Chapter VII:
Frequent Itemsets
& Association Rules

Information Retrieval & Data Mining

Universität des Saarlandes, Saarbrücken

Winter Semester 2011/12

Chapter VII:
Frequent Itemsets & Association Rules

VII.1 Definitions

Transaction data, frequent itemsets, closed and maximal itemsets, association rules

VII.2 The Apriori Algorithm

Monotonicity and candidate pruning, mining closed and maximal itemsets

VII.3 Mininig Association Rules

Apriori, hash-based counting & extensions

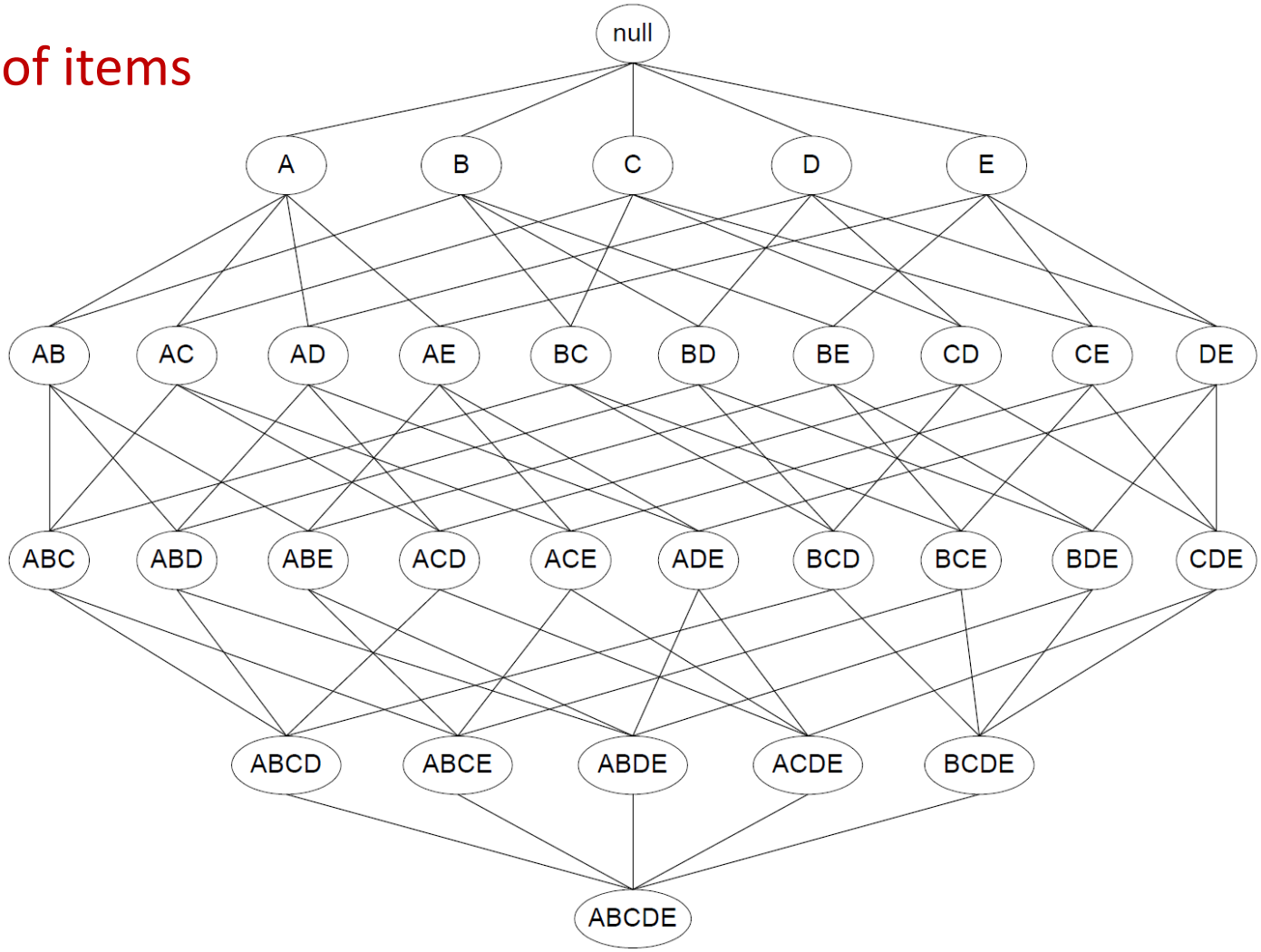
VII.4 Other measures for Association Rules

Properties of measures

Following Chapter 6 of
Mohammed J. Zaki, Wagner Meira Jr.: *Fundamentals of Data Mining Algorithms*.

VII.2 Apriori Algorithm for Mining Frequent Itemsets

Lattice of items



A Naïve Algorithm For Frequent Itemsets

- Generate all possible itemsets (lattice of itemsets):
  Start with 1-itemsets, 2-itemsets, ..., d-itemsets.

- Compute the frequency of each itemset from the data:
  Count in how many transactions each itemset occurs.

- If the support of an itemset is above **minsupp**
  then **report it as a frequent itemset**.

Runtime:

- Match every candidate against each transaction.
- For *M candidates* and $N=|D|$ *transactions*, the complexity is: $O(N\,M)$ => this is very expensive since $M = 2^{|I|}$

Speeding Up the Naïve Algorithm

- Reduce the **number of candidates** (M):
  - Complete search: $M=2^{|I|}$
  - Use pruning techniques to reduce M.

- Reduce the **number of transactions** (N):
  - Reduce size of N as the size of itemset increases.
  - Use vertical-partitioning of the data to apply the mining algorithms.

- Reduce the **number of comparisons** (N*M)
  - Use efficient data structures to store the candidates or transactions.
  - No need to match every candidate against every transaction.

