Recent Developments in Pattern Mining

Toon Calders

Discovery Science, October 30th 2012, Lyon

Outline

- Frequent Itemset Mining
  - Pattern Explosion Problem
  - Condensed Representations
    - Closed itemsets
    - Non-Derivable Itemsets
- Recent Approaches Towards Non-Redundant Pattern Mining
- Relations Between the Approaches

Association Rules

Minsup = 60%
Minconf = 80%

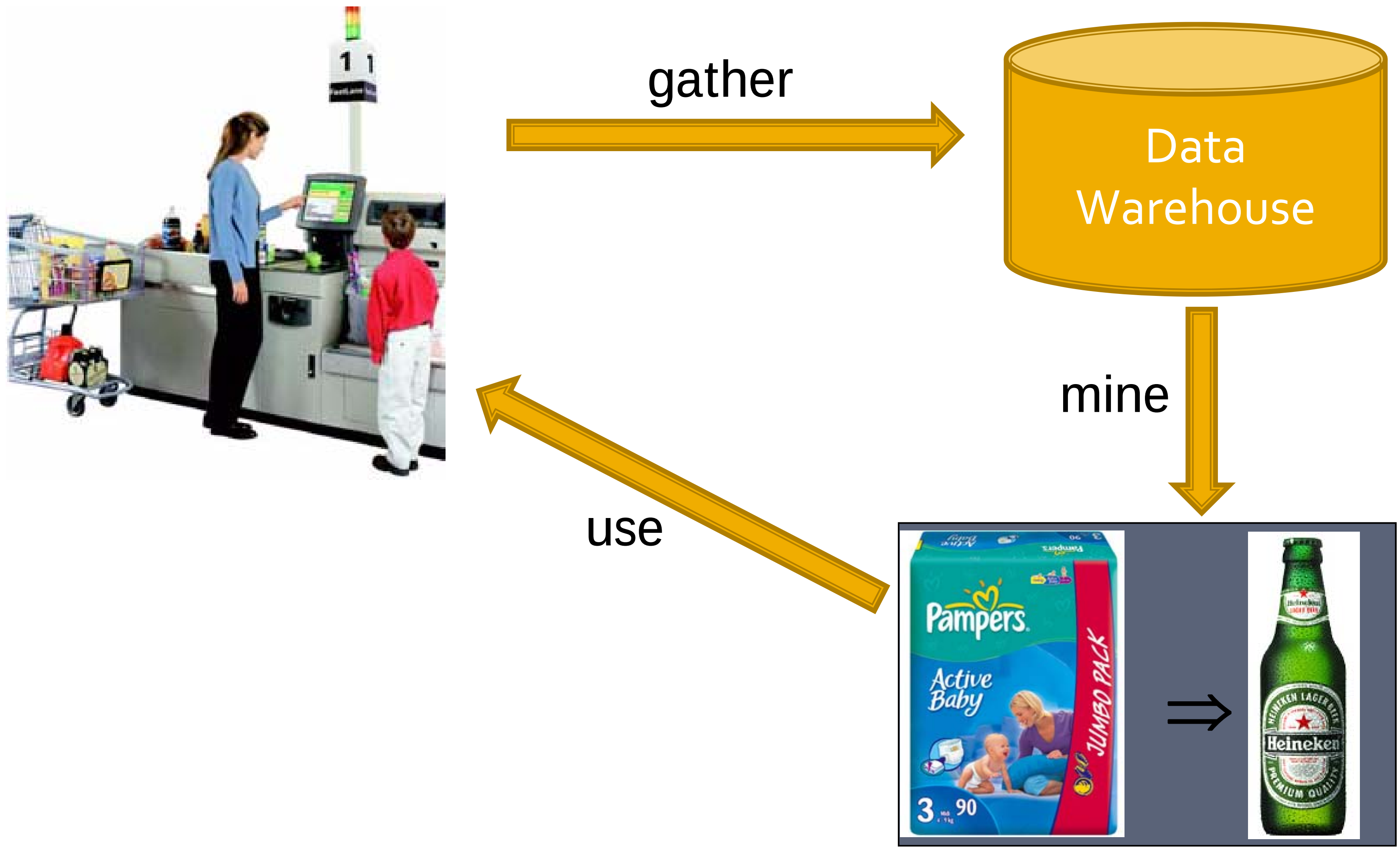
TID	Item
1	A,B,C,D
2	B,C,D
3	A,C,D
4	B,C,D
5	B,C

set	support
~~A~~	~~2~~
B	4
C	5
D	4

BD → C	100%
C → D	80%
D → C	100%
C → B	80%
B → C	100%

Mining **association rules** between sets of items in large databases
R **Agrawal**, T Imieliński, A Swami - ACM SIGMOD Record, 1993 - dl.acm.org
Cited by 11735

