# CS5112: Algorithms and Data Structures for Applications

## Association rules

Ramin Zabih

Cornell University

CORNELL TECH
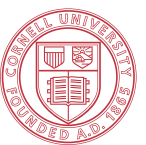
# Administrivia

- Prelim (midterm) date scheduled for Wednesday October 16
  - This would ruin your holiday weekend, so we're delaying it a week
  - In class, closed book on Wednesday October 23
  - Review session in class on October 21
- HW1 is due by 10/7
  - We will be more forceful about groups in the future
- Lectures will be recorded "Real Soon Now"

CORNELL TECH

# Lecture Outline

- Unsupervised learning and association rule mining

- Some logical identities

- Useful ways to think about (discrete) probabilities

- Frequent item set data mining

- The Apriori algorithm

# Learning from big data

- The term is ill-defined, but lots of important examples
  - Fraud detection in credit card transactions
  - Loyalty programs
  - Predicting consumer preferences
    - And exploiting them?
- Primarily rely on labeled data
  - "Supervised learning"
- Labeled data is **expensive**

# Unsupervised learning

- What can we learn in the absence of a labeled data set?

- Main unsupervised areas are:
  - Clustering (e.g., k-means)
  - Low dimensional structure (not covered in CS5112)
  - Associations (today's lecture)

- Digression: semi-supervised learning
  - Use clusters to propagate a sparse set of labels

# Useful logical identities

- Consider true/false propositions $p, q, r, \ldots$
- 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

| TID | Items |
|-----|-------|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

- Rule discovered: Coke$\rightarrow$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 (symmetry)

- 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

CORNELL
TECH

# Support and confidence

| TID | Items |
|-----|-------|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

- 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

  - Support of a rule is the support of LHS
    - Not all papers use this definition, sometimes it's the support of LHS ∪ RHS

- **Confidence** of an association rule is percentage of transactions where that rule is correct

  - Confidence of Milk→Bread is

**CORNELL TECH**

# 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 $q \wedge r$

- 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 $P(p,q,r) = P(p|q,r)P(q,r)$
  - Symmetry: $P(p,q,r) = P(r|p,q)P(p,q) = P(q|p,r)P(p,r)$

# Possible worlds

|  | Coffee | ¬Coffee |  |
|---|---|---|---|
| Tea | 15 | 5 | 20 |
| ¬Tea | 75 | 5 | 80 |
|  | 90 | 10 | 100 |

- Each possible world (= outcome) has some probability
  - This assigns probabilities to subset of possible worlds (= events)
    - Just add up their probabilities
  - Note: works well in the discrete case (only!)
- Joint probability $P(p, q, r)$ is the probability of the possible worlds where all 3 propositions are true
- Conditional probability $P(p|q, r)$ is the probability of the possible worlds where $p$ is true
  - Normalized by only considering possible worlds where $q, r$ are true

CORNELL
TECH

# Association rule learning

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

- We focus on finding sets with large support (why?)
  - Called **frequent (item) sets**

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

| TID | Items |
|-----|-------|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

{Milk,Diaper} $\rightarrow$ {Beer} ($s$=0.6, $c$=0.67)
{Milk,Beer} $\rightarrow$ {Diaper} (s=0.4, $c$=1.0)
{Diaper,Beer} $\rightarrow$ {Milk} (s=0.6, $c$=0.67)
{Beer} $\rightarrow$ {Milk,Diaper} (s=0.6, $c$=0.67)
{Diaper} $\rightarrow$ {Milk,Beer} ($s$=0.8, $c$=0.5)
{Milk} $\rightarrow$ {Diaper,Beer} ($s$=0.8, $c$=0.5)

# Beyond confidence

- Sometimes other measures are useful
- Motivating example:

| | Coffee | ¬Coffee | |
|---|---|---|---|
| Tea | 15 | 5 | 20 |
| ¬Tea | 75 | 5 | 80 |
| | 90 | 10 | 100 |

- $c = P(\text{Coffee}|\text{Tea}) = 0.75$
  - But $P(\text{Coffee}) = 0.9$
  - And $P(\text{Coffee}|\neg\text{Tea}) = 0.9375$

- **Lift** is one solution: $\dfrac{P(\text{Coffee}|\text{Tea})}{P(\text{Coffee})} = \dfrac{0.75}{0.9} < 1$

CORNELL TECH

# Lift and identities

- Formal definition: $\text{lift}(p{\rightarrow}q) = \dfrac{P(q|p)}{P(q)}$

  - Intuitively: how much more or less common is $q$ in the possible worlds where $p$ is true, compared to in general

    - Does buying $p$ make you more or less likely to buy $q$?

