

Association Analysis

Naren Ramakrishnan



Recap

- Supervised learning
 - Classification, Regression
- Unsupervised learning
 - Clustering, Dimensionality Reduction
- Time Series Analysis
 - Both supervised and unsupervised learning

Today

- Association analysis
 - Primarily unsupervised learning
 - One of the “new age” data mining problems
- Goes by other names
 - Market basket analysis
 - Mining transaction datasets
 - Itemset mining
 - Association rule mining

Example

- Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction

Market basket

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Examples of Association Rules

$\{\text{Diaper}\} \rightarrow \{\text{Beer}\},$
 $\{\text{Milk, Bread}\} \rightarrow \{\text{Eggs, Coke}\},$
 $\{\text{Beer, Bread}\} \rightarrow \{\text{Milk}\},$

Implication means co-occurrence,
not causality!

Another example

- Mining associations in electronic medical records

Property	Value
Number of patients	1,620,681
Number of diagnostic (ICD) codes	41,186,511
Number of procedure (CPT) codes	38,942,605
Max. number of codes in a record	10,430
Min. number of codes in a record	1
Max. span of a record in days	8202 days \approx 22.5 years
Min. span of a record in days	1

Describing the Relationship between Cat Bites and Human Depression Using Data from an Electronic Health Record

How do we find association rules?

- First
 - Find “frequent” itemsets $\{X, Y\}$
 - Defined by a support threshold
- Next
 - See if $X \rightarrow Y$ or $Y \rightarrow X$ hold
 - Defined by a confidence threshold

Frequent itemsets

- **Itemset**
 - A collection of one or more items
 - Example: {Milk, Bread, Diaper}
 - k-itemset
 - An itemset that contains k items
- **Support count (σ)**
 - Frequency of occurrence of an itemset
 - E.g. $\sigma(\{\text{Milk, Bread, Diaper}\}) = 2$
- **Support**
 - Fraction of transactions that contain an itemset
 - E.g. $s(\{\text{Milk, Bread, Diaper}\}) = 2/5$
- **Frequent Itemset**
 - An itemset whose support is greater than or equal to a *minsup* threshold

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Association rules

- Association Rule

- An implication expression of the form $X \rightarrow Y$, where X and Y are itemsets
- Example:
 $\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

- Rule Evaluation Metrics

- Support (s)
 - ◆ Fraction of transactions that contain both X and Y
- Confidence (c)
 - ◆ Measures how often items in Y appear in transactions that contain X

Example:

$\{\text{Milk, Diaper}\} \Rightarrow \text{Beer}$

$$s = \frac{\sigma(\text{Milk, Diaper, Beer})}{|T|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk, Diaper, Beer})}{\sigma(\text{Milk, Diaper})} = \frac{2}{3} = 0.67$$

All this sounds good..

- But how do we mine association rules from a given database?
 - Keep in mind that the database is likely to have billions of transactions and potentially millions of items

Observation 1

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Rules:

