Frequent Pattern Mining and the Apriori Algorithm: A Concise Technical Overview

<= Previous post
Next post =>

Tags: Algorithms, Apriori, Association Rules, Frequent Pattern Mining

This post provides a technical overview of frequent pattern mining algorithms (also known by a variety of other names), along with its most famous implementation, the Apriori algorithm.

By Matthew Mayo, KDnuggets.

Frequent pattern mining. Association mining. Correlation mining. Association rule learning. The Apriori algorithm.

These are all related, yet distinct, concepts that have been used for a very long time to describe an aspect of data mining that many would argue is the very essence of the term *data mining*: taking a set of data and applying statistical methods to find interesting and previously-unknown patterns within said set of data. We aren't looking to classify instances or perform instance clustering; we simply want to learn patterns of subsets which emerge within a dataset and across instances, which ones emerge frequently, which items are associated, and which items correlate with others. It's easy to see why the above terms become conflated.

So, let's have a look at this essential aspect of data mining. Foregoing the Apriori algorithm for now, I will simply use the term *frequent pattern mining* to refer to the big tent of concepts outlined above, even if somewhat flawed (and even if I personally prefer the less often used term *association mining*).

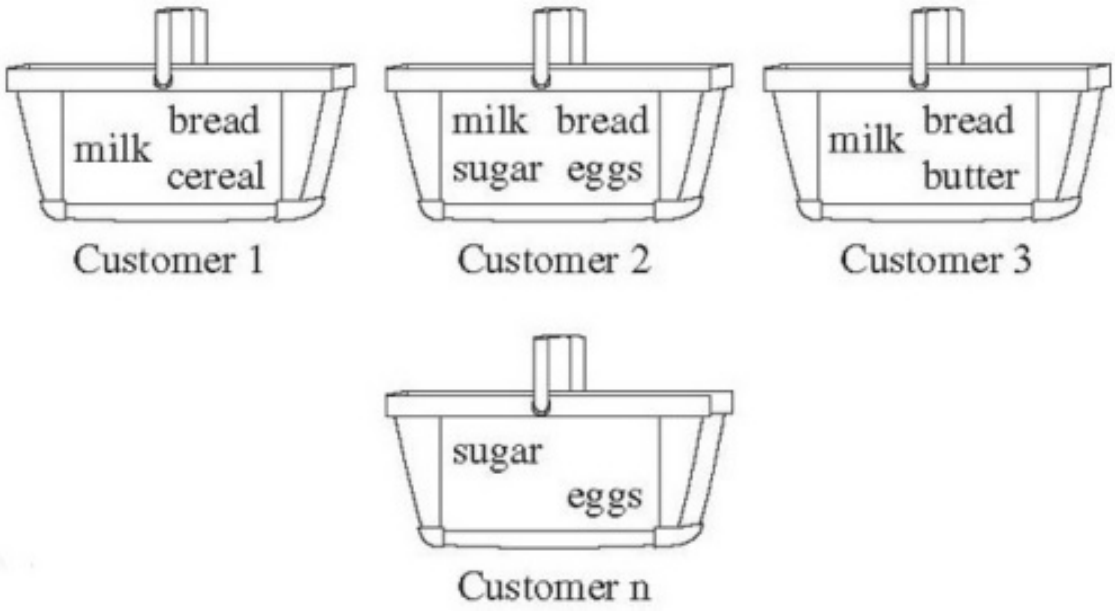

Fig.1: Market Basket Analysis (Han, Kamber & Pei).

Market Basket Analysis

Frequent patterns are patterns which appear frequently within a dataset (surprised?). A *frequent itemset* is one which is made up of one of these patterns, which is why frequent pattern mining is often alternately referred to as frequent itemset mining.

Frequent pattern mining is most easily explained by introducing *market basket analysis* (or affinity analysis), a typical usage for which it is well-known. Market basket analysis attempts to identify associations, or patterns, between the various items that have been chosen by a particular shopper and placed in their market basket, be it real or virtual, and assigns support and confidence measures for comparison. The value of this lies in cross-marketing and customer behavior analysis.

The generalization of market basket analysis is frequent pattern mining, and is actually quite similar to classification except that any attribute, or combination of attributes (and not just the *class*), can be predicted in association. As association does not require the pre-labeling of classes, it is a form of unsupervised learning.