Reducing the Number of Candidates

- Apriori principle (main observation):

– ***If an itemset is frequent, then all of its subsets must also be frequent.***

- Anti-monotonicity property (of support):

– ***The support of an itemset never exceeds the support of any of its subsets.***

Apriori Algorithm: Idea and Outline

Outline:

- Proceed in phases i=1, 2, ..., each making a single pass over D, and generate item set X with |X|=i in phase i;
- Use phase i-1 results to limit work in phase i:

Anti-monotonicity property (downward closedness):

For i-item-set X to be frequent,
each subset $X' \subseteq X$ with |X'|=i-1 must be frequent, too;

Worst-case time complexity still is exponential in |I| and linear in |D|*|I|, but usual behavior is linear in N=|D|.
(detailed average-case analysis is strongly data dependent, thus difficult)

Apriori Algorithm: Pseudocode

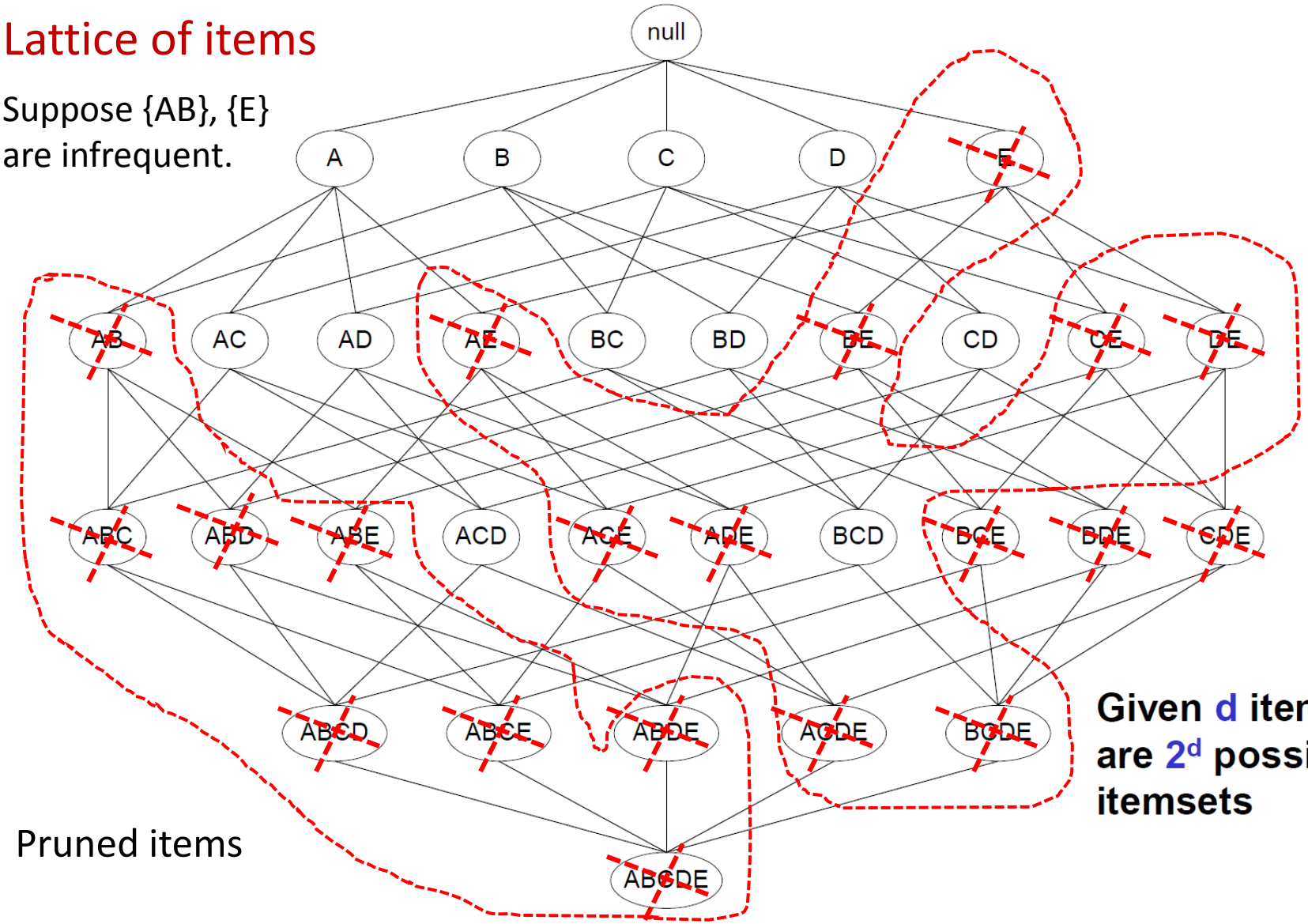
```
procedure apriori (D, min-support):
   L_1 = frequent 1-itemsets(D);
   for (k=2; L_{k-1} ≠ ∅; k++) {
      C_k = apriori-gen (L_{k-1}, min-support);
      for each t ∈ D {   // linear scan of D
         C_t = subsets of t that are in C_k;
         for each candidate c ∈ C_t  {c.count++} }; //end for
      L_k = {c ∈ C_k | c.count ≥ min-support} }; //end for
   return L = ∪_k L_k;    // returns all frequent item sets
```

```
procedure apriori-gen (L_{k-1}, min-support):
   C_k = ∅:
   for each itemset x_1 ∈ L_{k-1} {
      for each itemset x_2 ∈ L_{k-1} {
         if x_1 and x_2 have k-2 items in common and differ in 1 item {   // join
            x = x_1 ∪ x_2;
            if there is a subset s ⊆ x with s ∉ L_{k-1} {disregard x}    // infreq. subset
            else {add x to C_k} } } };
   return C_k;
```

Illustration For Pruning Infrequent Itemsets

Lattice of items

Suppose {AB}, {E} are infrequent.



Pruned items

Given d items, there are 2^d possible itemsets

Using Just One Pass over the Data

Idea:

Do not use the database for counting support after the 1st pass anymore!

