MAPPING THE FABRIC, THE GARMENT AND THE TOOLS

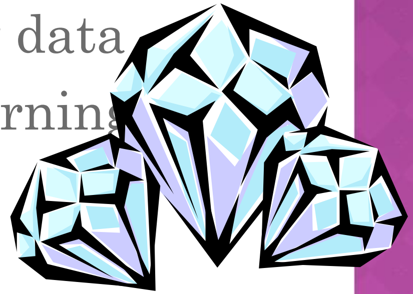
Dr. Anala Pandit
VJTI, Mumbai

Some Slides are taken from: Mining Massive Datasets course on Coursera

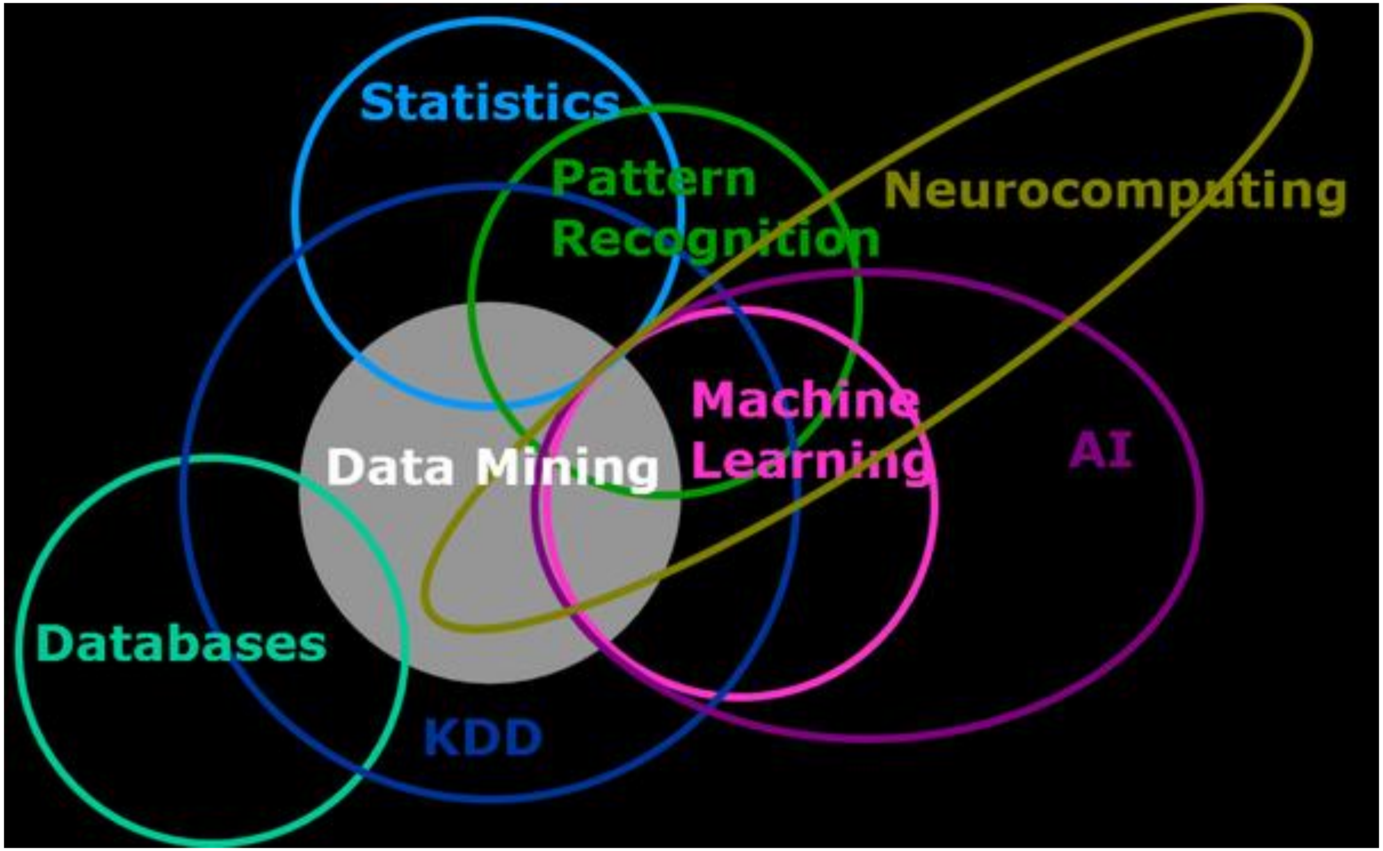
WHAT IS DATA MINING?