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

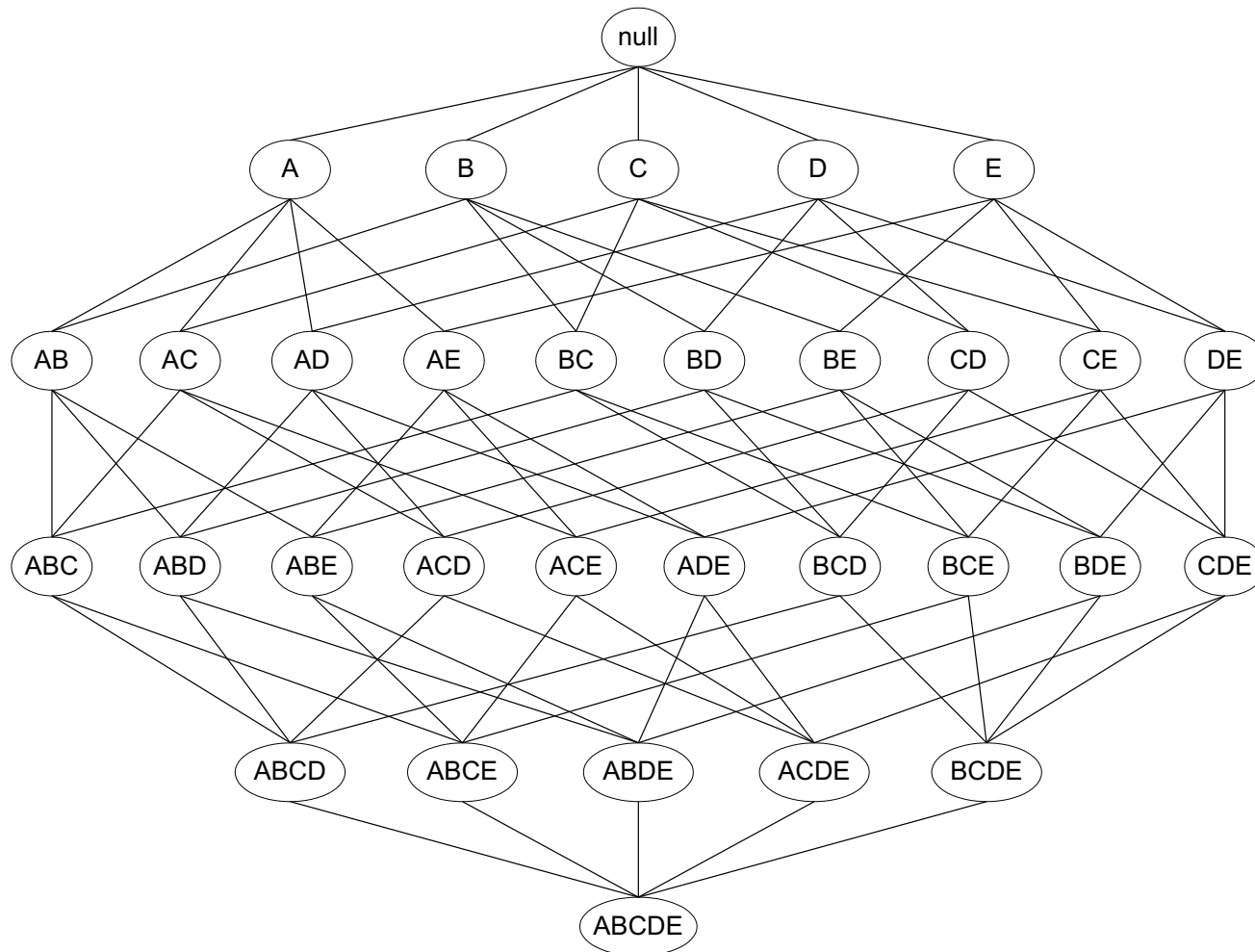
Observations:

- All the above rules are binary partitions of the same itemset:
 $\{\text{Milk, Diaper, Beer}\}$
- Rules originating from the same itemset have identical support but can have different confidence
- Thus, we may decouple the support and confidence requirements

