Data Mining & Machine Learning

CS37300
Purdue University

Nov 10, 2023

Pattern discovery

- Models describe entire dataset (or large part of it)
- Pattern characterize local aspects of data
- Pattern: predicate that returns “true” for the instances in the data where the pattern occurs and “false” otherwise
- Task: find descriptive associations between variables

Pattern in tabular data

- Primitive pattern: subset of all possible observations over variables $X_1,\ldots,X_d$
  - If X_k is categorical then $X_k = c$ is a primitive pattern
  - If X_k is ordinal then $X_k \leq c$ is a primitive pattern
- Start from primitive patterns and combine using logical connectives such as AND and OR
  - age<40 AND income<100,000
  - chips=1 AND (beer=1 OR soda=1)

Pattern space

- Set of legal patterns; defined through set of primitive patterns and operators to combine primitives
  - Example: If variable $X_1,\ldots,X_d$ are all binary we can define the space of patterns to be all conjunctions of the form
    $(X_{i1}=1)$ AND $(X_{i2}=1)$ AND ... AND $(X_{ik}=1)$

- Typically there is a generalization/specialization relationship between patterns
  - Pattern α is **more general** than pattern β, if whenever β occurs, α occurs as well. This also means that pattern β is **more specific** than pattern α
  - Examples:
    age<40 AND income<100,000 is more ***specific*** *than age<40*
    chips=1 is more ***general*** *than chips=1 AND (beer=1 OR soda=1)*
  - This property is used during search

Pattern discovery task

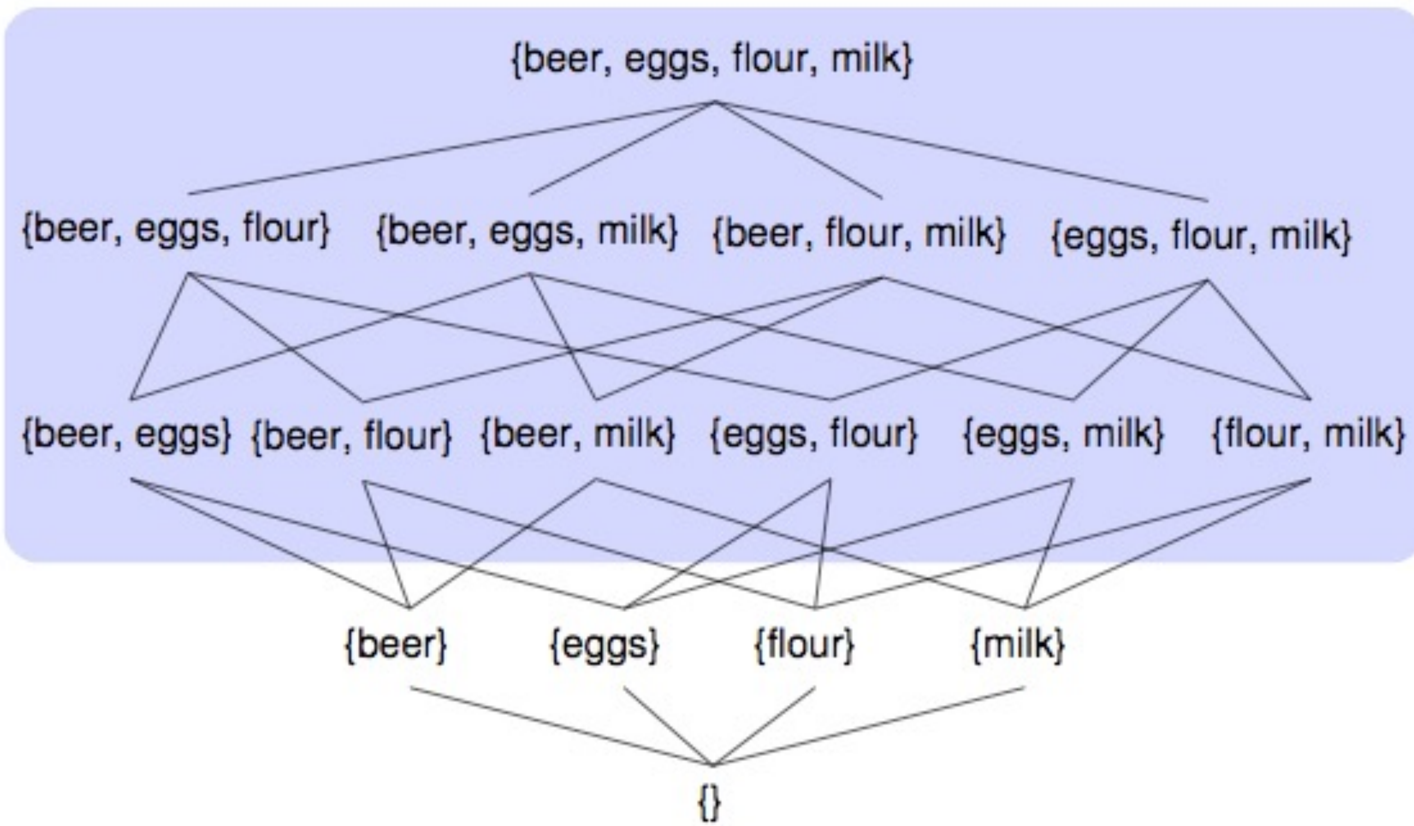
- Find all “interesting” patterns in the data
- Challenge: find the right balance between
  - Pattern complexity
  - Pattern impact
  - Computational complexity