- Data mining (knowledge discovery in databases):
  - Extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) information or patterns from data in large databases

- Alternative names and their "inside stories":
  - Data mining: a misnomer?
  - Knowledge discovery(mining) in databases (KDD), knowledge extraction, data/pattern analysis, data archeology, business intelligence, exploratory data analysis, data driven discovery, deductive learning

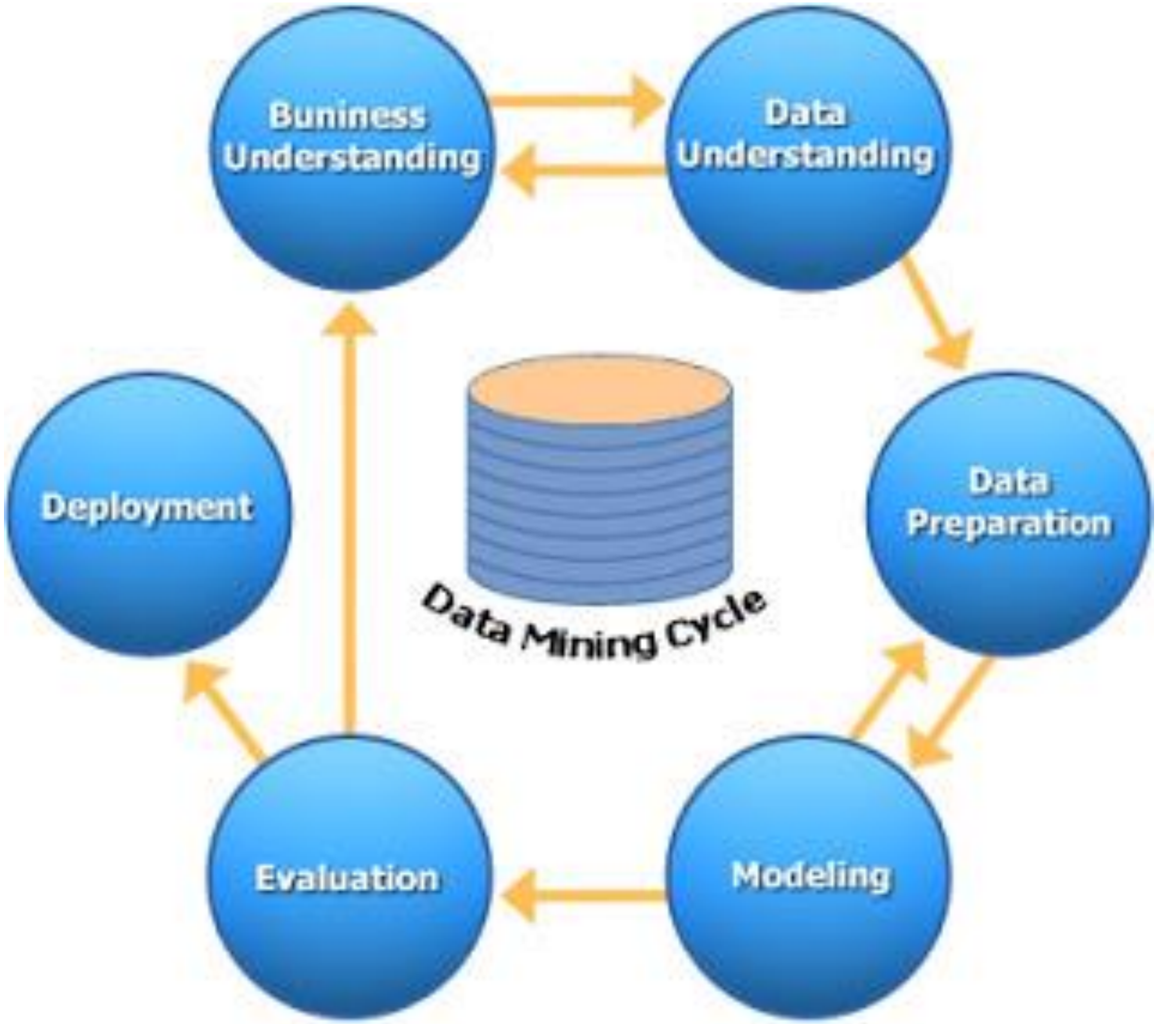


RELATED FIELDS



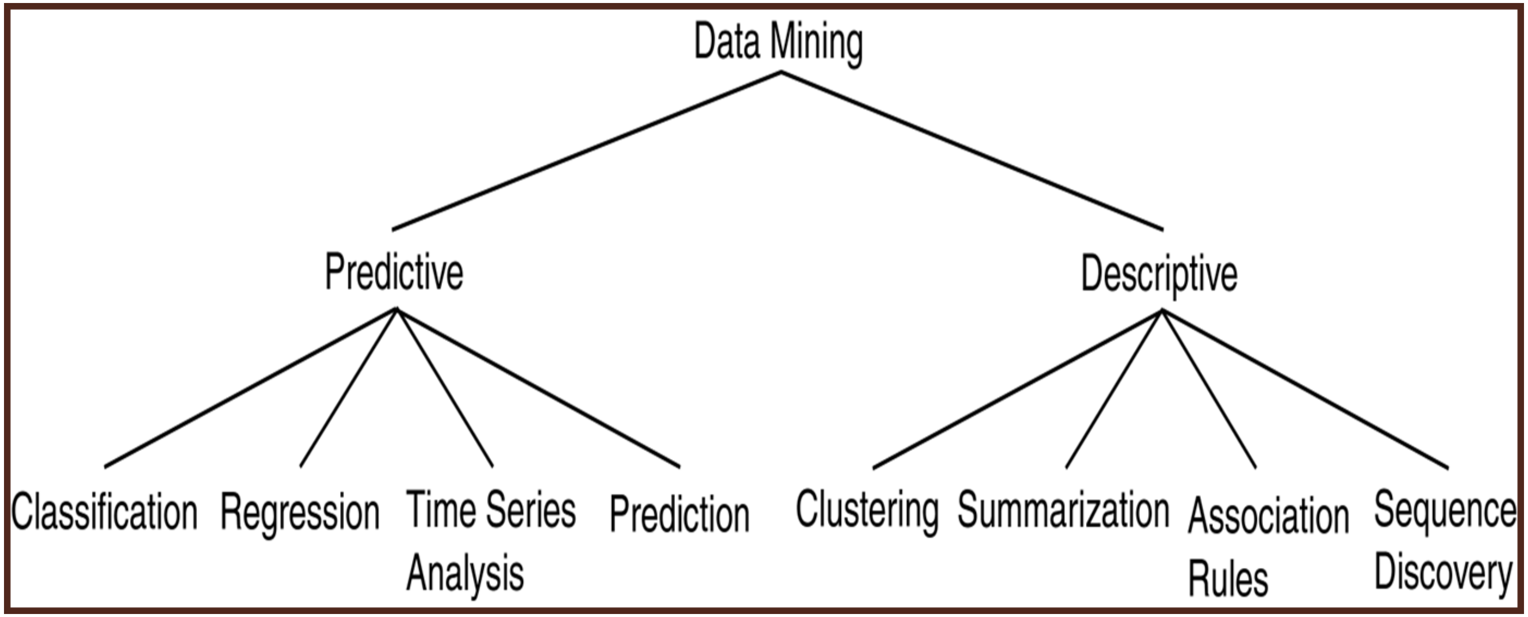
Source: https://www.quora.com/What-is-the-difference-between-artificial-intelligence-machine-learning-data-mining-and-business-intelligence-How-they-are-related

DATA MINING CYCLE

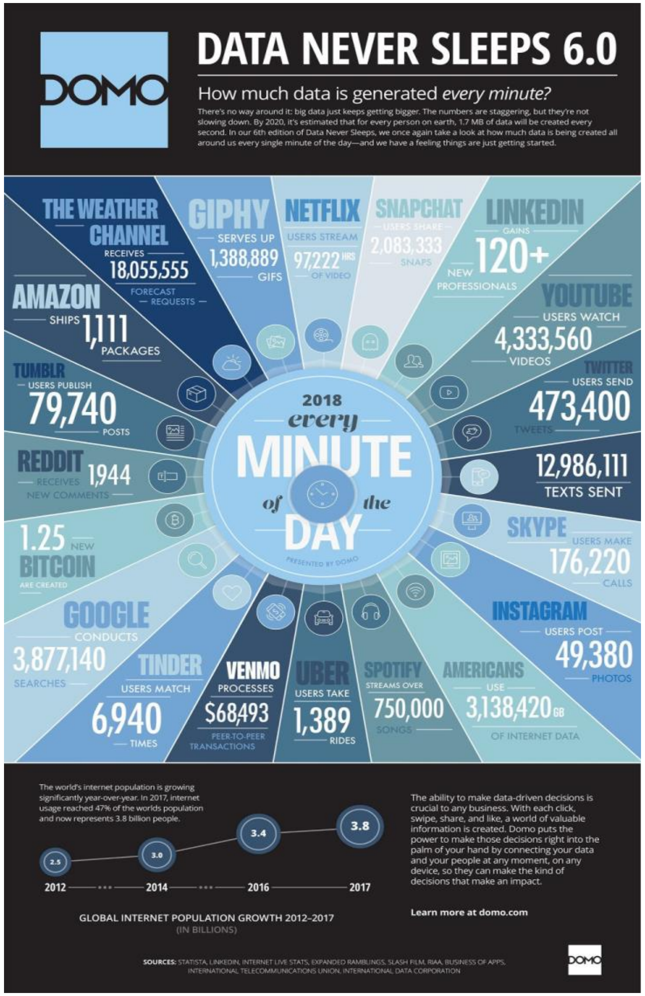


Source: http://webdataextraction.blogspot.com/2016/01/importance-of-data-mining-service-for.html