Confidence, Support, and Association Rules

If we think of the total set of items available in our set (sold at a physical store, at an online retailer, or something else altogether, such as transactions for fraud detection analysis), then each item can be represented by a Boolean variable, representing whether or not the item is present within a given "basket." Each basket is then simply a Boolean vector, possibly quite lengthy dependent on the number of available items. A dataset would then be the resulting matrix of all possible basket vectors.

This collection of Boolean basket vectors are then analyzed for associations, patterns, correlations, or whatever it is you would like to call these relationships. One of the most common ways to represent these patterns is via **association rules**, a single example of which is given below:

milk => bread [support = 25%, confidence = 60%]

How do we know how interesting or insightful a given rule may be? That's where support and confidence come in.

Support is a measure of *absolute frequency*. In the above example, the support of 25% indicates that, in our finite dataset, milk and bread are purchased together in 25% of all transactions.

Confidence is a measure of *correlative frequency*. In the above example, the confidence of 60% indicates that 60% of those who purchased milk also purchased bread.

In a given application, association rules are generally generated within the bounds of some predefined minimum threshold for both confidence and support, and rules are only considered interesting and insightful if they meet these minimum thresholds.

Apriori

Apriori enjoys success as the most well-known example of a frequent pattern mining algorithm. Given the above treatment of market basket analysis and item representation, Apriori datasets tend to be large, sparse matrices, with items (attributes) along the horizontal axis, and transactions (instances) along the vertical axis.

From an initial dataset of *n* attributes, Apriori computes a list of candidate itemsets, generally ranging from size 2 to *n*-1, or some other specified bounds. The number of possible itemsets of size *n*-(*n*+1) to *n*-1 that can be constructed from a dataset of size *n* can be determined as follows, using combinations:

$$\mathrm{C}(\mathrm{n}, \mathrm{n}-(\mathrm{n}+2))+\mathrm{C}(\mathrm{n}, \mathrm{n}-(\mathrm{n}+3))+\ldots+\mathrm{C}(\mathrm{n}, \mathrm{n}-1).$$

The above can also be expressed using the binomial coefficient.

Very large itemsets held within extremely large and sparse matrices can prove very computationally expensive.

```
INPUT: S, support where S = dataset, min_support = real
OUTPUT: Set of Frequent Itemsets
Require: S ≠ ∅, 0 ≤ min_support ≤ 1
 1: procedure GETFREQUENTITEMSETS
 2:     freqSets[ ] ← null
 3:     for all Itemsets i in S do
 4:         if support ≥ min_support then
 5:             freqSets[ ] ← i
 6:         end if
 7:     end for
 8: end procedure
```

Fig.2: Apriori Candidate Itemset Generation Algorithm

A support value is provided to the algorithm. First, the algorithm generates a list of candidate itemsets, which includes all of the itemsets appearing within the dataset. Of the candidate itemsets generated, an itemset can be determined to be frequent if the number of transactions that it appears in is greater than the support value.

```
INPUT: S where S = dataset
OUTPUT: Set of Candidate Itemsets
Require: S ≠ ∅
 1: procedure GENERATECANDIDATES
 2:     i ← 2
 3:     num ← NumAttributes(S)
 4:     candidates[ ] ← null
 5:     while i < num do
 6:         candidates[ ] ← all sets of size i, support
 7:         i ← i + 1
 8:     end while
 9: end procedure
```

Fig.3: Apriori Frequency Itemset Selection Algorithm

Explicit association rules can then trivially be generated by traversing the frequent itemsets, and computing associated confidence levels. Confidence is the proportion of the transactions containing item A which also contains item B, and is calculated as

$$\text{Confidence}(A \Rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$

ID	Rule	Support	Confidence
r1	$\{a, b, c\} \Rightarrow \{e\}$	0.5	1.0
r2	$\{a\} \rightarrow \{c, e, f\}$	0.5	0.66
r3	$\{a, b\} \rightarrow \{e, f\}$	0.5	1.0
r4	$\{b\} \rightarrow \{e, f\}$	0.75	0.75
r5	$\{a\} \rightarrow \{e, f\}$	0.75	1.0
r6	$\{c\} \rightarrow \{f\}$	0.5	1.0
r7	$\{a\} \rightarrow \{b\}$	0.5	0.66
...	...	...	...

