

Association Rules

Paulo J Azevedo

DI - Universidade do Minho,
INESC-TEC
2014-2022

MAP-i

Finding association in data

Roadmap

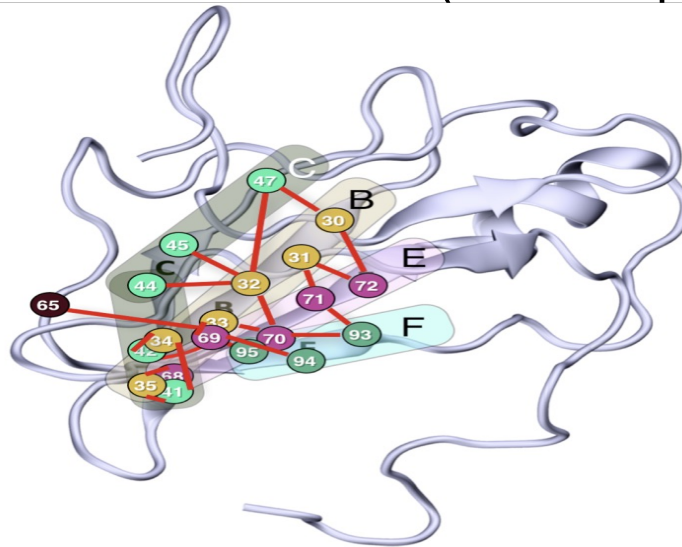
- Motivation
- Introduction to Association Rules
- Frequent Itemset Mining algorithms
 - Apriori e other Breadth-first variants
- Vertical Representations of data
 - Depth-first algorithm using vertical representation
- Interest measures,
- Pruning and selection of rules,
- Numeric attributes in association rules
- Subgroup Mining
 - Analysis of numeric properties using association rules,
 - Contrast Sets

Pattern Mining

- To identify interesting patterns in data. Patterns composed out of a specific configuration of a collection of atomic data elements.
 - {apples, oranges, yogurt} (frequent itemsets)
 - ATGCTTCGGCAA (DNA sequence)

– Graphs:

– etc...



– What is the meaning of *interesting*?

e.g. which occurs a significant number of times ...

Problem

- DataBase of
Ticket Data

- *Ex:*

1 1901,1881,199,901

2 901,1661

3 676,199,177,100

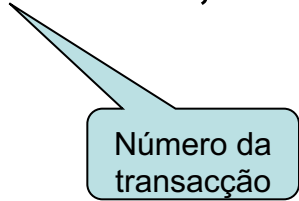
.....

...

120099 78,1881,199,8



item



Número da
transacção

- Market department of a supermarket chain aims to study consumers behavior.
- The department hold data representative of “shopping baskets “ (basket data)
- Type of question:
 - Which products are associated to the consumption of beer X.?
 - How to describe the population that consumes peanuts?
 - Where should be located the shelf for detergents?
 - How to relate products 1661 with 199?

How to represent the derived information?

- Rules to relate products (items),

Rule quality defined by statistical measures.

901 & 1661 → 67

A potential explosive number of rules can be derived!

All rules are meaningful?

How to obtain ?

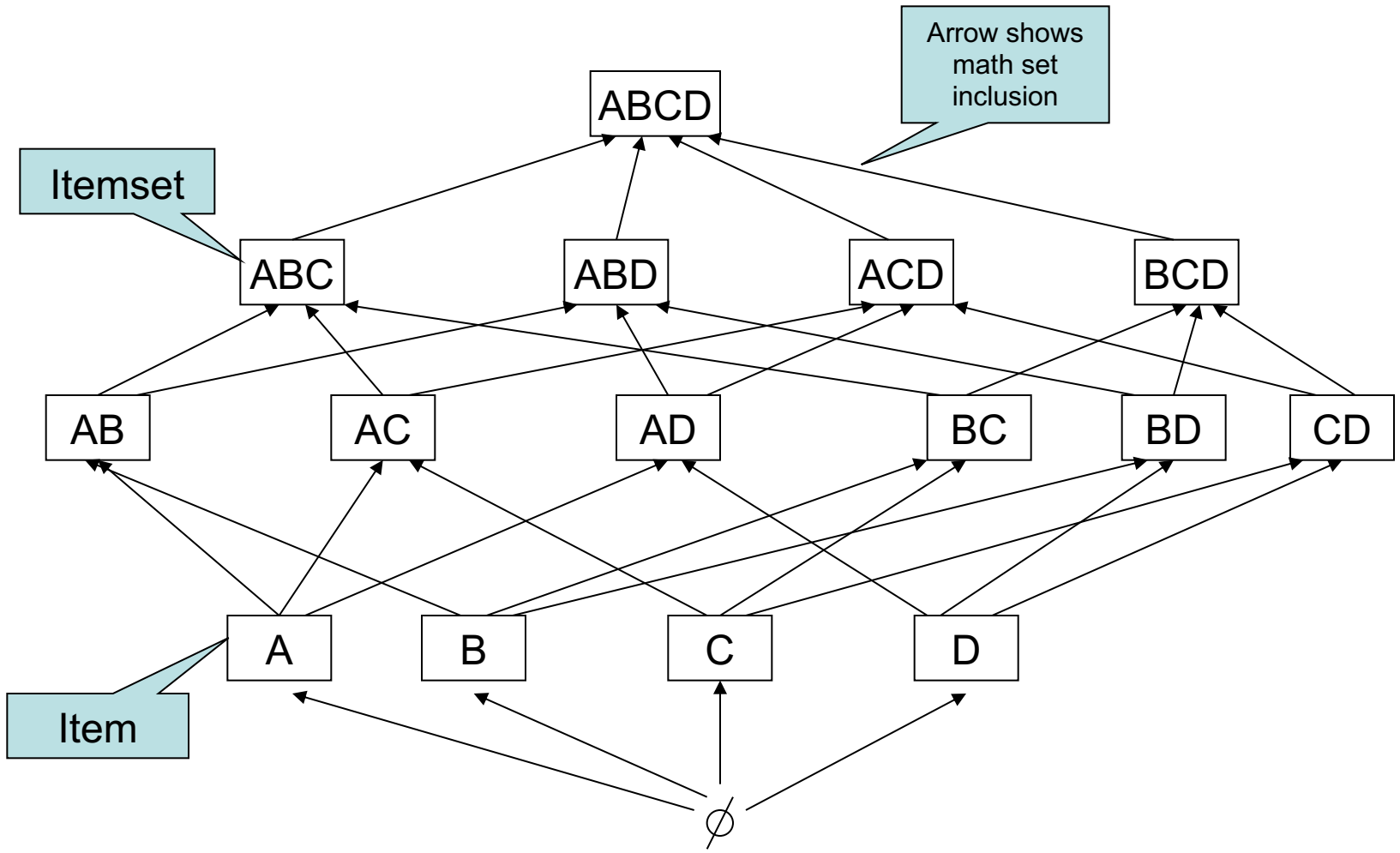
What is the most efficient procedure to apply?

How to select ?

How to discriminate "good" from "bad" rules?

How to organize ?

Watch the problem from a search space perspective



Interest Measures

- Typically an incidence measure is used to define which rules are significant (relevant).
- Support is the most popular (itemset counting).
- Rules are qualified by an interest measure (predict ability, strength, robustness of a rule).
- Confidence is normally used (conditional probability)
- Thus, an association rule:

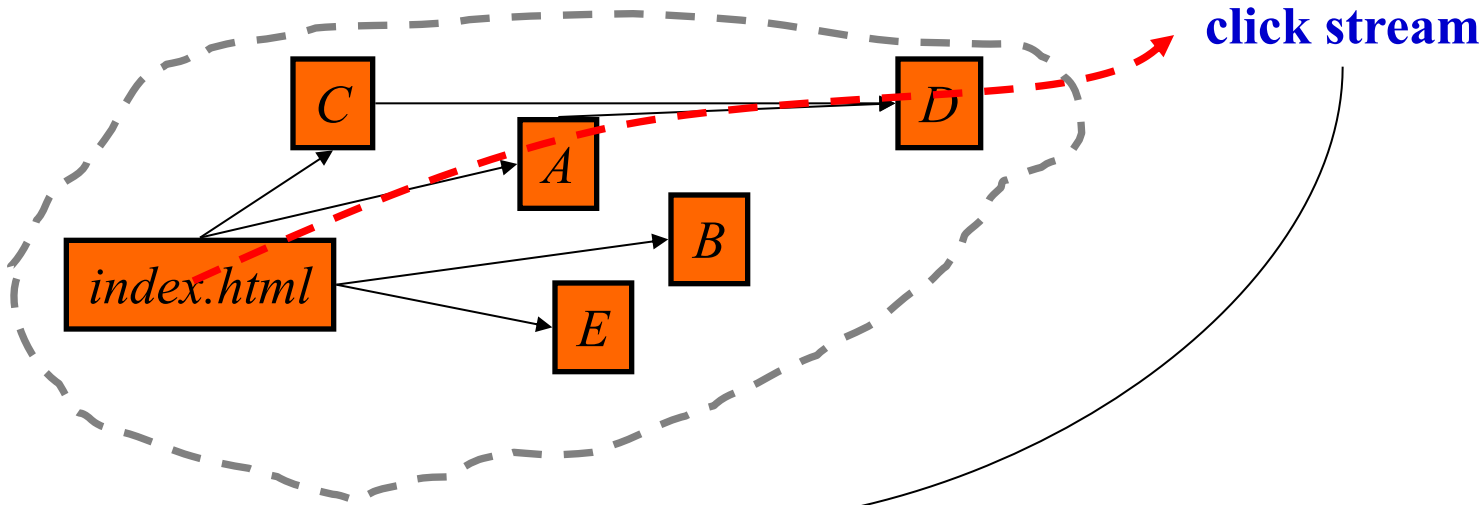
901 & 707 → 1088 (s=0.3, conf=0.9)

- Read as: *products 901, 707 e 1088 gather together in a single shop occur in 30% of the transactions. Furthermore, 90% of the transactions that contains 901 e 707 also have product 1088.*
- Another possible reading: *90% of the subpopulation defined by 901 e 707 consume product 1088.*
- Yet again: *the frequency of consume of products 901 e 707, when product 1088 is added, lowers to 90% of the initial counting.*

Applications

- Recommendation systems,
- Adaptative Web
 - Amazon: the site recommends new subjects of interest using the visited/bough user items.
 - Challenge *Netflix*: <http://www.netflixprize.com>
- Descriptive Data Mining (Protein residues Clusters),
- Spam Filtering,
- Categorical Prediction Models (classification),
- etc,

Application: Recommendations using AR



Obs.: `A` `D`

Rules:

`A` `B` `F` → `X` (conf: 0,8)

`A` `E` → `R` (conf: 0,7)

`A` `D` → `F` (conf: 0,6)

`A` → `D` (conf: 0,5)

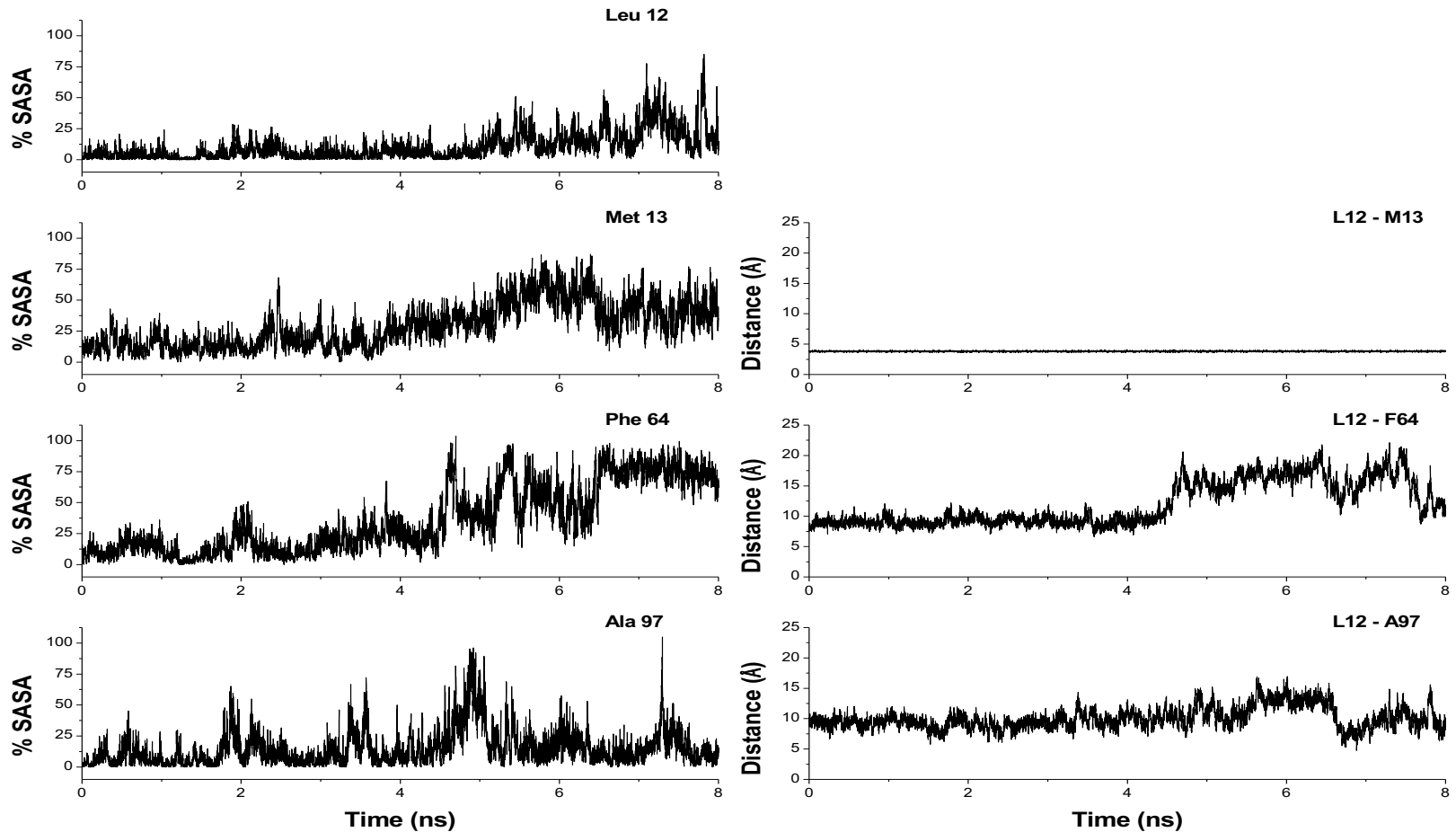
`D` → `X` (conf: 0,4)

Recommendations (top 2):

`F` (0,6)

`X` (0,4)

TTR Hydrophobic Clusters detection



Sup=0.34821 conf=1.0000

LEU_12=[00.00 : 25.00[←

PHE_64=[00.00 : 25.00[& MET_13=[00.00 : 25.00[& ALA_97=[00.00 : 25.00[

Rule Generation

- Compute confidence: $\text{conf}(A \rightarrow C) = s(A \ C) / s(A)$.
- Thresholds for conf and sup (minsup e minconf)
- Trivial Algorithm e.g:
Having ABC (check rule $AB \rightarrow C$),
 teste it, knowing $s(AB)$ and $s(ABC)$,
 if $s(ABC) / s(AB) \geq \text{minconf}$
Do this procedure for all
itemsets $\in \text{Power_set}(\{A,B,C\})$, where $\# \text{itemset} > 1$.

Compute Frequent terms (frequent itemsets)

- Algorithm naive:

Let $K = \{ \text{items in DB} \}$,

Compute $P(K)$ (Power_set),

Scan DB to count the occurrence of $P(K)$

Filter itemset in $P(K)$ that do not satisfy minsup.

- Intractable!!!!!!!!!!