What We Promised

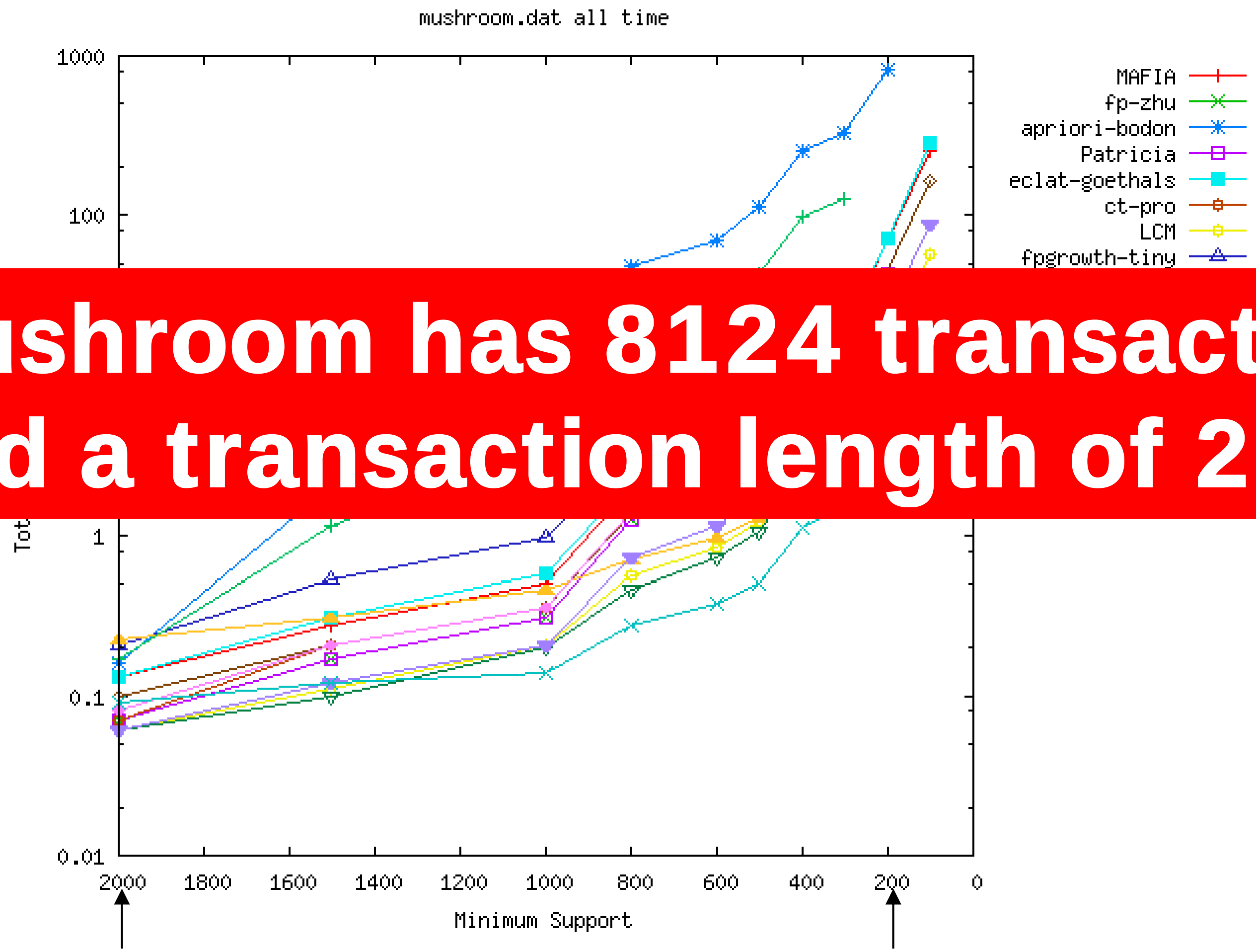


Popularity of the Topic

- Association rules gaining popularity

- Literally hundreds of algorithms:
  AIS, Apriori, AprioriTID, AprioriHybrid, FPGrowth, FPGrowth*, Eclat, dEclat, Pincer-search, ABS, DCI, kDCI, LCM, AIM, PIE, ARMOR, AFOPT, COFI, Patricia, MAXMINER, MAFIA, …

Pattern Explosion Problem



Mushroom has 8124 transactions, and a transaction length of 23

Over 50 000 patterns

Over 10 000 000 patterns

What We Actually Did

patterns

Redundancy Problem

- Frequent itemset / Association rule mining
  = find all itemsets / ARs satisfying thresholds

- Many are redundant

  smoker → lung cancer
  smoker, bald → lung cancer
  pregnant → woman
  pregnant, smoker → woman, lung cancer

Outline

- Frequent Itemset Mining
  - Pattern Explosion Problem
  - Condensed Representations
    - Closed itemsets
    - Non-Derivable Itemsets
- Recent Approaches Towards Non-Redundant Pattern Mining
- Relations Between the Approaches

Condensed Representations

A1	A2	A3	B1	B2	B3	C1	C2	C3
1	1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1

- Number of frequent itemsets = 21
- Need a compact representation

Discovering frequent closed itemsets for association rules

N **Pasquier**, Y Bastide, R Taouil, L Lakhal - Database Theory—ICDT'99, 1999

Condensed Representations

- Condensed Representation:
  "Compressed" version of the collection of all frequent itemsets (usually a subset) that allows for lossless regeneration of the complete collection.
  - Closed Itemsets (Pasquier et al, ICDT 1999)
  - Free Itemsets (Boulicaut et al, PKDD 2000)
  - Disjunction-Free itemsets (Bykowski and Rigotti, PODS 2001)

Condensed Representations: Reasoning with Probabilities

- How do supports interact?
- What information about unknown supports can we derive from known supports?
  - Concise representation: only store relevant part of the supports

Redundancies

- Agrawal et al. (Monotonicity)
  - $\text{Supp}(AX) \leq \text{Supp}(A)$

- Lakhal et al. (Closed sets)
  Boulicaut et al. (Free sets)
  - **If** $\text{Supp}(A) = \text{Supp}(AB)$
    Then $\text{Supp}(AX) = \text{Supp}(AXB)$

Redundancies

- Bayardo (MAXMINER)
  - Supp(ABX) ≥ Supp(AX) – $\underline{\text{(Supp(X)-Supp(BX))}}$

drop (X, B)

- Bykowski, Rigotti (Disjunction-free sets)

if Supp(ABC) = Supp(AB) + Supp(AC) – Supp(A)

then

Supp(ABCX) = Supp(ABX) + Supp(ACX) – Supp(AX)

Tight Bounds on Support

- General problem:
  - Given some supports, what can be derived for the supports of other itemsets?

Example:

$\text{supp}(AB) = 0.7$
$\text{supp}(BC) = 0.5$

$\text{supp}(ABC) \in [\ ?, ?\ ]$

Tight Bounds on Support

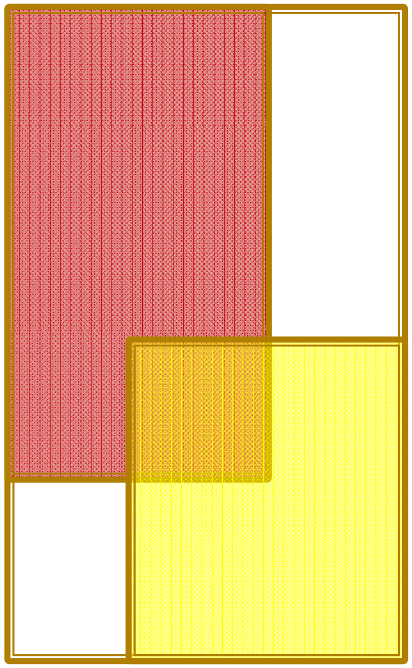
- General problem:
  - Given some supports, what can be derived for the supports of other itemsets?