Search problem

- Searching for all patterns is computationally intractable
- Consider market basket data where each row has 1000 binary variables
- How many possible patterns?
  - How many unique transactions?
    2^{1000}
  - How many subsets (patterns)?
    $2^{\wedge}(2^{1000})$

Transaction ID	beer	eggs	flour	milk
1	0	1	1	1
2	1	1	0	0
3	0	1	0	1
4	0	1	1	1
5	0	0	0	1

Example: lattice of itemsets (general to specific)



Solution

- Take advantage of the nature of the score function to prune parts of the search space and reduce run time
- What is the score function?
    - Patterns that occur frequently are often of interest.. thus score function often involves *frequency*

Finding frequent itemsets

- Find sets of items:
  - with large "support" i.e. patterns that occur with higher-than-threshold frequency or,
  - large "confidence" i.e. precision of rules is higher.
- Support is ***monotonic***
  - A subset of a frequent itemset must also be frequent
  - If {A,B} is a frequent itemset then both {A} and {B} are frequent itemsets as well
- **The Apriori principle**:
  - Iteratively find frequent itemsets with cardinality from 1 to k (k-itemset)
  - Prune any sets of size k that are not frequent

Apriori algorithm

- Classic algorithm for learning association rules that uses ***apriori principle*** to search efficiently for rules that meet support and confidence thresholds

- Given a pruned list of candidate frequent sets of size k
  - Algorithm performs a linear scan of the data to determine which of these sets are frequent
- Confirmed frequent sets of size k are combined to generate possible frequent sets of size k+1
  - Followed by another pruning step
  - Cardinality of largest frequent set is quite small for large support values
- Use frequent itemsets to form association rules

Sequential pattern mining

- **Task**:
    - Find frequent substring patterns
- **Data**:
    - Sequence data, biological data, text collections
- **Applications**:
    - Bioinformatics, text analysis, clustering, classification

Sequential patterns

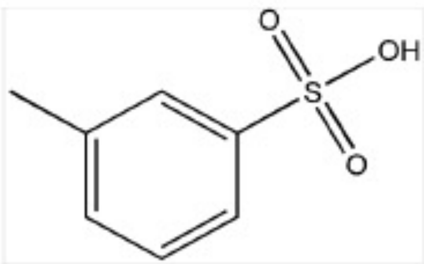
- Substring
- Regular expression
- Episode
  - Partially ordered collection of events occurring together
  - Can take time or sequential ordering into account but is insensitive to intervening events
  - For instance, headache followed by sense of disorientation within a given time period

