such decisions. Until recently, however, only global data about the cumulative sales during some time period (a day, a week, a month, etc.) was available on the computer. Progress in bar-code technology has made it possible to store the so called basket data that stores items purchased on a per-transaction basis. Basket data type transactions do not necessarily consist of items bought together at the same point of time. It may consist of items bought by a customer over a period of time. Examples include monthly purchases by members of a book club or a music club.

Several organizations have collected massive amounts of such data. These data sets are usually stored

*Current address: Computer Science Department, Rutgers University, New Brunswick, NJ 08903

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
SIGMOD 9/93/Washington, DC, USA
© 1993 ACM 0-89791-593-5/93/0005-0207...\$1.50

on tertiary storage and are very slowly migrating to database systems. One of the main reasons for the limited success of database systems in this area is that current database systems do not provide necessary functionality for a user interested in taking advantage of this information.

This paper introduces the problem of "mining" a large collection of basket data type transactions for association rules between sets of items with some minimum specified confidence, and presents an efficient algorithm for this purpose. An example of such an association rule is the statement that 90% of transactions that purchase bread and butter also purchase milk. The antecedent of this rule consists of bread and butter and the consequent consists of milk alone. The number 90% is the confidence factor of the rule.

The work reported in this paper could be viewed as a step towards enhancing databases with functionalities to process queries such as (we have omitted the confidence factor specification):

- Find all rules that have "Diet Coke" as consequent. These rules may help plan what the store should do to boost the sale of Diet Coke.
- Find all rules that have "bagels" in the antecedent. These rules may help determine what products may be impacted if the store discontinues selling bagels.
- Find all rules that have "sausage" in the antecedent and "mustard" in the consequent. This query can be phrased alternatively as a request for the additional items that have to be sold together with sausage in order to make it highly likely that mustard will also be sold.
- Find all the rules relating items located on shelves *A* and *B* in the store. These rules may help shelf planning by determining if the sale of items on shelf *A* is related to the sale of items on shelf *B*.
- Find the "best" *k* rules that have "bagels" in the consequent. Here, "best" can be formulated in terms of the confidence factors of the rules, or in terms

*SIGMOD Conference 1993

New problem

Mining Association Rules between Sets of Items in Large Databases

We introduced the problem of mining association rules between sets of items in a large database of customer transactions. Each transaction consists of items purchased by a customer in a visit. We are interested in finding those rules that have:

- Minimum transactional support s — the union of items in the consequent and antecedent of the rule is present in a minimum of $s\%$ of transactions in the database.
- Minimum confidence c — at least $c\%$ of transactions in the database that satisfy the antecedent of the rule also satisfy the consequent of the rule.

monthly purchases by members of a book club or a music club. Several organizations have collected massive amounts of such data. These data sets are usually stored and "mustard" in the consequent. This query can be phrased alternatively as a request for the additional items that have to be sold together with sausage in order to make it highly likely that mustard will also

An example of such an association rule is the statement that 90% of transactions that purchase bread and butter also purchase milk. The antecedent of this rule consists of bread and butter and the consequent consists of milk alone. The number 90% is the confidence factor of the rule.

***Discovering all relevant association rules**

New solution

Mining Association Rules between Sets of Items in Large Databases

We introduced the problem of mining association rules between sets of items in a large database of customer transactions. Each transaction consists of items purchased by a customer in a visit. We are interested in finding those rules that have:

- Minimum transactional support s — the union of items in the consequent and antecedent of the rule is present in a minimum of $s\%$ of transactions in the database.
- Minimum confidence c — at least $c\%$ of transactions in the database that satisfy the antecedent of the rule also satisfy the consequent of the rule.

monthly purchases by members of a book club or a music club. Several organisations have collected massive amounts of data and "mustard" in the consequent. This query can be phrased alternatively as a request for the additional items that have to be sold together with sausage in

We solve this problem by decomposing it into two subproblems:

1. Finding all itemsets, called *large* itemsets, that are present in at least $s\%$ of transactions.
2. Generating from each large itemset, rules that use items from the large itemset.

***Enumerating all frequent itemsets**

Leonardo DiCaprio Plays the Same Character Over and Over

	Troubled Romantic	Rich	Dies	Hidding Secret
				
				
				
				
				

*Source: mic.com

Leonardo DiCaprio
Plays the Same
Character Over and
Over

Troubled Romantic Rich Dies Hidding Secret



Troubled romantic + Rich
(supp = 0.4)

*Itemset

Leonardo DiCaprio Plays the Same Character Over and Over

	Troubled Romantic	Rich	Dies	Hidding Secret
				
				
				
				
				

Troubled romantic \rightarrow Rich
(supp = 0.4 / conf = 0.5)

*Association rule

Leonardo DiCaprio Plays the Same Character Over and Over

	Troubled Romantic	Rich	Dies	Hidding Secret
				
				
				
				
				

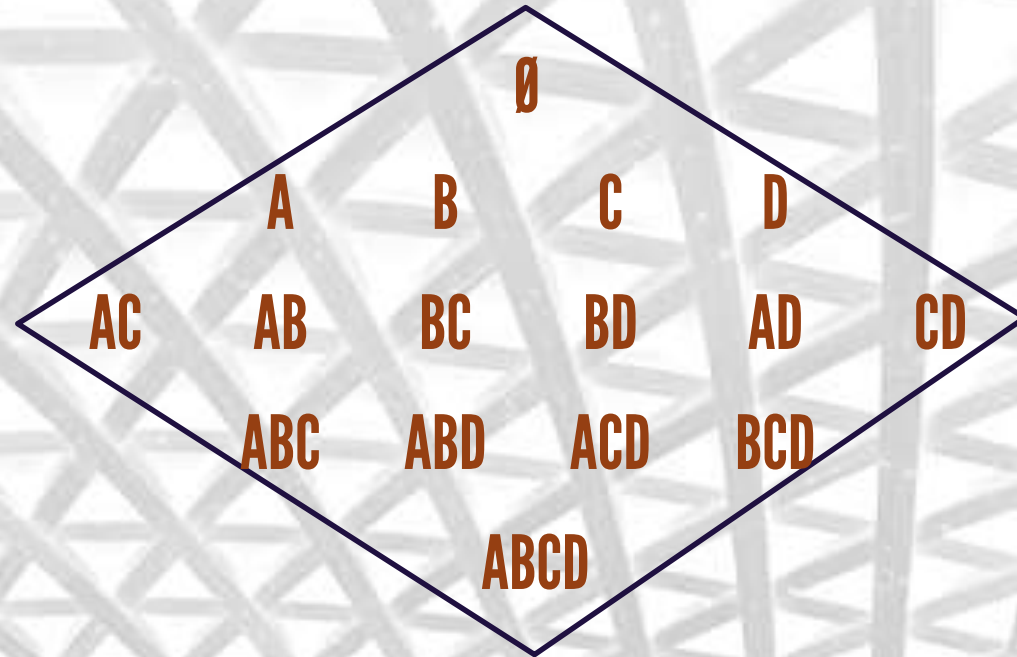
Troubled romantic → Rich
(supp = 0.4 / conf = 0.5)

Troubled romantic → Dies
(supp = 0.6 / conf = 0.75)

Leonardo DiCaprio Plays the Same Character Over and Over

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

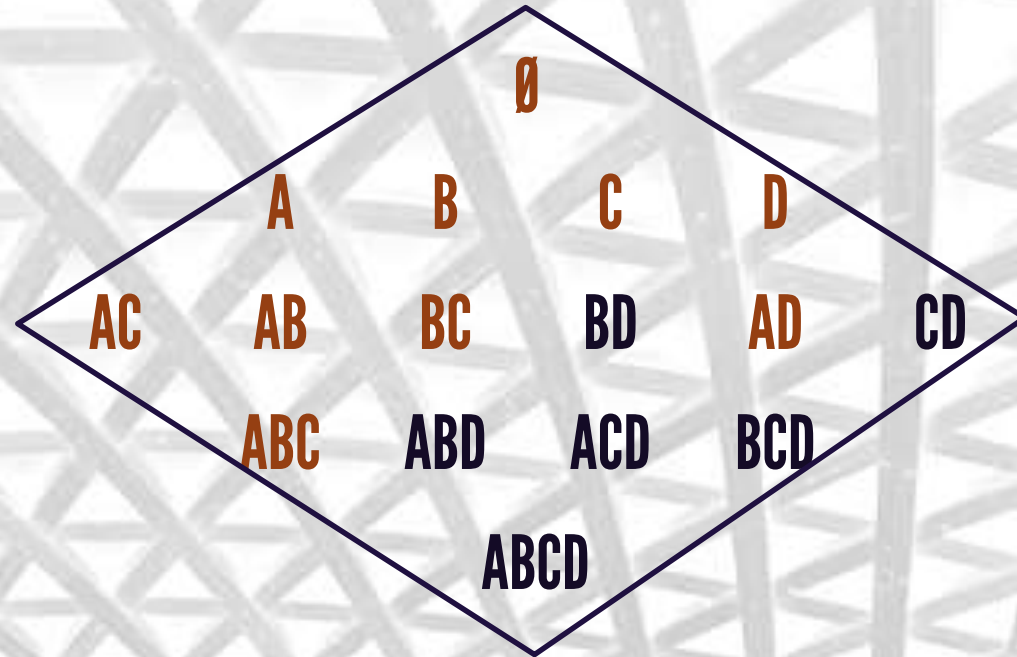
	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D



Frequent patterns

***Minimal frequency threshold = 1**

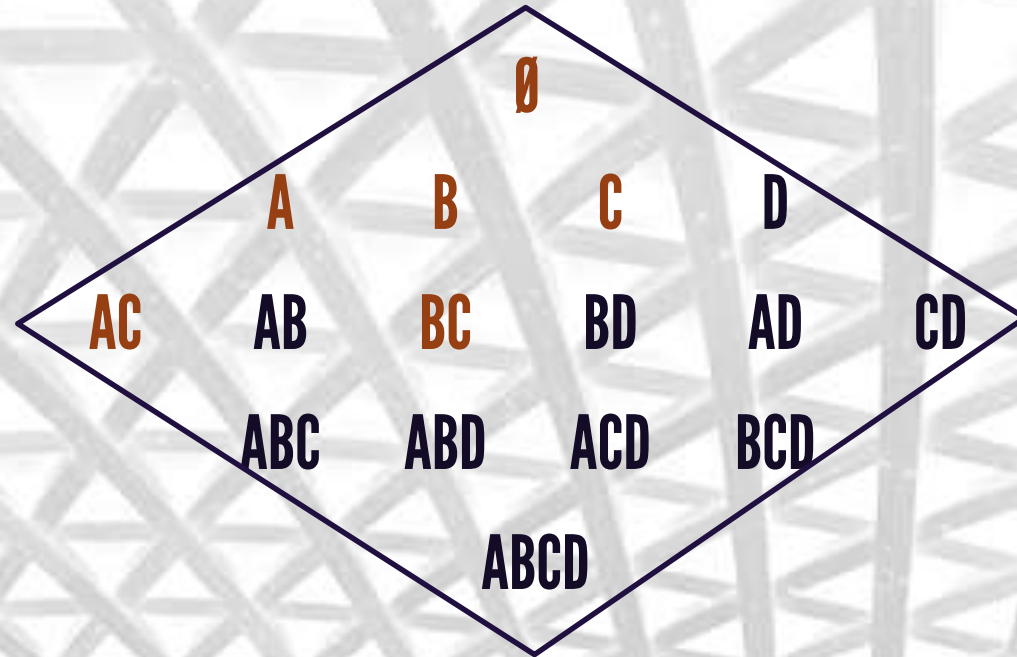
	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D



Frequent patterns

***Minimal frequency threshold = 2**

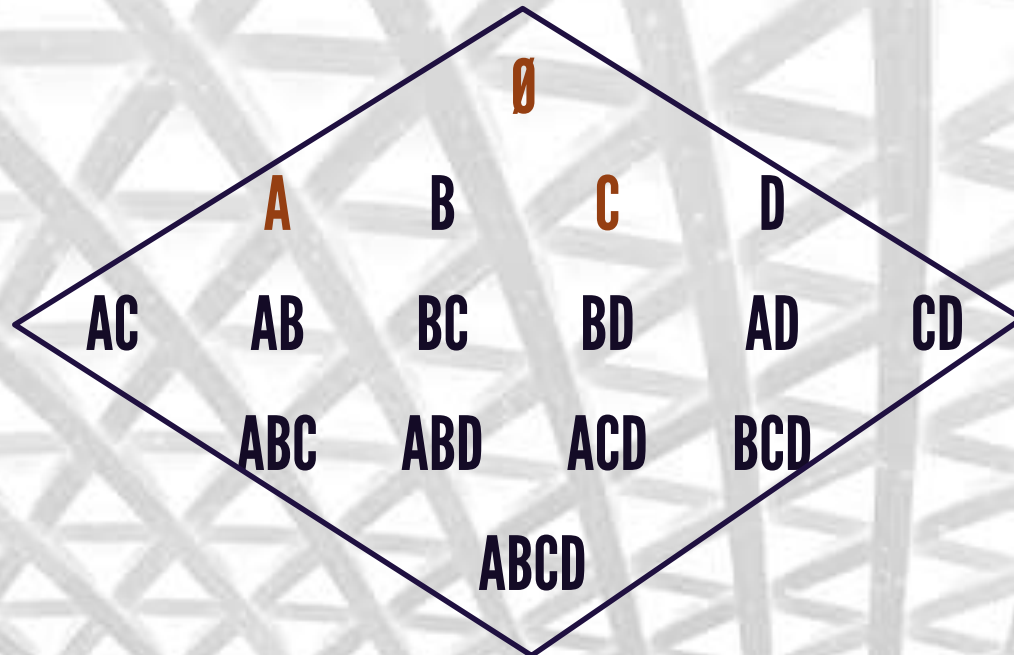
	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D



Frequent patterns

***Minimal frequency threshold = 3**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D



Frequent patterns

***Minimal frequency threshold = 4**



Exact solution

Exhaustive search

Speed of answer

Pattern Mining

***The footprint of databases**



Exact solution

Exhaustive search

Speed of answer

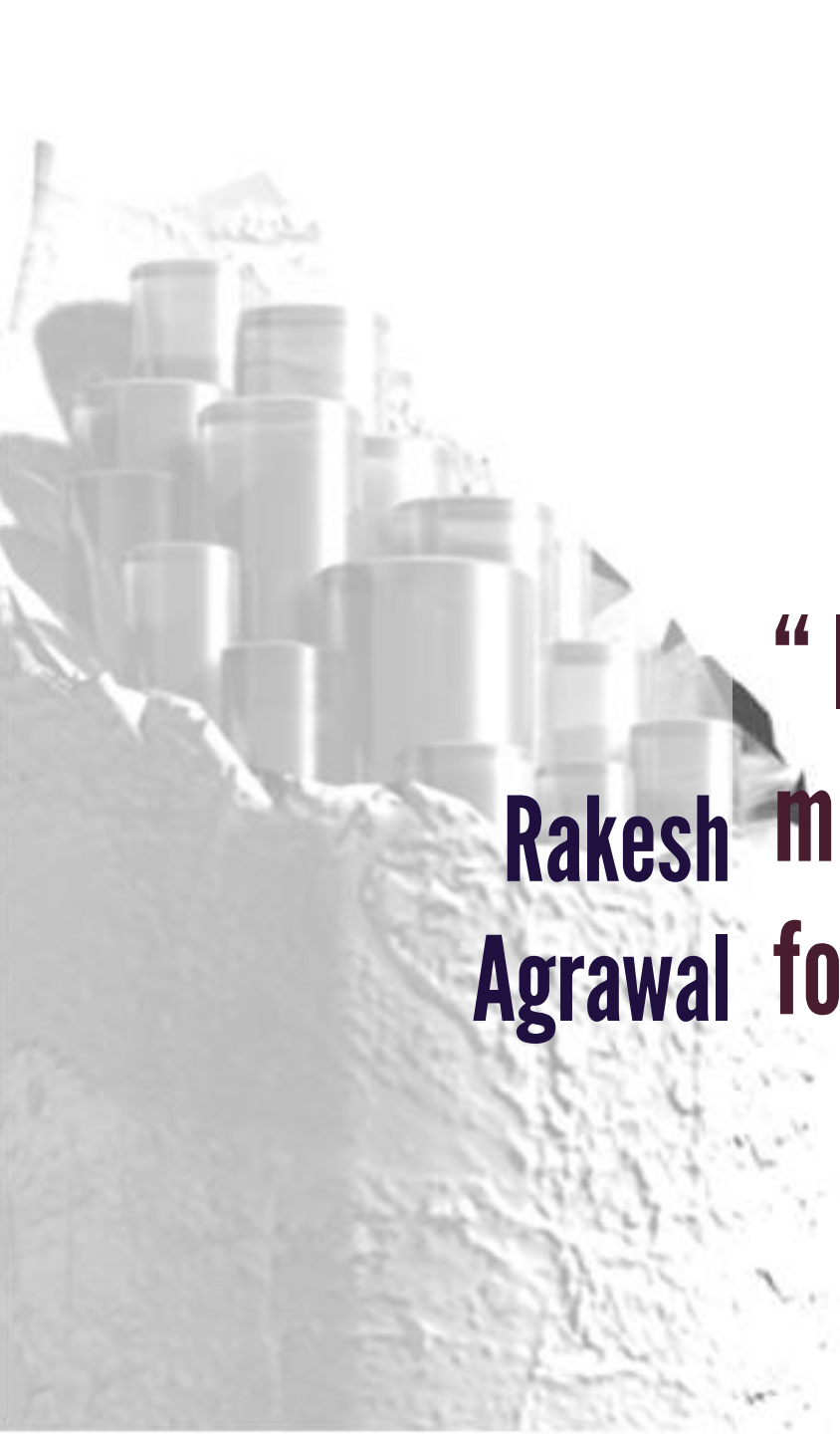
Approximate solution

Heuristic search

Quality of solution

Pattern Mining vs Artificial Intelligence

***The footprint of databases**



Rakesh Agrawal “ I’m a database person, so my view of data mining has been that it is essentially a richer form of querying.”

***The footprint of databases**

Impact of this seminal paper?

Mining Association Rules between Sets of Items in Large Databases

Rakesh Agrawal Tomasz Imielinski* Arun Swami

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95129

Abstract

We are given a large database of customer transactions. Each transaction consists of items purchased by a customer in a visit. We present an efficient algorithm that generates all significant association rules between items in the database. The algorithm incorporates buffer management and novel optimization and pruning techniques. We also present results of applying the algorithm to sales data obtained from a large retailing company, which shows the effectiveness of the algorithm.

1 Introduction

Consider a supermarket with a large collection of items. Typical business decisions that the management of the supermarket has to make include what to put on sale, how to design coupons, how to place merchandise on shelves in order to maximize the profit, etc. Analysis of past transaction data is a commonly used approach in order to improve the quality of such decisions. Until recently, however, only global data about the cumulative sales during some time period (a day, a week, a month, etc.) was available on the computer. Progress in bar-code technology has made it possible to store the so-called basket data that stores items purchased on a per-transaction basis. Basket data type transactions do not necessarily consist of items bought together at the same point of time. It may consist of items bought by a customer over a period of time. Examples include monthly purchases by members of a book club or a music club.

Several organizations have collected massive amounts of such data. These data sets are usually stored

*Current address: Computer Science Department, Rutgers University, New Brunswick, NJ 08903

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

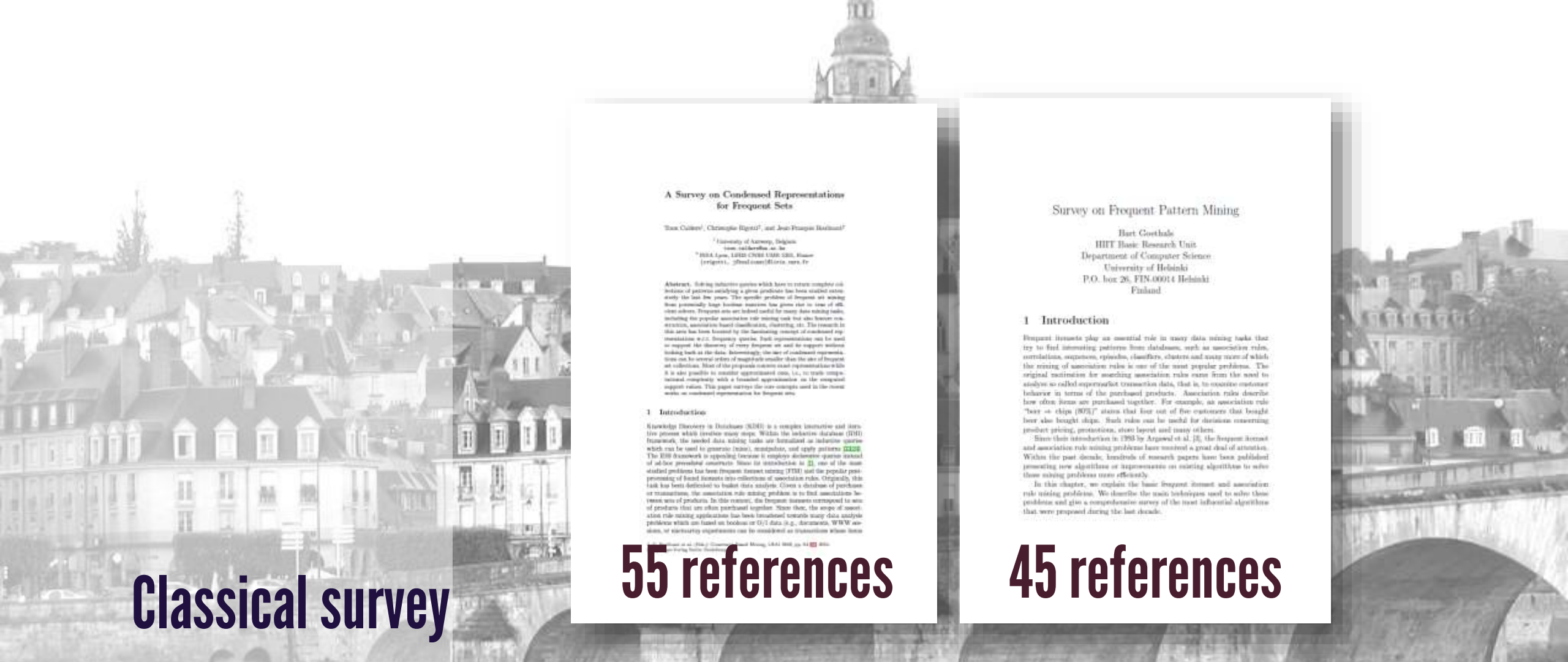
SIGMOD 9/93/Washington, DC, USA
© 1992 ACM 0-89791-593-5/93/0005-0207...\$1.50

on tertiary storage and are very slowly migrating to database systems. One of the main reasons for the limited success of database systems in this area is that current database systems do not provide necessary functionality for a user interested in taking advantage of this information.

This paper introduces the problem of "mining" a large collection of basket data type transactions for association rules between sets of items with some minimum specified confidence, and presents an efficient algorithm for this purpose. An example of such an association rule is the statement that 90% of transactions that purchase bread and butter also purchase milk. The antecedent of this rule consists of bread and butter and the consequent consists of milk alone. The number 90% is the confidence factor of the rule.

The work reported in this paper could be viewed as a step towards enhancing databases with functionalities to process queries such as (we have omitted the confidence factor specification):

- Find all rules that have "Diet Coke" as consequent. These rules may help plan what the store should do to boost the sale of Diet Coke.
- Find all rules that have "bagels" in the antecedent. These rules may help determine what products may be impacted if the store discontinues selling bagels.
- Find all rules that have "sausage" in the antecedent and "mustard" in the consequent. This query can be phrased alternatively as a request for the additional items that have to be sold together with sausage in order to make it highly likely that mustard will also be sold.
- Find all the rules relating items located on shelves *A* and *B* in the store. These rules may help shelf planning by determining if the sale of items on shelf *A* is related to the sale of items on shelf *B*.
- Find the "best" *k* rules that have "bagels" in the consequent. Here, "best" can be formulated in terms of the confidence factors of the rules, or in terms



Classical survey

A Survey on Condensed Representations for Frequent Sets

Thao Calmes¹, Christophe Rigotti², and Jean-François Bostain¹

¹University of Antwerp, Belgium
thao.calmes@ua.ac.be
²INRIA Lorraine, LORIA CNRS UMR 7503, France
rigotti@loria.fr, bostain@loria.fr

Abstract. Solving inductive queries which have to return complete sets instead of patterns satisfying a given predicate has been studied extensively the last few years. The specific problem of frequent set mining (more precisely large itemset mining) has given rise to a lot of efficient solutions. Frequent sets are indeed useful for many data mining tasks, including the popular association rule mining task but also feature construction, association-based classification, clustering, etc. The research in this area has been boosted by the fascinating concept of condensed representations w.r.t. frequency queries. Such representations can be used to support the discovery of every frequent set and to support various mining tasks on the data. Inevitably, the use of condensed representations can be several orders of magnitude smaller than the size of frequent set collections. Most of the proposed schemes used representations which it is also possible to construct approximately, i.e., to trade completeness with a bounded approximation on the consequent support ratios. This paper surveys the state-of-the-art in the construction of condensed representations for frequent sets.

