CS 8803 Big Data Systems and Analytics

Data Analytics @ Apache Mahout

Ling Liu
Professor
College of Computing
Georgia Institute of Technology

Adapted from Slides by Mahout team members, Mohamed Eltabakh@WPI

Data Analytics

- **Include machine learning and data mining: Theory + Algorithms + Software Tools**
  - Analyze/mine/summarize large datasets
  - Extract knowledge from past data
  - Predict trends in future data

Descriptive v.s. Predictive Analytics

Degree of Intelligence

- **Predictive Analytics**

Optimization	"What's the best that can happen?"
Predictive Modeling/ Forecasting	"What will happen next?"
Randomized Testing	"What happens if we try this?"
Statistical models	"Why is this happening?"

- **Descriptive Analytics**

Alerts	"What actions are needed?"
Query/drill down	"What exactly is the problem?"
Ad hoc reports	"How many, how often, where?"
Standard Reports	"What happened?"

- **Iterative Learning**
- **Interactive Learning**

Types of Analytic Models

- Unsupervised
  - Using unlabeled data, create function that predicts output
- Supervised
  - Learn labels of training dataset using statistical models
  - Using labeled training data, create function that predicts output of unseen inputs
- Semi-Supervised
  - Uses labeled and unlabeled data

Tools & Algorithms

- **Frequent Pattern Mining** (Association Rules Mining)
- **Clustering**
- **Collaborative Filtering**
  - Unsupervised learning based similarity functions
  - Supervised Learning based Similarity functions
- **Classification:**
  - Regression, SVM, Neural Networks, …
- Others…

Unsupervised Learning

Supervised Learning

Data Type Specific Analytics

- Text Mining
- Graph Analytics
- Spatial Data Analytics
- Temporal Data Analytics
  - Time Series Analysis
  - Trajectory Mining
- Multimedia Mining
  - Audio, Video, Image Pattern Matching
- … …

Supervised Learning or Unsupervised Learning

Big Data Analytics: Characterizations

- Lots of Data
- Feature Extractions in Big Data
  - Simple v.s. Complex
  - Intuitive v.s. Hidden
  - Shallow v.s. Deep
- Too big/costly for people to handle
  - People still can help

Lecture Outline

- Unsupervised Learning
  - Collaborative Filtering
  - Clustering
- Supervised Learning: Classification / ML
  - Decision Tree
  - Neural Network

Apache Mahout

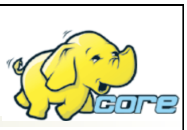
\+ (and other distributed techniques)

Machine Learning

=

- [1].Anil, Robin, Ted Dunning, and Ellen Friedman. Mahout in action. Manning, 2011.

Mahout Development: Motivation



- Large volumes of data are now available
- Platforms now exist to run computations over large datasets (Hadoop, HBase, Spark)
- Sophisticated analytics are needed to turn data into information that people can use
- Active research community and proprietary implementations of "machine learning" algorithms
- The world needs scalable implementations of ML + DM under open license - ASF

Apache Mahout

- Apache Software Foundation project
- Create scalable machine learning + Data Mining libraries
- **Why Mahout?**

  Many Open Source ML libraries :
  - Lack Community
  - Lack Documentation and Examples
  - Lack Scalability
  - Or are research-oriented

Data Analytics @ Mahout …

--Efficient in analyzing/mining data
--Do not scale

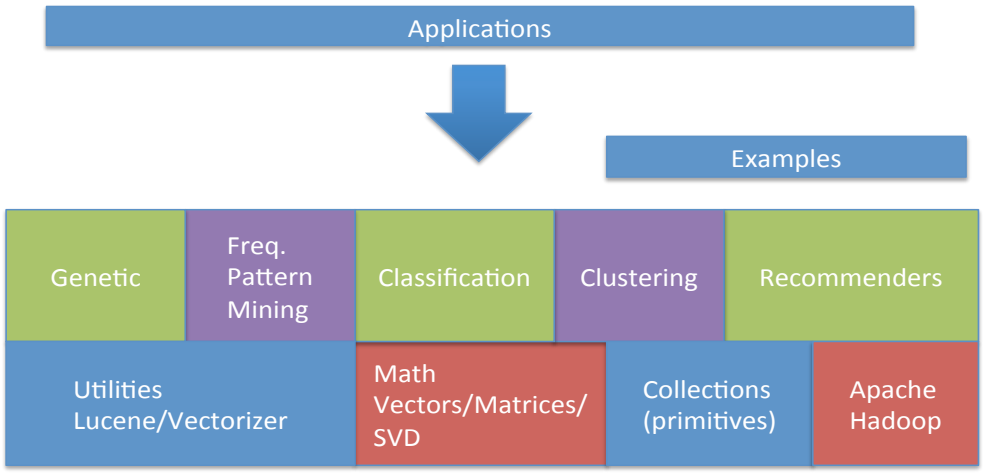
--Efficient in managing big data
--Does not analyze or mine the data

How to integrate these two worlds together

Apache Mahout

- Hadoop brings:
  - Map/Reduce API
  - HDFS
  - → scalability and fault-tolerance
- Mahout brings:
  - Library of machine learning + DM algorithms
  - Examples

Goal 1: Machine Learning



Goal 2: Scalability

- Be as fast and efficient as the possible given the intrinsic design of the algorithm
- Most Mahout implementations are Map Reduce enabled
- Work in Progress

Mahout Package

What can you do with Mahout right now?