- Better: Make use of the *downward closure property of support*:

$$\text{If } X \subseteq Y \text{ then } s(X) \geq s(Y)$$

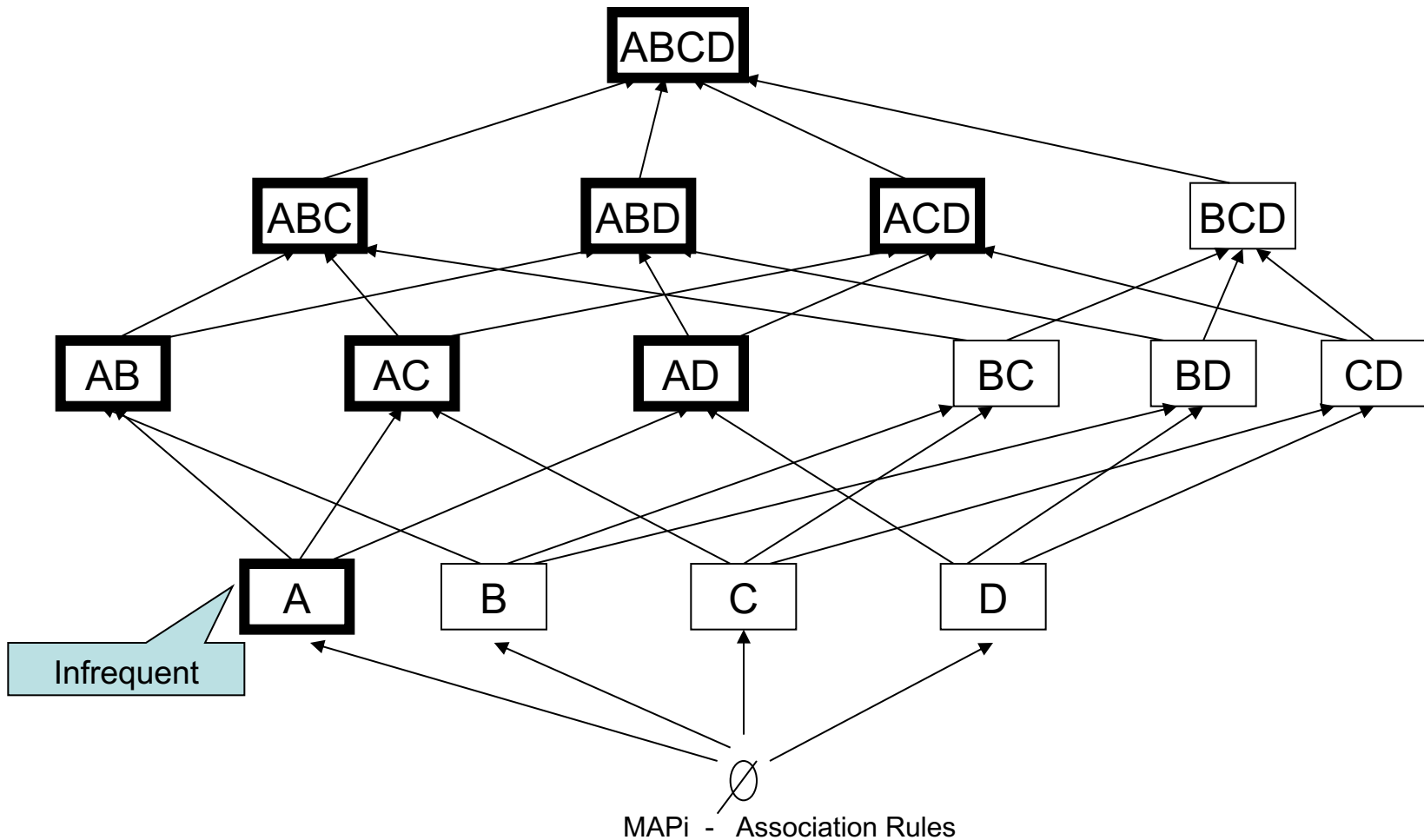
Apriori Algorithm [Agrawal & Srikant 94]

- $L_1 = \{ \text{1-items frequentes} \}$
- For($k=2; L_{k-1} \neq \{ \}$; $k++$) do
 - $C_{\{k\}} = \text{apriori_gen}(L_{\{k-1\}});$
 - forall transacções $t \in D$ do
 - $C_{\{t\}} = \text{subsets}(C_{\{k\}}, t)$
 - Forall candidatos $c \in C_{\{t\}}$ do
 - $c.\text{count}++;$
 - End
 - $L_{\{k\}} = \{ c \in C_{\{k\}} \mid c.\text{count} \geq \text{minsup} \}$
 - End

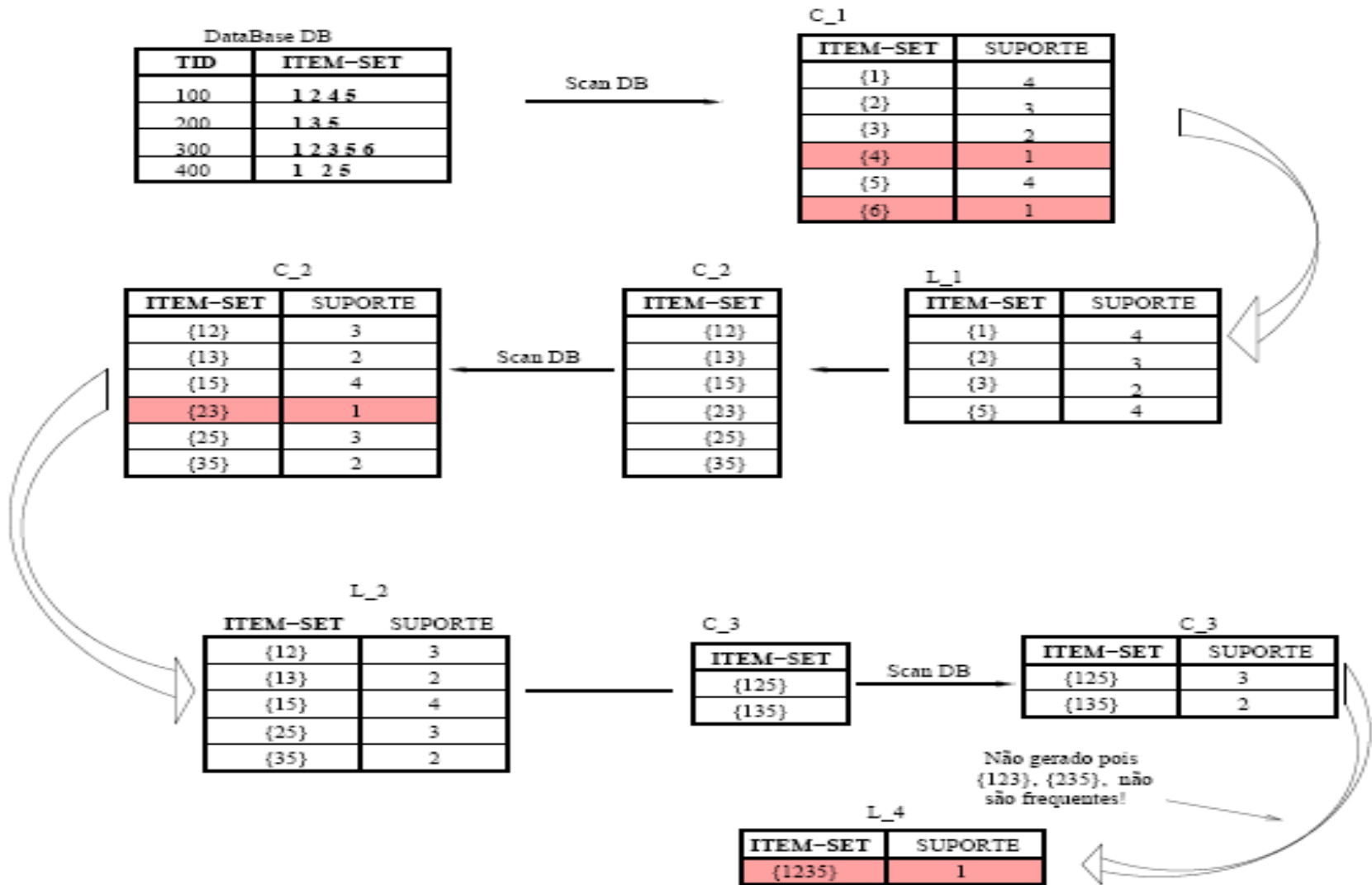
Answer= $\bigcup L_{\{k\}};$

Algorithm Bottom-up and breath-first. In **apriori_gen** the candidates to be counted are generated Only candidates that satisfy anti-monotonicity of support are considered. (An itemset is a candidate if all its subsets are frequent!!)

Using downward closure of support (anti-monotonic)



Apriori “in action...”



Algorithms for FIM

- Breath-First

Apriori

Partition

Dic

Sampling

- Depth-First

FP-growth

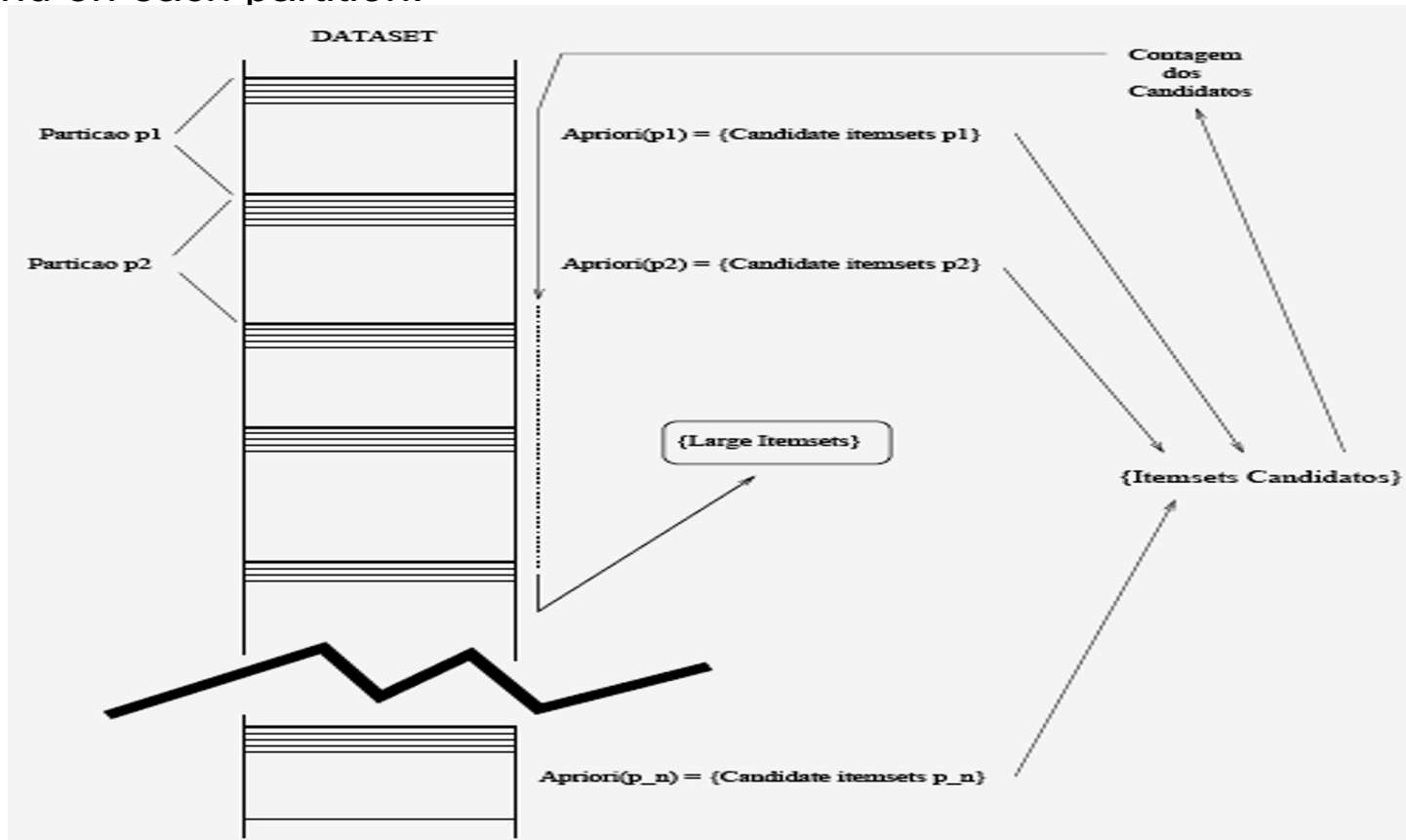
Inverted Matrix

Eclat

Depth-first expansion (CAREN)

Partition Algorithm

- Parallel version of *Apriori*. Defines several partitions in the dataset. *Apriori* algorithm applied to each different partition.
- The set of itemsets found on each partition is a superset of the frequent itemset in the dataset. A second scan is performed to count the candidates found on each partition.



Partition(2)

- Filtering on global candidates

$$\begin{aligned} S_i &= \sum(\text{local sup}_i) \\ S_{\min(i)} &= \sum(\text{local minsup}_i) \quad \text{where } i \text{ is frequent} \\ p &= \#\{\text{partitions}\} \\ p_i &= \#\{\text{partitions where } i \text{ is frequent}\} \end{aligned}$$

Before phase II of the algorithm:

- Check whether a given itemset i is frequent in all partitions.

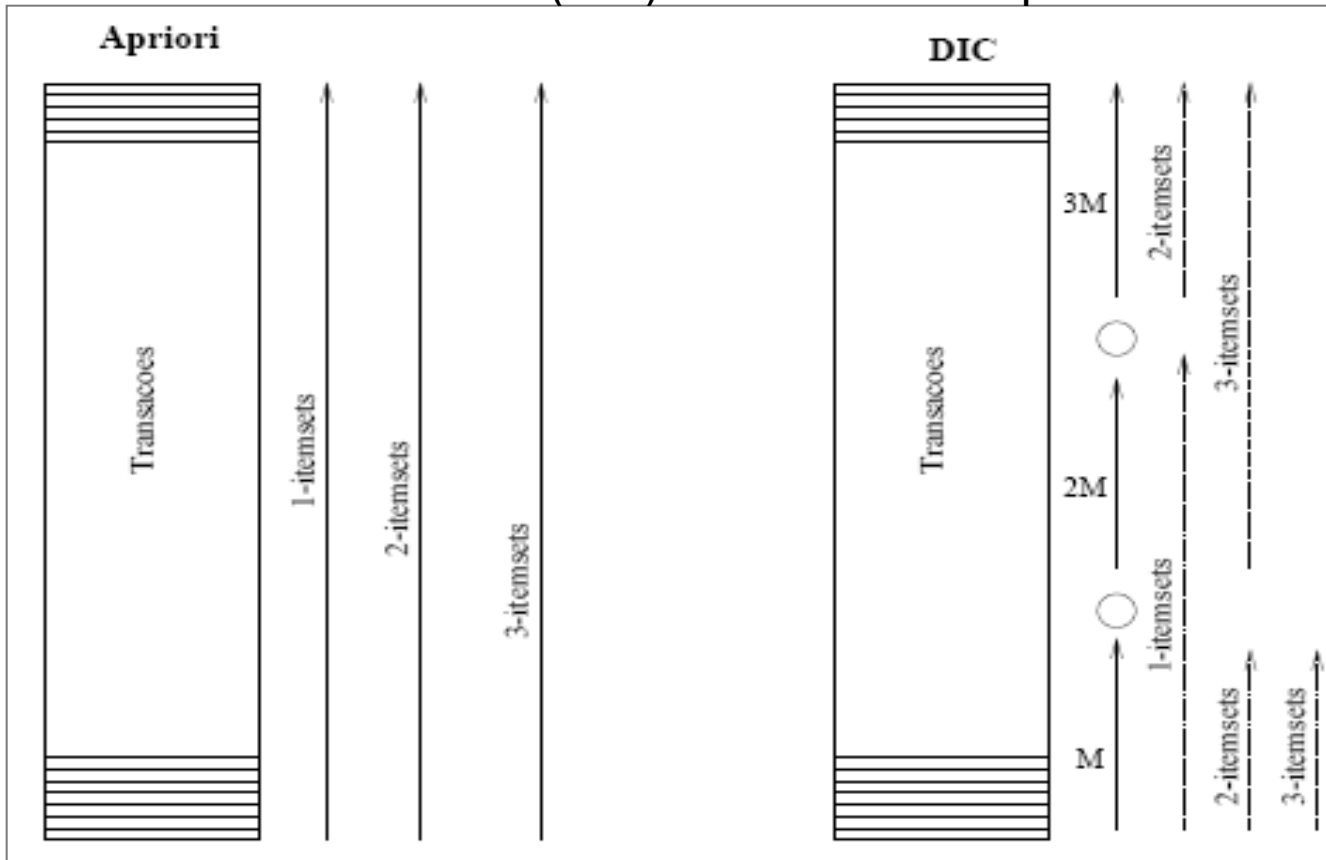
Then i can be removed from counting (it is frequent with Support = S_i)