Instead, use data structure C_k' for counting support in every step:

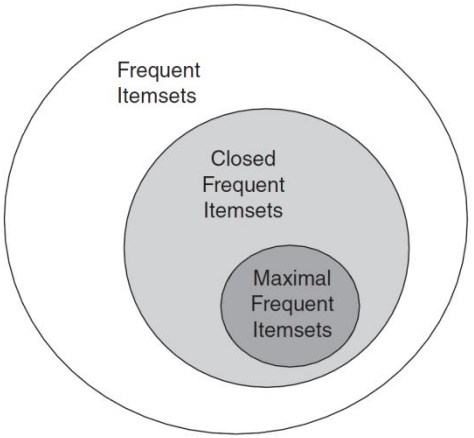
- C_k' = {<TID, $\{X_k\}$> | X_k is a potentially frequent k-itemset in transaction with id=TID}
- C_1': corresponds to the original database
- The member C_k' corresponding to transaction t is defined as <t.TID, $\{c \in C_k \mid c \text{ is contained in } t\}$>

AprioriTID Algorithm: PseudoCode

procedure apriori (D, min-support):
  L_1 = frequent 1-itemsets(D);
  C_1' = D;
  for (k=2; $L_{k-1} \neq \emptyset$; k++) {
    C_k = apriori-gen (L_{k-1}, min-support);
    C_k' = $\emptyset$
    for each t $\in$ C_{k-1}' { // linear scan of C_{k-1}' instead of D
      C_t = {c $\in$ C_k | t[c – c[k]]=1 and t[c – c[k-1]]=1};
      for each candidate c $\in$ C_t {c.count++};
      if ($C_t \neq \emptyset$) {C_k' = C_k' $\cup$ C_t};
    }; // end for
    L_k = {c $\in$ C_k | c.count $\geq$ min-support}
  }; // end for
  return L = $\cup_k$ L_k; // returns all frequent item sets

procedure apriori-gen (L_{k-1}, min-support):
  … // as before

Mining Maximal and Closed Frequent Itemsets with Apriori



Naïve Algorithm: (Bottum-Up Approach)

1) Compute all **frequent itemsets** using Apriori.

2) Compute all **closed itemsets** by checking all subsets of frequent itemsets found in 1).

3) Compute all **maximal itemsets** by checking all subsets of closed and frequent itemsets found in 2).

CHARM Algorithm (I)
for Mining Closed Frequent Itemsets
[Zaki, Hsiao: SIAM'02]

Basic Properties of Itemset-TID-Pairs:

Let t(X) denote the transaction ids associated with X.
Let $X_1 \leq X_2$ (for under any suitable order function, e.g., lexical order).

1) If $\mathbf{t(X_1) = t(X_2)}$, then $t(X_1 \cup X_2) = t(X_1) \cap t(X_2) = t(X_1) = t(X_2)$.
→ Replace X_1 with $X_1 \cup X_2$, remove X_2 from further consideration.

2) If $\mathbf{t(X_1) \subset t(X_2)}$, then $t(X_1 \cup X_2) = t(X_1) \cap t(X_2) = t(X_1) \neq t(X_2)$.
→ Replace X_1 with $X_1 \cup X_2$. Keep X_2, as it leads to a different closure.

3) If $\mathbf{t(X_1) \supset t(X_2)}$, then $t(X_1 \cup X_2) = t(X_1) \cap t(X_2) = t(X_2) \neq t(X_1)$.
→ Replace X_2 with $X_1 \cup X_2$. Keep X_1, as it leads to a different closure.

4) Else if $\mathbf{t(X_1) \neq t(X_2)}$, then $t(X_1 \cup X_2) = t(X_1) \cap t(X_2) \neq t(X_2) \neq t(X_1)$.
→ Do not replace any itemsets. Both X_1 and X_2 lead to different closures.

CHARM Algorithm (II)

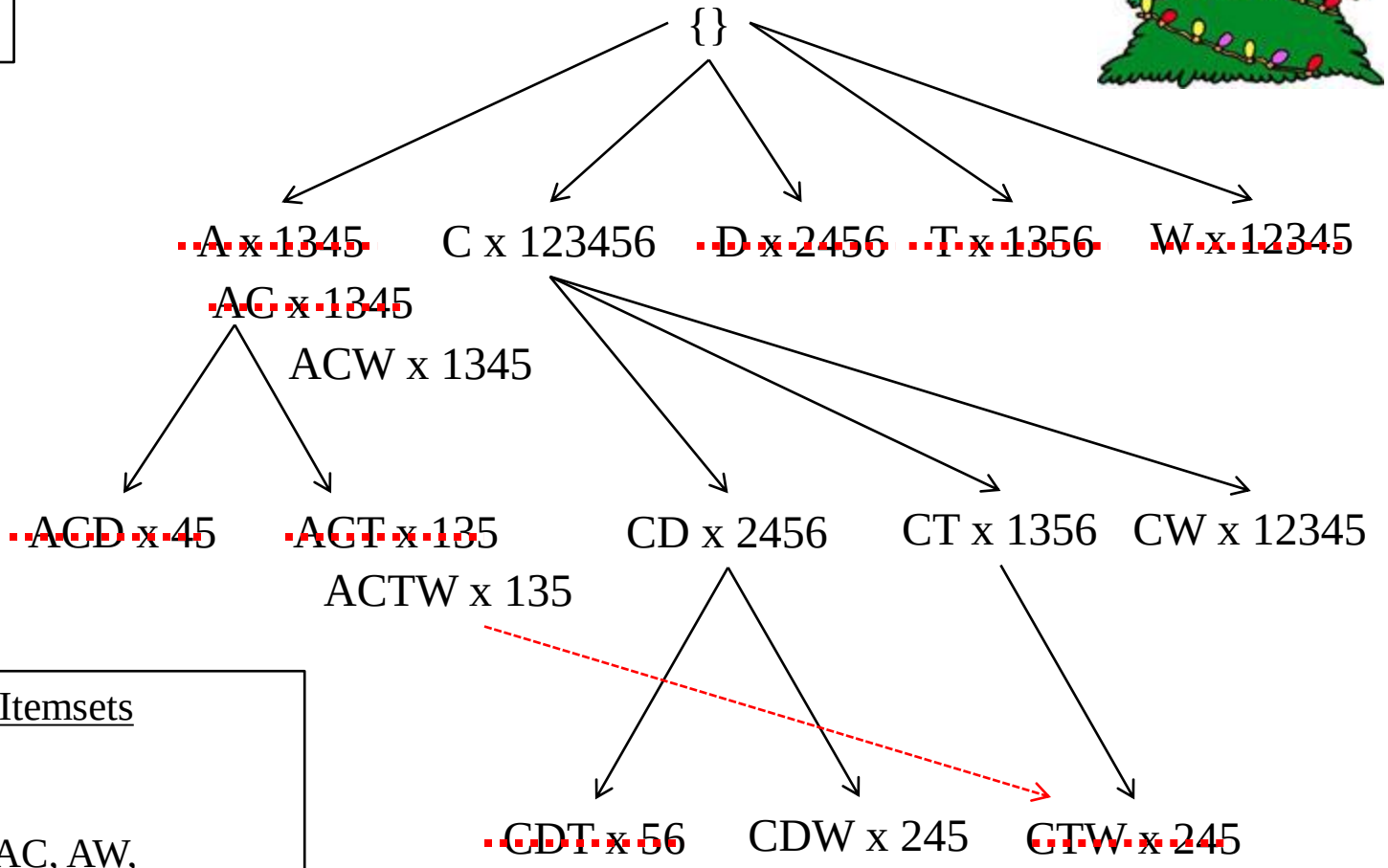
for Mining Closed Frequent Itemsets

[Zaki, Hsiao: SIAM'02]