- Collaborative Filtering
- Clustering
- Classification
- Frequent Pattern Mining (Association Rule Mining)
- Others
  - Outlier detection
  - Math libirary
    - Vectors, matrices, etc.
  - Noise reduction
  - Graph Processing

Current Code Base

- Matrix & Vector library
  - Memory resident sparse & dense implementations
  - Hama / Giraph for very large arrays
- Utilities
  - Distance Measures
  - Parameters
- Clustering
  - Canopy
  - K-Means
  - Mean Shift
- Collaborative Filtering
  - Taste
- Classification

History of Mahout

- Summer 2007
  - Developers needed scalable ML
  - Mailing list formed
- Community formed
  - Apache contributors
  - Academia & industry
  - Lots of initial interest
- Project formed under Apache Lucene
  - January 25, 2008

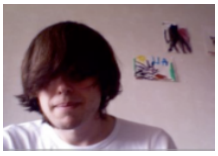
Georgia Institute of Technology
Georgia Tech College of Computing

Core Members of Mahout Team ….

Grant Ingersoll

Dawid Weiss

Otis Gospodetnic



Karl Wettin

Jeff Eastman

Ted Dunning

Erik Hatcher

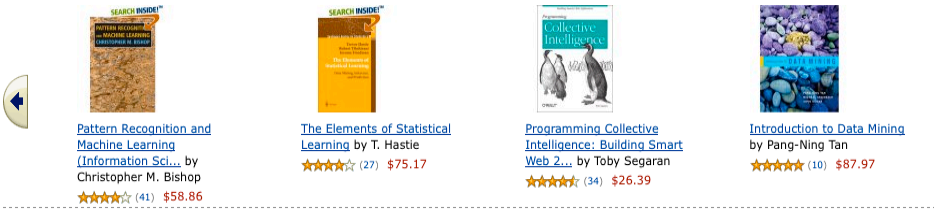
Isabel Drost

Collaborative Filtering

- Unsupervised
- Recommend people and products
  - User-User
    - User likes X, you might too
  - Item-Item
    - People who bought X also bought Y

Collaborative Filtering: An Example

Customers Who Bought This Item Also Bought



Pattern Recognition and Machine Learning (Information Sci... by Christopher M. Bishop
(41) $58.86

The Elements of Statistical Learning by T. Hastie
(27) $75.17

Programming Collective Intelligence: Building Smart Web 2... by Toby Segaran
(34) $26.39

Introduction to Data Mining by Pang-Ning Tan
(10) $87.97

Amazon.com

User-based Collaborative Filtering

- Idea: People who agreed in the past are likely to agree again
- To predict a user's opinion for an item, use the opinion of similar users
- Similarity between users is decided by looking at their overlap in opinions for other items

Example: User-based Collaborative Filtering

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5
User 3 	5	4	7	4	7
User 4 	7	1	7	3	8
User 5 	1	7	4	6	5
User 6 	8	3	8	3	7

Similarity between users

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5
User 4 	7	1	7	3	8

- How similar are users 1 and 2?
- How similar are users 1 and 4?
- … …
- How do you calculate similarity?

Similarity between users: simple way

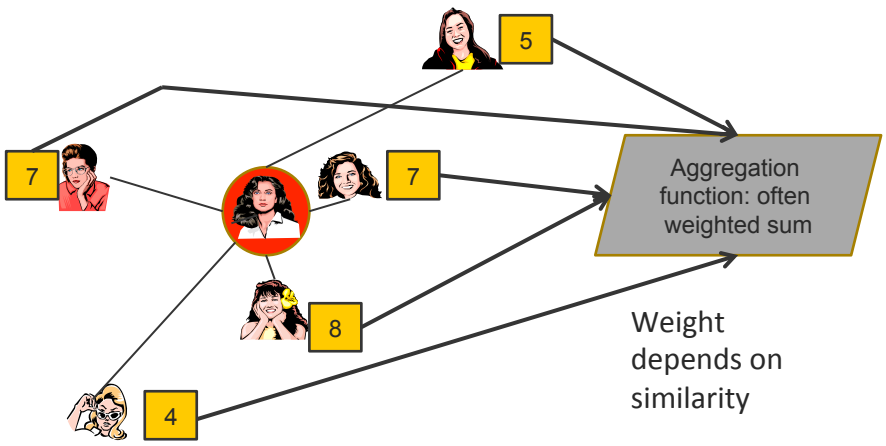
	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	8	1	?	2	7
User 2	2	?	5	7	5

- Only consider items both users have rated
- For each item:
  Calculate difference in the users' ratings
- Take the average of this difference over the items

Average j : Item j rated by User 1 and User 2:

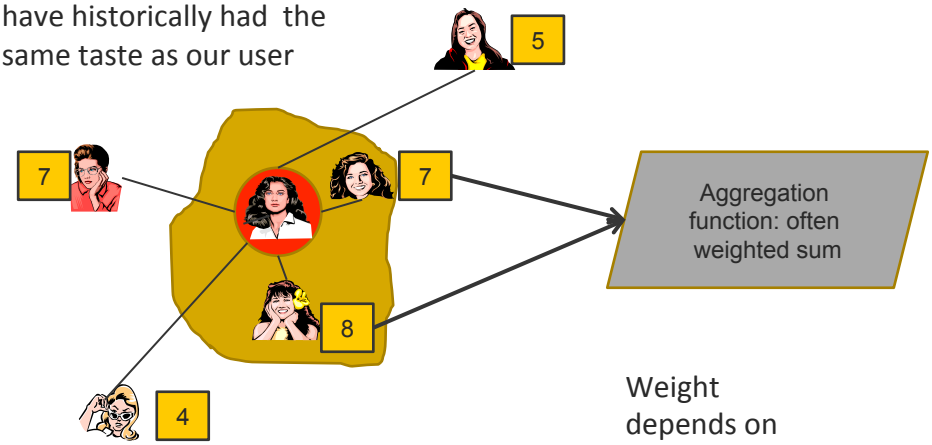
| rating (User 1, Item j) – rating (User 2, Item j) |

Algorithm 1: using entire matrix



Algorithm 2: K-Nearest-Neighbour

Neighbours are people who have historically had the same taste as our user



Weight depends on similarity

Vector Similarity

- Pearson product-moment correlation coefficient (PPMCC/PCC)
  - a measure of the linear correlation (dependence) between two variables *X* and *Y*, represented by a value between +1 and −1 inclusive, where
    - 1 is total positive correlation,
    - 0 is no correlation, and
    - −1 is total negative
  - a measure of the degree of linear dependence between two variables.