- Check whether $S_i + p_i < S_{\min(i)} + p$

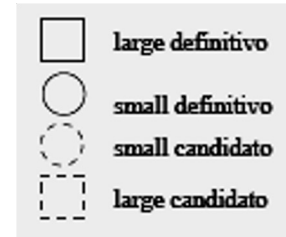
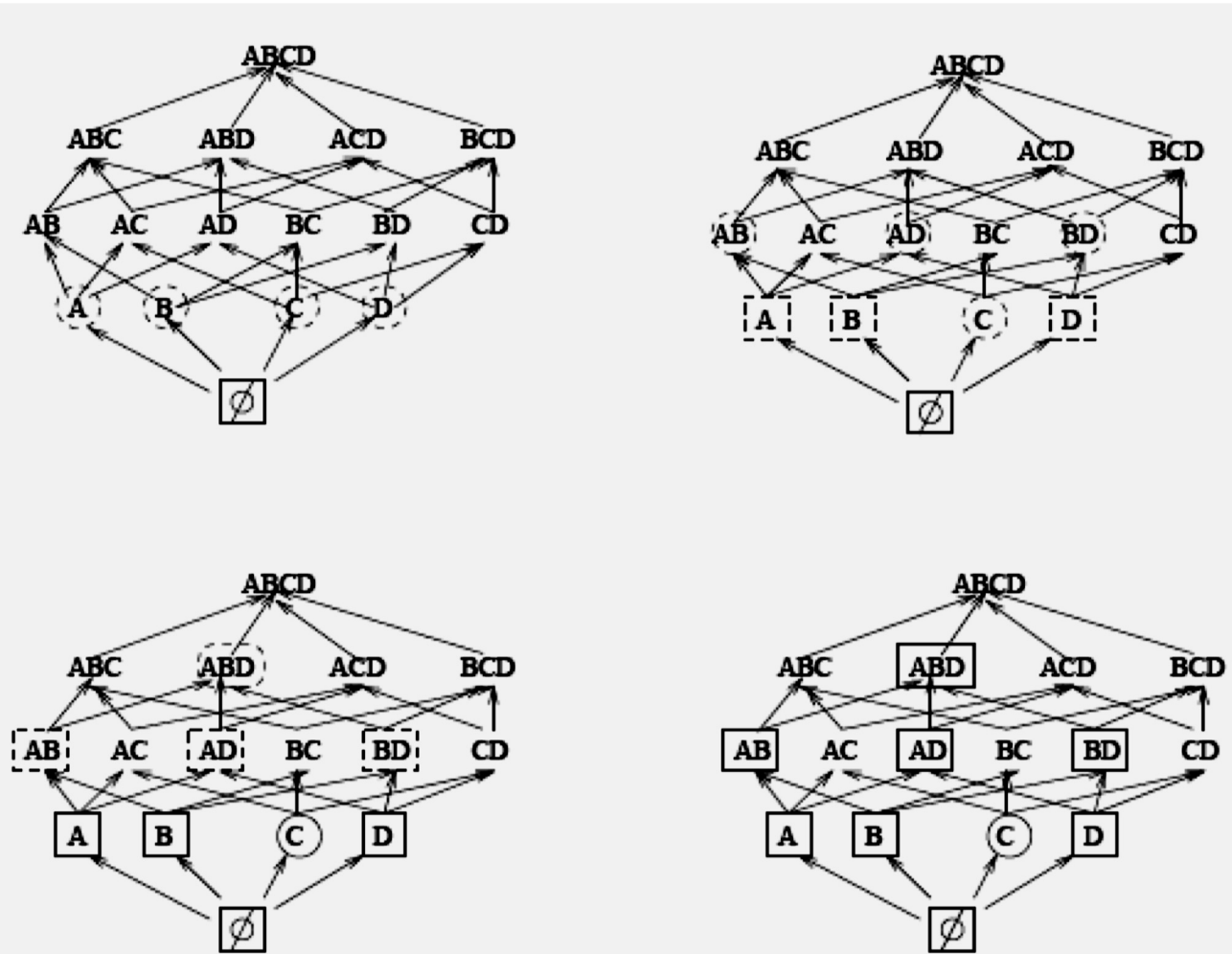
Then itemset i and all extensions can be removed from counting since it is infrequent and all its extension also infrequent.

Dynamic Itemset Counting (DIC)

Reduce the number of scans in the database using the idea of starting the counting of the N-itemset when all its subsets (N-1)-itemsets are frequent.



DIC(2)



- Large definite: frequente and counting
- Large candidate: frequente but being counted
- Small definite: already counted and not frequent
- Small candidato: being counted and still not frequent

Algorithms:

Dataset Representations

- Horizontal
 - Transactions are lists of items. Ex:
t12: 1,4,6,7,12,129,929
t15: 2,4,5,6,14,189,901
- Vertical
 - To represent the cover of an item on each transaction. Ex:
Tidlist(6) = [t12,t15,t24,t123,t300,...]
Tidlist(14) = [t15,t120,t541,...]
Tidlist(129)= [t12,t18,t45,...]

Vertical Representations

- Cover Lists

- Ideal for “sparse” data
- $Tidlist(I) = [t_4, t_9, t_{12}, t_{45}, t_{312}, \dots]$
- $s(I) = \#coverlist(I)$
- $Tidlist(A \cup B) = tidlist(A) \cap tidlist(B)$

- BitMaps

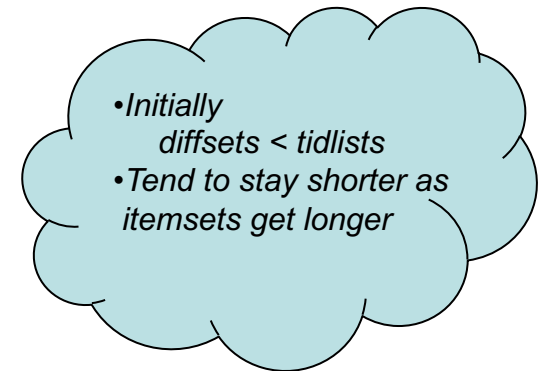
- Better suitable for “dense” data
- $bitmap(I) = \text{“}0010011100011000\text{”}$
- $s(I) = \text{bitcount}(bitmap(I))$
- $bitmap(A \cup B) = bitmap(A) \& bitmap(B)$

Bit (1s) counting

Bitwise logical and

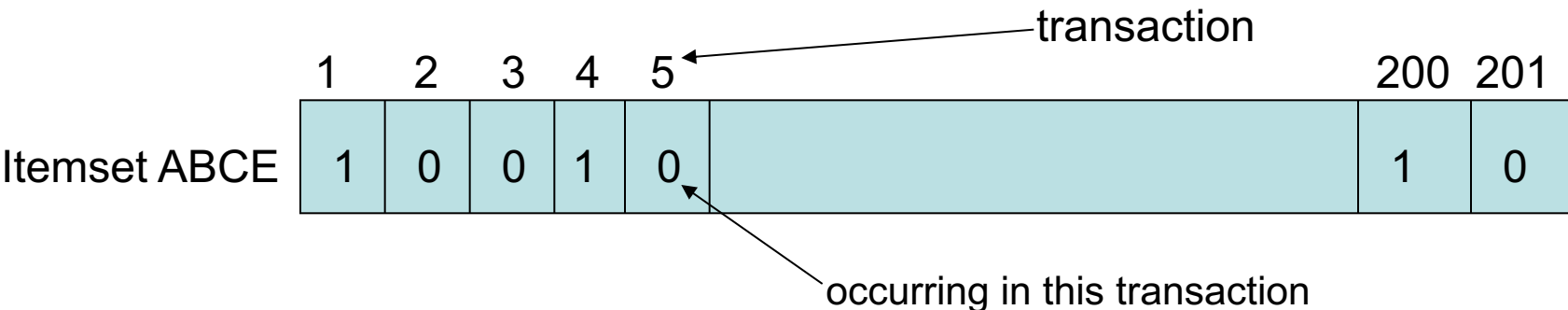
Vertical Representations (2)

- DiffSets (Highly scalable)
 - Instead of using the tidlist, only represent the “updates” to each tidlist to obtain support.
 - $\text{Diffset}(A \cup B) = \text{tidlist}(A) - \text{tidlist}(B)$ (elements of A that do not occur in B)
 - $s(AB) = s(A) - \#ds(AB)$
 - $ds(ABC) = ds(AC) - ds(AB)$
 - $s(ABC) = s(AB) - \#ds(ABC)$
- Exemplo:
 - $t(A) = [1,3,4,5]$, $t(B)=[1,2,3,4,5,6]$, $t(C)=[2,4,5,6]$.
 - $ds(AB)=[]$, $ds(AC)=[1,3]$, $ds(ABC)=[1,3]$,
 - $S(ABC)= 4 - 0 - 2 = 2$.



Depth-first Expansion (Caren)

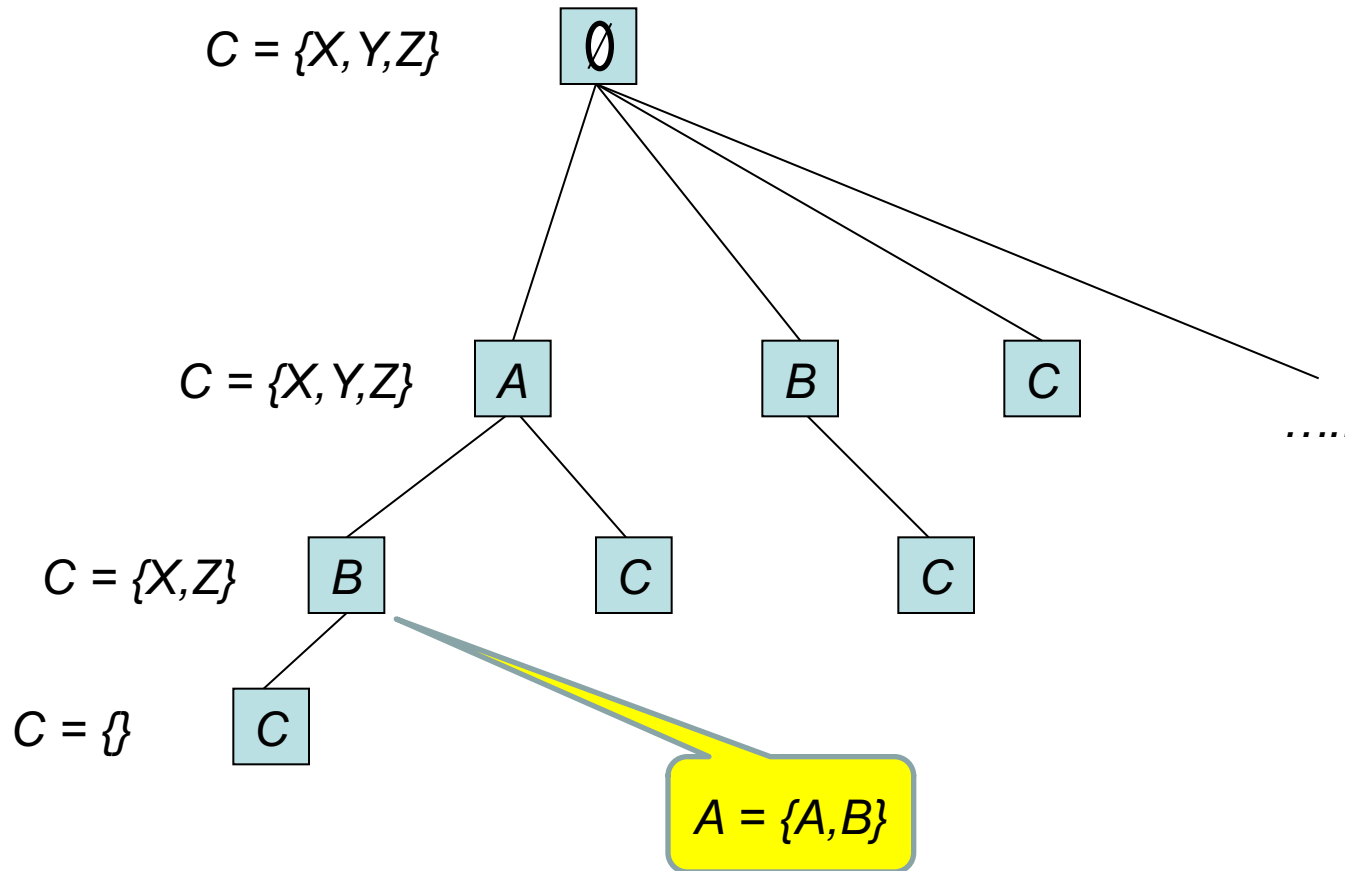
- Rules oriented algorithm instead of itemset oriented.
(The computed itemsets is immediately considered an antecedent of a rule which is also immediately computed). Therefore, after the frequent term is computed the rule(s) is (are) immediately computed.
- Single scan to the database :
 - Count frequent 1-items (items sorted by support increasing order).
 - bitmaps mounting (cover lists) of 1-items and count frequent 2-itemsets.
- Depth-first expansion: Expand itemsets by joining items (using defined items ordering) . Performed bitwise-and to obtain cover list of the new itemset. Make use of some testing to avoid redundant bitcounting operations (verify if 2-itemsets contained in the new itemsets are frequent, etc and other “tricks”).
Support counting ↔ bitcounting.



Depth-first Expansion (Caren)

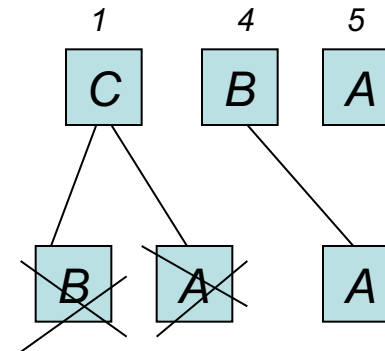
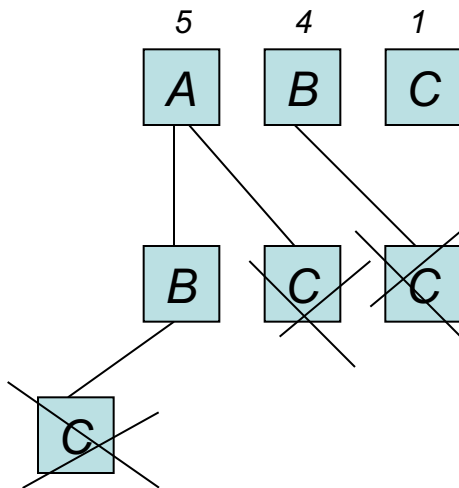
- General idea:
 - $A = \{\text{antecedent in construction}\}$,
 - $C = \{\text{list of consequences}\}$,
 - $F = \{\text{frequent items}\}$.
 - C can be a predefined list supplied by the user or dynamically to be $C = F / \{a: a \in A\}$,
 - Expand A and try all elements of C in A to derive rules that satisfy the different filters e.g. minsup, minconf, etc.
 - C is dynamically defined for each expansion branch i.e. certain elements are removed since they cannot be part of a rule (see rule-based algorithms).

Depth-first Expansion (Caren)



Pruning Mechanisms implemented in Caren (2.5)

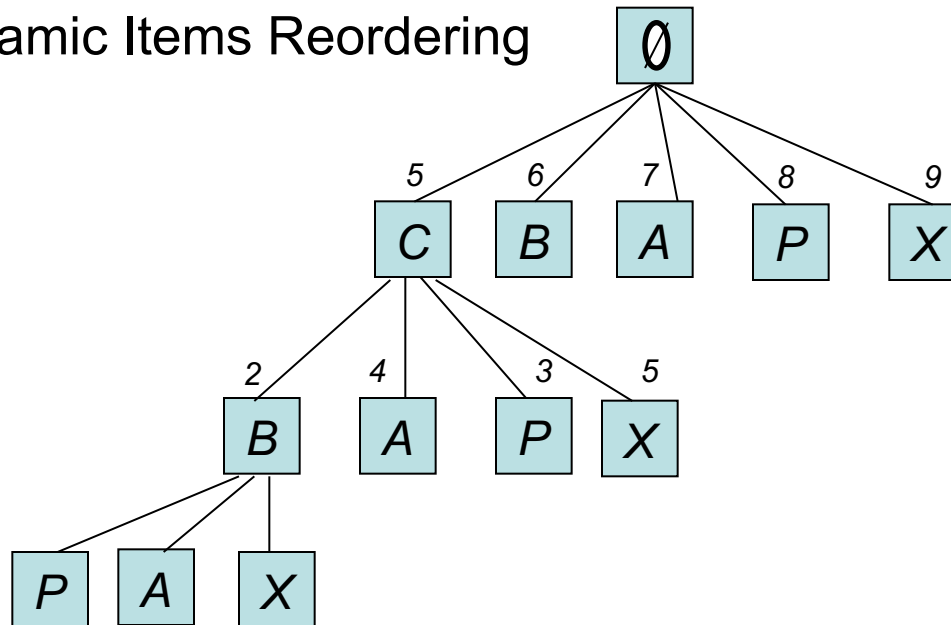
- Sorting the frequent terms using support ascendant order tends to yields smaller search spaces.
- Implements the idea of “earlier failure”,
- Depth-first expansion tends to form first itemsets with smaller support (with less chances of satisfying the minimal support filter).



Pruning mechanisms in Caren (2.5)

- Dynamically apply this reordering to each level of the depth-first expansion

- Dynamic Items Reordering



- Note! Order on level 2 changed: $s(CB) < s(CP) < s(CA) < s(CX)$
 - This eventually leads to find items that yield non frequent itemsets: Thus we do not propagate down these items in the depth-first expansion.