1 Introduction

Knowledge Discovery in Databases (KDD) is a complex iterative and iterative process which involves many steps. Within the inductive database (IDB) framework, the needed data mining tasks are formalized as inductive queries which can be used to generate (rules), summaries, and query patterns [22,23]. The IDB framework is appealing because it employs declarative queries instead of ad hoc procedural concrete lines for construction in [2], one of the most studied problems has been frequent itemset mining (FIM) and the popular preprocessing of formal contexts (pre-collections of association rules). Originally, this task has been dedicated to basket data analysis. Given a database of products or transactions, the association rule mining problem is to find associations between sets of products. In this context, the frequent itemsets correspond to sets of products that are often purchased together. Since then, the scope of association rule mining applications has been broadened towards many data analysis problems which can be based on boolean or $\{0,1\}$ data (e.g., documents, WWW sessions, of laboratory experiments can be considered as transactions where items

55 references

Survey on Frequent Pattern Mining

Bart Goethals
HIT Basic Research Unit
Department of Computer Science
University of Helsinki
P.O. box 26, FIN-00014 Helsinki
Finland

1 Introduction

Frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers, clusters and many more of which the mining of association rules is one of the most popular problems. The original motivation for searching association rules came from the need to analyze so called supermarket transaction data, that is, to describe customer behavior in terms of the purchased products. Association rules describe how often items are purchased together. For example, an association rule "beer \rightarrow chips (80%)" states that four out of five customers that bought beer also bought chips. Such rules can be useful for decisions concerning product pricing, promotion, store layout and many others. Since their introduction in 1983 by Agrawal et al. [1], the frequent itemset and association rule mining problems have received a great deal of attention. Within the past decade, hundreds of research papers have been published presenting new algorithms or improvements on existing algorithms to solve these mining problems more efficiently. In this chapter, we explain the basic frequent itemset and association rule mining problems. We describe the main techniques used to solve these problems and give a comprehensive survey of the most influential algorithms that were proposed during the last decade.

45 references

***Dozens of references**



Classical survey

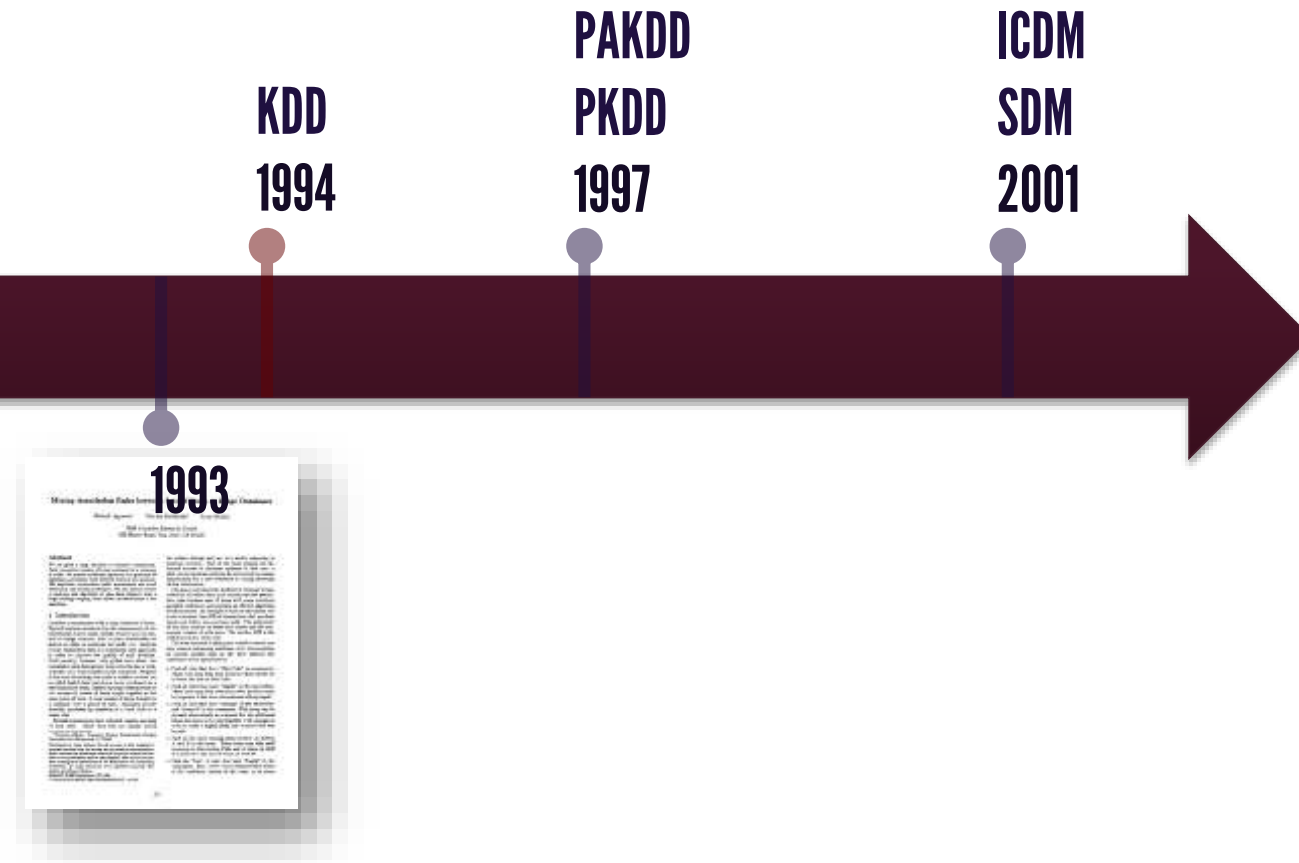
***Dozens of references**



Bibliometric survey

***Thousands of references**

Materials



***Data mining conferences ranked A**

Materials



Fast Algorithms for Mining Association Rules

Rakesh Agrawal Ramakrishnan Srikant*

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

Abstract

We consider the problem of discovering association rules between items in a large database of sales transactions. We present two new algorithms for solving this problem that are fundamentally different from the known algorithms. Empirical evaluation shows that these algorithms outperform the known algorithms by factors ranging from three for small problems to more than an order of magnitude for large problems. We also show how the best features of the two proposed algorithms can be combined into a hybrid algorithm, called AprioriHybrid. Scale-up experiments show that AprioriHybrid scales linearly with the number of transactions. AprioriHybrid also has excellent scale-up properties with respect to the transaction size and the number of items in the database.

1 Introduction

Progress in bar-code technology has made it possible for retail organizations to collect and store massive amounts of sales data, referred to as the *basket* data. A record in such data typically consists of the transaction date and the items bought in the transaction. Successful organizations view such databases as important pieces of the marketing infrastructure. They are interested in instituting information-driven marketing processes, managed by database technology, that enable marketers to develop and implement customized marketing programs and strategies [6].

The problem of mining association rules over basket data was introduced in [4]. An example of such a rule might be that 98% of customers that purchase

tires and auto accessories also get automotive services done. Finding all such rules is valuable for cross-marketing and attached mailing applications. Other applications include catalog design, add-on sales, store layout, and customer segmentation based on buying patterns. The databases involved in these applications are very large. It is imperative, therefore, to have fast algorithms for this task.

The following is a formal statement of the problem [4]: Let $I = \{I_1, I_2, \dots, I_m\}$ be a set of literals, called items. Let \mathcal{D} be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. Associated with each transaction is a unique identifier, called its *TID*. We say that a transaction T contains X , a set of some items in I , if $X \subseteq T$. An *association rule* is an implication of the form $X \Rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ holds in the transaction set \mathcal{D} with *confidence* c if $c\%$ of transactions in \mathcal{D} that contain X also contain Y . The rule $X \Rightarrow Y$ has *support* s in the transaction set \mathcal{D} if $s\%$ of transactions in \mathcal{D} contain $X \cup Y$. Our rules are somewhat more general than in [4] in that we allow a consequent to have more than one item.

Given a set of transactions \mathcal{D} , the problem of mining association rules is to generate all association rules that have support and confidence greater than the user-specified minimum support (called *minsup*) and minimum confidence (called *minconf*) respectively. Our discussion is neutral with respect to the representation of \mathcal{D} . For example, \mathcal{D} could be a data file, a relational table, or the result of a relational expression.

An algorithm for finding all association rules, called the *Apriori* algorithm, is presented in [4]. In this paper, we present two new algorithms, *Apriori* and *AprioriHybrid*, that differ fundamentally from these algorithms. We present experimental results showing

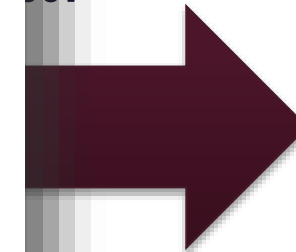
*Visiting from the Department of Computer Science, University of Wisconsin, Madison.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct or indirect commercial advantage, the VLDB copyright notice and the title of the publication, the name of the publisher, and the name of the author(s) appear on the copies.

Proceedings of the 20th VLDB Conference
Santiago, Chile, 1994

[Agrawal and Srikant, 1994]

DM
DM
001



*Not consider VLDB, CIKM, ICDE,...

Materials

KDI
199



Levelwise Search and Borders of Theories in Knowledge Discovery

HEIKKI MANNILA
HANNU TOIVONEN

Department of Computer Science, P.O. Box 26, FIN-00014 University of Helsinki, Finland

heikki.mannila@cs.helsinki.fi
hannu.toivonen@cs.helsinki.fi

Editor: Usama Fayyad

Received January 22, 1997; Revised July 10, 1997; Accepted July 11, 1997

Abstract. One of the basic problems in knowledge discovery in databases (KDD) is the following: given a data set r , a class \mathcal{L} of sentences for defining subgroups of r , and a selection predicate, find all sentences of \mathcal{L} deemed interesting by the selection predicate. We analyze the simple levelwise algorithm for finding all such descriptions. We give bounds for the number of database accesses that the algorithm makes. For this, we introduce the concept of the borders of a theory, a notion that turns out to be surprisingly powerful in analyzing the algorithm. We also consider the verification problem of a KDD process: given r and a set of sentences $S \subseteq \mathcal{L}$, determine whether S is exactly the set of interesting sentences about r . We show strong connections between the verification problem and the hypergraph transversal problem. The verification problem arises in a natural way when using sampling to speed up the generic discovery step in KDD.

Keywords: theory of knowledge discovery, association rules, episodes, integrity constraints, hypergraph transversals

1. Introduction

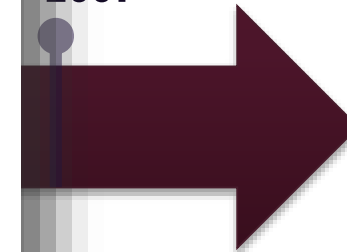
Knowledge discovery in databases (KDD), also called data mining, has recently received wide attention from practitioners and researchers. There are several attractive application areas for KDD, and it seems that techniques from machine learning, statistics, and databases can be profitably combined to obtain useful methods and systems for KDD. See, e.g., Fayyad et al. (1996), and Piatetsky-Shapiro and Frawley (1991) for general descriptions of the area.

The KDD area is and should be largely guided by (successful) applications. Still, theoretical work in the area is needed. In this paper we take some steps towards theoretical KDD. We consider a KDD process in which the analyzer first produces lots of potentially interesting rules, subgroup descriptions, patterns, etc., and then interactively selects the truly interesting ones from these. In this paper we analyze the first stage of this process: how to find all the potentially interesting rules in the database.

The problem we consider is the following: given a data set r and a class \mathcal{L} of sentences for defining subgroups of r , find all sentences of \mathcal{L} deemed interesting by the selection predicate.

if ψ then θ .

ICDM
SDM
2001



[Manila and Toivonen, 1997]

*Not consider DMKD, TKDE, KAIS,...

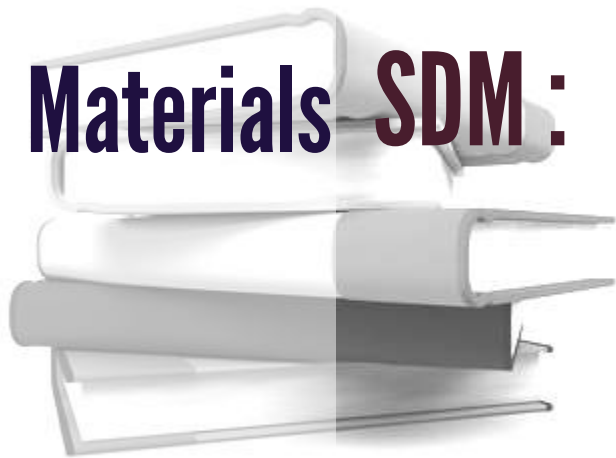
KDD : 1,905 since 1995

PKDD : 1,295 since 1997

PAKDD : 1,277 since 1998

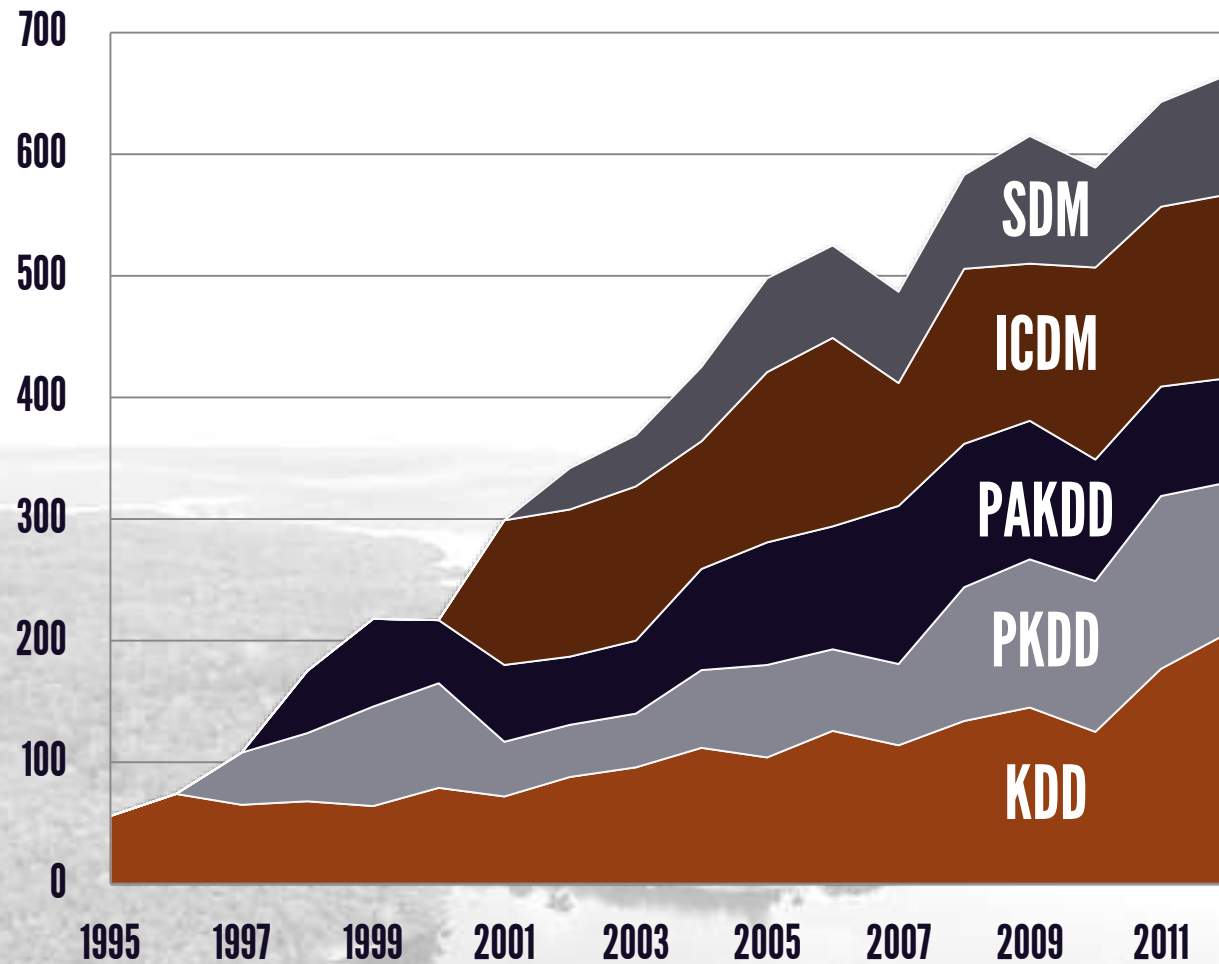
ICDM : 1,598 since 2001

Materials SDM : 813 since 2002



***6,888 publications from DBLP (1995-2012)**

**Faster,
Higher, Stronger**



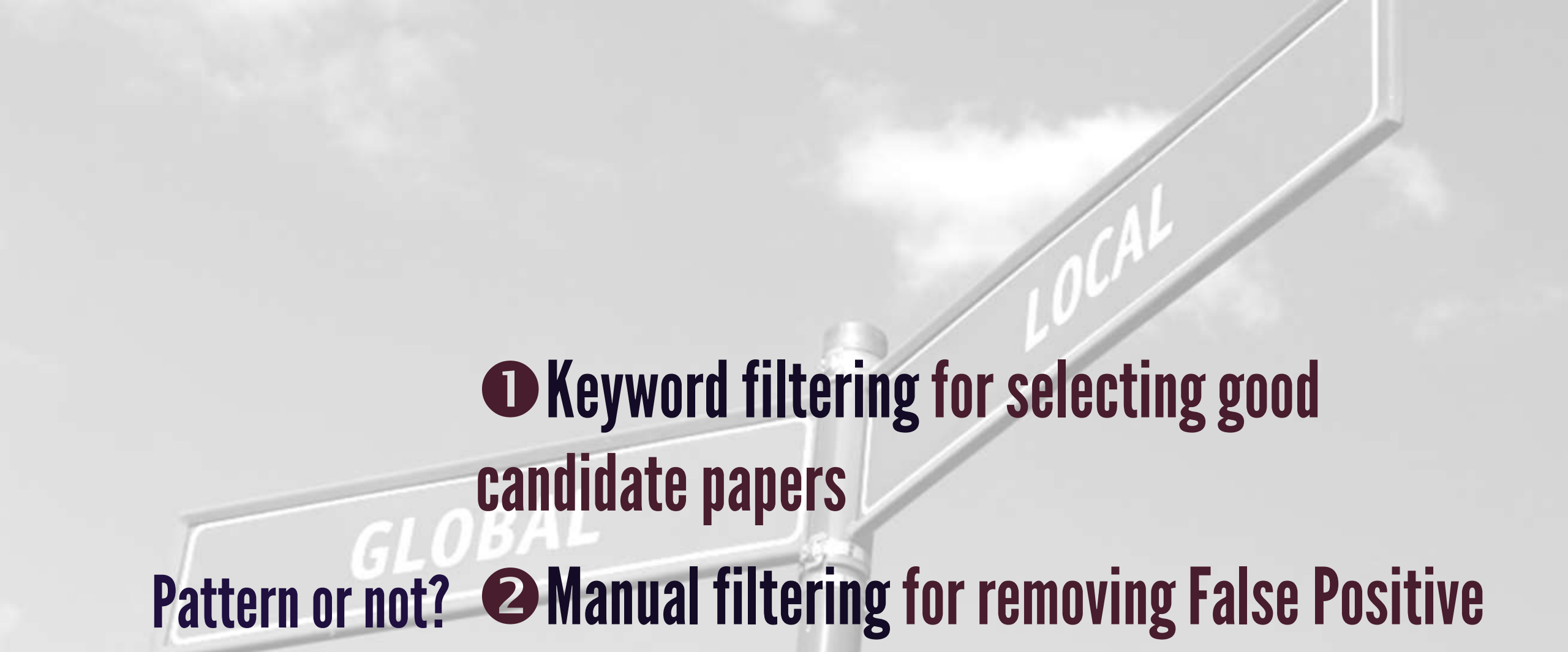


***Titles only**



Pattern or not? Fuzzy limit

***Discovery and use**



① Keyword filtering for selecting good candidate papers

Pattern or not? ② Manual filtering for removing False Positive

***Semi-automated topic assignment**

Language

Constraint

Pattern or not? Condensed Representation

***Dimensions of Pattern Mining**

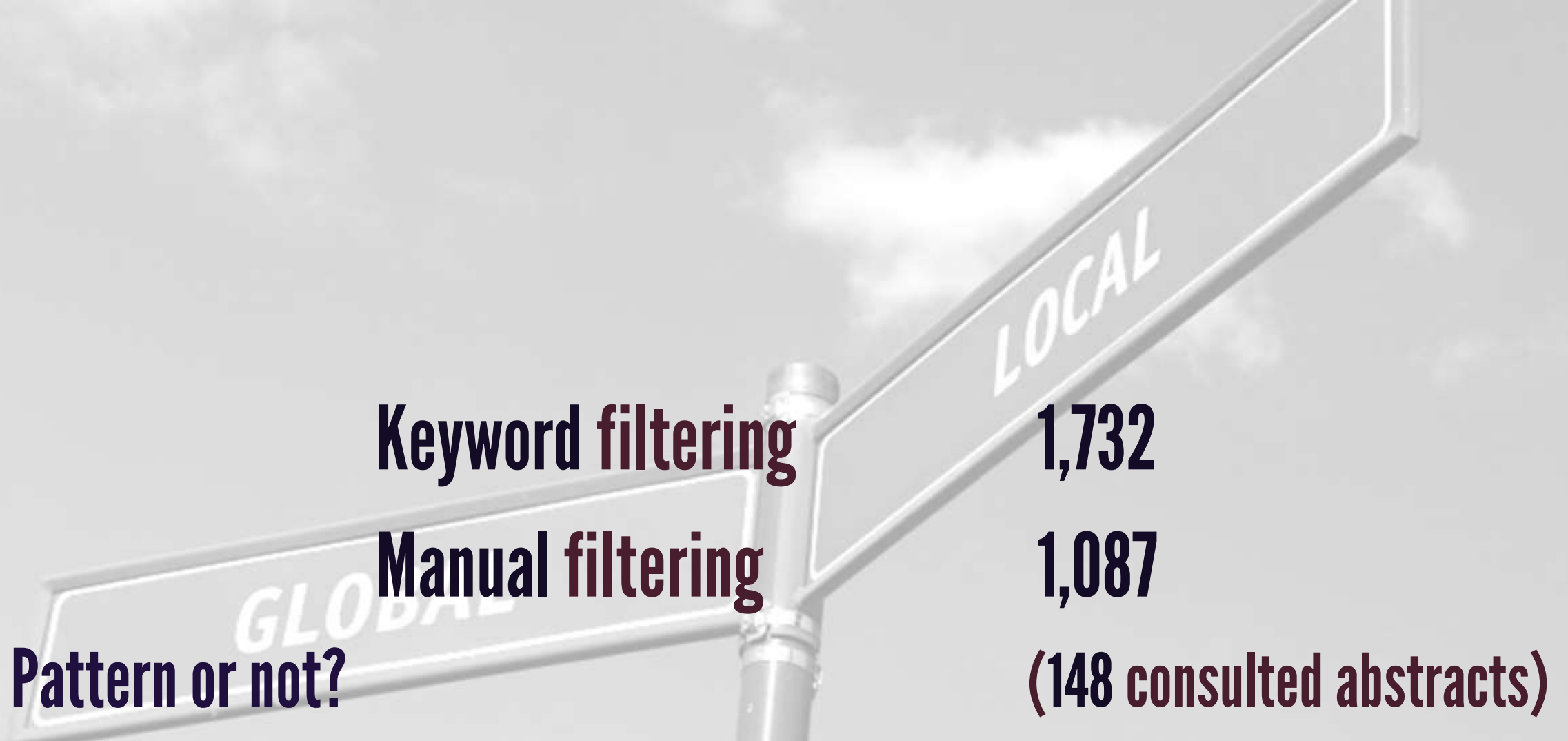


**pattern, item, sequence, rule, tree, graph, string,
stream, subgroup...**

**support (no Vector Machine), frequent,
monotone...**

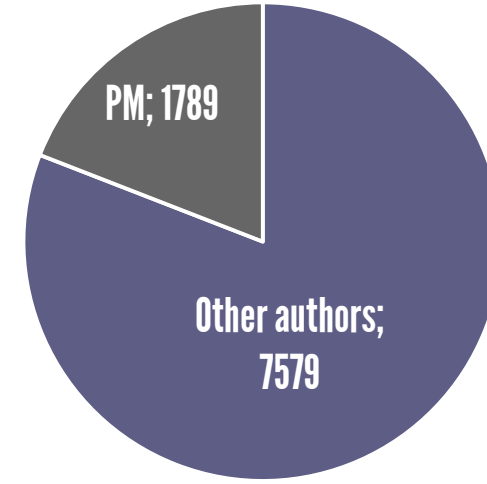
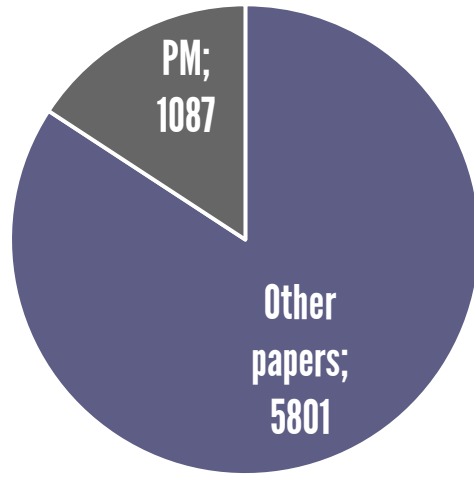
Pattern or not? free, generator, closed, condensed, concise