Person Correlation Coefficient (PCC)

- Measure the rating similarity (linear dependency) between ratings for active user (*a)* and another user (*u*)

$$c_{a,u} = \frac{\text{covar}(r_a, r_u)}{\sigma_{r_a}\sigma_{r_u}}$$

r_a and r_u are the ratings vectors for the m items rated by *Users a* and *u respectively*

$r_{i,j}$ is user i's rating for item j

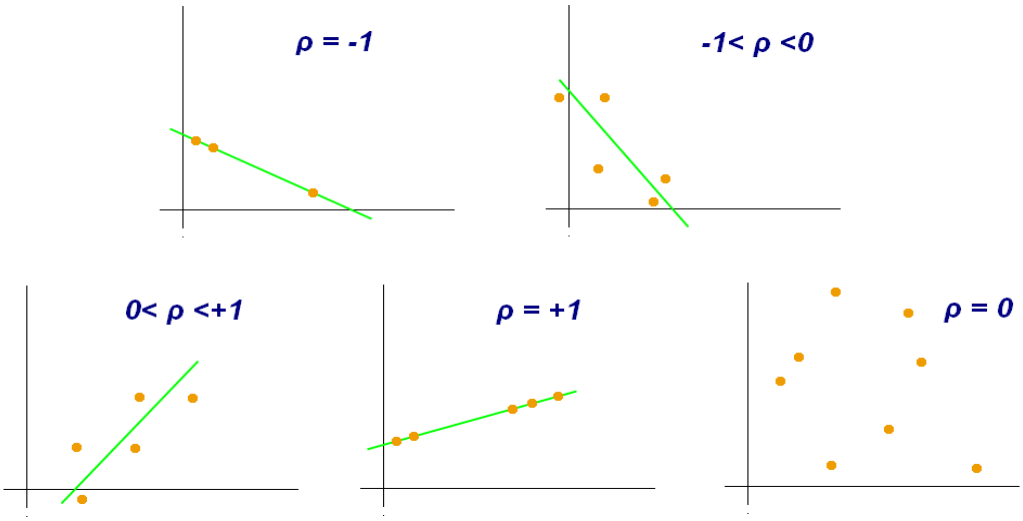
$$\text{covar}(r_a, r_u) = \frac{\sum_{i=1}^{m}(r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{m}$$

$$\bar{r}_x = \frac{\sum_{i=1}^{m} r_{x,i}}{m}$$

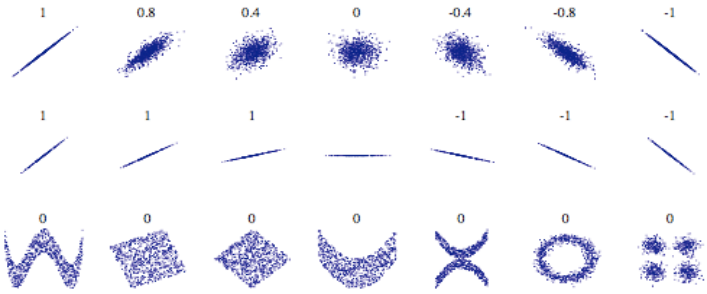
$$\sigma_{r_x} = \sqrt{\frac{\sum_{i=1}^{m}(r_{x,i} - \bar{r}_x)^2}{m}}$$

PCC Examples of scatter diagrams with different values of correlation coefficient (ρ)

Wikipedia.org



PCC: Another Example



- Several sets of (*x*, *y*) points, with the correlation coefficient of *x* and *y* for each set.
- The correlation reflects the non-linearity and direction of a linear relationship (top row),
- but not the slope of that relationship (middle), nor many aspects of nonlinear relationships (bottom).

Significance Weighting

- Important not to trust correlations based on very few co-rated items.
- Include *significance weights*, $s_{a,u}$, based on number of co-rated items, *m*.

$$w_{a,u} = s_{a,u} c_{a,u}$$

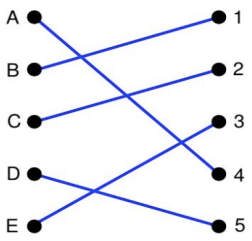
$$s_{a,u} = \begin{cases} 1 & \text{if } m > 50 \\ \frac{m}{50} & \text{if } m \leq 50 \end{cases}$$

Accuracy Problems with User-based CF

- **Cold Start**: hard to recommend items that have no user ratings or insufficient number of other user ratings in the system.
  - **First Rater**: Cannot recommend an item that has not been previously rated.
    - New items
    - Esoteric items
- **Sparsity**: If the user/ratings matrix is sparse, it is hard to find users that have rated the same items.
- **Popularity Bias**: Cannot recommend items to someone with unique tastes.
  - Tends to recommend popular items.

Ideas + Solution Approaches

- Finding useful user correlations beyond the rating network (user-item bipartite graph)
  - Social Networks
  - Context based User Relationships (co-author network with conferences, with subject Keywords, …)
- Finding indirect user rating correlations
  - Iterative Propagation (random walk, heat diffusion, etc.)
  - … …



Potential Innovative Project Ideas

Item-based Collaborative Filtering

- Idea: a user is likely to have the same opinion for similar items
  [same idea as in Content-Based Filtering]
- Similarity between items is decided by looking at how other users have rated them
  [different from Content-based, where item features are used]
- Advantage (compared to user-based CF):
  - Prevents User Cold-Start problem
  - Improves scalability (similarity between items is more stable than between users)

Example: Item-based Collaborative Filtering

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5
User 3 	5	4	7	4	7
User 4 	7	1	7	3	8
User 5 	1	7	4	6	5
User 6 	8	3	8	3	7

Georgia Institute of Technology
Georgia Tech College of Computing

Similarity between items

Item 3	Item 4	Item 5
?	2	7
5	7	5
7	4	7
7	3	8
4	6	5
8	3	7

- How similar are items 3 and 4?
- How similar are items 3 and 5?
- How do you calculate similarity?