Example:

supp(AB) = 0.7

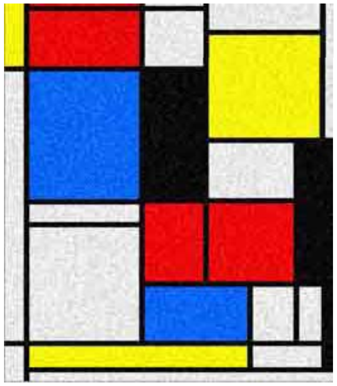
supp(BC) = 0.5

supp(ABC) $\in$ [0.2, 0.5]



Tight Bounds on Support

- The problem of finding tight bounds is hard to solve in general



Theorem

The following problem is NP-complete:

Given *itemsets I1, …, In, and supports s1, …, sn,*

Does there exist *a database D such that:*

for $j=1\ldots n$, $supp(I_j) = s_j$

Tight Bounds on Support

- Can be translated into a linear program
  - Introduce variable X_J for every itemset J
    $X_J \equiv$ fraction of transactions with items = J

TID	Items
1	A
2	C
3	C
4	A,B
5	A,B,C
6	A,B,C

Tight Bounds on Support

- Can be translated into a linear program
  - Introduce variable X_J for every itemset J
    $X_J \equiv$ fraction of transactions with items = J

TID	Items
1	A
2	C
3	C
4	A,B
5	A,B,C
6	A,B,C

$X_{\{\}} = 0$
$X_A = 1/6$
$X_B = 0$
$X_C = 2/6$
$X_{AB} = 1/6$
$X_{AC} = 0$
$X_{BC} = 0$
$X_{ABC} = 2/6$

Tight Bounds on Support

Give bounds on ABC

For a database D

In which
  supp(AB) = 0.7
  supp(BC) = 0.5

Minimize/maximize X_{ABC}

s.t.

$X_{\{\}}+X_A+X_B+X_C+X_{AB}+X_{AC}$
  $+X_{BC}+X_{ABC} = 1$

$X_{\{\}}, X_A, X_B, X_C, \ldots, X_{ABC} \geq 0$

$X_{AB}+X_{ABC} = 0.7$

$X_{BC}+X_{ABC} = 0.5$

Derivable Itemsets

- Given: Supp(I) **for all I $\subset$ J**
  Give tight [l,u] for J

Can be computed efficiently

- Without counting : Supp(J) $\in$ [l,u]
- J is a ***derivable itemset*** (DI) iff l = u
  - We **know** Supp(J) **exactly** without counting!

Summary – Condensed Rep's

- Considerably smaller than all frequent itemsets
  - Many redundancies removed
  - There exist efficient algorithms for mining them

- Yet, still way too many patterns generated
  - supp(A) = 90%, supp(B)=20%
    supp(AB) $\in$ [10%,20%]
    yet, supp(AB) = 18% not interesting

