Frequent Itemsets & Association Rules
(a.k.a. counting co-occurrences)
The Market-Basket Model

- **Baskets** = sets of purchases, **Items** = products;
- **Brick and Mortar**: Track purchasing habits
  - Chain stores have TBs of transaction data
  - Tie-in “tricks”, e.g., sale on diapers + raise price of beer
  - Need the rule to occur frequently, or no $$’s
- **Online**: People who bought $X$ also bought $Y$

**Input:**

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bread, Coke, Milk</td>
</tr>
<tr>
<td>2</td>
<td>Beer, Bread</td>
</tr>
<tr>
<td>3</td>
<td>Beer, Coke, Diaper, Milk</td>
</tr>
<tr>
<td>4</td>
<td>Beer, Bread, Diaper, Milk</td>
</tr>
<tr>
<td>5</td>
<td>Coke, Diaper, Milk</td>
</tr>
</tbody>
</table>

**Output:**

- Rules Discovered:
  - $\{\text{Milk}\} \rightarrow \{\text{Coke}\}$
  - $\{\text{Diaper, Milk}\} \rightarrow \{\text{Beer}\}$

Examples: Plagiarism, Side-Effects

- **Baskets** = sentences; **Items** = documents containing those sentences
  - Items that appear together too often could represent plagiarism
  - Notice items do not have to be “in” baskets

- **Baskets** = patients; **Items** = drugs & side-effects
  - Has been used to detect combinations of drugs that result in particular side-effects
  - **Requires extension:** Absence of an item needs to be observed as well as presence

*adapted from* J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, [http://www.mmds.org](http://www.mmds.org)
Example: Voting records

- **Baskets** = politicians;
  **Items** = party & votes

- Can extract set of votes most associated with each party (or subparty)

<table>
<thead>
<tr>
<th>Association Rule</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>{budget resolution = no, MX-missile=no, aid to El Salvador = yes } → {Republican}</td>
<td>91.0%</td>
</tr>
<tr>
<td>{budget resolution = yes, MX-missile=yes, aid to El Salvador = no } → {Democrat}</td>
<td>97.5%</td>
</tr>
<tr>
<td>{crime = yes, right-to-sue = yes, physician fee freeze = yes} → {Republican}</td>
<td>93.5%</td>
</tr>
<tr>
<td>{crime = no, right-to-sue = no, physician fee freeze = no} → {Democrat}</td>
<td>100%</td>
</tr>
</tbody>
</table>

Frequent Itemsets

• Simplest question: Find sets of items that appear together “frequently” in baskets

• Support $\sigma(X)$ for itemset $X$: Number of baskets containing all items in $X$
  • (Often expressed as a fraction of the total number of baskets)

• Given a support threshold $\sigma_{\text{min}}$, then sets of items $X$ that appear in at least $\sigma(X) \geq \sigma_{\text{min}}$ baskets are called \textit{frequent itemsets}

Example: Frequent Itemsets

- Items = \{milk, coke, pepsi, beer, juice\}
- Baskets
  - $B_1 = \{m, c, b\}$
  - $B_2 = \{m, p, j\}$
  - $B_3 = \{m, b\}$
  - $B_4 = \{c, j\}$
  - $B_5 = \{m, c, b\}$
  - $B_6 = \{m, c, b, j\}$
  - $B_7 = \{c, b, j\}$
  - $B_8 = \{b, c\}$
- Frequent itemsets ($\sigma(X) \geq 3$):
  - \{m\}:5, \{c\}:6, \{b\}:6, \{j\}:4, \{m,c\}: 3,
  - \{m,b\}:4, \{c,b\}:5, \{c,j\}:3, \{m,c,b\}:3
Association Rules

• If-then rules about the contents of baskets

• \( \{a_1, a_2, \ldots, a_k\} \rightarrow b \) means: “if a basket contains all of \( a_1, \ldots, a_k \) then it is likely to contain \( b \)”

• In practice there are many rules, want to find significant/interesting ones!