Similarity between items: simple way

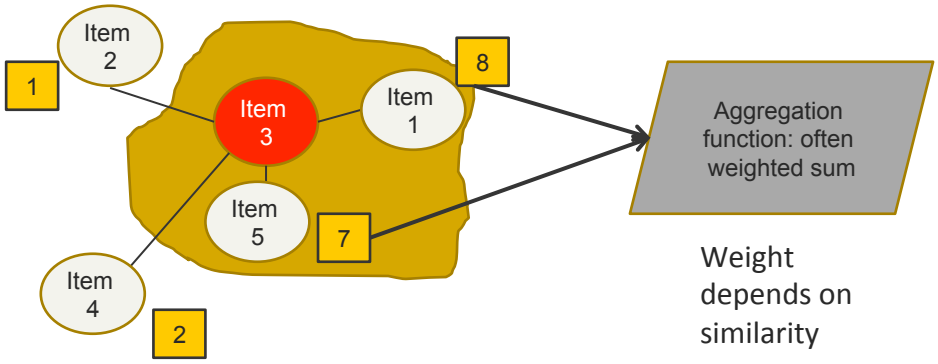
Item 3	Item 4
?	2
5	7
7	4
7	3
4	6
8	3

- Only consider users who have rated both items
- For each user:
  Calculate difference in ratings for the two items
- Take the average of this difference over the users

Average i : User i has rated Items 3 and 4: | rating (User i, Item 3) – rating (User i, Item 4) |

Algorithms

- Similar to the User-Based: can use K nearest-neighbours or the entire rating matrix



Problems with Item based CF

- **Cold Start**: Need enough other users already in the system to find a match.
  - **First Rater**: Cannot recommend an item that has no raters.
    - New items
    - Esoteric items
- **Sparsity**: If the user/ratings matrix is sparse, it is hard to find items that have been rated by the same set of users.
- **Popularity Bias**: Cannot recommend unpopular items to someone with unique tastes.
  - Majority rules

Taste: Movie Recommendation

- Given ratings by users of movies, recommend other movies
- Useful online materials
  - http://blog.trifork.com/2009/12/09/mahout-taste-part-one-introduction/
  - https://mahout.apache.org/users/recommender/userbased-5-minutes.html
  - https://mahout.apache.org/users/recommender/recommender-documentation.html

Ideas + Solution Approaches

- Finding useful correlations among items beyond the rating network (item correlation by users in terms of their rating similarity)
  - Item feature based correlation
    - Clustering or classification of items
    - …
  - Context based Item Relationships
    - Manufacture, utility, …
- Finding indirect item rating correlations
  - Iterative Propagation (random walk, heat diffusion, etc.)
  - … …

Potential Innovative Project Ideas

Applicability of Mahout for Large Data Sets

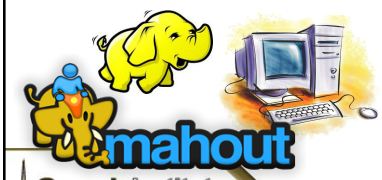
The Data: Million Song Data Set

- Large Data Set
  - 1,019,318 users
  - 384,546 MSD songs
  - 48,373,586 (user, song, count)
- Kaggle Competition: offline evaluation
  - Predict songs a user will listen
    - Training: 1M user listening history
    - Validation: 110K users
- "Martin L" blogged his methodology + results

Motivations

- Can Mahout easily be modified?
- Can Mahout perform well for this workload?
- Can Mahout produce accurate results?
- Can Mahout work 'out of box'?

- Hypothesis: 22 machines + Mahout > 1 guy



x 22 vs.

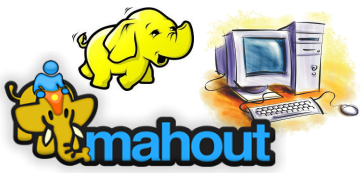
Georgia Institute of Technology

Georgia Tech College of Computing

What kind of Recommender?

- Format: `<userID, songID, count>`
- Users **interacting** with **items**
- Users express **preferences** towards items

- We can use Collaborative Filtering



x 22 vs.

Georgia Institute of Technology

Slides from Michael Sevilla

Georgia Tech College of Computing

Collaborative Filtering

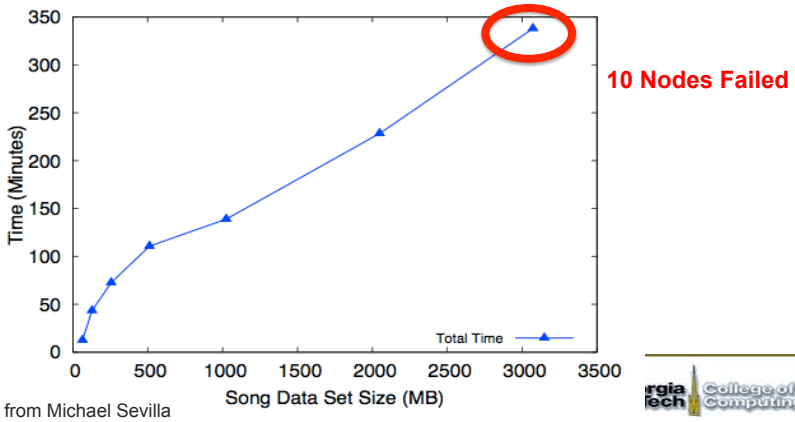
- Predicts preference of user towards an item
- Constructs a Top-N-Recommendation
  1. Parse input rating data
  2. Create user-item-matrix
  3. Predict missing entries

$$\begin{bmatrix} x_{00} & x_{10} & \cdots & & x_{n0} \\ x_{01} & \ddots & & & \\ \vdots & & x_{ij} & & \\ & & & \ddots & \\ x_{0m} & & & & x_{nm} \end{bmatrix}$$

How does Mahout scale the Collaborative Filtering jobs?