***Keywords of Pattern Mining**



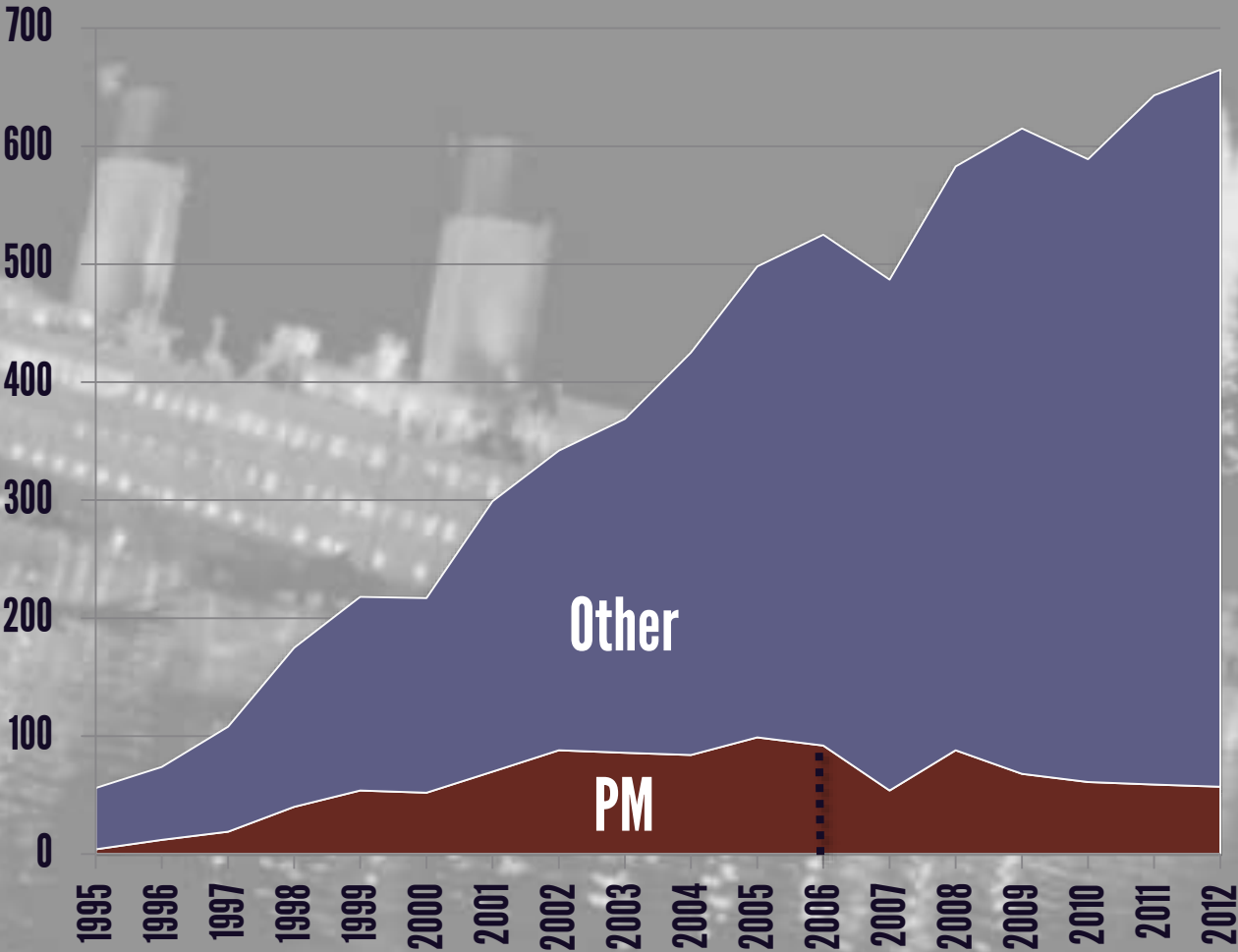
***5% of False Negative, around 258 papers**

**1/5th of authors and
1/6th of KDD
publications**



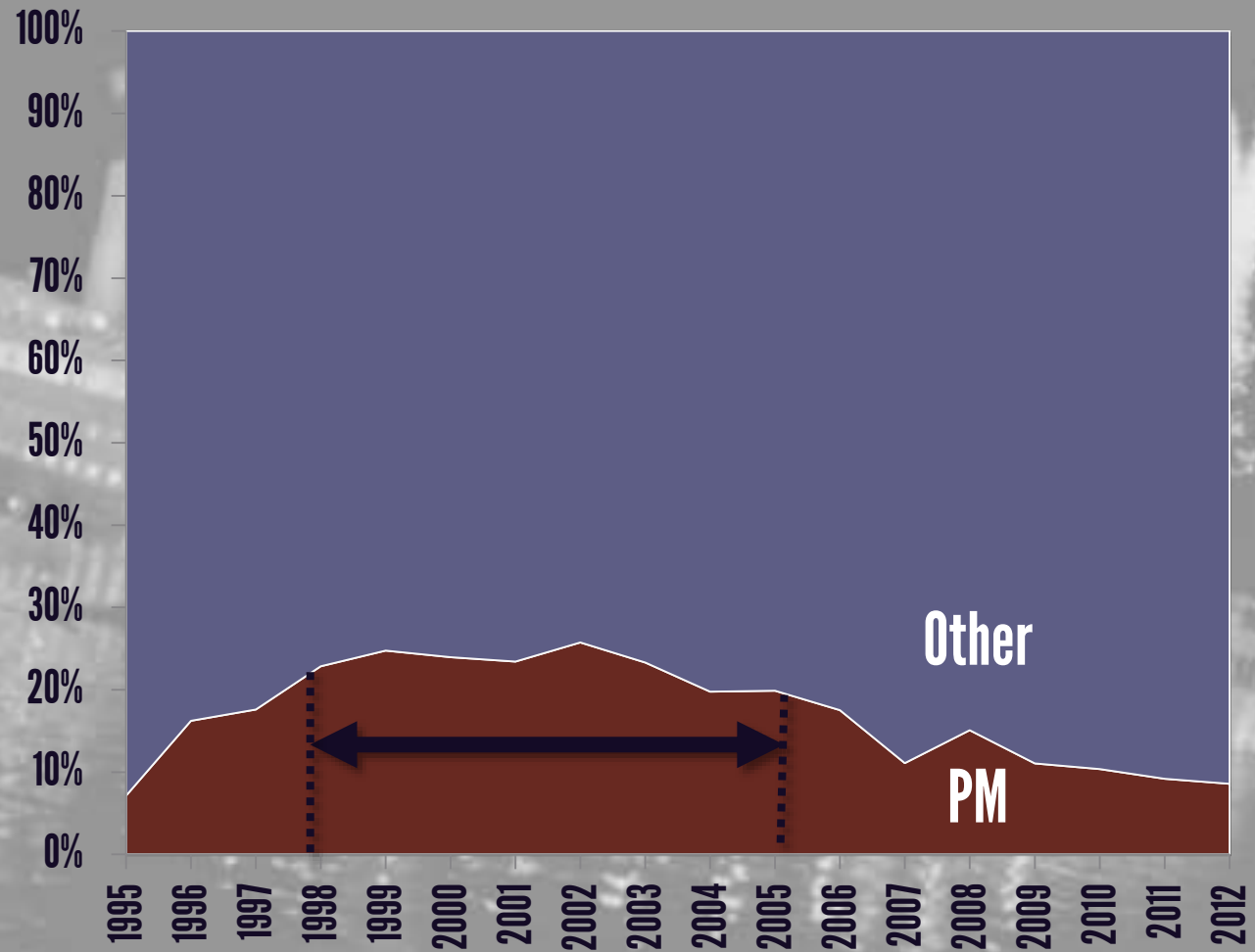
***PM is a true subfield of KDD**

The slowdown of Pattern Mining



***Turning point: 2006**

The slowdown of Pattern Mining



***Golden age: 1998-2005 (1 paper out of 5)**

What are the main principles of Pattern Mining?

rule **pattern** **frequent** **association** **mining** **itemset**

database **discovery** **data** **graph** **algorithm** **sequence** **efficient** **classification** **constraint** **set** **closed** **tree** **learning** **temporal** **stream** **subgraph** **approach** **model** **discovering** **time** **efficient** **interesting** **support** **representation** **measure** **energy** **large** **event** **site** **detector** **last** **iteration** **model** **graph** **model** **constraint** **classification** **sequence** **efficient** **interesting** **support** **representation** **measure** **energy** **large** **event** **site** **detector** **last** **iteration**

Typology of publications

Data Mining and Knowledge Discovery 1, 241-259 (1997)
© 1997 Kluwer Academic Publishers. Manufactured in The Netherlands.

Levelwise Search and Borders of Theories in Knowledge Discovery

HEIKKI MANNILA
HANSU TOIVONEN

heikki.mannila@cs.helsinki.fi
hansu.toivonen@cs.helsinki.fi

Department of Computer Science, P.O. Box 26, FIN-00014 University of Helsinki, Finland

Editor: Usama Fayyad

Received January 23, 1997; Revised July 23, 1997; Accepted July 11, 1997

Abstract. One of the basic problems in knowledge discovery in databases (KDD) is the following: given a data set r , a class C of sentences for defining subgroups of r , and a selection predicate, find all sentences of C deemed interesting by the selection predicate. We analyze the simple levelwise algorithm for finding all such descriptions. We give bounds for the number of database accesses that the algorithm makes. For this, we introduce the concept of the border of a theory, a notion that turns out to be surprisingly powerful in analyzing the algorithm. We also consider the verification problem of a KDD process: given r and a set of sentences $S \subseteq C$, determine whether S is exactly the set of interesting statements about r . We show strong connections between the verification problem and the hypergraph isomorphism problem. The verification problem arises in a natural way when using sampling to speed up the search discovery step in KDD.

Keywords: theory of knowledge discovery, association rules, apriori, integrity constraints, hypergraph isomorphism

1. Introduction

Knowledge discovery in databases (KDD), also called data mining, has recently received wide attention from practitioners and researchers. There are several attractive application areas for KDD, and it seems that techniques from machine learning, statistics, and databases can be profitably combined to obtain useful methods and systems for KDD. See, e.g., Fayyad et al. (1996), and Piatetsky-Shapiro and Frawley (1991) for general descriptions of the area.

The KDD area is and should be largely guided by (successful) applications. Still, theoretical work in the area is needed. In this paper we take some steps towards theoretical KDD. We consider a KDD process in which the analyzer first produces lots of potentially interesting rules, subgroup descriptions, patterns, etc., and then interactively selects the truly interesting ones from these. In this paper we analyze the first stage of this process: how to find all the potentially interesting rules in the database.

The intuitive idea behind this work is as follows. A lot of work in data mining can be formulated in terms of finding all rules of the form

if ψ then θ .

- 1 Language
- 2 Constraint
- 3 Condensed Representation (CR)

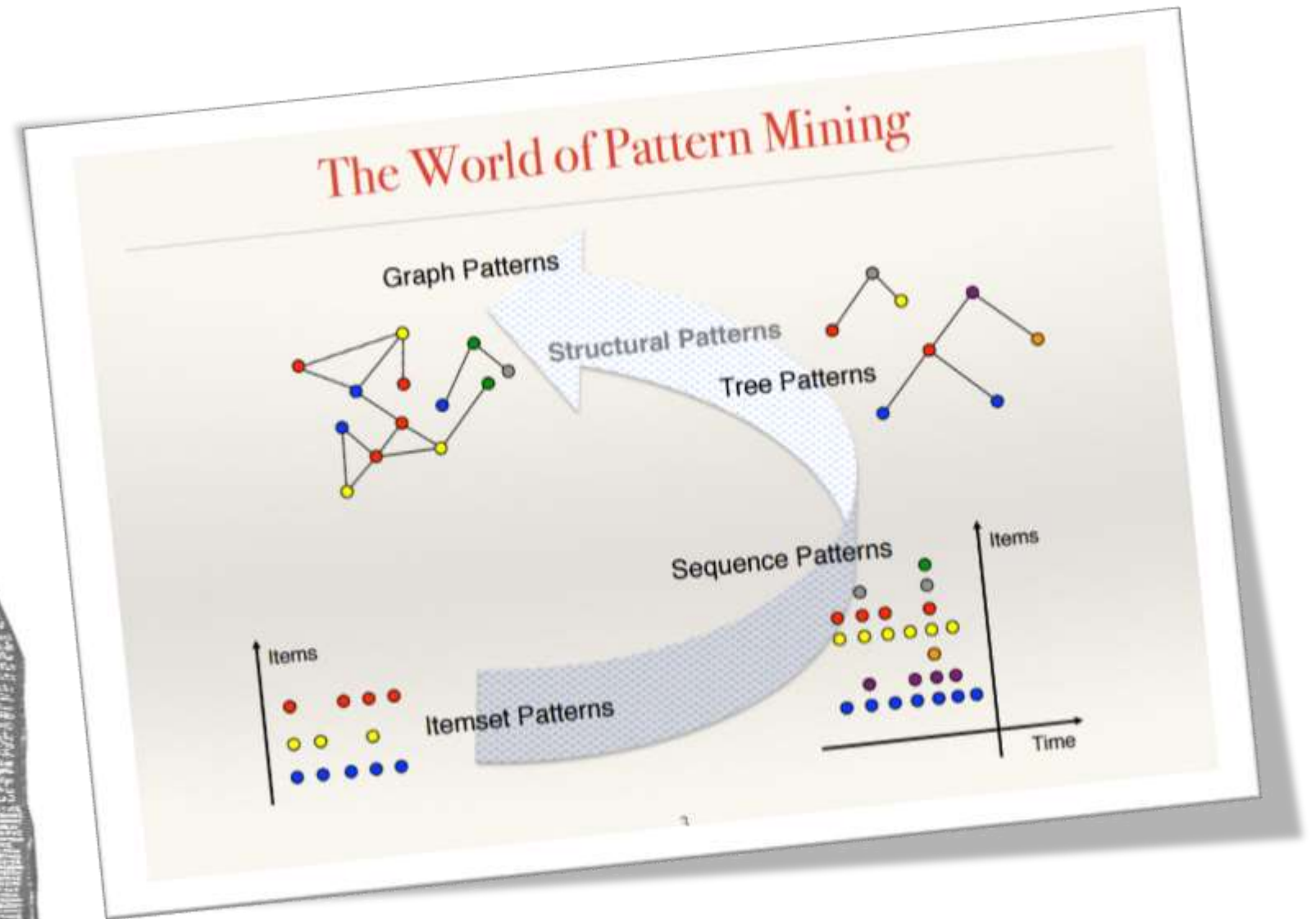
*Levelwise Search and Borders of Theories in Knowledge Discovery

[Mannila and Toivonen, 1997]

1



Language?



***Do you remember?**

1

Language?

The World of Pattern Mining

Items, itemsets and datasets

- **Items = a set of distinct literals**
 - Example: 🍷 🍺 🍻 🍾 🍹
 - Often denoted by a letter of alphabet
- **Itemset = any set of items**
 - Example: {🍷 🍺 🍻}
 - The whole set of itemsets is called the **language**
- **Dataset = multi-set of itemsets**
 - Example:

🍷	🍺	🍻	🍾	🍹
🍷	🍺	🍻	🍾	🍹
🍷	🍺	🍻	🍾	🍹
🍷	🍺	🍻	🍾	🍹

***Do you remember?**

1

Language?

The World of Pattern Mining

Items, itemsets and lists

Formal Notations of Sequence Data

- ◆ An *item* is denoted by lower-case letters

a, b, c, d, \dots

- ◆ An *itemset* is denoted by upper-case letters

$I = (abc), I' = (acd), I_1 = (bcef), \dots$

- ◆ A *sequence* is denoted by s (with prime and/or indice)

$s = \langle I_1 I_2 I_3 \rangle, s' = \langle (ab)(c)(bcd)(ac) \rangle, s_1 = \langle (ab)c(bc) \rangle \dots$

(We may ignore the parentheses for 1-item itemsets)

- ◆ A *sequence database* is a (large) set of sequences

ber?

8 itemsets

$\emptyset, A, B, C, AB, AC, BC, ABC$

80 sequential patterns

$\emptyset, \langle A \rangle, \langle AA \rangle, \langle AAA \rangle, \langle AAB \rangle, \langle AAC \rangle, \dots$

**Language
sophistication with
3 items and 3 as
maximal length**

238 subgraphs patterns

$\emptyset, A, AA, A-A, AAA, A-AA, A-A-A, \dots$

***Pattern explosion**

A black and white photograph of a nuclear mushroom cloud. The cloud is massive, with a thick, billowing base that spreads across the horizon. The top of the cloud is a dense, rounded cap that rises into the sky. The background shows a calm body of water in the foreground and a distant shoreline with some structures. The overall tone is somber and dramatic.

Pattern explosion

**Computational
challenges of
language
sophistication**

Pattern matching

Subgraph isomorphism checking



Pattern explosion

**Computational
challenges of
language
sophistication**

Pattern matching*

Subgraph isomorphism checking

***Does a database entry contain a pattern?**



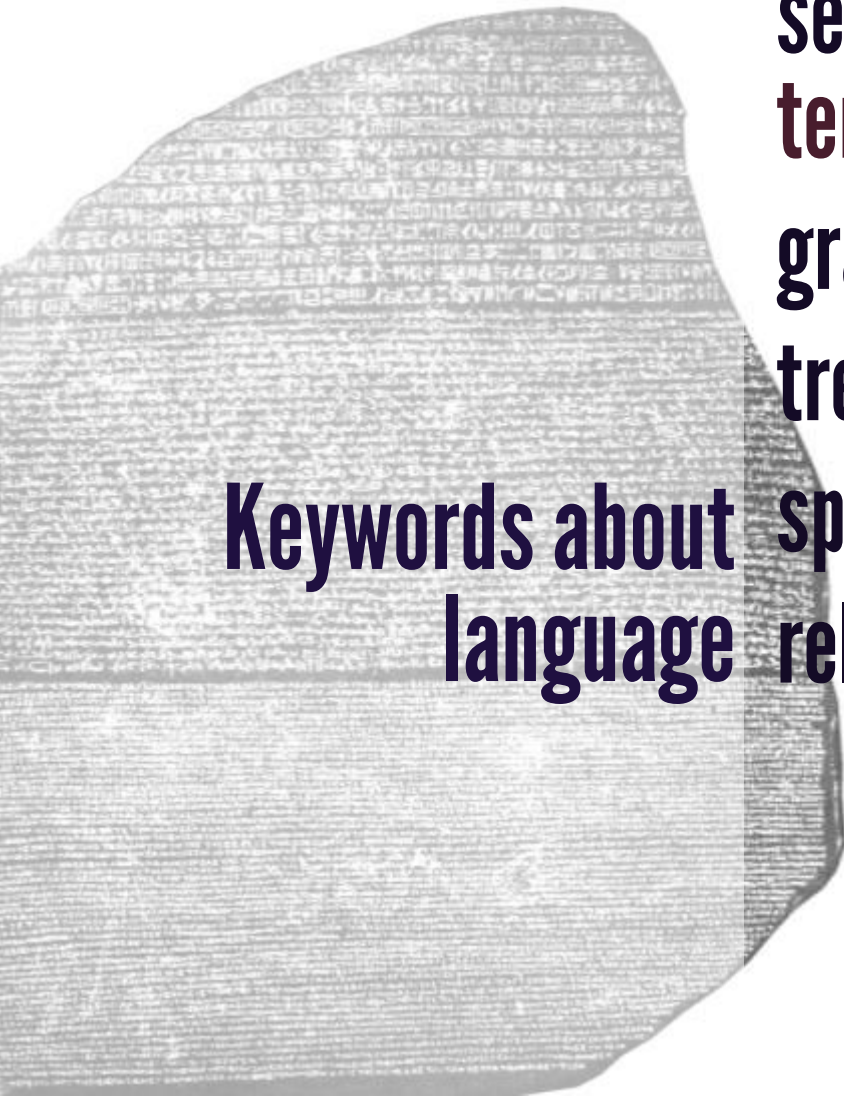
Pattern explosion

**Computational
challenges of
language
sophistication**

Pattern matching

Subgraph isomorphism checking*

***Does a graph contain a subgraph isomorphic to another graph?**



**Keywords about
language**

itemset, set

rule, association

**sequence, episode, string, stream, protein, periodic,
temporal**

graph, molecular, structure, network

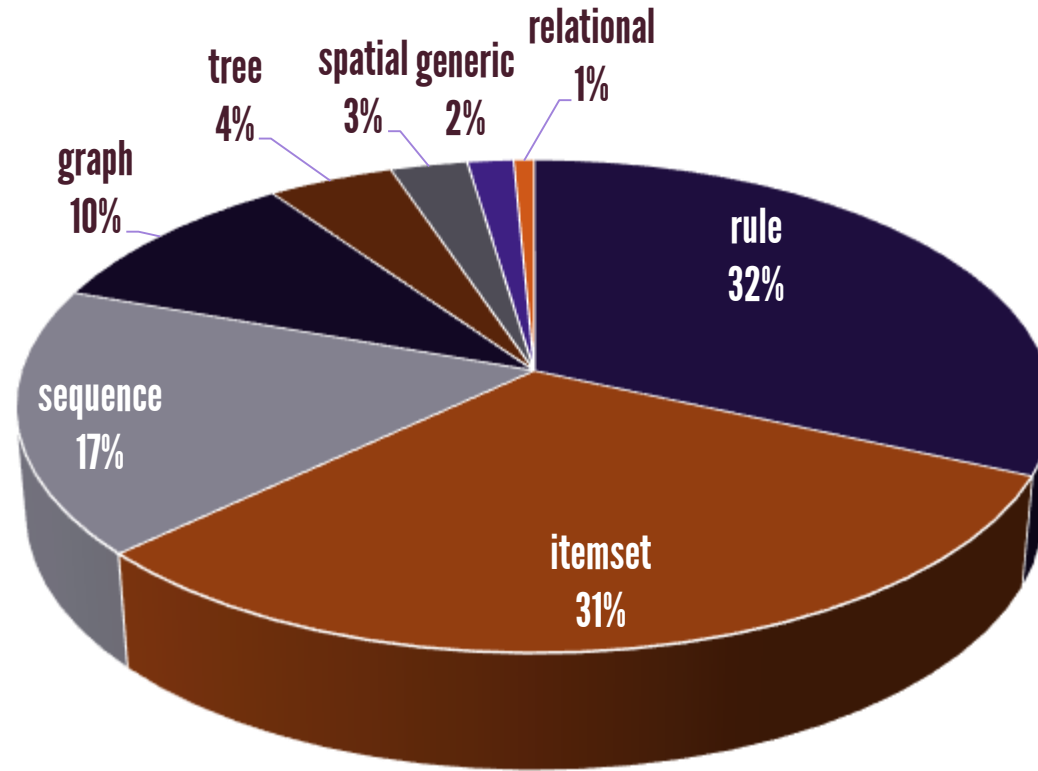
tree, xml

spatial, spatio-temporal

relational

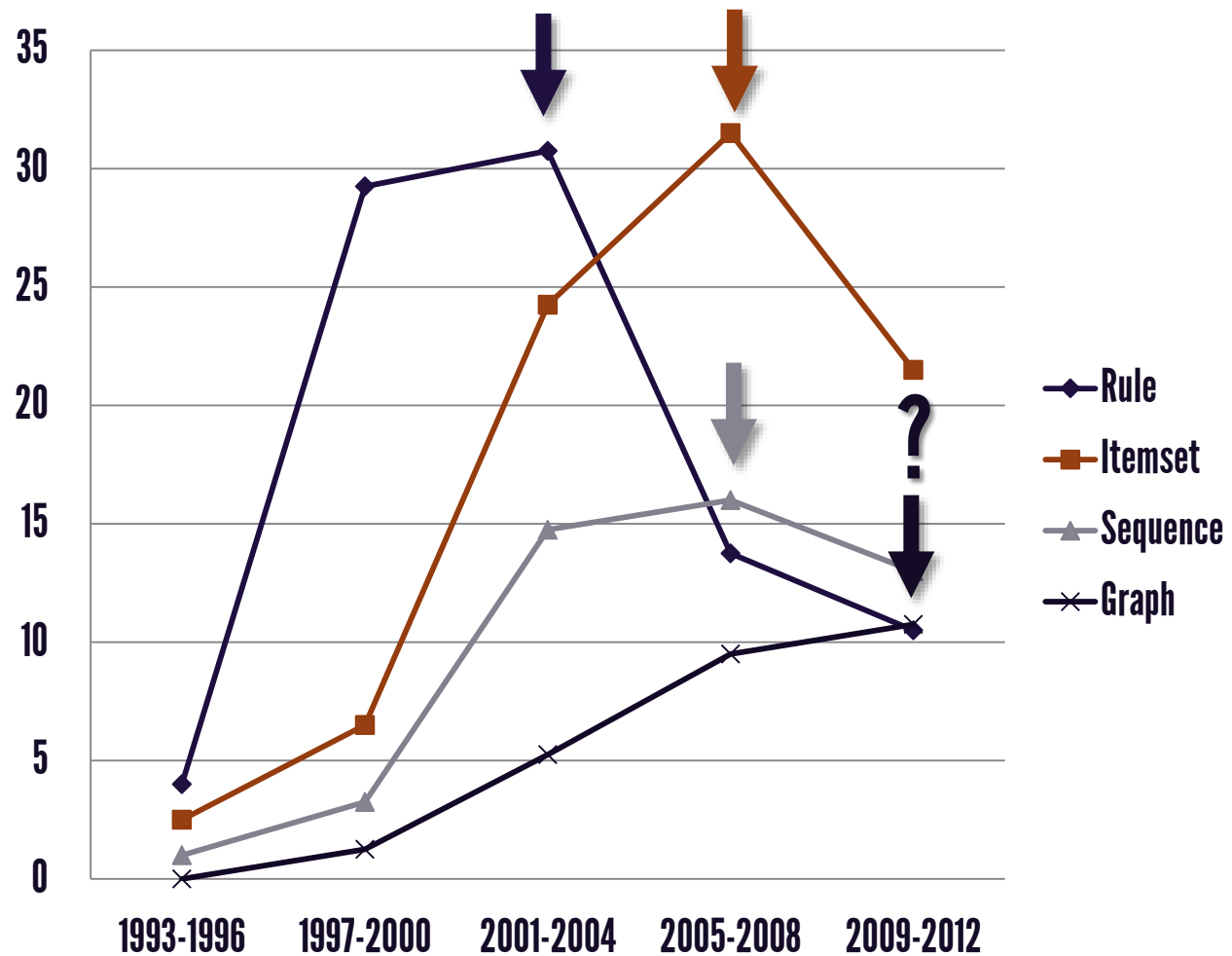
***Semi-automated topic assignment**

**Itemsets or rules for
63% of publications**



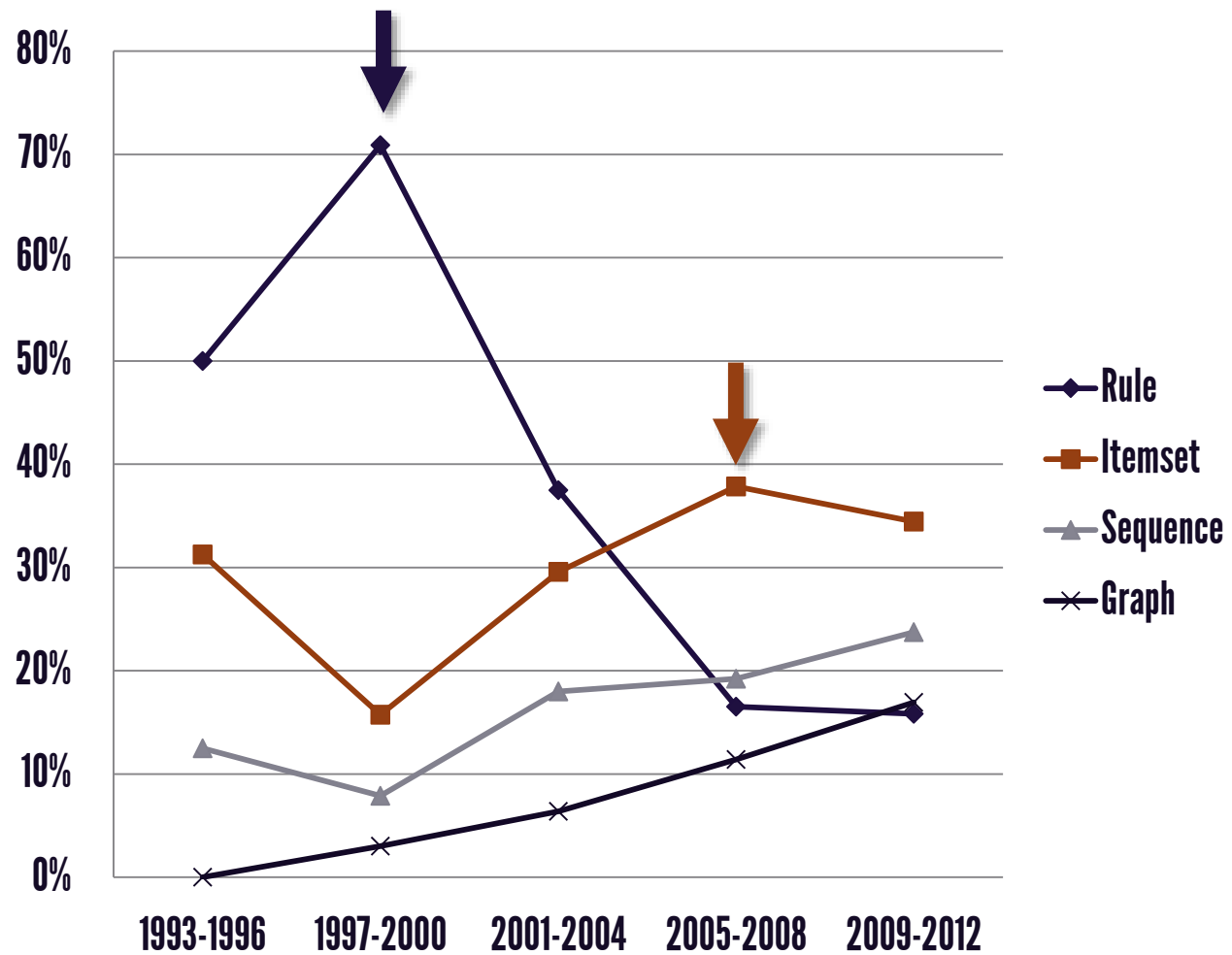
***Only 18 papers with generic language**