• **Confidence** of this association rule is the probability of \( B=\{b\} \) given \( A=\{a_1, \ldots, a_k\} \)

\[
\text{Support, } s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N};
\]

\[
\text{Confidence, } c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}. 
\]

Interest of Association Rules

- Not all high-confidence rules are interesting
  - The rule $A \rightarrow \text{milk}$ may have high confidence because milk is just purchased very often (independent of $A$)

- Interest Factor (or Lift) of a rule $A \rightarrow B$:

\[
Lift = \frac{c(A \rightarrow B)}{s(B)} \quad I(A, B) = \frac{s(A, B)}{s(A) \times s(B)}
\]
Example: Confidence and Interest

\[ B_1 = \{m, c, b\} \quad B_2 = \{m, p, j\} \]
\[ B_3 = \{m, b\} \quad B_4 = \{c, j\} \]
\[ B_5 = \{m, c, b\} \quad B_6 = \{m, c, b, j\} \]
\[ B_7 = \{c, b, j\} \quad B_8 = \{b, c\} \]

- Association rule: \( \{m\} \rightarrow b \)
  - Confidence = \(\frac{4}{5}\)
  - Interest Factor = \(\frac{1}{6} \cdot \frac{4}{5} = \frac{4}{30}\)
    - Item \(b\) appears in \(\frac{6}{8}\) of the baskets
    - Rule is not very interesting!

Many measures of interest

<table>
<thead>
<tr>
<th>Measure (Symbol)</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goodman-Kruskal (λ)</td>
<td>( \left( \sum_j \max_k f_{jk} - \max_k f_{+k} \right) / \left( N - \max_k f_{+k} \right) )</td>
</tr>
<tr>
<td>Mutual Information (M)</td>
<td>( \left( \sum_i \sum_j \frac{f_{ij}}{N} \log \frac{N f_{ij}}{f_{i+} f_{j+}} \right) / \left( - \sum_i \frac{f_{i+}}{N} \log \frac{f_{i+}}{N} \right) )</td>
</tr>
<tr>
<td>J-Measure (J)</td>
<td>( \frac{f_{11}}{N} \log \frac{N f_{11}}{f_{1+} f_{+1}} + \frac{f_{10}}{N} \log \frac{N f_{10}}{f_{1+} f_{+0}} )</td>
</tr>
<tr>
<td>Gini index (G)</td>
<td>( \frac{f_{1+}}{N} \times \left( \frac{f_{11}}{f_{1+}} \right)^2 + \left( \frac{f_{10}}{f_{1+}} \right)^2 \right] - \left( \frac{f_{+1}}{N} \right)^2 )</td>
</tr>
<tr>
<td>Laplace (L)</td>
<td>( \frac{f_{11} + 1}{f_{1+} + 2} )</td>
</tr>
<tr>
<td>Conviction (V)</td>
<td>( \frac{f_{1+} f_{+0}}{N f_{10}} )</td>
</tr>
<tr>
<td>Certainty factor (F)</td>
<td>( \frac{f_{11} f_{+1} - \frac{f_{+1}}{N}}{1 - \frac{f_{+1}}{N}} )</td>
</tr>
<tr>
<td>Added Value (AV)</td>
<td>( \frac{f_{11}}{f_{1+}} - \frac{f_{+1}}{N} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>( f_{11} )</th>
<th>( f_{10} )</th>
<th>( f_{1+} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{11} )</td>
<td>( f_{11} )</td>
<td>( f_{10} )</td>
<td>( f_{1+} )</td>
</tr>
<tr>
<td>( f_{01} )</td>
<td>( f_{01} )</td>
<td>( f_{00} )</td>
<td>( f_{0+} )</td>
</tr>
<tr>
<td>( f_{+1} )</td>
<td>( f_{+1} )</td>
<td>( f_{+0} )</td>
<td>( N )</td>
</tr>
</tbody>
</table>

Mining Association Rules

• Problem: Find all association rules with support $\geq s$ and confidence $\geq c$

• Note: Support of an association rule is the support of the set of items on the left side

• Hard part: Finding the frequent itemsets!

• If $\{i_1, i_2, \ldots, i_k\} \rightarrow j$ has high support and confidence, then both $\{i_1, i_2, \ldots, i_k\}$ and $\{i_1, i_2, \ldots, i_k, j\}$ will be “frequent”

Mining Association Rules

- **Step 1:** Find all frequent itemsets $I$
- **Step 2:** Rule generation
  - For every subset $A$ of $I$, generate a rule $A \rightarrow I \setminus A$
    - Since $I$ is frequent, $A$ is also frequent
    - **Variant 1:** Single pass to compute the rule confidence
      - $c(A,B \rightarrow C,D) = \sigma(A,B,C,D) / \sigma(A,B)$
    - **Variant 2:** If $A,B,C \rightarrow D$ is below confidence, then so is $A,B \rightarrow C,D$
      - Can generate “bigger” rules from smaller ones!
  - Output the rules above the confidence threshold