Mahout's Code

- Total Time
- ~ 12m, 43m, 1hr, 2hr, 4hr, >5hr ….



10 Nodes Failed

Slides from Michael Sevilla

CF Algorithms: The Scaling Problems (1)

- Matrix Computation
  - Large N users, Large M items
  - N * M matrix is huge
- Memory Resource Constraints
  - Need fast matrix algorithms for cases when the matrix is too big to fit into the memory of a single server
  - Need fast distributed matrix algorithms for cases when the matrix is too big to fit into the shared memory of a compute cluster (say 22 nodes)
- Skewed data distributions
  - Sparse matrix
  - Active users v.s. inactive users
  - Popular items v.s. unpopular items

CF Algorithms: The Scaling Problems (2)

- With Social network enhanced CF algorithms
  - N * M matrix is huge + sizes of the social networks → much higher computation complexity!
- Higher Memory Resource Demand
  - How to partition data and how to combine the intermediate results
  - How to minimize the communication cost for distributed CFs
- The data skewedness is more aggrevated
- How to combine the huge rating matrix with the social network effectively?

Potential Innovative Project Ideas

Tools & Algorithms

- **Collaborative Filtering**
- **Clustering Techniques**
- **Frequent Pattern Mining (Association Rules Mining)**

- **Classification Algorithms**
  - Decision Tree
  - SVM
  - Neural Networks

We Focus On…

- **Clustering** ➔ K-Means
- **Classification** ➔ Naïve Bayes
- **Frequent Pattern Mining** ➔ Apriori

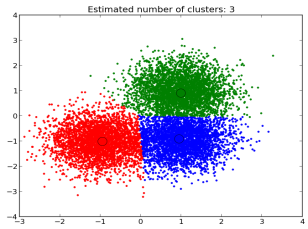
- **Technique logic**
- **How to implement in Hadoop**

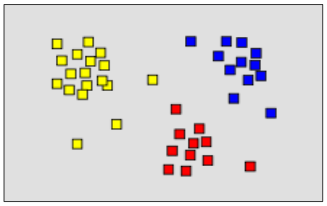
Clustering

- Unsupervised
- Find Natural Groupings
  - Documents
  - Search Results
  - People
  - Genetic traits in groups
  - Many, many more uses

Clustering

- Group similar objects together
- K-Means, Fuzzy K-Means, Density-Based,…
- Different distance measures
  - Manhattan, Euclidean, …





Clustering: An Example

McCain the show horse: Way off track
Seattle Post Intelligencer - **36 minutes ago**
By JOEL CONNELLY ABOARD THE now-jettisoned "Straight Talk Express," Sen. John McCain loved to talk with reporters about go-to heroes in history, none more than Theodore Roosevelt and Winston Churchill.
Candidates scramble to prepare for debate amid bailout crisis CNN International
McCain Decides to Participate in Debate New York Times
BBC News - Voice of America - Washington Post - AFP
all 5,183 news articles »


BBC News

WaMu's Bank Split From Holding Company, Sparing FDIC (Update1)
Bloomberg - **46 minutes ago**
By Linda Shen Sept. 26 (Bloomberg) -- Washington Mutual Inc.'s holding company was detached from its branches and deposits when JPMorgan Chase & Co.
Video: Wall Street watches Washington ReutersVideo
Update: JPMorgan takes over WaMu after snapping up assets Bizjournals.com
Los Angeles Times - CNNMoney.com - Wall Street Journal - MarketWatch
all 2,791 news articles »

Telegraph.co.uk

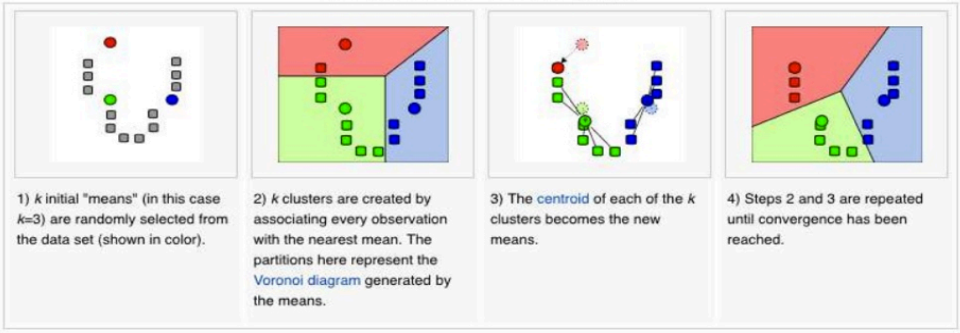
Russia warship heads to Africa after pirate attack
The Associated Press - **1 hour ago**
MOSCOW (AP) - A Russian warship on Friday rushed to intercept a Ukrainian vessel carrying 33 battle tanks and a hoard of ammunition that was seized by pirates off the Horn of Africa - a bold hijacking that again heightened fears about surging piracy ...
Russian Navy ship sent to combat pirates ABC Online
Somali pirates grab Ukrainain ship loaded with tanks Reuters
International Herald Tribune - Voice of America - CNN - Bloomberg
all 561 news articles »


The Southern Ledger

Google News

K-Means Algorithm

Demonstration of the standard algorithm



1) *k* initial "means" (in this case *k*=3) are randomly selected from the data set (shown in color).

2) *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3) The centroid of each of the *k* clusters becomes the new means.

4) Steps 2 and 3 are repeated until convergence has been reached.