Items: A C D T W

Transactions

1	ACTW
2	CDW
3	ACTW
4	ACDW
5	ACDTW
6	CDT



Support	Frequent Itemsets
100%	C
84%	W, CW
67%	A, D, T, AC, AW, CD, CT, ACW
50%	AT, DW, TW, ACT, ATW, CDW, CTW, ACTW

Done in 10 steps, found 7 closed & frequent itemsets!

VII.3 Mining Association Rules

Given:

- A set of **items** $I = \{x_1, ..., x_m\}$
- A set (bag) $D=\{t_1, ..., t_n\}$
  of **itemsets (transactions)** $t_i = \{x_{i1}, ..., x_{ik}\} \subseteq I$

Wanted:

Association rules of the form $\mathbf{X \Rightarrow Y}$ with $X \subseteq I$ and $Y \in I$ such that

- X is sufficiently often a subset of the itemsets t_i, and
- when $X \subseteq t_i$ then most frequently $Y \in t_i$ holds as well.

support $(X \Rightarrow Y)$ = absolute frequency of itemsets that contain X and Y

frequency $(X \Rightarrow Y) = support(X \Rightarrow Y) / |D| = \mathbf{P[XY]}$ relative frequency
frequency of itemsets that contain X and Y

confidence $(X \Rightarrow Y) = \mathbf{P[Y|X]}$ = relative frequency of itemsets
that contain Y provided they contain X

Support is usually chosen to be low (in the range of 0.1% to 1% frequency), confidence (aka. strength) in the range of 90% or higher.

Association Rules: Example

Market basket data ("sales transactions"):

t1 = {Bread, Coffee, Wine}
t2 = {Coffee, Milk}
t3 = {Coffee, Jelly}
t4 = {Bread, Coffee, Milk}
t5 = {Bread, Jelly}
t6 = {Coffee, Jelly}
t7 = {Bread, Jelly}
t8 = {Bread, Coffee, Jelly, Wine}
t9 = {Bread, Coffee, Jelly}

frequency (Bread $\Rightarrow$ Jelly) = 4/9
frequency (Coffee $\Rightarrow$ Milk) = 2/9
frequency (Bread, Coffee $\Rightarrow$ Jelly) = 2/9

confidence (Bread $\Rightarrow$ Jelly) = 4/6
confidence (Coffee $\Rightarrow$ Milk) = 2/7
confidence (Bread, Coffee $\Rightarrow$ Jelly) = 2/4

<u>Other applications:</u>

- book/CD/DVD purchases or rentals
- Web-page clicks and other online usage

etc. etc.

Mining Association Rules with Apriori

Given a frequent itemset X, find all non-empty subsets $Y \subset X$ such that $Y \rightarrow X - Y$ satisfies the minimum confidence requirement.

- If {A,B,C,D} is a frequent itemset, candidate rules are:
  $ABC \rightarrow D$, $ABD \rightarrow C$, $ACD \rightarrow B$, $BCD \rightarrow A$, $A \rightarrow BCD$,
  $B \rightarrow ACD$, $C \rightarrow ABD$, $D \rightarrow ABC$, $AB \rightarrow CD$, $AC \rightarrow BD$,
  $AD \rightarrow BC$, $BC \rightarrow AD$, $BD \rightarrow AC$, $CD \rightarrow AB$

- If $|X| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \emptyset$ and $\emptyset \rightarrow L$).

Mining Association Rules with Apriori

How to efficiently generate rules from frequent itemsets?

- In general, confidence does not have an anti-monotone property.
  conf(ABC → D) can be larger or smaller than conf(AB → D)
- But confidence of rules generated from the same itemset has an anti-monotone property!
- Example:
  X = {A,B,C,D}:
  conf(ABC → D) ≥ conf(AB → CD) ≥ conf(A → BCD)

Why?

→ **Confidence is anti-monotone w.r.t. number of items on the RHS of the rule!**

Apriori Algorithm For Association Rules

Outline:

- Proceed in phases i=1, 2, ..., each making a single pass over D, and generate rules $X \Rightarrow Y$ with frequent item set X (sufficient support) and |X|=i in phase i;
- Use phase i-1 results to limit work in phase i:

