

Data Mining

- The Extraction of useful information from data
- The automated extraction of hidden predictive information from (large) databases
- Business, Huge data bases, customer data, mine the data
 - Also Medical, Genetic, Astronomy, etc.
- Data sometimes unlabeled – unsupervised clustering, etc.
- Focuses on learning approaches which scale to massive amounts of data
 - and potentially to a large number of features
 - sometimes requires simpler algorithms with lower big-O complexities

Data Mining Applications

- Often seeks to give businesses a competitive advantage
- Which customers should they target
 - For advertising – more focused campaign
 - Customers they most/least want to keep
 - Most favorable business decisions
- Associations
 - Which products should/should not be on the same shelf
 - Which products should be advertised together
 - Which products should be bundled
- Information Brokers
 - Make transaction information available to others who are seeking advantages

Data Mining

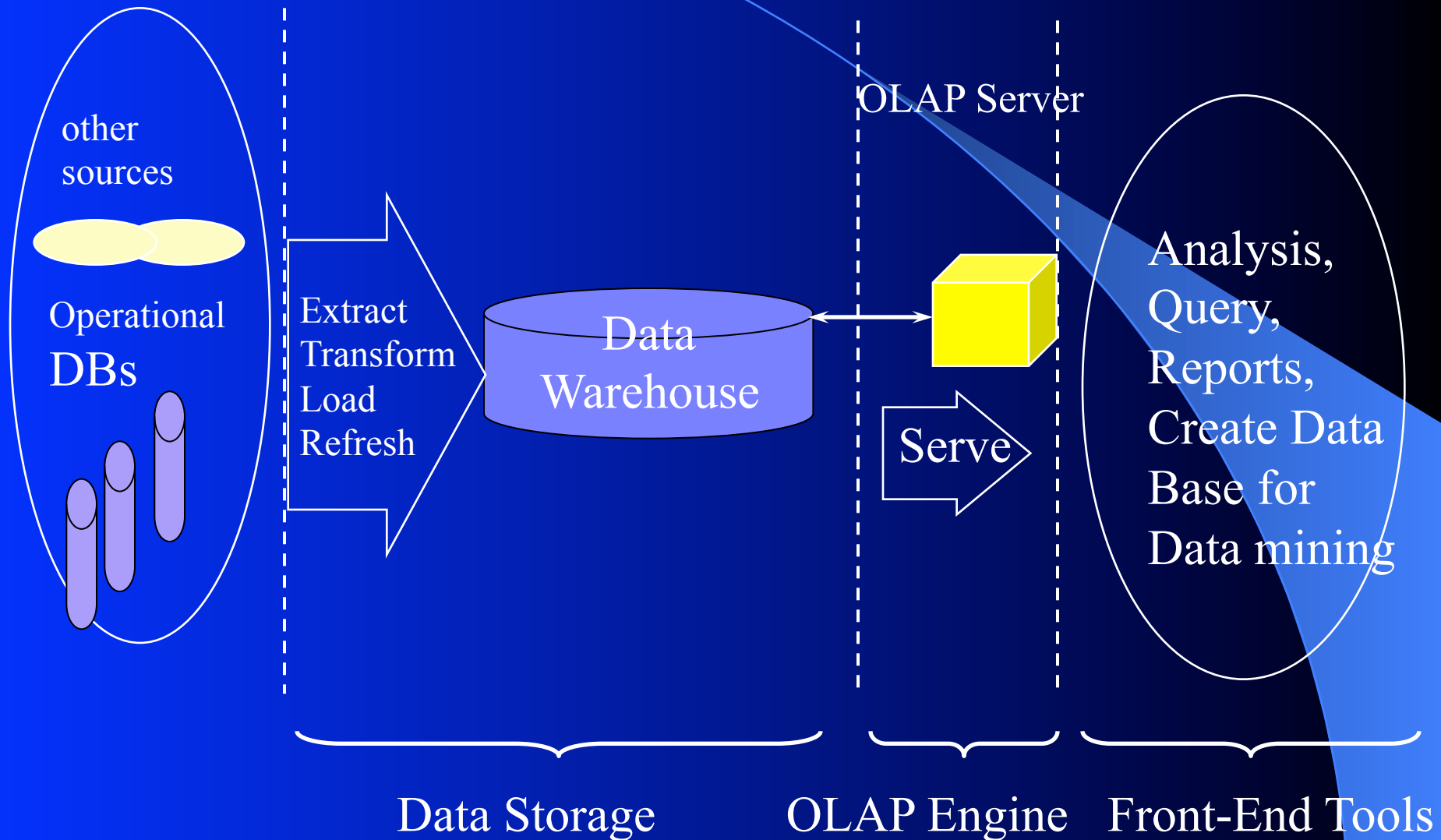
- Basically, a particular niche of machine learning applications
 - Focused on business and other large data problems
 - Focused on problems with huge amounts of data which needs to be manipulated in order to make effective inferences
 - “Mine” for “gems” of actionable information

