CMPS 566: Association Rule Mining

Vijay Raghavan

March 31, 2020

Agenda

- Association Mining
- Definitions and Example
- Apriori Algorithm
- Closed and Maximal Frequent Patterns
- Related Methods
- Correlation Analysis and Interestingness Measures

Association Mining

- Popular Data Mining Method.
- Attempts to find correlations between items and form 'interesting' rules.
- Market Basket Analysis
- Two step process
  - Find frequent itemsets
  - Form rules / patterns from frequent itemsets

Definitions

- The set of items $I = \{ i_1, i_2 \ldots i_m \}$ represents the set of all items in a transaction database.
- An itemset X is a subset of I.
- An itemset X is called k-itemset if its cardinality is k.

Definitions

- Support = probability that a transaction contains itemset X

- An itemset *X* is called a frequent itemset if support(X) > threshold (**minsup**).

Definitions

- Confidence is conditional probability that a transaction having X also contains Y

- An association rule is a pattern that states when X occurs, Y occurs satisfying a given confidence threshold (**minconf**).

Example: Frequent Itemset

- Transaction data

t1:	Beef, Chicken, Milk
t2:	Beef, Cheese
t3:	Cheese, Boots
t4:	Beef, Chicken, Cheese
t5:	Beef, Chicken, Clothes, Cheese, Milk
t6:	Chicken, Clothes, Milk
t7:	Chicken, Milk, Clothes

- Assume:
  - minsup = 30%

- Examples of frequent itemsets:
  - {Beef} [sup = 4/7]
  - {Chicken, Clothes, Milk} [sup = 3/7]

Example: Rules

- Transaction data

t1:	Beef, Chicken, Milk
t2:	Beef, Cheese
t3:	Cheese, Boots
t4:	Beef, Chicken, Cheese
t5:	Beef, Chicken, Clothes, Cheese, Milk
t6:	Chicken, Clothes, Milk
t7:	Chicken, Milk, Clothes

- Assume:
  - minsup = 30%
  - minconf = 80%
- Association rules from the itemset:
  - Clothes $\rightarrow$ Milk, Chicken, [sup = 3/7, conf = 3/3]
  - … …
  - Clothes, Chicken $\rightarrow$ Milk, [sup = 3/7, conf = 3/3]

Apriori

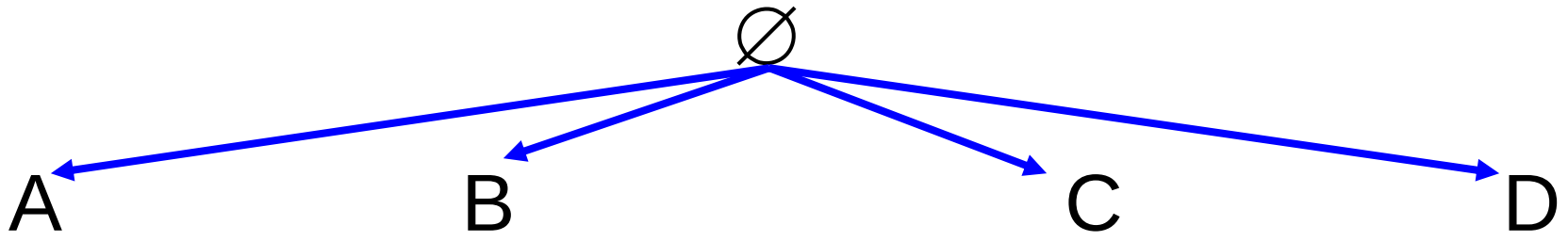
- First association mining algorithm
  - Iterative in nature
  - Level-wise search
  - Based on **Apriori Property**
    - All nonempty subsets of a frequent itemset must also be frequent. AKA, anti-monotonicity property
    - Note, in practice
      - If *support(milk)* < minsup,

        Then *support(milk* $\cup$ *Other)* < minsup

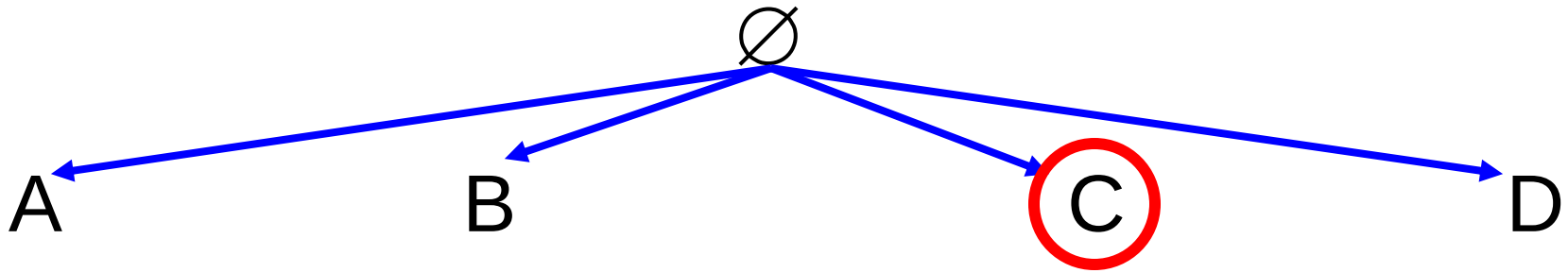
Apriori – Find Frequent Itemsets

- Basic Approach
  - Set K = 1
  - Create CandidateItems → All K-itemsets.
  - Do
    - Find Support for all K-level itemset
    - Store all K-level items that are frequent in Bag
    - Create new CandidateItems
      - Generate K+1 itemset from frequent K-itemsets
      - Use Apriori Principle to Prune invalid K+1 itemsets
    - K = K +1
  - Until CandidateItems is empty