Example: Mining Association Rules

\[ B_1 = \{m, c, b\} \quad B_2 = \{m, p, j\} \]
\[ B_3 = \{m, b\} \quad B_4 = \{c, j\} \]
\[ B_5 = \{m, c, b\} \quad B_6 = \{m, c, b, j\} \]
\[ B_7 = \{c, b, j\} \quad B_8 = \{b, c\} \]

Support threshold \( \sigma_{\text{min}} = 3 \), confidence \( c = 0.75 \)

1) Frequent itemsets:
\{b,m\}:4  \{c,m\}:3  \{b,c\}:5  \{c,j\}:3  \{m,c,b\}: 3

2) Generate rules:
\[ b \rightarrow m: \epsilon = 4/6 \quad b \rightarrow c: \epsilon = 5/6 \quad b,c \rightarrow m: \epsilon = 3/5 \]
\[ m \rightarrow b: \epsilon = 4/5 \quad \ldots \quad b,m \rightarrow c: \epsilon = 3/4 \]
\[ b \rightarrow c,m: \epsilon = 3/6 \]

Finding Frequent Item Sets

Given $k$ products, how many possible item sets are there?

Finding Frequent Item Sets

Answer: $2^k - 1 \rightarrow$ Cannot enumerate all possible sets

Observation: A-priori Principle

Subsets of a frequent item set are also frequent.
Corollary: Pruning of Candidates

If we know that a subset is not frequent, then we can ignore all its supersets.
A-priori Algorithm

Algorithm 6.1 Frequent itemset generation of the Apriori algorithm.

1: \( k = 1 \).
2: \( F_k = \{ i \mid i \in I \land \sigma(\{i\}) \geq N \times minsup \} \).  \{Find all frequent 1-itemsets\}
3: repeat
4: \( k = k + 1 \).
5: \( C_k = \text{apriori-gen}(F_{k-1}) \).  \{Generate candidate itemsets\}
6: for each transaction \( t \in T \) do
7: \( C_t = \text{subset}(C_k, t) \).  \{Identify all candidates that belong to \( t \}\}
8: for each candidate itemset \( c \in C_t \) do
9: \( \sigma(c) = \sigma(c) + 1 \).  \{Increment support count\}
10: end for
11: end for
12: \( F_k = \{ c \mid c \in C_k \land \sigma(c) \geq N \times minsup \} \).  \{Extract the frequent \( k \)-itemsets\}
13: until \( F_k = \emptyset \)
14: Result = \( \bigcup F_k \).

Generating Candidates $C_k$

1. **Self-joining**: Find pairs of sets in $L_{k-1}$ that differ by **one** element

2. **Pruning**: Remove all candidates with infrequent subsets
Example: Generating Candidates $C_k$

$B_1 = \{m, c, b\}$  \hspace{1cm}  $B_2 = \{m, p, j\}$
$B_3 = \{m, b\}$  \hspace{1cm}  $B_4 = \{c, j\}$
$B_5 = \{m, c, b\}$  \hspace{1cm}  $B_6 = \{m, c, b, j\}$
$B_7 = \{c, b, j\}$  \hspace{1cm}  $B_8 = \{b, c\}$

- **Frequent itemsets of size 2:**
  \{m,b\}:4, \{m,c\}:3, \{c,b\}:5, \{c,j\}:3

- **Self-joining:**
  \{m,b,c\}, \{b,c,j\}

- **Pruning:**
  \{b,c,j\} since \{b,j\} not frequent
Compacting the Output

• To reduce the number of rules we can post-process them and only output:
  
  • **Maximal frequent itemsets:**
    No immediate superset is frequent
    
    • Gives more pruning
  
  • **Closed itemsets:**
    No immediate superset has same count (> 0)
    
    • Stores not only frequent information, but exact counts
Example: **Maximal vs Closed**

\[
B_1 = \{m, c, b\} \quad B_2 = \{m, p, j\}
\]
\[
B_3 = \{m, b\} \quad B_4 = \{c, j\}
\]
\[
B_5 = \{m, c, b\} \quad B_6 = \{m, c, b, j\}
\]
\[
B_7 = \{c, b, j\} \quad B_8 = \{b, c\}
\]

**Frequent itemsets:**

- \(\{m\} : 5, \{c\} : 6, \{b\} : 6, \{j\} : 4\)  (Closed)
- \(\{m, c\} : 3, \{m, b\} : 4, \{c, b\} : 5, \{c, j\} : 3\)  (Maximal)
- \(\{m, c, b\} : 3\)
Example: Maximal vs Closed

- Maximal Frequent Itemsets
- Closed Frequent Itemsets
- Frequent Itemsets
Given a transaction \( t \), what are the possible subsets of size 3?

