Lecture 19: Association rules

Ramin Zabih

Some content from: Wikipedia/Google image search; Harrington;
Lecture Outline

• From last time: course grades, SimHash
• From supervised to unsupervised learning
• Some useful logical identities
• Frequent item set data mining
• The Apriori algorithm
Course grades

• The below is **NOT** a promise, just an educated guess
• Typically in a graduate course like CS5112, most students get some kind of an A or B
Angle similarity via SimHash

• Angle similarity via projection onto random vector
  – VERY important for machine learning, etc.

• Pick a random unit vector $r$, and check if the two inputs $x, y$ are on the same side of the half-space it defines

• Compute the dot products $\langle x, r \rangle, \langle y, r \rangle$
  – Do they have the same sign?
Dot product and hyperplanes

• For simplicity only consider vectors from the origin
• A vector \( \mathbf{v} \) defines a hyperplane of vectors perpendicular to \( \mathbf{v} \)
  – I.e., those vectors \( \mathbf{w} \) \( \langle \mathbf{v}, \mathbf{w} \rangle = 0 \)
• Divides vectors into those on either side of the hyperplane
  – Same side as \( \mathbf{v} \): \( \mathbf{w} | \langle \mathbf{v}, \mathbf{w} \rangle > 0 \) so hash value is +1
  – Opposite side of \( \mathbf{v} \): \( \mathbf{w} | \langle \mathbf{v}, \mathbf{w} \rangle < 0 \) so hash value is 0
• Easy to draw the 2D case
A bad LSH function and how to fix it

• This gives us a single bit per vector
• Which generates a really lousy LSH hash function
  – It only has 2 buckets!
• What goes wrong and how do we fix it?
• Same slice of the pizza!
2D case of SimHash

$v = 1$
Unsupervised learning

• What interesting things can we learn in the absence of a labeled data set?

• Labeled data is expensive
  – Semi-supervised learning

• Main unsupervised areas are:
  – Clusters (see: k-means algorithm)
  – Low dimensional structure (not covered in CS5112)
  – Associations (today’s lecture)
Useful logical identities

- Consider true/false propositions $p, q, r, ...$
- The below can be proved by, e.g. truth tables

\[(p \Rightarrow q) \equiv (\neg p \lor q) \equiv (\neg q \Rightarrow \neg p)\]

\[(p \land q \Rightarrow r) \equiv (p \land \neg r \Rightarrow \neg q)\]

\[(p \Rightarrow q \land r) \vdash (p \Rightarrow q)\]
Example transactions

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

- Rule discovered: Coke → Diaper
Things can go badly wrong...
Association rules

- Learn rules that are supported by your data
- Rules are co-occurrence, not causality!
  - Very clear in the propositional formulation
- Beer and diapers legend
  - What do you do with an association rule?
- In practice you don’t want too many of them
  - Need to act on them
Support and confidence

• Key ideas for association rules
• Have both a computational and probabilistic interpretation
• Support of an itemset is the percentage of the transactions containing that itemset
  – In our example, support of Milk is $\frac{4}{5} = .8$
  – Support of a rule is the support of LHS
    • Not all papers use this definition, sometimes it’s the support of LHS $\cup$ RHS
• Confidence of an association rule is percentage of transactions where that rule is correct
  – Confidence of Milk $\rightarrow$ Bread is $\frac{3}{4} = .75$
Probabilistic view

• “The basket contains beer” can be viewed as a proposition $p$, or as a 0/1 random variable
• Consider rule: $p$ generally follows from $q \land r$
• Can be viewed as the idea that $P(p|q, r)$ is large
  – Support is joint probability $P(p, q, r)$
• Confidence is conditional probability $P(p|q, r)$
  – Note that $P(p, q, r) = P(p|q, r)P(q, r)$
Association rule learning