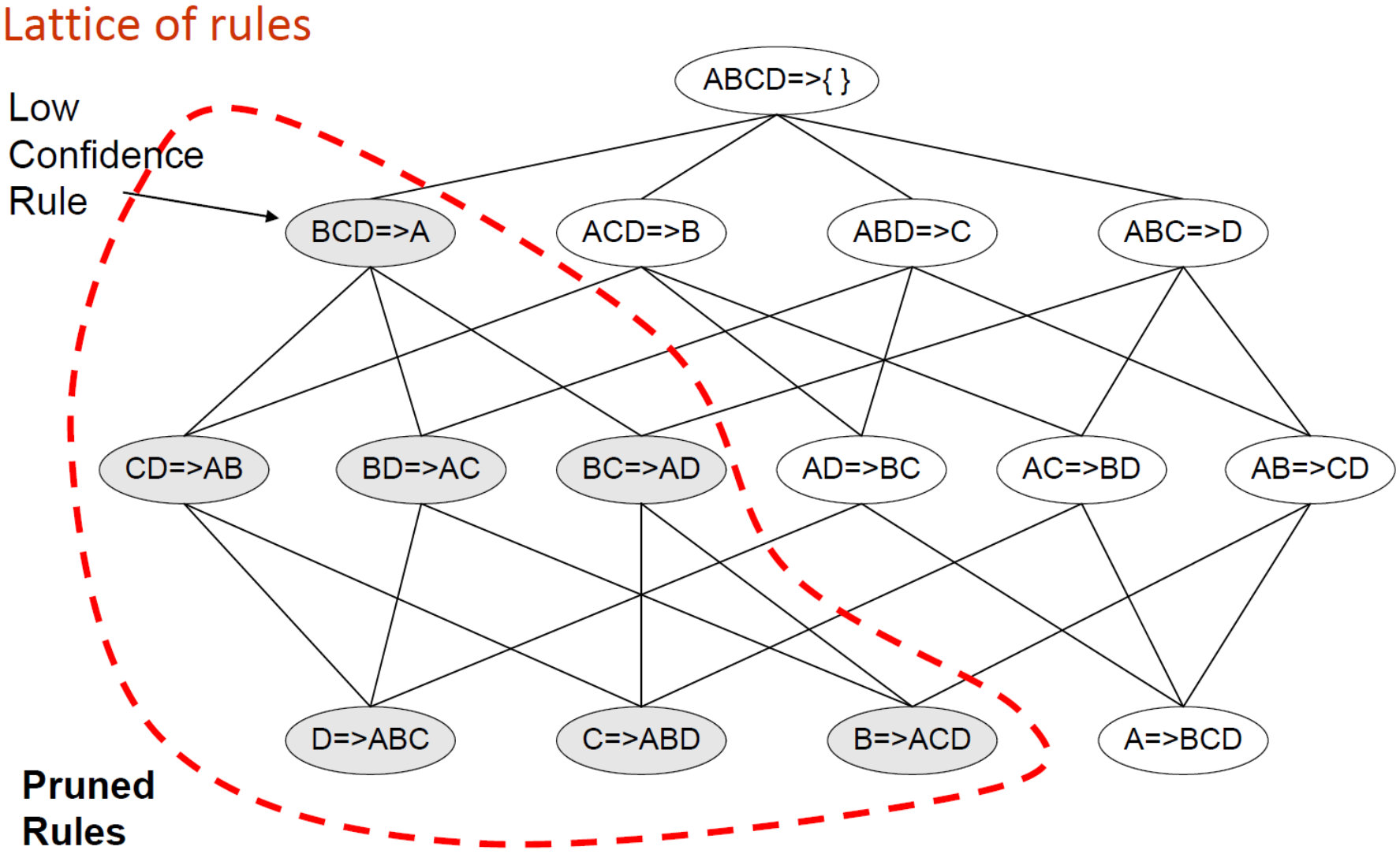
  Anti-monotonicity property (downward closedness):

  For i-item-set X to be frequent,

  each subset $X' \subseteq X$ with |X'|=i-1 must be frequent, too;

- Generate rules from frequent item sets;
- Test confidence of rules in final pass over D;

Illustration for Association Rule Mining



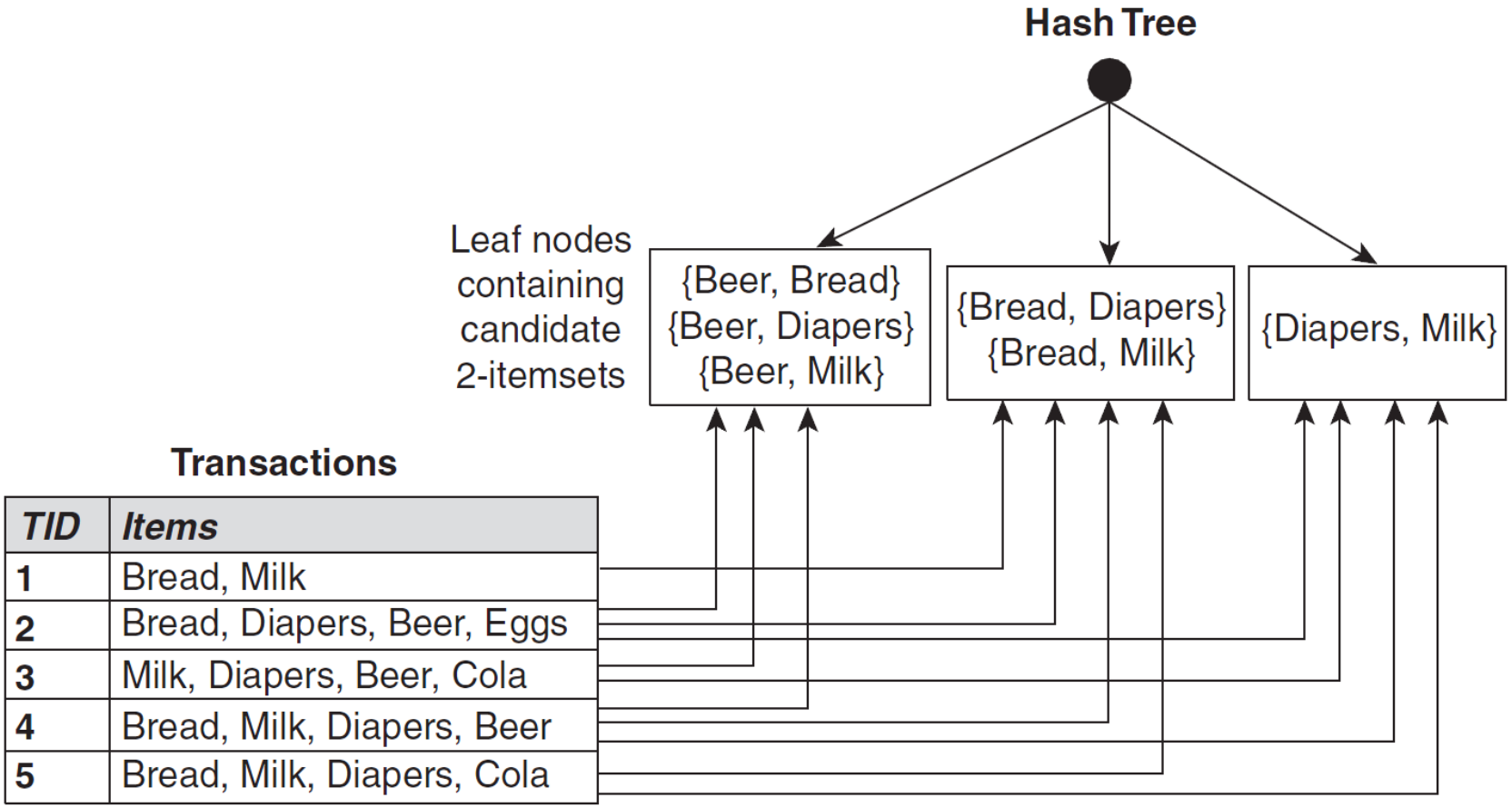
Algorithmic Extensions and Improvements

- **Hash-based counting** (computed during very first pass):
  map k-itemset candidates (e.g., for k=2) into hash table and maintain one count per cell; drop candidates with low count early.
- **Remove transactions** that don't contain frequent k-itemset for phases k+1, ...
- **Partition transactions** D:
  An itemset is frequent only if it is frequent in at least one partition.
- **Exploit parallelism** for scanning D.
- **Randomized (approximative) algorithms**:
  Find all frequent itemsets with high probability (using hashing, etc.).
- **Sampling** on a randomly chosen subset of D, then correct sample.