Make shorter Data warehouse slide and move
next two back

Data Warehouse

- Companies have large data warehouses of transactions
 - Records of sales at a store
 - On-line shopping
 - Credit card usage
 - Phone calls made and received
 - Visits and navigation of web sites, etc...
- Many/Most things recorded these days and there is potential information that can be mined to gain business improvements
 - For better customer service/support and/or profits
- Data Warehouse (DWH)
 - Separate from the operational data (OLTP – Online transaction processing)
 - Data comes from heterogeneous company sources
 - Contains static records of data which can be used and manipulated for analysis and business purposes
 - Old data is rarely modified, and new data is continually added
 - OLAP (Online Analytical Processing) – Front end to DWH allowing basic data base style queries
 - Useful for data analysis and data gathering and creating the task data base

The Big Picture: DBs, DWH, OLAP & DM



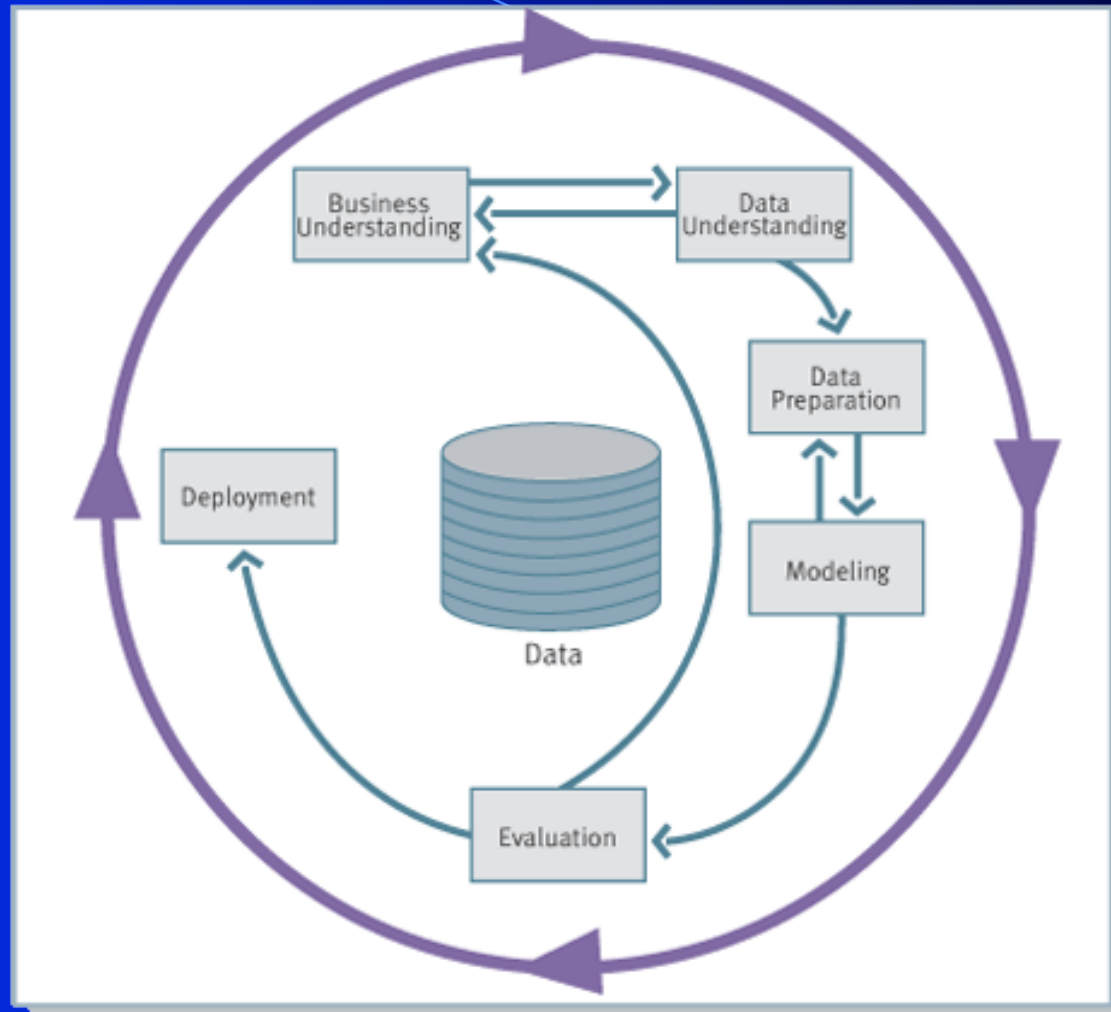
Data Mining Popularity

- Recent Data Mining explosion based on:
- Data available – Transactions recorded in data warehouses
 - From these warehouses specific databases for the goal task can be created
- Algorithms available – Machine Learning and Statistics
 - Including special purpose Data Mining software products to make it easier for people to work through the entire data mining cycle
- Computing power available
- Competitiveness of modern business – need an edge