DATA MINING MODELS AND TASKS



WHAT IS BIG DATA



DATA NEVER SLEEPS 6.0

How much data is generated *every minute?*

There's no way around it: big data just keeps getting bigger. The numbers are staggering, but they're not slowing down. By 2020, it's estimated that for every person on earth, 1.7 MB of data will be created every second. In our 6th edition of Data Never Sleeps, we once again take a look at how much data is being created all around us every single minute of the day—and we have a feeling things are just getting started.

DOMO
THE WEATHER CHANNEL RECEIVES 18,055,555 FORECAST REQUESTS
GIPHY SERVES UP 1,388,889 GIFS
NETFLIX USERS STREAM 97,222 HRS OF VIDEO
SNAPCHAT USERS SHARE 2,083,333 SNAPS
LINKEDIN GAINS 120+ NEW PROFESSIONALS
AMAZON SHIPS 1,111 PACKAGES
YOUTUBE USERS WATCH 4,333,560 VIDEOS
TUMBLR USERS PUBLISH 79,740 POSTS
TWITTER USERS SEND 473,400 TWEETS
REDDIT RECEIVES 1944 NEW COMMENTS
12,986,111 TEXTS SENT
1.25 NEW BITCOIN ARE CREATED
SKYPE USERS MAKE 176,220 CALLS
GOOGLE CONDUCTS 3,877,140 SEARCHES
INSTAGRAM USERS POST 49,380 PHOTOS
TINDER USERS MATCH 6,940 TIMES
VENMO PROCESSES $68,493 PEER-TO-PEER TRANSACTIONS
UBER USERS TAKE 1,389 RIDES
SPOTIFY STREAMS OVER 750,000 SONGS
AMERICANS USE 3,138,420 GB OF INTERNET DATA
2018 every MINUTE of the DAY
PRESENTED BY DOMO
2.5
3.0
3.4
3.8
2012
2014
2016
2017
GLOBAL INTERNET POPULATION GROWTH 2012–2017
(IN BILLIONS)

The world's internet population is growing significantly year-over-year. In 2017, internet usage reached 47% of the worlds population and now represents 3.8 billion people.

The ability to make data-driven decisions is crucial to any business. With each click, swipe, share, and like, a world of valuable information is created. Domo puts the power to make those decisions right into the palm of your hand by connecting your data and your people at any moment, on any device, so they can make the kind of decisions that make an impact.

