CSCI 550: Adv. Data Mining

14- Itemset mining

MONTANA STATE UNIVERSITY

FREQUENT PATTERN MINING

Frequent pattern mining refers to the task of extracting informative and useful patterns in massive and complex datasets. Patterns comprise sets of co-occurring attribute values, called itemsets, or more complex patterns, such as sequences.

The key goal is to discover hidden relationships in the data to better understand the interactions among the data points and attributes.

Frequent Itemset Mining

- In many applications one is interested in how often two or more objects of interest co-occur, the so-called itemsets.
- The prototypical application was market basket analysis, that is, to mine the sets of items that are frequently bought together at a supermarket by analyzing the customer shopping carts (the so-called "market baskets").
- Frequent itemset mining is a basic exploratory mining task, since the basic operation is to find the co-occurrence, which gives an estimate for the joint probability mass function.
- Once we mine the frequent sets, they allow us to extract association rules among the itemsets, where we make some statement about how likely are two sets of items to co-occur or to conditionally occur.
- What are other applications of itemset mining?

Frequent Itemsets: Terminology

- **Itemsets**: Let $I = \{x_1, x_2, \ldots, x_m\}$ be a set of elements called items. A set X ⊆ *I* is called an itemset. An itemset of cardinality (or size) k is called a k-itemset. Further, we denote by *I*(k) the set of all k-itemsets, that is, subsets of I with size k.
- **Tidsets:** Let $T = \{t_1, t_2, \ldots, t_n\}$ be another set of elements called transaction identifiers or tids. A set *T* ⊆ T is called a tidset.
- **Transactions:** A transaction is a tuple of the form (t, X), where t ∈ T is a unique transaction identifier, and X is an itemset.
- **Database:** A binary database D is a binary relation on the set of tids and items, that is, D ⊆ T × I. We say that tid t ∈ T contains item x ∈ I iff (t,x) ∈ D. In other words, (t,x) ∈ D iff x ∈ X in the tuple (t,x) . We say that tid t contains itemset $X = \{x_1, x_2, \ldots, x_k\}$ iff $(t,x_i) \in D$ for all i = 1,2, . . . ,k.

Market Basket Analysis Example

Transaction ID	Items
1	Toilet paper, beans, rice, milk, baby wipes,
2	Oat milk, beans, toilet paper, orange juice
3	Oat milk, milk, orange juice, toilet paper
4	Beans, toilet paper, baby wipes, diapers
5	Toilet paper, butter, baby wipes, diapers
6	Milk, toilet paper
7	Milk, rice
8	Beans, milk, rice, toilet paper
9	Milk, butter, diapers
10	Beans, rice, toilet paper

Support and Frequent Itemsets

- The support of an itemset X in a dataset D, denoted sup(X), is the number of transactions in D that contain X :

$$sup(x) = |t(x)|$$

- The relative support of X is the fraction of transactions that contain X:

$$\boxed{rsup(X) = \frac{sup(X)}{|\boldsymbol{D}|}}$$

- It is an estimate of the *joint probability of the items comprising X* .
- An itemset X is said to be frequent in D if sup(X) ≥ minsup, where minsup is a user defined minimum support threshold.
- The set F to denotes the set of all frequent itemsets, and F(k) denotes the set of frequent k-itemsets.
- For example: which sets of items are purchased at least 30% of the time?

Frequent Itemsets

Minimum support: minsup = 3

Transaction ID	Items
1	Toilet paper (9), beans (2), rice (8), milk (5), baby wipes (1) , diapers (4)
2	Oat milk (6), beans (2), toilet paper (9), orange juice (7)
3	Oat milk (6), milk (5), orange juice (7), toilet paper (9)
4	Beans (2), toilet paper (9), baby wipes (1), diapers (4)
5	Toilet paper (9), butter (3), baby wipes (1), diapers (4)
6	Milk (5), toilet paper (9)
7	Milk (5), rice (8)
8	Beans (2), milk (5), rice (8), toilet paper (9)
9	Milk (5), butter (3) , diapers (4)
10	Beans (2), rice (8), toilet paper (9)