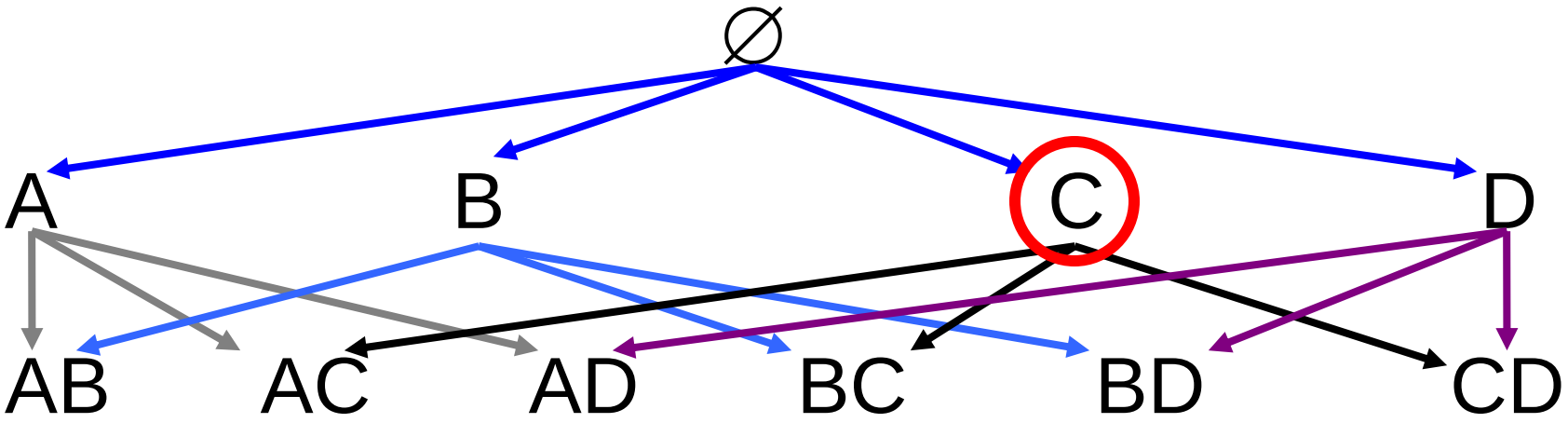
Concept Graph



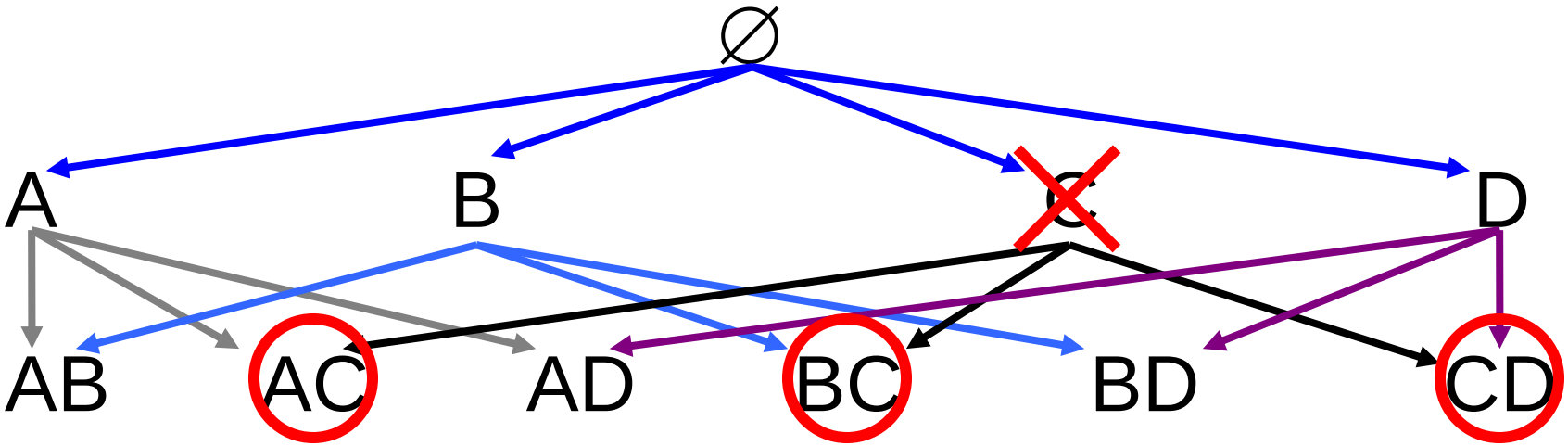
Concept Graph



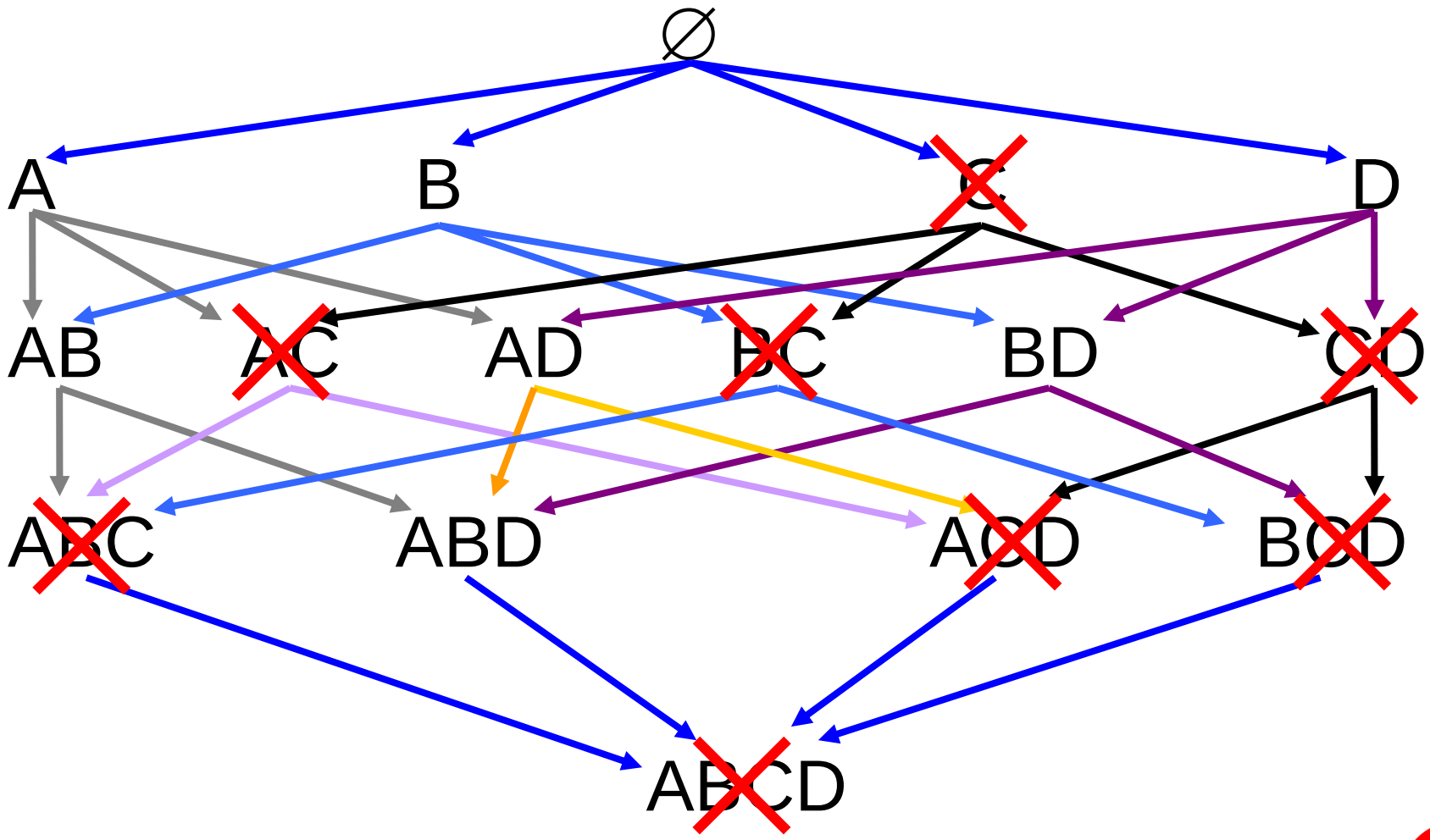
Concept Graph



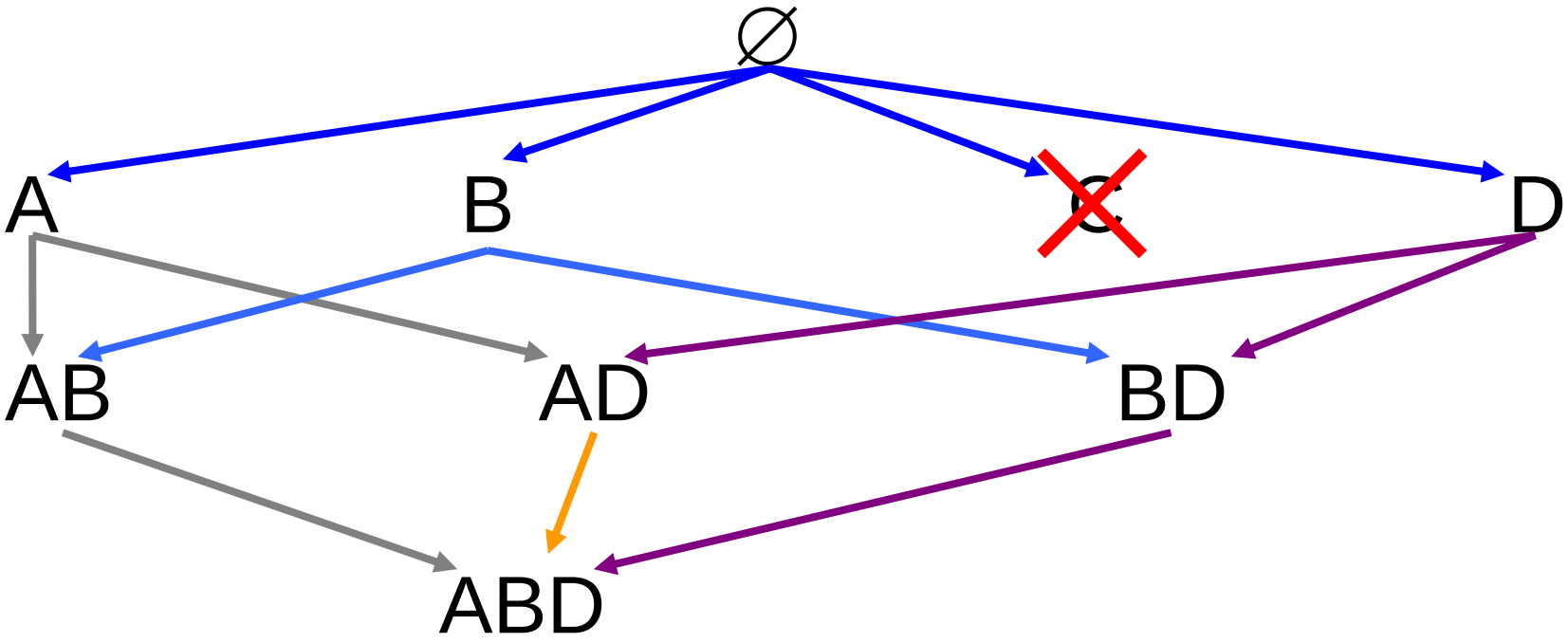
Concept Graph



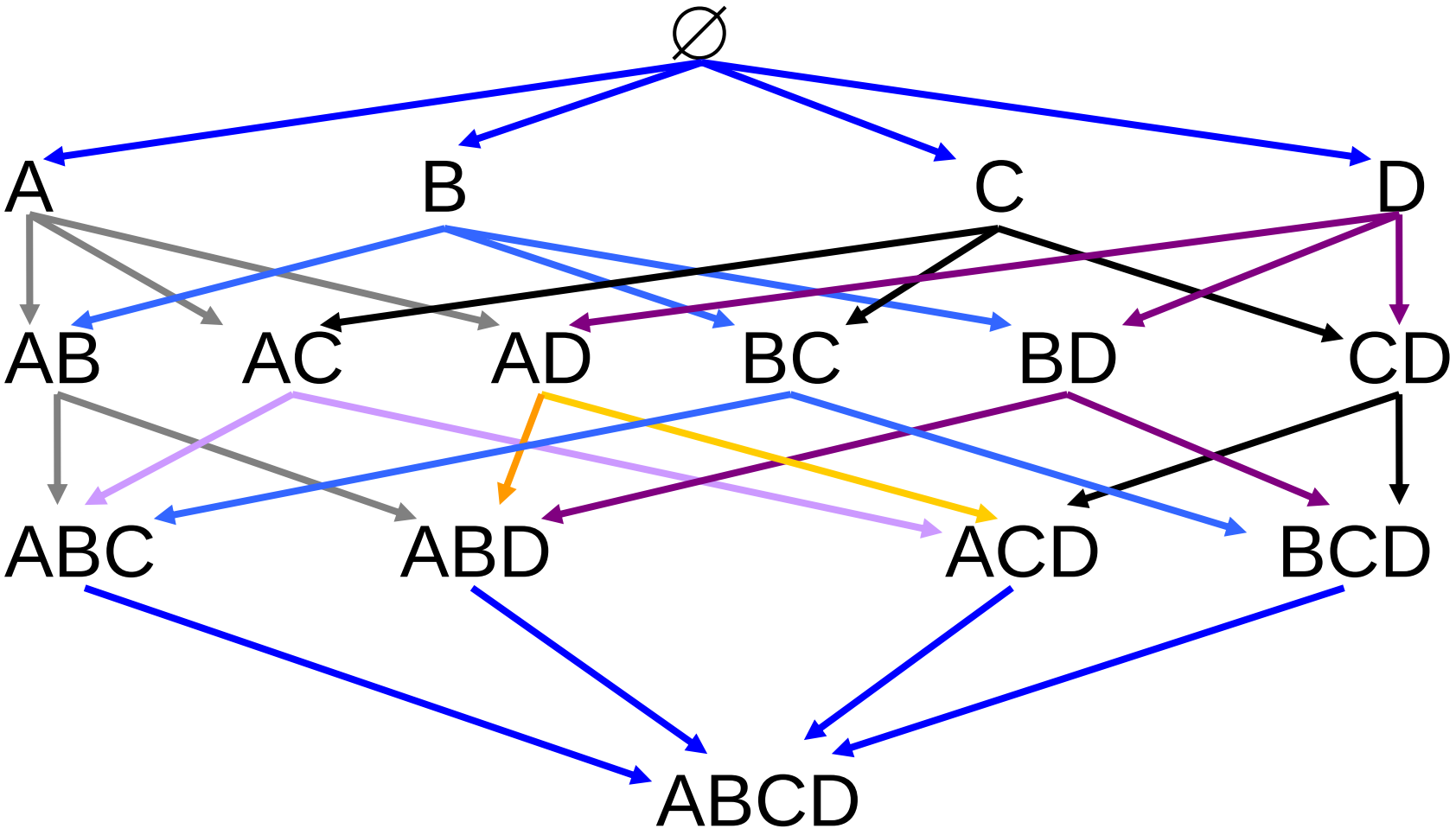
Concept Graph



Concept Graph



Concept Graph



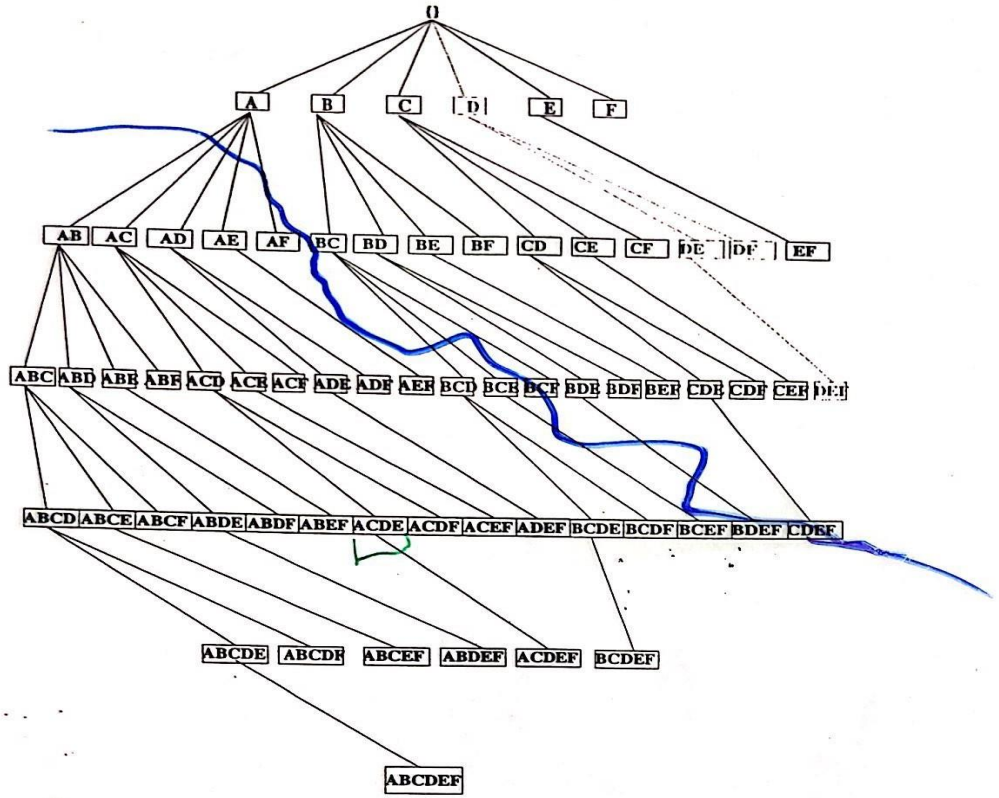


Figure 2.3: SE-tree for Itemset I = {A,B,C,D,E,F}

Algorithm: Apriori. Find frequent itemsets using an iterative level-wise approach based on candidate generation.