Inference rules for Counting

- Some algorithms make use of inference rules (and thus avoiding some computational effort) by deriving itemsets counting using the frequency of its subsets.
- Parent Equivalence Pruning (PEP) :

- (support inference)

Let X, Y, Z itemsets,

$$s(XY) = s(X) \Rightarrow s(XYZ) = s(XZ)$$

Will be very useful to detect productive and significant rules! **Parent Equivalence Pruning**

Pruning Mechanisms - Caren (2.5)

- Parent Equivalence Pruning (PEP):
 - Applies support inference,
 - Let P be an itemset and X an item. if $s(P) = s(PX)$ then for any $A \supseteq P$, $s(AX) = s(A)$,
 - We can remove X from the list of items used for expansion. This yields a drastic search space reduction!
 - At the end of the counting process, one makes use of all the items of type X , and apply this expansion to all frequent itemsets where the support equals the support of the original itemset,.
 - For each itemset one gets $2^{\#\text{items } X} - 1$ new frequent itemsets.
 - Example: $s(AB) = s(ABC)$, $s(AB) = s(ABD)$, $s(AB) = s(ABE)$, the list $pep = \{C, D, E\}$. If we get ABR we will derive $ABRC$, $ABRD$, $ABRE$, $ABRCDE$, $ABRCDE$, $ABRCDE$, $ABRCDE$ with support equals to $s(ABR)$.
 - This mechanism will be of utter importance for an efficient implementation of improvement (and other pruning methods).

Caren (2.5) Pruning mechanisms

- Controlling term expansion along the rule generation:
 - Let P be an itemset, $CONS$ a set of items predefined to be rule consequents,
 - If for $\forall c \in CONS, s(P \cup c) < \text{minsup}$
then $\forall X, s(P \cup X \cup c) < \text{minsup}$.
 - That is, if a term (P) for all defined consequents ($\forall c \in CONS$) do not imply any rule with minimal support then it is not worth to keep expanding this term!

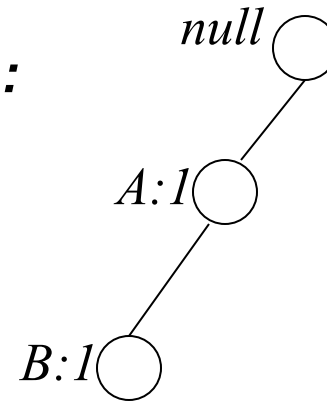
FP-Growth

- One of the most popular algorithms for frequent itemset mining.
- Makes use of an efficient dataset/database representation based on a tree-like structure - **FP-tree**.
- Two DB scans: 1st to count frequent items, 2nd to build the FP-tree.
- Once the FP-tree is built, the algorithm uses a *divide-and-conquer* recursive approximation to obtain the frequent itemsets.

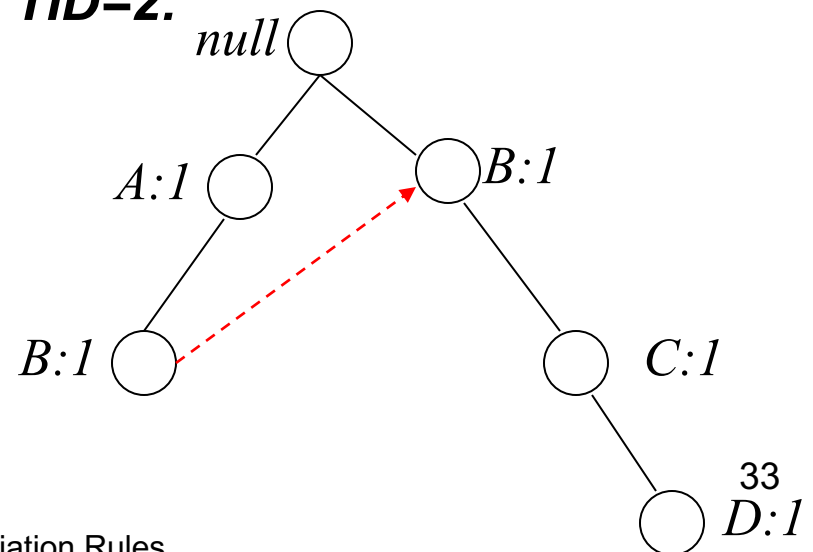
Building the FP-Tree structure

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{B,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

After reading TID=1:



After reading TID=2:

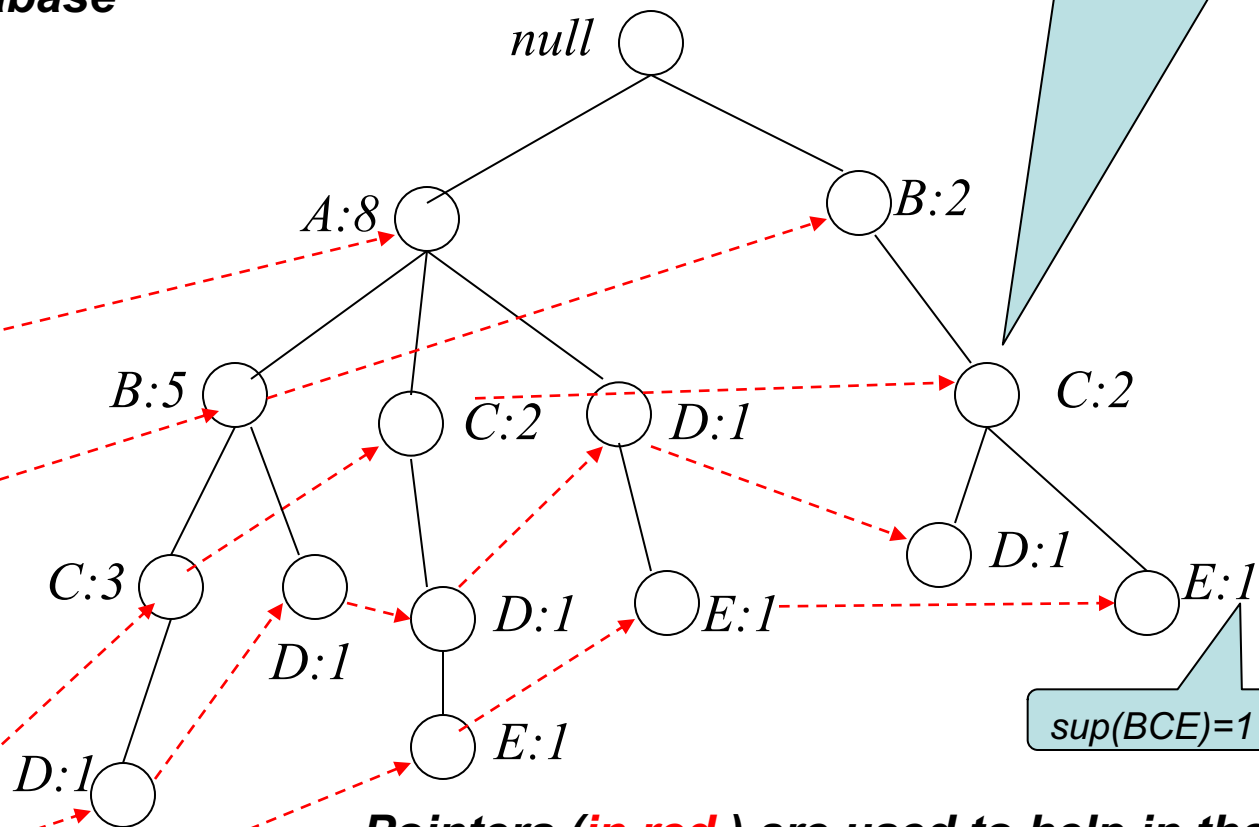


Building the FP-Tree (2)

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

Transaction Database

Items sorted by support descendent order. This yields a higher probability for node sharing between branches in the FP-tree



Header table

Item	Pointer
A	→
B	→
C	→
D	→
E	→

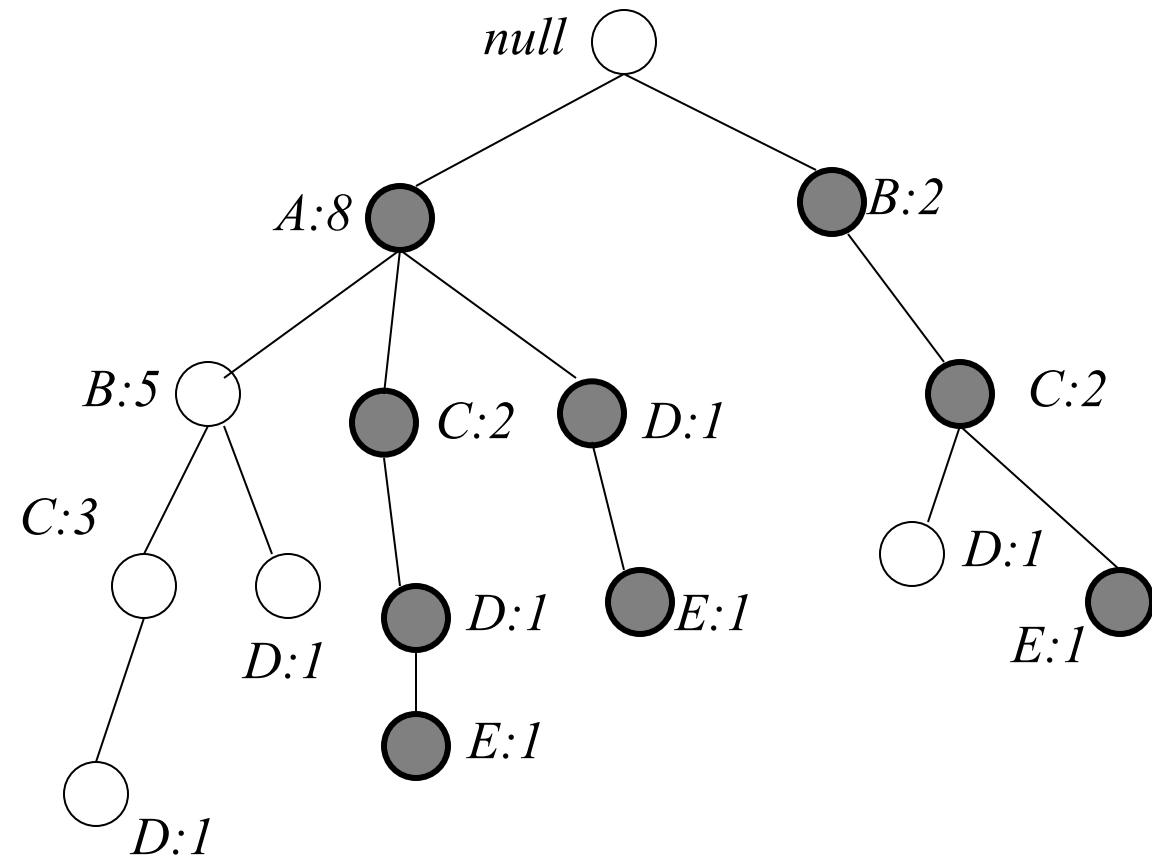
$sup(BCE)=1$

Pointers (in red) are used to help in the frequent terms generation.

FP-growth: Frequent Itemsets

- All itemsets can be derived from the FP-tree,
- These are obtained by following the node links that start at the header table.
- Start by the least frequent item.
- In a glance, the algorithm is:
 1. For each item A
 2. Get all itemsets that contain A (conditional pattern base)
 3. Get a FP-tree for these itemsets (conditional FP-tree)
 4. Get 2-itemsets from the conditional FP-tree.
 5. For each 2-itemset repeat step 2.

FP-growth – Deriving frequent itemsets



Build a conditional pattern base for E:

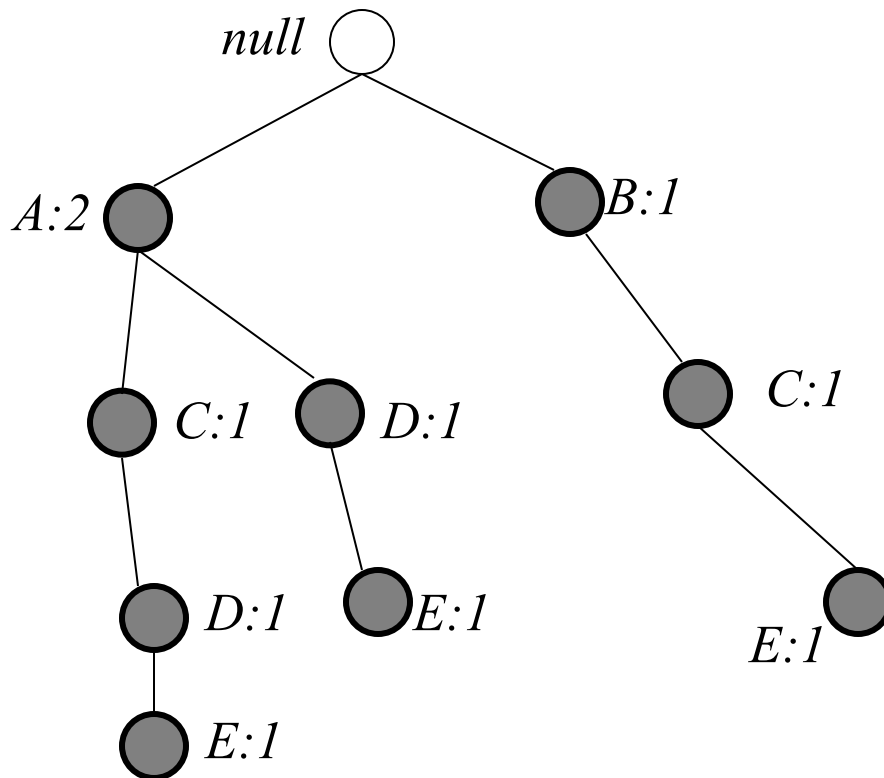
$P = \{(A:1, C:1, D:1),$
 $(A:1, D:1),$
 $(B:1, C:1)\}$

Apply FP-growth recursively in P

NOTE: minsup(abs) = 2

FP-growth

Conditional tree for E:



Conditional Pattern base for E:

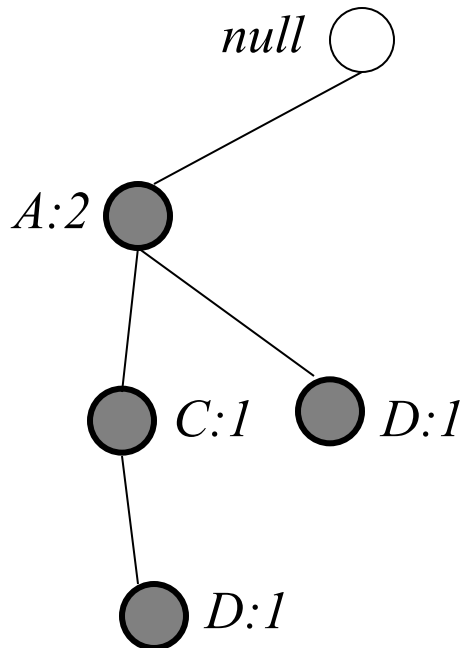
$P = \{(A:1, C:1, D:1, E:1),$
 $(A:1, D:1, E:1),$
 $(B:1, C:1, E:1)\}$

de E =3: {E} is frequent

**Apply recursively
FP-growth in P**

FP-growth

**Conditional tree for D
inside the conditional
tree for E:**



**Conditional pattern base
of D inside the
conditional base of E:**

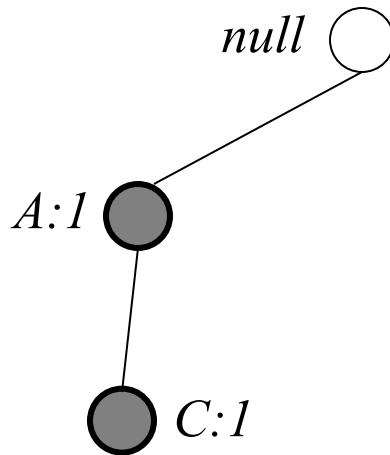
$$P = \{(A:1, C:1, D:1), \\ (A:1, D:1)\}$$

**#D =2: {D,E} is a frequent
itemset**

**Apply recursively FP-
growth em P**

FP-growth

**Conditional tree for C
inside D, having this
one inside E:**



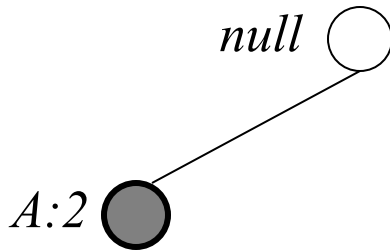
**Conditional pattern base
for C inside D inside E:**

$$P = \{(A:1, C:1)\}$$

**#C = 1: {C,D,E} is not
frequent!**

FP-growth

**Conditional tree for A
inside D inside E:**



#A = 2: {A,D,E} is frequent

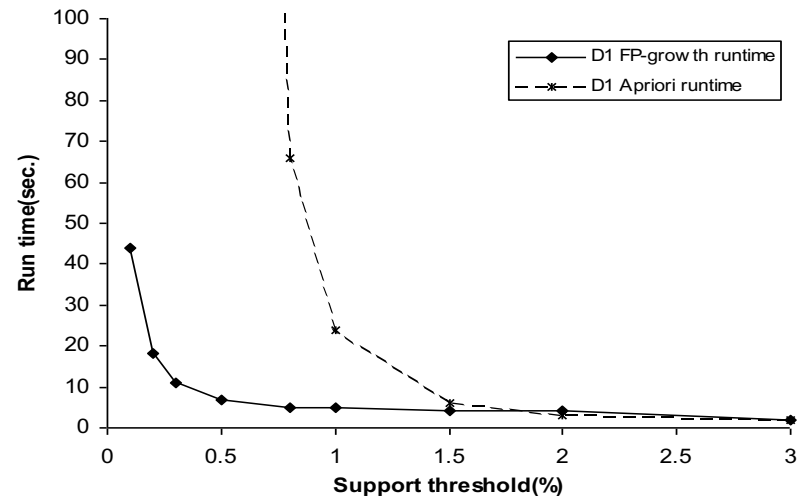
Next step:

**Build conditional tree for C
inside the conditional tree of E**

**Proceed until exploring all the
conditional tree for A (which
only has node A)**

Benefits of the FP-tree

- Performance shows that:
 - FP-growth is an order of magnitude faster than Apriori.
- Features:
 - No candidate generation, no candidate test
 - Uses a compact data structure
 - Removes the need for several consecutive database scans
 - Basic operations are counting and FP-tree built.



