

# Data Mining

## Association Analysis: Basic Concepts and Algorithms

---

---

### Lecture Notes for Chapter 7



# Contents

범주형/연속형 속성 처리

# 범주형/연속형 속성

- 지금까지 **asymmetric binary variables**에 대한 연관 분석을 공부함  
→ 이제 **categorical / continuous attribute**에 적용 필요함

| Session Id | Country   | Session Length (sec) | Number of Web Pages viewed | Gender | Browser Type | Buy |
|------------|-----------|----------------------|----------------------------|--------|--------------|-----|
| 1          | USA       | 982                  | 8                          | Male   | IE           | No  |
| 2          | China     | 811                  | 10                         | Female | Netscape     | No  |
| 3          | USA       | 2125                 | 45                         | Female | Mozilla      | Yes |
| 4          | Germany   | 596                  | 4                          | Male   | IE           | Yes |
| 5          | Australia | 123                  | 9                          | Male   | Mozilla      | No  |
| ...        | ...       | ...                  | ...                        | ...    | ...          | ... |

## Example of Association Rule:

$\{\text{Number of Pages} \in [5, 10) \wedge (\text{Browser} = \text{Mozilla})\} \rightarrow \{\text{Buy} = \text{No}\}$

# 범주형/연속형 속성

- 실제 연관규칙이 적용되는 데이터베이스 레코드는 범주형 혹은 연속형 속성이 많음
- 범주형 속성(Categorical Attributes)
  - 속성의 값이 범주(category)로 나타나는 경우를 일컬음
  - 예제: 성별, 전공, 특기
- 연속형 속성(Continuous Attributes)
  - 속성의 값이 숫자로 나타나는 경우를 일컬음
  - 예제: 나이, 몸무게, 연봉

# 범주형 속성 처리

- 범주형 속성을 **Asymmetric binary** 변수로 변환함
  - Binary variable has only two states: 0 or 1
  - A binary variable is **symmetric** if both of its states are equally valuable, that is, there is no preference on which outcome should be coded as 1.
  - A binary variable is **asymmetric** if the **outcome of the states are not equally important**, such as positive or negative outcomes of a disease test.
- Introduce a new “item” for each distinct attribute-value pair
  - Example: replace Browser Type attribute with
    - ✓ **Browser Type = Internet Explorer**
    - ✓ **Browser Type = Mozilla**
    - ✓ **Browser Type = Mozilla**

# 범주형 속성 처리 예제

Table 7.1. Internet survey data with categorical attributes.

| Gender | Level of Education | State      | Computer at Home | Chat Online | Shop Online | Privacy Concerns |
|--------|--------------------|------------|------------------|-------------|-------------|------------------|
| Female | Graduate           | Illinois   | Yes              | Yes         | Yes         | Yes              |
| Male   | College            | California | No               | No          | No          | No               |
| Male   | Graduate           | Michigan   | Yes              | Yes         | Yes         | Yes              |
| Female | College            | Virginia   | No               | No          | Yes         | Yes              |
| Female | Graduate           | California | Yes              | No          | No          | Yes              |
| Male   | College            | Minnesota  | Yes              | Yes         | Yes         | Yes              |
| Male   | College            | Alaska     | Yes              | Yes         | Yes         | No               |
| Male   | High School        | Oregon     | Yes              | No          | No          | No               |
| Female | Graduate           | Texas      | No               | Yes         | No          | No               |
| ...    | ...                | ...        | ...              | ...         | ...         | ...              |

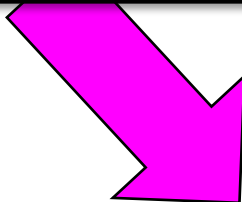


Table 7.2. Internet survey data after binarizing categorical and symmetric binary attributes.

| Male | Female | Education = Graduate | Education = College | ... | Privacy = Yes | Privacy = No |
|------|--------|----------------------|---------------------|-----|---------------|--------------|
| 0    | 1      | 1                    | 0                   | ... | 1             | 0            |
| 1    | 0      | 0                    | 1                   | ... | 0             | 1            |
| 1    | 0      | 1                    | 0                   | ... | 1             | 0            |
| 0    | 1      | 0                    | 1                   | ... | 1             | 0            |
| 0    | 1      | 1                    | 0                   | ... | 1             | 0            |
| 1    | 0      | 0                    | 1                   | ... | 1             | 0            |
| 1    | 0      | 0                    | 1                   | ... | 0             | 1            |
| 1    | 0      | 0                    | 1                   | ... | 0             | 1            |
| 1    | 0      | 0                    | 0                   | ... | 0             | 1            |
| 0    | 1      | 1                    | 0                   | ... | 0             | 1            |
| ...  | ...    | ...                  | ...                 | ... | ...           | ...          |

# 범주형 속성 처리

- 주요 이슈

- 만일, 범주형 속성이 매우 많은 값을 가진다면?
  - ◆ **Example:** attribute country has more than 200 possible values
  - ◆ 많은 속성값은 매우 적은 support 값을 가질 수 있음
    - 해결책: Aggregate the low-support attribute values
- 만약 속성값 분포가 한쪽으로 심하게 편향되었다면( highly skewed)?
  - ◆ **Example:** 95% of the visitors have “Buy = No.”
  - ◆ Most of the items will be associated with (Buy=No) item
    - 해결책: 매우 빈발한 항목은 drop 함. → 즉, 매우 많은 빈도수를 가지는 항목은 새로운 정보를 가지지 않기 때문