Iterative algorithm until converges

K-Means Algorithm (4 steps)

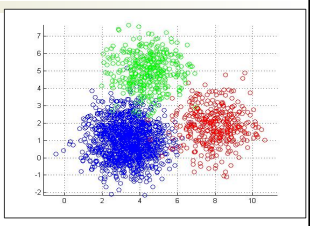
- **Step 1:** Select K points at random (Centers)
- **Step 2**: For each data point, assign it to the closest center
  - Now we formed K clusters
- **Step 3:** For each cluster, re-compute the centers
  - E.g., in the case of 2D points ➔
    - X: average over all x-axis points in the cluster
    - Y: average over all y-axis points in the cluster
- **Step 4:** If the new centers are different from the old centers (previous iteration) ➔ Go to Step 2

K-means Algorithms: Psudo Code

- **Input:** a database *D*, of *m* records, *r*1, ..., *rm* and a desired number of clusters *k*
- **Output:** set of *k* clusters that minimizes the squared error criterion

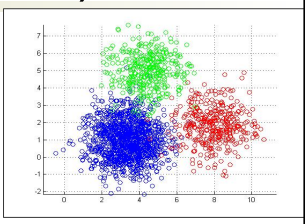
- **Begin**
- Randomly choose k records as the centroids for the k clusters;
- repeat
assign each record ri to a cluster such that the distance between ri
- and the cluster centroid (mean) is the smallest among the k clusters;
recalculate the centroid (mean) for each cluster based on the records
- assigned to the cluster;
- until no change;
- **End**;

K-Means in MapReduce



- **Input**
  - Dataset (set of points in 2D) --Large
  - Initial centroids (K points) --Small
- **Map Side**
  - Each map reads the K-centroids + one block from dataset
  - Assign each point to the closest centroid
  - Output <centroid, point>

K-Means in MapReduce (Cont'd)



- **Reduce Side**
  - Gets all points for a given centroid
  - Re-compute a new centroid for this cluster
  - Output: <new centroid>
- **Iteration Control**
  - Compare the old and new set of K-centroids
    - If similar ➔ Stop
    - Else
      - ★ If max iterations has reached ➔ Stop
      - ★ Else ➔ Start another Map-Reduce Iteration

K-means Clustering in Mahout

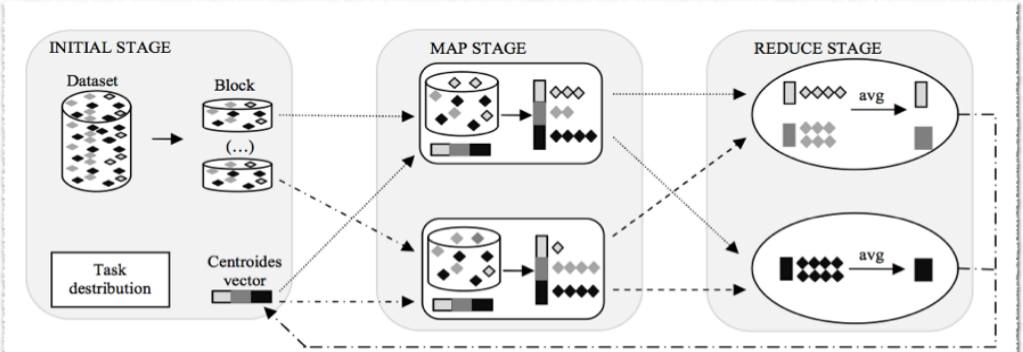


Figure 1 – The three stages of a K-means job.

- [3].K-means Clustering in the Cloud -- A Mahout Test, R. M. Esteves et al.,IEEE Advanced Information Networking and Applications , 2011,

K-means Clustering in Mahout

The dataset is from the 1999 KDD cup.
It has 4,940,000 records, with 41 attributes and 1 label (converted to numerical. A 1.1 GB dataset was used. This file was randomly segmented into smaller files.

Data File (%)	kmeans MN (sec)	kmeans SN (sec)	Iterations to converge	Gain (%)
6	43.9	41.3	72	-6%
12	45.3	52.8	78	16%
25	46.4	88.5	72	91%
50	48.6	149.1	71	207%
100	70.3	316.7	56	351%

Table 1. Times per iteration and gain with the file size. MN - 5 Multi node. SN - Single node.

- [3].K-means Clustering in the Cloud -- A Mahout Test, R. M. Esteves et al.,IEEE Advanced Information Networking and Applications , 2011,

K-means Clustering in Mahout

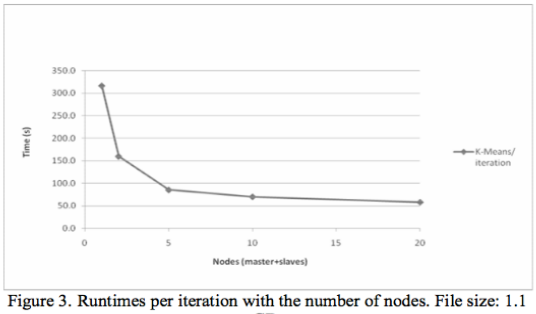


Figure 3. Runtimes per iteration with the number of nodes. File size: 1.1 GB.

Nodes	K-means (min)	Gain (%)
1	296	
2	149	98%
5	80	271%
10	66	351%
20	54	444%

Table 2. Converging times varying the number of nodes. File size: 1.1 GB. Gain comparing with 1 node.

- [3].K-means Clustering in the Cloud -- A Mahout Test, R. M. Esteves et al.,IEEE Advanced Information Networking and Applications , 2011,

Tools & Algorithms

- **Collaborative Filtering**
- **Clustering Techniques**
- **Frequent Pattern Mining (Association Rules Mining)**
- **Classification Algorithms**
  - Decision Tree
  - SVM
  - Neural Networks

Frequent Pattern Mining

- Association Rules Mining
- Very common problem in Market-Basket applications
- Given a set of items I ={milk, bread, jelly, …}
- Given a set of transactions where each transaction contains a subset of items
  - t1 = {milk, bread, water}
  - t2 = {milk, nuts, butter, rice}

Frequent Pattern Mining

- Given a set of items I ={milk, bread, jelly, …}
- Given a set of transactions where each transaction contains subset of items
  - t1 = {milk, bread, water}
  - t2 = {milk, nuts, butter, rice}

What are the itemsets frequently sold together ??

