What Is Pattern Discovery?
Why Is It Important?

What Is Pattern Discovery?

- What are patterns?
  - Patterns: A set of items, subsequences, or substructures that occur frequently together (or strongly correlated) in a data set
  - Patterns represent intrinsic and important properties of datasets
- Pattern discovery: Uncovering patterns from massive data sets
- Motivation examples:
  - What products were often purchased together?
  - What are the subsequent purchases after buying an iPad?
  - What code segments likely contain copy-and-paste bugs?
  - What word sequences likely form phrases in this corpus?

Pattern Discovery: Why Is It Important?

- Finding inherent regularities in a data set
- Foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Mining sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: Discriminative pattern-based analysis
  - Cluster analysis: Pattern-based subspace clustering
- Broad applications
  - Market basket analysis, cross-marketing, catalog design, sale campaign analysis, Web log analysis, biological sequence analysis

Basic Concepts: Frequent Patterns and Association Rules

Basic Concepts: Frequent Itemsets (Patterns)

- Itemset: A set of one or more items
- k-itemset: $X = \{x_1, \ldots, x_k\}$
- *(absolute) support (count)* of X: Frequency or the number of occurrences of an itemset X (단순히) 갯수
- *(relative) support*, *s:* The fraction of transactions that contains X (i.e., the probability that a transaction contains X) 5개 transaction 중 'Bear'가 3개 있으면 60%.
- An itemset X is *frequent* if the support of X is no less than a *minsup* threshold (denoted as σ)

5개의 transaction.

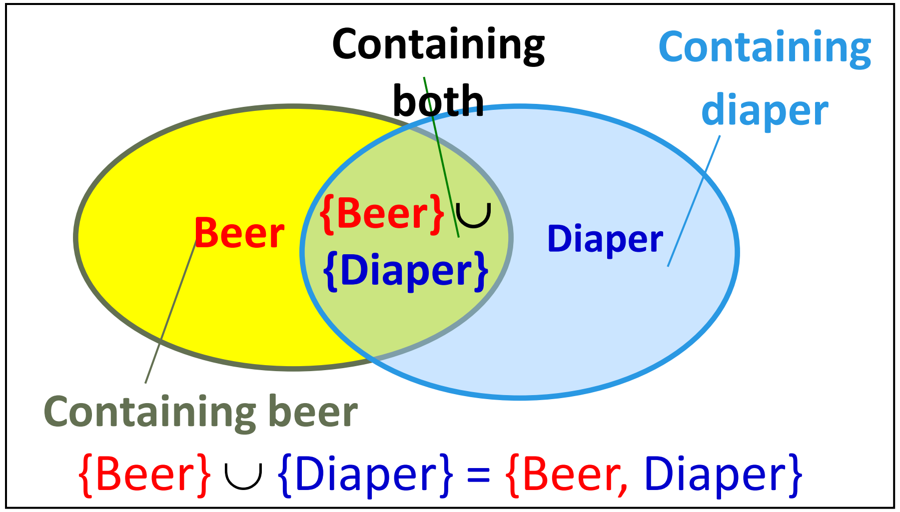
Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

- Let *minsup = 50%* 50% 이상만 찾아라.
- Freq. 1-itemsets:
  - Beer: 3 (60%); Nuts: 3 (60%)
  - Diaper: 4 (80%); Eggs: 3 (60%)
- Freq. 2-itemsets:
  - {Beer, Diaper}: 3 (60%)

X item과 Y item의 결합도(degree of coupling)을 한 눈에 알아보기 위한 방법

From Frequent Itemsets to Association Rules

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



Note: Itemset: X ∪ Y, a subtle notation!

- Association rules: *X* → *Y* (s, c)

  X를 구입하면 Y에 대한 support(확률)와 confidence(조건부확률)은 무엇인가?

  - Support, *s*: The probability that a transaction contains X ∪ Y

    두 item 사이 순서(sequence) 정보는 상관 없음.

  - Confidence, *c: The* conditional probability that a transaction containing X also contains *Y*

    순서 정보가 있음. X를 먼저 사고 난 다음!

  - c = sup(X ∪ Y) / sup(X)
- **Association rule mining**: Find all of the rules, *X* → *Y*, with minimum support and confidence
- Frequent itemsets: Let *minsup = 50%*
  - Freq. 1-itemsets: Beer: 3, Nuts: 3, Diaper: 4, Eggs: 3
  - Freq. 2-itemsets: {Beer, Diaper}: 3
- Association rules: Let *minconf = 50%*

  양방향으로 모두 표현해야 함.
  (1) X - > Y(S, C)
  (2) Y - > X(S, C)

  - *Beer* → *Diaper* (60%, 100%)
  - *Diaper* → *Beer* (60%, 75%)

(Q: Are these all rules?)