Rule Examples

Association Rules ...

Sup = 0.01500	Conf = 0.37500	oranges	←	bananas & peaches
Sup = 0.03900	Conf = 0.30000	oranges	←	peaches
Sup = 0.01000	Conf = 0.28571	oranges	←	bananas & potatoes
Sup = 0.01000	Conf = 0.28571	oranges	←	peaches & potatoes

- What kind of information one can extract from these patterns ?
- how one should read these rules...
- Predict ability?
- Metric (measures) reading
- How the described population is characterized...
- Redundancy

Interest Measures

- Lift
- Conviction
- Leverage
- χ^2
- Reliability
- etc

$$Lift(A \rightarrow C) = \frac{conf(A \rightarrow C)}{s(C)}$$

$$conv(A \rightarrow C) = \frac{1 - s(C)}{1 - conf(A \rightarrow C)}$$

$$leve(A \rightarrow C) = s(A \cup C) - s(A) * s(C)$$

$$R(A \rightarrow C) = |conf(A \rightarrow C) - s(C)|$$

χ^2 test between antecedent and consequent

$$importance(A \rightarrow C) = \log \left(\frac{conf(A \rightarrow C)}{conf(\neg A \rightarrow C)} \right)$$

SQL Server 2000: used measure =
0 → independency
< 0 → positive association,
> 0 → negative association.

Interest Measures (2)

Confidence:

- measures conditional probability
P(C) given A
- Tends to shed light on uncorrelated rules
(spurious rules).

$$\text{conf}(A \rightarrow C) = \frac{s(A \cup C)}{s(A)}$$

Laplace:

- confidence estimator that considers support
- becomes more pessimistic along smaller values of $s(A)$
- similar problems arise as with confidence.

$$\text{lapl}(A \rightarrow C) = \frac{s(A \cup C) + 1}{s(A) + 2}$$

Lift:

- Measures distance to independence between A e C
- varies in $[0, +\infty[$
- 1 \rightarrow independence,
- Values far from 1 \rightarrow show evidence that
A supplies information about C.
- measures co-occurrence (not implication)
- simetric measure!

$$\text{Lift}(A \rightarrow C) = \frac{\text{conf}(A \rightarrow C)}{s(C)}$$

Interest Measures (3)

Conviction:

- aims to repair weakness in conf and lift
- varies in $[0.5, +\infty[$
- tries to capture degree of implication between A and C
- its directional i.e. $\text{conv}(A \rightarrow C) \neq \text{conv}(C \rightarrow A)$
- value 1 represents independence
- motivation: (logical implication): $A \rightarrow C \Leftrightarrow \neg A \cup C \Leftrightarrow \neg(A \cap \neg C)$
- measures when $(A \cap \neg C)$ deviates from independence.
- inverts the ratio between $s(A \cup \neg C)$ and $s(A) \times s(\neg C)$ to cope with negation
- Excellent measure for classification.
- ratio between expected frequency of rule prediction error (A occurs without C), assuming independence between A e C, and the observed frequency of incorrect prediction e.g. **conv=1.5** means that a rule would be wrong 1.5 of the times as often (50% more) if the association between A e C was purely random chance.

$$\text{conv}(A \rightarrow C) = \frac{1 - s(C)}{1 - \text{conf}(A \rightarrow C)}$$

Leverage:

- varies in $] -0.25, 0.25[$
- measures the number of extra cases obtained
In relation to the expected support (independence)

$$\text{leve}(A \rightarrow C) = s(A \cup C) - s(A) \times s(C)$$

χ^2 test:

- Measures statistical independence between antecedent
And consequent
- does not capture correlation strength between A and C
- Only supports the decision on independence.

$$\chi^2 = \sum_{r_i \in R} \frac{(O(r) - E[r])^2}{E[r]}$$

Interest Measures (4)

Jaccard:

- measures the “overlap” degree between cases covering A and cases of C
- varies between [0,1]
- measures the distance between A and C using the fraction between the cases covered by both and cases covered by just one (A or C).
- high values denote overlapping.

$$jacc(A \rightarrow C) = \frac{sup(A \cup C)}{sup(A) + sup(C) - sup(A \cup C)}$$

Cosine:

- also an overlapping measure between A and C
- regard A and C as two vectors
- 1, vectors coincide
- 0, vectors do not overlap (also varies in [0,1])

$$cos(A \rightarrow C) = \frac{sup(A \cup C)}{\sqrt{sup(A) \times sup(C)}}$$

$$MI(A \rightarrow C) = \frac{\sum_i \sum_j sup(A_i \cup C_j) \times \log\left(\frac{sup(A_i \cup C_j)}{sup(A_i) \times sup(C_j)}\right)}{\min(\sum_i -sup(A_i) \times \log(sup(A_i)), \sum_j -sup(C_j) \times \log(sup(C_j)))}$$

Mutual Info:

- measures a uncertainty reduction in the consequent when one “knows about” the antecedent.
- is symmetric
- based on Shanahan’s entropy!

Flaws in the *Confidence* measure

Confidence may be unable to detect independence.

Rule *eggs* → *milk* may have conf=80% but we could know that the consumption of eggs is independent from *milk*.

Independence between A and C:

$$s(A \cup C) = s(A) \times s(C)$$

Also negative/positive dependency.

One can use X^2 to measure correlation between antecedent and consequent.

$$X^2 = \sum_{r_i \in R} \frac{(O(r) - E[r])^2}{E[r]}$$

Example: Apply X^2 using conf=95% and 1 degree of freedom, If $X^2 \geq 3.84$ then reject the independence hypothesis, (check table for $\alpha=0.05$ and 1 degree. Value is 3.84)

Weakness of the support-confidence framework

- Minimal support is context dependent and sometime hard to define.
- Certain problems demand very low minimal support thresholds e.g. caviar → champagne
- Possible solution: find the *k-optimal rules* (being optimal in a specific interest measure sense)
- High minimal support and confidence can refrain to derive some interesting rules;
- Confidence can assign high interest to non correlated rules (as seen before!)
- Alternative measures lead to similar problems.

False Discoveries

- Let ρ be a rule that satisfies a set of constraints φ in relation to a distribution Θ , found in a dataset D .
- There is a risk of finding ρ that satisfies φ in relation to D but not in relation to Θ . (all pruning method studied before have this problem!)
- Statistical Hypothesis tests :
 - H_0 : $\neg\rho$ is true
 - If ρ is found then have a *false discovery* - Type I error (**unsoundness**)
 - For all ρ that does not satisfy H_0 and is not derived - Type II error (**incompleteness**)

Rule Selection and Pruning

- A FIM algorithm (even with minimal support filtering) can derive millions of rules.

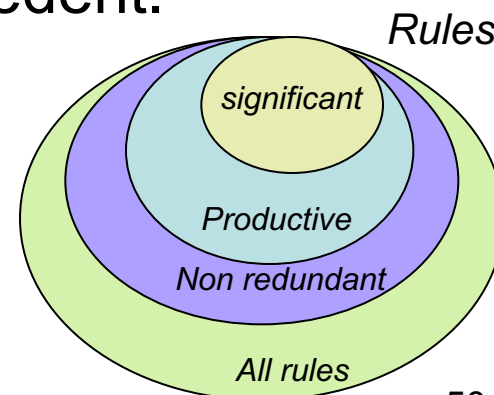
We can even have $\#\{\text{rules}\} \gg \#\{\text{transactions}\} !!!$

- Majority of rules are derived by chance (in a statistical sense). Notion of *false discoveries*
- Non correlated rules (where the antecedent and the consequent are independents)
- Arise of *Redundant Rules* (Zaki00).

Rules contain items in the antecedent that are justified by other items also present in the antecedent.

Ex (pregnant \rightarrow woman):

- pregnant & woman \rightarrow swollen_feet
- Discard *redundant rule* $x \rightarrow y$ if:
 - $\exists z \in x : s(x \rightarrow y) = s(x - z \rightarrow y)$



Rules Pruning

- Identifying *improvement* in rules

Conf = 0.300 oranges ← bananas & peaches
Conf = 0.315 oranges ← peaches

- Definition of improvement:

- A more specific rule has to induce an added value in terms of its interest measure.

$$\text{imp}(A \rightarrow C) = \min(\forall A' \subset A : \text{met}(A \rightarrow C) - \text{met}(A' \rightarrow C))$$

met can be = {conf, lift, conv, X², etc}

- If improvement > 0 we say that the rules are *productive*.

Statistical Significance

- An alternative to define a minimal improvement is to apply a statistical significant test : discard rules that are *non significant* (Webb, Magnum Opus)
- A rule $x \rightarrow y$ is *non significant* if
 - There exists another rule $x - z \rightarrow y$ where the value $\text{met}(x \rightarrow y) - \text{met}(x - z \rightarrow y)$ is not significantly high (where $\text{met}(x \rightarrow y) > \text{met}(x-z \rightarrow y)$)
- A frequentist hypothesis test is used (e.g. Binomial distribution test) to evaluate significance. But Binomial computes probabilities with replacement!

Significant Rules

(First version)

- Example for $\alpha=1\%$
- Process: discard the most specific rule if the probability of obtaining, by chance, the set of transactions that cover this rules is bigger than α , assuming that the data is obtained from a random sample and that the true measure value (null hypothesis) is the one of the most general rule.
- Rules (ntrans=1000, $H_0=0.907$):

A & B \rightarrow C (antsup=272,s=255,conf=0.938)
 A \rightarrow C (antsup=333,s=302,conf=0.907)

Note: The Binomial test assumes that expectation is error-free i.e. H_0 is the same for all tests.

sucess

#cases

H.null

H.alternative

1 - α

Computes probabilities with replacement!

- `Binom.test(255,272,0.907,>H0,conf=0.99)`, gets p-value=0.04608
- First rule is non significant (for $\alpha=1\%$) in relation to the second, since adding B to the antecedent does not increases significantly the confidence (conf) value.

Significant Patterns (Webb'07)

- Two approaches :
 - Within-search
 - Using holdout data
- Holdout approach:
 - Divide data into *exploratory* data and *holdout* data.
 - Exploratory data used to derive rules
 - holdout data used for testing.
- Alternative (within search):
 - Compute search space size
 - Apply statistical tests with *Bonferroni* adjustment.
- In both: Apply hypothesis tests to each rule (obtain a *p-value* for each one).
 - Reject H_0 if $p\text{-value} \leq \alpha$.

Fisher Test for Significant Rules

Sensitive to the size of both tested population!
(assumes no replacement)

	y	$\neg y$	
X	a	b	$a+b$
$X-Z$	c	d	$c+d$
	$a+c$	$b+d$	

$$\begin{aligned}
 a &= s(x \cup y) \\
 b &= s(x \cup \neg y) \\
 c &= s(x-z \cup \neg z \cup y) \\
 d &= s(x-z \cup \neg z \cup \neg y)
 \end{aligned}$$

Uses *Fisher exact Test*, p -value($x \rightarrow y, x-z \rightarrow y$):

$$p = \sum_{i=0}^{\min(b,c)} \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{(a+b+c+d)!(a+i)!(b-i)!(c-i)!(d+i)!}$$

p (p -value) is the probability of finding the values (or more extreme values) observed in the contingency table i.e. along the diagonal a, d .

Fisher Test for Significant Rules (2)

$$H_0: p(y|x) \leq p(y|x-z)$$

- *Fisher exact Test*,

- p-value($x \rightarrow y, x-z \rightarrow y$):

*Computes the probability of the observed values obtained from the occurrence of **x & y** (or larger values) given the number of occurrences of **x-z & y** if $P(y|x) == P(y|x-z)$. Sampling without replacement is assumed!*

- Accept $x \rightarrow y$ if all p-value $\leq \alpha$

- *Webb* only applies this test between each rule $x \rightarrow y$ and its direct generalizations. That is rules:

- $\{\} \rightarrow y$ and

- $x-z \rightarrow y$ such that $|x-z| = n - 1$, being $|x| = n$.

- Note that $x \rightarrow y$ has $2^{|x|} - 1$ generalizations!!

Significant Patterns (3)

(Multiple Hypothesis)

- Multiple hypothesis testing.
Risk for type I is not more than α .
- Probability for occurring type I error rises with the number of tests. For n tests $\alpha_{real} = 1 - (1 - \alpha)^n$
- Use *Bonferroni* adjustment (correct α for n tests as $\kappa = \alpha/n$)
– To thin filter!
- *Holm* adjustment (k em vez de α).
 - Requires sorting the p-values in ascendent order and to have available all these values before computing de adjustment value (k).
 - For n tests,

$$k = \max(p_i : \forall_{1 \leq j \leq i} p_j \leq \frac{\alpha}{n - j + 1})$$

Significant Patterns (4)

(Caren implementation)

- Use *Bonferroni* adjustment (correct α for n tests as being $\kappa = \alpha/n$).
- Use layered critical values,
- Instead of a global cutoff that corrects the initial α , calculate several α'_L for each layer L .

$$\alpha'_L = \frac{\alpha}{(L_{\max} \times S_L)}$$

where S_L is the number of possible rules that one can derive from the given dataset with L items in the antecedent. L_{\max} is the max number of items permit in a rule's antecedent.

We can assure that:

$$\alpha \geq \sum_{L=1}^{L_{\max}} \alpha'_L \times S_L$$

Rule based versus Itemset based algorithms

- Itemset based:
 - Derive itemsets considering the supplied set of constraints e.g. minsup, items in the antecedent, etc
 - Generate rules from these itemsets considering the defined constraints in the consequent
- Rule based:
 - Consider a list of consequents
 - Derive the itemsets representing the antecedents that satisfy the constraints;
 - Use *new opportunities of pruning* (let $A \rightarrow c$, the rule to be derived):
 - In improvement, if $1 - \text{conf}(A \rightarrow c) < \text{minimp}$, withdraw c from the list of consequents;
 - In fisher, if $\text{Fisher}(\text{sup}(A \rightarrow c), \text{conf}(A \rightarrow c), \text{sup}(A \rightarrow c), 1) > \alpha$, withdraw c from the list of consequents;
 - If $\text{sup}(A \rightarrow c) < \text{minsup}$, withdraw c from the list of consequents;
 - If the list of consequents is empty then stop expansion of itemset A .

*Major optimization!
Enables the control of the
term expansion (itemset)
through the rule
generation process.*

Non Categorical Data

(Processing during itemsets generation)

- In an attribute/value format, numerical attribute (or any other non categorical quantity e.g. hierarchies) can lead to the derivation of a very large number of items.
- It also leads to a large number of rules. These rules tend to be very specific (most of the time without any interest value). Thus, instead of:

class=1 \leftarrow colesterol = high & age=29

class=1 \leftarrow colesterol = high & age=32

class=1 \leftarrow colesterol = high & age=41

one should have:

class=1 \leftarrow colesterol = high & age \in [29,41]

- *Catch 22 situation*: items representing narrow intervals imply lower support items which leads to not deriving low incidence rules.
- On the other hand, large intervals imply low confidence rules. Joining values of an attribute into a single interval arises loss of information.!

Dealing with numerical values

- Pre-processing:
 - Interval values. E.g. define intervals where class value is preserved.
 - Binarization; each attribute is converted into two values. A cut value is selected.
- During processing (Decision trees) :
 - Binarization: Select a cut value among the set of values within the sub-tree. The chosen one is the value that maximizes gain! (it is always one that is in between class values transition).
 - Recursively apply this principle.

Numerical Data

Age	Coolest	Blood	Class
23	High	2.33	A
24	Low	2.39	B
27	High	2.21	A
27	Low	1.09	A
29	Low	2.02	A
30	Low	2.98	C
31	Low	3.01	C
31	High	1.98	B
33	low	2.09	B

Supervised discretization: Special attribute drives the process.

Ex: **Age**: [23-23],[24-24],[27-29],[30-31],[33-33]

On **Age** < 29, **Age** ≥ 29.