Data Mining Process Model

- You will use this basic process in your group project
 1. Identify and define the task (e.g. business problem)
 2. Gather and Prepare the Data
 - Build Data Base for the task
 - Select/Transform/Derive features
 - Analyze and Clean the Data
 3. Build and Evaluate the Model(s) – Using training and test data
 4. Deploy the Model(s) and Evaluate business related Results
 - Data visualization tools
 5. Iterate through this process to gain continual improvements
 - both initially and during life of task
 - Improve/adjust features and/or machine learning approach

Data Mining Process Model - Cycle



Monitor, Evaluate, and update deployment

Data Science and Big Data

- Interdisciplinary field about scientific methods, processes and systems to extract knowledge or insights from data
 - Machine Learning
 - Statistics/Math
 - CS/Database/Algorithms
 - Visualization
 - Parallel Processing
 - Etc.
- Increasing demand in industry!

Group Projects

- Review timing and expectations
 - Progress Report
 - Time purposely available between Decision Tree and Instance Based projects to keep going on the group project
 - Gathering, Cleaning, Transforming the Data can be the most critical part of the project, so get that going early!!
 - Then plenty of time to try some different ML models and some iterations on your Features/ML approaches to get improvements
 - Final report and presentation
- Questions?

Association Analysis – Link Analysis

- Used to discover relationships in large databases
- Relationships represented as *association rules*
 - Unsupervised learning, any data set
- One example is *market basket analysis* which seeks to understand more about what items are bought together
 - This can then lead to improved approaches for advertising, product placement, etc.
 - Example Association Rule: {Cereal} \Rightarrow {Milk}

Transaction ID and Info	Items Bought
1 and (who, when, etc.)	{Ice cream, milk, eggs, cereal}
2	{Ice cream}
3	{milk, cereal, sugar}
4	{eggs, yogurt, sugar}
5	{Ice cream, milk, cereal}

Association Discovery

- Association rules are not causal, show correlations
- k -itemset is a subset of the possible items – {Milk, Eggs} is a 2-itemset
- Which itemsets does transaction 3 contain
- Association Analysis/Discovery seeks to find frequent itemsets

TID	Items Bought
1	{Ice cream, milk, eggs, cereal}
2	{Ice cream}
3	{milk, cereal, sugar}
4	{eggs, yogurt, sugar}
5	{Ice cream, milk, cereal}

Association Rule Quality

$$\text{support}(X) = \frac{|\{t \in T : X \subseteq t\}|}{|T|}$$

$$\text{support}(X \Rightarrow Y) = \frac{|\{t \in T : (X \cup Y) \subseteq t\}|}{|T|}$$

$$\text{confidence}(X \Rightarrow Y) = \frac{|\{t \in T : (X \cup Y) \subseteq t\}|}{|\{t \in T : X \subseteq t\}|}$$

$$\text{lift}(X \Rightarrow Y) = \frac{\text{confidence}(X \Rightarrow Y)}{\text{support}(Y)}$$

TID	Items Bought
1	{Ice cream, milk, eggs, cereal}
2	{Ice cream}
3	{milk, cereal, sugar}
4	{eggs, yogurt, sugar}
5	{Ice cream, milk, cereal}

- $t \in T$, the set of all transactions, and X and Y are itemsets
- Rule quality measured by support and confidence
 - Without sufficient support (frequency), rule will probably overfit, and also of little interest, since it is rare
 - Note $\text{support}(X \Rightarrow Y) = \text{support}(Y \Rightarrow X) = \text{support}(X \cup Y)$
 - Note that $\text{support}(X \cup Y)$ is support for itemsets where both X **and** Y occur
 - Confidence measures reliability of the inference (to what extent does X imply Y)
 - $\text{confidence}(X \Rightarrow Y) \neq \text{confidence}(Y \Rightarrow X)$
 - Support and confidence range between 0 and 1
 - Lift: Lift is high when $X \Rightarrow Y$ has high confidence and the consequent Y is less common, Thus lift suggests ability for X to infer a less common value with good probability

Association Rule Discovery Defined

- User supplies two thresholds
 - *minsup* (Minimum required support level for a rule)
 - *minconf* (Minimum required confidence level for a rule)
- Association Rule Discovery: Given a set of transactions T , find all rules having support $\geq \text{minsup}$ and confidence $\geq \text{minconf}$
- How do you find the rules?
- Could simply try every possible rule and just keep those that pass
 - Number of candidate rules is exponential in the size of the number of items
- Standard Approaches - Apriori
 - 1st find frequent itemsets (Frequent itemset generation)
 - Then return rules within those frequent itemsets that have sufficient confidence (Rule generation)
 - Both steps have an exponential number of combinations to consider
 - Number of itemsets exponential in number of items m (power set: 2^m)
 - Number of rules per itemset exponential in number of items in itemset ($n!$)