Sophistication and marginalization of languages

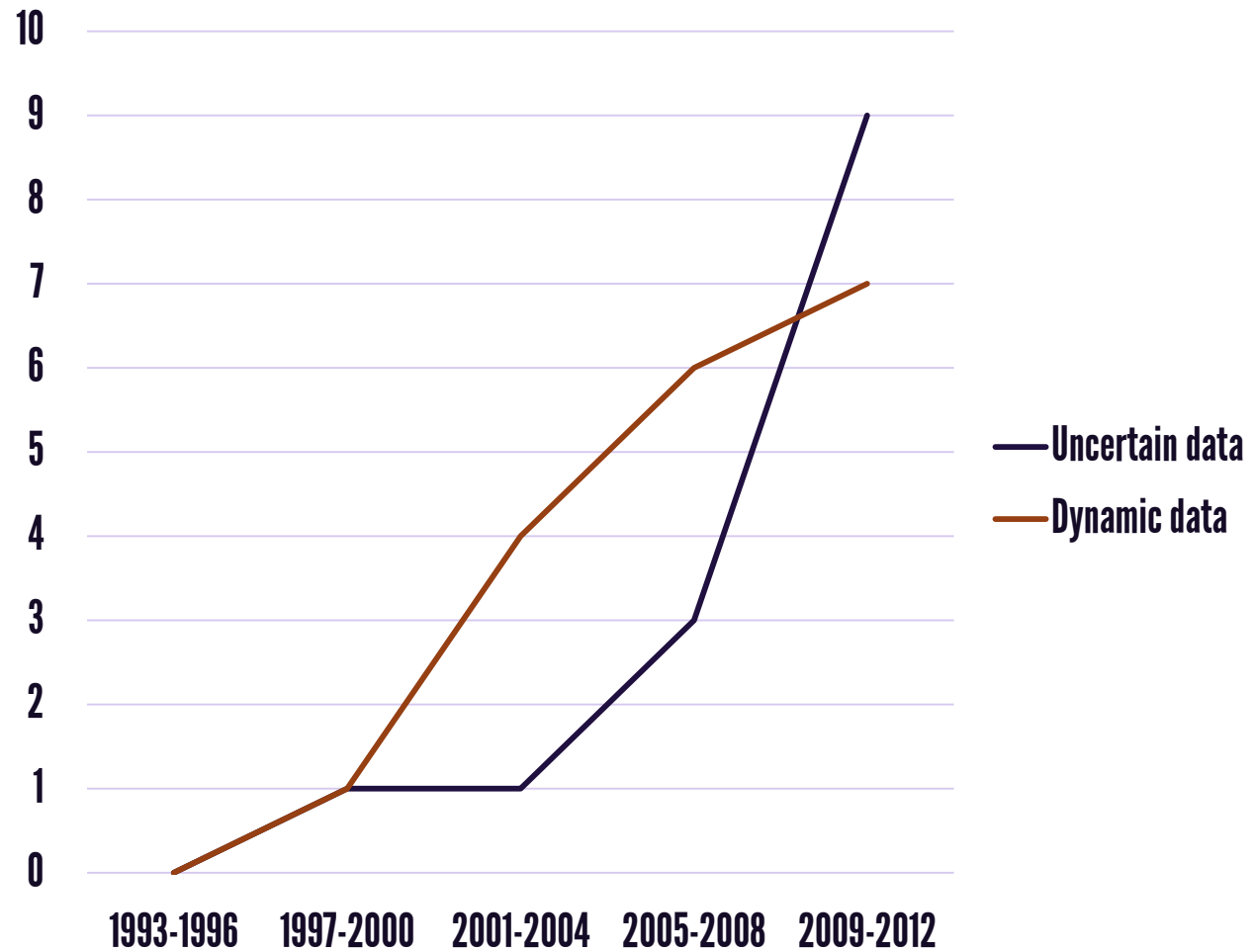


***Next language: spatio-temporal patterns?**

Progress of sequences and graphs

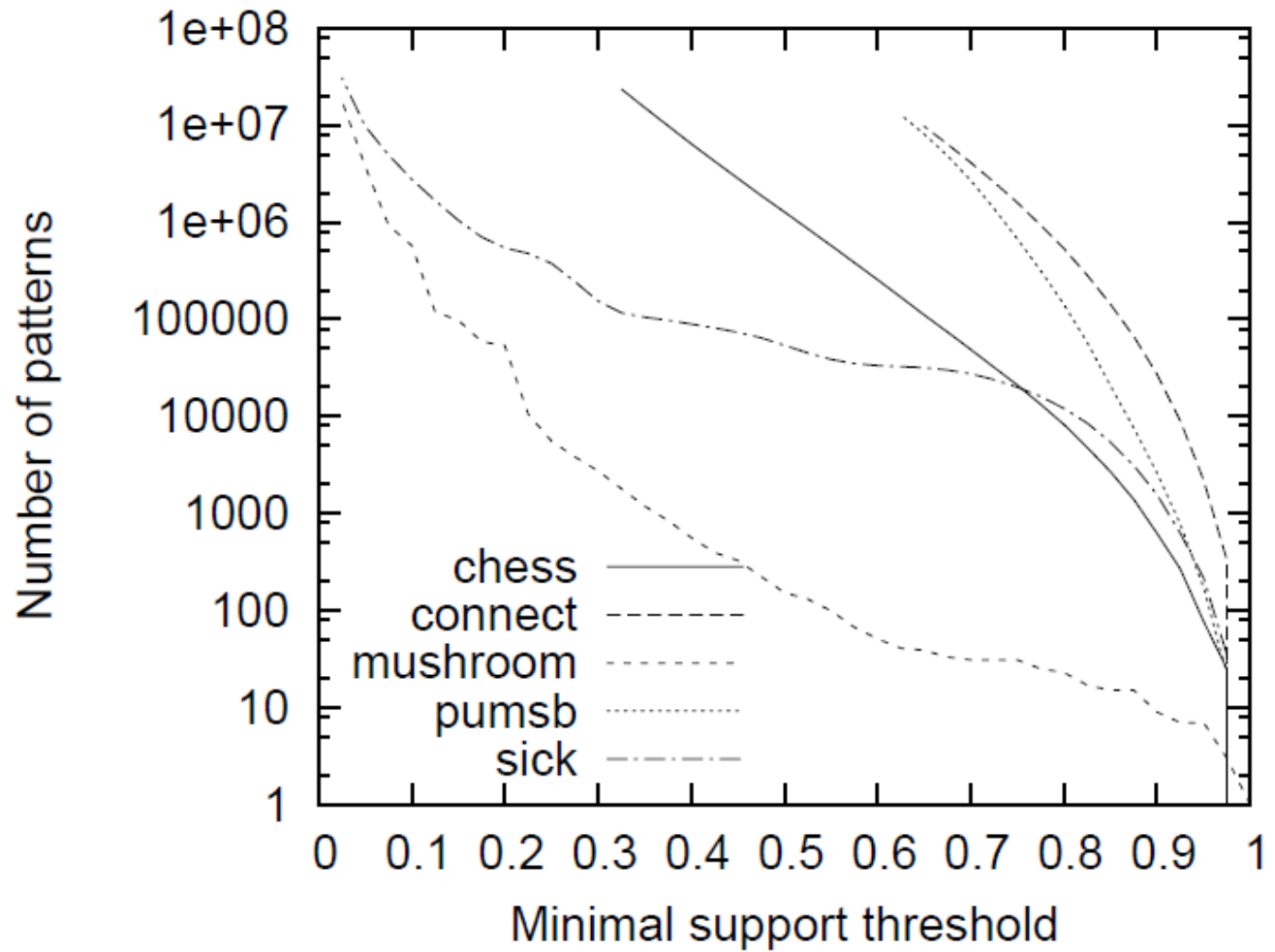


**Next challenge:
complex data**



***Uncertain, dynamic, massive, heterogeneous data**

Pattern explosion of frequent patterns (even with itemsets)



***How to reduce the number of patterns?**

Two strategies
against pattern
explosion

Useful Patterns (UP'10) ACM SIGKDD Workshop Report

Jilles Vreeken
Department of Mathematics
and Computer Science
University of Antwerp
Belgium
jilles.vreeken@ua.ac.be

Nikolaj Tatti
Department of Mathematics
and Computer Science
University of Antwerp
Belgium
nikolaj.tatti@ua.ac.be

Bart Goethals
Department of Mathematics
and Computer Science
University of Antwerp
Belgium
bart.goethals@ua.ac.be

ABSTRACT

We provide a summary of the workshop on Useful Patterns (UP'10) held in conjunction with the ACM SIGKDD 2010, on July 25th in Washington, DC, USA. We report in detail on the motivation, goals, and the research issues addressed in the talks at this full-day workshop. More information can be found at: <http://www.usefulpatterns.org>

1. MOTIVATION

Pattern mining is an important aspect of data mining, concerned with finding local structure in data. Traditionally, the focus of research in pattern mining has been on completeness and efficiency. That is, trying to find all potentially interesting patterns as fast as possible. This focus, important as it is, has led our attention away from the most important aspect of the exercise: leading to useful results. To emphasize this, let us consider the following example.

Pattern mining in action, an example

Say a domain expert wants to extract novel knowledge from some data at hand. Or, more specifically, the expert wants to know what patterns are present in the data.

Typically, such data is complex, high-volume, and high-dimensional, and includes a mix of variables that are binary, categorical, hierarchical, or real-valued. Before the expert can apply, say a frequent itemset mining algorithm, the data has to be transformed into a binary matrix. For the numerical attributes, for instance, this involves discretizing the attributes into bins; a non-trivial step, in which potentially important information is easily lost.

Once this conversion is complete, the expert is ready to apply the pattern mining algorithm of choice. Before the mining can commence, however, she first has to define the constraints the patterns need to fulfill, including the main parameter in frequent set mining: the minimal support threshold. Not knowing what the correct value is, she at first sets the threshold at high level. This results in a boring result—the returned patterns mostly represent single items and some trivial associations she already knew. Disappointed by these results, she lowers the threshold somewhat, and starts the algorithm all over. In virtually no time at all, a gargantuan number of patterns is returned, many of which are much larger than the original data set.

Moreover, these patterns are typically presented as a text-file. Nevertheless, let us assume our expert patiently considers the result. Not knowing how to sift through these patterns, she sorts them on frequency, and starts from the top. At first, she sees the same singleton patterns, and as she manually lowers the file further she starts to see spurious patterns that can be explained either by singletons or by the trivial associations discovered in the first run. Ideally she would have been given just the most informative patterns. This result should be manageable, and allow the expert to zoom in on particular patterns of interest if needs be. After awkwardly considering the many, many discovered patterns, our expert actually finds an interesting pattern, or at least something that is surprising to what is already known. However, the only information typically readily available on this pattern, is the pattern itself, and how often it occurs. To the expert, this is not good enough, as she wants to know where the pattern occurs in the data, and whether there is anything interesting happening in those parts of the data, perhaps explaining the pattern, or making it even more interesting. In other words, we need to go back to the data. However, as of yet, there has not been much attention to how this should be done, nor are there tools available to assist in this matter. Further, since the data was transformed, exploring it with regard to a pattern is not trivial.

Making pattern mining useful

All things considered, even when constrained of the potential, in the above case the expert would not be very impressed by the usefulness of pattern mining. Unlike in other fields of data mining, such as clustering, in pattern mining presentation and visualization has not been a priority. However, even when we forget about presentation to a user, patterns are not yet as useful as they could be. While they provide highly detailed descriptions of phenomena in data, it remains difficult to make good use of them in, say, classification or clustering. While this is mostly due to the huge number of discovered patterns, making the result succinct at least, it does pose interesting research questions like "how to select patterns such that they are useful". Techniques that summarize the result exist, but focus primarily on being able to present the result in a way that is easy to interpret, rather than on targeting the usability of the result. In this workshop, we discuss into techniques that aim at all sets of patterns that are useful, where usefulness is defined in terms of their intended use. In short, it is exactly this kind of research, and these experiences and practices that we discussed at UP.

[Vreeken et al., 2010]

Constraint
Focusing on the most useful
patterns for the data expert

Condensed Representation
Removing all redundant
patterns

*Useful Patterns (UP'10) ACM SIGKDD Workshop

2

Constraint?



Support and confidence

- **Support of X** = proportion of transactions in dataset D containing X

$$\text{supp}(X, D) = |\{t \in D / X \subseteq t\}| / |D|$$

- **Support of $X \rightarrow Y$**

$$\text{supp}(X \rightarrow Y, D) = \text{supp}(X \cup Y, D)$$

- **Confidence of $X \rightarrow Y$**

$$\text{conf}(X \rightarrow Y, D) = \text{supp}(X \cup Y, D) / \text{supp}(X, D)$$

27/06/2016

Data mining - Local pattern discovery

25

***Do you remember, again?**

2

Constraint?



Emerging patterns: definition

- Growth rate of X in D_i :
 $gr_i(X,D) = \text{supp}(X,D_i) / \text{supp}(X,D-D_i)$
- X is an emerging pattern iff $gr_i(X,D) \geq \rho$ ($\rho > 1$)

Efficient Mining of Emerging Patterns: Discovering Trends and Differences. Dong et Li, KDD 1999.

Data mining - Local pattern discovery

27/06/2016

25

78

*Do you remember, again?

2

Constraint?



Downward and upward closure

specialization

Itemssets satisfying an anti-monotone constraint

Itemssets satisfying a monotone constraint

$freq(X, D) \geq t$
 $min(X.val) \geq t$
 $max(X.val) \leq t$
 $sum(X.val) \leq t$
 $X \subseteq \{A, B, C\}$
...

$freq(X, D) \leq t$
 $min(X.val) \leq t$
 $max(X.val) \geq t$
 $sum(X.val) \geq t$
 $X \supseteq \{A, B, C\}$
...

27/06/2011

26/10/2011

Data mining - Local pattern discovery

84

Do you remember, again?

Troubled Romantic Rich Dies Hidding Secret



Interest of constraints



Troubled romantic → Rich (conf = 0.5)

Troubled romantic → Dies (conf = 0.75)

Troubled Romantic **Rich** **Dies** **Hidding Secret**



Troubled romantic → Rich
(conf = 0.5 / lift = 0,8)

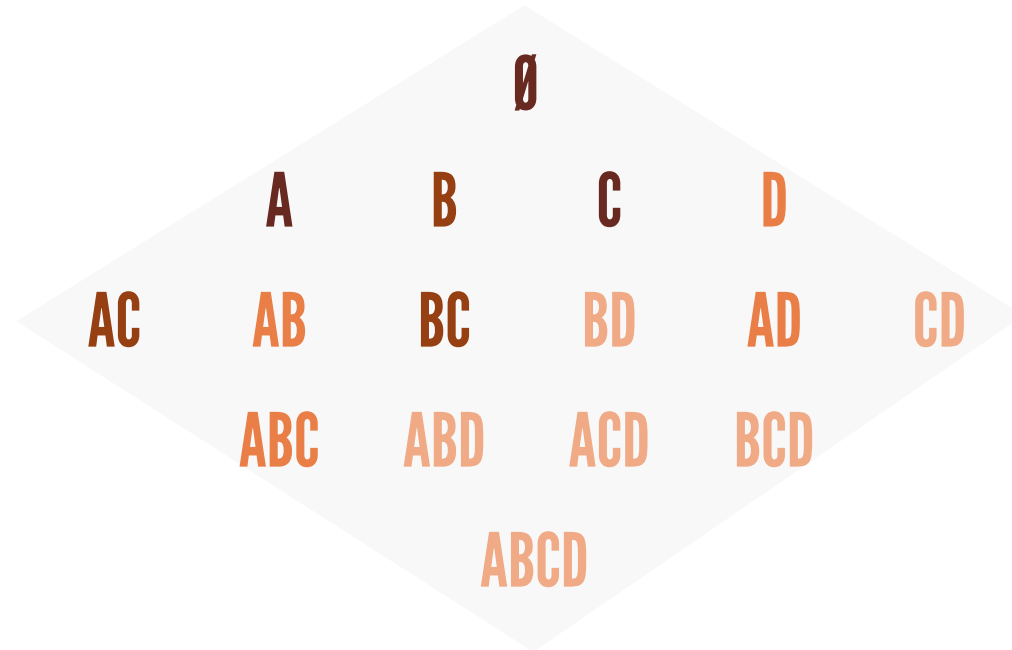
Troubled romantic → Dies
(conf = 0.75 / lift = 0,6)

Interest of constraints



	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

$$freq(X) \geq minfreq$$



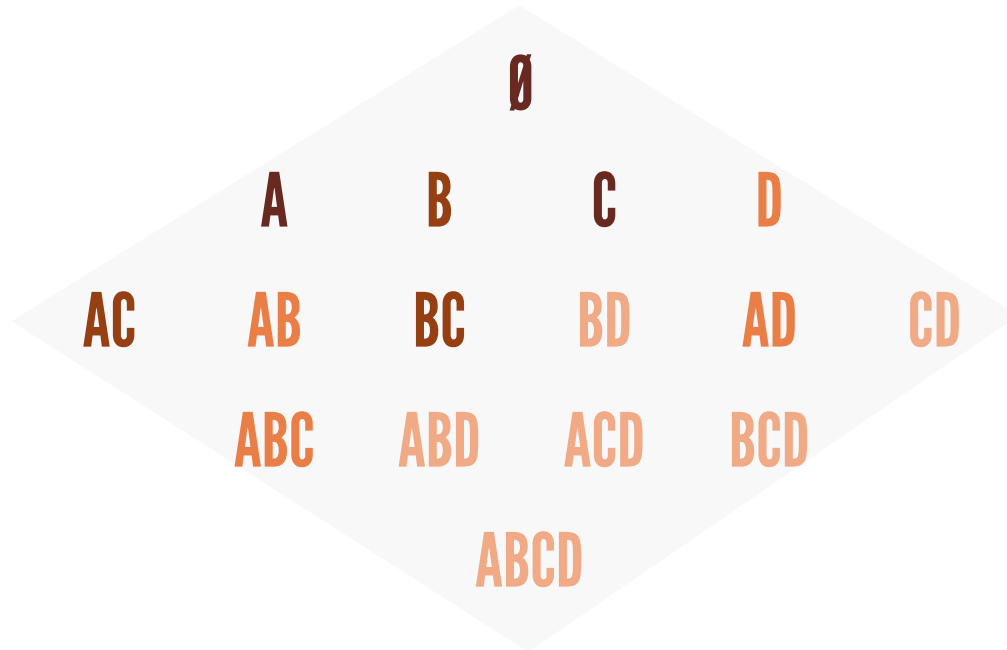
Challenge of constraints



***How to prune the search space for frequency?**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

$$freq(X) \geq minfreq$$



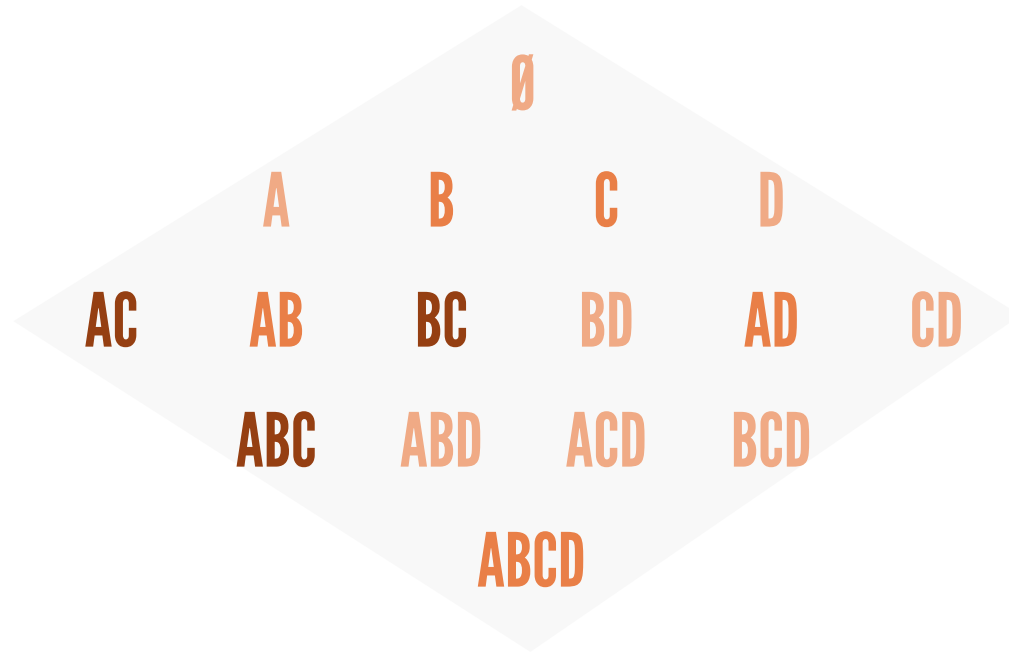
Challenge of constraints



***How to prune the search space for frequency?
Easy due to the downward closure**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