# 연속형 속성 처리

- 데이터에는 당연히 연속형 속성도 있음
- 예:
  - $\text{Age} \in [21, 35) \wedge \text{Salary} \in [70\text{k}, 120\text{k}) \rightarrow \text{Buy}$
  - $\text{Salary} \in [70\text{k}, 120\text{k}) \wedge \text{Buy} \rightarrow \text{Age}: \mu=28, \sigma=4$
- 연속형 속성을 처리하는 방법
  - 이산화 기반(Discretization-based) 방법
  - 통계 기반(Statistics-based) 방법
  - Non-discretization 기법
    - Min-Apriori 기법



# 연속형 속성 처리 예제

**Table 7.3.** Internet survey data with continuous attributes.

| Gender | ... | Age | Annual Income | No. of Hours Spent Online per Week | No. of Email Accounts | Privacy Concern |
|--------|-----|-----|---------------|------------------------------------|-----------------------|-----------------|
| Female | ... | 26  | 90K           | 20                                 | 4                     | Yes             |
| Male   | ... | 51  | 135K          | 10                                 | 2                     | No              |
| Male   | ... | 29  | 80K           | 10                                 | 3                     | Yes             |
| Female | ... | 45  | 120K          | 15                                 | 3                     | Yes             |
| Female | ... | 31  | 95K           | 20                                 | 5                     | Yes             |
| Male   | ... | 25  | 55K           | 25                                 | 5                     | Yes             |
| Male   | ... | 37  | 100K          | 10                                 | 1                     | No              |
| Male   | ... | 41  | 65K           | 8                                  | 2                     | No              |
| Female | ... | 26  | 85K           | 12                                 | 1                     | No              |
| ...    | ... | ... | ...           | ...                                | ...                   | ...             |

**Table 7.4.** Internet survey data after binarizing categorical and continuous attributes.

| Male | Female | ... | Age < 13 | Age ∈ [13, 21) | Age ∈ [21, 30) | ... | Privacy = Yes | Privacy = No |
|------|--------|-----|----------|----------------|----------------|-----|---------------|--------------|
| 0    | 1      | ... | 0        | 0              | 1              | ... | 1             | 0            |
| 1    | 0      | ... | 0        | 0              | 0              | ... | 0             | 1            |
| 1    | 0      | ... | 0        | 0              | 1              | ... | 1             | 0            |
| 0    | 1      | ... | 0        | 0              | 0              | ... | 1             | 0            |
| 0    | 1      | ... | 0        | 0              | 0              | ... | 1             | 0            |
| 1    | 0      | ... | 0        | 0              | 1              | ... | 1             | 0            |
| 1    | 0      | ... | 0        | 0              | 0              | ... | 0             | 1            |
| 1    | 0      | ... | 0        | 0              | 0              | ... | 0             | 1            |
| 0    | 1      | ... | 0        | 0              | 1              | ... | 0             | 1            |
| ...  | ...    | ... | ...      | ...            | ...            | ... | ...           | ...          |

# 이산화(Discretization) 기반 방법

- 이산화 기반 방법에는 Unsupervised 방법과 supervised 방법 존재함
- 비감독(unsupervised) 방법
  - Equal-width binning
  - Equal-depth binning
  - Clustering
- 감독(supervised) 방법

Attribute values,  $v$

| Class     | $V_1$ | $V_2$ | $V_3$ | $V_4$ | $V_5$ | $V_6$ | $V_7$ | $V_8$ | $V_9$ |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Anomalous | 0     | 0     | 20    | 10    | 20    | 0     | 0     | 0     | 0     |
| Normal    | 150   | 100   | 0     | 0     | 0     | 100   | 100   | 150   | 100   |

bin1                      bin2                      bin3

# 이산화(Discretization) 기반 방법

- Size of the discretized intervals affect support & confidence

{Refund = No, (Income = \$51,250)} → {Cheat = No}

{Refund = No, (60K ≤ Income ≤ 80K)} → {Cheat = No}

{Refund = No, (0K ≤ Income ≤ 1B)} → {Cheat = No}

- If intervals too small
  - ◆ may not have enough support
- If intervals too large
  - ◆ may not have enough confidence
- Potential solution: use all possible intervals

# 이산화(Discretization) 기반 방법

- Execution time

- 만약 범위가  $k$  구간으로 나뉘진다면  $k(k-1)/2$ 의 이진 항목들이 모든 가능한 구간을 나타내기 위해 생성되어야 함 → 많은 비용 소비됨

- Too many rules

{Refund = No, (Income = \$51,250)} → {Cheat = No}

{Refund = No, (51K ≤ Income ≤ 52K)} → {Cheat = No}

{Refund = No, (50K ≤ Income ≤ 60K)} → {Cheat = No}

# 통계 기반 방법, Min-Apriori 기법

- 통계 기반 방법

- 정량적 연관 규칙은 모집단의 통계적 특성을 추론하는데 사용가능
- 연관규칙의 결론부가 통계적 속성(평균, 표준편차 등)을 갖는 연속형 속성으로 나타남
- 예제:  $\text{Salary} \in [70k, 120k) \wedge \text{Buy} \rightarrow \text{Age}: \mu=28, \sigma=4$

- Min-Apriori 기법 (비이산화 방법)

- 연속형 속성들 중에서 연관성을 찾는 방법  $\rightarrow$  단어들 사이의 연관성 등

# 통계 기반 방법

- 사례:

Browser=Mozilla  $\wedge$  Buy=Yes  $\rightarrow$  Age:  $\mu=23$

- 연관규칙의 결론부는 통계적 속성을 가짐

- mean, median, standard deviation, etc.