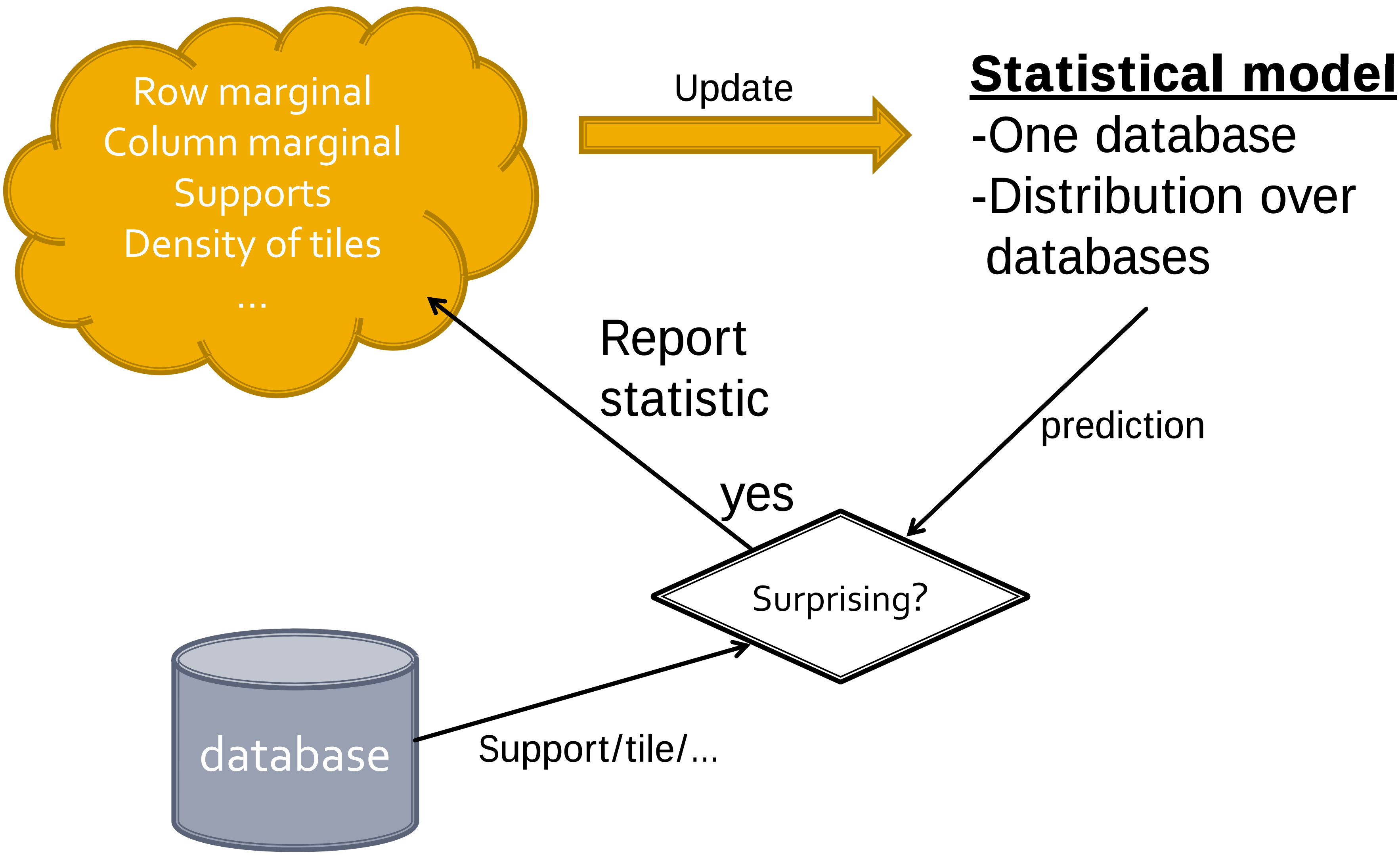
Outline

- Frequent Itemset Mining
- Recent Approaches Towards Non-Redundant Pattern Mining
  - Statistically based
  - Compression based
- Relations Between the Approaches

Statistical Approaches

- We have background knowledge
  - Supports of some itemsets
  - Column/row marginals
- Influences our “expectation” of the database
  - Not every database equally likely
- Surprisingness:
  - How does real support correspond to expectation?

Statistical Approaches



Different Variants

- Types of background knowledge
  - Supports, marginals, densities of regions
- Mapping background knowledge to statistical model
  - Distribution over databases; one distributions representing a database
- Way of computing surprisingness

Types of Background Knowledge

- Row and column marginals

A	B	C	
0	0	0	0
0	1	1	2
0	1	1	2
1	1	0	2
1	0	0	1
1	1	1	3
3	3	3	

Column marginals

Row marginals

Types of Background Knowledge

- Row and column marginals

A	B	C	Row marginals
?	?	?	0
?	?	?	2
?	?	?	2
?	?	?	2
?	?	?	1
?	?	?	3
3	3	3	

Column marginals

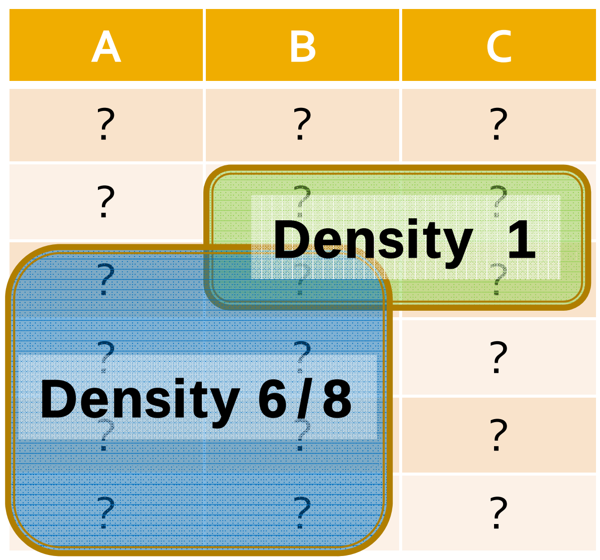
Types of Background Knowledge

- Density of tiles

A	B	C
0	0	0
0	1	1
0	1	1
1	1	0
1	0	0
1	1	1

Types of Background Knowledge

- Density of tiles



Statistical Model - Uniform

- Consider all databases that satisfy the constraints
- Uniform distribution over these databases
  - Gionis et al: row and column marginals
  - Hanhijärvi et al: extension to supports

A. Gionis, H. Mannila, T. Mielikäinen, P. Tsaparas: Assessing data mining results via swap randomization. TKDD 1(3): (2007)

S. Hanhijärvi, M. Ojala, N. Vuokko, K. Puolamäki, N. Tatti, H. Mannila: Tell Me Something I Don’t Know: Randomization Strategies for Iterative Data Mining. ACM SIGKDD (2009)

Statistical Model - Uniform

$$\begin{array}{l} 1\ 1\ 1 \rightarrow 3 \\ 1\ 1\ 1 \rightarrow 3 \\ 0\ 1\ 1 \rightarrow 2 \\ 1\ 0\ 0 \rightarrow 1 \\ 0\ 1\ 0 \rightarrow 1 \\ \downarrow\ \downarrow\ \downarrow \\ 3\ 4\ 3 \end{array} \quad \Longrightarrow \quad \text{supp(BC)} = 60\%$$

- Is this support surprising given the marginals?

Statistical Model - Uniform

1 1 1
1 1 1
0 1 1
1 0 0
0 1 0

supp(BC) = 60%

1 1 1
1 1 1
1 1 0
0 1 0
0 0 1

supp(BC) = 40%

1 1 1
1 1 1
0 1 1
1 0 0
0 1 0

supp(BC) = 60%

1 1 1
1 1 1
0 1 1
0 1 0
1 0 0

supp(BC) = 60%

1 1 1
1 1 1
1 0 1
0 1 0
0 1 0

supp(BC) = 40%

Statistical Model - Uniform

1 1 1
1 1 1
0 1 1
1 0 0
0 1 0

→ supp(BC) = 60%

- Is this support surprising given the marginals?
  No!
  - p-value = P(supp(BC) ≥ 60% | marginals) = 60%
  - E[supp(BC)] = 60% x 60% + 40% x 40% = 52%

Statistical Model - Uniform

- Estimation of p-value via simulation (MC)
- Uniform sampling from databases with same marginals is non-trivial
  - MCMC

