COSC 290 Discrete Structures

Lecture 2: Sets

Prof. Michael Hay
Friday, Jan. 26, 2018
Colgate University

Plan for today

1. Sets
2. Frequent itemset mining

Sets

Sets

A *set* is an unordered collection of objects.

- $\mathit{Fruits} := \{\text{banana}, \text{apple}, \text{pear}\}$
- Membership: $\mathit{apple} \in \mathit{Fruits}$ is True
- Subset:
  $\{\text{banana}, \text{pear}\} \subseteq \mathit{Fruits}$
  $\{\text{banana}, \text{orange}\} \not\subseteq \mathit{Fruits}$
- Cardinality: $|\mathit{Fruits}| = 3$
- Defining a set by...
  - Enumeration:

$$\mathit{SingleDigitOdds} := \{1, 3, 5, 7, 9\}$$

  - Abstraction:

$$\mathit{SingleDigitOdds} := \{x : x \in \mathbb{Z} \text{ and } 0 \leq x \leq 10 \text{ and } x \bmod 2 = 1\}$$

or

$$\mathit{SingleDigitOdds} := \{2x + 1 : x \in \mathbb{Z} \text{ and } 0 \leq x \leq 4\}$$

Set equality

Let A and B be sets. A and B are equal, denoted $A = B$, if A and B have exactly the same elements.

A little more formally, $A = B$ if every $x \in A$ is also an element of B **and** if every $y \in B$ is also an element of A.

Poll everywhere

On a device of your choice, go to `pollev.com/cosc290`

Poll: equal sets

$$R := \{1+1, 2+2, 3+3, 4+4\}$$
$$S := \{8, 4, 8, 2, 6, 4\}$$
$$T := \{2, 4, 6, 8\}$$

Which sets are equal?

A) R and S only
B) S and T only
C) R and T only
D) R, S and T
E) None are equal

Vote here: `pollev.com/cosc290`

Poll: equal sets 2

$$R := \{2, 4, 6, 8\}$$
$$S := \{x \in \mathbb{Z}^{>0} : x \bmod 2 = 0 \text{ and } x < 10\}$$
$$T := \{2x : x \in \mathbb{Z}^{>0} \text{ and } x < 10\}$$

Which sets are equal? Choose the best answer.

A) R and S only
B) S and T only
C) R and T only
D) R, S and T
E) None are equal

Set operations

$A = \{1,3,5,7\}$ and $B = \{1,2,3,4\}$ and let universe $\mathcal{U}$ be single digit positive integers.

- Union: $A \cup B = ?$
- Intersection: $A \cap B = ?$
- Difference: $A - B = ?$
- Complement: $\sim A = ?$
  (Note: complement always defined with respect to universe $\mathcal{U}$)

Venn diagram on the board.

Poll: size of union

Let S and T be two sets with $|S| = m$ and $|T| = n$ and suppose we know that $m < n$. What is the smallest cardinality for $S \cup T$?

In other words, $|S \cup T|$ must be at least...

A) 0
B) m
C) n
D) $n + m$
E) $n \times m$

Poll: size of intersection

Let S and T be two sets with $|S| = m$ and $|T| = n$ and suppose we know that $m < n$. What is the largest cardinality for $S \cap T$?

A) 0
B) m
C) n
D) $n + m$
E) $n \times m$

Powerset

The powerset of a set S is the set of all subsets of S.

(Note: this includes the empty set because $\emptyset \subseteq S$.)

Notation: We will use $\mathcal{P}(S)$ to denote the powerset of a set S.

Example

Suppose $S \coloneqq \{1,2,3\}$, what is $\mathcal{P}(S)$?

$$\mathcal{P}(S) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$$

Frequent itemset mining

Frequent itemset mining

Data on consumer purchases: a list of n transactions $T := t_1, \ldots, t_n$ where each transaction t_i is represented as a *set* of items purchased.

Example:

$$\begin{aligned} t_1 &= \{\,\text{soy milk}, \text{coffee}\,\} \\ t_2 &= \{\,\text{milk}, \text{orange juice}, \text{cocoa puffs}\,\} \\ &\ldots \\ t_n &= \{\,\text{organic tofu}, \text{broccoli}, \text{coffee}, \text{soy milk}\,\} \end{aligned}$$

Goal: find sets of items that were frequently purchased together.

Three key ideas

1. Itemset support
2. *A priori* principle
3. Candidate generation

Frequent itemsets

Transactions $T := t_1, t_2, \ldots, t_n$ where t_i is the set of items purchased in the i^{th} transaction.

An itemset c is simply a set of items. Example: $c := \{\,\text{coffee}, \text{milk}\,\}$.

A frequent itemset is an itemset whose *support* is above some threshold (e.g., 1% of all transactions).

Recall that the support of an itemset is the number of transactions in which these items were purchased together.

The task is to (efficiently) find all frequent itemsets.

Support

The support of an itemset is the number of transactions in which these items appear together.

Let's express this using set notation,

$$\sigma(c) := |\{\, t_i : c \subseteq t_i, t_i \in T \,\}|$$

All items

Let I be the set of all items that appear in any transaction.

Example

If $T := t_1, t_2, t_3$ is as follows:

$$t_1 = \{\, \text{soy milk}, \text{coffee} \,\}$$
$$t_2 = \{\, \text{milk}, \text{orange juice}, \text{cocoa puffs} \,\}$$
$$t_3 = \{\, \text{tofu}, \text{broccoli}, \text{coffee}, \text{soy milk} \,\}$$

Then I would be

$$I := \{\, \text{soy milk}, \text{coffee}, \text{milk}, \text{orange juice}, \text{cocoa puffs}, \text{tofu}, \text{broccoli} \,\}$$

Frequent itemset mining: slow version

First attempt at algorithm:

- Try every possible combination of items from I.
- For each combo, calculate its support.
- If support is high enough (above some threshold), add it to result.
- Output result.