- 규칙 생성 방법:

- 먼저, target variable를 정한 후, 이를 나머지 데이터와 별도로 생각함

- 분리된 "나머지 데이터"에 대해서 frequent itemset을 찾음

- "나머지 데이터"로부터 찾은 frequent itemset을 사용하여, "target variable"에 대한 통계적 속성을 찾음

- {Annual Income > \$100K, Shop Online = Yes}  $\rightarrow$  Age: Mean = 38

- 이 규칙은 인터넷 사용자의 연수입이 10만 달러보다 많고 정기적으로 온라인 쇼핑을 하는 사람의 평균 나이는 38세라는 것을 말함

- 해당 연관 규칙의 정당성을 확인하기 위해 통계 테스트 수행(가설 검증 등)



# 통계 기반 방법

- 연관 규칙 유용성 테스트

- 연관 규칙에 포함되는 트랜잭션으로부터 계산된 통계량이 해당 규칙에 포함되지 않은 트랜잭션으로부터 계산된 것과 다른 경우에만 해당 연관 규칙은 유용함
- Compare the statistics for segment of population covered by the rule vs segment of population not covered by the rule:

$$A \rightarrow B: \mu \quad \text{vs.} \quad \bar{A} \rightarrow B: \mu'$$

- 통계적 가설 검증(Statistical hypothesis testing):

- ◆ 귀무가설(Null hypothesis):  $H_0: \mu' = \mu + \Delta$

- ◆ 대립가설(Alternative hypothesis):  $H_1: \mu' > \mu + \Delta$

- ◆ Z has zero mean and variance 1 under null hypothesis

- ◆ A는 frequent itemset, B는 연속형 target 속성

- ◆ n1은 A를 지지하는 트랜잭션 수, n2는 A를 지지하지 않는 트랜잭션 수

- ◆ s1은 A를 지지하는 트랜잭션 중에서 B에 대한 표준편차, s2는 A를 지지하지 않는 트랜잭션 중에서 B에 대한 표준편차

$$Z = \frac{\mu' - \mu - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

# 통계 기반 방법

- 통계적 가설 검증(계속)

$$A \rightarrow B: \mu \quad \text{vs.} \quad \bar{A} \rightarrow B: \mu'$$

- $\mu$ 와  $\mu'$ 의 차이가 어떤 사용자~지정된 임계값  $\Delta$ 보다 더 큰가?를 검사함

- ◆ 통계적 가설 검증에서 귀무가설과 대립 가설, 두개의 반대되는 명제가 주어짐

- (일반적으로 귀무가설을 기각하고 대립가설을 채택하기 위해선 귀무가설이 잘못되었다는 것을 입증함)

- ◆ 데이터로부터 수집된 증거에 근거하여 가설 검증은 위 두개의 가설중에서 어느것이 수락되어야하는지를 결정함

- ◆ 이 경우,  $\mu < \mu'$ 을 가정하면,

- ◆ 귀무가설  $H_0: \mu' = \mu + \Delta$ 이며, 대립가설  $H_1: \mu' > \mu + \Delta$  임.

- ◆ 여기서 어느 가설이 수락되어야 하는지를 결정하기 위해 아래의 Z 통계량을 계산함

$$Z = \frac{\mu' - \mu - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

- Z는 평균 0과 분산 1을 갖는 표준 정규 분포
- 계산된 Z의 값은 기각값(critical value)  $Z_\alpha$ 와 대조/비교함. 기각값은 신뢰도 수준에 따라 결정됨
- 만약  $Z > Z_\alpha$ 이면 귀무가설 기각됨  $\rightarrow$  위의 정량적 연관 규칙은 유용하다고 결론 내림
- 아니면, 평균에서 차이가 통계적으로 유의미하다는 것을 보여주기 위한 충분한 증거가 데이터에 없음을 의미함



# 통계 기반 방법

- 사례:

연관규칙: Browser=Mozilla  $\wedge$  Buy=Yes  $\rightarrow$  Age:  $\mu=23$

- Rule is interesting **if difference between  $\mu$  and  $\mu'$  is greater than 5 years (i.e.,  $\Delta = 5$ )**
- For  $r$ , suppose  $n_1 = 50$ ,  $s_1 = 3.5$
- For  $r'$  (complement):  $n_2 = 250$ ,  $s_2 = 6.5$

$$Z = \frac{\mu' - \mu - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} = \frac{30 - 23 - 5}{\sqrt{\frac{3.5^2}{50} + \frac{6.5^2}{250}}} = 3.11$$

- For 1-sided test at 95% confidence level, **critical Z-value for rejecting null hypothesis is 1.64.**
- **Since Z is greater than 1.64, r is an interesting rule**