$$freq(X) \times |X| \geq minarea$$



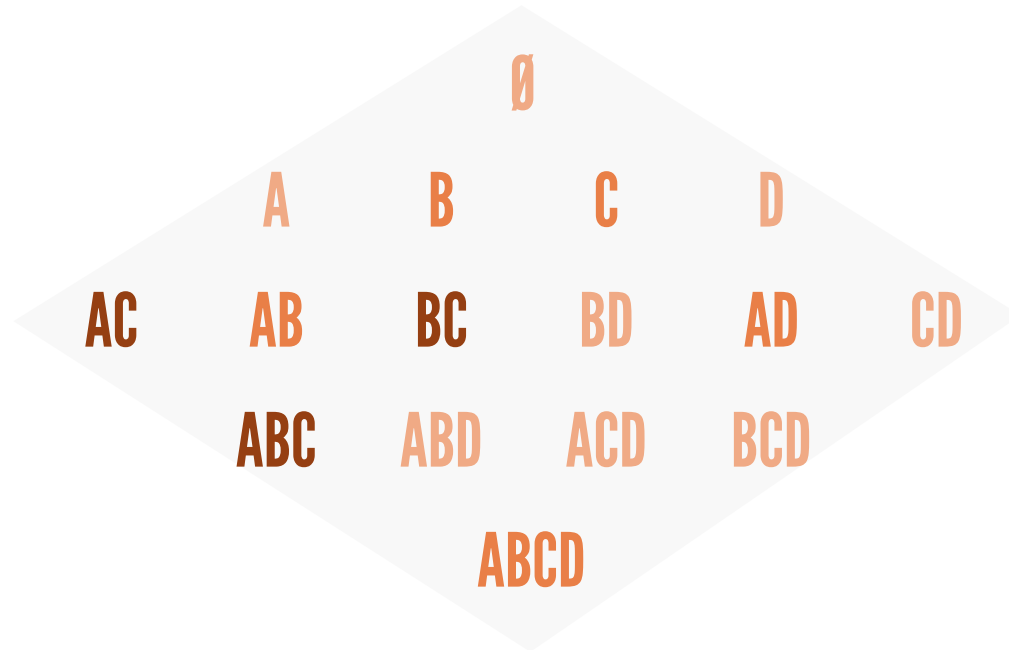
Challenge of constraints



***How to prune the search space for area?**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

$$freq(X) \times |X| \geq minarea$$



Challenge of constraints



***How to prune the search space for area?**

Relax the area constraint by $freq(X) \times 4 \geq minarea$

regularity, frequent, support

contrast, emerging, discriminative

exception, abnormal, surprising, anomaly, unexpected

utility

significant, chi-square, correlated

interesting, relevant

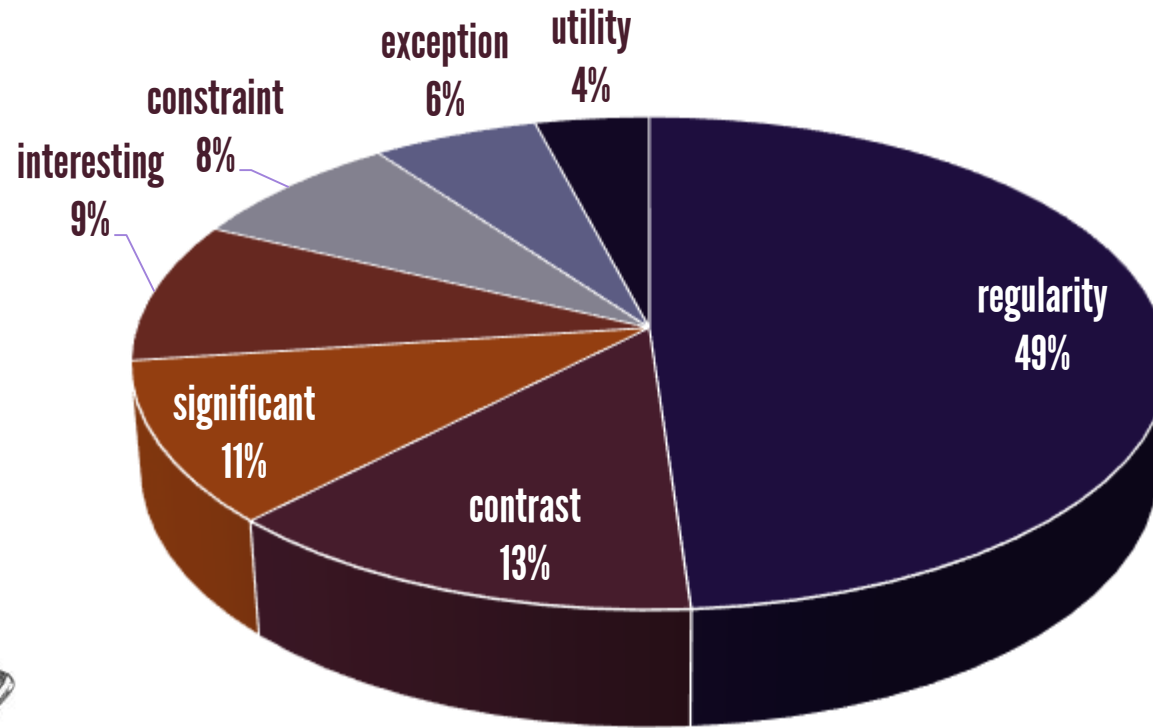
**Keywords about
constraints**

generic, monotone, anti-monone, constrained



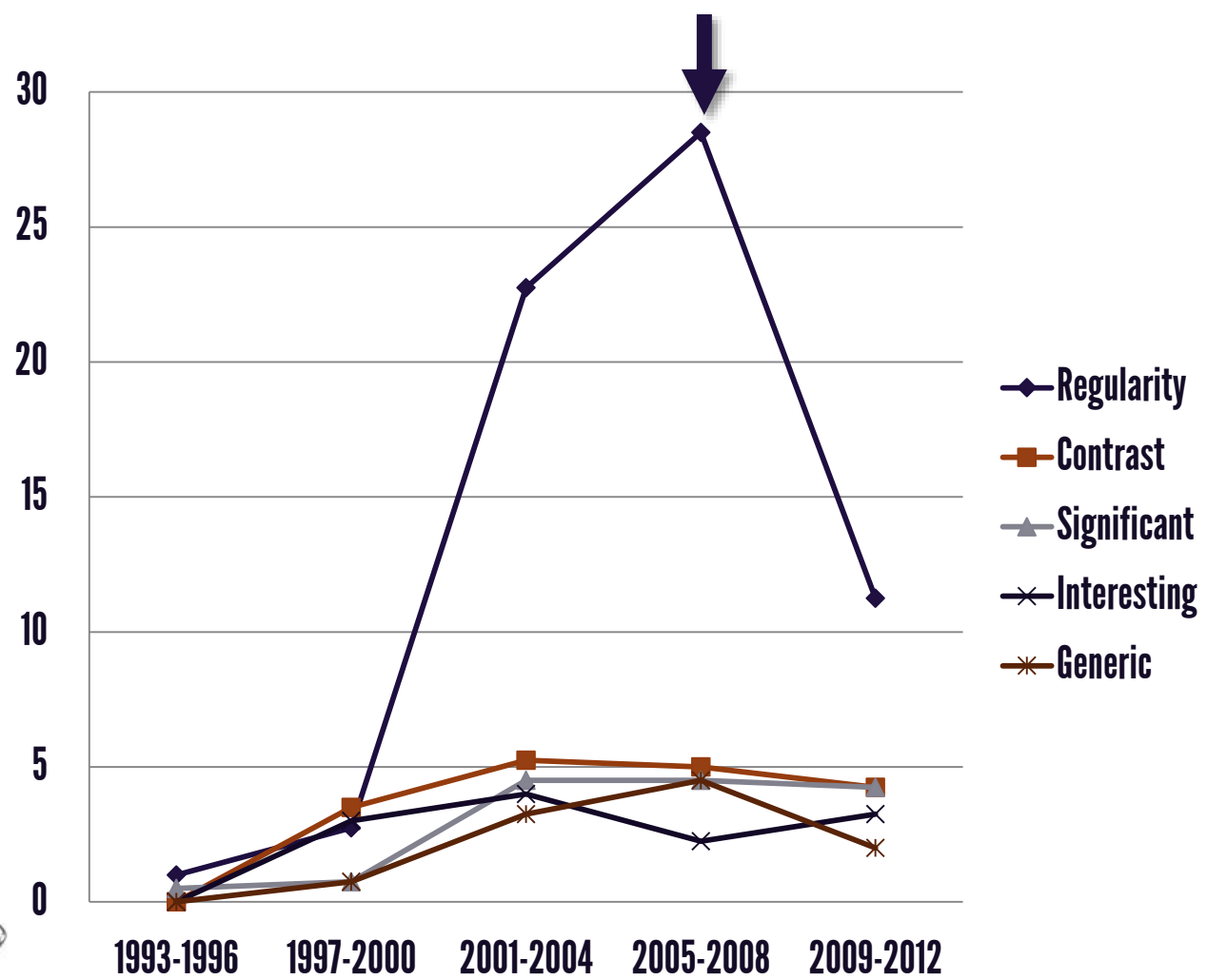
***Semi-automated assignment topic**

**Frequent patterns in
49% of publications**

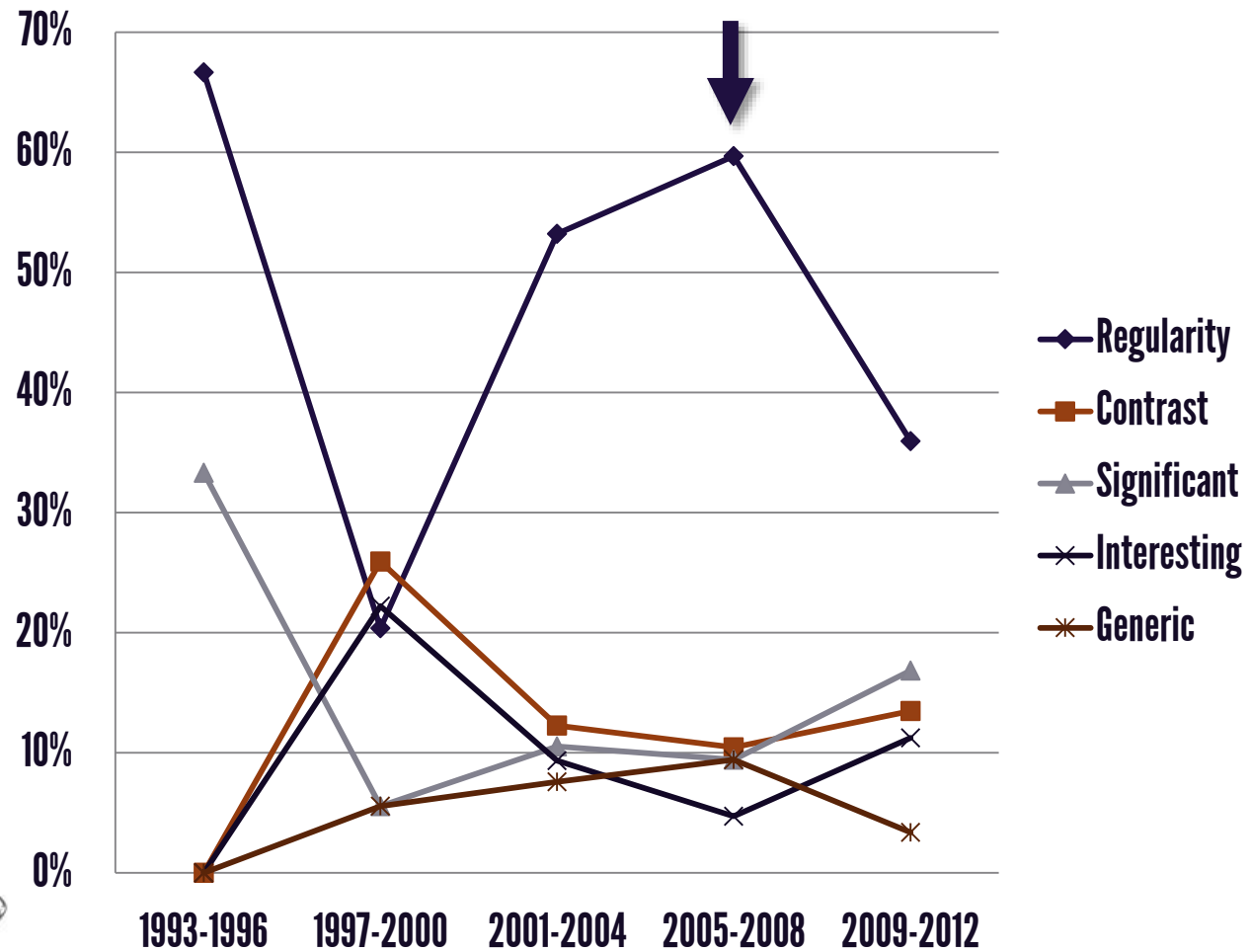


***Speed preferred to quality**

**Stable except for
frequency**



Progress of contrastive and significant patterns



***Only 42 papers with generic constraints**

Interestingness

Piatetsky-Shapiro

Agrawal

2003

Han et al.

2007

“we need work to bring in some notion of ‘here is my idea of what is interesting,’ and pruning the generated rules based on that input.”

Rakesh Agrawal Speaks Out
on Where the Data Mining Field Is Going, Where It Came From, How to Choose Problems and Open Up New Fields, Our Responsibilities to Society as Technologists, What Industry Owes Academia, and More

by Marianne Winsler



Rakesh Agrawal
<http://www.stanford.edu/~rakesh/>

Welcome to this installment of ACM SIGMOD Journal's series of interviews with distinguished members of the database community. The Marianne Winsler and today we are in San Diego, home of the 2007 SIGMOD and PDSN conference. I have here with me Rakesh Agrawal, who is a member of the research staff at IBM Almaden Research Center. Rakesh is well known for his work on data mining, for which he has received the SIGMOD Association Award and the SIGMOD Journalism Award. Rakesh is an IBM Fellow and an IEEE Fellow, and his PhD is from the University of Wisconsin. So, Rakesh, welcome!

Thanks, Marianne.

Rakesh, a paper of yours with Aron Baum and Thomas Dietterich on association rule mining just won the Best of Papers Award at this year's SIGMOD conference, so the paper that has had the most impact among those that appeared in the SIGMOD conference in recent years. Since that pioneering paper appeared, what has surprised you most about the development of the field of data mining?

When the three of us did the association rules paper, the truth is that we were a little skeptical about even sending the paper to SIGMOD. We thought the ideas were simple and the reviewers might reject the paper, saying, "There is not enough depth in the paper". What finally convinced us to send it is was the fact that the paper was solving a real problem, and the solution was a fairly general abstraction into which a lot of problems could be cast. I never imagined the different ways people have used association rules and the various extensions others have made to our paper. I read new papers on association rules and say, "Hey, I wish I could have done some of these things!"

David DeWitt once said that Tom Gray wrote a paper on data cubes and a few years later we had 200 papers on data cubes. Have too many people rushed to jump on the data mining bandwagon, or is the situation different here?

Interestingness

Piatetsky-Shapiro

Agrawal

2003

Han et al.

2007

The Grand Challenges Today

“A General Theory of Interestingness. What makes this rule, pattern more interesting than another?”

Summary from the KDD-03 Panel --
Data Mining: The Next 10 Years

Umasri M. Fayyad
IBM Corp
3500 American Ave, 4th Floor
Belmont, MA 02458 USA
www.DMGroup.com
Fayyad at DMGroup.com

Gregory Piatetsky-Shapiro
KDDgroup
www.kddgroup.com
gregory at kddgroup.com

Ramasamy Uthurusamy
General Motors
Warren, MI, USA
www.gm.com
uramy at gm.com

I. Introduction

The KDD-03 is a summary of the panel held at the 9th International Conference on Data Mining and Knowledge Discovery (KDD-03) on August 17, 2003 in Washington, D.C. The panel participants included the following:

- Umasri Fayyad of IBM Corp (Panel Organizer)
- Robert Agrawal of IBM Almaden Research Center
- David Dreyfus of AT&T Systems Laboratory
- Gregory Piatetsky-Shapiro of KDDgroup
- Engel Kuhlmann of the University of Wisconsin - Madison
- Ramasamy Uthurusamy of General Motors

The goal of the panel was to gather representative from academia and industry and to provide advice for the field through the study of a decade and a half of KDD research. As all have seen a significant growth in demand for data mining technology driven by a deluge of data. We have observed data mining growing as a leading research enterprise. However, we still struggle on two important fronts: the scientific and the commercial. On the scientific front, data mining still needs to reach a maturity level of attacking clearly well-posed, tractable problems. On the commercial front, the large opportunity has not yet been met with adequate tools and solutions. This panel was an attempt to address the possible future directions for Data Mining and KDD.

II. Position Statements

Position Statements by U. Fayyad (DMV Group)
Position Statements by T. Piatetsky-Shapiro (KDDgroup)

The field is well-served, and the level of interest in it seems to be increasing dramatically. The most serious challenge for the field is to figure out what the field is the state of which we are currently studying research from different fields. This is perhaps easiest in the beginning, but I never realize we have what it takes to change it in the future.

I think that the biggest remaining front from the scientific perspective is the lack of a fundamental theory of a clear and well-understood concepts of problems and challenges. The challenge is to consider at the very same level of interest we are putting data practitioners and people answering in applications. There still exist that a large variety of fields. But within the scientific frontiers and the commercial solution progress, the field cannot give and collect.

Some of the other issues I have, as a data miner, involve around the following observations:

- In a typical data mining scenario, I spend most of my time extracting and manipulating data, not really doing any mining-related experiments.
- The real "big bang" I have heard in my group after mining several applications, and it seems every time it is repeated and repeated, about Data Mining, again.
- There is little theory, or even supporting practice, to make us do. It is all a "black art" only. We need a theory for what we do, and how we do it. The theory can drive supporting topics of the data mining field to advance effectively.
- Without the practitioners of the research community, we are doomed to a state of badly over-predicting, and we may forget that there are so many people that other fields to support.
- The same problems we see too many other fields. We need to start using these things and we need to put the other side working about Data. How our problems are fundamentally to those and of great scientific interest to them.
- Industry-related issues: very interesting, but, some open for study.

Below is a summary of some of the position. We follow it with a summary of some of the topics discussed during the panel.

SIKDDO Symposium
Volume 1, Issue 2 - Page 181

Interestingness

Piatetsky-Shapiro

Agrawal

2003

Han et al.

2007

Data Min Knowl Disc (2007) 15:55–66
DOI 10.1007/s10618-006-0251-1

Frequent pattern mining: current status and future directions

Jiawei Han · Hong Cheng · Dong Xin · Xifeng Yan

Received: 22 June 2006 / Accepted: 8 November 2006 / Published online: 27 January 2007
Springer Science+Business Media, LLC 2007

Abstract Frequent pattern mining has been a focused theme in data mining research for over a decade. Abundant literature has been dedicated to this research and tremendous progress has been made, ranging from efficient and scalable algorithms for frequent itemset mining in transaction databases to numerous research frontiers, such as sequential pattern mining, structured pattern mining, correlation mining, associative classification, and frequent pattern-based clustering, as well as their broad applications. In this article, we provide a brief overview of the current status of frequent pattern mining and discuss a few promising research directions. We believe that frequent pattern mining research has substantially broadened the scope of data analysis and will have deep impact on data mining methodologies and applications in the long run. However, there are still some challenging research issues that need to be solved before frequent pattern mining can claim a cornerstone approach in data mining applications.

Keywords Frequent pattern mining · Association rules · Data mining research · Applications

“it is still not clear what kind of patterns will give us satisfactory pattern sets in both compactness and representative quality”