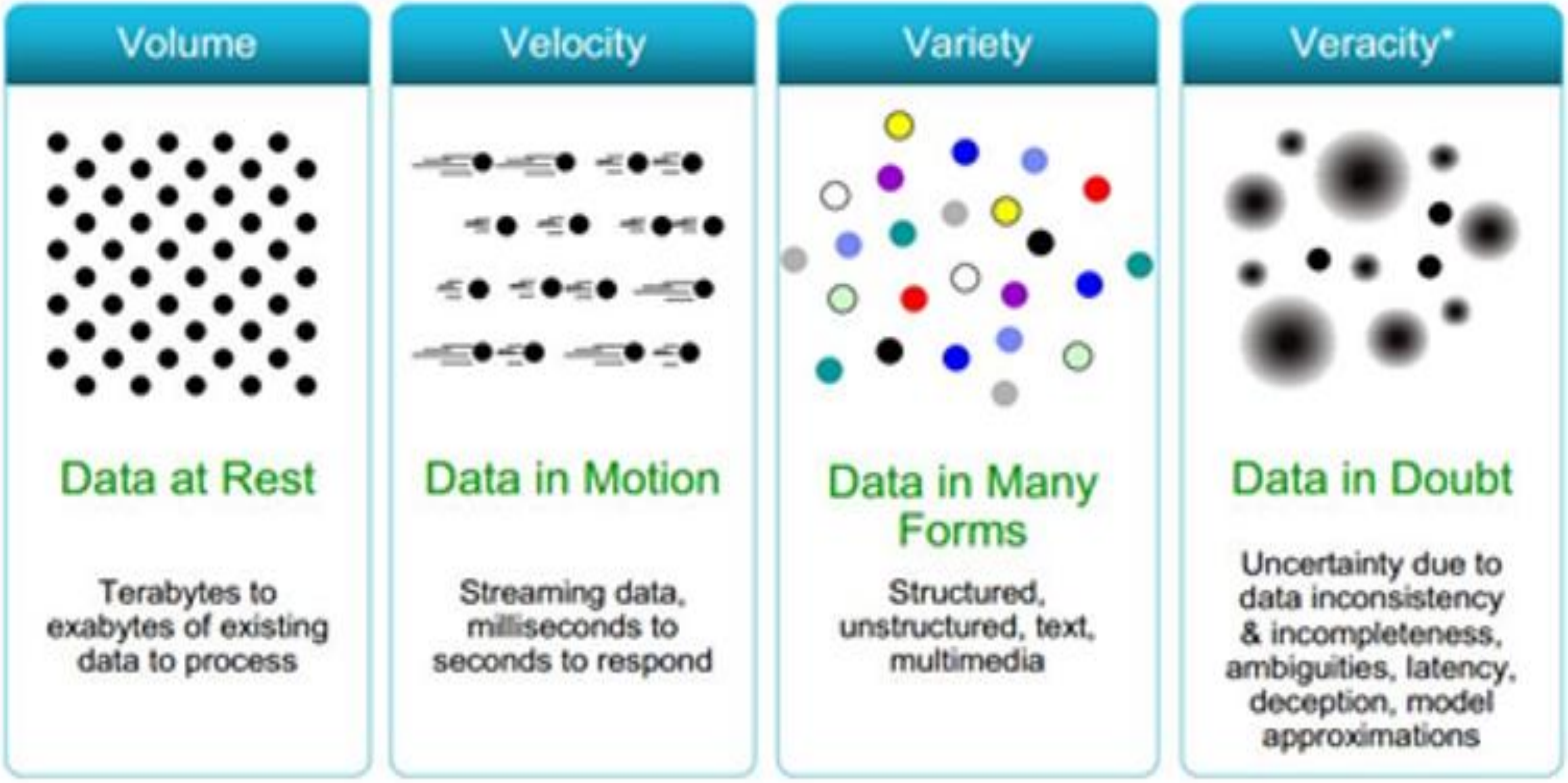
Learn more at domo.com

SOURCES: STATISTA, LINKEDIN, INTERNET LIVE STATS, EXPANDED RAMBLINGS, SLASH FILM, RIAA, BUSINESS OF APPS, INTERNATIONAL TELECOMMUNICATIONS UNION, INTERNATIONAL DATA CORPORATION

"Big data is high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making." -- Gartner

Over 2.5 quintillion bytes of data are created every single day, and it's only going to grow from there. By 2020, it's estimated that 1.7MB of data will be created every second for every person on earth."

CHARACTERISTICS OF BIG DATA



Volume

Data at Rest

Terabytes to exabytes of existing data to process

Velocity

Data in Motion

Streaming data, milliseconds to seconds to respond

Variety

Data in Many Forms

Structured, unstructured, text, multimedia

Veracity*

Data in Doubt

Uncertainty due to data inconsistency & incompleteness, ambiguities, latency, deception, model approximations

ISSUES IN APPLYING DATA MINING ALGORITHMS TO BIG DATA

- Data mining Algorithms require the data to be in main memory for processing.
- However, the Big data cannot fit into main memory.
- So you may need to fetch the relevant data from disk for every iteration of processing
- → Disk thrashing?
- Other issues like: Unstructured nature of data, missing, noisy data. These can be handled during Data Preparation phase

HOW TO HANDLE?

OBJECTIVE:

- To obtain results for large data with minimal disk accesses and highest possible accuracy.
- In short: Map the fabric with the garment you want to create and the tool you want to use to create

SOLUTIONS:

- Use samples.
- Partition and run algorithms on each partition and combine the results. (Map Reduce?)
- Summarize the results of each partition and use summaries to combine the results.
- Find variants based on problem such that space complexity is reduced.

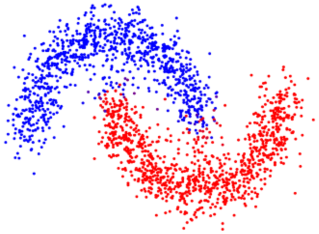
EXAMPLE 1

- **Clustering**: Distance calculations require all points to be in memory. Initially k centroids (clusters) are selected. Distance is calculated of each point from centroid and point is assigned to closest cluster. Centroids are recalculated and process is repeated until the clusters are stable. Complexity is O(knId).