# Min-Apriori (Han et al)

- Data set이 연속형 속성(continuous attribute)을 가지는 경우에 적용 가능
- 특히 연속형 속성들간의 연관성을 찾고자 하는 경우 사용
- 사례:
  - 텍스트 문서에서 단어 연관성을 찾는 문제
  - 문서-단어 행렬(Document-term matrix)에서 각 엔트리는 주어진 문서에서 나타나는 단어의 정규화 빈도수 카운트 값을 의미함

| TID | W1 | W2 | W3 | W4 | W5 |
|-----|----|----|----|----|----|
| D1  | 2  | 2  | 0  | 0  | 1  |
| D2  | 0  | 0  | 1  | 2  | 2  |
| D3  | 2  | 3  | 0  | 0  | 0  |
| D4  | 0  | 0  | 1  | 0  | 1  |
| D5  | 1  | 1  | 1  | 0  | 2  |

**W1 and W2 tends to appear together in the same document**

# Min-Apriori

- 아래의 “문서-단어 행렬”을 다음과같이 변경함

| TID | W1 | W2 | W3 | W4 | W5 |
|-----|----|----|----|----|----|
| D1  | 2  | 2  | 0  | 0  | 1  |
| D2  | 0  | 0  | 1  | 2  | 2  |
| D3  | 2  | 3  | 0  | 0  | 0  |
| D4  | 0  | 0  | 1  | 0  | 1  |
| D5  | 1  | 1  | 1  | 0  | 2  |

- 위 “문서-단어 행렬”을 0/1 matrix로 변형한 후, 기존의 알고리즘 적용
  - ◆ 기존의 알고리즘은 binary variable에 적용되는 것이었음
- 단어간 연관성을 찾고자 함

# Min-Apriori

- 단어간 연관성 찾을 수 있음
  - If we simply sum up its frequency, support count will be greater than total number of documents!
    - ◆ Normalize the word vectors – e.g., using  $L_1$  norm
    - ◆ Each word has a support equals to 1.0

| TID | W1 | W2 | W3 | W4 | W5 |
|-----|----|----|----|----|----|
| D1  | 2  | 2  | 0  | 0  | 1  |
| D2  | 0  | 0  | 1  | 2  | 2  |
| D3  | 2  | 3  | 0  | 0  | 0  |
| D4  | 0  | 0  | 1  | 0  | 1  |
| D5  | 1  | 1  | 1  | 0  | 2  |

Normalize



| TID | W1   | W2   | W3   | W4   | W5   |
|-----|------|------|------|------|------|
| D1  | 0.40 | 0.33 | 0.00 | 0.00 | 0.17 |
| D2  | 0.00 | 0.00 | 0.33 | 1.00 | 0.33 |
| D3  | 0.40 | 0.50 | 0.00 | 0.00 | 0.00 |
| D4  | 0.00 | 0.00 | 0.33 | 0.00 | 0.17 |
| D5  | 0.20 | 0.17 | 0.33 | 0.00 | 0.33 |

각 word 갯수로 normalize 함

# Min-Apriori

- New definition of support:

$$\text{sup}(C) = \sum_{i \in T} \min_{j \in C} D(i, j)$$

| TID | W1   | W2   | W3   | W4   | W5   |
|-----|------|------|------|------|------|
| D1  | 0.40 | 0.33 | 0.00 | 0.00 | 0.17 |
| D2  | 0.00 | 0.00 | 0.33 | 1.00 | 0.33 |
| D3  | 0.40 | 0.50 | 0.00 | 0.00 | 0.00 |
| D4  | 0.00 | 0.00 | 0.33 | 0.00 | 0.17 |
| D5  | 0.20 | 0.17 | 0.33 | 0.00 | 0.33 |

Example:

Sup(W1, W2, W3)

$$= \min_{D1} (0.40, 0.33, 0.00) + \min_{D2} (0.00, 0.00, 0.33) + \dots + \min_{D5} (0.20, 0.17, 0.33)$$

$$= 0 + 0 + 0 + 0 + 0.17$$

$$= 0.17$$

# Min-Apriori

- Min-Apriori에서의 정의된 지지도 척도는 문서에서 단어 연관성을 찾는 데 적합한 아래의 특성 가짐
  - 한 단어의 정규화 지지도가 증가함에 따라 지지도는 단조형 증가
  - 그 단어를 포함하는 문서 개수가 증가함에 따라 지지도 단조형 증가
  - 지지도는 비단조형 특성 가짐
    - ◆ 예를 들어 itemset  $\{A,B\}$ 와  $\{A,B,C\}$ 가 있다면,  $\min(\{A,B\}) \geq \min(\{A,B,C\})$ 이므로  $s(\{A,B\}) \geq s(\{A,B,C\})$  임(itemset에 속한 단어 수가 증가하면 최소값을 찾기때문에 지지도는 감소할 수 밖에 없음)

# Anti-monotone property of Support

| TID | W1   | W2   | W3   | W4   | W5   |
|-----|------|------|------|------|------|
| D1  | 0.40 | 0.33 | 0.00 | 0.00 | 0.17 |
| D2  | 0.00 | 0.00 | 0.33 | 1.00 | 0.33 |
| D3  | 0.40 | 0.50 | 0.00 | 0.00 | 0.00 |
| D4  | 0.00 | 0.00 | 0.33 | 0.00 | 0.17 |
| D5  | 0.20 | 0.17 | 0.33 | 0.00 | 0.33 |