Compressed Representation: Closed Patterns and Max-Patterns

Challenge: There Are Too Many Frequent Patterns!

Long pattern은
너무 많은 frequent pattern을 가진다.

- A long pattern contains a combinatorial number of sub-patterns
- How many frequent itemsets does the following TDB_1 contain?
  - $TDB_{1:}$ T_1: $\{a_1, \ldots, a_{50}\}$; T_2: $\{a_1, \ldots, a_{100}\}$
  - Assuming (absolute) *minsup* = 1
  - Let's have a try

    1-itemsets: $\{a_1\}$: 2, $\{a_2\}$: 2, …, $\{a_{50}\}$: 2, $\{a_{51}\}$: 1, …, $\{a_{100}\}$: 1,

    2-itemsets: $\{a_1, a_2\}$: 2, …, $\{a_1, a_{50}\}$: 2, $\{a_1, a_{51}\}$: 1 …, …, $\{a_{99}, a_{100}\}$: 1,

    …, …, …, …

    99-itemsets: $\{a_1, a_2, \ldots, a_{99}\}$: 1, …, $\{a_2, a_3, \ldots, a_{100}\}$: 1

    100-itemset: $\{a_1, a_2, \ldots, a_{100}\}$: 1
  - In total: $\binom{100}{1} + \binom{100}{2} + \ldots + \binom{100}{100} = 2^{100} - 1$ sub-patterns!

A too huge set for any computer to compute or store!

Expressing Patterns in Compressed Form: Closed Patterns

- How to handle such a challenge?
- Solution 1: **Closed patterns**: A pattern (itemset) X is closed if X is *frequent*, and there exists *no super-pattern* Y ⊃ X, *with the same support* as X
  - Let Transaction DB TDB_1: T_1: $\{a_1, …, a_{50}\}$; T_2: $\{a_1, …, a_{100}\}$
  - Suppose *minsup* = 1. How many closed patterns does TDB_1 contain?
    - Two: P_1: "$\{a_1, …, a_{50}\}$: 2"; P_2: "$\{a_1, …, a_{100}\}$: 1"
- Closed pattern is a lossless compression of frequent patterns
  - Reduces the # of patterns but does not lose the support information!
  - You will still be able to say: "$\{a_2, …, a_{40}\}$: 2", "$\{a_5, a_{51}\}$: 1"

Closed pattern:
parent pattern이 없는 child pattern과 같다.

Close patterns.

* 길이에 상관없는 frequent pattern을 찾는 것이 목적이라면 closed pattern mining 기법을 사용하는 것이 좋다.

* 정보손실이 없는 압축이라는 점에서 huffman coding과 유사하다.

* 중복이 없는 core frequent pattern을 찾을 수 있다.

* ex) pattern_1{a}, pattern_2{abc} 여기서 {a}는 mining할 필요가 없다. {abc}(core)로부터 만들 수 있다.

* 진짜 필요한 알짜베기(core) frequent pattern들만 찾아주기 때문에. 컴퓨터 리소스를 절약할 수 있다.

Expressing Patterns in Compressed Form: Max-Patterns

- Solution 2: **Max-patterns**: A pattern X is a max-pattern if X is frequent and there exists no frequent super-pattern Y ↄ X
- Difference from close-patterns?
  - Do not care the real support of the sub-patterns of a max-pattern
  - Let Transaction DB TDB_1: T_1: $\{a_1, \ldots, a_{50}\}$; T_2: $\{a_1, \ldots, a_{100}\}$
  - Suppose *minsup* = 1. How many max-patterns does TDB_1 contain?
    - One: P: "$\{a_1, \ldots, a_{100}\}$: 1"
- Max-pattern is a lossy compression!
  - We only know $\{a_1, \ldots, a_{40}\}$ is frequent
  - But we do not know the real support of $\{a_1, \ldots, a_{40}\}$, …, any more!
- Thus in many applications, mining close-patterns is more desirable than mining max-patterns

Max pattern

* 주어진 시스템이나 DB에서 only the longest frequent pattern들을 찾는 것이 목적이다.

* 특정 쓰임새가 있는 경우 the longest frequent pattern들만 찾아도 유용할 것이다.

* 정보 손실이 있는 압축 방법이다.

* 모든 super frequent pattern에서, 모든 frequent pattern들을 generate하지 못하기 때문에 정보 손실이 있다.

Recommended Readings

- R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases”, in Proc. of SIGMOD'93
- R. J. Bayardo, “Efficiently mining long patterns from databases”, in Proc. of SIGMOD'98
- N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, “Discovering frequent closed itemsets for association rules”, in Proc. of ICDT'99
- J. Han, H. Cheng, D. Xin, and X. Yan, “Frequent Pattern Mining: Current Status and Future Directions”, Data Mining and Knowledge Discovery, 15(1): 55-86, 2007