Frequent itemset mining: slow version

Input: A list of transactions $T := t_1, \dots, t_n$, *minsup*
Output: The set of all frequent itemsets.

```
Let I be the set of all items that appear in any transaction t_i
Result := ∅                                  ▷ Result will store all frequent itemsets
for each candidate itemset c ∈ P(I) do
    s := σ(c)                                ▷ compute support for c
    if s ≥ n × minsup then                   ▷ itemset c is frequent
        Result := Result ∪ { c }             ▷ add c to Result
return Result
```

What is inefficient about this approach?

A priori principle

The *a priori* principle: if an itemset c is frequent, then all subsets of c must also be frequent.

We can equivalently say, if any subset of c is infrequent, then itemset c *cannot* be frequent.

Example

Example: consider the itemset c

$$c := \{\text{organic tofu}, \text{meat lover's frozen pizza}, \text{broccoli}\}$$

and its subset c',

$$c' := \{\text{organic tofu}, \text{meat lover's frozen pizza}\}$$

If c' is infrequent, then c cannot be frequent.

So what?

Towards a more efficient algorithm

1. Start with single items:
   - Keep only those items that are frequent.
2. Then look at pairs:
   - *A priori* principle: only need to consider pairs of frequent items
   - Keep only pairs that are frequent
3. Then look at triples:
   - *A priori* principle: only need to consider triples such that every pair from triple is frequent
   - Keep only triples that are frequent
4. Continue with size 4 itemsets, etc.

Candidate frequent itemsets

Suppose we have already calculated F_2, the set of all frequent 2-itemsets (itemsets that contain two items).

We can use the *a priori* principle to generate candidate 3-itemsets.

An itemset c is a *candidate* if every subset of c is frequent.

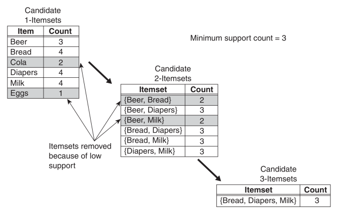
Let C_3 be the set of all *candidate* 3-itemsets.

Example

Should $c := \{\text{tofu}, \text{coffee}, \text{cocoa puffs}\}$ be in C_3?

c should be in C_3 if and only if all subsets of c are in F_2: $\{\text{tofu}, \text{coffee}\} \in F_2$ and $\{\text{tofu}, \text{cocoa puffs}\} \in F_2$ and $\{\text{coffee}, \text{cocoa puffs}\} \in F_2$.

Example



Item	Count
Beer	3
Bread	4
Cola	2
Diapers	4
Milk	4
Eggs	1

Itemset	Count
{Beer, Bread}	2
{Beer, Diapers}	3
{Beer, Milk}	2
{Bread, Diapers}	3
{Bread, Milk}	3
{Diapers, Milk}	3

Itemset	Count
{Bread, Diapers, Milk}	3

Figure 6.5. Illustration of frequent itemset generation using the *Apriori* algorithm.

Credit: Figure from *Introduction to Data Mining* by Tan, Steinbach, and Kumar, 2005.https://www-users.cs.umn.edu/~kumar001/dmbook/ch6.pdf

Frequent itemset mining: faster version

Input: A list of transactions $T := t_1, \ldots, t_n$, *minsup*
Output: The set of all frequent itemsets.

```
k = 1
Let F be set of frequent items (items that occur in at least
    n × minsup transactions)
F_k = { {i} : i ∈ F }                     ▷ put each frequent item in a set by itself
repeat
    k = k + 1
    C_k = generateCandidates(F_{k−1}, F)   ▷ using a priori principle
    F_k = ∅
    for all candidate itemset c ∈ C_k do
        s := σ(c)                          ▷ compute support for c
        if s ≥ n × minsup then             ▷ itemset c is frequent
            F_k := F_k ∪ { c }             ▷ add c to F_k
until F_k = ∅                              ▷ no size k itemsets were frequent
return F_1 ∪ F_2 ∪ ··· ∪ F_k
```


Poll: the set of all items

Suppose we have a collection of n transactions, $t_1, \ldots, t_n$. Example:

$$t_1 = \{\text{soy milk}, \text{coffee}\}$$
$$t_2 = \{\text{milk}, \text{orange juice}, \text{cocoa puffs}\}$$
$$\ldots$$
$$t_n = \{\text{organic tofu}, \text{broccoli}, \text{coffee}, \text{soy milk}\}$$

Let I represent the set of all items purchased in at least one transaction. Which of the following is a correct definition for I?

A) $I := t_1 \cup t_2 \cup \cdots \cup t_n$
B) $I := t_1 + t_2 + \cdots + t_n$
C) $I := t_1 \cap t_2 \cap \cdots \cap t_n$
D) $I := \{t_1, t_2, \ldots, t_n\}$
E) None of the above / More than one of the above

Poll: frequent itemset

Suppose we have a collection of n transactions, $t_1, \ldots, t_n$. Example:

$$t_1 = \{\text{soy milk}, \text{coffee}\}$$
$$t_2 = \{\text{milk}, \text{orange juice}, \text{cocoa puffs}\}$$
$$\ldots$$
$$t_n = \{\text{organic tofu}, \text{broccoli}, \text{coffee}, \text{soy milk}\}$$

Suppose that c is a frequent itemset. Consider this statement:

$$c \in \mathcal{P}(t_1 \cup t_2 \cup \cdots \cup t_n)$$

Choose the best answer:

A) This statement must be true.
B) This statement may be true.
C) This statement must be false.
D) This statement is not well defined.