Apriori Algorithm

- The support for the rule $X \Rightarrow Y$ is the same as the support of the itemset $X \cup Y$
 - Assume $X = \{\text{milk, eggs}\}$ and $Y = \{\text{cereal}\}$. $C = X \cup Y$
 - All the possible rule combinations of itemset C have the same support (# of possible rules exponential in width of itemset: $|C|!$)
 - $\{\text{milk, eggs}\} \Rightarrow \{\text{cereal}\}$
 - $\{\text{milk}\} \Rightarrow \{\text{cereal, eggs}\}$
 - $\{\text{eggs}\} \Rightarrow \{\text{milk, cereal}\}$
 - $\{\text{milk, cereal}\} \Rightarrow \{\text{eggs}\}$
 - $\{\text{cereal, eggs}\} \Rightarrow \{\text{milk}\}$
 - $\{\text{cereal}\} \Rightarrow \{\text{milk, eggs}\}$
- Do they have the same confidence?
- So rather than find common rules we can first just find all itemsets with support $\geq \text{minsup}$
 - These are called frequent itemsets
 - After that we can find which rules within the common itemsets have sufficient confidence to be kept

Support-based Pruning

- Apriori Principle: If an itemset is frequent, then all subsets of that itemset will be frequent
 - Note that subset refers to the items in the itemset
- If an itemset is not frequent, then any superset of that itemset will also not be frequent

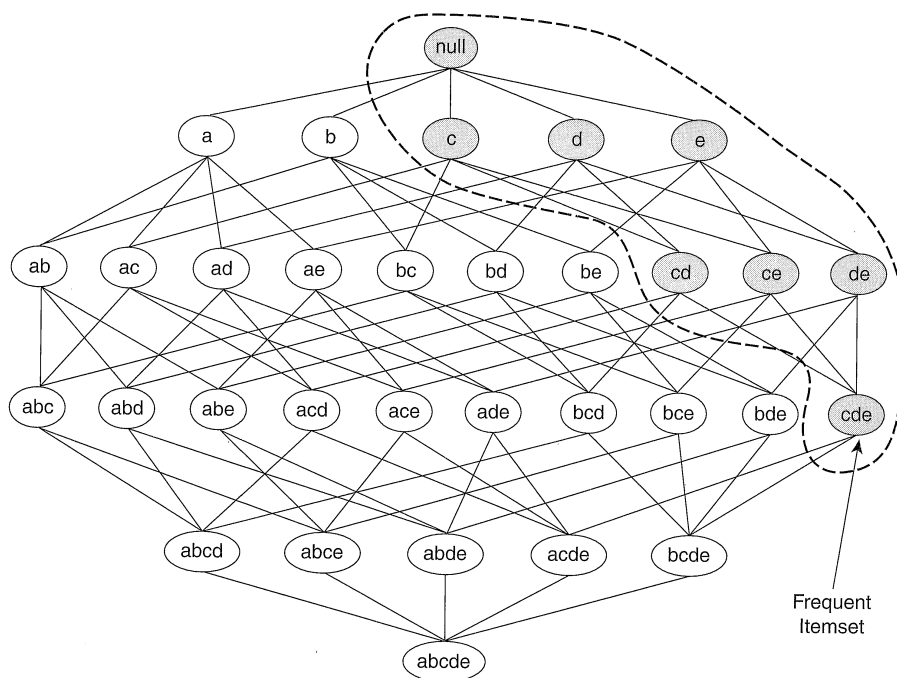


Figure 6.3. An illustration of the *Apriori* principle. If $\{c, d, e\}$ is frequent, then all subsets of this itemset are frequent.

