Artificial Intelligence:
Search & Mining

2015 人工知能: 探索とマイニング

Introduction to Data Mining

Kevin Duh

2015-05-19

Today's Agenda

Introduction to Data Mining

Frequent Itemset Mining

Apriori Algorithm

What is Data Mining?

- Data is all around us:
  - Your photo/video collection
  - Text and multimedia from the Web
  - Credit card transactions
  - DNA sequencing database
  - Facebook social graph

What is Data Mining?

- Data is all around us:
  - Your photo/video collection
  - Text and multimedia from the Web
  - Credit card transactions
  - DNA sequencing database
  - Facebook social graph
- Data Mining = a set of methods for acquiring useful knowledge from data

Topics in Data Mining

1. Discovering Frequent Patterns
2. Cluster & Outlier Analysis
3. Classification/Prediction

Topics in Data Mining

1. Discovering Frequent Patterns
2. Cluster & Outlier Analysis
3. Classification/Prediction

Is Data Mining part of Artificial Intelligence? Depends on who you ask.

Example: Supermarket

Suppose you’re a supermarket owner, and you have data on what customers bought

Example: Supermarket

Suppose you're a supermarket owner, and you have data on what customers bought

1. Discovering Frequent Patterns:
   - What items are frequently bought together? Put them on nearby shelves.

Example: Supermarket

Suppose you're a supermarket owner, and you have data on what customers bought

1. Discovering Frequent Patterns:
   - What items are frequently bought together? Put them on nearby shelves.
2. Cluster & Outlier Analysis
   - What kinds of customer types exist?

Example: Supermarket

Suppose you're a supermarket owner, and you have data on what customers bought

1. Discovering Frequent Patterns:
   - What items are frequently bought together? Put them on nearby shelves.
2. Cluster & Outlier Analysis
   - What kinds of customer types exist?
3. Classification/Prediction
   - Given a particular customer profile, predict if ad campaign will be effective.

We'll focus on Discovering Patterns

1. Discovering Frequent Patterns
   - We'll discuss how to discover frequent and interesting patterns from various data: sets, sequences, and graphs
   - Emphasis on efficient algorithms
2. Cluster & Outlier Analysis
3. Classification/Prediction
   - See Prof. Nakamura's Big Data Analysis & Prof. Ukita's Pattern Recognition course
   - Emphasis on statistical methods

Emphasis on Efficient Algorithms

- Simple way to discover frequent patterns: Enumerate and count all possible patterns

Emphasis on Efficient Algorithms

- Simple way to discover frequent patterns: Enumerate and count all possible patterns
- But too many patterns!
- Similar to Search, we need efficient algorithms to solve the problem

Today's Agenda

Introduction to Data Mining

Frequent Itemset Mining

Apriori Algorithm

Problem Definition

- Given a finite set of **items** $\{A, B, C, \ldots\}$

Problem Definition

- Given a finite set of **items** $\{A, B, C, \ldots\}$
- in several **baskets**, e.g.
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$

Problem Definition

- Given a finite set of **items** $\{A, B, C, \ldots\}$
- in several **baskets**, e.g.
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$
- Find the **frequent itemsets**, i.e. sets of items appearing in s baskets or more

Example

- Find itemsets that appear in $s = 3$ or more baskets:
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$
- Answer:

Example

- Find itemsets that appear in $s = 3$ or more baskets:
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$
- Answer:
  - $\{A\}$: 3

Example

- Find itemsets that appear in $s = 3$ or more baskets:
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$
- Answer:
  - $\{A\}$: 3
  - $\{B\}$: 4

Example

- Find itemsets that appear in $s = 3$ or more baskets:
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$
- Answer:
  - $\{A\}$: 3
  - $\{B\}$: 4
  - $\{E\}$: 3

Example

- Find itemsets that appear in $s = 3$ or more baskets:
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$
- Answer:
  - $\{A\}$: 3
  - $\{B\}$: 4
  - $\{E\}$: 3
  - $\{A, B\}$: 3

Example

- Find itemsets that appear in $s = 3$ or more baskets:
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$
- Answer:
  - $\{A\}$: 3
  - $\{B\}$: 4
  - $\{E\}$: 3
  - $\{A, B\}$: 3
  - $\{B, E\}$: 3

Example

- Find itemsets that appear in $s = 3$ or more baskets:
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$
- Answer:
  - $\{A\}$: 3
  - $\{B\}$: 4
  - $\{E\}$: 3
  - $\{A, B\}$: 3
  - $\{B, E\}$: 3

Problem Definition (rigorous version)

- We are given several baskets, each containing several items.

Problem Definition (rigorous version)

- We are given several baskets, each containing several items.
- Let I be an itemset. The **support** of I is the number of baskets that contain I

Problem Definition (rigorous version)

- We are given several baskets, each containing several items.
- Let I be an itemset. The **support** of I is the number of baskets that contain I
- We specify a number s as threshold, and say I is a **frequent itemset** if its support is s or more.