Frequent Itemsets

Minimum support: minsup = 3

Which sets of items are purchased at least 30% of the time? → "minsup" = 3

Transaction ID	Items
1	Toilet paper (9), beans (2), rice (8), milk (5), baby wipes (1) , diapers (4)
2	Oat milk (6), beans (2), toilet paper (9), orange juice (7)
3	Oat milk (6), milk (5), orange juice (7), toilet paper (9)
4	Beans (2), toilet paper (9), baby wipes (1), diapers (4)
5	Toilet paper (9), butter (3), baby wipes (1), diapers (4)
6	Milk (5), toilet paper (9)
7	Milk (5), rice (8)
8	Beans (2), milk (5), rice (8), toilet paper (9)
9	Milk (5), butter (3) , diapers (4)
10	Beans (2), rice (8), toilet paper (9)

Frequent itemsets of size 1: {1}, {2}, {4}, {5}, {8},{9}

Frequent itemsets of size 2: {1,4}, {1,9}, {2,8}, {2,9}, {4,9}, {5,8},{5,9},{8,9}

Frequent itemsets of size 3: {1,4,9}, {2,8,9}

Itemset Mining Algorithms: Brute Force

- The brute-force algorithm enumerates all the possible itemsets $X \subseteq I$, and for each such subset determines its support in the input dataset D. The method comprises two main steps: (1) candidate generation and (2) support computation.
- **Candidate Generation:** This step generates all the subsets of I, which are called candidates, as each itemset is potentially a candidate frequent pattern. The candidate itemset search space is clearly exponential because there are $2^{|I|}$ potentially frequent itemsets.
- **Support Computation:** This step computes the support of each candidate pattern X and determines if it is frequent. For each transaction <t, i (t)> in the database, we determine if X is a subset of i(t). If so, we increment the support of X.
- **Computational Complexity:** Support computation takes time $O(|I| \cdot |D|)$ in the worst case, and because there are $O(2^{|I|})$ possible candidates, the computational complexity of the brute-force method is $O(|I| \cdot |D| \cdot 2^{|I|})$.

Itemset Mining Algorithms: Brute Force

BruteForce ($\mathbf{D}$, $\mathcal{I}$, *minsup*):

$\mathcal{F} \leftarrow \emptyset$ `// set of frequent itemsets`

foreach $X \subseteq \mathcal{I}$ **do**

  $sup(X) \leftarrow$ ComputeSupport $(X, \mathbf{D})$

  if $sup(X) \geq minsup$ **then**

    $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, sup(X))\}$

return $\mathcal{F}$

ComputeSupport $(X, \mathbf{D})$:

$sup(X) \leftarrow 0$

foreach $\langle t, \mathbf{i}(t) \rangle \in \mathbf{D}$ **do**

  if $X \subseteq \mathbf{i}(t)$ **then**

    $sup(X) \leftarrow sup(X) + 1$

return $sup(X)$

Itemset Mining Algorithms: Apriori

- If $X \subseteq Y$, then $sup(X) \geq sup(Y)$, which leads to the following two observations:
- (1) if X is frequent, then any subset $Y \subseteq X$ is also frequent, and
- (2) if X is not frequent, then any superset $Y \supseteq X$ cannot be frequent.
- The Apriori algorithm utilizes these two properties to significantly improve the brute-force approach. It employs a level-wise or breadth-first exploration of the itemset search space, and prunes all supersets of any infrequent candidate, as no superset of an infrequent itemset can be frequent. It also avoids generating any candidate that has an infrequent subset.
- This will eliminate computing frequency of sets that have no chance of being frequent.