Graph mining

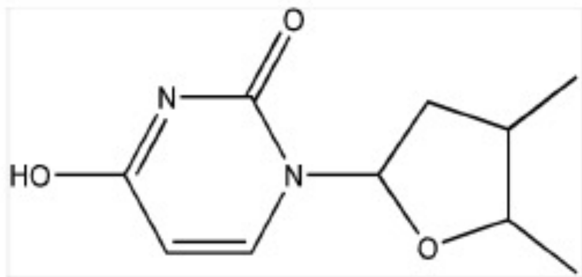
- **Task**:
  - Find frequent subgraph patterns
- **Data**:
  - Graph databases or relational databases
- **Applications**:
  - Graph indexing, similarity search, clustering, classification

Subgraphs

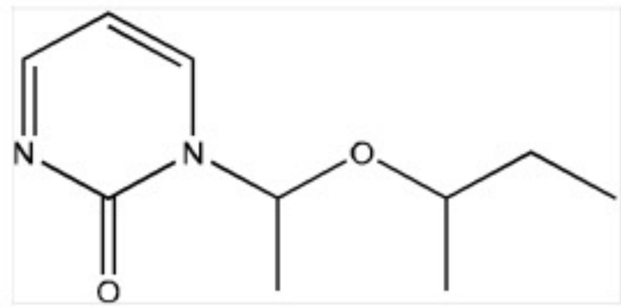
GRAPH DATASET



(A)



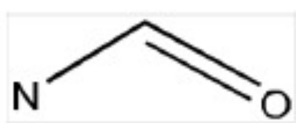
(B)



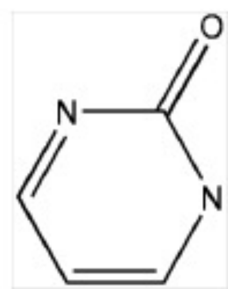
(C)

FREQUENT PATTERNS
(MIN SUPPORT IS 2)

(1)



(2)



Association rule mining

- **Task**:
  - Find frequent patterns, associations, correlations, or causal structures among items
- **Data**:
  - Transaction databases, relational database, or other information repositories
- **Applications**:
  - Market basket analysis, cross-marketing, catalog design, loss-leader analysis, clustering, classification

Rule

- A rule is an expression of the form $\theta \rightarrow \phi$
- Association rules:
  - All variables are binary
  - Probabilistic statement about the co-occurrence of certain events in the database
- Mining rules
  - Number of rules grows exponentially with dimensions
  - How to find patterns in an efficient manner?
  - How to determine which rules are interesting?

Association rules

- Data
  - Basket: customer transaction; items: products
  - Basket: document; items: words
  - Basket: web pages; items: links
- Find all rules of the form $\theta \rightarrow \phi$ that satisfy the following constraints:
  - Support of the rule is greater than threshold *s*
  - Confidence of the rule is greater than threshold *c*
  - For instance, 98% of people who purchase tires and auto accessories also have automotive service done

Rule evaluation

- **Support** (also known as frequency)
  - $s(\theta \rightarrow \phi) = fr(\theta \wedge \phi)$
  - Number of samples which have antecedent θ and consequent ϕ, divided by total number of samples.
- **Confidence** (also known as accuracy)
  - $c(\theta \rightarrow \phi) = p(\phi \mid \theta) = fr(\theta \wedge \phi) \,/\, fr(\theta)$
  - Number of samples which have antecedent θ and consequent ϕ, divided by number of samples which have antecedent θ.

