Association Rules Small Transactions Example

Rei Sanchez-Arias, Ph.D.

Introduction to Association Rules Mining

Recap

Motivation

Market basket analysis is an association rule method that identifies associations in *transactional data*. It is an **unsupervised machine learning** technique used for **knowledge discovery**. This analysis results in a set of association rules that **identify patterns of relationships among items**.

A rule can typically be expressed in the form:
{peanut butter, jelly} $\rightarrow$ {bread}

The above rule states that if both peanut butter and jelly are purchased, then bread is also likely to be purchased. **Transactional data** can be extremely large both in terms of the quantity of transactions and the number of items monitored. Given k items that can either appear or not appear in a set, there are 2^k possible itemsets that must be searched for rules.

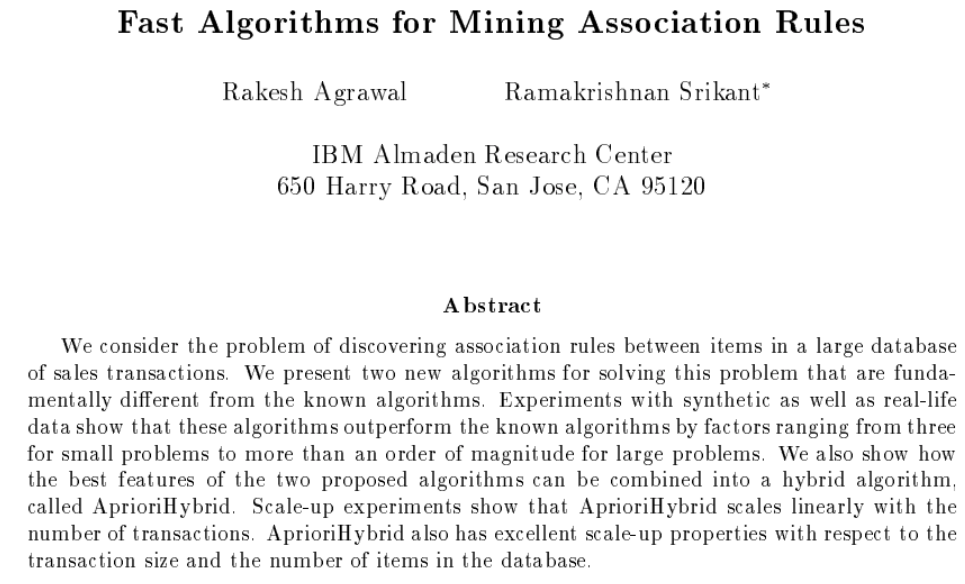
Motivation (2)

Thus, even if a retailer only has 100 distinct items, he could have $2^{100} = 1.267651 \times 10^{30}$ itemsets to evaluate, which is quite an impossible task. However, a **smart rule learner** algorithm can take advantage of the fact that in reality, many of the potential item combinations are rarely found in practice.

For example, if a retailer sells both paints and dairy products, a set of {paint, butter} are *extremely unlikely to be common*. By ignoring these rare cases, it makes it possible to limit the scope of the search for rules to a much more manageable size.

R. Agrawal and R. Srikant

R. Agrawal and R. Srikant introduced the **apriori algorithm**: it utilizes a simple prior belief (hence the name a priori) about the properties of frequent items. Using this a priori belief, **all subsets of frequent items must also be frequent**. This makes it possible to limit the number of rules to search for.



Fast Algorithms for Mining Association Rules

Rakesh Agrawal Ramakrishnan Srikant*

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

Abstract

We consider the problem of discovering association rules between items in a large database of sales transactions. We present two new algorithms for solving this problem that are fundamentally different from the known algorithms. Experiments with synthetic as well as real-life data show that these algorithms outperform the known algorithms by factors ranging from three for small problems to more than an order of magnitude for large problems. We also show how the best features of the two proposed algorithms can be combined into a hybrid algorithm, called AprioriHybrid. Scale-up experiments show that AprioriHybrid scales linearly with the number of transactions. AprioriHybrid also has excellent scale-up properties with respect to the transaction size and the number of items in the database.

Presented at the 20th int. conf. very large databases, VLDB, 1994

For example, the set {paint, butter} can only be frequent if {paint} and {butter} both occur frequently. Conversely, if neither {paint} nor {butter} are frequent, then any set containing these two items can be excluded from the search.