1	1	1		1	1	1
1	1	1		1	1	1
0	1	1	$\Rightarrow$	1	0	1
1	0	0		0	1	0
0	1	0		0	1	0

Statistical Model - Uniform

1 1 1
1 1 1
1 1 0
0 0 1
0 1 0

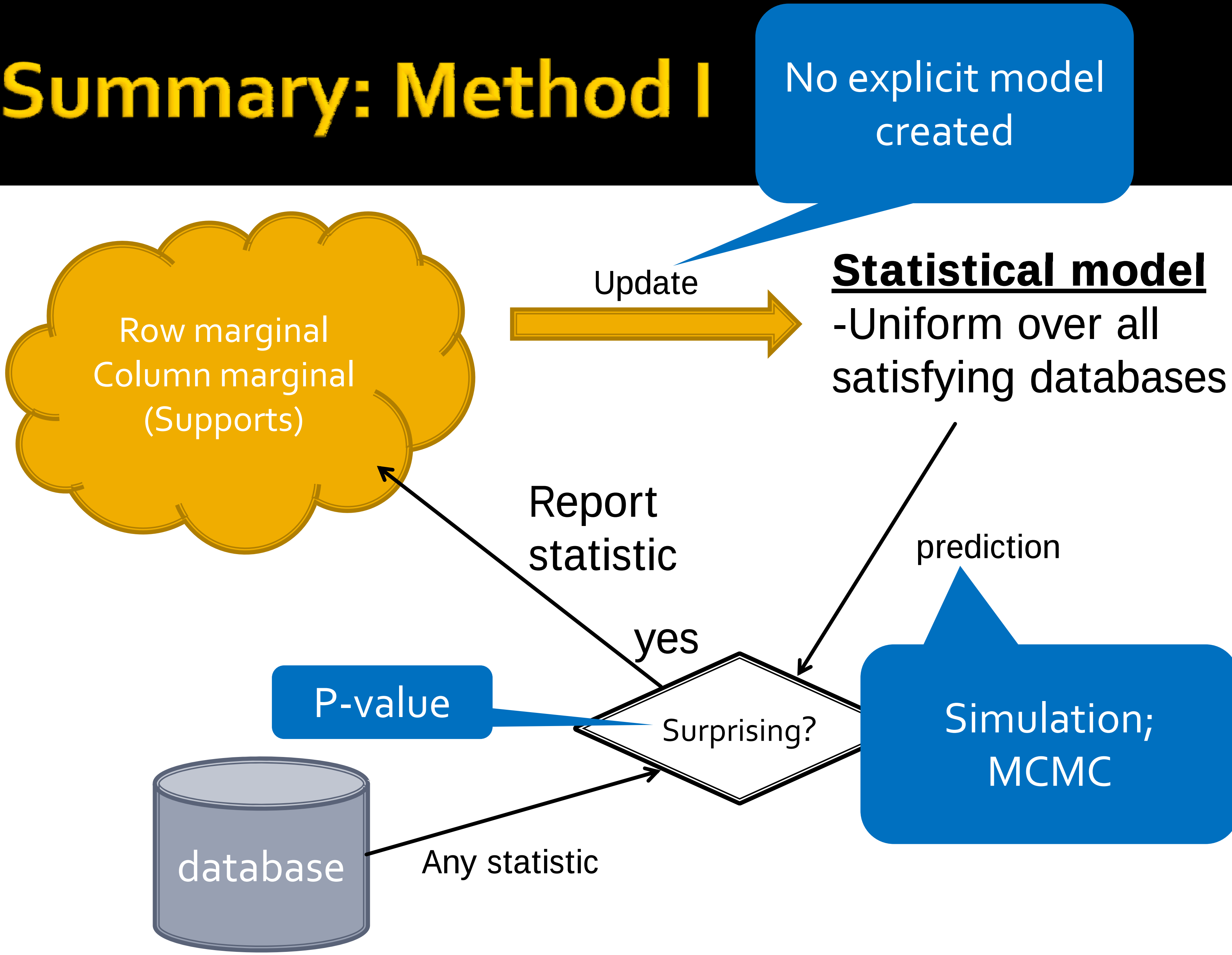
1 1 1
1 1 1
1 1 0
0 1 0
0 0 1

1 1 1
1 1 1
0 1 1
1 0 0
0 1 0

1 1 1
1 1 1
1 0 1
0 1 0
0 1 0

1 1 1
1 1 1
0 1 1
0 1 0
1 0 0

Summary: Method I



Statistical Model II – MaxEnt

- Database ≡ probability distribution
- $p(t=X) = |\{ t \in D \mid t=X \}|/|D|$
- Pick the one with maximal entropy
  - $H(p) = -\Sigma_X\, p(t=X) \log(p(t=X))$

Example:

supp(A) = 90%

supp(B) = 20%

A	B	prob
0	0	10%
0	1	0%
1	0	70%
1	1	20%

H = 1.157

A	B	Prob
0	0	0%
0	1	10%
1	0	80%
1	1	10%

H = 0.992

A	B	prob
0	0	8%
0	1	2%
1	0	72%
1	1	18%

H = 1.19

Why MaxEntropy?

- $H(p) = -\Sigma_X \, p(t{=}X) \log(p(t{=}X))$
  - $-\log(p(t{=}X))$
    denotes space required to encode X, given an optimal Shannon encoding for the distribution p; characterizes the *information content of X*
  - p(t=X) denotes the probability that event t=X occurs
  - H(p) = expected number of bits needed to encode transactions

Why MaxEntropy?

principle of maximum entropy

- if nothing is known about a distribution except that it belongs to a certain class, pick distribution with the largest entropy. Maximizing entropy minimizes the amount of prior information built into the distribution.