THE APPROACHES

- If the data is normally distributed and clusters are expected to be aligned with the axis then, **Bradley Fayyad Reina or BFR algorithm**

- If the data is not normally distributed and clusters are not aligned with the axis,
- Centroid-based approach (using d_{mean}) considers only one point as representative of a cluster - the cluster centroid.
- All-points approach (based on d_{min}) makes the clustering algorithm extremely sensitive to outliers and to slight changes in the position of data points.
- Both of them can't work well for non-spherical or arbitrary shaped clusters, **use CURE algorithm**

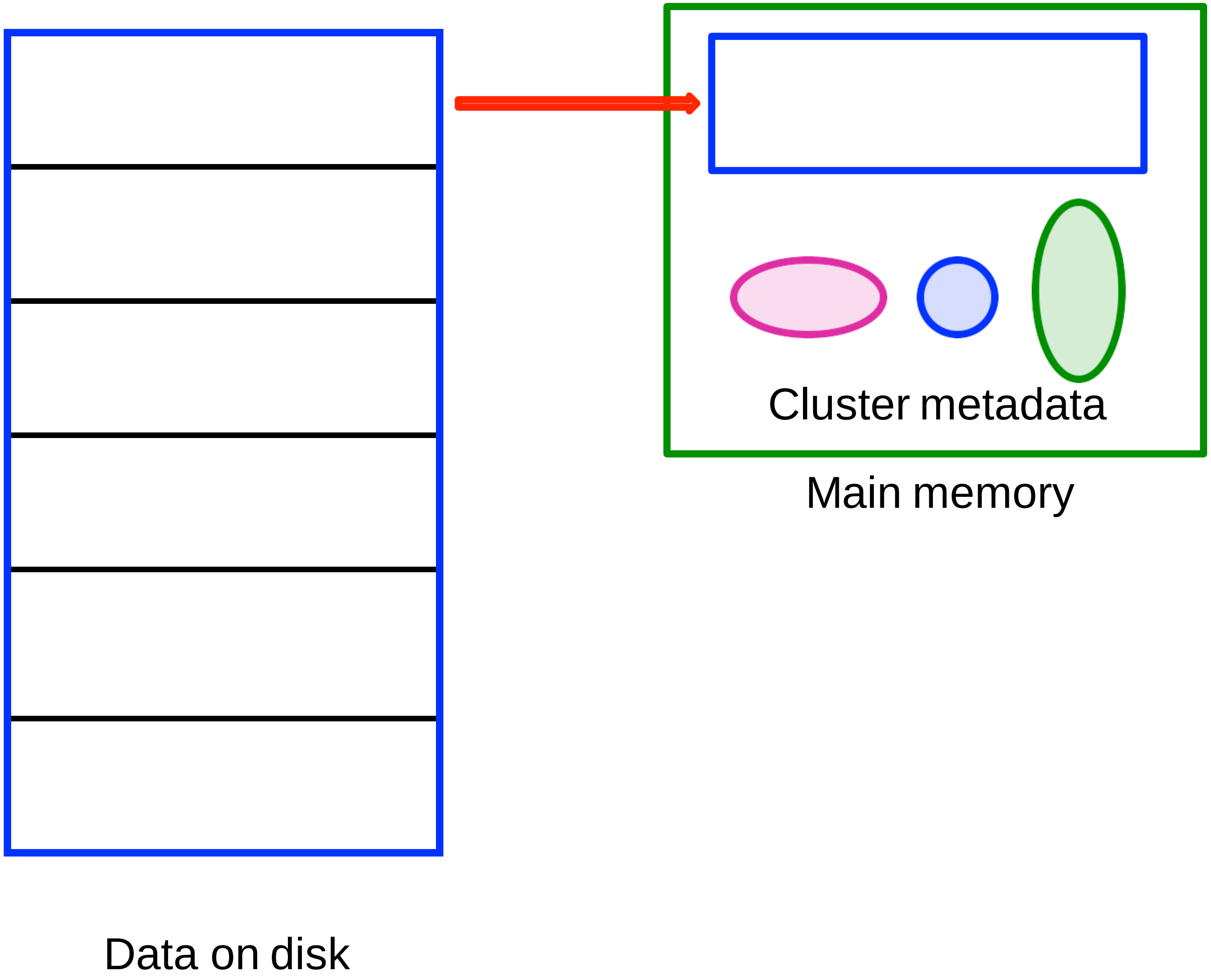


BFR ALGORITHM

Track of 3 sets of points

- **Discard set (DS):**
  - Points close enough to a centroid to be summarized
- **Compression set (CS):**
  - Groups of points that are close together but not close to any existing centroid
  - These points are summarized, but not assigned to a cluster
- **Retained set (RS):**
  - Isolated points waiting to be assigned to a compression set

BFR Algorithm: Overview



THE "MEMORY-LOAD" OF POINTS

- Find those points that are "**sufficiently close**" to a cluster centroid and add those points to that cluster and the **DS**
  - These points are so close to the centroid that they can be summarized and then discarded

- Use any main-memory clustering algorithm to cluster the remaining points and the old **RS**
  - Clusters go to the **CS**; outlying points to the **RS**