Exact solution

Exhaustive search

Speed of answer

Approximated solution

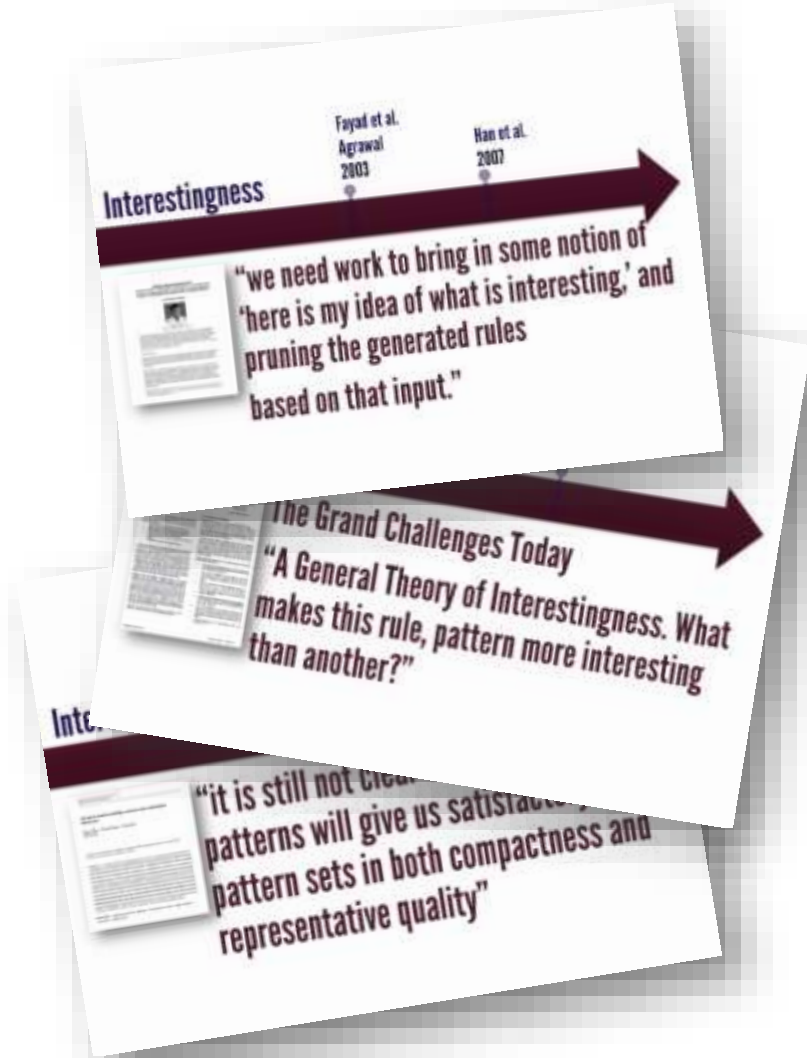
Heuristic search

Quality of solution

Pattern Mining vs Artificial Intelligence

***The footprint of databases**

1993

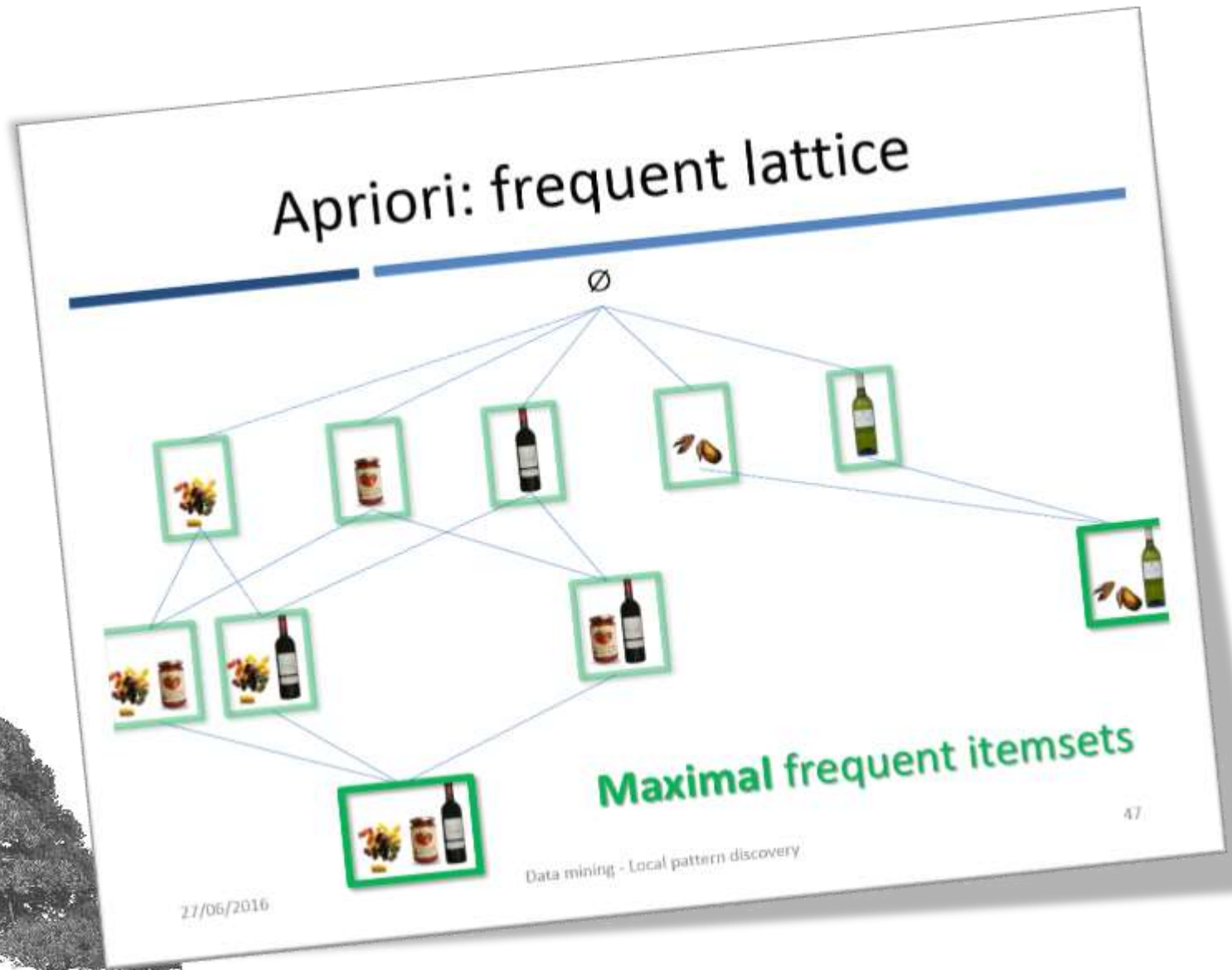


now*

***Quality of solution more important than speed of answer**

3

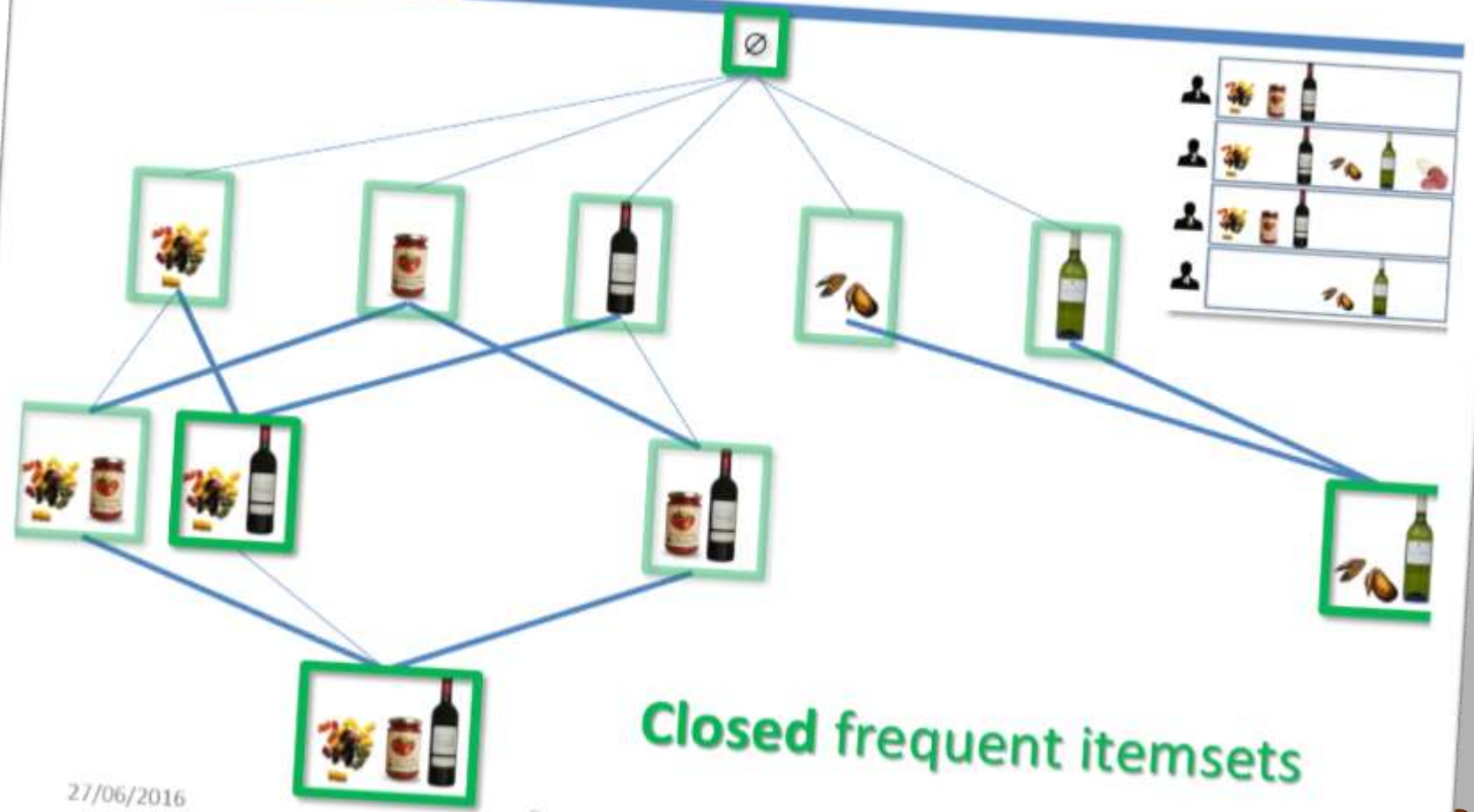
Condensed representations



***Do you remember, again and again?**

3

Apriori: frequent lattice



Condensed representations

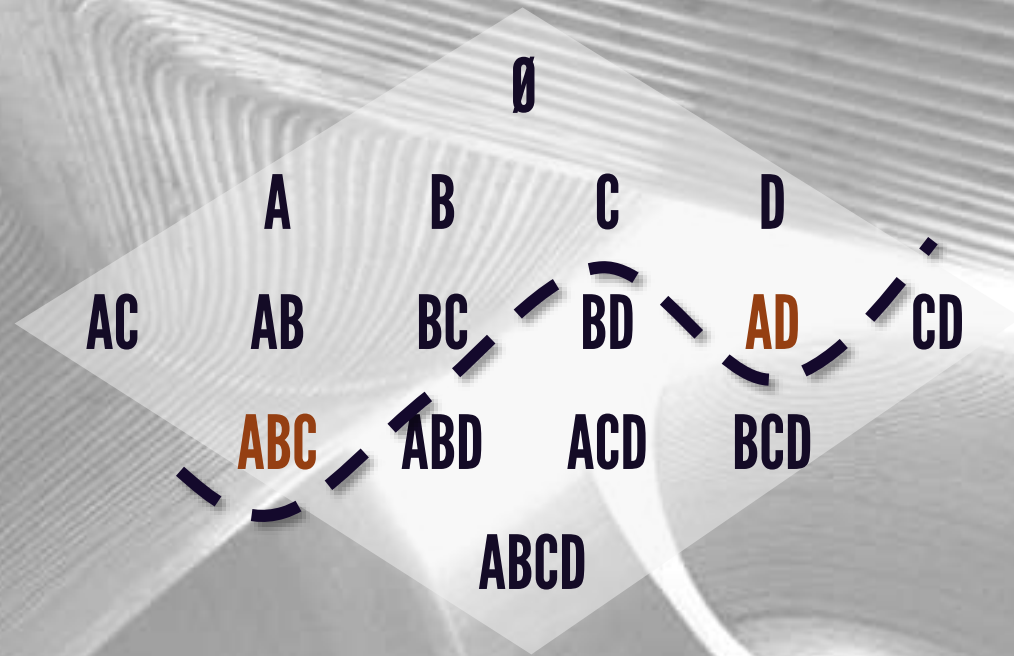
Closed frequent itemsets

27/06/2016

Data mining - Local pattern discovery

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

Maximal patterns

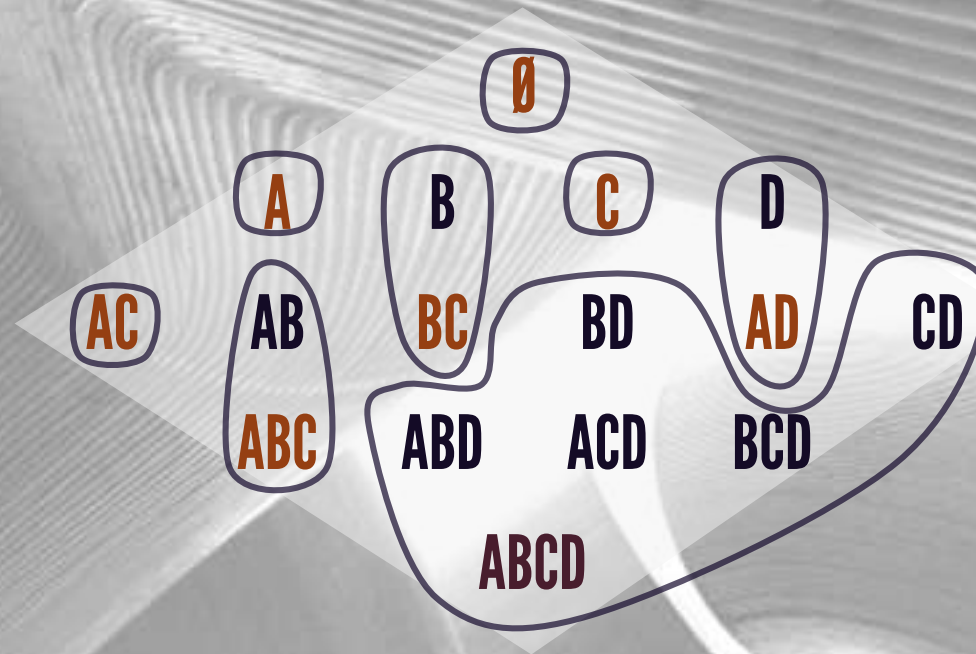


2 patterns with minfreq = 2

***larger frequent patterns w.r.t inclusion**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

Closed patterns

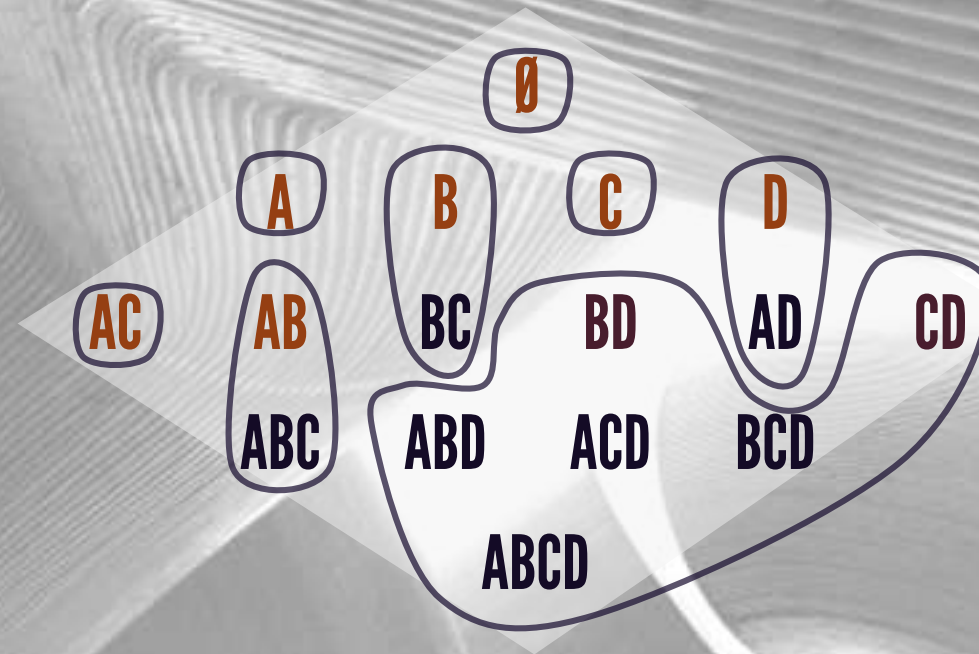


7 patterns with minfreq = 2

***maximal patterns of equivalence classes**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

Free patterns



7 patterns with minfreq = 2

***minimal patterns of equivalence classes**

More condensed representations

Non-derivable
itemsets
2002

K-free itemsets
2003

Mining All Non-Derivable Frequent Itemsets

Touan Calders¹
University of Antwerp, Belgium
Bart Goethals
University of Limburg, Belgium

Abstract

Recent studies on frequent itemset mining algorithms resulted in significant performance improvements. However, if the minimal support threshold is set too low, or the data is highly correlated, the number of frequent itemsets itself can be prohibitively large. To overcome this problem, recently several proposals have been made to construct a concise representation of the frequent itemsets, instead of mining all frequent itemsets. The main goal of this paper is to identify redundancies in the set of all frequent itemsets and to exploit these redundancies in order to reduce the result of a mining operation. We present deduction rules to derive tight bounds on the support of candidate itemsets. We show how the deduction rules allow for constructing a minimal representation for all frequent itemsets. We also present connections between our proposed and recent proposals for concise representations and we give the results of experiments on real-life datasets that show the effectiveness of the deduction rules. In fact, the experiments even show that in many cases, first mining the concise representation, and then creating the frequent itemsets from this representation outperforms existing frequent set mining algorithms.

1 Introduction

The frequent itemset mining problem [1] is by now well known. We are given a set of items I and a database D of subsets of I , together with a unique identifier. The elements of D are called transactions. An itemset $f \subseteq I$ is some set of items; its support in D , denoted by $\text{support}(f, D)$, is defined as the number of transactions in D that contains all items of f ; and an itemset is called α -frequent in D if its support in D exceeds $\alpha \cdot |D|$ and α is assumed when they are clear from the context. The problem now, given a minimal support threshold α and a set of items I , is to find all frequent itemsets in D .

¹Research Assistant of the Fund for Scientific Research - Flanders (FWO/Vlaanderen).

[Calders and Goethals, 2002]

Minimal k-Free Representations of Frequent Sets

Touan Calders¹ and Bart Goethals²
¹ University of Antwerp, Belgium
² Ghent Institute for Information Technology, Ghent

Abstract. Due to the potentially massive amount of frequent sets that can be generated from transactional databases, recent studies have demonstrated the need for concise representations of all frequent sets. These studies resulted in several successful algorithms that only generate a limited subset of the frequent sets. In this paper, we present a unifying framework encompassing most known concise representations. Because of the deeper understanding of the different proposals that obtained, we are able to generate new, possibly more concise, representations. These theoretical results are supported by several experiments showing the practical applicability.


1 Introduction

The frequent itemset mining problem is by now well known [1]. We are given a set of items I and a database D of subsets of I . The elements of D are called transactions. An itemset $f \subseteq I$ is some set of items; its support in D , denoted $\text{support}(f, D)$, is defined as the number of transactions in D that contains all items of f . An itemset is called α -frequent in D if its support in D exceeds $\alpha \cdot |D|$ and the database D and the minimal support α are assumed when they are clear from the context. The goal is now, given a minimal support threshold and a database, to find all frequent itemsets. The set of all frequent itemsets is denoted $F(D, \alpha)$; the set of infrequent sets is denoted $\bar{F}(D, \alpha)$.

Recent studies on frequent itemset mining algorithms resulted in significant performance improvements. However, if the minimal support threshold is set too low, or the data is highly correlated, the number of frequent itemsets itself can be extremely large. To overcome this problem, recently several proposals have been made to construct a concise representation [2] of the frequent itemsets, instead of mining all frequent itemsets: Closed sets [4, 13–15], Free sets [6], Disjunction-Free Sets [3, 16], Generalized Disjunction-Free Generalized [12, 13], and Non-Derivable Itemsets [8].

A Concise Representation of frequent sets is a subset of all frequent sets with their supports that contains all information about frequent itemsets. These studies demonstrated the value of a concise representation in a more general context. Our definition resembles theirs, but for reasons of simplicity we only concentrate on representations that are exact, and the frequent itemsets.

[Calders and Goethals, 2003]



closed

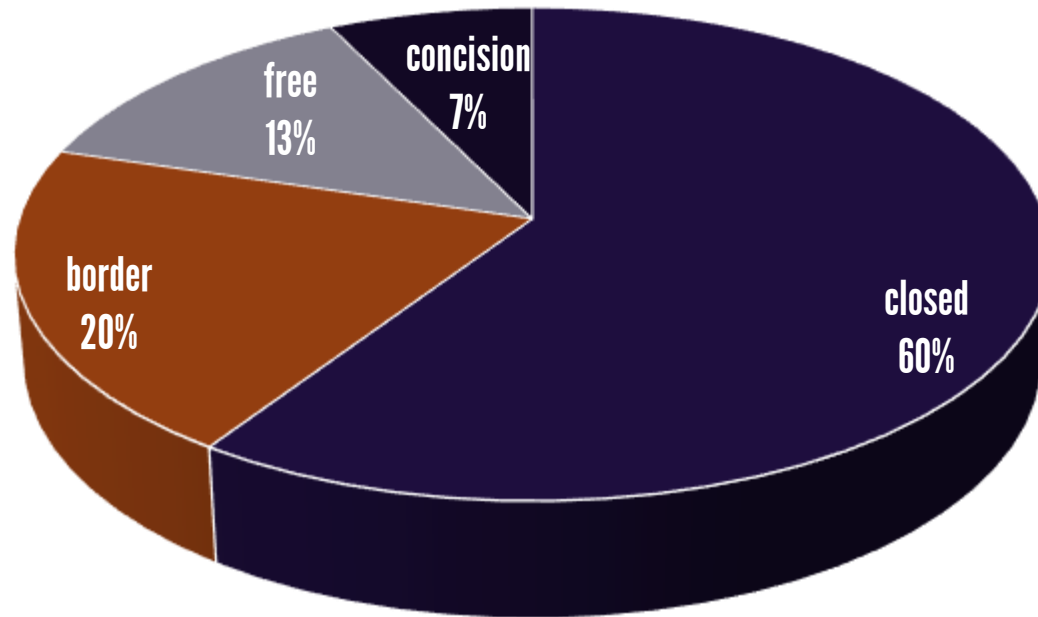
Keywords about condensed representations

border, maximal, minimal

free, generator, non-derivable

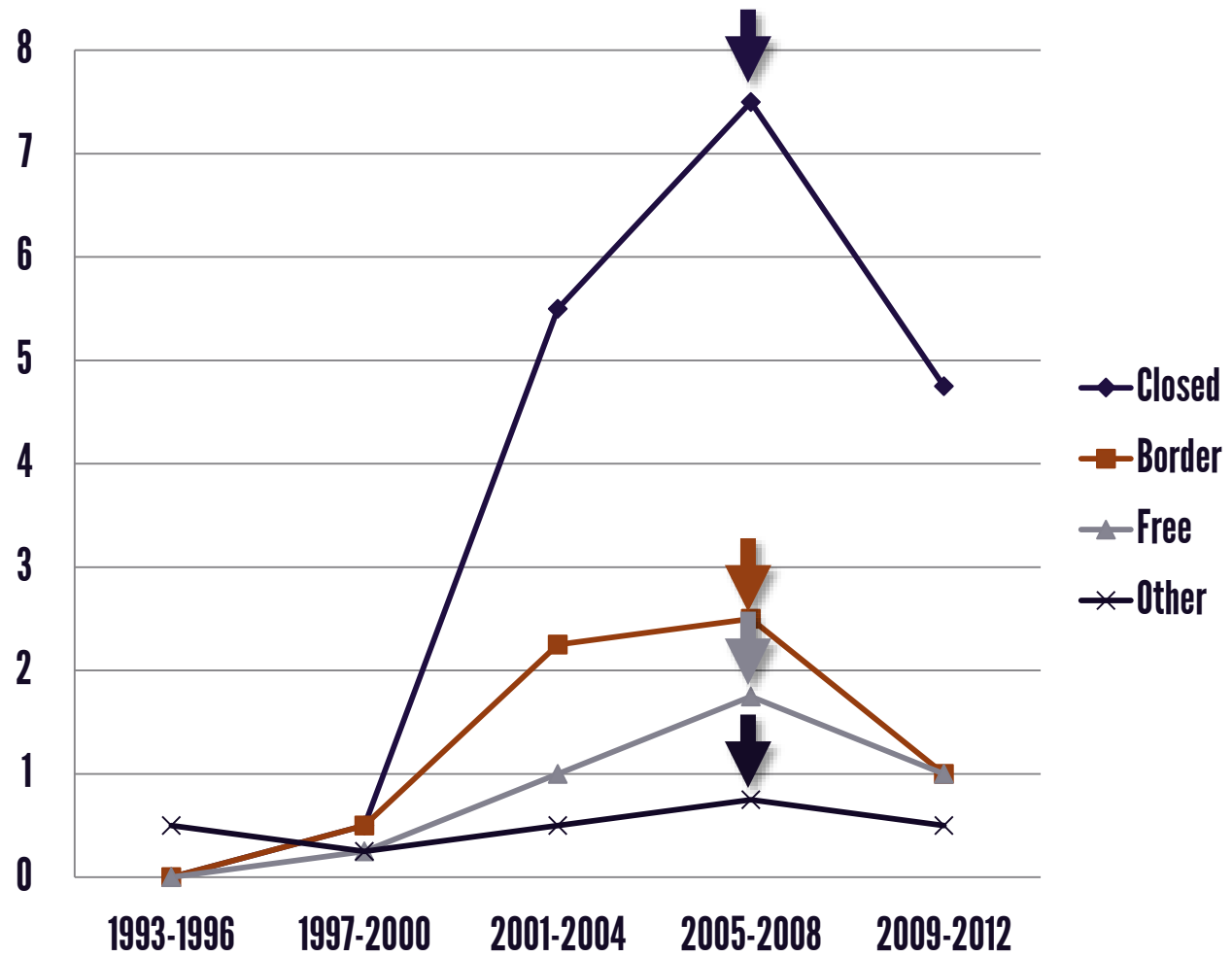
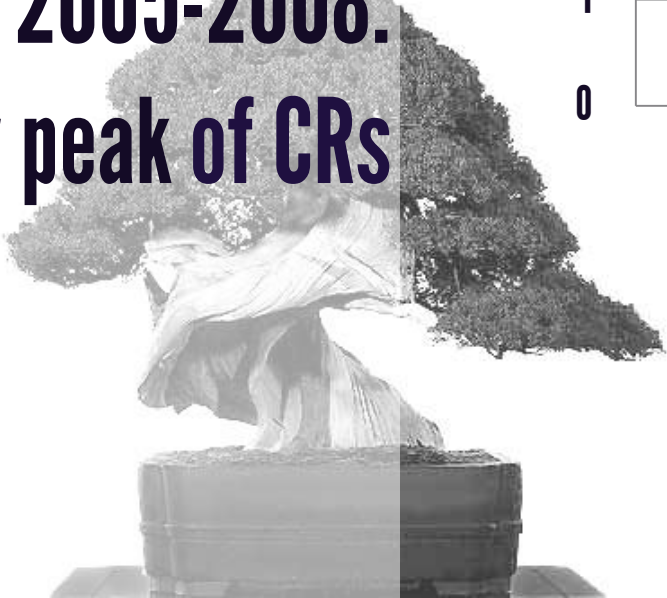
***Semi-automated topic assignment**

**Closed patterns in
60% of CR**



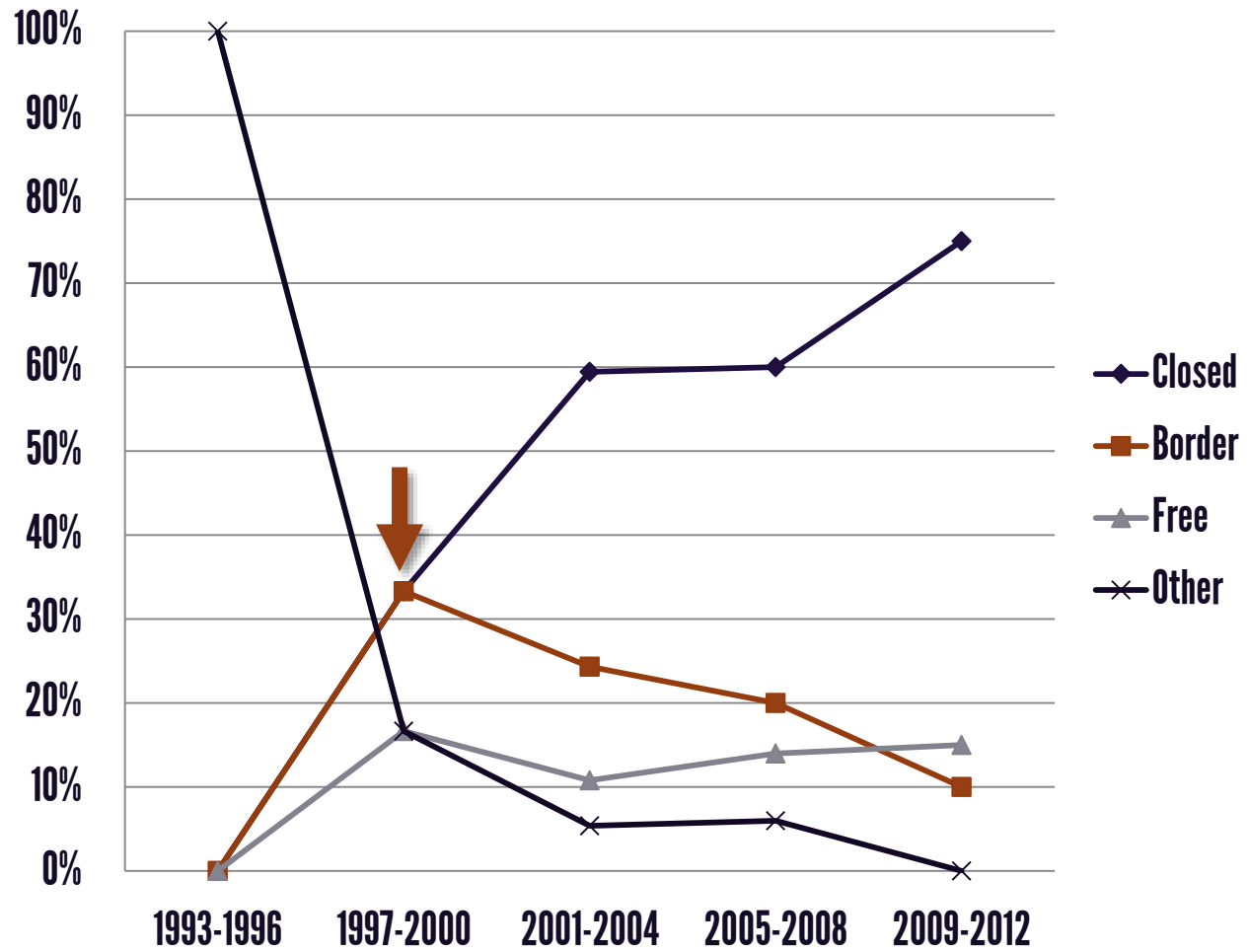
***Why this success?**

2005-2008: Activity peak of CRs



***Pattern-based models instead of CRs**

Progress of free itemsets



Pattern-based classifiers

CBA
1998



***Two-phases: pattern extraction & model construction**

Pattern-based models

Krimp
2006

The Chosen Few
MINI
2007

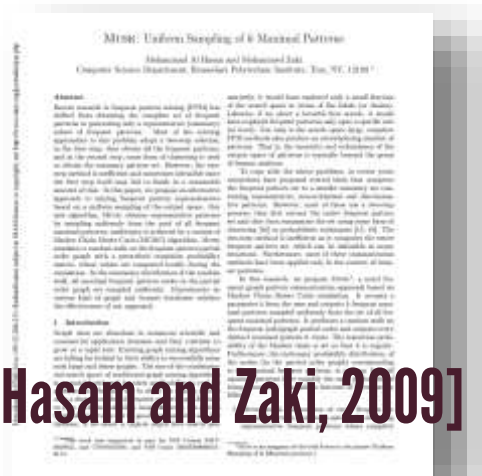


***Two-phases: pattern extraction & model construction**

Pattern-based models

MUSK
2009

Local pattern
sampling
2011



[Hasam and Zaki, 2009]



[Boley et al., 2011]

*Sampling

Stop completeness!

**Useful Patterns
ACM SIGKDD
Workshop
2010**

**ECMLPKDD most-
influential paper award
2012**

**“Please, please stop making new algorithms for
mining all patterns”**

Toon Calders

1993

Exact solution	Approximate solution
Exhaustive search	Heuristic search
Speed of answer	Quality of solution

Pattern Mining vs Artificial Intelligence

**The footprint of databases*

Pattern-based models

MUSK 2009

Local pattern sampling 2011

Useful Patterns ACM SIGKDD Workshop 2010

ECMLPKDD most-influential paper award 2012

Stop completeness!

"Please, please stop making new algorithms for mining all patterns"

Toon Calders

now*

***Approximate solution and Heuristic search!**

What are the recent trends of Pattern Mining?

Pattern mining as an optimization problem

Top-k frequent patterns
2000

Skypatterns
2011

Optimal patterns
Dominance programming
2015

Mining N -most Interesting Itemsets

Ada Wu-chao Fu, Ruihong Wang-wai Kwong, Jian Tong

Department of Computer Science and Engineering
The Chinese University of Hong Kong, Hong Kong
{ada.wu, rkwong}@cs.cuhk.edu.hk

Abstract. Previous methods on mining association rules require users to input a minimum support threshold. However, there can be too many or too few resulting rules if the threshold is set inappropriately. It is difficult for end-users to find the suitable threshold. In this paper, we propose a different setting in which the user does not provide a support threshold, but instead indicates the amount of results that is required.

1 Introduction

In recent years, there has been a lot of studies in association rule mining. An example of such a rule is:

[Fu et al., 2000]

where x is the set of items, A is the set of items that the item x is purchased with, and B is the set of items that the item x is not purchased with. This rule indicates that a certain percentage of people that buy biscuits also buy orange juice at the same time, and there are quite many people buying both biscuits and orange juice.

Typically, this method requires the users to specify the minimum support threshold, which in the above example is the minimum percentage of transactions buying both biscuits and orange juice in order for the rule to be generated. However, it is difficult for the users to set this threshold to obtain the result they want. If the threshold is too small, a very large amount of results are mined. It is difficult to select the useful information. If the threshold is set too large, there may not be any result. Users would not have much idea about how large the threshold should be. Here, we study an approach where the user can set a threshold on the amount of results instead of the threshold.

