Apriori Association Rule Tutorial

September 21, 2025

1 Introduction

Apriori is a classic algorithm for mining frequent itemsets and association rules from transactional datasets. By leveraging the downward closure property (all subsets of a frequent itemset are frequent), Apriori iteratively expands candidate itemsets while pruning the search space, ultimately producing rules ranked by support, confidence, and lift.

2 Theory and Formulas

2.1 Support, Confidence, and Lift

Given an itemset X, support measures how often it appears:

$$\operatorname{supp}(X) = \frac{|\{T \mid X \subseteq T,\ T \in \mathcal{D}\}|}{|\mathcal{D}|}. \tag{1}$$

For a rule $X \Rightarrow Y$ with $X \cap Y = \emptyset$, confidence and lift are defined as

$$\operatorname{conf}(X \Rightarrow Y) = \frac{\operatorname{supp}(X \cup Y)}{\operatorname{supp}(X)}, \tag{2}$$

$$\operatorname{lift}(X \Rightarrow Y) = \frac{\operatorname{conf}(X \Rightarrow Y)}{\operatorname{supp}(Y)}. \tag{3}$$

Lift greater than 1 indicates positive association; values below 1 suggest substitution effects.

2.2 Apriori Iteration

Apriori proceeds level-wise:

1. Generate candidate 1-itemsets and discard those with support below `min_supp`.
2. At level k, join frequent $(k-1)$-itemsets to create candidate k-itemsets.
3. Prune candidates whose $(k-1)$-subsets are not frequent (Apriori property).
4. Scan the database to compute support; retain itemsets meeting `min_supp`.
5. Repeat until no candidates remain, then derive rules satisfying `min_conf`.

The number of database scans equals the largest frequent itemset size, making support thresholds and dataset sparsity critical for efficiency.

2.3 Evaluation Metrics

Beyond support and confidence, measures such as conviction, leverage, and lift help prioritize rules. Conviction is

$$\mathrm{conv}(X \Rightarrow Y) = \frac{1 - \mathrm{supp}(Y)}{1 - \mathrm{conf}(X \Rightarrow Y)}, \tag{4}$$

penalizing rules that fail frequently. Visual analysis of metric distributions aids in selecting thresholds that balance rule volume and significance.

3 Applications and Tips

- **Market basket analysis**: discover product bundles for promotions and store layout optimization.
- **Recommendation systems**: augment collaborative filtering with interpretable co-purchase rules.
- **Fraud and anomaly detection**: identify unusual item combinations indicative of risky behavior.
- **Best practices**: discretize continuous variables, remove ubiquitous items that dominate support, tune `min_supp` and `min_conf` iteratively, and validate rules with hold-out data or domain experts.

4 Python Practice

The script `gen_apriori_figures.py` synthesizes transactional data, mines frequent itemsets using a simple Apriori implementation, and visualizes the support–confidence landscape and lift distribution for discovered rules.

Listing 1: Excerpt from gen$_{a}priori_{f}igures.py$

```python
from itertools import combinations

rules = []
for itemset, support in frequent_itemsets.items():
    for split in range(1, len(itemset)):
        for lhs in combinations(itemset, split):
            rhs = tuple(sorted(set(itemset) - set(lhs)))
            conf = support / frequent_itemsets[lhs]
            lift = conf / frequent_itemsets[rhs]
            rules.append((lhs, rhs, support, conf, lift))
```