Input: Database, D, of transactions; minimum support threshold, *min_sup*.

Output: L, frequent itemsets in D.

Method:

```
(1)  L1 = find_frequent_1-itemsets(D);
(2)  for (k = 2; Lk−1 ≠ φ; k++) {
(3)      Ck = apriori_gen(Lk−1, min_sup);
(4)      for each transaction t ∈ D { // scan D for counts
(5)          Ct = subset(Ck, t); // get the subsets of t that are candidates
(6)          for each candidate c ∈ Ct
(7)              c.count++;
(8)      }
(9)      Lk = {c ∈ Ck|c.count ≥ min_sup}
(10) }
(11) return L = ∪k Lk;
```

```
procedure apriori_gen(L_{k-1}: frequent (k − 1)-itemsets; min_sup: minimum support threshold)
(1)   for each itemset l_1 ∈ L_{k-1}
(2)       for each itemset l_2 ∈ L_{k-1}
(3)           if (l_1[1] = l_2[1]) ∧ (l_1[2] = l_2[2]) ∧ ... ∧ (l_1[k − 2] = l_2[k − 2]) ∧ (l_1[k − 1] < l_2[k − 1]) then {
(4)               c = l_1 ⋈ l_2; // join step: generate candidates
(5)               if has_infrequent_subset(c, L_{k-1}) then
(6)                   delete c; // prune step: remove unfruitful candidate
(7)               else add c to C_k;
(8)           }
(9)   return C_k;

procedure has_infrequent_subset(c: candidate k-itemset; L_{k-1}: frequent (k − 1)-itemsets);
              // use prior knowledge
(1)   for each (k − 1)-subset s of c
(2)       if s ∉ L_{k-1} then
(3)           return TRUE;
(4)   return FALSE;
```

Figure 6.5 The Apriori algorithm for discovering frequent itemsets for mining Boolean association rules.