We observe that solutions to multiple data mining problems including mining association rules [2, 4], mining correlation [3], and subspace clustering [5], are based on the discovery of large itemsets, i.e., itemsets with support greater than a user-specified threshold. Also, the mining of large itemsets is the most difficult part in the above methods. Therefore, we would like to mine the interesting itemsets instead of interesting association rules with the constraint on the number of large itemsets instead of the minimum support threshold value. The

2011 11th IEEE International Conference on Data Mining

Mining Dominant Patterns in the Sky

Arnaud Soulet¹, Chedly Elghal², Marc Plateron³ and Bruno Chérollet⁴

¹Université François Rabelais de Tours, 41, EA 2400, F-37000, France
²INRIA Nancy Grand-Est, France
³Université de Lyon, CNRS, INSA de Lyon, UMR5216, F-69622, France
⁴Université de Caen Normandie, CNRS, GREYC UMR6072, F-14032, France

Abstract. Pattern discovery is at the core of numerous data mining tasks. Although many methods focus on efficiency in pattern mining, they still suffer from the problem of choosing a threshold that influences the final extraction result. The goal of our study is to make the results of pattern mining useful from a non-expert point of view. To this end, we propose that the pattern discovery process the idea of skyline queries in order to mine skyline patterns in a threshold-free manner because the skyline patterns satisfy a natural property of dominance: they are only less a global interest but also have attributes that are really interesting by the user. In this work, we first establish theoretical relationships between pattern condensed representations and skyline patterns existing. We also show that it is possible to compute automatically a subset of minimum length in the user query which allows the patterns to be condensed and then facilitates the computation of the skyline patterns. This forms the basis for a novel approach to mining skyline patterns. We illustrate the efficiency of our approach over several data sets including a case from domain knowledge where the goal is to find a set of dominant patterns in a multi-dimensional space.

[Soulet et al., 2011]

The process of extracting useful patterns from data, called pattern mining, is an important tool for data analysis and has been used in a wide range of applications and domains such as bioinformatics [1] or recommendation [2]. Since the pioneering works of Agrawal et al. [3], Maniatis et al. [4], a large amount of work has been developed and many pattern extraction problems are now identified and understood from both theoretical and computational perspectives.

Most existing pattern mining approaches minimize patterns with respect to a given set of constraints that range from extremely simple to very complex. For instance, given a transaction database, a well-known “very” pattern mining task is to compute all itemsets (i.e., sets of items) that appear in at least k transactions. Another mining approach is to extract from a given graph database all subgraphs that have a diameter larger than f , connectivity higher than c , and where each vertex has a degree bounded by d . In fact, the community has made great efforts on sophisticated algorithms pushing the complexity away from the mining process [5]. But it has paid less attention to how to define constraints in pattern, many constraints entail choosing of

threshold values such as the well-known minimum frequency. This notion of “hardcoding” has serious drawbacks. Unless specific domain knowledge is available, the choice is often arbitrary and may lead to a very large number of extracted patterns which can reduce the success of any subsequent data analysis. This drawback is obviously even deeper when several thresholds are needed and have to be combined. A second drawback is the stronger conservative spirit: a pattern is either above or below the threshold. What about patterns that respect only some thresholds? With this paradigm it is very difficult to apply subtle selection mechanisms. There are very few works such as [6] which propose to introduce a soft-edge criterion into the mining process. Other studies based user preferences at the mining task in order to limit the number of extracted patterns such as the top- k patterns [7, 8]. By associating each pattern with a score, we can cover the user defined interest, and several algorithms are proposed in a single framework to mine top- k patterns in terms of top- k itemsets, top- k subgraphs, top- k itemsets and top- k itemsets and top- k itemsets.

In this work, we focus on making the results of pattern mining useful from a non-expert point of view. To this end, we integrate into the pattern discovery process the idea of skyline queries [9] in order to mine skyline patterns in a threshold-free manner. Such queries have attracted considerable attention due to their importance in multi-criteria decision making. Roughly speaking, in a multi-dimensional space where a performance is defined for each dimension, a point A dominates another point B if A is better (i.e., more preferred) than B in at least one dimension, and is not worse than B in every other dimension. For example, in a multi-criteria space of patterns we prefer a pattern A over a pattern B if A is longer and has a high confidence. In this paper, we propose a dominance ordering pattern mining approach. We first extract from a given graph database all subgraphs that have a diameter larger than f , connectivity higher than c , and where each vertex has a degree bounded by d . In fact, the community has made great efforts on sophisticated algorithms pushing the complexity away from the mining process [5]. But it has paid less attention to how to define constraints in pattern, many constraints entail choosing of

Modeling and Mining Optimal Patterns using Dynamic CSP

Willy Ugarte^{*}, Patrick Bonnamant^{*}, Bruno Chérollet^{*} and Saurabh Lodha^{*}

^{*}GREYC Laboratory (CNRS UMR 6072)
Université de Caen Normandie, 14032 CAEN,
{[urarte,bonnamant,loalha]}@univ-caen.fr

Abstract. We introduce the notion of Optimal Pattern (OP), defined as the best pattern according to a given user preference, and show that OPs encompass many data mining patterns. Thus, we propose a generic method based on a Dynamic Constraint Satisfaction Problem to mine OPs, and we show that any OP is characterized by a finite constraint set and a set of constraints to be dynamically added. Finally, we perform an experimental study comparing our approach to all best methods on several types of OPs.

Keywords: Pattern Mining, Optimization, Dynamic CSP.

1. INTRODUCTION

Relationships between constraint programming (CP) and data mining have recently received considerable attention [1], [2], [3]. This success likely comes from the declarative and flexible model provided by this new framework. Hypotheses and patterns that analysts seek to discover are specified in terms of constraints. Then, powerful CP solvers attempt to produce the complete set of patterns satisfying the constraints. One of the main objectives of constraint programming is to generate a complete set of solutions to a given problem. This approach falls into the general trend to produce sets of patterns satisfying properties on the whole set of patterns [4] which is a promising road to discover useful patterns.

There are several propositions in the literature to extract sets of patterns defined by constraints based on several patterns such as pattern mining [5], minimum mining [6], top- k [7], concept clustering [8], suboptimization mining [2], to name a few. Some of these methods share the idea of reducing the search space by pruning patterns that do not satisfy the constraints. However, this approach is not very efficient because it requires to compute all patterns that satisfy the constraints. In this paper, we propose a new approach to mine OPs. We first extract from a given graph database all subgraphs that have a diameter larger than f , connectivity higher than c , and where each vertex has a degree bounded by d . In fact, the community has made great efforts on sophisticated algorithms pushing the complexity away from the mining process [5]. But it has paid less attention to how to define constraints in pattern, many constraints entail choosing of

constraints that can be expressed in our framework. Other methods based on CP are devoted to specific sets of patterns such as top- k [8] or skypatterns [9].

The non-exclusive list of data mining methods to extract sets of patterns shows the importance of this research activity. Many other methods for mining sets of patterns can be designed. But what kinds of sets of patterns can be produced? What about these sets of patterns w.r.t. usual pattern mining methods? This paper addresses these issues.

The key idea is to model sets of patterns thanks to the notion of preference. Then, we define the optimal patterns (OPs) which are the best patterns w.r.t. the preference (an OP is a pattern for which there are no preferred patterns). A major result is that numerous data mining problems can be modeled in this framework including well-known tasks (e.g. condensed representation of patterns [11], [12], sub-optimal subgroups [13]) but, more interestingly, the problems indicated above and many others as we will see in the rest of the paper. Taking the notion of OP as the core of our framework, we propose a new framework to mine OPs which is based on a dynamic CSP (Constraint Satisfaction Problem) framework. The main principle is as follows: when a solution is found, a constraint is dynamically added to the current CSP so that a better solution and then successively reduce the search space. The process stops when no better solution can be found. Finally, experiments show that our approach compares well with all best methods despite its generic modeling.

Section II introduces the notion of OPs, defined as the best patterns according to a given user preference. Section III shows that OPs encompass many data mining patterns. Section IV presents the generic approach to mining OPs. Section V illustrates the efficiency of our approach on several types of OPs. Finally, Section VI concludes the paper.

We search a few well-known results from the pattern mining area. Let T be a set of binary items.

- An itemset s is a non-empty subset of T , i.e., $s \subseteq T$.
- An itemset s is a non-empty subset of T , i.e., $s \subseteq T$.

***Focusing on the best patterns**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D



Top-k pattern mining

***Finding the k patterns maximizing an interestingness measure**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D



Top-k pattern mining

***Finding the 3 most frequent patterns: \emptyset (5), A (4), C (4)**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

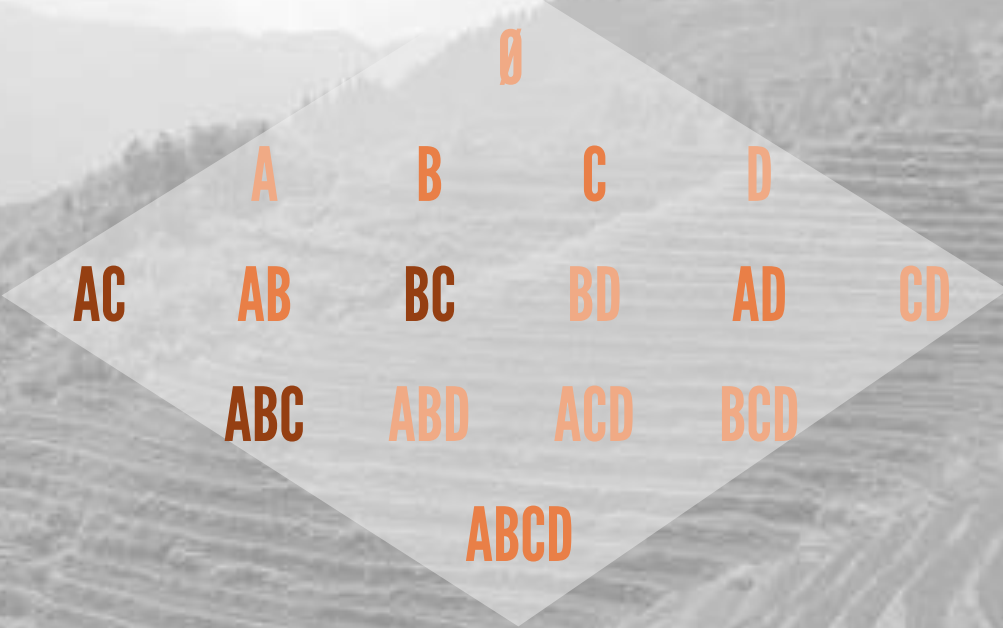


Top-k pattern mining

***Finding the 3 most frequent patterns: \emptyset (5), A (4), C (4)**

****Easy due to anti-monotone property of frequency**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D



Top-k pattern mining

***Finding the 3 patterns maximizing area: AC (6), BC (6), ABC (6)**

****Branch&Bound method**

Top-k pattern mining

Compact

Threshold free

Best patterns

Not fast*

No diversity

***Exact resolution is costly / sometimes heuristic search (beam search)**

Top-k pattern mining

Compact

Threshold free

Best patterns

Not fast

No diversity*

***Diversity issue: top-k patterns often very similar**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

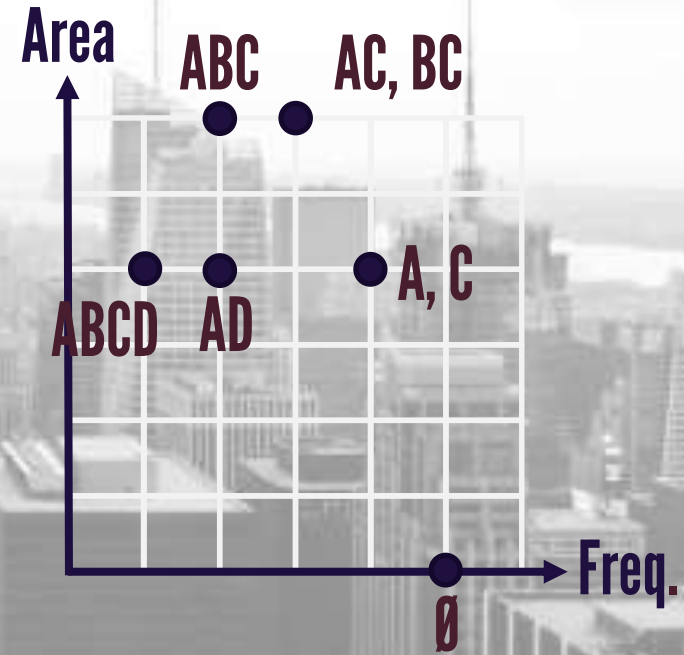
Pattern	Freq.	Area	
\emptyset	5	0	Top frequent
A	4	4	
C	4	4	
AC	3	6	Top area
BC	3	6	
AD	2	4	
ABC	2	6	
ABCD	1	4	

Skyline pattern mining

***How to find a trade-off between several criteria?**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

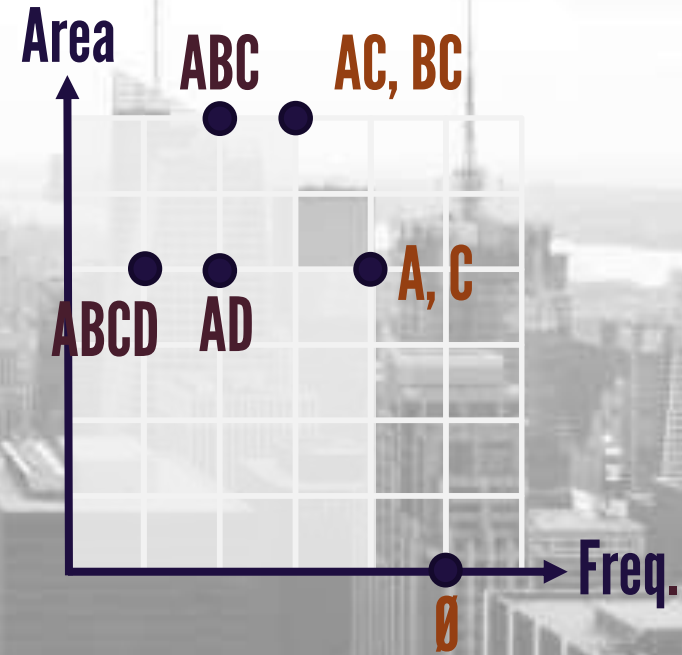
Pattern	Freq.	Area
\emptyset	5	0
A	4	4
C	4	4
AC	3	6
BC	3	6
AD	2	4
ABC	2	6
ABCD	1	4



Skyline pattern mining

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

Pattern	Freq.	Area
\emptyset	5	0
A	4	4
C	4	4
AC	3	6
BC	3	6
AD	2	4
ABC	2	6
ABCD	1	4

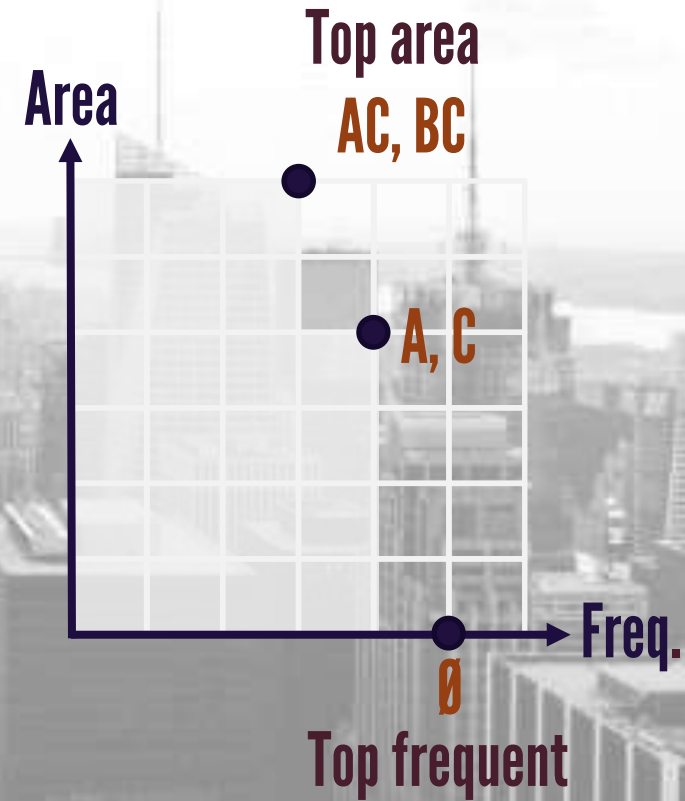


Skyline pattern mining

***Dominated space**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

Pattern	Freq.	Area
\emptyset	5	0
A	4	4
C	4	4
AC	3	6
BC	3	6
AD	2	4
ABC	2	6
ABCD	1	4

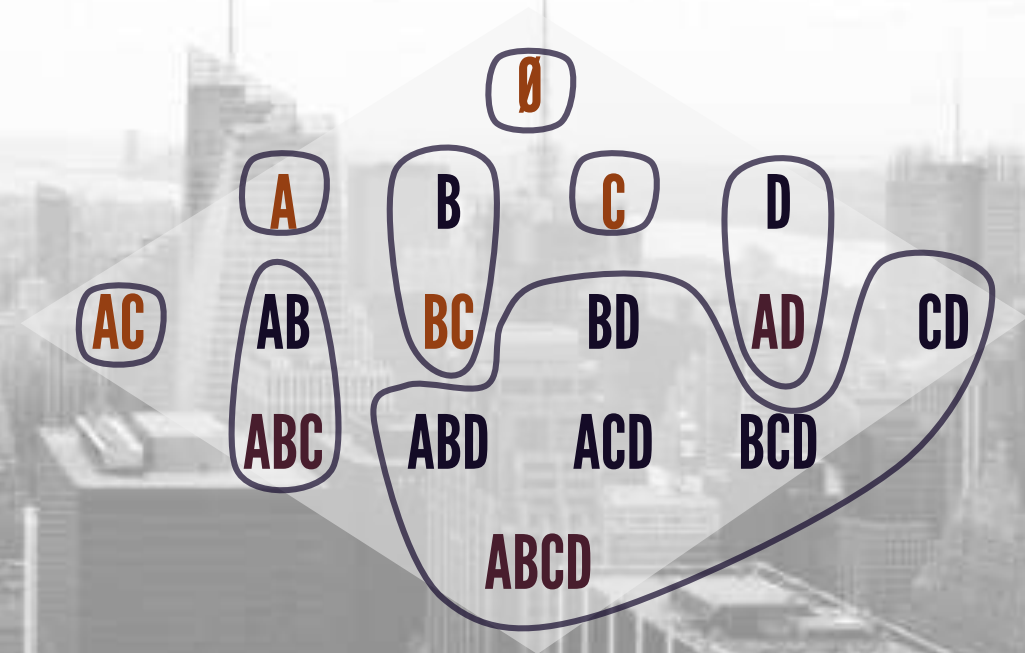


Skyline pattern mining

***Skypatterns = non-dominated patterns**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

Pattern	Freq.	Area
\emptyset	5	0
A	4	4
C	4	4
AC	3	6
BC	3	6
AD	2	4
ABC	2	6
ABCD	1	4



Skyline pattern mining

***Skypatterns are closed patterns**



Maximal patterns

Closed patterns

**Dominance
programming for
optimal patterns**

Top-k patterns

Skypatterns

***A pattern is optimal if it is not dominated by another.**



Maximal patterns*

Closed patterns

**Dominance
programming for
optimal patterns**

Top-k patterns

Skypatterns

***Dominance relation = inclusion**



Maximal patterns

Closed patterns*

**Dominance
programming for
optimal patterns**

Top-k patterns

Skypatterns

***Dominance relation = inclusion at same frequency**



Maximal patterns

Closed patterns

**Dominance
programming for
optimal patterns**

Top-k patterns*

Skypatterns

***Dominance relation = order induced by the interestingness measure**



Maximal patterns

Closed patterns

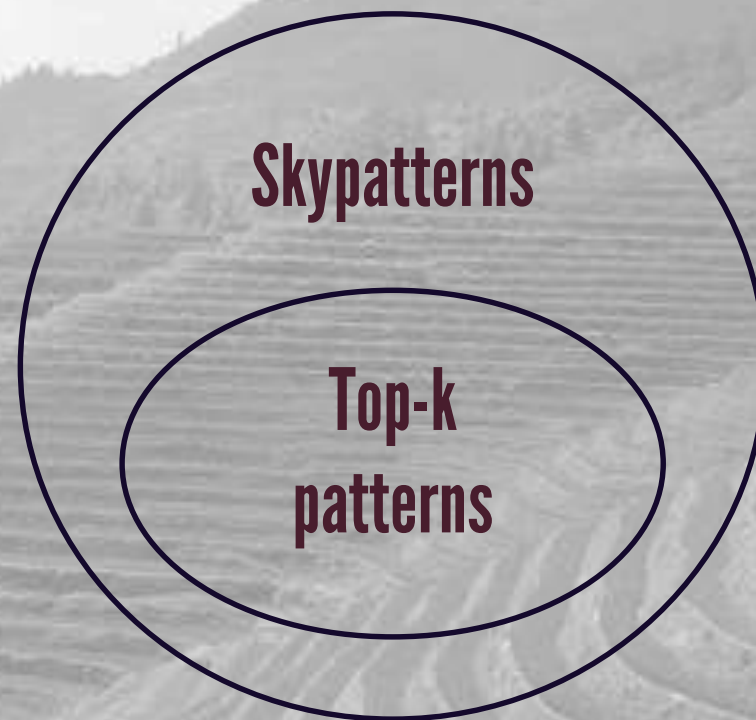
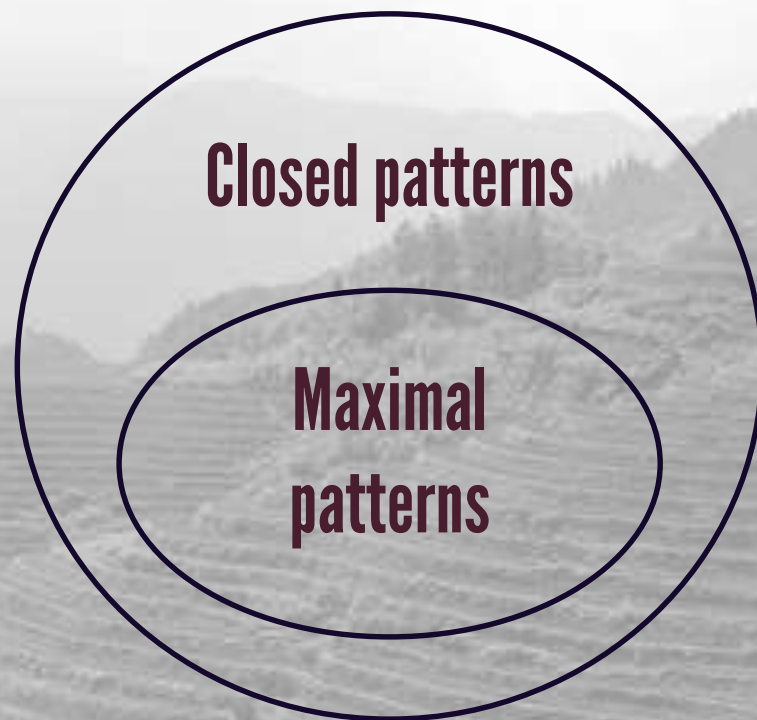
**Dominance
programming for
optimal patterns**

Top-k patterns

Skypatterns*

***Dominance relation = Pareto domination**

**Dominance
programming for
optimal patterns**



Pattern sampling

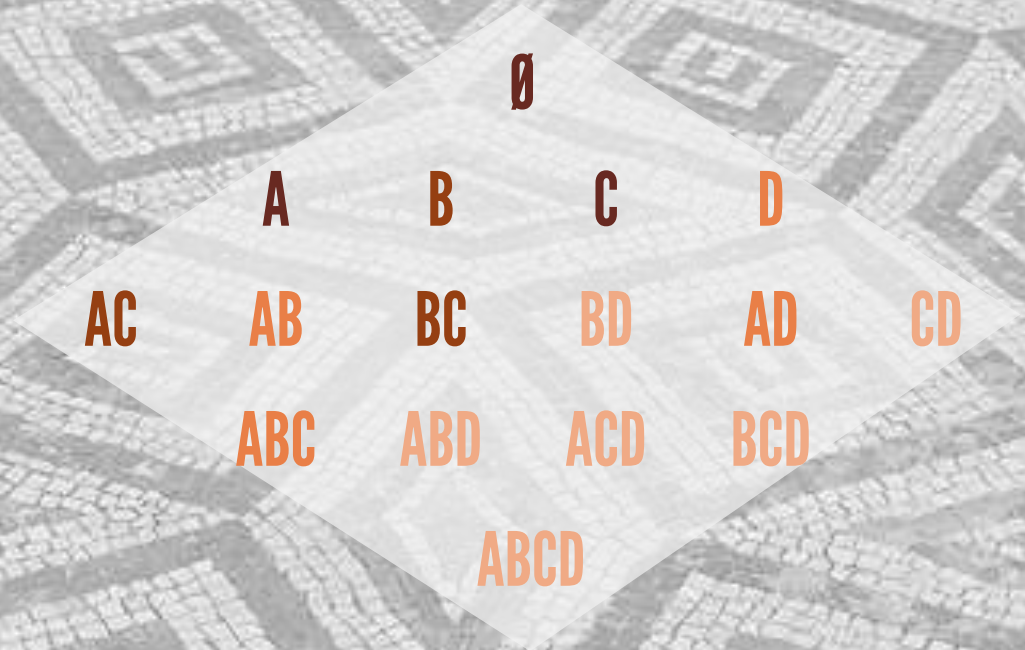
MUSK
2009

Local pattern sampling
2011



*Sampling

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D



Pattern A (freq = 4)
has twice more
chance to be drawn
than pattern D
(freq = 2)

Pattern sampling

*Picking k patterns randomly with a probability proportional to an interestingness measure

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

Stochastic methods [Hasan and Zaki, 2009]