...

Mostly concerned about reducing disk I/O cost
(for TByte databases of large wholesalers or phone companies).

Hash-based Counting of Itemsets



Transactions

TID	*Items*
1	Bread, Milk
2	Bread, Diapers, Beer, Eggs
3	Milk, Diapers, Beer, Cola
4	Bread, Milk, Diapers, Beer
5	Bread, Milk, Diapers, Cola

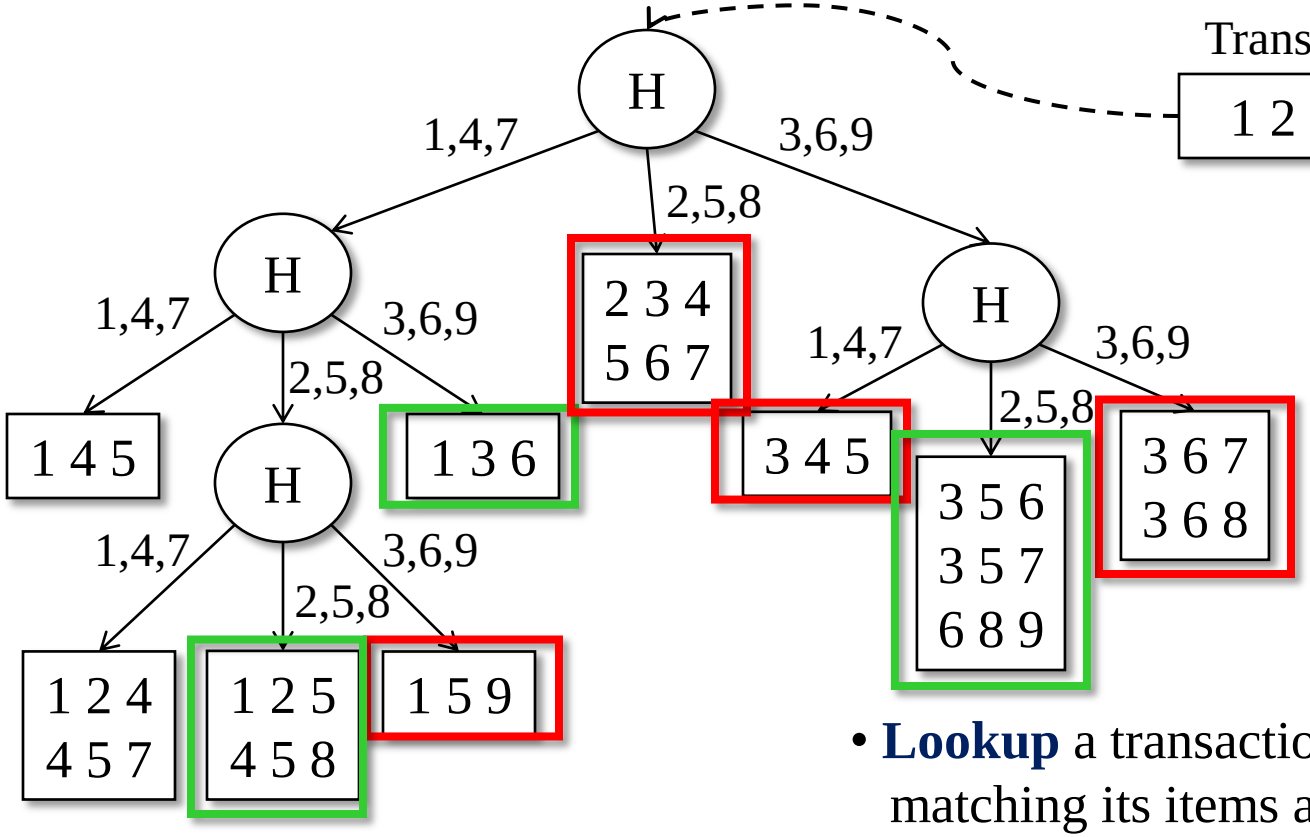
- During the main loop of Apriori, the support of candidate itemsets is calculated by matching each candidate against each transaction.
- This step can be accelerated by matching a candidate only against transactions that are relevant for this candidate (i.e., the ones that are contained in the *same bucket*).

Hash-Tree Index for Itemsets

Hash-tree for 3-itemsets:

- Inner nodes denote same hash-function H(p) = p mod 3
- Leaf nodes contain all candidate 3-itemsets

- **Build** hash-tree index by splitting candidate itemsets according to H
- Stop splitting into subsets if current split contains only one element



- **Lookup** a transaction by iteratively matching its items against H
- Check for containment if a leaf is reached

Extensions and Generalizations of Association Rules

- **Quantified rules**: consider quantitative attributes of item in transactions
  (e.g., wine between \$20 and \$50 $\Rightarrow$ cigars, or
  age between 30 and 50 $\Rightarrow$ married, etc.)
- **Constrained rules**: consider constraints other than count thresholds,
  (e.g., count itemsets only if average or variance of price exceeds ...)
- **Generalized aggregation rules**: rules referring to aggr. functions other than count (e.g., sum(X.price) $\Rightarrow$ avg(Y.age))
- **Multilevel association rules**: considering item classes
  (e.g., chips, peanuts, bretzels, etc., belonging to class "snacks")
- **Sequential patterns** (e.g., customers who purchase books in some order):
  combine frequent sequences $x_1\ x_2\ \ldots\ x_n$ and $x_2\ \ldots\ x_n\ x_{n+1}$
  into frequent-sequence candidate $x_1\ x_2\ \ldots\ x_n\ x_{n+1}$
- From strong rules to **interesting rules**:
  consider also lift (aka. interest) of rule X $\Rightarrow$Y: P[XY] / P[X]P[Y]
- **Correlation rules** (see next slides)

VII.4 Other Measures For Association Rule Mining

Limitations of support and confidence:

(a) Many interesting items might fall below minsupp threshold!
(b) Confidence ignores the support of the itemset in the consequent!