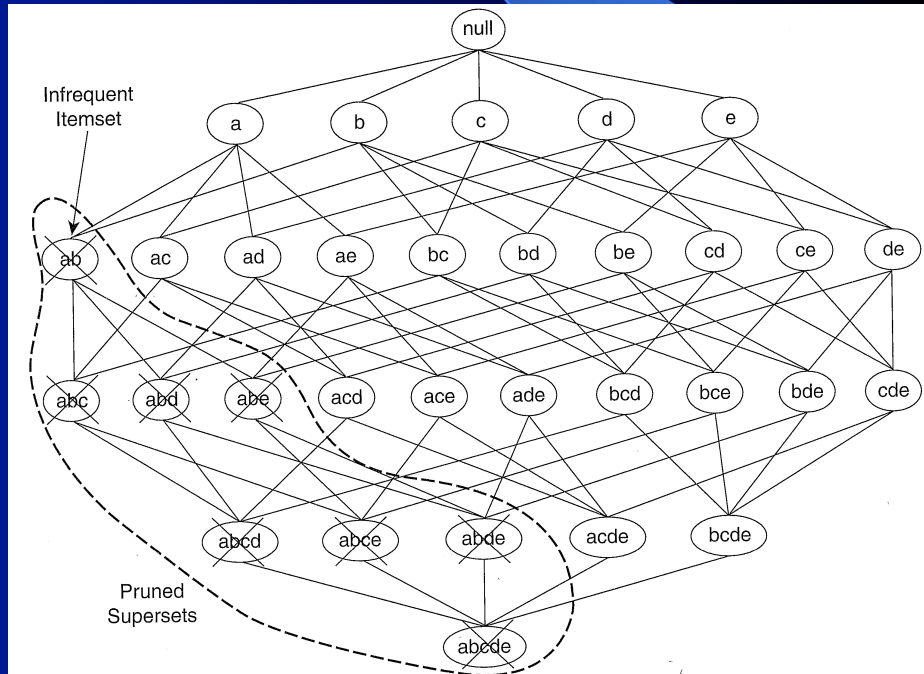


Figure 6.4. An illustration of support-based pruning. If $\{a, b\}$ is infrequent, then all supersets of $\{a, b\}$ are infrequent.

Apriori: Breadth First Search

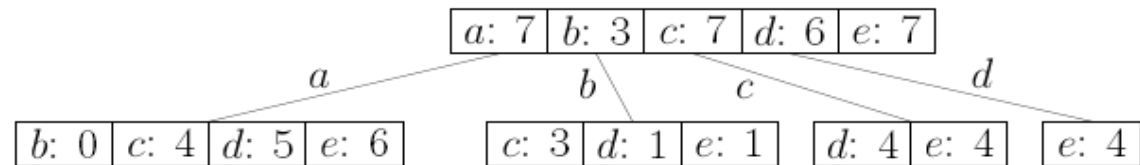
- 1: $\{a, d, e\}$
- 2: $\{b, c, d\}$
- 3: $\{a, c, e\}$
- 4: $\{a, c, d, e\}$
- 5: $\{a, e\}$
- 6: $\{a, c, d\}$
- 7: $\{b, c\}$
- 8: $\{a, c, d, e\}$
- 9: $\{c, b, e\}$
- 10: $\{a, d, e\}$

a: 7	b: 3	c: 7	d: 6	e: 7
------	------	------	------	------

- Example transaction DB with 5 items and 10 transactions
- Minsup = 30%, at least 3 transaction must contain the itemset
- For each itemset at the current level of the tree (depth k) go through each of the n transactions and update tree itemset counts accordingly
- All 1-itemsets are kept since all have support $\geq 30\%$

Apriori: Breadth First Search

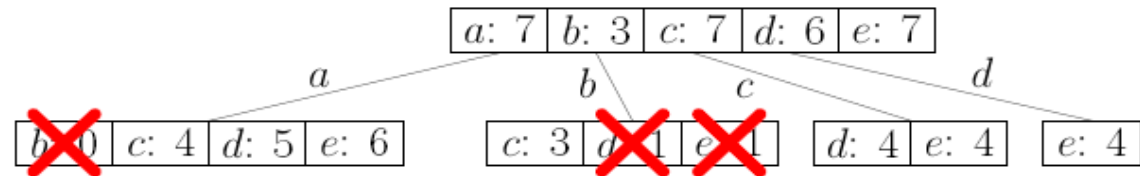
- 1: {a, d, e}
- 2: {b, c, d}
- 3: {a, c, e}
- 4: {a, c, d, e}
- 5: {a, e}
- 6: {a, c, d}
- 7: {b, c}
- 8: {a, c, d, e}
- 9: {c, b, e}
- 10: {a, d, e}



- Generate level 2 of the tree (all possible 2-itemsets)
- Normally use lexical ordering in itemsets to generate/count candidates more efficiently
 - (a,b), (a,c), (a,d), (a,e), (b,c), (b,d), ..., (d,e)
 - When looping through n transactions for (a,b), can stop if a not first in the set, etc.
- Number of tree nodes will grow exponentially if not pruned
- Which ones can we prune assuming minsup = .3?