Itemset Mining Algorithms: Apriori

Apriori (D, $\mathcal{I}$, *minsup*)**:**

$\mathcal{F} \leftarrow \emptyset$

$\mathcal{C}^{(1)} \leftarrow \{\emptyset\}$ `// Initial prefix tree with single items`

foreach $i \in \mathcal{I}$ **do** Add i as child of $\emptyset$ in $\mathcal{C}^{(1)}$ with $sup(i) \leftarrow 0$

$k \leftarrow 1$ `// k denotes the level`

while $\mathcal{C}^{(k)} \neq \emptyset$ **do**

    ComputeSupport ($\mathcal{C}^{(k)}, \boldsymbol{D}$)

    foreach *leaf* $X \in \mathcal{C}^{(k)}$ **do**

        if $sup(X) \geq minsup$ **then** $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, sup(X))\}$

        else remove X from $\mathcal{C}^{(k)}$

    $\mathcal{C}^{(k+1)} \leftarrow$ ExtendPrefixTree ($\mathcal{C}^{(k)}$)

    $k \leftarrow k + 1$

return $\mathcal{F}^{(k)}$

Itemset Mining Algorithms: Apriori

ComputeSupport $(\mathcal{C}^{(k)}, \boldsymbol{D})$:

foreach $\langle t, \boldsymbol{i}(t)\rangle \in \boldsymbol{D}$ **do**
  foreach *k-subset* $X \subseteq \boldsymbol{i}(t)$ **do**
    if $X \in \mathcal{C}^{(k)}$ **then** $sup(X) \leftarrow sup(X) + 1$

ExtendPrefixTree $(\mathcal{C}^{(k)})$:

foreach *leaf* $X_a \in \mathcal{C}^{(k)}$ **do**
  foreach *leaf* $X_b \in \text{sibling}(X_a)$, *such that* $b > a$ **do**
    $X_{ab} \leftarrow X_a \cup X_b$
    `// prune if there are any infrequent subsets`
    if $X_j \in \mathcal{C}^{(k)}$, ***for all*** $X_j \subset X_{ab}$, *such that* $|X_j| = |X_{ab}| - 1$ **then**
      Add X_{ab} as child of X_a with $sup(X_{ab}) \leftarrow 0$
  if *no extensions from* X_a **then**
    remove X_a and its ancestors with no extensions from $\mathcal{C}^{(k)}$
return $\mathcal{C}^{(k)}$

Apriori Algorithm: Details

- Let C(k) denote the prefix tree comprising all the candidate k-itemsets. Apriori begins by inserting the single items into an initially empty prefix tree to populate C(1).
- The support for the current candidates is obtained via ComputeSupport procedure that generates k-subsets of each transaction in the database D, and for each such subset it increments the support of the corresponding candidate in C(k) if it exists. Next, we remove any infrequent candidate.
- The leaves of the prefix tree that survive comprise the set of frequent k-itemsets F(k), which are used to generate the candidate (k +1)-itemsets for the next level. The ExtendPrefixTree procedure employs prefix-based extension for candidate generation.
- Given two frequent k-itemsets Xa and Xb with a common k −1 length prefix, that is, given two sibling leaf nodes with a common parent, we generate the (k +1)-length candidate Xab = Xa ∪Xb. This candidate is retained only if it has no infrequent subset. Finally, if a k-itemset Xa has no extension, it is pruned from the prefix tree, and we recursively prune any of its ancestors with no k-itemset extension, so that in C(k) all leaves are at level k.
- If new candidates were added, the whole process is repeated for the next level. This process continues until no new candidates are added.

Itemset Mining Algorithms: Apriori
Minimum support: minsup = 3

Transaction ID	Items
1	Toilet paper, beans, rice, milk, baby wipes, diapers
2	Oat milk, beans, toilet paper, orange juice
3	Oat milk, milk, orange juice, toilet paper
4	Beans, toilet paper, baby wipes, diapers
5	Toilet paper, butter, baby wipes, diapers
6	Milk, toilet paper
7	Milk, rice
8	Beans, milk, rice, toilet paper
9	Milk, butter, diapers
10	Beans, rice, toilet paper

→

Candidate Set	Support
{Baby Wipes}	3
{Beans}	5
{Butter}	2
{Diapers}	4
{Milk}	6
{Oat Milk}	2
{Orange Juice}	2
{Rice}	4
{Toilet Paper}	8