Non supervised: Process is independent from any other attributes.

Ex: **Age**: [23-27],[29-31],[33-33]

Discretização

- Supervised:
 - Fayyad & Irani: Entropy oriented
 - Class intervals (caren)
 - Chi-Merge
- Non supervised:
 - Equi-depth
 - Equi-width
 - Srikant (caren)
 - K-means

Fayyad & Irani

(in Pre-processing)

- Sort values of the attribute to be discretized,
- Define the cut points – values where class values change,
- For each point, compute Information Gain:

$$E(A, T; S) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2) . \quad Gain(A, T; S) = Ent(S) - E(A, T; S),$$

- Choose point with the biggest Gain value,
- Check if MDL condition is satisfied:
 - In the positive case Stop,
 - Otherwise recursively apply the process on the left and right to the chosen point.

Fayyad & Irani

Following this idea, one wants to minimize the size of the “theory” plus the quantity of information required to specify its exceptions. This is related to Kolmogorov Complexity. In our case, the “theory” is the splitting point.

- Conditional Minimum Description Length:

Required correction due to the need to transmit which classes correspond to the upper and lower intervals.

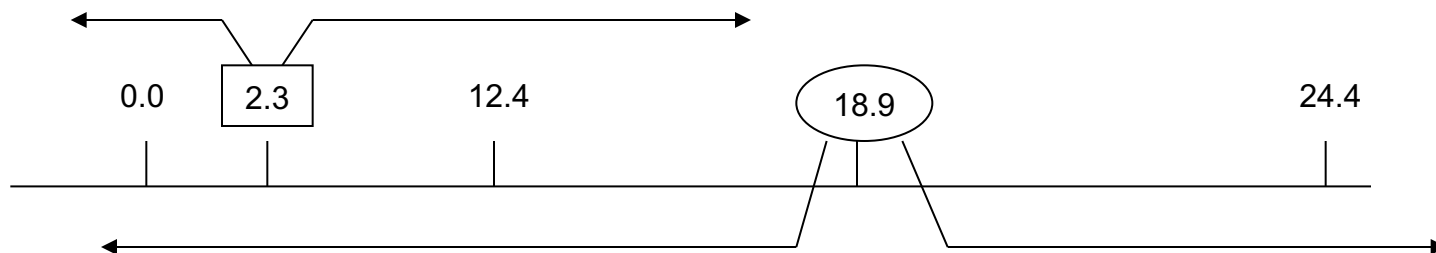
$$Gain(A, T; S) < \frac{\log_2(N - 1)}{N} + \frac{\Delta(A, T; S)}{N}$$

$$\Delta(A, T; S) = \log_2(3^k - 2) - [k \cdot Ent(S) - k_1 \cdot Ent(S_1) - k_2 \cdot Ent(S_2)],$$

$N = \#S, k = \#classes, k_i = \#classes \text{ em } S_i$

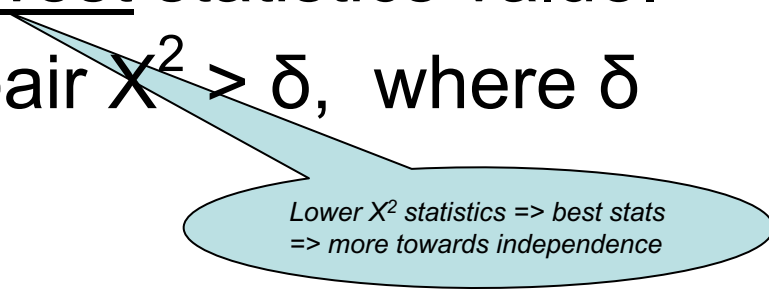
Required information to specify the split point.

- Process stops when this condition is satisfied.



ChiMerge

- Join adjacent intervals that yield independence on the class attribute.
- Measure independence using the χ^2 test. Choose the pair with the lowest statistics value.
- Stop when for the chosen pair $\chi^2 > \delta$, where δ is given by the user.
- Initially the attribute values are sorted. Each value corresponds to a unitary interval.



Lower χ^2 statistics => best stats
=> more towards independence

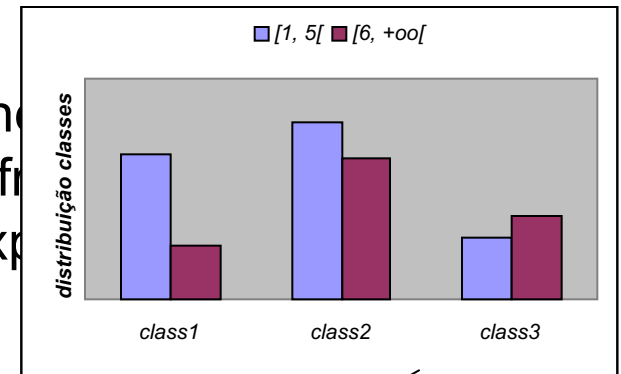
ChiMerge

- Compute χ^2

$$\chi^2 = \sum_{i=1}^2 \sum_{j=1}^k \frac{(A_{ij} - E_{ij})^2}{E_{ij}}$$

- Being:

- k is the number of classes, N the size of the data
- A_{ij} is the number of instances in interval i for class j
- $E_{ij} = \#(\text{interval } i) \times \#(\text{class } j) / N$, that is, expected



- Degrees of freedom = $\overbrace{(\#intervals - 1)}^{= 1} \times (\#classes - 1)$
- If the test yields dependency then the difference in the classes distribution between intervals is statistical significant. Once, the intervals should stay separated.

Srikant & Agrawal

- Performed during the execution of the FIM algorithm.
- Begins with a predefined number of intervals. This number is obtained by using a loss information measure (*Partial Completeness*).
- Join of adjacent intervals is controlled by the support of the resulting interval.
- The new derived items live together with the original items (intervals).
- This yields a varying number of information levels regarding the numeric attribute (with more or less granularity).

Join of Intervals

Dom(X) = {1,2,3,4}

Base intervals:

X=[1,1], sup=0.1

X=[2,2], sup=0.2

X=[3,3], sup=0.2

X=[4,4], sup=0.5

then we can have:

X=[1,2], sup=0.3

X=[3,4], sup=0.7

but also:

X=[1,3], sup=0.5

X=[2,4], sup=0.9

and even

X=[1,4], sup=1

These intervals are only generated if the user provided maximal support is satisfied!

A large number of items can be generated. For n values (or intervals) of an attribute one potentially has $O(n^2)$ items that contain a value/interval of that attribute.

A similar problem can arise with rules and confidence!

Partial Completeness

We would like to evaluate the degree of information loss that arises with the definition of the number of intervals. This measure enables to define the ideal number of intervals regarding the assumed admissible information loss.

Let C a set of itemsets from dataset D . For each $K > 1$, we say that P is K -complete in relation to C if:

- $P \subseteq C$,
- $X \in P$ and $X' \subseteq X \rightarrow X' \in P$,
- $\forall X \in C, \exists Z$ such that:
 - Z is a generalization of X and $s(Z) \leq K \times s(X)$,
 - $\forall Y \subseteq X, \exists Y'$ such that:
 - Y' is a generalization of Y and $s(Y') \leq K \times s(Y)$.

Example

num	itemset	sup
1	X=[20,30]	10
2	X=[20,40]	12
3	X=[20,50]	15
4	Y=[1,5]	10
5	Y=[1,7]	12
6	X=[20,30],Y=[1,5]	8
7	X=[20,40],Y=[1,7]	10

Itemsets 2,3,5,7 form a P 1.5-complete set.

Any itemset from attribute X has a generalization in this set where the support is at most 1.5 times superior to the support of X

Equi-depth intervals (with the same number of observed numeric values) will minimize the required number of intervals to establish a given K level of *partial completeness*.

Number of Intervals

$$num = \frac{2 * n}{ms * (k - 1)}$$

n is the number of numeric attributes. ms minimal support

As the value of completeness K decreases, the information loss decreases and the number of intervals increases.

Note that:

- To control the intervals join process one makes use of a maximal support threshold that by default equals ms .

Example

X	Y	CLASS
1	11	A
1	11	B
2	13	B
2	13	A
3	15	C
3	15	A
7	18	B
7	18	B
9	23	B
9	23	A

$$ms = 0.2 \quad K = 8$$

$$num = \frac{2 * 2}{(0.2 * 7)} = 3$$

base intervals:

$$X=[1,2], X=[3,7], X=[9,9]$$
$$Y=[11,13], Y=[15,18], Y=[23,23]$$

If max sup is 0.6, we can get through joining :

$$X=[3,9] \quad e \quad Y=[15,23]$$

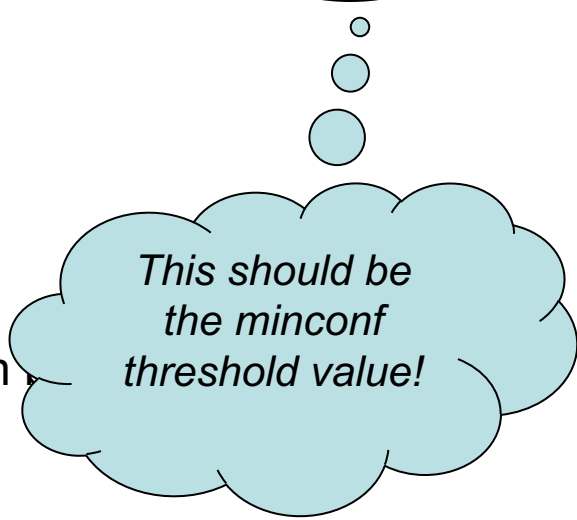
Rule generation from a K-complete set

Let:

- P is K -complete in relation to C
- R_C are rules that contain itemsets from C (minconf = mc)
- R_P are rules that contain itemsets from P (minconf = mc/K)

Then:

- $\forall a \rightarrow b \in R_C, \exists \underline{a} \rightarrow \underline{b} \in R_P$ such that:
 - \underline{a} is a generalization of a , \underline{b} is a generalization from b
 - $s(\underline{a} \rightarrow \underline{b}) \leq K \times s(a \rightarrow b)$
 - $\text{conf}(\underline{a} \rightarrow \underline{b}) \geq \text{conf}(a \rightarrow b) / K$
 - $\text{conf}(\underline{a} \rightarrow \underline{b}) \leq K * \text{conf}(a \rightarrow b)$



This should be the minconf threshold value!

Framework for SubGroup Mining

- To derive subgroups using association rules algorithms;
- Algorithms are *rule-based*.
- Detect deviation (*interest*) using statistical significance;
- Control specialization (*overfitting*) using the same type of statistical tests;
- Several types of rules depending on the specific application.

Identifying Interesting Subgroups

- Derive rules to identify *interesting* subpopulations that occur in the studied data

Subgroup_describing_characteristics → poi

- Property of interest (poi) can be a categorical attribute or numerical, a constraint expression or even a contrast!
- Several statistics computed for a rule

Association Rules with a Numeric Property of Interest

Main idea: Derive rules where the consequent represents a numerical property (which is the target of our study).

Examples:

Sex=female → Wage: mean=\$7.9 (overall mean=\$9.02)

non-smoker & wine-drinker → life-expectancy=85 (overall=80)

Association Rules with numerical properties (cont)

- Several proposals
 - Quantitative Association Rules (Aumann & Lindell99)
 - Impact Rules (Webb 2001)
 - Distribution Rules (Jorge & Azevedo 2006)
- Common idea to all proposals:

derive rules that represent the behaviour of the numerical property in an *interesting subpopulation*. Different proposals for the notion of interesting rule!

Association Rules with numerical properties (cont)

- Definition of interesting subpopulation.
 - QAR, makes use of a *z-test to confirm* interest (validity) of the rule. z-test between $\text{mean}_J(\text{Tx})$ and $\text{mean}(\text{D-Tx})$ with $\alpha=0.05$.

Rules of the type: $\text{subset}(X) \rightarrow \text{Mean}_J(\text{Tx})$ where
 $\text{Mean}_J(\text{Tx}) \neq \text{Mean}(\text{D-Tx})$

Complement
of Tx

$\text{z.test}(\mu_0, \text{observ}, \sigma)$: Used to find the significant differences between mean μ_0 and sample mean.

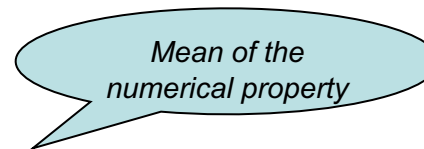
Computes the probability of the sample mean, obtained assuming the population mean and standard deviation (μ_0 e σ), being larger than the observed mean (assumes Gaussian Distribution)!

Concludes whether the sample belongs to the studied population.

Association Rules with numerical properties (cont)

Impact Rules (Webb)

- *Interest* refers to the notion of *impact*. Search for impact rules that maximize the definition of *impact*.
- Uses a *t-test* to evaluate significance: (tends to *z-test* with the increase in the number of degrees of freedom). More suitable to small samples. A procedure to compare means.
- Notion of Impact:



Mean of the numerical property

$$\text{Impact}(IR) = (\text{Mean}(IR) - \overline{poi}) \times |\text{cover}(\text{ant}(IR))|$$

Distribution Rules

- The consequent is a Distribution,
- Uses goodness-of-fit *Kolmogorov-Smirnov* test, to evaluate rule's interest.
- Notion of *interest*: Rule is interesting if *p-value* of

General population
Distribution

Rule's subgroup
distribution

$$\text{ks-test}(\text{apriori}, \text{rules-dist}) < \alpha$$

is less than a given threshold α supplied by the user.

Main Idea

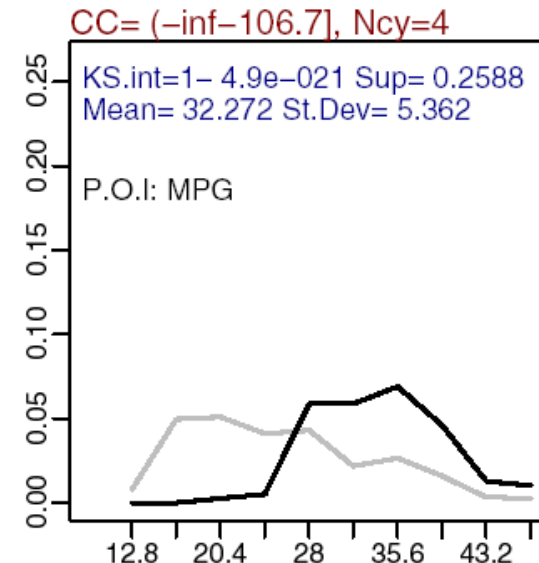
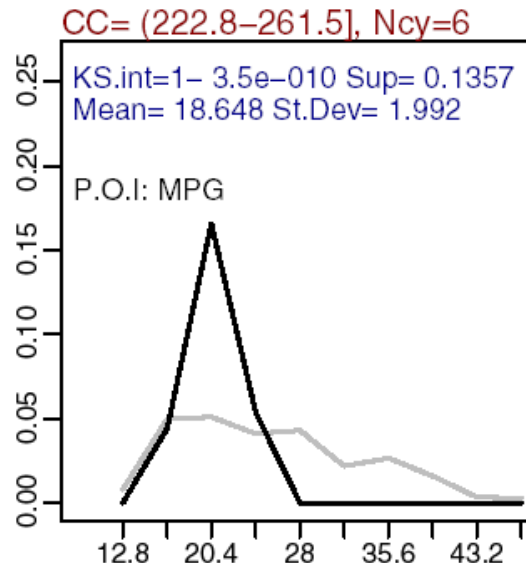
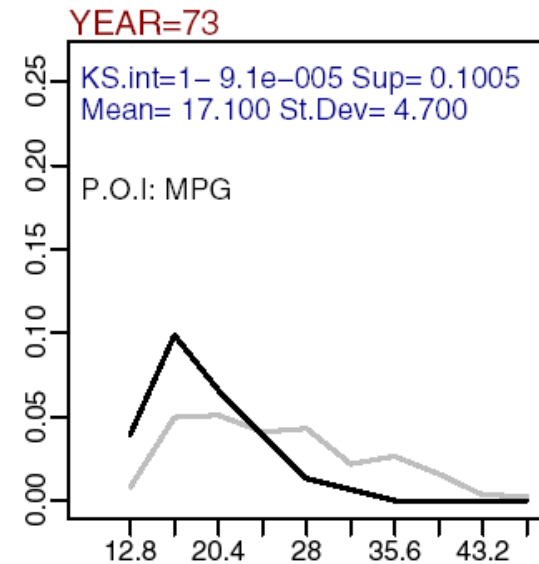
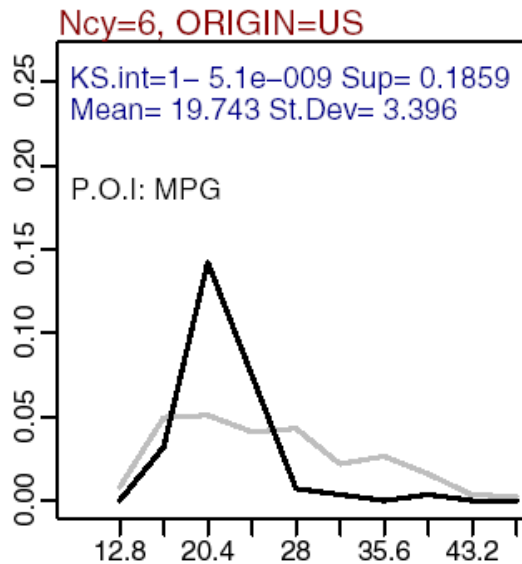
- Derive rules where the consequent is the distribution of the numeric property to be studied. The antecedent describes the subpopulation.
- Compare the *apriori* distribution (general population) against subgroup distribution (using the ks-test()).
- **Ex:** Ant-Sup = 0.14482

AGE={46/1,48/1,51/2,52/2,54/1,55/1,57/2,58/1,59/3,60/2,61/2,62/2,63/3,64/4,65/4,66/4,67/3,68/4,69/2,70/6,72/6,73/4,75/3,76/7,77/5,78/3,79/1,80/2,81/1,82/4,83/2,84/3,86/3,90/1 } ← ***TAVC=1 & DIAB=0***

Describes the distribution of AGE for the subpopulation (which represents 14,5% of the studied population) that had a stroke type 1 (AVC 1) and is non diabetic.