Support and Confidence

Let $I = \{i_1, i_2, \ldots, i_d\}$ be the set of all items in a market basket data and $T = \{t_1, t_2, \ldots, t_N\}$ be the set of all transactions. Each transaction t_i contains a subset of items chosen from I. In association analysis, a collection of zero or more items is termed an **itemset**. If an itemset contains k items, is called a k-itemset.

A transaction t_j is said to contain an itemset X if X is a subset of t_j. An important property of an itemset is its *support count*, which refers to the number of transactions that contain a particular itemset. Mathematically, the support count, $\sigma(X)$, for an itemset X can be stated as follows:

$$\sigma(X) = |t_i : X \subseteq t_i, \quad t_i \in T|$$

where the symbol $|\cdot|$ denotes the number of elements in a set.

Support and Confidence (2)

Support : This measures how frequently an itemset occurs in the data:

$$\text{Support}(X) = \frac{\text{Count}(X)}{N} = \frac{\sigma(X)}{N}$$

where X represents an item and N represents the total number of transactions.

An itemset X is called *frequent* if the support is greater than some user-defined threshold (sometimes referred to as *minsup*)

Confidence : This measures the algorithm's predictive power or accuracy. It is calculated as the support of item X and Y divided by the support of item X.

$$\text{Confidence}(X \to Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}$$

Support and Confidence (3)

Confidence measures how frequently items in Y appear in transactions containing X

$$\text{Confidence}(X \to Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

Important to notice that $\text{Confidence}(X \to Y) \neq \text{Confidence}(Y \to X)$

Lift: the lift of a rule is defined as

$$\text{Lift}(X \to Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X) \cdot \text{Support}(Y)}$$

Greater lift values ($\gg 1$) indicate stronger associations.

Apriori Algorithm: how it works

(1) Identify all itemsets that meet a minimum support threshold

This process occurs in multiple iterations. Each successive iteration evaluates the support of storing a set of **increasingly large items**. The first iteration involves evaluating the set of of 1-itemsets. The second iteration involves evaluating the set of 2-itemsets, and so on. The result of each iteration k is a set of k-itemsets that meet the minimum threshold. All itemsets from iteration k are combined in order to **generate candidate itemsets** for evaluation in iteration $k + 1$.

The apriori principle can eliminate some of the items before the next iteration begins. For example, if {A}, {B}, and {C} are frequent in iteration 1, but {D} is not, then the second iteration will only consider the itemsets {A, B}, {A, C}, and {B, C}.

(2) Create rules from these items that meet a minimum confidence threshold.

Small example: calculations

Small Example

Trace the results of using the Apriori algorithm on the grocery store example below with **support threshold** $s = 33.34\%$ and **confidence threshold** $c = 60\%$. Show the candidate and frequent itemsets for each database scan. Enumerate all the final frequent itemsets. Also indicate the association rules that are generated and highlight the strong ones, sort them by confidence.

Transaction ID	Items
T1	Hot-dogs, Buns, Ketchup
T2	Hot-dogs, Buns
T3	Hot-dogs, Coke, Chips
T4	Chips, Coke
T5	Chips, Ketchup
T6	Hot-dogs, Coke, Chips

Given the support threshold is $s = 33.34\%$, this indicates that the threshold is to consider at least 2 transactions

$$s = 0.3334 \leq \text{Support}(X) = \frac{\sigma(X)}{N} = \frac{\sigma(X)}{6}$$

so we need $\sigma(X) \geq 2$.

Calculations

Transaction ID	Items
T1	Hot-dogs, Buns, Ketchup
T2	Hot-dogs, Buns
T3	Hot-dogs, Coke, Chips
T4	Chips, Coke
T5	Chips, Ketchup
T6	Hot-dogs, Coke, Chips

- Get candidate k-itemsets X (with $\sigma(X)$)
- Find frequent k-itemsets

$k = 1$ **candidates** k itemsets: {Hot-dogs} (4), {Buns} (2), {Ketchup} (2), {Coke} (3), {Chips} (4)

$k = 2$ **candidates** k itemsets: {Hot-dogs, Buns} (2), ~~{Hot-dogs, Ketchup}~~ (1),{Hot-dogs, Coke} (2), {Hot-dogs, Chips} (2),~~{Buns, Ketchup}~~ (1), ~~{Buns, Coke}~~ (0),~~{Buns, Chips}~~ (0), ~~{Ketchup, Coke}~~ (0), ~~{Ketchup, Chips}~~ (1),{Coke, Chips} (3)

$k = 3$ **candidates**: {Hot-dogs, Coke, Chips} (2)

Note that {Hot-dogs, Buns, Coke} and {Hot-dogs, Buns, Chips} are *not candidates* when $k = 3$ because their **subsets** {Buns, Coke} and {Buns, Chips} are **not frequent**.

