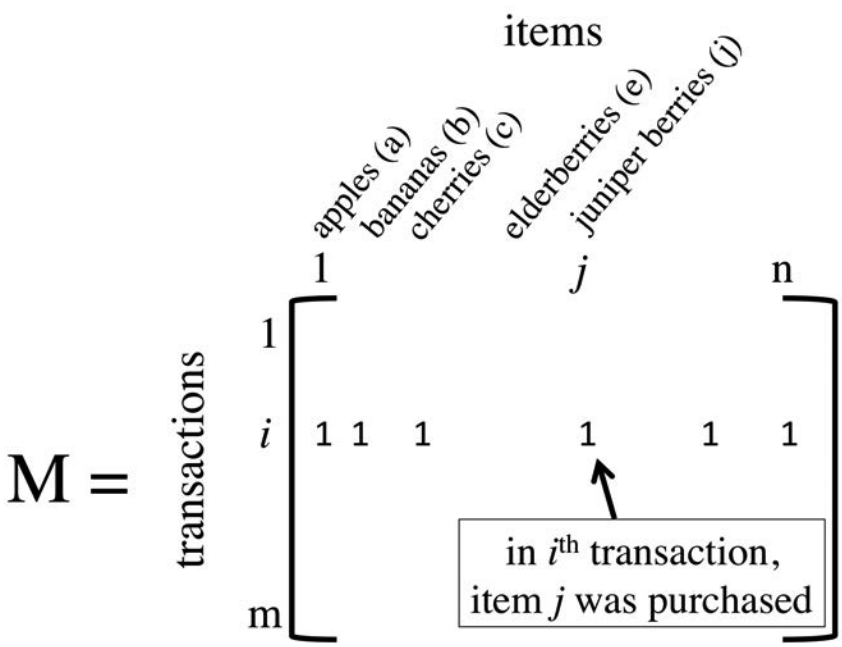
Rule Mining and the Apriori Algorithm

MIT 15.097 Course Notes

Cynthia Rudin

The Apriori algorithm - often called the "first thing data miners try," but somehow doesn't appear in most data mining textbooks or courses!

Start with market basket data:



Some important definitions:

- **Itemset**: a subset of items, e.g., (bananas, cherries, elderberries), indexed by $\{2, 3, 5\}$.
- **Support** of an itemset: number of transactions containing it,

$$\text{Supp(bananas, cherries, elderberries)} = \sum_{i=1}^{m} M_{i,2} \cdot M_{i,3} \cdot M_{i,5}.$$

- **Confidence** of rule $a \to b$: the fraction of times itemset b is purchased when itemset a is purchased.

$$\begin{aligned} \text{Conf}(a \to b) &= \frac{\text{Supp}(a \cup b)}{\text{Supp}(a)} = \frac{\#\text{times } a \text{ and } b \text{ are purchased}}{\#\text{times } a \text{ is purchased}} \\ &= \hat{P}(b|a). \end{aligned}$$

We want to find all **strong rules**. These are rules $a \to b$ such that:

$$\text{Supp}(a \cup b) \geq \theta, \text{ and } \text{Conf}(a \to b) \geq \text{minconf}.$$

Here θ is called the **minimum support threshold**.

The support has a monotonicity property called *downward closure*:

$$\text{If Supp}(a \cup b) \geq \theta \text{ then Supp}(a) \geq \theta \text{ and Supp}(b) \geq \theta.$$

That is, if $a \cup b$ is a frequent item set, then so are a and b.

$$\begin{aligned} \text{Supp}(a \cup b) &= \#\text{times } a \text{ and } b \text{ are purchased} \\ &\leq \#\text{times } a \text{ is purchased} = \text{Supp}(a). \end{aligned}$$

Apriori finds all frequent itemsets (a such that $\text{Supp}(a) \geq \theta$). We can use Apriori's result to get all strong rules $a \to b$ as follows:

- For each frequent itemset ℓ:
  - Find all nonempty subsets of ℓ
  - For each subset a, output $a \to \{\ell \setminus a\}$ whenever

$$\frac{\text{Supp}(\ell)}{\text{Supp}(a)} \geq \text{minconf}.$$

Now for Apriori. Use the downward closure property: generate all k-itemsets (itemsets of size k) from $(k-1)$-itemsets. It's a breadth-first-search.

$\boxed{\text{Example:}}$
$\theta = 10$

	apples	bananas	cherries		elderberries		grapes
1-itemsets:	a	b	c	d	e	~~f~~	g
supp:	25	20	30	45	29	5	17

2-itemsets:	~~{a,b}~~	{a,c}	{a,d}	{a,e}	...	~~{e,g}~~
supp:	7	25	15	23		3

3-itemsets:	{a,c,d}	{a,c,e}	{b,d,g}	...
supp:	15	22	15	

4-itemsets:	{a,c,d,e}
supp:	12

Apriori Algorithm:

Input: Matrix M

L_1={frequent 1-itemsets; i such that Supp(i) $\geq \theta$}.

For $k = 2$, while $L_{k-1} \neq \emptyset$ (while there are large $k-1$-itemsets), $k++$

- C_k = apriori_gen(L_{k-1}) generate candidate itemsets of size k
- $L_k = \{c : c \in C_k,\ \text{Supp}(c) \geq \theta\}$ frequent itemsets of size k (loop over transactions, scan the database)

end

Output: $\bigcup_k L_k$.