**Example:**

$$\text{Sup}(W1) = 0.4 + 0 + 0.4 + 0 + 0.2 = 1$$

$$\text{Sup}(W1, W2) = 0.33 + 0 + 0.4 + 0 + 0.17 = 0.9$$

$$\text{Sup}(W1, W2, W3) = 0 + 0 + 0 + 0 + 0.17 = 0.17$$

Itemset에 속한 단어 증가시 support  
값 줄어드는 특성 확인

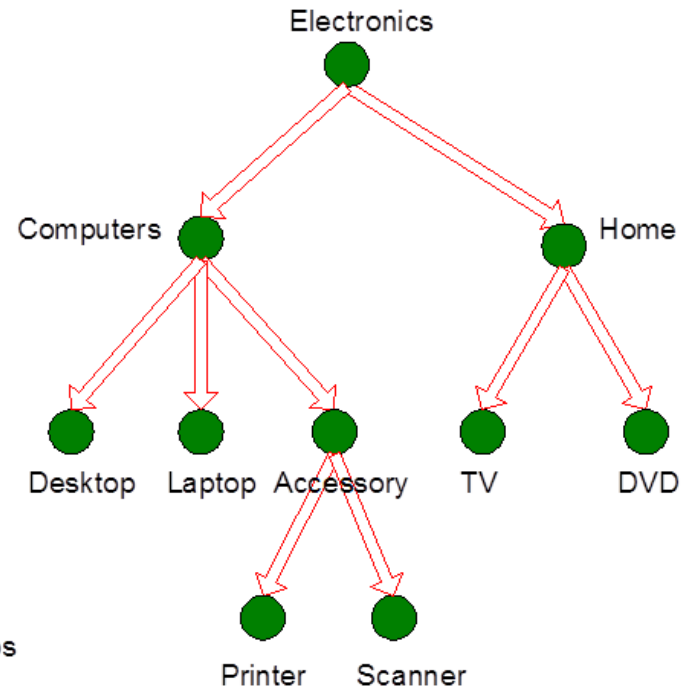
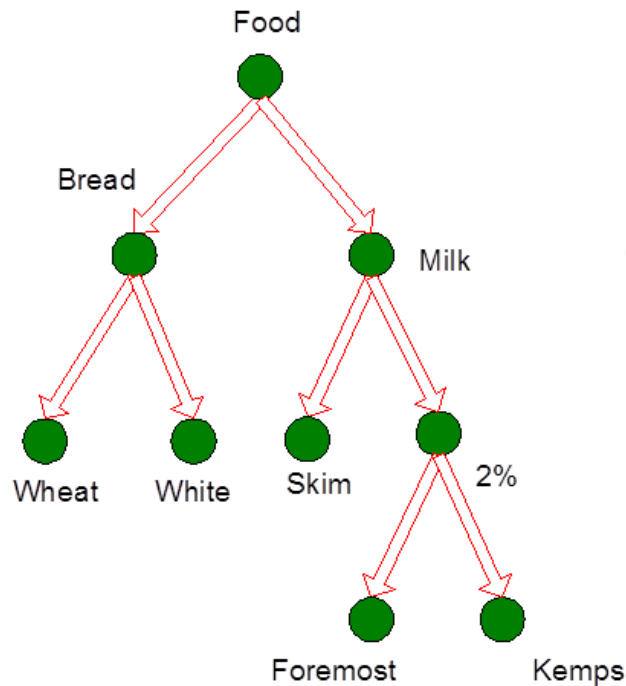
# Contents

## 다단계 연관 규칙



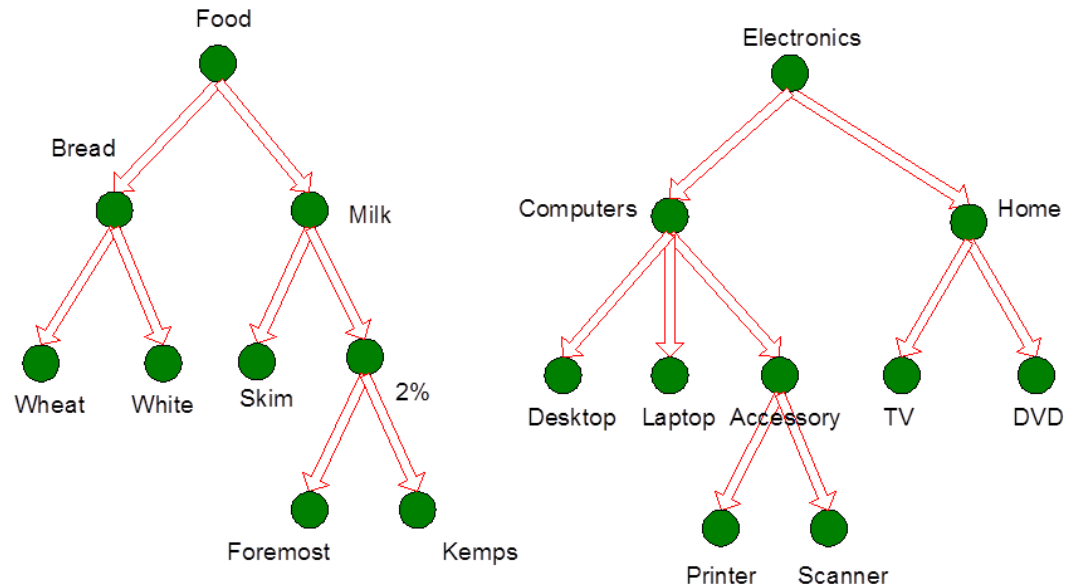
# 개념 계층(Concept Hierarchy)