Start with frequent item sets of size k=1

Itemset Mining Algorithms: Apriori
Minimum support: minsup = 3

Transaction ID	Items
1	Toilet paper (9), beans (2), rice (8), milk (5), baby wipes (1) , diapers (4)
2	Oat milk (6), beans (2), toilet paper (9), orange juice (7)
3	Oat milk (6), milk (5), orange juice (7), toilet paper (9)
4	Beans (2), toilet paper (9), baby wipes (1), diapers (4)
5	Toilet paper (9), butter (3), baby wipes (1), diapers (4)
6	Milk (5), toilet paper (9)
7	Milk (5), rice (8)
8	Beans (2), milk (5), rice (8), toilet paper (9)
9	Milk (5), butter (3) , diapers (4)
10	Beans (2), rice (8), toilet paper (9)

→

Candidate Set	Support
{1}	3
{2}	5
{3}	2
{4}	4
{5}	6
{6}	2
{7}	2
{8}	4
{9}	8

Assign an ID to each item

Itemset Mining Algorithms: Apriori
Minimum support: minsup = 3

Frequent item sets of size 1: {1}, {2}, {4}, {5}, {8},{9}

Transaction ID	Items
1	Toilet paper (9), beans (2), rice (8), milk (5), baby wipes (1) , diapers (4)
2	Oat milk (6), beans (2), toilet paper (9), orange juice (7)
3	Oat milk (6), milk (5), orange juice (7), toilet paper (9)
4	Beans (2), toilet paper (9), baby wipes (1), diapers (4)
5	Toilet paper (9), butter (3), baby wipes (1), diapers (4)
6	Milk (5), toilet paper (9)
7	Milk (5), rice (8)
8	Beans (2), milk (5), rice (8), toilet paper (9)
9	Milk (5), butter (3) , diapers (4)
10	Beans (2), rice (8), toilet paper (9)

Candidate Set	Support
{1}	3
{2}	5
{3}	2
{4}	4
{5}	6
{6}	2
{7}	2
{8}	4
{9}	8

Select frequent itemsets (those that appear 3 times or more)

Itemset Mining Algorithms: Apriori
Minimum support: minsup = 3

Frequent item sets of size 1: {1}, {2}, {4}, {5}, {8},{9}

Transaction ID	Items
1	Toilet paper (9), beans (2), rice (8), milk (5), baby wipes (1) , diapers (4)
2	Oat milk (6), beans (2), toilet paper (9), orange juice (7)
3	Oat milk (6), milk (5), orange juice (7), toilet paper (9)
4	Beans (2), toilet paper (9), baby wipes (1), diapers (4)
5	Toilet paper (9), butter (3), baby wipes (1), diapers (4)
6	Milk (5), toilet paper (9)
7	Milk (5), rice (8)
8	Beans (2), milk (5), rice (8), toilet paper (9)
9	Milk (5), butter (3) , diapers (4)
10	Beans (2), rice (8), toilet paper (9)

Candidate Set	Support
{1,2}	2
{1,4}	3
{1,5}	1
{1,8}	1
{1,9}	3
{2, 4}	2
{2, 5}	2
{2, 8}	3
{2, 9}	5
{4, 5}	2
{4, 8}	1
{4, 9}	3
{5,8}	3
{5, 9}	4
{8,9}	3

Generate new candidates of size k+1

Itemset Mining Algorithms: Apriori
Minimum support: minsup = 3

Frequent item sets of size 1: {1}, {2}, {4}, {5}, {8},{9}

Frequent itemsets of size 2: {1,4}, {1,9}, {2,8}, {2,9}, {4,9}, {5,8},{5,9},{8,9}