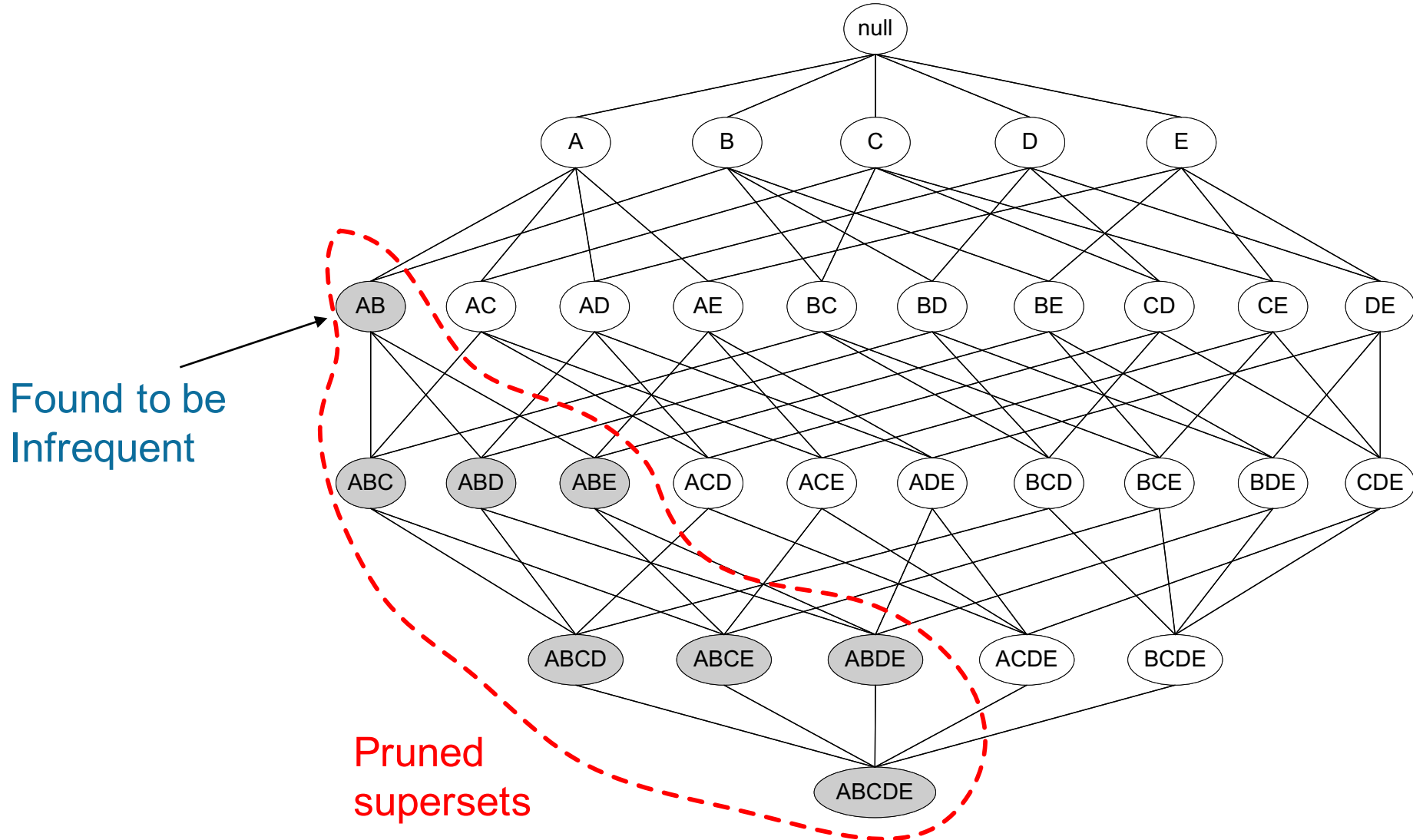
Two step approach

- Two-step approach:
 1. Frequent Itemset Generation
 - Generate all itemsets whose support \geq minsup
 2. Rule Generation
 - Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset
- Frequent itemset generation is still computationally expensive!

Searching for sets



Observation 2



The Apriori principle

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

Items (1-itemsets)



Itemset	Count
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk,Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	3

If every subset is considered,
 ${}^6C_1 + {}^6C_2 + {}^6C_3 = 41$
With support-based pruning,
 $6 + 6 + 1 = 13$

More formally

- Support is ***anti-monotone***
 - If an itemset X does not have support, no superset of X can have support

The Apriori algorithm

- Let $k=1$
- Generate frequent itemsets of length 1
- Repeat until no new frequent itemsets are identified
 - Generate length $(k+1)$ candidate itemsets from length k frequent itemsets
 - Prune candidate itemsets containing subsets of length k that are infrequent
 - Count the support of each candidate by scanning the DB
 - Eliminate candidates that are infrequent, leaving only those that are frequent