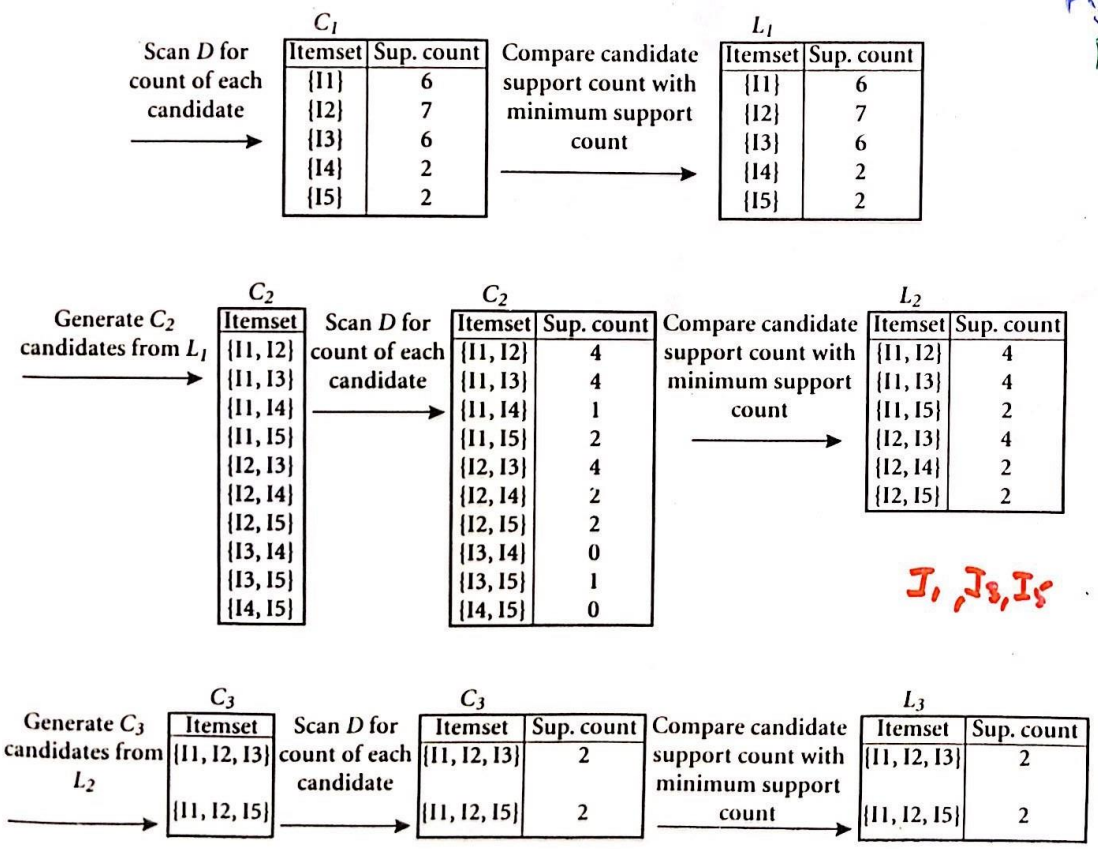
TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Table 5.1

Figure 6.2 Transactional data for an *AllElectronics* branch.

1. **The join step:** To find L_k, a set of **candidate** k-itemsets is generated by joining L_{k-1} with itself. This set of candidates is denoted C_k. Let l_1 and l_2 be itemsets in L_{k-1}. The notation $l_i[j]$ refers to the jth item in l_i (e.g., $l_1[k-2]$ refers to the second to the last item in l_1). By convention, Apriori assumes that items within a transaction or itemset are sorted in lexicographic order. The join, $L_{k-1} \bowtie L_{k-1}$, is performed, where members of L_{k-1} are joinable if their first $(k-2)$ items are in common. That is, members l_1 and l_2 of L_{k-1} are joined if $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \ldots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1])$. The condition $l_1[k-1] < l_2[k-1]$ simply ensures that no duplicates are generated. The resulting itemset formed by joining l_1 and l_2 is $l_1[1]l_1[2] \ldots l_1[k-1]l_2[k-1]$.
2. **The prune step:** C_k is a superset of L_k, that is, its members may or may not be frequent, but all of the frequent k-itemsets are included in C_k. A scan of the database to determine the count of each candidate in C_k would result in the determination of L_k (i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to L_k). C_k, however, can be huge, and so this could involve heavy computation. To reduce the size of C_k, the Apriori property is used as follows. Any $(k-1)$-itemset that is not frequent cannot be a subset of a frequent k-itemset. Hence, if any $(k-1)$-subset of a candidate k-itemset is not in L_{k-1}, then the candidate cannot be frequent either and so can be removed from C_k. This subset testing can be done quickly by maintaining a hash tree of all frequent itemsets.