Problem Definition (rigorous version)

- We are given several baskets, each containing several items.
- Let I be an itemset. The **support** of I is the number of baskets that contain I
- We specify a number s as threshold, and say I is a **frequent itemset** if its support is s or more.
- Goal: find **all** such frequent itemsets

Example (again)

- We are given:
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$

Example (again)

- We are given:
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$
- 1-item Itemsets & their support:
  - $\{A\}$: 3, $\{B\}$: 4, $\{C\}$: 1, $\{D\}$: 1, $\{E\}$: 3, $\{F\}$: 2

Example (again)

- We are given:
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$
- 1-item Itemsets & their support:
  - $\{A\}$: 3, $\{B\}$: 4, $\{C\}$: 1, $\{D\}$: 1, $\{E\}$: 3, $\{F\}$: 2
- 2-item Itemsets & their suppport:
  - $\{A, B\}$: 3, $\{A, C\}$: 1, $\{A, D\}$: 1, $\{A, E\}$: 2, $\{A, F\}$: 1, $\{B, C\}$: 1, $\{B, D\}$: 1, ...

Example (again)

- We are given:
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$
- 1-item Itemsets & their support:
  - $\{A\}$: 3, $\{B\}$: 4, $\{C\}$: 1, $\{D\}$: 1, $\{E\}$: 3, $\{F\}$: 2
- 2-item Itemsets & their suppport:
  - $\{A, B\}$: 3, $\{A, C\}$: 1, $\{A, D\}$: 1, $\{A, E\}$: 2, $\{A, F\}$: 1, $\{B, C\}$: 1, $\{B, D\}$: 1, ...
- 3-item Itemsets & their suppport:
  - $\{A, B, C\}$: 1, $\{A, B, D\}$: 1, $\{A, B, E\}$: 2, $\{A, B, F\}$: 1, $\{A, C, D\}$: 0, ...

Brute-force Solution

For each possible Itemset I:

Brute-force Solution

For each possible Itemset I:

1. Count the support of I

Brute-force Solution

For each possible Itemset I:

1. Count the support of I
2. If support is larger than s, report I as frequent

How many Itemsets are possible?

- If we have n items
  1. Number of 1-item Itemsets: n
  2. Number of 2-item Itemsets: $\binom{n}{2}$
  3. Number of 3-item Itemsets: $\binom{n}{3}$
  4. Number of k-item Itemsets: $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

How many Itemsets are possible?

- If we have n items
  1. Number of 1-item Itemsets: n
  2. Number of 2-item Itemsets: $\binom{n}{2}$
  3. Number of 3-item Itemsets: $\binom{n}{3}$
  4. Number of k-item Itemsets: $\binom{n}{k} = \frac{n!}{k!(n-k)!}$
- It's impossible to enumerate! e.g.
  - $\binom{10}{3} = 120$
  - $\binom{20}{3} = 1,140$
  - $\binom{40}{3} = 9,980$
  - $\binom{80}{3} = 82,160$
  - $\binom{160}{3} = 669,920$

Brute-force Solution doesn't work!

For each possible Itemset I: $\leftarrow$ TOO MANY!

1. Count the support of I
2. If support is larger than s, report I as frequent

Today's Agenda

Introduction to Data Mining

Frequent Itemset Mining

Apriori Algorithm

Monotonicity Principle

- If a set I is frequent, then every subset of I is also frequent.

Monotonicity Principle

- If a set I is frequent, then every subset of I is also frequent.
- Why?
  1. Let $J \subseteq I$. e.g. $I = \{A, B, C\}, J = \{A, C\}$
  2. Every basket that contains I must contain J. So support of $J \geq$ support of I.
  3. If I is frequent (support $\geq s$), then so is J.

Monotonicity Principle (Contrapositive version)

- If a set I is frequent, then every subset of I is also frequent.
- If I is not frequent, then no superset of I can be frequent.
  - e.g. if support($\{A, B\}$)$<\ s$, then:
  - support($\{A, B, C\}$)$<\ s$
  - support($\{A, B, D\}$)$<\ s$
  - support($\{A, B, X\}$)$<\ s$ for any X
  - support($\{A, B, X, Y\}$)$<\ s$ for any X, Y

Apriori Algorithm (main idea)

- Exploits the Monotonicity Principle.
- Don't enumerate every itemset.
- If an itemset I is not frequent, don't enumerate any superset of I.

Reference:
Rakesh Agrawal and Ramakrishnan Srikant, "Fast algorithms for mining association rules in large databases," Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pp.487-499, 1994.