The Apriori algorithm

- Let $k=1$
- Generate frequent itemsets of length 1
- Repeat until no new frequent itemsets are identified
 - Generate length $(k+1)$ candidate itemsets from length k frequent itemsets – HOW?
 - Prune candidate itemsets containing subsets of length k that are infrequent
 - Count the support of each candidate by scanning the DB
 - Eliminate candidates that are infrequent, leaving only those that are frequent

Generating $(k+1)$ itemsets from k -itemsets

- To generate a $(k+1)$ itemset
 - Pick 2 k -itemsets that have the same $(k-1)$ prefix
 - Merge them!
- Example
 - (A,B,C) and (A,B,D) are merged to form (A,B,C,D)

Rule generation

- Given a frequent itemset L , find all non-empty subsets $f \subset L$ such that $f \rightarrow L - f$ satisfies the minimum confidence requirement

– If $\{A,B,C,D\}$ is a frequent itemset, candidate rules:

$ABC \rightarrow D,$	$ABD \rightarrow C,$	$ACD \rightarrow B,$	$BCD \rightarrow A,$
$A \rightarrow BCD,$	$B \rightarrow ACD,$	$C \rightarrow ABD,$	$D \rightarrow ABC$
$AB \rightarrow CD,$	$AC \rightarrow BD,$	$AD \rightarrow BC,$	$BC \rightarrow AD,$
$BD \rightarrow AC,$	$CD \rightarrow AB,$		

- If $|L| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \emptyset$ and $\emptyset \rightarrow L$)

Rule generation

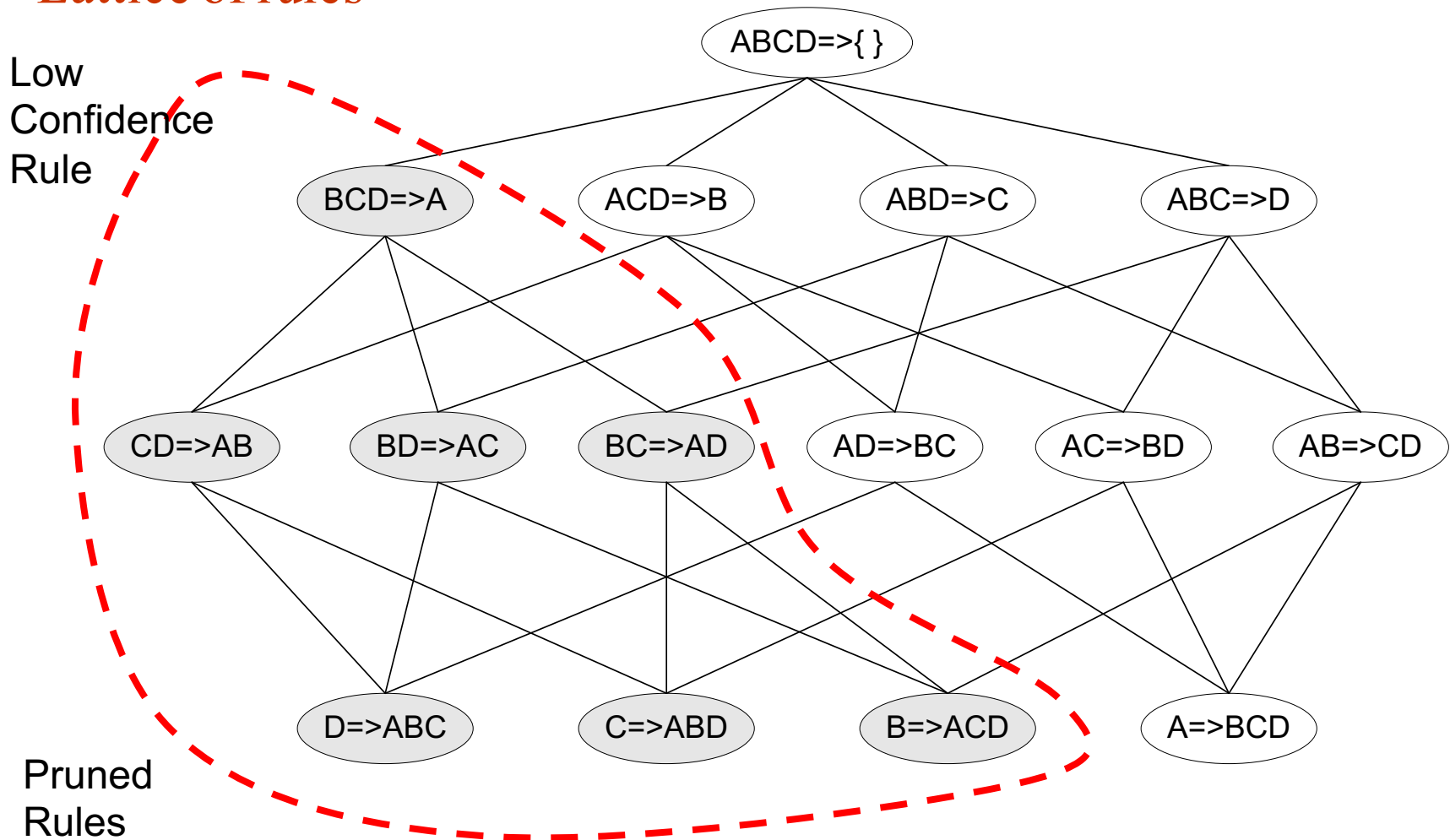
- How to efficiently generate rules from frequent itemsets?
 - In general, confidence does not have an anti-monotone property
 - $c(ABC \rightarrow D)$ can be larger or smaller than $c(AB \rightarrow D)$
 - But confidence of rules generated from the same itemset has an anti-monotone property
 - e.g., $L = \{A, B, C, D\}$:

$$c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$$

- Confidence is anti-monotone w.r.t. number of items on the RHS of the rule

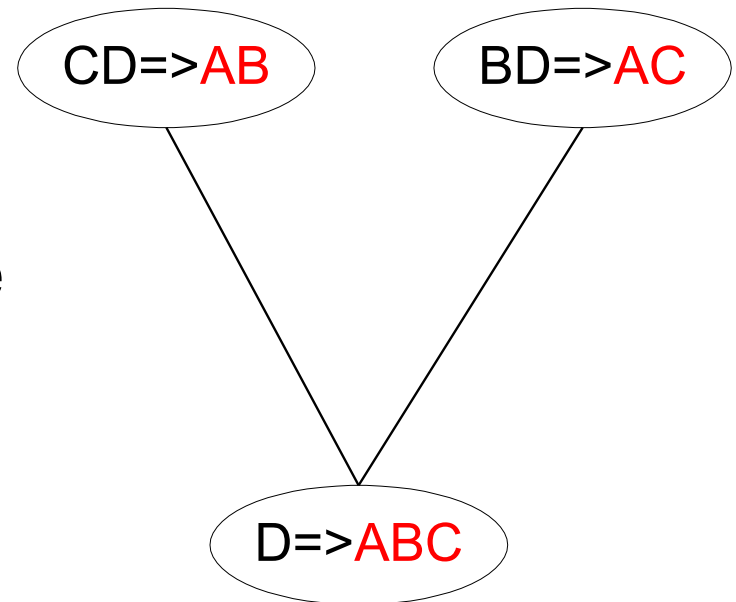
Rule generation (from a single itemset)