• All rules with support $\geq s$ and confidence $\geq c$

• We focus on finding sets with large support
  – Called frequent (item) sets

• Many rules from same item set, different $c$

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bread, Milk</td>
<td>{Milk,Diaper} $\rightarrow$ {Beer} ($s=0.6, c=0.67$)</td>
</tr>
<tr>
<td>2</td>
<td>Bread, Diaper, Beer, Eggs</td>
<td>{Milk,Beer} $\rightarrow$ {Diaper} ($s=0.4, c=1.0$)</td>
</tr>
<tr>
<td>3</td>
<td>Milk, Diaper, Beer, Coke</td>
<td>{Diaper,Beer} $\rightarrow$ {Milk} ($s=0.6, c=0.67$)</td>
</tr>
<tr>
<td>4</td>
<td>Bread, Milk, Diaper, Beer</td>
<td>{Beer} $\rightarrow$ {Milk,Diaper} ($s=0.6, c=0.67$)</td>
</tr>
<tr>
<td>5</td>
<td>Bread, Milk, Diaper, Coke</td>
<td>{Diaper} $\rightarrow$ {Milk,Beer} ($s=0.6, c=0.67$)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{Milk} $\rightarrow$ {Diaper,Beer} ($s=0.8, c=0.5$)</td>
</tr>
</tbody>
</table>
Beyond confidence

• Sometimes other measures are useful
• Motivating example:

<table>
<thead>
<tr>
<th></th>
<th>Coffee</th>
<th>¬Coffee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tea</td>
<td>15</td>
<td>5</td>
</tr>
<tr>
<td>¬Tea</td>
<td>75</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>10</td>
</tr>
</tbody>
</table>

• $c = P(\text{Coffee}|\text{Tea}) = 0.75$
  – But $P(\text{Coffee}) = 0.9$
  – And $P(\text{Coffee}|¬\text{Tea}) = 0.9375$

• **Lift** is one solution: $\frac{P(\text{Coffee}|\text{Tea})}{P(\text{Coffee})} = \frac{0.75}{0.9} < 1$
PB&J example

- Item set is \( \{P, J, B\} \)
- Consider the rule \( \{P, J\} \rightarrow B \)
- Support of 0.03 for LHS means \( P, J \) in 3% of transactions
- Confidence of 0.82 for rule means 82% of transactions that purchase \( P, J \) also purchase \( B \)
- If \( B \) had support of 43% then the rule has a lift of 1.95
Fields of sets

• Consider a set with $n$ elements
• We can arrange all of its $2^n$ subsets into a lattice
  – Via union and intersection
• This structure is called a field of sets
Example

Figure 11.2  All possible itemsets from the available set {0, 1, 2, 3}

Harrington, *Machine Learning in Action*
The A Priori Principle

- Problem: exponentially many item sets
- As we grow an item set, its support goes down
- If an item set is frequent, all of its subsets are frequent
- If an item set is infrequent, all of its supersets are infrequent
Example

Figure 11.3  All possible itemsets shown, with infrequent itemsets shaded in gray. With the knowledge that the set \{2,3\} is infrequent, we can deduce that \{0,2,3\}, \{1,2,3\}, and \{0,1,2,3\} are also infrequent, and we don’t need to compute their support.

Harrington, *Machine Learning in Action*
Apriori algorithm

- Given a support threshold and a set of transactions
- Find frequent single items
- To go from frequent $k$ tuples to frequent $k + 1$ tuples, combine with frequent single items for candidates
  - Ex: from 2-tuples to 3-tuples
- Stop when no more frequent tuples
From frequent item sets to rules

• In bricks and mortar situations, usually require about 1% support and 50% confidence

• Given a frequent item set with \( k \) elements, there are \( k - 1 \) logically equivalent rules
  – Of the form \( p_1 \land p_2 \land \cdots p_{k-1} \Rightarrow p_k \)

• We know that the LHS is frequent, so we can easily calculate the confidence of this rule
Apriori plus and minus

- Plus: Fast, runs on huge data sets, easy to interpret
- Rules with high confidence but low support are missed
  - Classic example: vodka → caviar
Extension: PCY algorithm

• Park-Chen-Yu speedup of apriori
• Use a hash table to store counts of pairs
• Hash on the pair
• Collisions: may think something is frequent even if it is not
  – But you can use hashing to eliminate a ton of computation
• What does this remind you of?