Statistical Model II – MaxEnt

- How to compute the MaxEnt distribution?
  - Recall: linear programming formulation

$$\mathbf{MAX(-\ X_{\{\}}\ log(X_{\{\}})\ -\ X_A\ log(X_A)}$$
$$\mathbf{X_B\ log(X_B)\ -\ X_{AB}\ log(X_{AB}))}$$

$$X_{\{\}} + X_A + x_B + X_{AB} = 1$$
$$X_{\{\}}, X_A, x_B, X_{AB} \geq 0$$

$$X_A + X_{AB} = 0.9$$
$$X_B + X_{AB} = 0.2$$

Statistical Model II – MaxEnt

THEOREM 2 (THEOREM 3.1 IN [4]). *Given a collection of itemsets* $\mathcal{C} = \{X_i\}_{i=1}^{k}$ *with frequencies* $fr(X_i)$, *let us define* $\mathcal{P} = \{p \mid p(X_i = 1) = fr(X_i)\}$. *If there is a distribution in* $\mathcal{P}$ *that has only non-zero entries, then the maximum entropy distribution* p^* *can be written as*

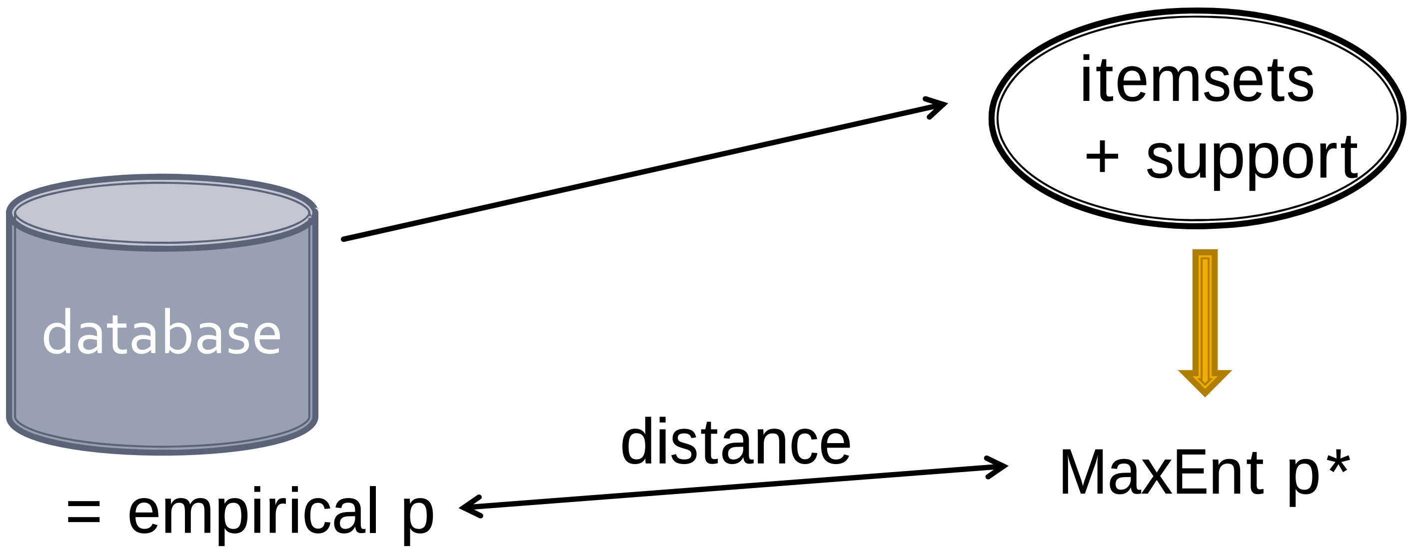
$$p^*(A = t) = u_0 \prod_{X \in \mathcal{C}} u_X^{S_X(t)},$$

where $u_X \in \mathbb{R}$, *and* u_0 *is a normalization factor.*

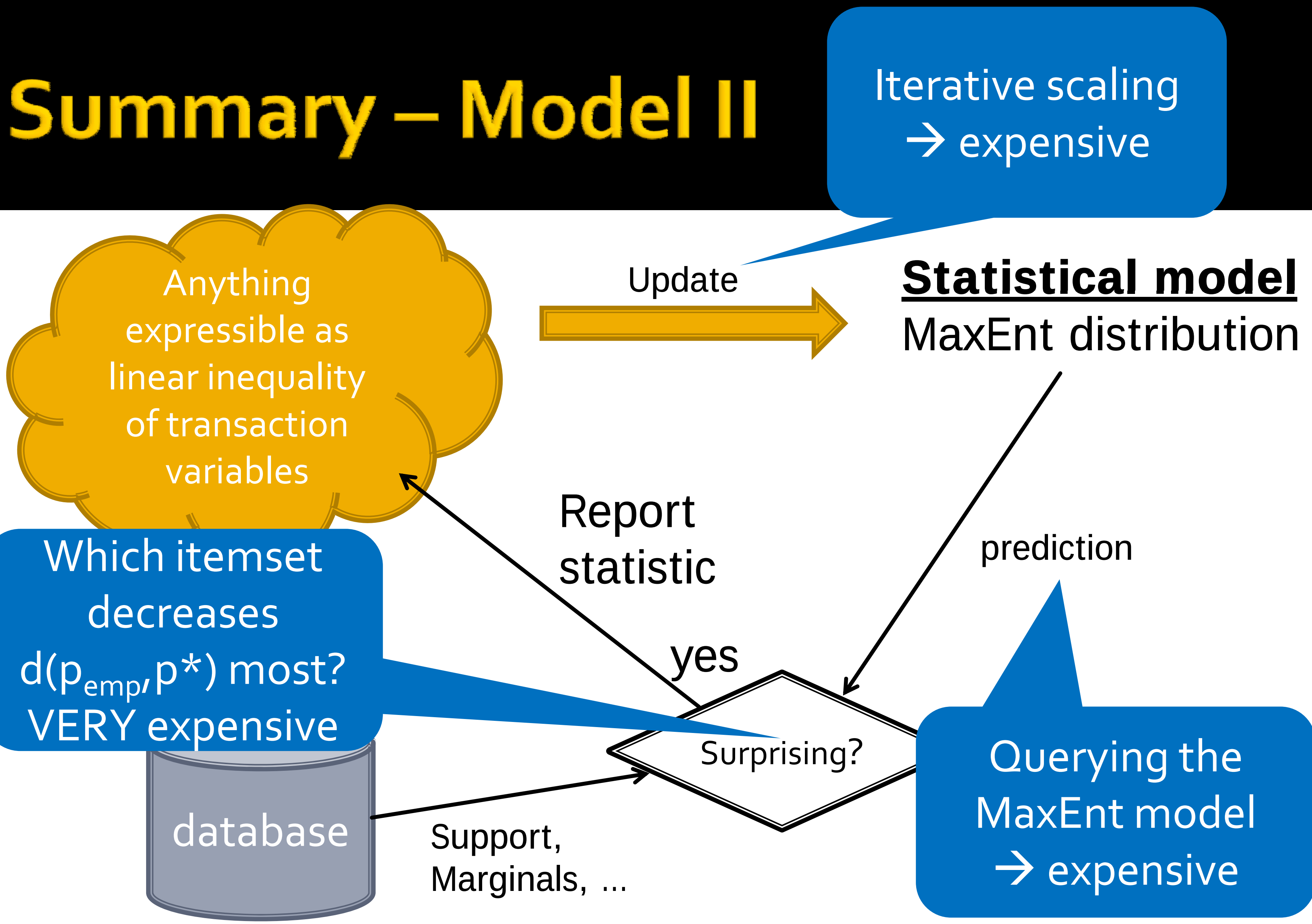
- Get u_0, and u_X for all X → *iterative scaling*
- Works with any constraint that can be expressed as lin. ineq. of transaction variables