% of transactions in which the itemset appears >= α

Common Use Cases

- Recommend friends/dates/products
- Classify content into predefined groups
- Find similar content
- Find associations/patterns in actions/behaviors
- Identify key topics/summarize text
  - Documents and Corpora
- Detect anomalies/fraud
- Ranking search results
- Others?

FRM and Association Rule Mining

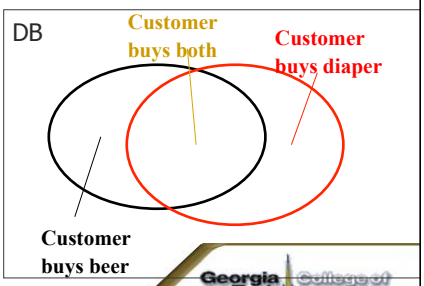
- Association rule mining:
  - Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transactional databases, relational databases, and other information repositories
- Examples
  - People buy beers on Fridays will usually also buy popcorns.
  - People with symptom X often will develop symptom Y over the time period T.
  - What is the percentage of students who took both cs6675AIA and cs8803BDS?

Association Rule Problem

- Given a set of items $I=\{I_1,I_2,...,I_m\}$ and a database of transactions $D=\{t_1,t_2, ..., t_n\}$ where $t_i=\{I_{i1},I_{i2}, ..., I_{ik}\}$ and $I_{ij} \in I$, the ***Association Rule Problem*** is to identify all association rules $X \Rightarrow Y$ with a minimum support and confidence.

NOTE:

- Support(X)=count(X)
- Support of $X \Rightarrow Y$ is same as support of $X \cup Y$.



Association Rule Techniques

1. Find Large Frequent Itemsets.
2. Generate rules from frequent itemsets.

Sup(X→Y) = Sup(XUY) > minsupport

$$\text{conf}(X \rightarrow Y) = \frac{\sup(X \cup Y)}{\sup(X)} > \text{minconf.}$$

Association Rules Example

Transaction	Items
t_1	Bread,Jelly,PeanutButter
t_2	Bread,PeanutButter
t_3	Bread,Milk,PeanutButter
t_4	Beer,Bread
t_5	Beer,Milk

$$conf(X \Rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)}$$

I = { Beer, Bread, Jelly, Milk, PeanutButter}

Support of Bread→PeanutButter is 60%

3/5 transactions buy bread and peanutbutter

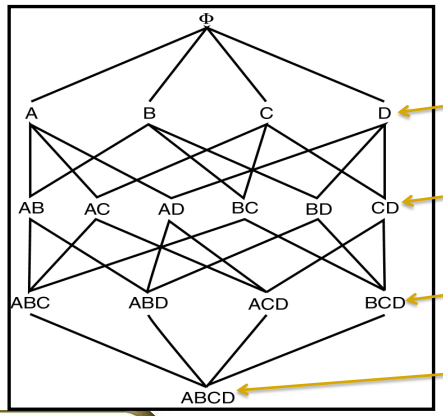
Confidence of Bread→PeanutButter is 75%

among 4 buy bread, 3/4 buys bread and peanutbutter

How to find frequent item sets

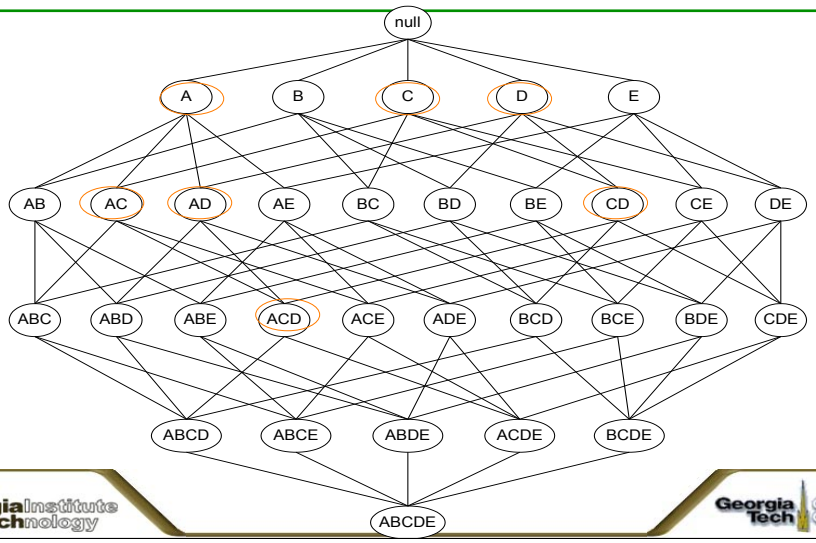
- **Naïve Approach**
  - Enumerate all possible itemsets and then count each one

Finding frequent item-sets for a given set of transactions is computationally expensive