Note also that normally, there is no need to go to $k = 4$ since the longest transaction has only 3 items.

Calculations (cont.)

Pass	Candidate k-itemsets X (with $\sigma(X)$)	Frequent k-itemsets
$k = 1$	{Hot-dogs} (4), {Buns} (2), {Ketchup} (2), {Coke} (3), {Chips} (4)	{Hot-dogs}, {Buns}, {Ketchup}, {Coke}, {Chips}
$k = 2$	{Hot-dogs, Buns} (2),~~{Hot-dogs, Ketchup}~~ (1), {Hot-dogs, Coke} (2), {Hot-dogs, Chips} (2), ~~{Buns, Ketchup}~~ (1),~~{Buns, Coke}~~ (0), ~~{Buns, Chips}~~ (0), ~~{Ketchup, Coke}~~ (0), ~~{Ketchup, Chips}~~ (1),{Coke, Chips} (3)	{Hot-dogs, Buns}, {Hot-dogs, Coke}, {Hot-dogs, Chips}, {Coke, Chips}
$k = 3$	{Hot-dogs, Coke, Chips} (2)	{Hot-dogs, Coke, Chips}
$k = 4$	{}	{}

All Frequent Itemsets:

- {Hot-dogs}, {Buns}, {Ketchup}, {Coke}, {Chips},
- {Hot-dogs, Buns}, {Hot-dogs, Coke}, {Hot-dogs, Chips}, {Coke, Chips},
- {Hot-dogs, Coke, Chips}

Possible Rules

Association Rules: (highlighted rules satisfy the minimum confidence level)

{Hot-dogs, Buns} would generate:

- {Hot-dogs} $\rightarrow$ {Buns} (confidence: 2/4=0.5)
- **{Buns} $\rightarrow$ {Hot-dogs} (confidence: 2/2=1)**

{Hot-dogs, Coke} would generate:

- {Hot-dogs} $\rightarrow$ {Coke} (confidence: 0.5)
- **{Coke} $\rightarrow$ {Hot-dogs} (confidence: 2/3=0.66)**

{Hot-dogs, Chips} would generate:

- {Hot-dogs} $\rightarrow$ {Chips} (confidence: 0.5)
- {Chips} $\rightarrow$ {Hot-dogs} (confidence: 2/4 = 0.5)

{Coke, Chips} would generate:

- **{Coke} $\rightarrow$ {Chips} (confidence: 3/3 = 1)**
- **{Chips} $\rightarrow$ {Coke} (confidence: 3/4 = 0.75)**

{Hot-dogs, Coke, Chips} would generate:

- {Hot-dogs} $\rightarrow$ {Coke, Chips} (confidence: 0.5)
- **{Coke} $\rightarrow$ {Hot-dogs, Chips} (confidence: 2/3 = 0.66)**
- {Chips} $\rightarrow$ {Hot-dogs, Coke} (confidence: 0.5)
- **{Hot-dogs, Coke} $\rightarrow$ {Chips} (confidence: 2/2 = 1)**
- **{Hot-dogs, Chips} $\rightarrow$ {Coke} (confidence: 2/2 = 1)**
- **{Chips, Coke} $\rightarrow$ {Hot-dogs} (confidence: 2/3 = 0.66)**

Possible Rules (cont.)

With the confidence threshold set to 60%, the strong association rules are (sorted by confidence):

1. {Coke} $\rightarrow$ {Chips} (confidence: 3/3 = 1)
2. {Buns} $\rightarrow$ {Hot-dogs} (confidence: 2/2=1)
3. {Hot-dogs, Coke} $\rightarrow$ {Chips} (confidence: 2/2 = 1)
4. {Hot-dogs, Chips} $\rightarrow$ {Coke} (confidence: 2/2 = 1)
5. {Chips}$\to$ {Coke} (confidence: 3/4 = 0.75)
6. {Coke} $\rightarrow$ {Hot-dogs} (confidence: 2/3=0.66)
7. {Coke} $\rightarrow$ {Hot-dogs, Chips} (confidence: 2/3 = 0.66)
8. {Chips, Coke} $\rightarrow$ {Hot-dogs} (confidence: 2/3 = 0.66)