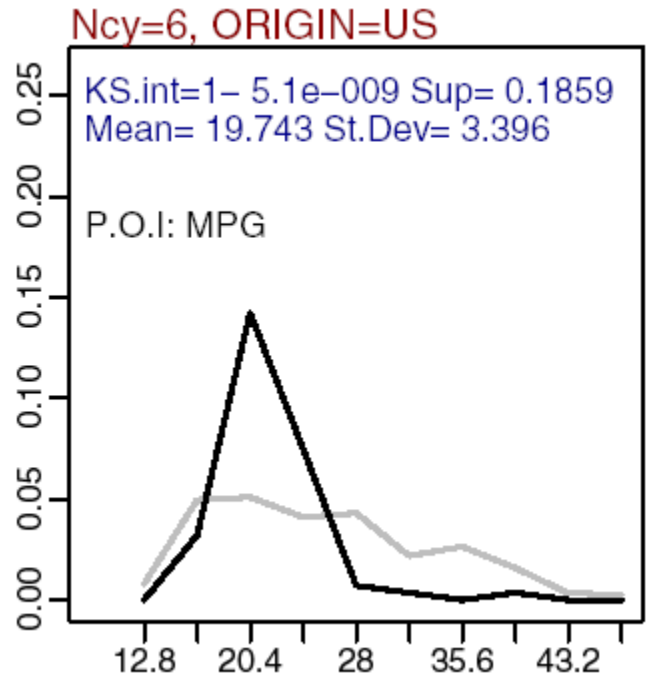
Distribution Rule presentation

- property of interest
- each DR is a plot
- distribution plot
 - frequency polygon
 - static binning
- distribution statistics
- comparison with default distribution



Compute interest of a DR

- KS-interest:
 - Given a rule $A \rightarrow y = D_{y|A}$, its KS-interest is $1-p$,
 - p is the p -value of the KS test comparing $D_{y|A}$ and $D_{y|\emptyset}$



•KS-improvement

- value added by the refinements of a rule
- $\text{imp}(A \rightarrow B)$ is

$$\min(\{\text{KS-interest}(A \rightarrow B) - \text{KS-interest}(A_s \rightarrow B) \mid A_s \subseteq A\})$$

Applications

- Descriptive data mining
 - dataset: Determinants of Wages from the 1985 Current Population Survey in the United States, a.k.a. Wages
 - property of interest: WAGE
- Rule discovery
 - min-sup=0.1, KS-int=0.95
 - minimal KS-improvement of 0.01
 - numerical attributes in the antecedent were pre-discretized
 - compact internal representation of rules
 - rules can be output as text or graphically

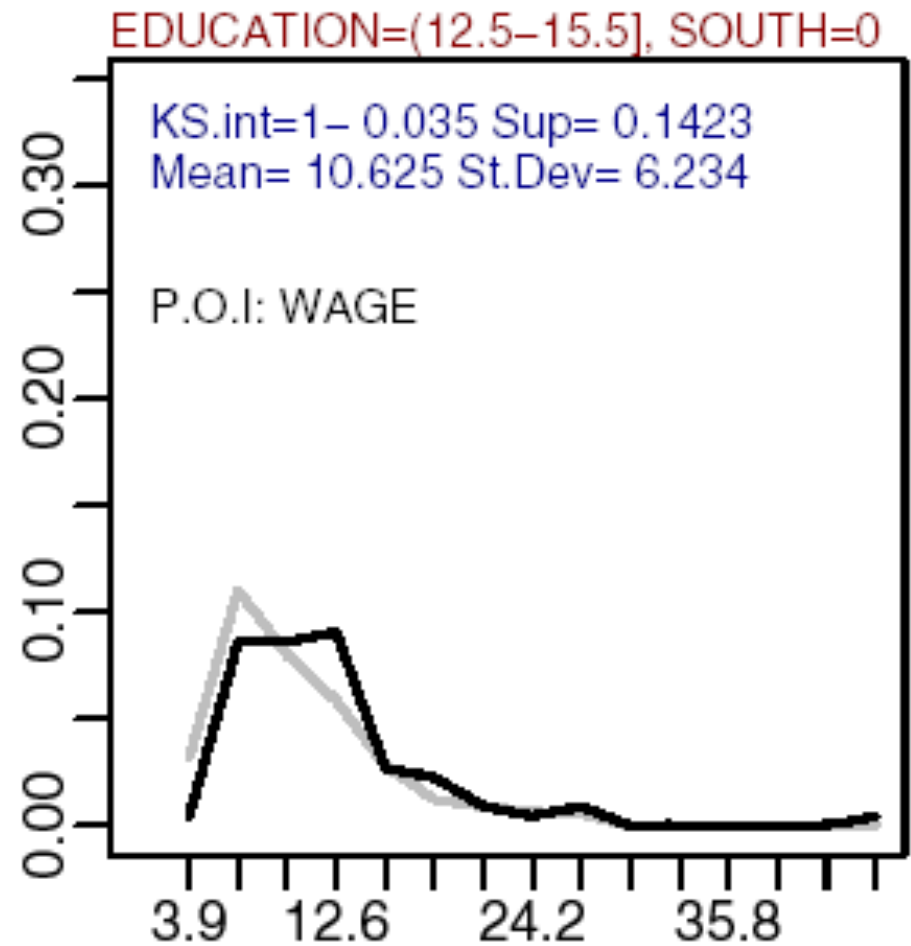
Sup=0.118 KS.int=1-0.0085 Mean=10.982 St.Dev=6.333

EDUCATION=(12.5-15.5] & SOUTH=0 & RACE=3

-> WAGE={ 3.98/1,4.0/1,4.17/1,4.5/1,4.55/1,4.84/1,5.0/1,5.62/1,5.65/1,5.8/1,6.0/1,6.25/4,7.14/1,7.5/1,7.67/1,7.7/1,7.96/1,
8.0/2,8.4/1,8.56/1,8.63/1,8.75/1,8.9/1,9.22/1,9.63/1,9.75/1,9.86/1,10.0/3,10.25/1,10.5/1,10.53/1,10.58/1,10.61/1,
11.11/1,11.25/2,12.0/1,12.47/1,12.5/4,13.07/1,13.75/1,13.98/1,14.29/1,15.0/1,16.0/1,16.14/1,16.42/1,17.25/1,17.86/1,
18.5/1,21.25/1,22.5/1,26.0/1,44.5/1 }

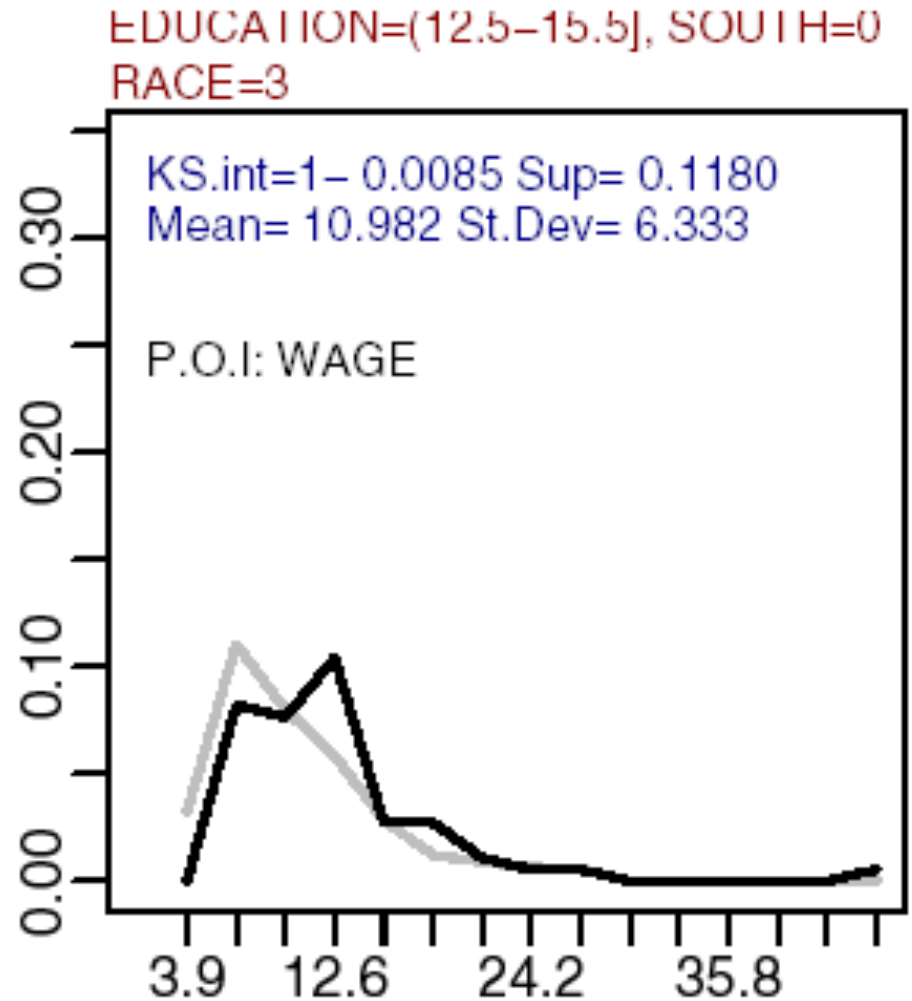
Using Distribution Rules

- antecedent
 - people with 13 to 15 years of education
 - not from the south
- consequent
 - wage distribution is better than the whole population but still concentrated on the same interval



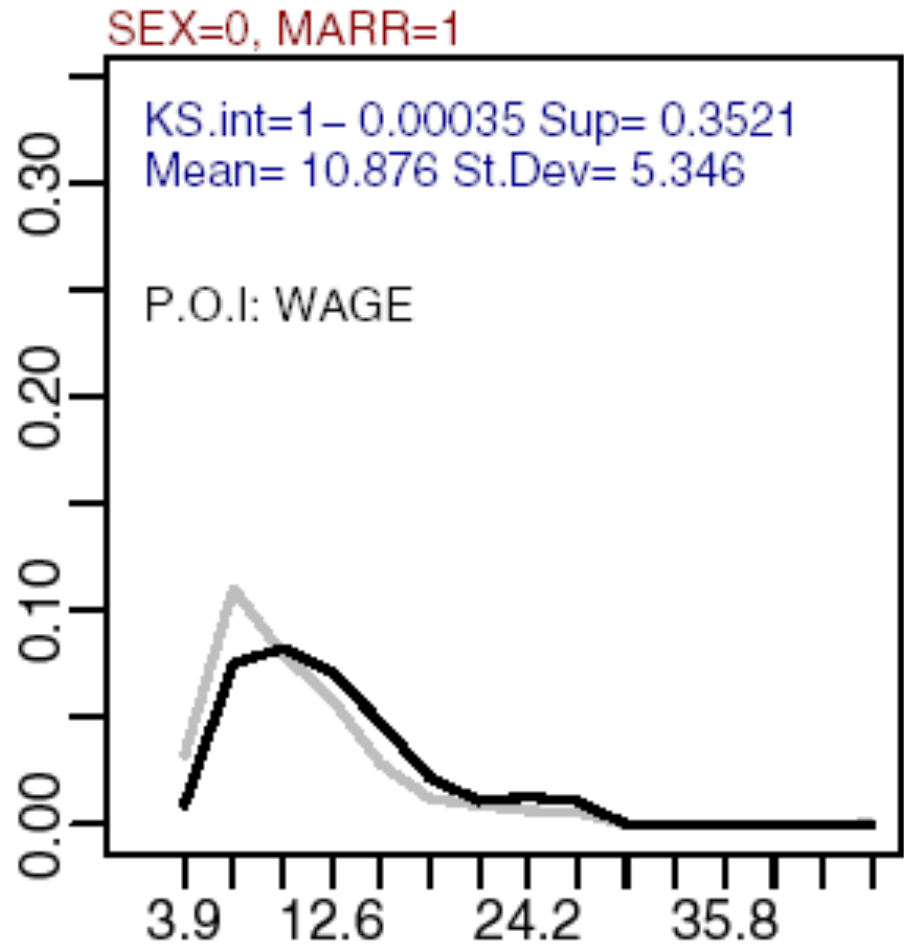
Using Distribution Rules

- antecedent
 - refinement of previous
 - race is white
- consequent
 - wage distribution is even better than before
 - KS-improvement is higher than 0.01
 - the wages still are concentrated on the same interval as before



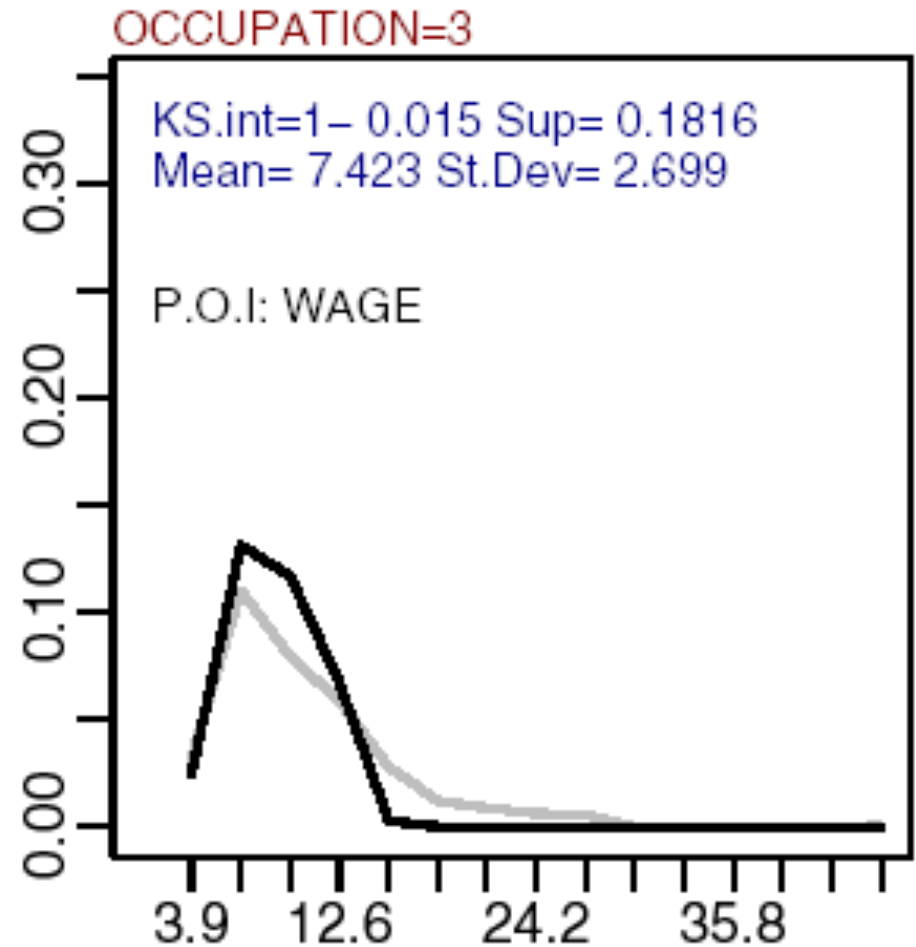
Using Distribution Rules

- antecedent
 - married males
- consequent
 - less interesting
 - still signif. different