Random walk on lattice

Two-step direct method [Boley et al., 2011]

Pick a transaction + pick an itemset of this

Pattern sampling transaction

***Two main families of methods**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

Stochastic methods [Hasan and Zaki, 2009]

Random walk on lattice

Two-step direct method [Boley et al., 2011]*

Pick a transaction + pick an itemset of this

Pattern sampling transaction

***More uniform**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D



\emptyset, A, C, AC



\emptyset, A, D, AD



$\emptyset, A, B, C, AB, AC, BC, ABC$



\emptyset, B, C, BC



$\emptyset, A, B, C, D, AB, AC, AD, BC, BD, CD, ABC, ABD, ACD, BCD, ABCD$

**Direct pattern
sampling**

[Boley et al., 2011]

***Consider all itemsets contain in each transaction**

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D



\emptyset , **A**, C, AC



\emptyset , **A**, **D**, AD



\emptyset , **A**, B, C, AB, AC, BC, ABC



\emptyset , B, C, BC



\emptyset , **A**, B, C, **D**, AB, AC, AD, BC, BD, CD, ABC, ABD, ACD, BCD, ABCD

Direct pattern sampling

[Boley et al., 2011]

Pattern A (freq = 4) appears twice more than pattern D (freq = 2)

*Consider all itemsets contain in each transaction

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D



4 $\emptyset, \mathbf{A}, C, AC$



4 $\emptyset, \mathbf{A}, \mathbf{D}, AD$



8 $\emptyset, \mathbf{A}, B, C, AB, AC, BC, ABC$



4 \emptyset, B, C, BC



16 $\emptyset, \mathbf{A}, B, C, \mathbf{D}, AB, AC, AD, BC, BD, CD, ABC, ABD, ACD, BCD, ABCD$

Direct pattern
sampling

[Boley et al., 2011]

Pattern A (freq = 4)
has twice more
chance to be drawn
than pattern D
(freq = 2)

*Count the number of itemsets

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D



1/9 $\emptyset, \mathbf{A}, C, AC$



1/9 $\emptyset, \mathbf{A}, \mathbf{D}, AD$



2/9 $\emptyset, \mathbf{A}, B, C, AB, AC, BC, ABC$



1/9 \emptyset, B, C, BC



4/9 $\emptyset, \mathbf{A}, B, C, \mathbf{D}, AB, AC, AD, BC, BD, CD, ABC, ABD, ACD, BCD, ABCD$

Direct pattern
sampling

[Boley et al., 2011]

Pattern A (freq = 4)
has twice more
chance to be drawn
than pattern D
(freq = 2)

*Normalize

	Troubled Romantic	Rich	Dies	Hidding Secret
	A		C	
	A			D
	A	B	C	
		B	C	
	A	B	C	D

Direct pattern sampling

[Boley et al., 2011]



1/9 \emptyset, A, C, AC



1/9 \emptyset, A, D, AD



2/9 $\emptyset, A, B, C, AB, AC, BC, ABC$



1/9 \emptyset, B, C, BC



4/9 $\emptyset, A, B, C, D, AB, AC, AD, BC, BD, CD, ABC, ABD, ACD, BCD, ABCD$

Pattern A (freq = 4) has twice more chance to be drawn than pattern D (freq = 2)

- *Pick a transaction proportionally to the distribution
- **Pick uniformly an itemset within this transaction

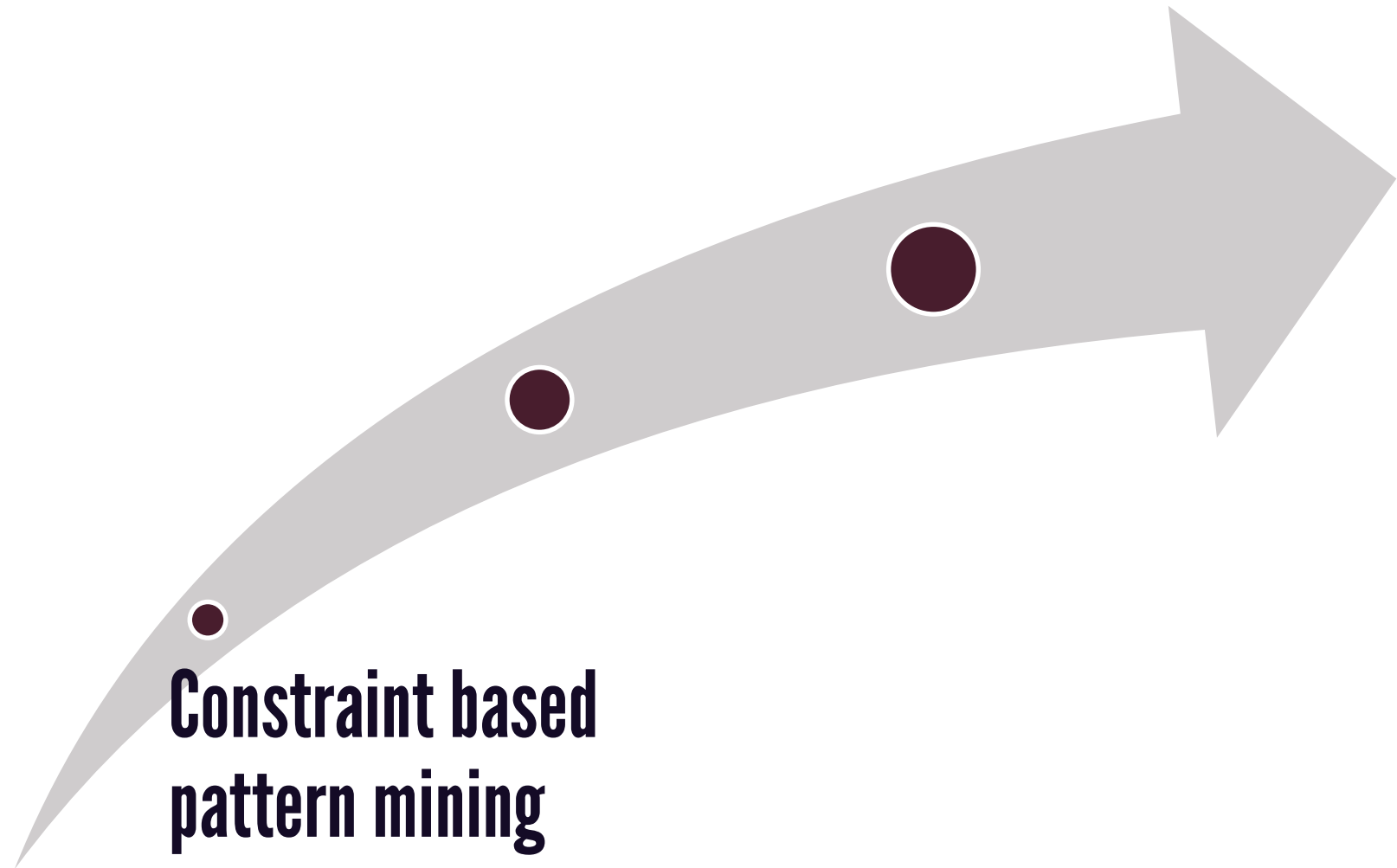
Pattern sampling



Compact
Threshold free
Diversity
Very fast

**Patterns far from
optimality**

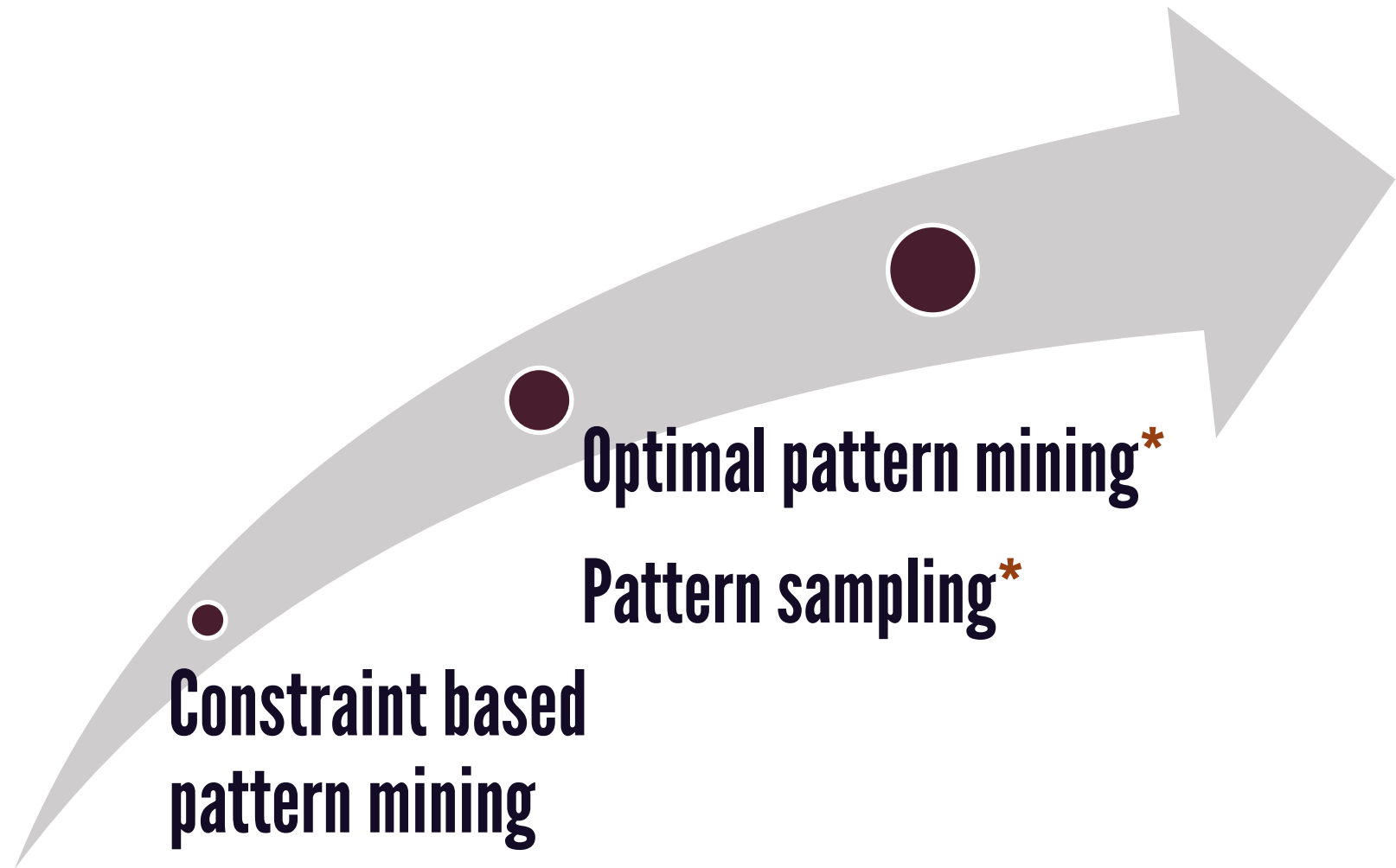
Ease of use



**Constraint based
pattern mining**

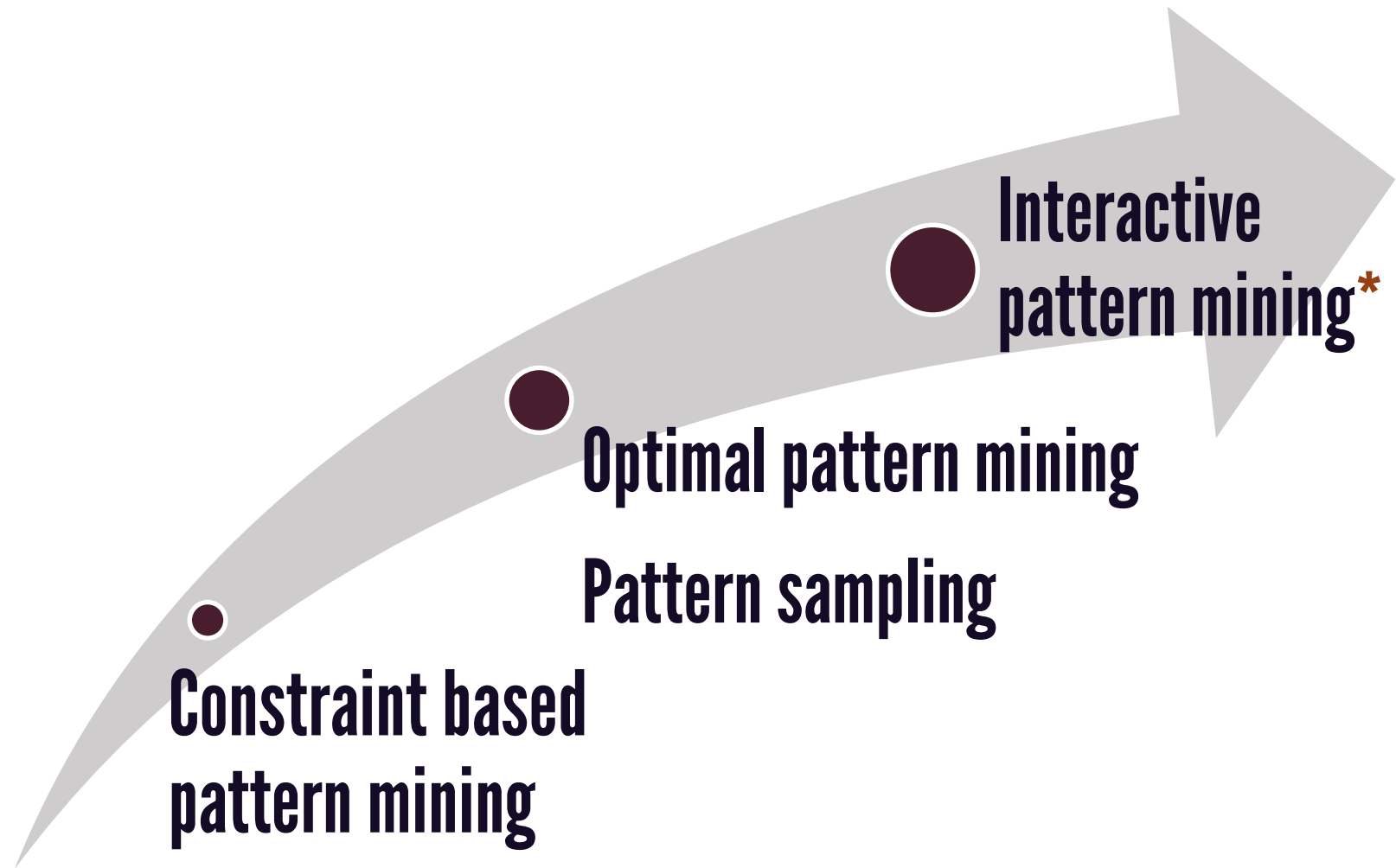
***No algorithm specification**

Ease of use



***No user-specified threshold**

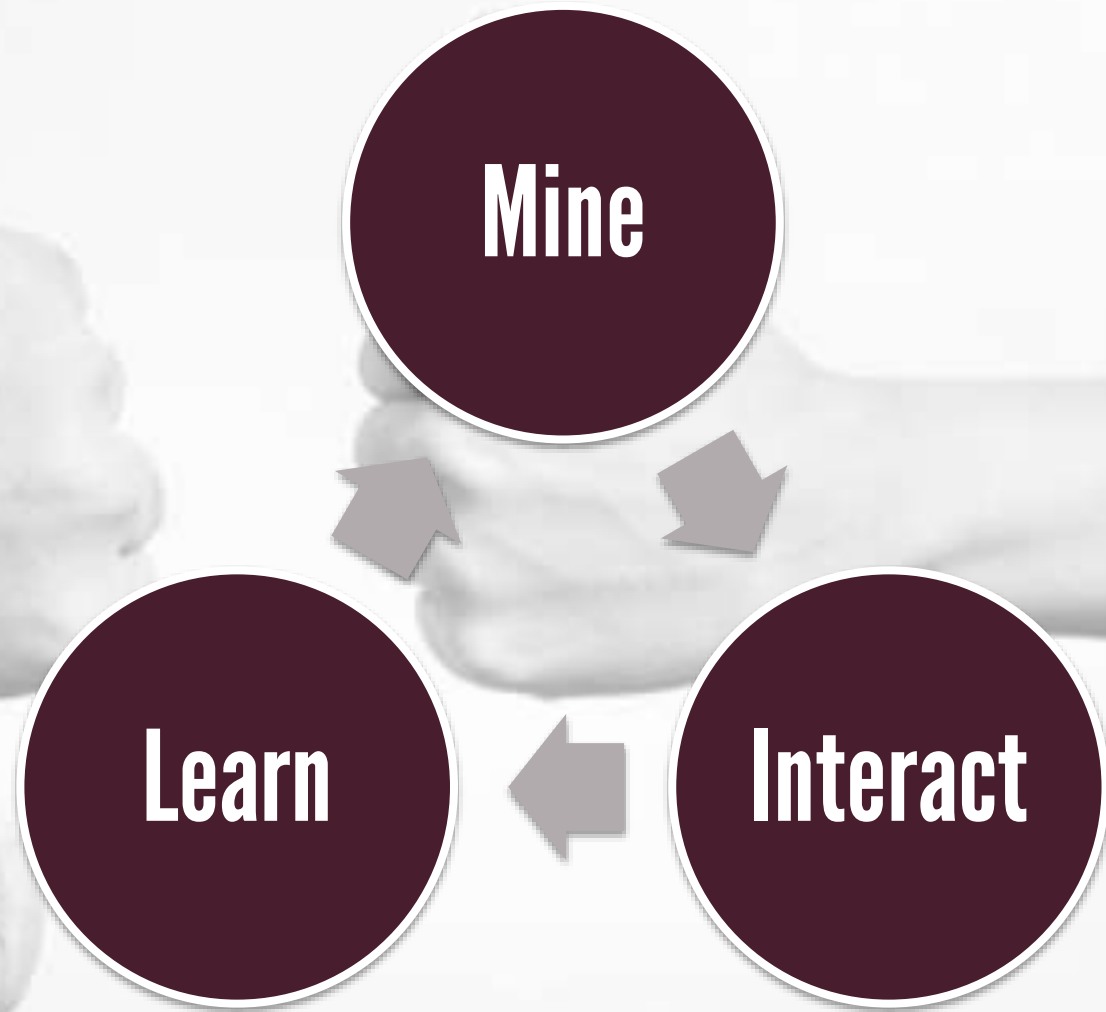
Ease of use



Pattern sampling

***No user-specified measure**

**Interactive data
exploration using
pattern mining**
[van Leeuwen 2014]



**Interactive data
exploration using
pattern mining
[van Leeuwen 2014]**

Feedback integration

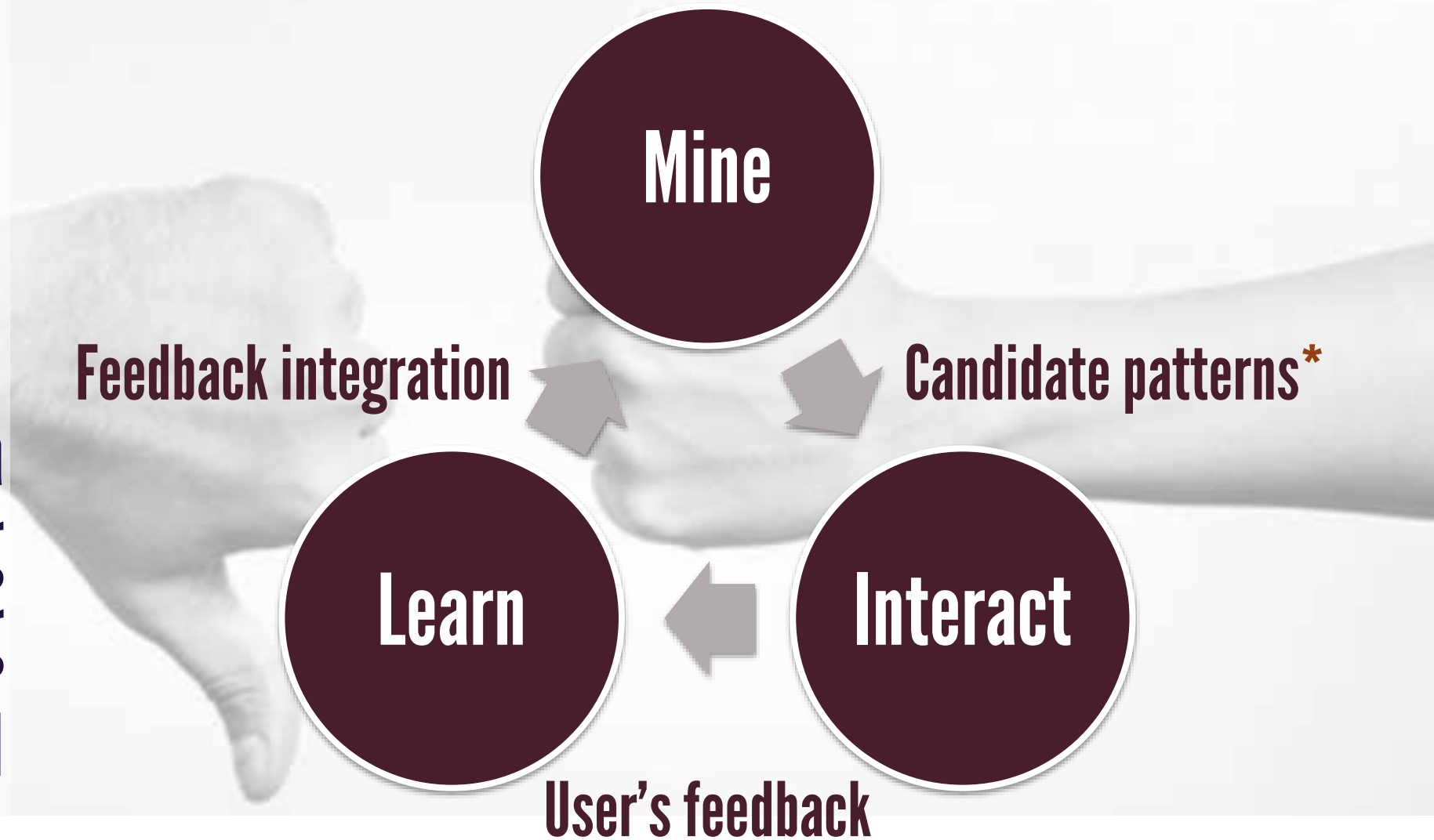


Candidate patterns



User's feedback

**Interactive data
exploration using
pattern mining**
[van Leeuwen 2014]



***Active learning vs useful pattern mining**

**Interactive data
exploration using
pattern mining**
[van Leeuwen 2014]

Feedback integration



Candidate patterns



User's feedback*

***Explicit feedback vs implicit feedback**

**Interactive data
exploration using
pattern mining**
[van Leeuwen 2014]

Feedback integration*




Candidate patterns



User's feedback

***How to update the target of the mining method?**




**Discovering Interesting Patterns Through User's
Interactive Feedback [Xin et al., 2006]***

**Interactive Pattern Mining on Hidden Data: A Sampling-
based Solution [Bhuiyan et al., 2012]****

Mining step? **Active Preference Learning for Ranking Patterns [Dzyuba
et al., 2013]****

***Offline mining of all frequent patterns**

****Online mining by integrating preferences**



**Discovering Interesting Patterns Through User's
Interactive Feedback [Xin et al., 2006]**

**Interactive Pattern Mining on Hidden Data: A Sampling-
based Solution [Bhuiyan et al., 2012] ***

**Mining step? Active Preference Learning for Ranking Patterns [Dzyuba
et al., 2013] ****

***Pattern sampling**

****Optimal pattern mining via beam search**



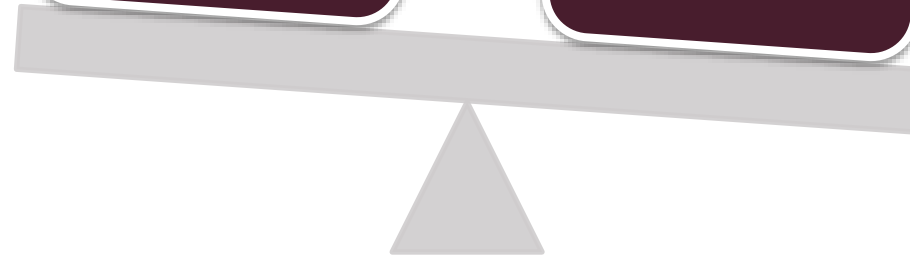
**Optimal pattern
mining**


**Pattern
sampling**

**Best
patterns**

Very fast

Diversity





**Discovering Interesting Patterns Through User's
Interactive Feedback [Xin et al., 2006]***

**Interactive Pattern Mining on Hidden Data: A Sampling-
based Solution [Bhuiyan et al., 2012]****

**Learning step? Active Preference Learning for Ranking Patterns [Dzyuba
et al., 2013]***

***Ranking over all patterns = learning to rank problem**

****Weight on items**

Conclusion

Frequent pattern mining
1990s

Constraint-based
pattern mining
2000s

Optimal pattern
mining
Early 2010s

Declarative pattern
mining
Early 2010s

Interactive pattern
mining
Now

Retrieval era

Exploratory analysis era

Performance issue*

The more, the better

Data-driven

Quality issue

The less, the better

User-driven

*Faster

Conclusion

Frequent pattern mining
1990s

Constraint-based
pattern mining
2000s

Optimal pattern
mining
Early 2010s

Declarative pattern
mining
Early 2010s

Interactive pattern
mining
Now

Retrieval era

Exploratory analysis era

Performance issue

Quality issue*

The more, the better

The less, the better*

Data-driven

User-driven

*Faster, better

Conclusion

Frequent pattern mining
1990s

Constraint-based
pattern mining
2000s

Optimal pattern
mining
Early 2010s

Declarative pattern
mining
Early 2010s

Interactive pattern
mining
Now

Retrieval era

Exploratory analysis era

Performance issue

The more, the better

Data-driven

Quality issue

The less, the better*

User-driven*

*Faster, better, easier



Bruno Crémilleux, François Rioult



Arnaud Giacometti, Dominique Li, Patrick Marcel



Marc Plantevit

Acknowledgements



Chedy Raïssi



Thank you!