Apriori: Breadth First Search

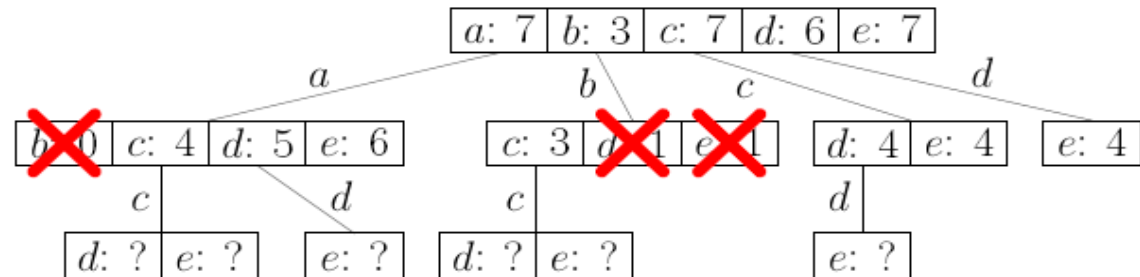
- 1: {a, d, e}
- 2: {b, c, d}
- 3: {a, c, e}
- 4: {a, c, d, e}
- 5: {a, e}
- 6: {a, c, d}
- 7: {b, c}
- 8: {a, c, d, e}
- 9: {c, b, e}
- 10: {a, d, e}



- Generate level 2 of the tree (all possible 2-itemsets)
- Use lexical ordering in itemsets to generate/count candidates more efficiently
 - (a,b), (a,c), (a,d), (a,e), (b,c), (b,d), ..., (d,e)
 - When looping through n transactions for (a,b), can stop if a not first in the set, etc.
- Number of tree nodes will grow exponentially if not pruned
- Which ones can we prune assuming minsup = .3?

Apriori: Breadth First Search

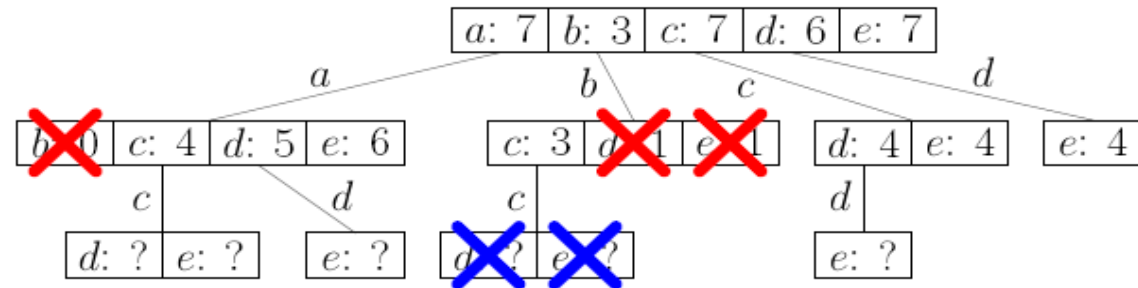
- 1: {a, d, e}
- 2: {b, c, d}
- 3: {a, c, e}
- 4: {a, c, d, e}
- 5: {a, e}
- 6: {a, c, d}
- 7: {b, c}
- 8: {a, c, d, e}
- 9: {c, b, e}
- 10: {a, d, e}



- Generate level 3 of the tree (all 3-itemsets with frequent parents)
- Before calculating the counts, check to see if any of these newly generated 3-itemsets, contain an infrequent 2-itemset. If so we can prune it before we count since it must be infrequent
 - A k -itemset contains k subsets of size $k-1$
 - It's parent in the tree is only one of those subsets
 - Are there any candidates we can delete?

Apriori: Breadth First Search

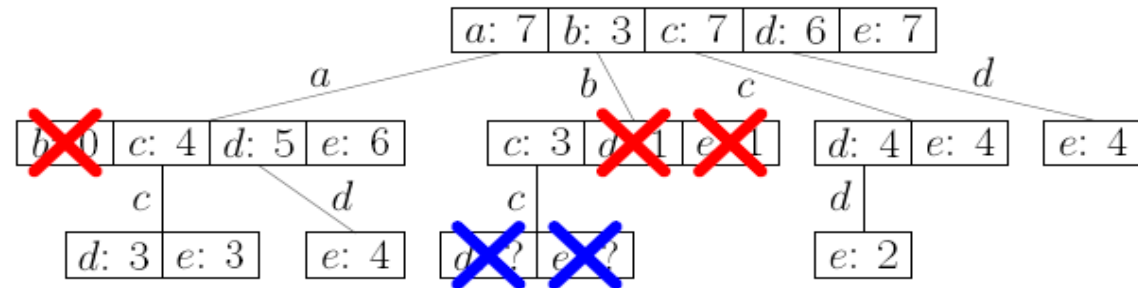
- 1: {a, d, e}
- 2: {b, c, d}
- 3: {a, c, e}
- 4: {a, c, d, e}
- 5: {a, e}
- 6: {a, c, d}
- 7: {b, c}
- 8: {a, c, d, e}
- 9: {c, b, e}
- 10: {a, d, e}



- The item sets {b, c, d} and {b, c, e} can be pruned, because
 - {b, c, d} contains the infrequent item set {b, d} and
 - {b, c, e} contains the infrequent item set {b, e}.