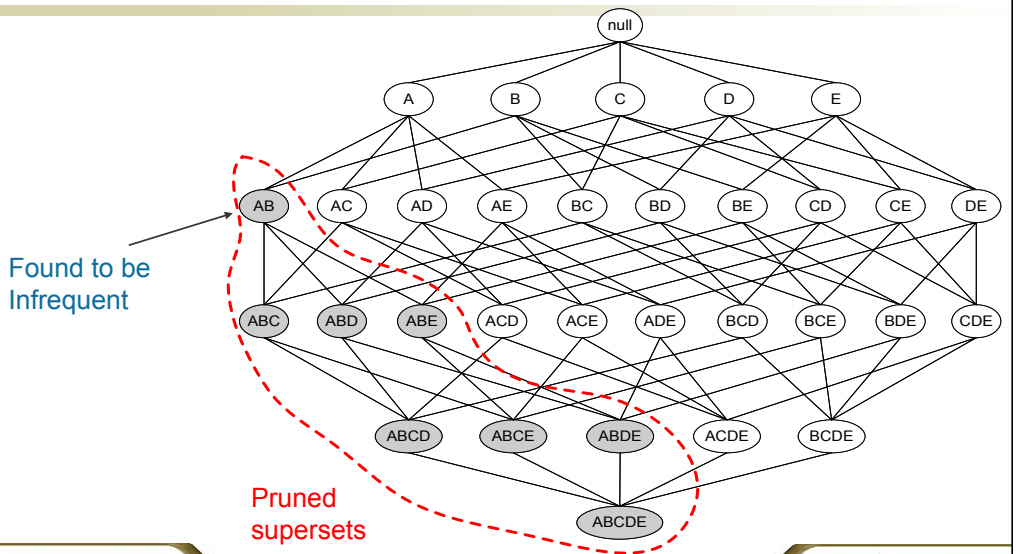


Can we optimize? Apriori Optimization

Apriori Principle: All subsets of a frequent itemset must be frequent



Illustrating Apriori Principle



Factors Affecting Complexity

- Choice of minimum support threshold
  - lowering support threshold results in more frequent itemsets
  - this may increase number of candidates and max length of frequent itemsets
- Dimensionality (number of items) of the data set
  - more space is needed to store support count of each item
  - if number of frequent items also increases, both computation and I/O costs may also increase
- Size of database
  - since Apriori makes multiple passes, run time of algorithm may increase with number of transactions
- Average transaction width
  - transaction width increases with denser data sets
  - This may increase max length of frequent itemsets and traversals of hash tree (number of subsets in a transaction increases with its width)

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, 14 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,		

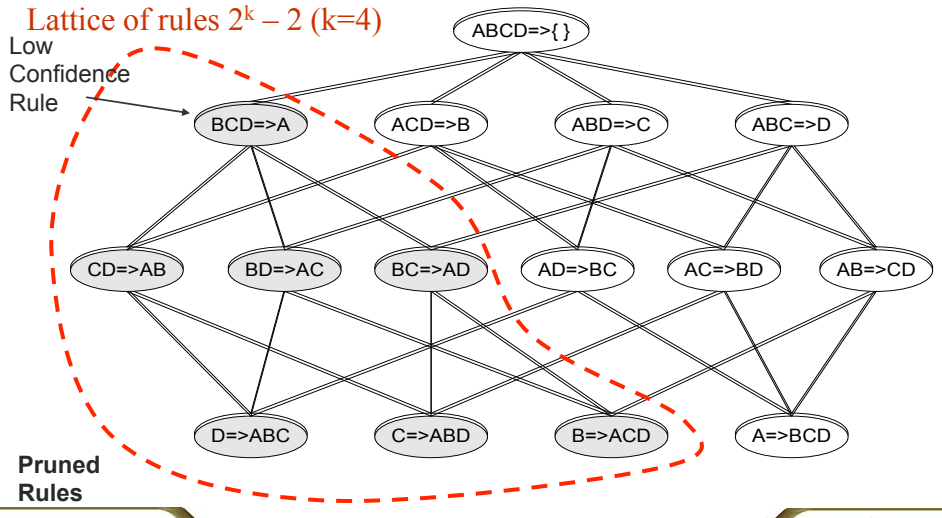
- If |L| = k, then there are $2^k - 2$ candidate association rules (ignoring L $\rightarrow \varnothing$ and $\varnothing \rightarrow$ L)

Rule Generation: Optimization

- How to efficiently generate rules from frequent itemsets?
  - In general, confidence does not have an anti-monotone property
    c(ABC →D) can be larger or smaller than c(AB →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 for Apriori Algorithm



Summary: Reducing Association Rule Complexity

- Two properties are used to reduce the search space for association rule generation.
  - **Downward Closure**
    - A subset of a large itemset must also be large
  - **Anti-monotonicity**
    - A superset of a small itemset is also small. This implies that the itemset does not have sufficient support to be considered for rule generation.

Apriori Algorithm

- **Executes in scans (iterations), each scan has two phases**
  - Given a list of candidate itemsets of size n, count their appearance and find frequent ones
  - From the frequent ones generate candidates of size n+1 *(previous property must hold)*
    - All subsets of size n must be frequent to be a candidate
  - Start the algorithm where n =1, then repeat

Use the property to reduce the number of itemsets to check

FPM in Hadoop

- Can Mahout be applicable to Large Data Sets?
- How to efficiently implement FPM as map-reduce jobs?

Available DataSets

- UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets.html
- ArXiv (http://arxiv.org): Open access to 965,645 e-prints
- Yelp Dataset Challenge! http://www.yelp.com/dataset_challenge/
- Million Song Dataset Challenge http://www.kaggle.com/c/msdchallenge
- TREC Datasets for information retrieval
- Web Graphs (http://law.di.unimi.it/datasets.php)

Resource

[1]Anil, Robin, Ted Dunning, and Ellen Friedman. Mahout in action. Manning, 2011.
[2] http://www.orzota.com/apache-mahout-and-machine-learning/
[3] K-means Clustering in the Cloud -- A Mahout Test, R. M. Esteves et al.,IEEE Advanced Information Networking and Applications , 2011,
[4] https://mahout.apache.org/
[5] http://www.ibm.com/developerworks/java/library/j-mahout/
[6] Sean Owen, Robin Anil, Ted Dunning and Ellen Friedman,Mahout in action,Manning Publications; Pap/Psc edition (October 14, 2011)
[7] From Mahout Hands on, by Ted Dunning and Robin Anil, OSCON 2011, Portland
[8]"Programming Collective Intelligence" by Segaran
[9]"Data Mining - Practical Machine Learning Tools and Techniques" by Witten and Frank
[10]"Taming Text" by Ingersoll and Morton
[11]http://lucene.apache.org/mahout

- Hadoop - http://hadoop.apache.org

[12]http://cwiki.apache.org/MAHOUT
[13]mahout-{user|dev}@lucene.apache.org

- http://www.lucidimagination.com/search/p:mahout

Questions