Transaction, \( t \) (items are sorted)

Level 1

Level 2

Level 3

Subsets of 3 items

Hash Tree for Itemsets

Hash Function

Candidate Hash Tree

15 candidate 3-itemsets, distributed across 9 leaf nodes

15 candidate 3-itemsets, distributed across 9 leaf nodes

Hash Tree for Itemsets

15 candidate 3-itemsets, distributed across 9 leaf nodes

Subset Operation

Subset Operation Using Hash Tree

1 5 9
1 4 5
1 3 6
3 4 5
3 6 7
3 6 8
3 5 6
3 5 7
6 8 9
2 3 4
5 6 7
1 2 4
4 5 7
1 2 5
4 5 8
1 2 3 5 6
1 + 2 3 5 6
2 + 3 5 6
3 + 5 6
1, 4, 7
2, 5, 8
3, 6, 9

Hash Function

transaction

Subset Operation

Hash Function

transaction

Subset Operation

Match transaction against 11 out of 15 candidates

Apriori: Bottlenecks

1. Set $k = 0$
2. Define $C_1$ as all size 1 item sets
3. **While $C_{k+1}$ is not empty**
4. Set $k = k + 1$
5. Scan DB to determine subset $L_k \subseteq C_k$ with support $\geq s$
6. Construct candidates $C_{k+1}$ by combining sets in $L_k$ that differ by 1 element

(I/O limited) (Memory limited)
Apriori: Bottlenecks

1. Set $k = 0$
2. Define $C_1$ as all size 1 item sets
3. **While $C_{k+1}$ is not empty**
4. Set $k = k + 1$
5. **Scan DB to determine subset $L_k \subseteq C_k$** (I/O limited) with support $\geq s$
6. **Construct candidates $C_{k+1}$ by combining sets in $L_k$ that differ by 1 element** (Memory limited)
FP-Growth Algorithm – Overview

• Apriori requires one pass for each $k$ (2+ on first pass for PCY variants)
• Can we find all frequent item sets in fewer passes over the data?

FP-Growth Algorithm:

• *Pass 1*: Count items with support $\geq s$
• Sort frequent items in descending order according to count
• *Pass 2*: Store all frequent itemsets in a frequent pattern tree (FP-tree)
• Mine patterns from FP-Tree
FP-Tree Construction

Transaction ID 1

null

a: 1

b: 1

Transaction ID 2

null

a: 1

b: 1

c: 1

d: 1

e: 1

Transaction ID 3

null

a: 2

b: 1

c: 1

d: 1

e: 1

Transaction ID 10

null

a: 8

b: 2

c: 2

d: 5

e: 1

f: 1

g: 1

h: 1

m: 1

n: 1

Mining Patterns from the FP-Tree

**Step 1: Extract subtrees ending in each item**

**Full Tree**

**Subtree e**

**Subtree d**

**Subtree c**

**Subtree b**

**Subtree a**

a: 8, b: 7, c: 6, d: 5, e: 3, f: 1, g: 1, h: 1, m: 1, n: 1

Mining Patterns from the FP-Tree

Step 2: Construct Conditional FP-Tree for each item

Conditional Pattern Base for e
acd: 1, ad: 1, bc: 1

Conditional Node Counts
a: 2, b: 1, c: 2, d: 2

- Calculate counts for paths ending in e
- Remove leaf nodes
- Prune nodes with count ≤ s

Mining Patterns from the FP-Tree

Step 3: Recursively mine conditional FP-Tree for each item

Conditional e

Subtree de

Conditional de

Subtree ce

Subtree ae

Mining Patterns from the FP-Tree

### Conditional Pattern Base

<table>
<thead>
<tr>
<th>Suffix</th>
<th>Conditional Pattern Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>acd:1; ad:1; bc:1</td>
</tr>
<tr>
<td>d</td>
<td>abc:1; ab:1; ac:1; a:1; bc:1</td>
</tr>
<tr>
<td>c</td>
<td>ab:3; a:1; b:2</td>
</tr>
<tr>
<td>b</td>
<td>a:5</td>
</tr>
<tr>
<td>a</td>
<td>ϕ</td>
</tr>
</tbody>
</table>