Lattice of rules



Rule generation details

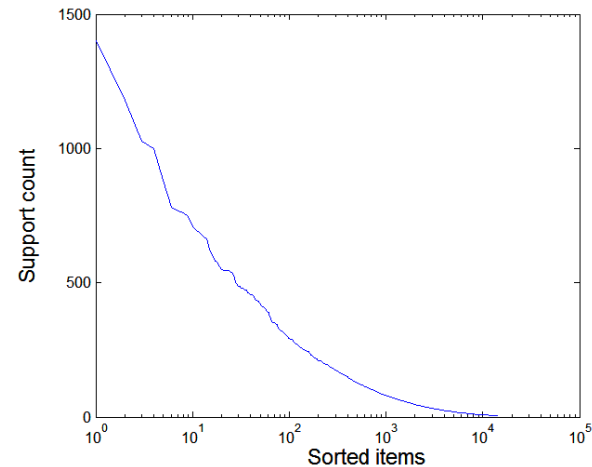
- Candidate rule is generated by merging two rules that share the same prefix in the rule consequent
- $\text{join}(\text{CD} \Rightarrow \text{AB}, \text{BD} \Rightarrow \text{AC})$ would produce the candidate rule $\text{D} \Rightarrow \text{ABC}$
- Prune rule $\text{D} \Rightarrow \text{ABC}$ if its subset $\text{AD} \Rightarrow \text{BC}$ does not have high confidence



Practical issues

- Many real datasets have skewed support distributions

- Too small support vs
- Too large support



- Association analysis tends to produce too many rules!
 - Use interestingness measures to prune/select rules

What we have learnt thus far

- Association analysis
 - A new age data mining problem
 - Data mining is simply “smart counting and book-keeping”
 - Efficiency of search is important!

Recommender systems

Amazon.com: DVD: The Hitchhiker's Guide to the Galaxy (1982) - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Mail News RSS

Address <http://www.amazon.com/exec/obidos/ASIN/B00005YUNJ/qid%3D1083871010/sr%3D11-1/ref%3Dsr%5F11%5F1/002-9924222-238005c> Go Links

Google amazon Search Web Search Site PageRank 55 blocked Options amazon

Jewelry & Watches

WELCOME YOUR STORE BOOKS APPAREL & ACCESSORIES ELECTRONICS TOYS & GAMES MUSIC COMPUTER & VIDEO GAMES DVD SEE MORE STORES Your Gold Box

ADVANCED SEARCH BROWSE GENRES TOP SELLERS NEW & FUTURE RELEASES DVD ESSENTIALS MOVIE SHOWTIMES TODAY'S DEALS USED DVDS

SEARCH

DVD GO!

WEB SEARCH

GO! Powered by Google

DVD INFORMATION

Explore This DVD

[buying info](#)

[technical information](#)

[editorial reviews](#)

[customer reviews](#)

RECENTLY VIEWED

The Hitchhiker's Guide to the Galaxy (1982)

The DVD cover features a green alien character in a red suit, holding a sign that says 'HITCHHIKER'S GUIDE TO THE GALAXY'. The background is dark with some space-themed elements.

List Price: ~~\$34.98~~

Price: **\$28.68** & This item ships for **FREE** with Super Saver Shipping. [See details.](#)

You Save: **\$6.30 (18%)**

Availability: Usually ships within 24 hours

Want it delivered **Monday, May 10?** Order it in the next 23 hours and 12 minutes, and choose **One-Day Shipping** at checkout. [See details.](#)

[see larger picture](#)

18 used & new from **\$26.21**

Edition: DVD

[See more product details](#)

Better Together

Buy this DVD with [Red Dwarf - Series 1 & 2 DVD](#) ~ Chris Barrie today!

Total List Price: ~~\$104.90~~

READY TO BUY?

[Add to Shopping Cart](#)

or

[Sign in](#) to turn on 1-Click ordering.

MORE BUYING CHOICES

[Add to Wish List](#)

[Add to Wedding Registry](#)

Don't have one? We'll set one up for you.

Recommender systems

Amazon.com: DVD: The Hitchhiker's Guide to the Galaxy (1982) - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://www.amazon.com/exec/obidos/ASIN/B00005YUNJ/qid%3D1083871010/sr%3D11-1/ref%3Dsr%5F11%5F1/002-9924222-2380050> Go Links »

Google amazon Search Web Search Site PageRank 55 blocked Options amazon

[information](#)
[editorial reviews](#)
[customer reviews](#)

[See more product details](#)

Don't have one?
We'll set one up for you.