Fig 6.3
Fig 5.3



I1, I3, I5

1. Join: $C_3 = L_2 \bowtie L_2 = \{\{I1,I2\}, \{I1,I3\}, \{I1,I5\}, \{I2,I3\}, \{I2,I4\}, \{I2,I5\}\} \bowtie \{\{I1,I2\}, \{I1,I3\}, \{I1,I5\}, \{I2,I3\}, \{I2,I4\}, \{I2,I5\}\} = \{\{I1,I2,I3\}, \{I1,I2,I5\}, \{I1,I3,I5\}, \{I2,I3,I4\}, \{I2,I3,I5\}, \{I2,I4,I5\}\}$.
2. Prune using the Apriori property: All nonempty subsets of a frequent itemset must also be frequent. Do any of the candidates have a subset that is not frequent?
   - The 2-item subsets of {I1,I2,I3} are {I1,I2}, {I1,I3}, and {I2,I3}. All 2-item subsets of {I1,I2,I3} are members of L_2. Therefore, keep {I1,I2,I3} in C_3.
   - The 2-item subsets of {I1,I2,I5} are {I1,I2}, {I1,I5}, and {I2,I5}. All 2-item subsets of {I1,I2,I5} are members of L_2. Therefore, keep {I1,I2,I5} in C_3.
   - The 2-item subsets of {I1,I3,I5} are {I1,I3}, {I1,I5}, and {I3,I5}. {I3,I5} is not a member of L_2, and so it is not frequent. Therefore, remove {I1,I3,I5} from C_3.
   - The 2-item subsets of {I2,I3,I4} are {I2,I3}, {I2,I4}, and {I3,I4}. {I3,I4} is not a member of L_2, and so it is not frequent. Therefore, remove {I2,I3,I4} from C_3.
   - The 2-item subsets of {I2,I3,I5} are {I2,I3}, {I2,I5}, and {I3,I5}. {I3,I5} is not a member of L_2, and so it is not frequent. Therefore, remove {I2,I3,I5} from C_3.
   - The 2-item subsets of {I2,I4,I5} are {I2,I4}, {I2,I5}, and {I4,I5}. {I4,I5} is not a member of L_2, and so it is not frequent. Therefore, remove {I2,I4,I5} from C_3.
3. Therefore, $C_3 = \{\{I1,I2,I3\}, \{I1,I2,I5\}\}$ after pruning.

Figure 6.4 Generation of candidate 3-itemsets, C_3, from L_2 using the Apriori property.

Closed and Maximal Frequent Patterns

Based on *various extensions* to association mining: Association mining can be extended to correlation analysis, where the absence or presence of correlated items can be identified. It can also be extended to mining *maxpatterns* (i.e., maximal frequent patterns) and *frequent closed itemsets*. A **maxpattern** is a frequent pattern, p, such that any proper superpattern[5] of p is not frequent. A **frequent closed itemset** is a *frequent* closed itemset where an itemset c is *closed* if there exists no proper superset of c, c', such that every transaction containing c also contains c'. Maxpatterns and frequent closed itemsets can be used to substantially reduce the number of frequent itemsets generated in mining.

equality of support

Apriori property: All nonempty subsets of a frequent itemset must also be frequent. The Apriori property is based on the following observation. By definition, if an itemset I does not satisfy the minimum support threshold, *min_sup*, then I is not frequent, that is, $P(I) < min\_sup$. If an item A is added to the itemset I, then the resulting itemset (i.e., $I \cup A$) cannot occur more frequently than I. Therefore, $I \cup A$ is not frequent either, that is, $P(I \cup A) < min\_sup$.

I - is an itemset

anti - monotonic

closed-frequent itemsets

Ex 6.2

$\langle a_1, a_2, \ldots, a_{100} \rangle$

$\langle a_1, a_2, \ldots a_{50} \rangle$

min_sup = 1

$\langle a_1, a_2, \ldots, a_{100} \rangle$ → included as closed-Itemset with support = 1

$\langle a_1, a_2, \ldots, a_{50} \rangle$ → included as closed Itemset with support = 2

Max Patterns
or
Maximal Frequent Patterns

Itemsets

$\{I_1, I_2, I_5\}$, $\{I_1, I_2, I_3\}$,

$\{I_2, I_4\}$

Closed Frequent Itemsets

$\{I_1\}$, $\{I_2\}$, $\{I_3\}$, $\{I_1, I_2\}$, $\{I_1, I_3\}$

$\{I_2, I_3\}$ $\{I_2, I_4\}$, $\{I_1, I_2, I_3\}$, $\{I_1, I_2, I_5\}$

Apriori – Find Rules

- For each frequent itemset *Z*
  - Generate all nonempty subsets of *Z*
  - For each nonempty subset *L* of *Z*
    - Rule is L $\rightarrow$ (Z – L)
    - Calculate confidence.
    - If confidence > minconf, accept rule.

5.4

Example 6.2 Let's try an example based on the transactional data for *AllElectronics* shown in Figure 6.2. Suppose the data contain the frequent itemset $l = \{I1, I2, I5\}$. What are the association rules that can be generated from l? The nonempty subsets of l are $\{I1, I2\}$, $\{I1, I5\}$, $\{I2, I5\}$, $\{I1\}$, $\{I2\}$, and $\{I5\}$. The resulting association rules are as shown below, each listed with its confidence:

$$I1 \wedge I2 \Rightarrow I5, \quad confidence = 2/4 = 50\%$$
$$I1 \wedge I5 \Rightarrow I2, \quad confidence = 2/2 = 100\%$$
$$I2 \wedge I5 \Rightarrow I1, \quad confidence = 2/2 = 100\%$$
$$I1 \Rightarrow I2 \wedge I5, \quad confidence = 2/6 = 33\%$$
$$I2 \Rightarrow I1 \wedge I5, \quad confidence = 2/7 = 29\%$$
$$I5 \Rightarrow I1 \wedge I2, \quad confidence = 2/2 = 100\%$$

If the minimum confidence threshold is, say, 70%, then only the second, third, and last rules above are output, since these are the only ones generated that are strong.