- 개념 계층: 특정한 영역에서 정의된 여러 개체들 또는 개념들의 다중 계층 조직임.
  - Concept hierarchy는 specific한 데이터가 상위레벨에서 추상화되므로 분류에서 의미가 있을 수 있음
  - 주로, “is-a” 관계를 구성하는 taxonomy 형식을 가짐
  - Milk는 food의 한종류. DVD는 electronics의 한 종류 등
  - 개념 계층은 directed acyclic graph로 표현됨



# 개념 계층(Concept Hierarchy)

- 개념 계층을 연관 분석에 사용하는 경우의 주된 특성(장단점)
  - 계층의 더 낮은 레벨에 있는 항목들은 어떤 frequent itemset을 나타내기 위해 충분한 support를 가지지 않을 수도 있다. → 개념계층을 사용하지 않으면 이들의 유용한 패턴을 놓칠 수 있음
  - 그런데, 개념계층의 상위레벨 규칙은 유용하지 않을 수도 있음. → 예를 들어, (electronics → food)는 고객의 실 구매 성향에 대한 정보를 주지 않음



# 다단계 연관 규칙 마이닝(1/4)

- 개념 계층을 연관 분석에 사용하는 경우의 주된 특성(장단점)
  - 더 높은 레벨에 있는 항목들은 더 낮은 레벨에 있는 항목보다 더 높은 지지도를 가지는 경향 있음 → 이에, 만약 지지도 임계값을 너무 높게 두면 단지 높은 레벨 항목들을 수반하는 패턴만 추출됨
  - 개념계층의 도입은 더 많은 항목과 더 넓은 트랜잭션을 다루는 일이 되므로 연관 분석 알고리즘의 계산시간을 늘임
  - 개념 계층의 도입에 의해 중복된 규칙들이 산출될 수 있음

# 다단계 연관 규칙 마이닝(2/4)

- 기본 성질: 개념 계층에 의해 지지도/신뢰도는 어떻게 변하나?

• If  $\sigma(X1 \cup Y1) \geq \text{minsup}$ ,

and **X is parent of X1, Y is parent of Y1**

then  $\sigma(X \cup Y1) \geq \text{minsup}$ ,  $\sigma(X1 \cup Y) \geq \text{minsup}$ ,  $\sigma(X \cup Y) \geq \text{minsup}$

• If  $\text{conf}(X1 \Rightarrow Y1) \geq \text{minconf}$

then  $\text{conf}(X1 \Rightarrow Y) \geq \text{minconf}$

$$\text{conf}(X1 \Rightarrow Y1) = \frac{\sigma(X1 \cup Y1)}{\sigma(X1)}$$

$$\text{conf}(X1 \Rightarrow Y) = \frac{\sigma(X1 \cup Y)}{\sigma(X1)}$$

# 다단계 연관 규칙 마이닝(3/4)

- 접근법 1

- 상위 계층의 항목들을 각 트랜잭션에 추가하고, 기존 알고리즘을 사용하여 연관룰을 만든다

- 예제

- Original Transaction: {skim milk, wheat bread}
- Augmented Transaction:  
{skim milk, wheat bread, milk, bread, food}

- 이슈

- 상위 계층 항목은 다른 항목들에 비해서 높은 지지도 카운트를 가짐
  - → 만일, 주어진 지지도 threshold(즉, min supp)가 낮다면 너무 많은 규칙들이 생성될 것이다.
- 즉, 트랜잭션의 차원이 높아지는 문제가 있음

# 다단계 연관 규칙 마이닝(4/4)

- 접근법 2

- 먼저 최상위 계층에서 빈발 항목집합을 생성한 후,
- 다음으로 그 아래 계층에서 빈발 항목집합을 생성
- 위 과정을 “만족할 만한 정도의 의미 있는 규칙”을 찾을 때 까지 반복한다.

- 이슈

- 많은 I/O가 필요하여 마이닝 시간이 무척 많이 걸리게 된다.
- 교차-단계의 연관 규칙을 찾지 못할 수 있다.

(예를 들어, 가정부는 상위 계층, 결론부는 하위 계층인 연관 규칙을 찾지 못할 수 있다.)