THE "MEMORY-LOAD" OF POINTS

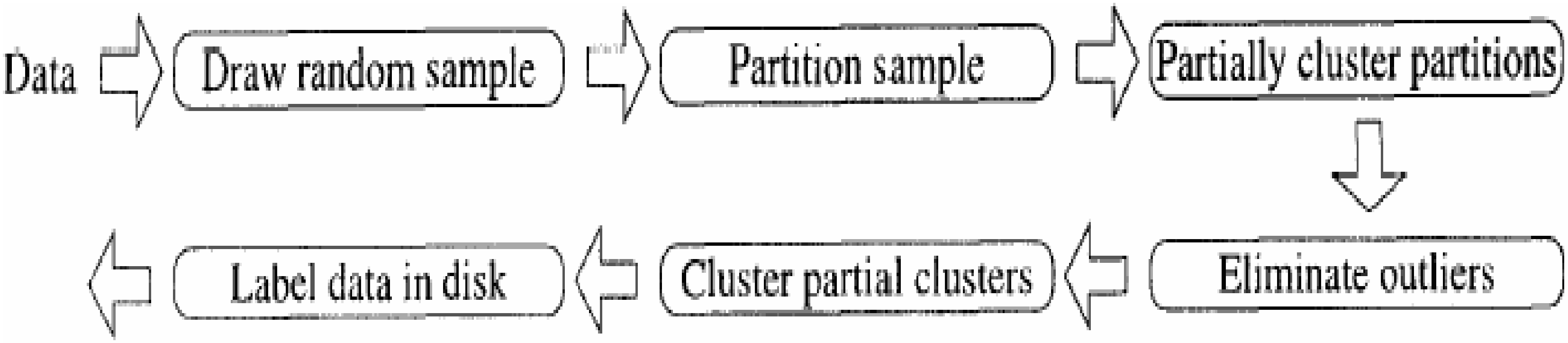
- **DS set:** Adjust statistics of the clusters to account for the new points
  - Add ***N***s, ***SUM***s, ***SUMSQ***s
- Consider merging compressed sets in the **CS**
- If this is the last round, merge all compressed sets in the **CS** and all **RS** points into their nearest cluster

THE CURE ALGORITHM

- **CURE (Clustering Using REpresentatives):**
  - Assumes a Euclidean distance
  - Allows clusters to assume any shape
  - **Uses a collection of representative points to represent clusters**

  Contributions:
- CURE can identify both spherical and non-spherical clusters.
  - It chooses a number of well scattered points as *representatives* of the cluster instead of one point - centroid.
- CURE uses *random sampling* and *partitioning* to speed up clustering.

OVERVIEW OF CURE



Data

Draw random sample

Partition sample

Partially cluster partitions

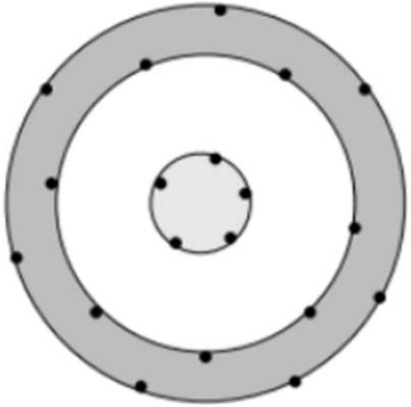
Eliminate outliers

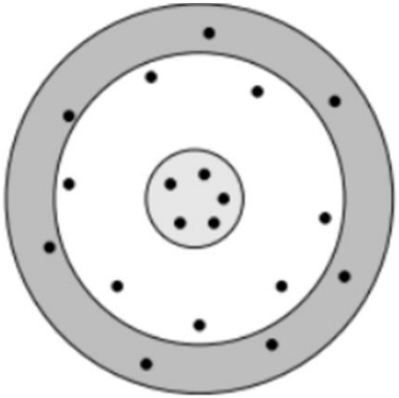
Cluster partial clusters

Label data in disk

CURE ALGORITHM: HIERARCHICAL CLUSTERING ALGORITHM

- For each cluster, m well scattered points within the cluster are chosen , and then shrinking them toward the mean of the cluster by a fraction α.
- The distance between two clusters is then the distance between the closest pair of representative points from each cluster.
- The m representative points attempt to capture the physical shape and geometry of the cluster. **Shrinking** the scattered points by fraction **α** toward the mean gets rid of surface abnormalities and mitigates the effects of outliers.