Transaction ID	Items
1	Toilet paper (9), beans (2), rice (8), milk (5), baby wipes (1) , diapers (4)
2	Oat milk (6), beans (2), toilet paper (9), orange juice (7)
3	Oat milk (6), milk (5), orange juice (7), toilet paper (9)
4	Beans (2), toilet paper (9), baby wipes (1), diapers (4)
5	Toilet paper (9), butter (3), baby wipes (1), diapers (4)
6	Milk (5), toilet paper (9)
7	Milk (5), rice (8)
8	Beans (2), milk (5), rice (8), toilet paper (9)
9	Milk (5), butter (3) , diapers (4)
10	Beans (2), rice (8), toilet paper (9)

Candidate Set	Support
{1,2}	2
{1,4}	3
{1,5}	1
{1,8}	1
{1,9}	3
{2, 4}	2
{2, 5}	2
{2, 8}	3
{2, 9}	5
{4, 5}	2
{4, 8}	1
{4, 9}	3
{5,8}	3
{5, 9}	4
{8,9}	3

Select frequent itemsets (those that appear 3 times or more)

Itemset Mining Algorithms: Apriori
Minimum support: minsup = 3

Frequent item sets of size 1: {1}, {2}, {4}, {5}, {8},{9}

Frequent itemsets of size 2: {1,4}, {1,9}, {2,8}, {2,9}, {4,9}, {5,8},{5,9},{8,9}

Frequent itemsets of size 3: {1,4,9}, {2,8,9}

Transaction ID	Items
1	Toilet paper (9), beans (2), rice (8), milk (5), baby wipes (1) , diapers (4)
2	Oat milk (6), beans (2), toilet paper (9), orange juice (7)
3	Oat milk (6), milk (5), orange juice (7), toilet paper (9)
4	Beans (2), toilet paper (9), baby wipes (1), diapers (4)
5	Toilet paper (9), butter (3), baby wipes (1), diapers (4)
6	Milk (5), toilet paper (9)
7	Milk (5), rice (8)
8	Beans (2), milk (5), rice (8), toilet paper (9)
9	Milk (5), butter (3) , diapers (4)
10	Beans (2), rice (8), toilet paper (9)

Candidate Set	Support
{1,4, 9}	3
{2,8,9}	3
{5,8,9}	2
{2,4,9}	2
{2,5,8}	2
{4,5,9}	1

At this point, no more candidates can be generated - we've found all frequent item sets

Association Rules

- An association rule is an expression $X \xrightarrow{s,c} Y$
- where X and Y are itemsets and they are disjoint, that is, $X,Y \subseteq I$, and $X \cap Y = \emptyset$.
- Let the itemset $X \cup Y$ be denoted as XY.
- The support of the rule is the number of transactions in which both X and Y co-occur as subsets:

$$s = sup(X \longrightarrow Y) = |\boldsymbol{t}(XY)| = sup(XY)$$

- The relative support of the rule is defined as the fraction of transactions where X and Y co-occur, and it provides an estimate of the joint probability of X and Y :

$$rsup(X \longrightarrow Y) = \frac{sup(XY)}{|\boldsymbol{D}|} = P(X \wedge Y)$$

- The conf idence of a rule is the conditional probability that a transaction contains Y given that it contains X:

$$c = conf(X \longrightarrow Y) = P(Y|X) = \frac{P(X \wedge Y)}{P(X)} = \frac{sup(XY)}{sup(X)}$$

Generating Association Rules

- Given a frequent itemset $Z \in F$, we look at all proper subsets $X \subset Z$ to compute rules of the form

$$X \xrightarrow{s,c} Y, \text{ where } Y = Z \setminus X$$

- where $Z \setminus X = Z - X$.
- The rule must be frequent because

$$s = sup(XY) = sup(Z) \geq minsup$$

- We compute the confidence as follows

$$\boxed{c = \frac{sup(X \cup Y)}{sup(X)} = \frac{sup(Z)}{sup(X)}}$$

- If $c \geq minconf$, then the rule is a strong rule. On the other hand, if conf $(X \rightarrow Y) < c$, then conf $(W \rightarrow Z \setminus W) < c$ for all subsets $W \subset X$, as $sup(W) \geq sup(X)$. We can thus avoid checking subsets of X.