# Contents

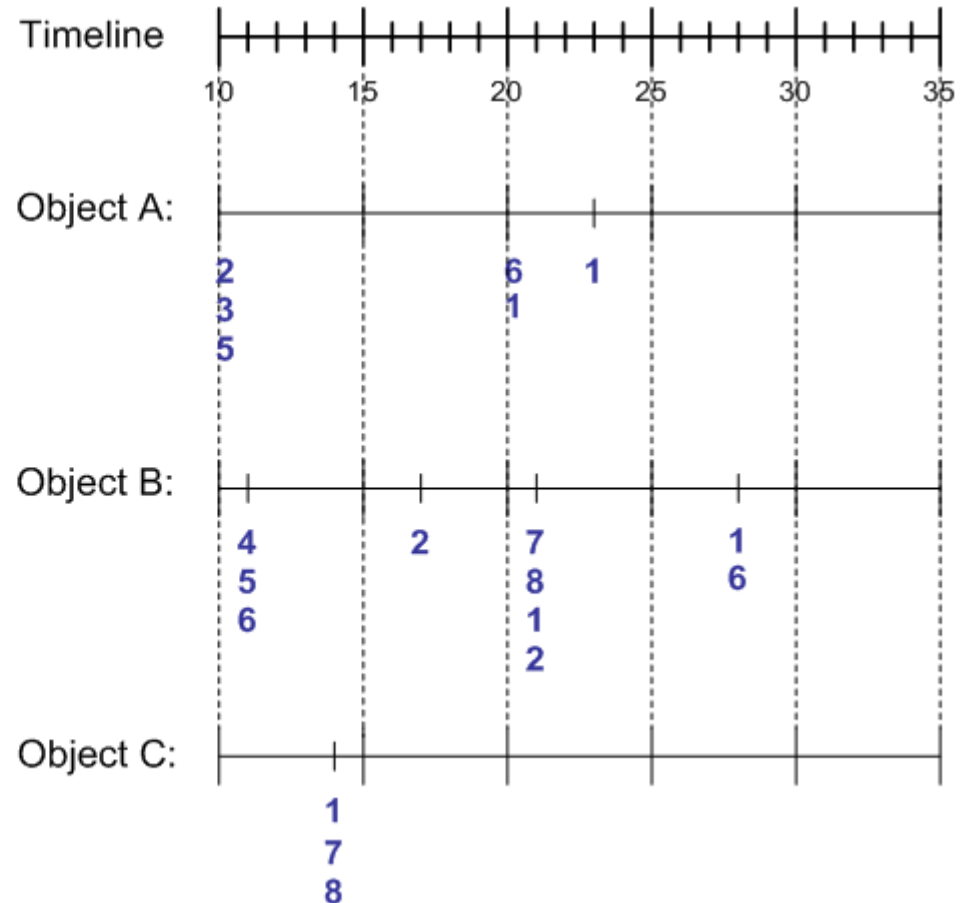
순차 패턴

# 시퀀스 데이터 (Sequence Data)

- Row에 주어진 시간에 특정 객체와 연관된 사건 발생을 기록함

Sequence Database:

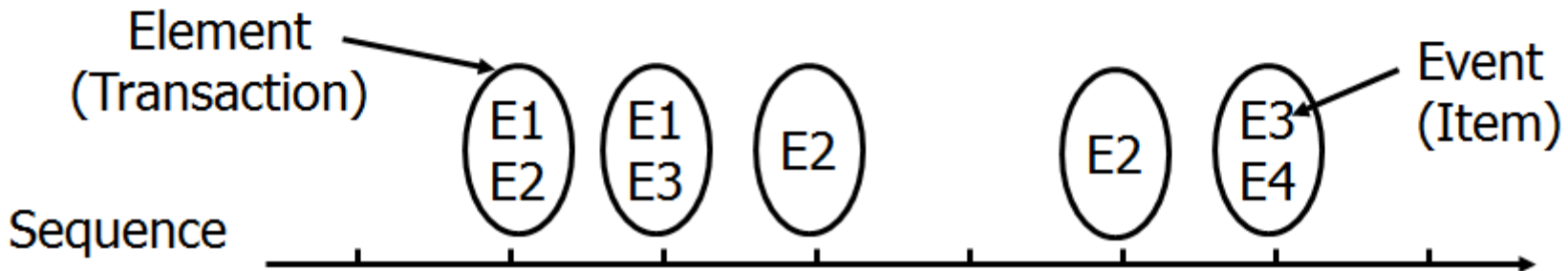
| Object | Timestamp | Events     |
|--------|-----------|------------|
| A      | 10        | 2, 3, 5    |
| A      | 20        | 6, 1       |
| A      | 23        | 1          |
| B      | 11        | 4, 5, 6    |
| B      | 17        | 2          |
| B      | 21        | 7, 8, 1, 2 |
| B      | 28        | 1, 6       |
| C      | 14        | 1, 8, 7    |





# 시퀀스 데이터 예제

| Sequence Database | Sequence                                      | Element (Transaction)  | Event (Item)                             |
|-------------------|---|--|--|
| Customer          | Purchase history of a given customer          | A set of items bought by a customer at time t                            | Books, diary products, CDs, etc          |
| Web Data          | Browsing activity of a particular Web visitor | A collection of files viewed by a Web visitor after a single mouse click | Home page, index page, contact info, etc |
| Event data        | History of events generated by a given sensor | Events triggered by a sensor at time t                                   | Types of alarms generated by sensors     |
| Genome sequences  | DNA sequence of a particular species          | An element of the DNA sequence   | Bases A,T,G,C                            |



# 시퀀스의 정의

- 시퀀스란 원소(혹은 트랜잭션)들의 순서 리스트이다.

(A sequence is an ordered list of elements (transactions))

$$s = \langle e_1 e_2 e_3 \dots \rangle$$

- 각 원소는 사건(혹은 항목)들의 모임을 포함한다

$$e_i = \{i_1, i_2, \dots, i_k\}$$

- 각 원소는 특정 시간 혹은 장소를 속성으로 가질 수 있다.

- 시퀀스의 길이  $|s|$ 는 시퀀스에 포함된 원소의 개수이다.

( $|s|$  = the number of elements of the sequence  $s$ )

- $k$ -시퀀스( $k$ -sequence)란  $k$ 개 사건(항목)을 포함하는 시퀀스이다.