  - When lift=1 the two items being purchased are independent

  - When lift<1 the items are substitutes

  - When lift>1 then the items complement each other

- Note: $\text{lift}(p{\rightarrow}q) = \text{lift}(q{\rightarrow}p)$ (because: algebra)

# 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
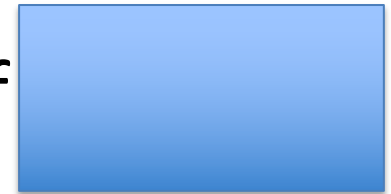
# Fields of sets

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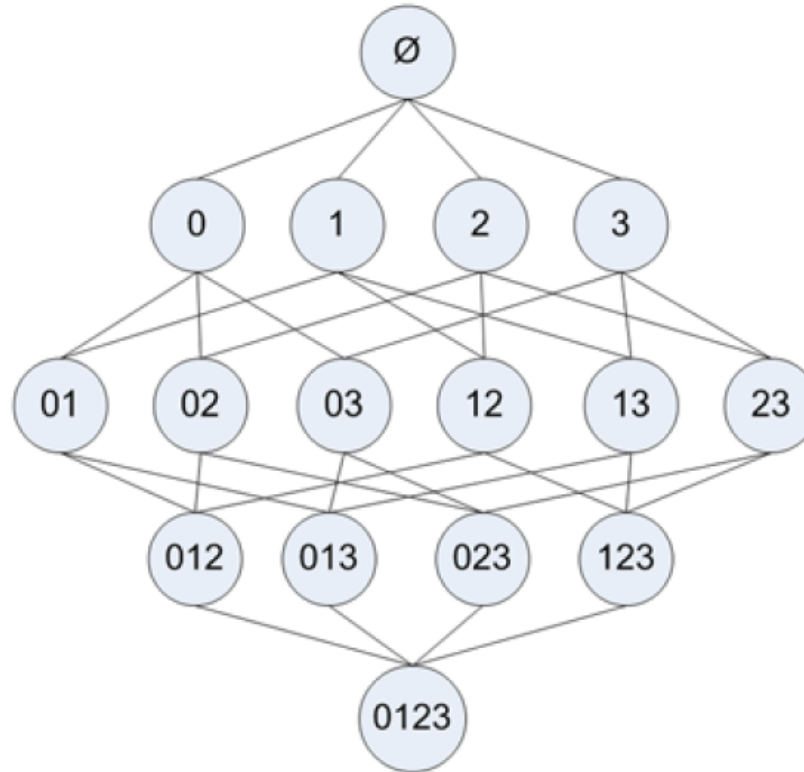- This structure is called a field of sets

CORNELL
TECH

# 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
  - How could beer and diapers be popular, if beer were not popular?

- If an item set is infrequent, all of its supersets are infrequent
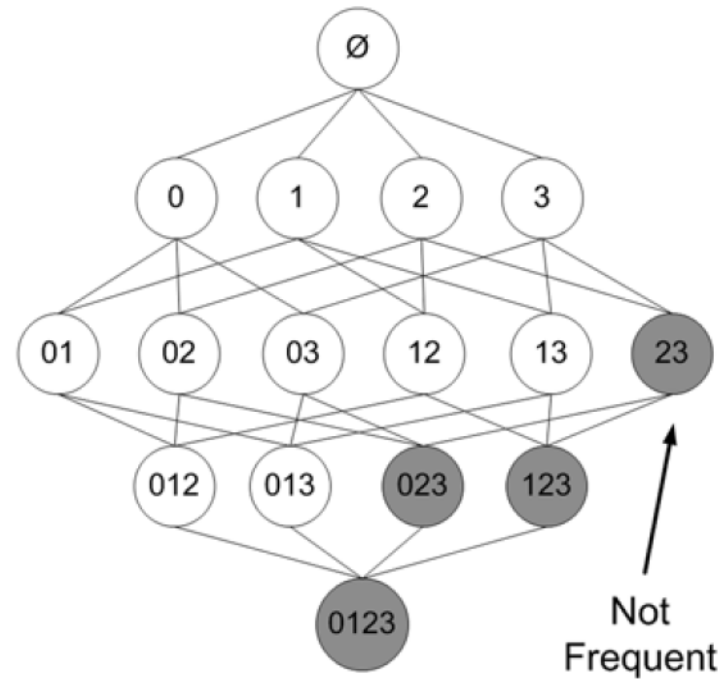  - If beer and coke is infrequent, so is beer and coke and diapers

# 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 (possibly) frequent $k + 1$ tuples, combine $k$-tuples with frequent single items
  - Ex: from 2-tuples to 3-tuples
  - Combining a frequent single item and a frequent 2-tuple can lead to an infrequent 3-tuple
- 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 $p_1 \wedge p_2 \wedge \cdots p_{k-1} \Rightarrow p_k$

- 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 $\rightarrow$ caviar
- Very widely cited paper, many algorithmic variants
  - Ex: a transaction with no frequent itemsets can be ignored later
  - Including, for example, dynamic programming!