Consider contingency table (assume n=100 transactions):

	T	¬T	
C	20	70	90
¬C	5	5	10
	25	75	100

Consider the rule: **tea ⇒ coffee**
→ support(tea ⇒ coffee) = 20
→ confidence(tea ⇒ coffee) = 0.8

But support of coffee alone is 90, and of tea alone it is 25. That is, drinking coffee makes you less likely to drink tea, and drinking tea makes you less likely to drink coffee!
→ Tea and coffee have **negative correlation**!

Correlation Rules

Example for strong, but misleading association rule:
tea $\Rightarrow$ coffee with confidence 80% and support 20
But support of coffee alone is 90, and of tea alone it is 25
$\rightarrow$ tea and coffee have negative correlation!

Consider contingency table (assume n=100 transactions):

	T	¬T	
C	20	70	90
¬C	5	5	10
	25	75	100

$\rightarrow$ {T, C} is a frequent and correlated item set

$$\chi^2(C,T) = \sum_{X \in \{C,\overline{C}\}} \sum_{Y \in \{T,\overline{T}\}} \frac{(supp(X \wedge Y) - supp(X)\,supp(Y)/n)^2}{supp(X)\,supp(Y)/n}$$

Correlation rules are **monotone** (upward closed):
If the set X is correlated then every superset X' $\supset$ X is correlated, too.

Correlation Rules

Example for strong, but misleading association rule:

tea ⇒ coffee with confidence 80% and support 20

But support of coffee alone is 90, and of tea alone it is 25

→ tea and coffee have negative correlation!

Consider contingency table (assume n=100 transactions):

	T	¬T	
C	20	70	90
¬C	5	5	10
	25	75	100

E[C]=0.9
E[T]=0.25
E[(T-E[T])2]=1/4 * 9/16 +3/4 * 1/16= 3/16=Var(T)
E[(C-E[C])2]=9/10 * 1/100 +1/10 * 1/100 = 9/100=Var(C)
E[(T-E[T])(C-E[C])]=
2/10 * 3/4 * 1/10
– 7/10 * 1/4 * 1/10
– 5/100 * 3/4 * 9/10
+ 5/100 * 1/4 * 9/10 =
60/4000 – 70/4000 – 135/4000 + 45/4000 = – 1/40 = Cov(C,T)
ρ(C,T) = – 1/40 * 4/sqrt(3) * 10/3 ≈ -1/(3*sqrt(3)) ≈ – 0.2

Correlated Item Set Algorithm

```
procedure corrset (D, min-support, support-fraction, significance-level):
  for each x ∈ I compute count O(x);
  initialize candidates := ∅; significant := ∅;
  for each item pair x, y ∈ I with O(x) > min-support and O(y) > min-support {
      add (x,y) to candidates};
  while (candidates  ≠ ∅) {
      notsignificant := ∅;
      for each itemset x ∈ candidates {
          construct contingency table T;
          if (percentage of cells in T with count > min-support
            is at least support-fraction) {  // otherwise too few data for chi-square
              if  (chi-square value for T ≥ significance-level)
              {add X to significant} else {add X to notsignificant} } };  // if/for

      candidates := itemsets with cardinality k such that
                    every subset of cardinality k-1 is in notsignificant;
                    // only interested in correlated itemsets of min. cardinality
  }; //while
  return significant;
```

Examples of Contingency Tables

General form:
(for pair of variables A, B)

	B	¬B	
A	f_{11}	f_{10}	f_{1+}
¬A	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	N

Examples for binary cont. tables:

Example	f_{11}	f_{10}	f_{01}	f_{00}
E_1	8123	83	424	1370
E_2	8330	2	622	1046
E_3	3954	3080	5	2961
E_4	2886	1363	1320	4431
E_5	1500	2000	500	6000
E_6	4000	2000	1000	3000
E_7	9481	298	127	94
E_8	4000	2000	2000	2000
E_9	7450	2483	4	63
E_{10}	61	2483	4	7452

Symmetric Measures for Itemset {A,B}

Measure (Symbol)	Definition
Correlation (ϕ)	$\frac{N f_{11} - f_{1+} f_{+1}}{\sqrt{f_{1+} f_{+1} f_{0+} f_{+0}}}$
Odds ratio (α)	$\left(f_{11} f_{00}\right) / \left(f_{10} f_{01}\right)$
Kappa (κ)	$\frac{N f_{11} + N f_{00} - f_{1+} f_{+1} - f_{0+} f_{+0}}{N^2 - f_{1+} f_{+1} - f_{0+} f_{+0}}$
Interest (I)	$\left(N f_{11}\right) / \left(f_{1+} f_{+1}\right)$
Cosine (IS)	$\left(f_{11}\right) / \left(\sqrt{f_{1+} f_{+1}}\right)$
Piatetsky-Shapiro (PS)	$\frac{f_{11}}{N} - \frac{f_{1+} f_{+1}}{N^2}$
Collective strength (S)	$\frac{f_{11} + f_{00}}{f_{1+} f_{+1} + f_{0+} f_{+0}} \times \frac{N - f_{1+} f_{+1} - f_{0+} f_{+0}}{N - f_{11} - f_{00}}$
Jaccard (ζ)	$f_{11} / \left(f_{1+} + f_{+1} - f_{11}\right)$
All-confidence (h)	$\min \left[\frac{f_{11}}{f_{1+}}, \frac{f_{11}}{f_{+1}}\right]$