5 Result

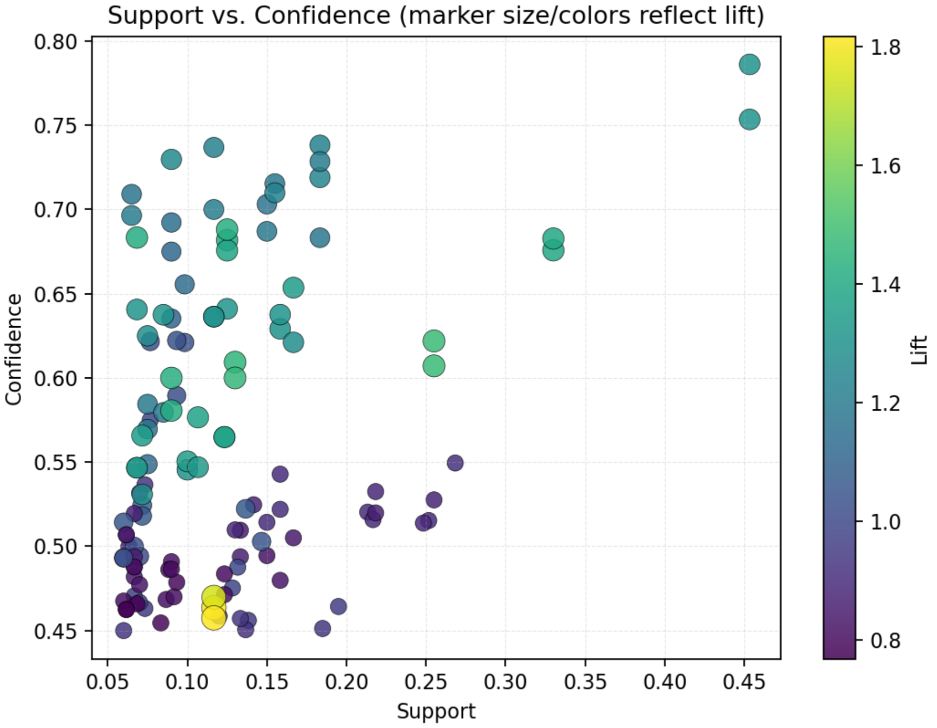

Figure 1: Support vs. confidence for mined rules, with marker size proportional to lift

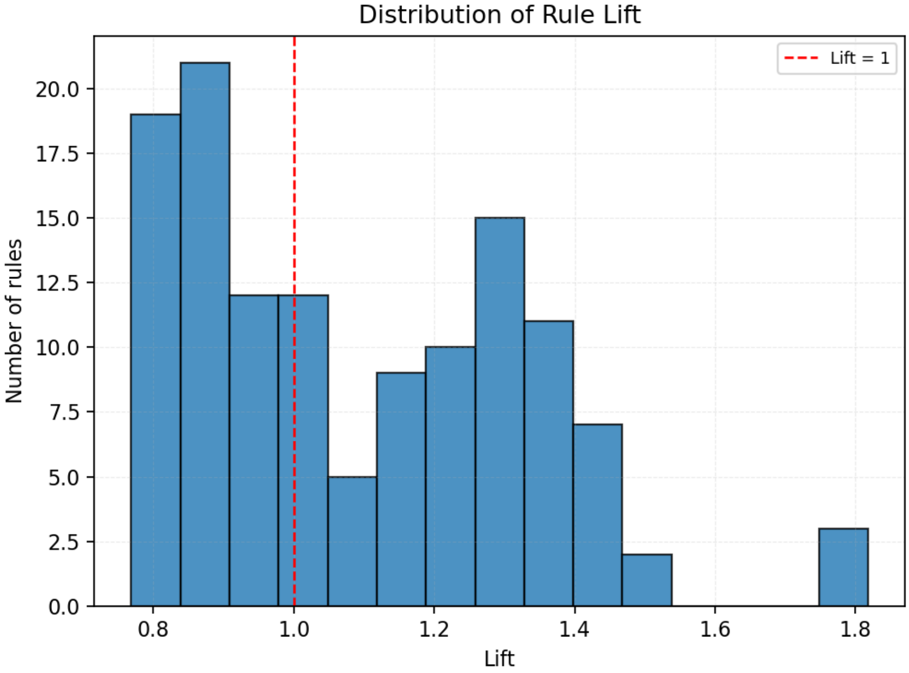

Figure 2: Distribution of rule lift values highlighting high-association pairs

6 Summary

Apriori exploits the downward closure property to mine interpretable association rules from transactional data. Effective deployment hinges on carefully chosen support and confidence thresholds, complementary metrics like lift or conviction, and domain validation. The synthetic example illustrates how visual diagnostics reveal the trade-offs when tuning rule-mining parameters.