Statistical Model II – MaxEnt

- Score a collection of itemsets:
  - Build MaxEnt distribution for these itemsets
  - Compare to empirical distribution
    - E.g., Kullback-Leibler divergence, BIC, …



Summary – Model II



Michael Mampaey, Nikolaj Tatti, Jilles Vreeken: Tell me what i need to know: succinctly summarizing data with itemsets. KDD 2011: 573-581

Statistical Model III – MaxEnt

- Original database is n x m
  - Consider all 0-1 databases of size n x m
  - Every database has a probability
    ➔ distribution over databases
- $E(supp(J)) = \sum_D p(D)\, supp(J,D)$
- Select distribution p that maximizes entropy and satisfies the constraints *in expectation*

Tijl De Bie: Maximum entropy models and subjective interestingness: an application to tiles in binary databases. DMKD Vol. 23(3): 407-446 (2011)

Statistical Model III – MaxEnt

- Depending on the type of constraints finding MaxEnt distribution is easy; e.g.,
  - density of a given tile
  - row and column marginals
  - Anything expressible as a linear constraint in the variables D[i,j]

- Does not work for frequency constraints!
  - supp(ab) = 5 ➔ D[1,a]*D[1,b] + D[2,a]*D[2,b] + … = 5

Summary – Statistical Methods

- Depending on background knowledge → expectation underlying database changes
- Different ways to model
  - Uniform over all consistent databases
  - MaxEnt consistent database
  - Satisfy constraints in expectation; MaxEnt distribution over all databases

Summary – Statistical Methods

- All models have pro and cons
  - Uniform is hard to extend to new types of constraints
  - MaxEnt approaches easier to extend, as long as constraints can be expressed linearly
  - All approaches are extremely computationally demanding
    - MaxEnt II seems most realistic

Outline

- Frequent Itemset Mining
- Recent Approaches Towards Non-Redundant Pattern Mining
  - Statistically based
  - Compression based
- Relations Between the Approaches

Minimal Description Length

- A good model helps us to compress the data and is compact
  - Let L(M) be the description length of the model,
  - Let L(D|M) be the size of the data when compressed by the model

- Find a model M that minimizes:
  $$L(M) + L(D|M)$$
- Explicit trade-off; increasing model complexity:
  - Increases L(M),
  - Decreases L(D|M)

Minimal Description Length

- We can use patterns to code a database

TID	Items
1	A
2	C
3	C
4	A,B
5	A,B,C
6	A,B,C

pattern	code
A	1
B	2
C	3
AB	4

L(M)

TID	Items
1	1
2	3
3	3
4	4
5	3,4
6	3,4

L(D|M)

- Find set of patterns that minimizes L(M)+L(D|M)

Minimal Description Length

- Rank itemsets according to how well they can be used to compress the dataset
  - Property of a set of patterns
- The "Krimp" algorithm was the first to use this paradigm in itemset mining
  - Assumes a seed set of patterns
  - A subset of these patterns is selected to form the "code book"
  - The best codebook is the one that gives the best compression

[Krimp: mining itemsets that compress]