Apriori: Breadth First Search

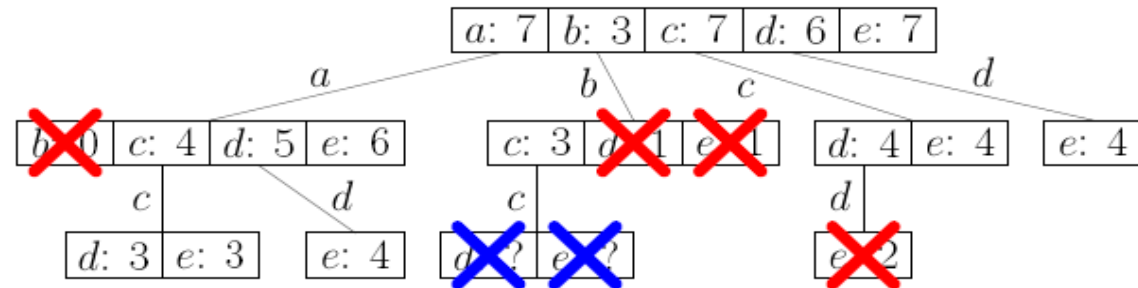
- 1: {a, d, e}
- 2: {b, c, d}
- 3: {a, c, e}
- 4: {a, c, d, e}
- 5: {a, e}
- 6: {a, c, d}
- 7: {b, c}
- 8: {a, c, d, e}
- 9: {c, b, e}
- 10: {a, d, e}



- Only the remaining four item sets of size 3 are evaluated.

Apriori: Breadth First Search

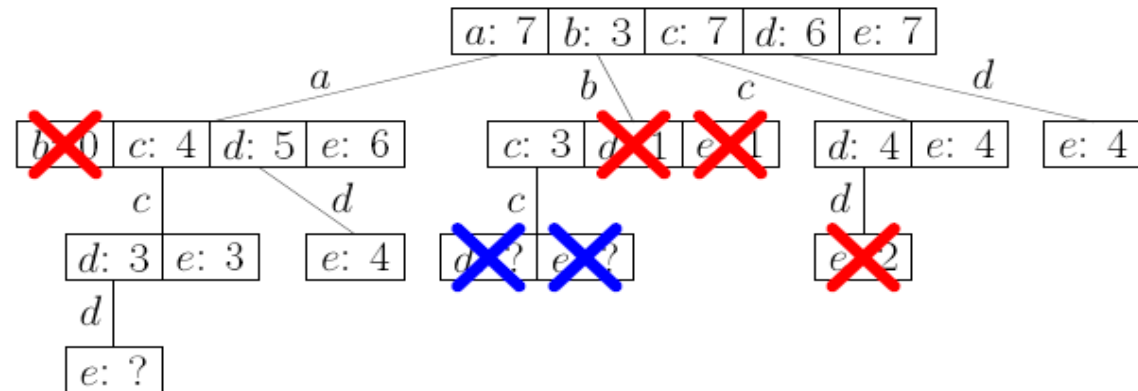
- 1: {a, d, e}
- 2: {b, c, d}
- 3: {a, c, e}
- 4: {a, c, d, e}
- 5: {a, e}
- 6: {a, c, d}
- 7: {b, c}
- 8: {a, c, d, e}
- 9: {c, b, e}
- 10: {a, d, e}



- Minimum support: 30%, i.e., at least 3 transactions must contain the item set.
- Infrequent item set: {c, d, e}.

Apriori: Breadth First Search

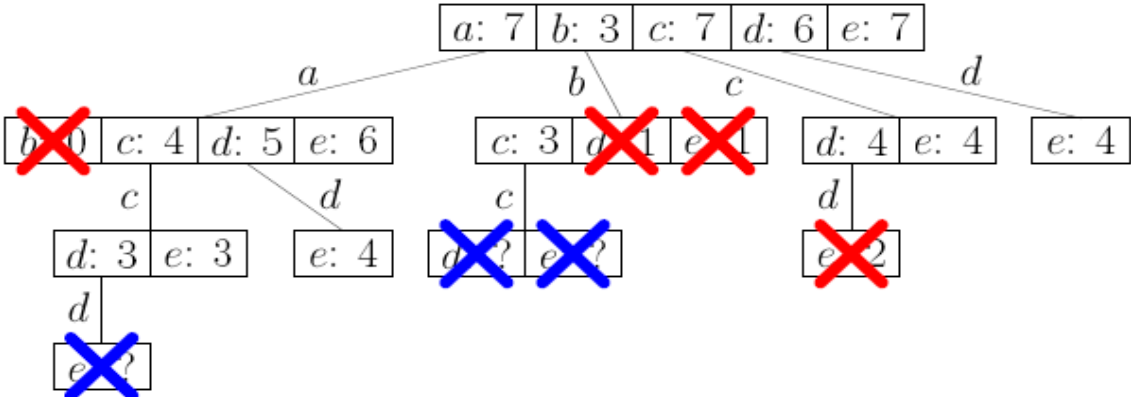
- 1: {a, d, e}
- 2: {b, c, d}
- 3: {a, c, e}
- 4: {a, c, d, e}
- 5: {a, e}
- 6: {a, c, d}
- 7: {b, c}
- 8: {a, c, d, e}
- 9: {c, b, e}
- 10: {a, d, e}



- Generate candidate item sets with 4 items (parents must be frequent).
- Before counting, check whether the candidates contain an infrequent item set.