(A  $k$ -sequence is a sequence that contains  $k$  events (items))

# 시퀀스의 예제

---

- 웹 시퀀스 (Web Sequence)

- < {Homepage} {Electronics} {Digital Cameras} {Canon Digital Camera}  
{Shopping Cart} {Order Confirmation} {Return to Shopping} >

- 도서관에서 대여된 책들의 순서

- <{Fellowship of the Ring} {The Two Towers}  
{Return of the King}>

# 서브시퀀스 정의와 순차 패턴

- 시퀀스 내에 포함된 시퀀스를 서브시퀀스라 부른다.
  - **Definition:** A sequence  $\langle a_1 a_2 \dots a_n \rangle$  is contained in another sequence  $\langle b_1 b_2 \dots b_m \rangle$  ( $m \geq n$ ) if there exist integers  $i_1 < i_2 < \dots < i_n$  such that  $a_1 \subseteq b_{i_1}$ ,  $a_2 \subseteq b_{i_2}$ , ...,  $a_n \subseteq b_{i_n}$

| Data sequence                             | Subsequence                     | Contain? |
|---|---------------------------------|----------|
| $\langle \{2,4\} \{3,5,6\} \{8\} \rangle$ | $\langle \{2\} \{3,5\} \rangle$ | Yes      |
| $\langle \{1,2\} \{3,4\} \rangle$         | $\langle \{1\} \{2\} \rangle$   | No       |
| $\langle \{2,4\} \{2,4\} \{2,5\} \rangle$ | $\langle \{2\} \{4\} \rangle$   | Yes      |

- 서브시퀀스  $w$ 의 지지도는  $w$ 를 포함하는 시퀀스의 비율을 나타냄  
(The support of a subsequence  $w$  is defined as the fraction of data sequences that contain  $w$ )
- **순차 패턴(sequential pattern)**은 빈발 서브시퀀스(지지도가 minsup 이상인 서브시퀀스)를 의미함 (A sequential pattern is a frequent subsequence (i.e., a subsequence whose support is  $\geq \text{minsup}$ ))

# 순차 패턴 마이닝 정의

- 다음이 주어졌을 때
  - 시퀀스들로 구성된 데이터베이스
  - 사용자가 제시한 최소 지지도  $minsup$
- 다음 작업을 수행하라.
  - 지지도가  $minsup$  이상인 모든 서브시퀀스를 찾아라.

- Given:
  - a database of sequences
  - a user-specified minimum support threshold,  $minsup$
- Task:
  - Find all subsequences with support  $\geq minsup$

# 순차 패턴 마이닝 예제

| Object | Timestamp | Events  |
|--------|-----------|---------|
| A      | 1         | 1,2,4   |
| A      | 2         | 2,3     |
| A      | 3         | 5       |
| B      | 1         | 1,2     |
| B      | 2         | 2,3,4   |
| C      | 1         | 1, 2    |
| C      | 2         | 2,3,4   |
| C      | 3         | 2,4,5   |
| D      | 1         | 2       |
| D      | 2         | 3, 4    |
| D      | 3         | 4, 5    |
| E      | 1         | 1, 3    |
| E      | 2         | 2, 4, 5 |

*Minsup* = 50%

## Examples of Frequent Subsequences:

- < {1,2} >                    s=60%
- < {2,3} >                    s=60%
- < {2,4}>                    s=80%
- < {3} {5}>                    s=80%
- < {1} {2} >                    s=80%
- < {2} {2} >                    s=60%
- < {1} {2,3} >                    s=60%
- < {2} {2,3} >                    s=60%
- < {1,2} {2,3} >                    s=60%

# 순차 패턴 마이닝 방법

- Apriori 원리를 활용

## Step 1:

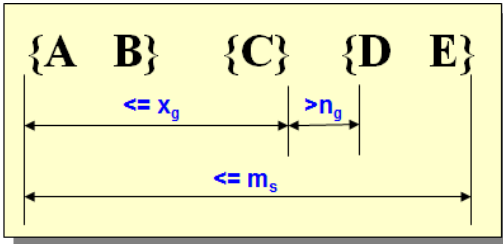
- Make the first pass over the sequence database  $D$  to yield all the 1-element frequent sequences

## Step 2:

Repeat until no new frequent sequences are found

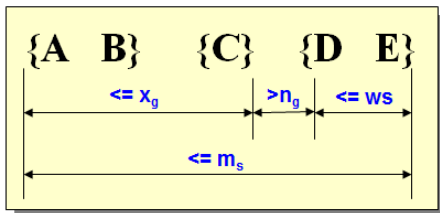
- **Candidate Generation:**
  - ◆ Merge pairs of frequent subsequences found in the  $(k-1)$ th pass to generate candidate sequences that contain  $k$  items
- **Candidate Pruning:**
  - ◆ Prune candidate  $k$ -sequences that contain infrequent  $(k-1)$ -subsequences
- **Support Counting:**
  - ◆ Make a new pass over the sequence database  $D$  to find the support for these candidate sequences
- **Candidate Elimination:**
  - ◆ Eliminate candidate  $k$ -sequences whose actual support is less than  $minsup$

# 시간 제약 요건



$x_g$ : max-gap  
 $n_g$ : min-gap  
 $m_s$ : maximum span

- **ms: Maximum span**
- The maximum allowed time difference between the earliest event and the latest event in the entire sequence.



$x_g$ : max-gap  
 $n_g$ : min-gap  
 $ws$ : window size  
 $m_s$ : maximum span

- **ng: Min-gap**
- The minimum allowed time difference between two consecutive elements in a sequence
- **xg: Maxgap**
- The maximum allowed time difference between two consecutive elements in a sequence
- **ws: Window Size**
- The maximum allowed time difference of the latest and earliest occurrences of events in any element of a sequential pattern.  
If,  $ws=0$  all events in the same element of a pattern must occur simultaneously