### Frequent Itemsets

<table>
<thead>
<tr>
<th>Suffix</th>
<th>Frequent Itemsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>{e}, {d,e}, {a,d,e}, {c,e}, {a,e}</td>
</tr>
<tr>
<td>d</td>
<td>{d}, {c,d}, {b,c,d}, {a,c,d}, {b,d}, {a,b,d}, {a,d}</td>
</tr>
<tr>
<td>c</td>
<td>{c}, {b,c}, {a,b,c}, {a,c}</td>
</tr>
<tr>
<td>b</td>
<td>{b}, {a,b}</td>
</tr>
<tr>
<td>a</td>
<td>{a}</td>
</tr>
</tbody>
</table>

FP-Growth vs Apriori

Simulated data 10k baskets, 25 items on average

(from: Han, Kamber & Pei, Chapter 6)
FP-Growth vs Apriori

<table>
<thead>
<tr>
<th>File</th>
<th>Apriori</th>
<th>FP-Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Market Basket test file</td>
<td>3.66 s</td>
<td>3.03 s</td>
</tr>
<tr>
<td>&quot;Real&quot; test file (1 Mb)</td>
<td>8.87 s</td>
<td>3.25 s</td>
</tr>
<tr>
<td>&quot;Real&quot; test file (20 Mb)</td>
<td>34 m</td>
<td>5.07 s</td>
</tr>
<tr>
<td>Whole &quot;real&quot; test file (86 Mb)</td>
<td>4+ hours (Never finished, crashed)</td>
<td>8.82 s</td>
</tr>
</tbody>
</table>

http://singularities.com/blog/2015/08/apriori-vs-fpgrowth-for-frequent-item-set-mining
FP-Growth vs Apriori

Advantages of FP-Growth

• Only 2 passes over dataset
• Stores “compact” version of dataset
• No candidate generation
• Faster than A-priori

Disadvantages of FP-Growth

• The FP-Tree may not be “compact” enough to fit in memory

Used in practice: PFP (a distributed version of FP-growth)
Exploratory Data Analysis (demo)
Counting in the Shell

```
grep '^#c' publications.txt \ | sed 's/^#c//\n | uniq -c | sort -nr \ > venue_counts.txt
```

46993 CoRR
13835 IEICE Transactions
13260 ICRA
10978 Discrete Mathematics
...

Counting in the Shell

awk 'BEGIN {sum=0} {sum=sum+$1; print sum}'\nvenue_counts.txt  > venue_cumsum.txt

46993
60828
74088
85066
...

...
Spinning up Jupyter (Docker)

docker run -it --rm -p 8888:8888 \ 
-v "$PWD":/home/jovyan/work \ 
jupyter/pyspark-notebook

[I 15:59:40.962 NotebookApp] The Jupyter Notebook is running at: http://[all ip addresses on your system]:8888/?token=90c08be4b2cecb020965c0fe7160049b56412869f7f5f5f8

[I 15:59:40.962 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[C 15:59:40.962 NotebookApp] Copy/paste this URL into your browser when you connect for the first time, to login with a token:

http://localhost:8888/?token=90c08be4b2cecb020965c0fe7160049b56412869f7f5f5f8
Spinning up Jupyter (Docker)

```python
In [2]:
    import numpy as np
    import matplotlib.pyplot as plt
    plt.style.use('fivethirtyeight')
    %matplotlib inline

In [4]:
    pub_cumsum = np.loadtxt('venue_cumsum.txt')

In [5]:
    plt.plot(pub_cumsum)
    plt.xlabel('number of publication venues')
    plt.ylabel('number of publications')

Out[5]: <matplotlib.text.Text at 0x7f6a9fa91f28>
```
Counting with Spark

In [1]:
   import pyspark
   from pyspark import SparkContext
   from pyspark.mllib.fpm import FPGrowth
   SparkContext.setSystemProperty('spark.executor.memory', '2g')
   sc = pyspark.SparkContext('local[*]')

In [2]:
   import re
   rdd = sc.textFile('publications.txt')
   small_rdd = rdd.sample(False, 1e-3)
   venue_rdd = small_rdd.filter(
       lambda l: re.match('^#c(.*[^c]*c[.]*)', l).map(
           lambda l: re.match('^#c(.*[^c]*c[.]*)', l).group(1))
   venue_counts = venue_rdd.countByValue()

In [4]:
   svc = sorted(venue_counts, key=venue_counts.get, reverse=True)
   for v in svc[:5]:
       print(v, venue_counts[v])

CoRR 57
IEICE Transactions 20
ICRA 14
Discrete Mathematics 13
IEEE Transactions on Information Theory 12