Apriori Algorithm (example run)

- Find frequent itemsets ($s = 3$):
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$

1. First pass (enumerate all 1-item)
   - $\{A\}$: 3, $\{B\}$: 4, $\{C\}$: 1, $\{D\}$: 1, $\{E\}$: 3, $\{F\}$: 2

Apriori Algorithm (example run)

- Find frequent itemsets ($s = 3$):
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$

1. First pass (enumerate all 1-item)
   - $\{A\}$: 3, $\{B\}$: 4, $\{C\}$: 1, $\{D\}$: 1, $\{E\}$: 3, $\{F\}$: 2
2. Second pass (enumerate only 2-item sets where both items are frequent)

Apriori Algorithm (example run)

- Find frequent itemsets ($s = 3$):
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$

1. First pass (enumerate all 1-item)
   - $\{A\}$: 3, $\{B\}$: 4, $\{C\}$: 1, $\{D\}$: 1, $\{E\}$: 3, $\{F\}$: 2
2. Second pass (enumerate only 2-item sets where both items are frequent)
   - $\binom{3}{2} = 3$ vs. $\binom{6}{2} = 15$

Apriori Algorithm (example run)

- Find frequent itemsets ($s = 3$):
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$

1. First pass (enumerate all 1-item)
   - $\{A\}$: 3, $\{B\}$: 4, $\{C\}$: 1, $\{D\}$: 1, $\{E\}$: 3, $\{F\}$: 2
2. Second pass (enumerate only 2-item sets where both items are frequent)
   - $\binom{3}{2} = 3$ vs. $\binom{6}{2} = 15$
   - $\{A, B\}$: 3, $\{A, E\}$: 2, $\{B, E\}$: 3

Apriori Algorithm (example run)

- Find frequent itemsets ($s = 3$):
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$

1. First pass (1-item itemsets)
   - $\{A\}$: 3, $\{B\}$: 4, $\{C\}$: 1, $\{D\}$: 1, $\{E\}$: 3, $\{F\}$: 2
2. Second pass (2-item itemsets)
   - $\{A, B\}$: 3, $\{A, E\}$: 2, $\{B, E\}$: 3

Apriori Algorithm (example run)

- Find frequent itemsets ($s = 3$):
  - Basket 1: $\{A, B, D\}$
  - Basket 2: $\{A, B, C, E\}$
  - Basket 3: $\{B, E, F\}$
  - Basket 4: $\{A, B, E, F\}$

1. First pass (1-item itemsets)
   - $\{A\}$: 3, $\{B\}$: 4, $\{C\}$: 1, $\{D\}$: 1, $\{E\}$: 3, $\{F\}$: 2
2. Second pass (2-item itemsets)
   - $\{A, B\}$: 3, $\{A, E\}$: 2, $\{B, E\}$: 3
3. Third pass (3-item itemsets)
   - only enumerate $\{A, B, E\}$: 2
   - No more frequent itemsets, so stop.

Apriori Algorithm (general flow)

Alternate between:

- L_k: set of **truly frequent** itemsets of size k
- C_k: set of **candidate** itemsets of size k
  - constructed from L_{k-1}, avoids all possible enumerations

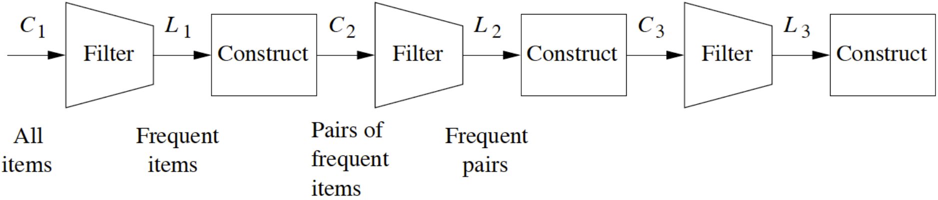

Figure from Rajamaran et. al., Mining of Massive Datasets, chapter 6

Applications of Frequent Itemset Mining

Supermarket example: What items are frequently bought together?

- cereal and milk

Applications of Frequent Itemset Mining

Supermarket example: What items are frequently bought together?

- cereal and milk
- pasta and tomato sauce and salad

Applications of Frequent Itemset Mining

Supermarket example: What items are frequently bought together?

- cereal and milk
- pasta and tomato sauce and salad
- diaper and beer?

Applications of Frequent Itemset Mining

Supermarket example: What items are frequently bought together?

- cereal and milk
- pasta and tomato sauce and salad
- diaper and beer?
  - Parents who buy diaper likely drink at home rather than outside

Summary

1. What's **Data Mining**? Methods for acquiring useful knowledge from data

Summary

1. What's **Data Mining**? Methods for acquiring useful knowledge from data
2. **Frequent Itemset Mining**: Given many baskets of items, find itemsets that appear in more than s baskets

Summary

1. What's **Data Mining**? Methods for acquiring useful knowledge from data
2. **Frequent Itemset Mining**: Given many baskets of items, find itemsets that appear in more than s baskets
3. **Monotonicity Principle**: If itemset I is not frequent, no superset of I can be.

Summary

1. What's **Data Mining**? Methods for acquiring useful knowledge from data
2. **Frequent Itemset Mining**: Given many baskets of items, find itemsets that appear in more than s baskets
3. **Monotonicity Principle**: If itemset I is not frequent, no superset of I can be.
4. **Apriori Algorithm**: construct candidates C_k from truly frequent itemsets of smaller size L_{k-1}

Next Week

Sequence Mining

- Extending Frequent Itemset Mining to Sequence data (e.g. DNA, text strings)
- Other methods that can be even more efficient than the Apriori Algorithm