J Vreeken, M van Leeuwen, A Siebes - Data Mining and Knowledge …, 2011

Minimal Description Length

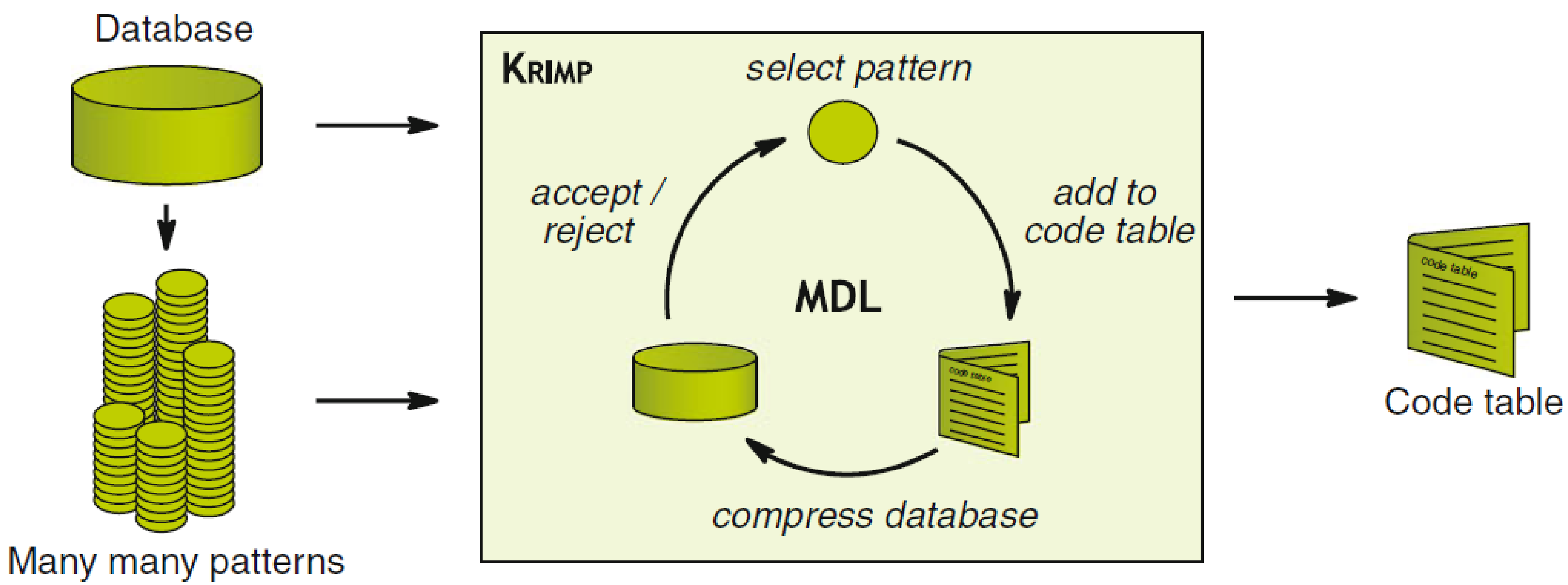


Fig. 4 KRIMP in action

Figure of Vreeken et al.

Summary MDL-Based Methods

- Select set of patterns that best compresses the dataset as the result
  - Model of the dataset; the main “building blocks”
  - Patterns will have little overlap → transaction partially covered by AB benefits little from ABC
  - Returned patterns are useful to describe the data

Summary MDL-Based Methods

- MDL method is NOT parameter-free!
  - Way of encoding has a great influence on the result
  - Encoding exploits patterns one expects to see
    - E.g., Encode errors explicitly?
- In most cases:
  - Finding best set of patterns is intractable and does not allow for approximation

Outline

- Frequent Itemset Mining
- Recent Approaches Towards Non-Redundant Pattern Mining
- Relations Between the Approaches

Relations Between the Approaches

- Actually, the three approaches are tightly connected

- Maximum likelihood principle:
  - Prior distribution over models $P(M)$
  - Posterior distribution:

$$P(M|D) = P(D|M).P(M) / P(D)$$
$$\propto P(D|M).P(M)$$

  - Pick model that maximizes $P(M|D)$

  = model maximizing $\log(P(D|M)) + \log(P(M))$

Relations Between the Approaches

- Let $Q(D|M) = 2^{-L(D|M)}$
  - If code is optimal $Q(D|M)$ is a probability
- Otherwise: normalize
  - $W(M) := \Sigma_{D'}\ 2^{-L(D'|M)}$
- $P(D|M) := Q(D|M) / W(M)$

- Prior distribution over models:
  - $P(M) := 2^{-L(M)}\ W(M) / W$
  - $W = \Sigma_{M'}\ 2^{-L(M')}\ W(M')$

Relations Between the Approaches

- $P(D|M) := 2^{-L(D|M)} / W(M)$
  $P(M) := 2^{-L(D|M)} W(M) / W$
- Maximum likelihood principle:
  Pick M that maximizes $\log(P(D|M)\, P(M))$
  $= \log(2^{-L(D|M)} / W(M)\; 2^{-L(M)}\, W(M) / W)$
  $= -L(D|M) - L(M) - \log(W)$

- Select model minimizing
  $L(D|M) + L(M)$

Relations Between the Approaches

- Hence, encoding the model and the data given the model are "just" fancy ways of expressing distributions
  - Higher $L(D|M)$ = lower $P(D|M)$
  - $W(M)$ expresses how useful M is to encode databases
  - Higher $W(M)$ = higher $P(M)$
  - Higher $L(M)$ = lower $P(M)$

Relations Between the Approaches

- MaxEnt Model I
  - Patterns = model
  - Model → distribution p_M maximizing

    $$H(p) = -\Sigma_X p(t=X) \log(p(t=X))$$
  - Scoring the model: compare p_M to the empirical distribution
  - E.g., KL-divergence

Relations Between the Approaches

- Other way of looking at it:
  - Let's compress the database using M
  - We make an optimal code; code length for an itemset X equals $-\log(p_M(X))$
  - $L(D|M) = \Sigma_{t \in D} -\log(p_M(t))$

    $= -\Sigma_X \, p_{emp}(X) \log(p_M(X))$

$$KL(p_{emp} \,||\, p_M) = \Sigma_X \, p_{emp}(X) \log(p_{emp}(X) / p_M(X))$$

$$= L(D|M) - H(p_{emp})$$

Minimizing KL-divergence = minimizing L(D|M)

Summary: Relations

- Both statistical approach and minimal description length approach can be seen as instances of Bayesian learning
  - MDL
    - L(M) → model prior
    - L(D|M) → likelihood
  - Statistical approach
    - Probability → optimal code → encoding length

Conclusion

- Original pattern mining definition suffers from the pattern explosion problem
  - Frequency $\neq$ interestingness
  - Redundancy among patterns

- First approach: Condensed representations
  - Removing redundancies based on support interaction
  - Does not account for "expectation"

Conclusion

- Recent approaches based on statistical models
  - Background knowledge → information about underlying database
  - Influences what is surprising
- Different ways to interpret constraints
  - Uniform vs Maximal entropy
  - One database vs distribution over databases

Conclusion

- MDL-based methods
  - Use patterns to encode dataset
  - Optimize encoding length patterns + encoding length of the data given the patterns

- Essentially all methods similar in spirit in a mathematical sense
  - Different ways to encode prior distributions
  - Yet, at a practical level quite different

Future?

- Make these approaches more practical
  - Currently do not scale well
  - Look at compression algorithms
- Non-redundant patterns directly from data
  - Give up on exactness, but with guarantees
  - Exploit data size instead of fighting it
    → Converge to solution
- Extend to other pattern domains
  - Sequences, graphs, dynamic graphs

Thank you!