Association Rule Mining Algorithm

AssociationRules ($\mathcal{F}$, *minconf*):

foreach $Z \in \mathcal{F}$, *such that* $|Z| \geq 2$ **do**

    $\mathcal{A} \leftarrow \{X \mid X \subset Z, X \neq \emptyset\}$

    while $\mathcal{A} \neq \emptyset$ **do**

        $X \leftarrow$ maximal element in $\mathcal{A}$

        $\mathcal{A} \leftarrow \mathcal{A} \setminus X$ `// remove` X `from` $\mathcal{A}$

        $c \leftarrow sup(Z)/sup(X)$

        if $c \geq minconf$ **then**

            print $X \longrightarrow Y$, $sup(Z)$, c

        else

            $\mathcal{A} \leftarrow \mathcal{A} \setminus \{W \mid W \subset X\}$

            `// remove all subsets of` X `from` $\mathcal{A}$

Association Rule Mining

Transaction ID	Items
1	Toilet paper (9), beans (2), rice (8), milk (5), baby wipes (1) , diapers (4)
2	Oat milk (6), beans (2), toilet paper (9), orange juice (7)
3	Oat milk (6), milk (5), orange juice (7), toilet paper (9)
4	Beans (2), toilet paper (9), baby wipes (1), diapers (4)
5	Toilet paper (9), butter (3), baby wipes (1), diapers (4)
6	Milk (5), toilet paper (9)
7	Milk (5), rice (8)
8	Beans (2), milk (5), rice (8), toilet paper (9)
9	Milk (5), butter (3) , diapers (4)
10	Beans (2), rice (8), toilet paper (9)

Frequent itemsets of size 1: {1}, {2}, {4}, {5}, {8},{9}

Frequent itemsets of size 2: {1,4}, {1,9}, {2,8}, {2,9}, {4,9}, {5,8},{5,9},{8,9}

Frequent itemsets of size 3: {1,4,9}, {2,8,9}

Association Rule Mining

Transaction ID	Items
1	Toilet paper (9), beans (2), rice (8), milk (5), baby wipes (1) , diapers (4)
2	Oat milk (6), beans (2), toilet paper (9), orange juice (7)
3	Oat milk (6), milk (5), orange juice (7), toilet paper (9)
4	Beans (2), toilet paper (9), baby wipes (1), diapers (4)
5	Toilet paper (9), butter (3), baby wipes (1), diapers (4)
6	Milk (5), toilet paper (9)
7	Milk (5), rice (8)
8	Beans (2), milk (5), rice (8), toilet paper (9)
9	Milk (5), butter (3) , diapers (4)
10	Beans (2), rice (8), toilet paper (9)

Frequent itemsets of size 1: {1}, {2}, {4}, {5}, {8},{9}

Frequent itemsets of size 2: {1,4}, {1,9}, {2,8}, {2,9}, {4,9}, {5,8},{5,9},{8,9}

Frequent itemsets of size 3: {1,4,9}, {2,8,9}

Example: $\{2,8\} \rightarrow \{9\}$

Association Rule Mining

Transaction ID	Items
1	Toilet paper (9), beans (2), rice (8), milk (5), baby wipes (1) , diapers (4)
2	Oat milk (6), beans (2), toilet paper (9), orange juice (7)
3	Oat milk (6), milk (5), orange juice (7), toilet paper (9)
4	Beans (2), toilet paper (9), baby wipes (1), diapers (4)
5	Toilet paper (9), butter (3), baby wipes (1), diapers (4)
6	Milk (5), toilet paper (9)
7	Milk (5), rice (8)
8	Beans (2), milk (5), rice (8), toilet paper (9)
9	Milk (5), butter (3) , diapers (4)
10	Beans (2), rice (8), toilet paper (9)

Frequent itemsets of size 1: {1}, {2}, {4}, {5}, {8},{9}

Frequent itemsets of size 2: {1,4}, {1,9}, {2,8}, {2,9}, {4,9}, {5,8},{5,9},{8,9}