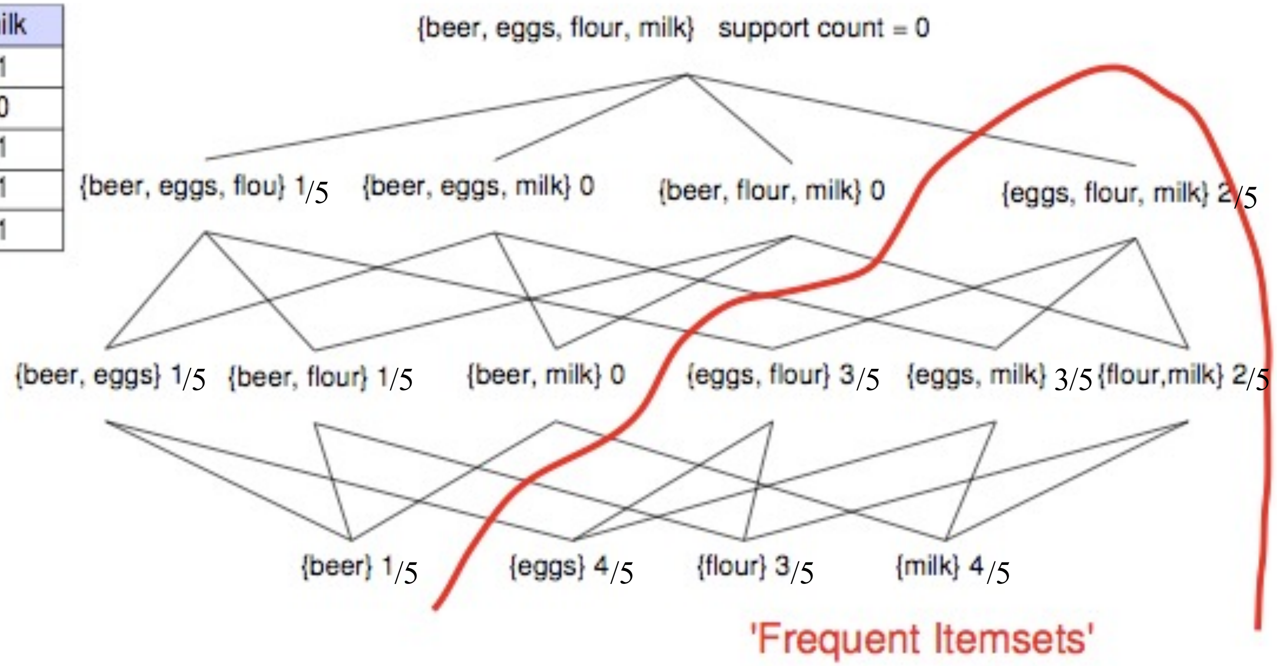
Rule evaluation

- **Support** (also known as frequency)
  - $s(\theta \rightarrow \phi) = fr(\theta \wedge \phi)$
  - Number of samples which have antecedent θ and consequent ϕ, divided by total number of samples.
- **Confidence** (also known as accuracy)
  - $c(\theta \rightarrow \phi) = p(\phi \mid \theta) = fr(\theta \wedge \phi) / fr(\theta)$
  - Number of samples which have antecedent θ and consequent ϕ, divided by number of samples which have antecedent θ.
- Which has higher confidence?
  - 1. flour → eggs
  - 2. eggs → flour

Transaction ID	beer	eggs	flour	milk
1	0	1	1	1
2	1	1	1	0
3	0	1	0	1
4	0	1	1	1
5	0	0	0	1

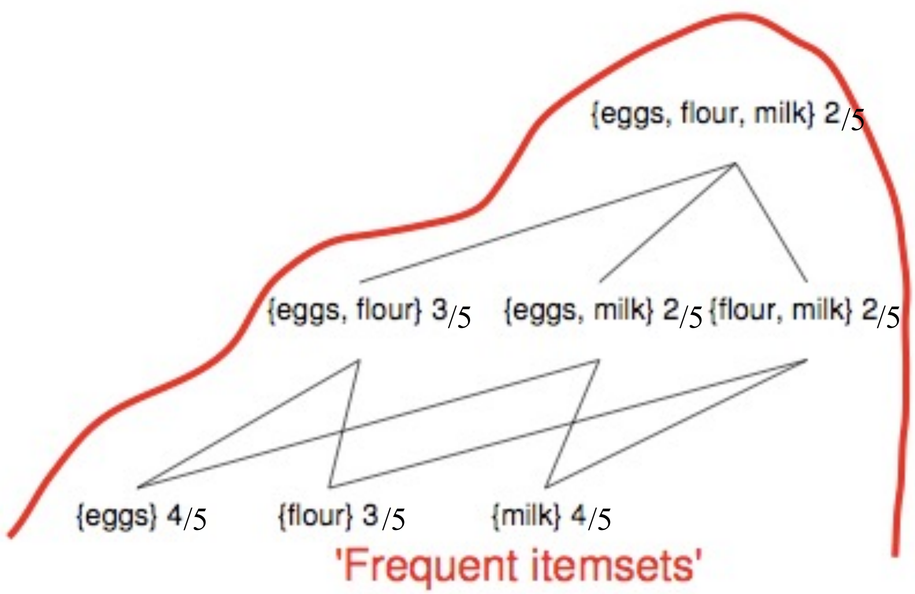
Example

Transaction ID	beer	eggs	flour	milk
1	0	1	1	1
2	1	1	1	0
3	0	1	0	1
4	0	1	1	1
5	0	0	0	1