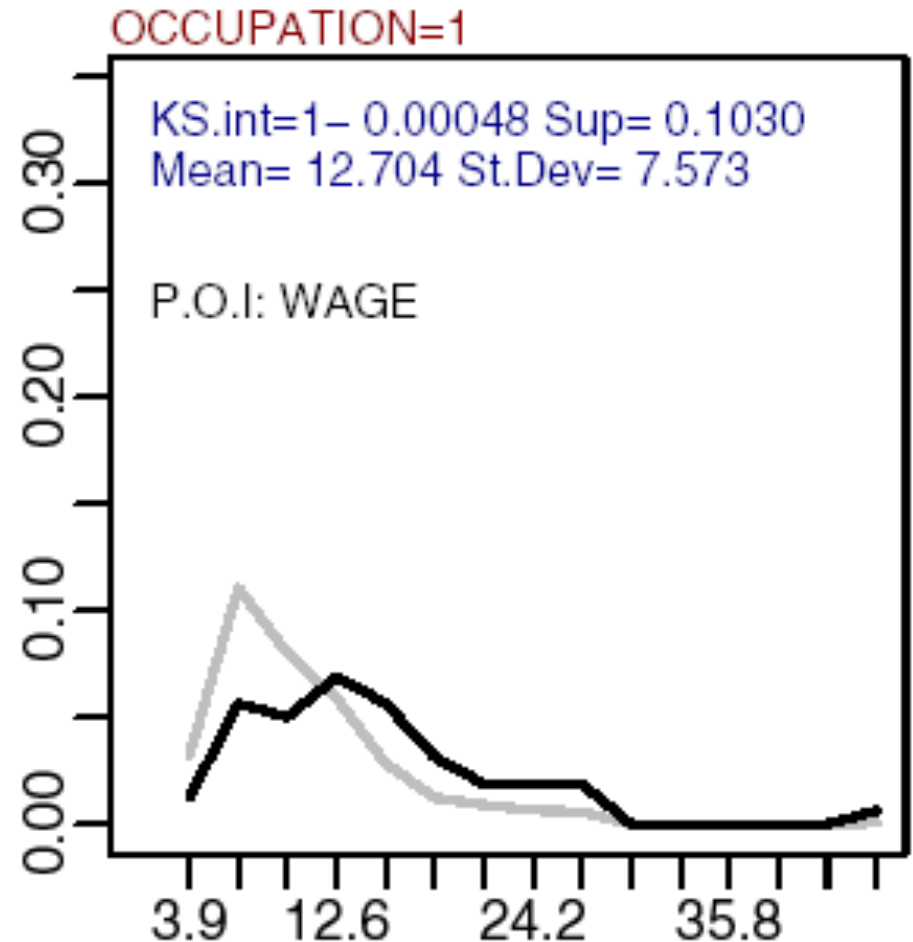
Using Distribution Rules

- antecedent
 - Occupation=Clerical
- consequent
 - concentrated on lower income



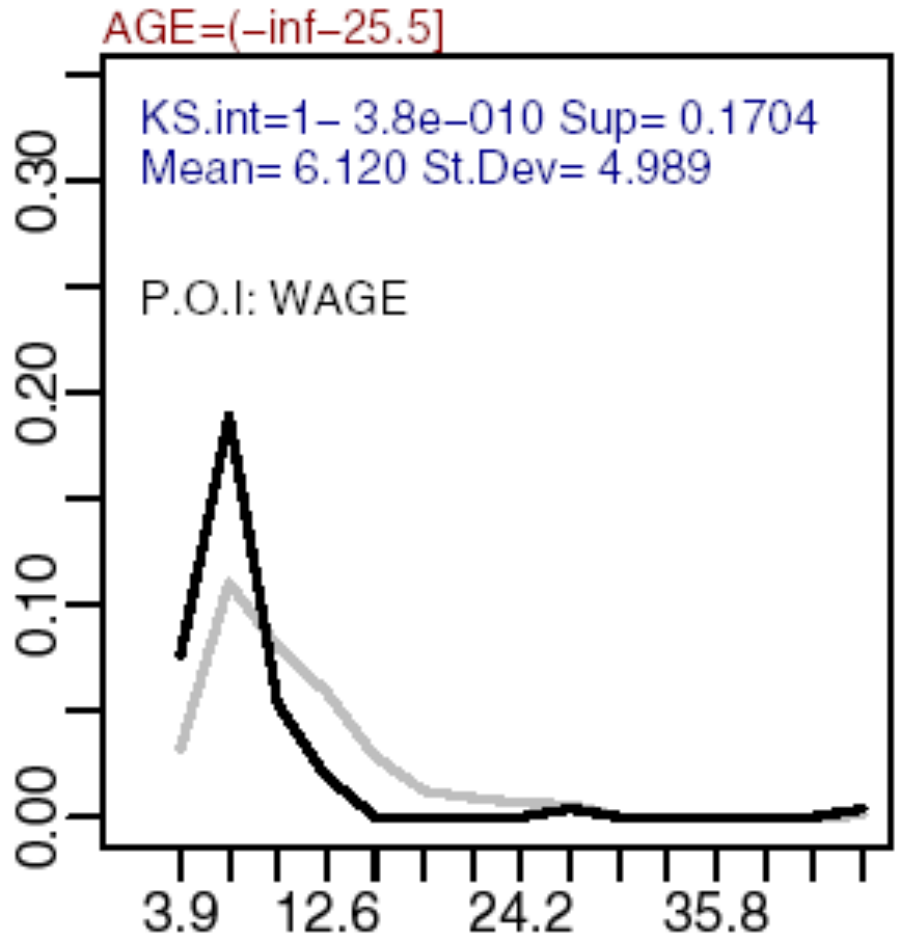
Using Distribution Rules

- antecedent
 - Occupation=Management
- consequent
 - clearly better wage distribution
 - we also observe a slightly lifted right tail



Using Distribution Rules

- antecedent
 - young people
- consequent
 - lower wages, very concentrated
 - some secondary modes are suggested



Max Leverage Rules

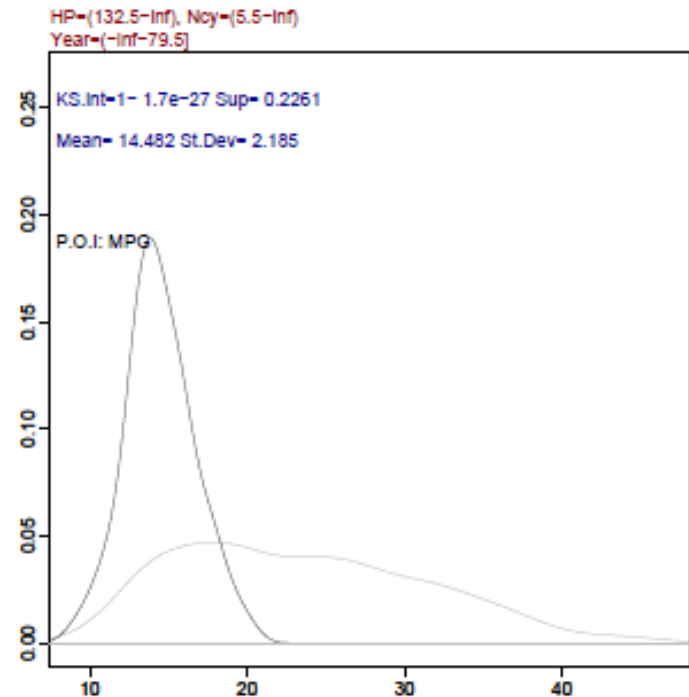
- [Jorge&Azevedo2011]
- Rules of the type:

$$\text{ant} \rightarrow A \in I,$$

where I is the interval that defines the maximal value for leverage (add value) of A for the antecedent ant , where A is our poi (property of interest).

- The rule is derived from the associated distribution rule.
- The intervals that maximize *leverage* I (*added value*) are obtained from the *KS* test.
- $AV(A \rightarrow C) = \text{conf}(A \rightarrow C) - \text{sup}(C)$.

Max Leverage Rules Example



(Cov=0.226 Lev=0.148 AV=0.653 Conf=0.922)

HP=(132.5-inf) & Ncy=(5.5-inf) & Year=(-inf-79.5] → MPG < 18

- The above rule states that cars with horse power (HP) above 132.5, more than 5 cylinders (Ncy) and made (Year)) before 1980 tend to have a miles per gallon (MPG) value below 18 when compared to a generic car.
- In fact, the rule says that a generic car will only have such a bad performance with much lower probability (0.653 lower to be precise, as it is given by the added value AV)

Contrast Sets

- *Rules for Contrast Sets* [Azevedo2010]
- Describe the difference between contrasting groups.
- A contrast set is a conjunction of characteristics that describes a subpopulation which occurs with different proportions along different groups.
- Examples:
 - Different temporal instances (sales in 1998 versus 1999),
 - Different locations (find distinct characteristics for the location of a gene x in human DNA in relation to mice DNA),
 - Along different classes (difference between brunettes and blonds).

RCS

- The characteristics of the subpopulation to be found (*contrast sets*) are interesting (significant) if the proportions of the individual occurrences along groups are significantly distinct.
- i.e. subpopulation is *not independent* to group belonging. Significance is computed using a Fisher exact test.

Gsup = 0.17191 | 0.04121 p = 1.1110878451E-017
Gsup = 0.17191 | 0.01681 p = 3.0718399575E-040
Sup(CS) = 0.03097

education=Doctorate >> education=Masters
education=Doctorate >> education=Bachelors
← workclass=State-gov & class > 50K.

In our case
defined by
attribute
education

- Specialization of a *contrast set* is controlled also through a Fisher test.

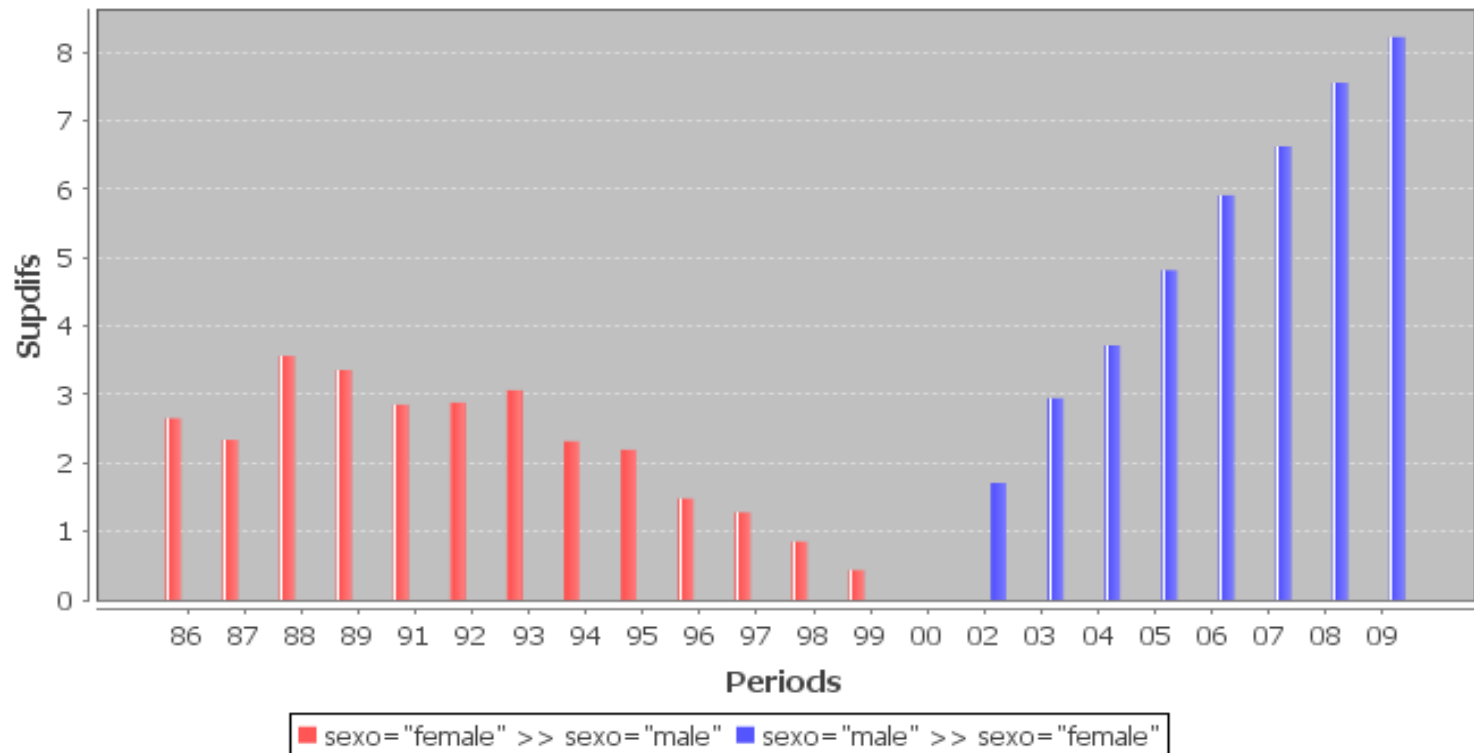
Case Study

Data representing employment from the Portuguese private sector between 1986 and 2009.

Ant: educ="5-9"

Stability (sexo="female" >> sexo="male"): 0.55

Stability (sexo="male" >> sexo="female"): 0.26

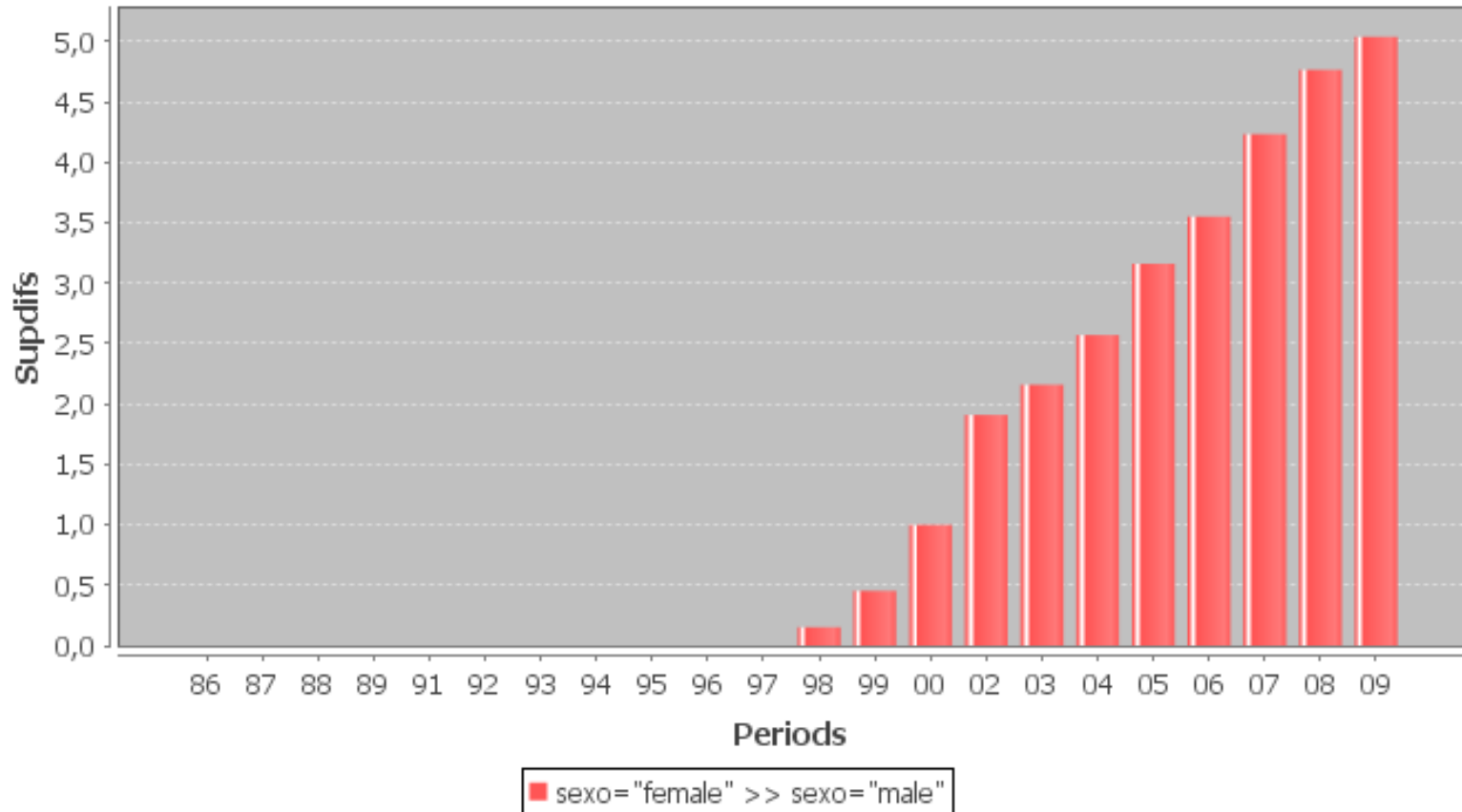


- *Contrast on individuals with basic (lower) education*

Case Study

Ant: educ=">12"

Stability (sexo="female" >> sexo="male"): 0.48



- *Contrast found on individuals with higher education*

Conclusions

- Several algorithms to compute associations between atomic elements in the data (items),
- Derive rules that describe association between atomic elements of the data.
- Selecting interesting and significant rules,
- Dealing with non categorical data

- Analysis of numerical properties of interest