Frequent itemsets of size 3: {1,4,9}, {2,8,9}

Example: {2,8}→{9}

{2,8} has support 3 and {2,8,9} has support 3

Association Rule Mining

Transaction ID	Items
1	Toilet paper (9), beans (2), rice (8), milk (5), baby wipes (1) , diapers (4)
2	Oat milk (6), beans (2), toilet paper (9), orange juice (7)
3	Oat milk (6), milk (5), orange juice (7), toilet paper (9)
4	Beans (2), toilet paper (9), baby wipes (1), diapers (4)
5	Toilet paper (9), butter (3), baby wipes (1), diapers (4)
6	Milk (5), toilet paper (9)
7	Milk (5), rice (8)
8	Beans (2), milk (5), rice (8), toilet paper (9)
9	Milk (5), butter (3) , diapers (4)
10	Beans (2), rice (8), toilet paper (9)

Frequent itemsets of size 1: {1}, {2}, {4}, {5}, {8},{9}

Frequent itemsets of size 2: {1,4}, {1,9}, {2,8}, {2,9}, {4,9}, {5,8},{5,9},{8,9}

Frequent itemsets of size 3: {1,4,9}, {2,8,9}

Example: {2,8}→{9}

{2,8} has support 3 and {2,8,9} has support 3

So we say the rule {2,8}→{9} has support 3 and confidence $\frac{3}{3} = 1$

Association Rule Mining

Transaction ID	Items
1	Toilet paper (9), beans (2), rice (8), milk (5), baby wipes (1) , diapers (4)
2	Oat milk (6), beans (2), toilet paper (9), orange juice (7)
3	Oat milk (6), milk (5), orange juice (7), toilet paper (9)
4	Beans (2), toilet paper (9), baby wipes (1), diapers (4)
5	Toilet paper (9), butter (3), baby wipes (1), diapers (4)
6	Milk (5), toilet paper (9)
7	Milk (5), rice (8)
8	Beans (2), milk (5), rice (8), toilet paper (9)
9	Milk (5), butter (3) , diapers (4)
10	Beans (2), rice (8), toilet paper (9)

Frequent itemsets of size 1: {1}, {2}, {4}, {5}, {8},{9}

Frequent itemsets of size 2: {1,4}, {1,9}, {2,8}, {2,9}, {4,9}, {5,8},{5,9},{8,9}

Frequent itemsets of size 3: {1,4,9}, {2,8,9}

Example: {9}→{2,8}

{9} has support 8 and {2,8,9} has support 3

So we say the rule {9}→{2,8} has support 3 and confidence $\frac{3}{8} = 0.375$

Association Rule Mining

Transaction ID	Items
1	Toilet paper (9), beans (2), rice (8), milk (5), baby wipes (1) , diapers (4)
2	Oat milk (6), beans (2), toilet paper (9), orange juice (7)
3	Oat milk (6), milk (5), orange juice (7), toilet paper (9)
4	Beans (2), toilet paper (9), baby wipes (1), diapers (4)
5	Toilet paper (9), butter (3), baby wipes (1), diapers (4)
6	Milk (5), toilet paper (9)
7	Milk (5), rice (8)
8	Beans (2), milk (5), rice (8), toilet paper (9)
9	Milk (5), butter (3) , diapers (4)
10	Beans (2), rice (8), toilet paper (9)

Frequent itemsets of size 1: $\{1\}, \{2\}, \{4\}, \{5\}, \{8\}, \{9\}$

Frequent itemsets of size 2: $\{1,4\}, \{1,9\}, \{2,8\}, \{2,9\}, \{4,9\}, \{5,8\}, \{5,9\}, \{8,9\}$

Frequent itemsets of size 3: $\{1,4,9\}$, $\{2,8,9\}$

Example: $\{4\} \rightarrow \{1\}$

$\{4\}$ *has support 4 and* $\{1,4\}$ *has support 3*

So we say the rule $\{4\} \rightarrow \{1\}$ *has support 3 and confidence* $\frac{3}{4} = 0.75$