support threshold = 0.3

Example



			Confidence
{eggs}	→	{flour}	$3/4 = 0.75$
{flour}	→	{eggs}	$3/3 = 1$
{eggs}	→	{milk}	$2/4 = 0.5$
{milk}	→	{eggs}	$2/4 = 0.5$
{flour}	→	{milk}	$2/3 = 0.67$
{milk}	→	{flour}	$2/4 = 0.5$
{eggs, flour}	→	{milk}	$2/3 = 0.67$
{eggs, milk}	→	{flour}	$2/2 = 1$
{flour, milk}	→	{eggs}	$2/2 = 1$
{eggs}	→	{flour, milk}	$2/4 = 0.5$
{flour}	→	{eggs, milk}	$2/3 = 0.67$
{milk}	→	{eggs, flour}	$2/4 = 0.5$

Association rules

- Knowledge representation?
  - **If-then rules**
- Score function?
  - **Support, confidence**
- Search?
  - **Exhaustive search**
    Returns all rules that exceed support and confidence thresholds, pruning (based on apriori principle) is used to eliminate portions of search space
- Optimal?
  - Yes. Guaranteed to find all rules that exceed specified thresholds.

Evaluation

- Association rules algorithms usually return many, many rules
  - Many are uninteresting or redundant
    (For instance, ABC→D and AB→D may have same support and confidence)
- How to quantify interestingness?
  - Objective: statistical measures
  - Subjective: *unexpected* and/or *actionable* patterns
    (requires domain knowledge)

Drawback of support

- Support suffers from the **rare item problem** (Liu et al.,1999)
    - Infrequent items not meeting minimum support are ignored which is problematic if rare items are important
    - For instance, rarely sold products which account for a large part of revenue or profit
- Support falls rapidly with itemset size. A threshold on support favors short itemsets

Drawback of confidence

	Coffee	$\overline{\text{Coffee}}$	
Tea	15	5	20
$\overline{\text{Tea}}$	75	5	80
	90	10	100

Association Rule: Tea → Coffee

Confidence = P(Coffee|Tea) = 15/20 = 0.75, which is high

but rule is misleading since P(Coffee|$\overline{\text{Tea}}$) = 75 / 80 = 0.9375

Statistical-based measures

- Lift: $$\frac{P(Y \mid X)}{P(Y)}$$

- Piatetsky-Shapiro: $$P(X,Y) - P(X)P(Y)$$

- ϕ-coefficient: $$\frac{P(X,Y) - P(X)P(Y)}{\sqrt{P(X)(1-P(X))P(Y)(1-P(Y))}}$$

Lift example

	Coffee	$\overline{\text{Coffee}}$	
Tea	15	5	20
$\overline{\text{Tea}}$	75	5	80
	90	10	100

Association Rule: Tea $\rightarrow$ Coffee

Confidence = P(Coffee|Tea) = 15/20 = 0.75, which is high

but rule is misleading since P(Coffee|$\overline{\text{Tea}}$) = 75 / 80 = 0.9375

P(Coffee) = 0.9

Lift = P(Coffee|Tea)/P(Coffee) = 0.75 / 0.9 = 0.8333 < 1 (negatively associated)

Lift example 2

	Coffee	$\overline{\text{Coffee}}$	
Tea	15	5	20
$\overline{\text{Tea}}$	75	5	80
	90	10	100

Association Rule: $\overline{\text{Tea}} \rightarrow \text{Coffee}$

Confidence = $P(\text{Coffee}|\overline{\text{Tea}})$ = 75 / 80 = 0.9375

P(Coffee) = 0.9

Lift = $P(\text{Coffee}|\overline{\text{Tea}})/P(\text{Coffee})$ = 0.9375 / 0.9 = 1.0417 > 1 (positively associated)