Asymmetric Measures For Rule A ⇒ B

Measure (Symbol)	Definition
Goodman-Kruskal (λ)	$\left(\sum_j \max_k f_{jk} - max_k f_{+k}\right) / \left(N - \max_k f_{+k}\right)$
Mutual Information (M)	$\left(\sum_i \sum_j \frac{f_{ij}}{N} \log \frac{N f_{ij}}{f_{i+} f_{+j}}\right) / \left(-\sum_i \frac{f_{i+}}{N} \log \frac{f_{i+}}{N}\right)$
J-Measure (J)	$\frac{f_{11}}{N} \log \frac{N f_{11}}{f_{1+} f_{+1}} + \frac{f_{10}}{N} \log \frac{N f_{10}}{f_{1+} f_{+0}}$
Gini index (G)	$\frac{f_{1+}}{N} \times \left(\frac{f_{11}}{f_{1+}}\right)^2 + \left(\frac{f_{10}}{f_{1+}}\right)^2] - \left(\frac{f_{+1}}{N}\right)^2$ $+ \frac{f_{0+}}{N} \times [\left(\frac{f_{01}}{f_{0+}}\right)^2 + \left(\frac{f_{00}}{f_{0+}}\right)^2] - \left(\frac{f_{+0}}{N}\right)^2$
Laplace (L)	$\left(f_{11} + 1\right) / \left(f_{1+} + 2\right)$
Conviction (V)	$\left(f_{1+} f_{+0}\right) / \left(N f_{10}\right)$
Certainty factor (F)	$\left(\frac{f_{11}}{f_{1+}} - \frac{f_{+1}}{N}\right) / \left(1 - \frac{f_{+1}}{N}\right)$
Added Value (AV)	$\frac{f_{11}}{f_{1+}} - \frac{f_{+1}}{N}$

Consistency of Measures

Ranking of tables according to symmetric measures

	ϕ	α	κ	I	IS	PS	S	ζ	h
E_1	1	3	1	6	2	2	1	2	2
E_2	2	1	2	7	3	5	2	3	3
E_3	3	2	4	4	5	1	3	6	8
E_4	4	8	3	3	7	3	4	7	5
E_5	5	7	6	2	9	6	6	9	9
E_6	6	9	5	5	6	4	5	5	7
E_7	7	6	7	9	1	8	7	1	1
E_8	8	10	8	8	8	7	8	8	7
E_9	9	4	9	10	4	9	9	4	4
E_{10}	10	5	10	1	10	10	10	10	10

Ranking of tables according to asymmetric measures

	λ	M	J	G	L	V	F	AV
E_1	1	1	1	1	4	2	2	5
E_2	2	2	2	3	5	1	1	6
E_3	5	3	5	2	2	6	6	4
E_4	4	6	3	4	9	3	3	1
E_5	9	7	4	6	8	5	5	2
E_6	3	8	6	5	7	4	4	3
E_7	7	5	9	8	3	7	7	9
E_8	8	9	7	7	10	8	8	7
E_9	6	4	10	9	1	9	9	10
E_{10}	10	10	8	10	6	10	10	8

- Rankings may vary substantially!
- Many measures provide conflicting information about quality of a pattern.
- Want to define generic properties of measures.

Properties of Measures

Definition (Inversion Property):
An objective measure M is invariant under the *inversion operation* if its value remains the same when exchanging the frequency counts f_{11} with f_{00} and f_{10} with f_{01}.

Definition (Null Addition Property):
An objective measure M is invariant under the *null addition operation* if it is not affected by increasing f_{00}, while all other frequency counts stay the same.

Definition (Scaling Invariance Property):
An objective measure M is invariant under the *row/column scaling* operation if M(T) = M(T'), where T is a contingency table with frequency counts $[f_{11}, f_{10}, f_{01}, f_{00}]$, T' is a contingency table with frequency counts $[k_1k_3f_{11}, k_2k_3f_{10}, k_1k_4f_{01}, k_2k_4f_{00}]$, and k_1, k_2, k_3, k_4 Are positive constants.

Example: Confidence and the Inversion Property

Recall the general form:

confidence($A \Rightarrow B$) := P[B|A]

	B	¬B	
A	f_{11}	f_{10}	f_{1+}
¬A	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	N

$= f_{11}/f_{1+} = f_{11} / f_{11}+ f_{10}$

$\neq f_{00} / f_{00} + f_{10} = f_{00}/f_{+0}$

(Inversion)

Counter example:

confidence($T \Rightarrow C$)

	T	¬T	
C	20	70	90
¬C	5	5	10
	25	75	100

$= 20/25 = 0.8 \neq 5/90 = 0.055$

Simpson's Paradox (I)

Consider the following correlation between people buying an HTDV (H) and an exercise machine (E):

	E	¬E	
H	99	81	180
¬H	54	66	120
	153	147	300

confidence($H \Rightarrow E$) = 99/180 = 0.55
confidence($\neg H \Rightarrow E$) = 54/120 = 0.45

→ Customers who buy an HDTV are more likely to buy an exercise machine than those who do not buy an HDTV.

Simpson's Paradox (II)

Consider stratified data by including additional variables
(data split two groups: college students and working employees):

		E	¬E	Total
Students (44)	H	1	9	10
	¬H	4	30	34
Employees (256)	H	98	72	170
	¬H	50	36	86

confidence(H ⇒ E) = 1/10 = 0.10 =: a/b

confidence(¬H ⇒ E) = 4/34 = 0.12 =: c/d

confidence(H ⇒ E) = 98/170 = 0.57 =: p/q

confidence(¬H ⇒ E) = 50/86 = 0.58 =: r/s

H and E are positively correlated in the combined data but negatively correlated in each of the strata!
When pooled together, the confidences of H ⇒ E and ¬H ⇒ E are (a+p)/(b+q) and (c+r)/(d+s), respectively.
Simpson's paradox occurs when: (a+p)/(b+q) > (c+r)/(d+s)

Summary of Section VII

Mining frequent itemset and association rules is a **versatile tool** for many applications (e-commerce, user recommendations, etc.).

One of the most basic building blocks in data mining for identifying interesting **correlations** among items/objects **based on co-occurrence statistics**.

Complexity issues mostly due to the huge amount of possible combinations of candidate itemsets (and rules), also expensive when amount of transactions is huge and needs to be read from disk.

Apriori builds on **anti-monotonicity property of support**, whereas confidence does not generally have this property (however pruning is possible to some extent within a given itemset).

Many **quality measures** considered in the literature, each with different properties.

Additional Literature:

M. J. Zaki and C. Hsiao: CHARM: An efficient algorithm for closed itemset mining. SIAM'02.