Apriori

- Time-consuming
  - Can generate large number of candidates
  - Repeatedly scans the database

- A number of algorithms exist that speed up process
  - Normally focus on the Frequent Itemset generation step.

Related Frequent Patterns Mining Methods

Create hash table H_2 using hash function $h(x, y) = ((order\ of\ x) \times 10 + (order\ of\ y))\ mod\ 7$

H_2

bucket address	0	1	2	3	4	5	6
bucket count	2	2	4	2	2	4	4
bucket contents	{I1, I4} {I3, I5}	{I1, I5} {I1, I5}	{I2, I3} {I2, I3} {I2, I3} {I2, I3}	{I2, I4} {I2, I4}	{I2, I5} {I2, I5}	{I1, I2} {I1, I2} {I1, I2} {I1, I2}	{I1, I3} {I1, I3} {I1, I3} {I1, I3}

Figure 6.6 Hash table, H_2, for candidate 2-itemsets: This hash table was generated by scanning the transactions of Figure 6.2 while determining L_1 from C_1. If the minimum support count is, say, 3, then the itemsets in buckets 0, 1, 3, and 4 cannot be frequent and so they should not be included in C_2.

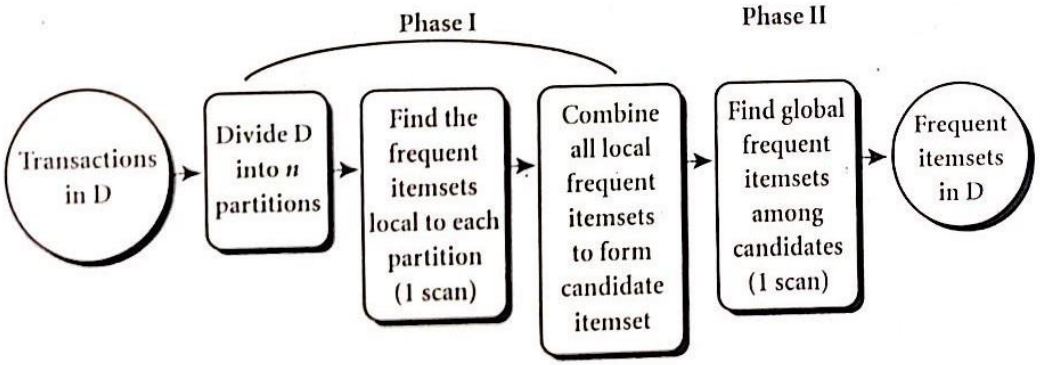


Figure 6.7 Mining by partitioning the data.

5.6

$n_1 \; n_2 \; n_3 \rightarrow n$

Correlation Analysis and Pattern Interestingness Measures

Rules

- Support-confidence framework
  - Generates a large number of rules (normally)
  - Not all rules interesting
    - Birds → Fly
- What is interesting?

Interesting - Lift

- Have rule **A → B**
- Lift(A, B)

- Result
  - Lift(A,B) > 1 : Positive Correlation (keep)
  - Lift(A,B) = 1 : Independant
  - Lift(A,B) < 1 : Negative Correlation

Example 6.7 To help filter out misleading "strong" associations of the form $A \Rightarrow B$, we need to study how the two itemsets, A and B, are correlated. Let $\overline{game}$ refer to the transactions of Example 6.6 that do not contain computer games, and $\overline{video}$ refer to those that do not contain videos. The transactions can be summarized in a contingency table. A contingency table for the data of Example 6.6 is shown in Table 6.4. From the table, we can see that the probability of purchasing a computer game is $P(\{game\}) = 0.60$, the probability of purchasing a video is $P(\{video\}) = 0.75$, and the probability of purchasing both is $P(\{game, video\}) = 0.40$. By Equation (6.22), $P(\{game, video\})/(P(\{game\}) \times P(\{video\})) = 0.40/(0.60 \times 0.75) = 0.89$. Since this value is less than 1, there is a negative correlation between the occurrence of $\{game\}$ and $\{video\}$. The numerator is the likelihood of a customer purchasing both, while the denominator is what the likelihood would have been if the two

Table 6.4 A 2 × 2 contingency table summarizing the transactions with respect to computer game and video purchases.

	game	$\overline{game}$	Σ *row*
video	4,000	3,500	7,500
$\overline{video}$	2,000	500	2,500
Σ_{col}	6,000	4,000	10,000

$$\frac{P(A,B)}{P(A)\cdot P(B)}$$

$$P(A,B) = P(B/A)\cdot P(A)$$

$$\frac{P(A,B)}{P(A)} = P(B/A)$$

$$\frac{P(A,B)}{P(A)\cdot P(B)} = \frac{P(B/A)}{P(B)} = \frac{P(A/B)}{P(A)}$$

Light — Basic Value

$$A \Rightarrow B \;:\; \frac{P(B/A)}{P(B)}$$

$$B \Rightarrow A \;:\; \frac{P(A/B)}{P(A)}$$

i) All-Confidence

$$\min \{ P(A/B), P(B/A) \}$$

ii) Kulczynski Measure

$$\frac{1}{2} \times [P(A/B) + P(B/A)]$$

iii) Cosine Measure

$$\sqrt{P(A/B) \cdot P(B/A)}$$

iv) χ^2