<p>Apriori: Breadth First Search</p>

- 1: $\{a, d, e\}$
- 2: $\{b, c, d\}$
- 3: $\{a, c, e\}$
- 4: $\{a, c, d, e\}$
- 5: $\{a, e\}$
- 6: $\{a, c, d\}$
- 7: $\{b, c\}$
- 8: $\{a, c, d, e\}$
- 9: $\{c, b, e\}$
- 10: $\{a, d, e\}$



- The item set $\{a, c, d, e\}$ can be pruned, because it contains the infrequent item set $\{c, d, e\}$.
- Consequence: No candidate item sets with four items.
- Fourth access to the transaction database is not necessary.

- Frequent itemsets are: {a,c}, {a,c,d}, {a,c,e}, {a,d}, {a,d,e}, {a,e}, {b,c}, {c,d}, {c,e}, {d,e}

Rule Generation

- Frequent itemsets were: $\{a,c\}$, $\{a,c,d\}$, $\{a,c,e\}$, $\{a,d\}$, $\{a,d,e\}$, $\{a,e\}$, $\{b,c\}$, $\{c,d\}$, $\{c,e\}$, $\{d,e\}$
- For each frequent itemset generate the possible rules and keep those with confidence $\geq \text{minconf}$
- First itemset $\{a,c\}$ gives possible rules
 - $\{a\} \Rightarrow \{c\}$ with confidence $4/7$ and
 - $\{c\} \Rightarrow \{a\}$ with confidence $4/7$
- Second itemset $\{a,c,d\}$ leads to six possible rules
- Just as with frequent itemset generation, we can use pruning and smart lexical ordering to make rule generation more efficient
 - Project? – Search pruning tricks (312) vs ML

Illustrative Training Set

Would if we had real valued data?
What are steps for this example?

Risk Assessment for Loan Applications

Client #	Credit History	Debt Level	Collateral	Income Level	RISK LEVEL
1	Bad	High	None	Low	HIGH
2	Unknown	High	None	Medium	HIGH
3	Unknown	Low	None	Medium	MODERATE
4	Unknown	Low	None	Low	HIGH
5	Unknown	Low	None	High	LOW
6	Unknown	Low	Adequate	High	LOW
7	Bad	Low	None	Low	HIGH
8	Bad	Low	Adequate	High	MODERATE
9	Good	Low	None	High	LOW
10	Good	High	Adequate	High	LOW
11	Good	High	None	Low	HIGH
12	Good	High	None	Medium	MODERATE
13	Good	High	None	High	LOW
14	Bad	High	None	Medium	HIGH

Running Apriori (I)

- Choose *MinSupport* = .4 and *MinConfidence* = .8
- 1-Itemsets (Level 1):
 - (CH=Bad, .29) (CH=Unknown, .36) (CH=Good, .36)
 - (DL=Low, .5) (DL=High, .5)
 - (C=None, .79) (C=Adequate, .21)
 - (IL=Low, .29) (IL=Medium, .29) (IL=High, .43)
 - (RL=High, .43) (RL=Moderate, .21) (RL=Low, .36)

Running Apriori (II)

- 1-Itemsets = $\{(DL=Low, .5); (DL=High, .5); (C=None, .79); (IL=High, .43); (RL=High, .43)\}$
- 2-Itemsets = $\{(DL=High + C=None, .43)\}$
- 3-Itemsets = $\{\}$
- Two possible rules:
 - $DL=High \Rightarrow C=None$
 - $C=None \Rightarrow DL=High$
- Confidences:

– $Conf(DL=High \Rightarrow C=None) = .86$	<u>Retain</u>
– $Conf(C=None \Rightarrow DL=High) = .54$	Ignore

Summary

- Association Analysis useful in many real world tasks
 - Not a classification approach, but a way to understand relationships in data and use this knowledge to advantage
- Also standard classification and other approaches
- Data Mining continues to grow as a field
 - Data and features issues
 - Gathering, Selection and Transformation, Preparation, Cleaning, Storing
 - Data visualization and understanding
 - Outlier detection and handling
 - Time series prediction
 - Web mining
 - etc.