Fig.4: Sample Association Rules with Support and Confidence
(Source: An introduction to frequent pattern mining, by Philippe Fournier-Viger.

The manner in which Apriori works is quite simple; it computes all of the rules that meet minimum support and confidence values. The number of possible potential rules increases exponentially with the number of items in the itemset. Since the computation of new rules does not rely on previously computed rules, the Apriori algorithm provides an opportunity for parallelism to offset computation time.

Check out the results of a recent KDnuggets poll outlining other algorithms which are available and in use today.

Related:

- Support Vector Machines: A Concise Technical Overview
- Comparing Clustering Techniques: A Concise Technical Overview
- Association Rules and the Apriori Algorithm: A Tutorial

<= Previous post
Next post =>

Top Stories Past 30 Days

Most Popular

1. **Why the Future of ETL Is Not ELT, But EL(T)**
2. **20 Core Data Science Concepts for Beginners**
3. **How to Get Into Data Science Without a Degree**
4. **TabPy: Combining Python and Tableau**
5. **How to Acquire the Most Wanted Data Science Skills**
6. **Top Python Libraries for Deep Learning, Natural Language Processing & Computer Vision**
7. **Introduction to Data Engineering**

Most Shared

1. **How to Get Into Data Science Without a Degree**
2. **Why the Future of ETL Is Not ELT, But EL(T)**
3. **20 Core Data Science Concepts for Beginners**
4. **Top Python Libraries for Deep Learning, Natural Language Processing & Computer Vision**
5. **Learn Deep Learning with this Free Course from Yann LeCun**
6. **AI, Analytics, Machine Learning, Data Science, Deep Learning Research Main Developments in 2020 and Key Trends for 2021**
7. **A Rising Library Beating Pandas in Performance**

Latest News

- Monte Carlo integration in Python
- How to easily check if your Machine Learning model is f...
- SQL vs NoSQL: 7 Key Takeaways
- Can you trust AutoML?
- XGBoost: What it is, and when to use it

- KDnuggets 20:n48, Dec 23: Crack SQL Interviews; MLOps-...

Top Stories
Last Week

Most Popular

1. **20 Core Data Science Concepts for Beginners**

2. **A Rising Library Beating Pandas in Performance**
3. **R or Python? Why Not Both?**
4. **Why the Future of ETL Is Not ELT, But EL(T)**
5. **10 Python Skills They Dont Teach in Bootcamp**
6. **Introduction to Data Engineering**
7. **Essential Math for Data Science: Probability Density and Probability Mass Functions**

Most Shared

1. **20 Core Data Science Concepts for Beginners**
2. **A Rising Library Beating Pandas in Performance**
3. **Main 2020 Developments and Key 2021 Trends in AI, Data Science, Machine Learning Technology**
4. **R or Python? Why Not Both?**
5. **Artificial Intelligence in Modern Learning System : E-Learning**
6. **Essential Math for Data Science: Probability Density and Probability Mass Functions**
7. **10 Python Skills They Dont Teach in Bootcamp**

More Recent Stories

- KDnuggets 20:n48, Dec 23: Crack SQL Interviews; MLOps-...
- The Future of Cloud is Now
- Resampling Imbalanced Data and Its Limits
- Feature Store vs Data Warehouse
- 5 strategies for enterprise machine learning for 2021
- Top 9 Data Science Courses to Learn Online
- Production Machine Learning Monitoring: Outliers, Drift, Expla...
- MLOps Is Changing How Machine Learning Models Are Developed
- Fast and Intuitive Statistical Modeling with Pomegranate
- Optimization Algorithms in Neural Networks
- MLOps – “Why is it required?” and “What it...
- Navigate the road to Responsible AI
- Top 2020 Stories: 24 Best (and Free) Books To Understand Machi...
- ebook: Fundamentals for Efficient ML Monitoring
- Undersampling Will Change the Base Rates of Your Model’s...
- Crack SQL Interviews
- 8 Places for Data Professionals to Find Datasets
- Top tweets, Dec 09-15: Main 2020 Developments, Key 2021 Tre...
- How to use Machine Learning for Anomaly Detection and Conditio...
- Industry 2021 Predictions for AI, Analytics, Data Science, Mac...

x