The subroutine apriori_gen joins L_{k-1} to L_{k-1}.

apriori_gen Subroutine:

Input: L_{k-1}

Find all pairs of itemsets in L_{k-1} where the first $k-2$ items are identical.

Union them (lexicographically) to get $C_k^{\text{too big}}$,

$$\text{e.g.,}\{a,b,c,d,e,f\},\{a,b,c,d,e,g\} \rightarrow \{a,b,c,d,e,f,g\}$$

Prune: $C_k = \{c \in C_k^{\text{too big}}$, all $(k-1)$-subsets c_s of c obey $c_s \in L_{k-1}\}$.

Output: C_k.

Example of Prune step: consider $\{a,b,c,d,e,f,g\}$ which is in $C_k^{\text{too big}}$, and I want to know whether it's in C_k. Look at $\{\cancel{a},b,c,d,e,f,g\}$, $\{a,\cancel{b},c,d,e,f,g\}$,$\{a,b,\cancel{c},d,e,f,g\}$, $\{a,b,c,\cancel{d},e,f,g\}$, etc. If any are not in L_6, then prune $\{a,b,c,d,e,f,g\}$ from L_7.

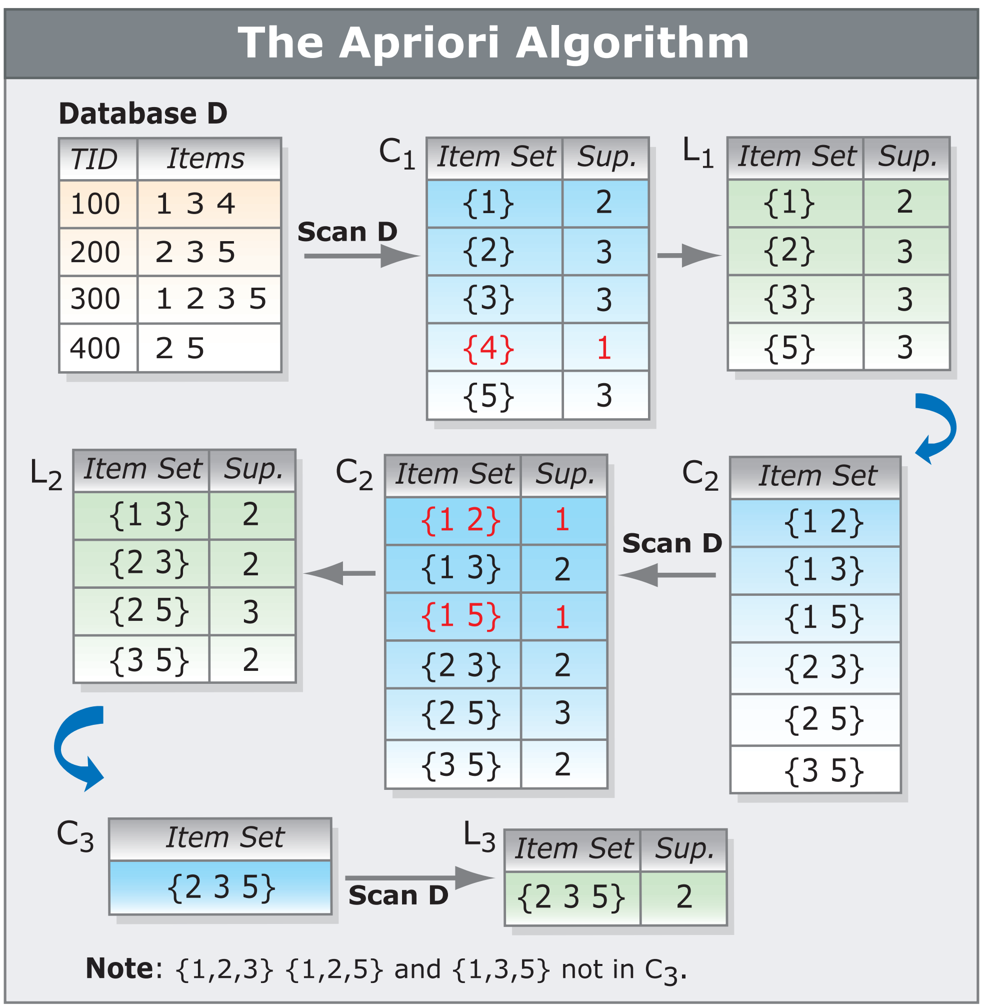


Image by MIT OpenCourseWare, adapted from Osmar R. Zaïane.

- $\boxed{\text{Apriori scans the database at most how many times?}}$
- Huge number of candidate sets. ☹
- Spawned huge number of apriori-like papers.

What do you do with the rules after they're generated?

- Information overload (give up)
- Order rules by "interestingness"
  - Confidence
$$\hat{P}(b|a) = \frac{\text{Supp}(a \cup b)}{\text{Supp}(a)}$$
  - "Lift"/"Interest"
$$\frac{\hat{P}(b|a)}{\hat{P}(b)} = \frac{\text{Supp}(b)}{1 - \frac{\text{Supp}(a \cup b)}{\text{Supp}(a)}}$$
    ⋮
  - Hundreds!

Research questions:

- mining more than just itemsets (e.g, sequences, trees, graphs)
- incorporating taxonomy in items
- boolean logic and "logical analysis of data"
- Cynthia's questions: Can we use rules within ML to get good predictive models?