Better Together

Buy this DVD with [Red Dwarf - Series 1 & 2 DVD](#) ~ Chris Barrie today!
Total List Price: \$104.99
Buy Together Today: \$86.01
[Buy both now!](#)

Customers who bought this DVD also bought:

- [The Adventures of Buckaroo Banzai Across the 8th Dimension \(Special Edition\) DVD](#) ~ Peter Weller ([Rate it](#))
- [Red Dwarf - Series 3 & 4 DVD](#) ([Rate it](#))
- [Dark Star DVD](#) ~ Dre Pahich ([Rate it](#))
- [Hyperspace DVD](#) ([Rate it](#))

← **Rate it?**

[Explore Similar Items: 20 in DVD, 20 in Books, and 19 in Video](#)

Product Details


- Starring:** [Simon Jones](#), [David Dixon](#), [See more](#)
- Director:** [Alan J.W. Bell](#)
- Encoding:** Region 1 (U.S. and Canada only. This DVD will probably NOT be viewable in other countries. Read more about [DVD formats](#).)
- Format:** Color
- Rated:** NR
- Studio:** BBC Video

RECENTLY VIEWED

- [Being John Malkovich DVD](#) ~ John Cusack ([Rate it](#))
- [Pi DVD](#) ~ Sean Gullette ([Rate it](#))
- [Master and Commander - The Far Side of the World \(Full Screen Edition\) DVD](#) ([Rate it](#))
- [The Color of Magic](#) by Terry Pratchett (Author) ([Rate it](#))

[See more in the Page You Made](#)

Featured Item:



Recommender systems



machine learning



All

News

Videos

Books

Images

More

Settings

Tools

About 74,200,000 results (0.55 seconds)

Machine learning - Wikipedia

https://en.wikipedia.org/wiki/Machine_learning ▼ Wikipedia ▼

Machine learning is the subfield of computer science that enables computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959)

Portal:Machine learning · List of machine learning ...

Cached

Similar

Machine Learning

How do they work?

- Two basic threads
 - Content-based filtering
 - Tom likes aviation movies, so Tom will like Sully
 - Collaborative filtering
 - Tom likes the movies that Sally likes and Sally liked Sully, so Tom will like Sully
 - Suffers from “cold start” problems

Collaborative filtering

- Two basic flavors
 - User-based
 - Item-based
- Both are types of nearest neighbor reasoning! 😊

Example

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

First measure similarities between users

A popular similarity measure in user-based CF: Pearson correlation

a, b : users


$r_{a,p}$: rating of user a for item p

P : set of items, rated both by a and b

Possible similarity values between -1 and 1; \bar{r}_a, \bar{r}_b = user's average ratings

$$\text{sim}(a, b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}}$$

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1



sim = 0,85
sim = 0,70
sim = -0,79

Make a prediction

- To predict the rating for user a for product p , find others who have rated p and scale their ratings by their similarity to a

$$pred(a, p) = \bar{r}_a + \frac{\sum_{b \in N} sim(a, b) * (r_{b,p} - \bar{r}_b)}{\sum_{b \in N} sim(a, b)}$$

Item-based CF

- User-based CF has scalability issues if there are many more users than items
- Alternative idea
 - Find similarities between items

How this works

- Example
 - Look for items similar to item5
 - Take Alice's ratings for these items to predict her rating for item5

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

How Amazon works

- Purportedly uses item-based collaborative filtering
- Pre-compute item similarities
 - They are more stable than user similarities
 - Neighborhood used at run-time is small since each user has rated only a small number of items

What we have learnt thus far

- Two broad classes of association methods
 - Itemset mining
 - Recommender systems
- We have seen only the most basic/vanilla versions of these methods
 - Significant variations and optimizations abound!