CURE ALGORITHM: PARTITIONING FOR SPEEDUP

- In the first pass:
  - Partition the sample space into p partitions, each of size n/p.
  - Partially cluster each partition until the final number of clusters in each partition reduces to n/pq, q > 1.
  - If we predefine the number of clusters to k then total number of representative points will be mk*p
  - Eg: m = 4, k =100, p =5 then clustering problem now has 2000 points only.
  - Run CURE on these 2000 points. But make sure that the k you have chosen in earlier step is much larger than the number of final clusters that you want

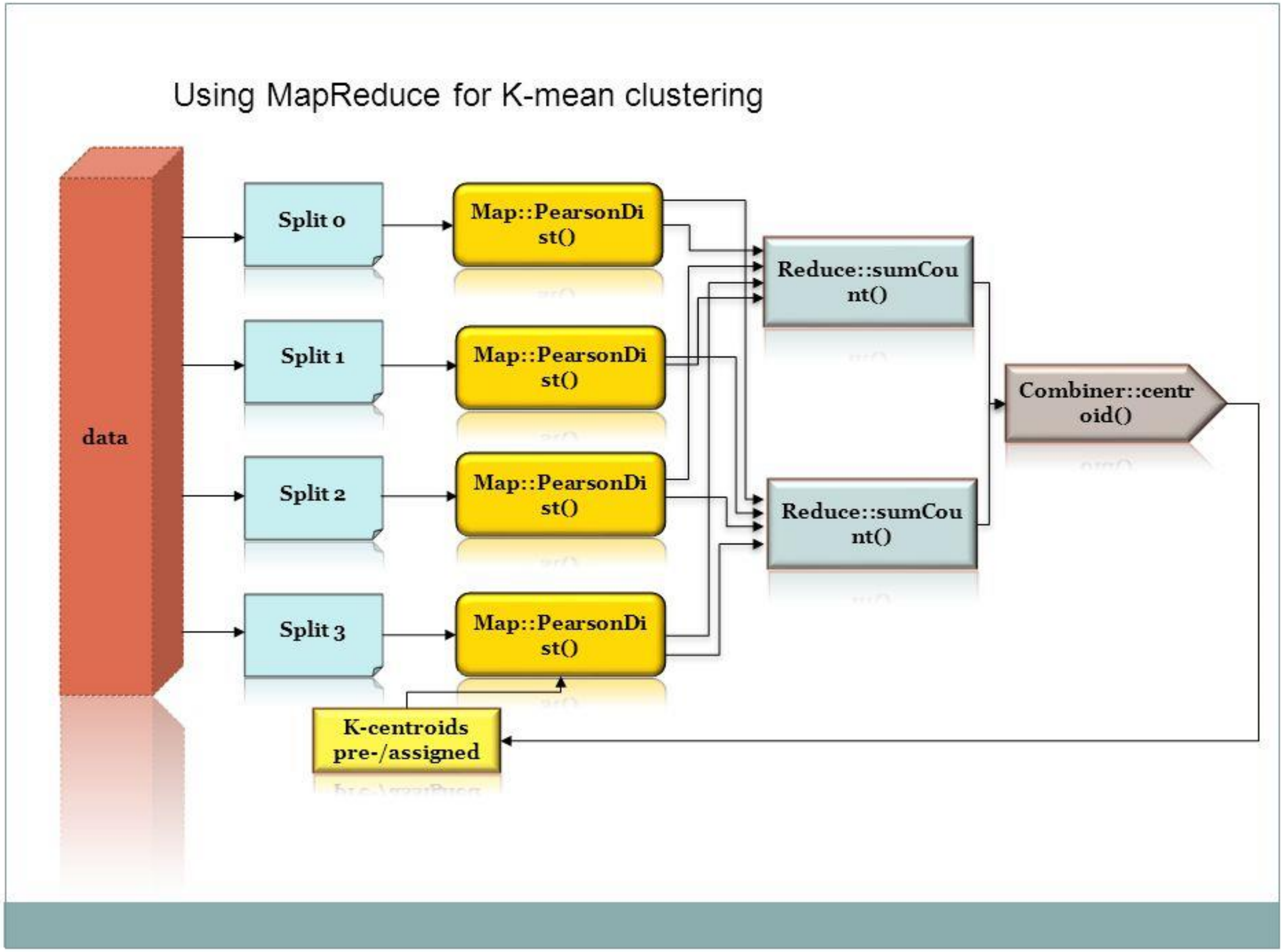
CURE ALGORITHM: LABELING DATA ON DISK

- Process of Sampling has excluded majority of data points and they must be assigned to the clusters formed in the earlier phase
- Each data point in the secondary memory is assigned to the cluster containing the representative point closest to it.
- The last step of CURE is point assignment. Each point p is brought from secondary storage and compared with the representative points. We assign p to the cluster of the representative point that is closest to p.

ADVANTAGES

- Reduce execution time
- Reduce the input size and ensure it fits in main-memory by storing only the representative points for each cluster as input to the clustering algorithm.

K MEANS: MAP REDUCE VERSION



Source: Haddop File system : B. Ramamurthy

EXAMPLE 2

- **Association rule mining:**
  - Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.

- **Applications:**
  - Basket data analysis, cross-marketing, catalog design, loss-leader analysis, clustering, classification, etc.

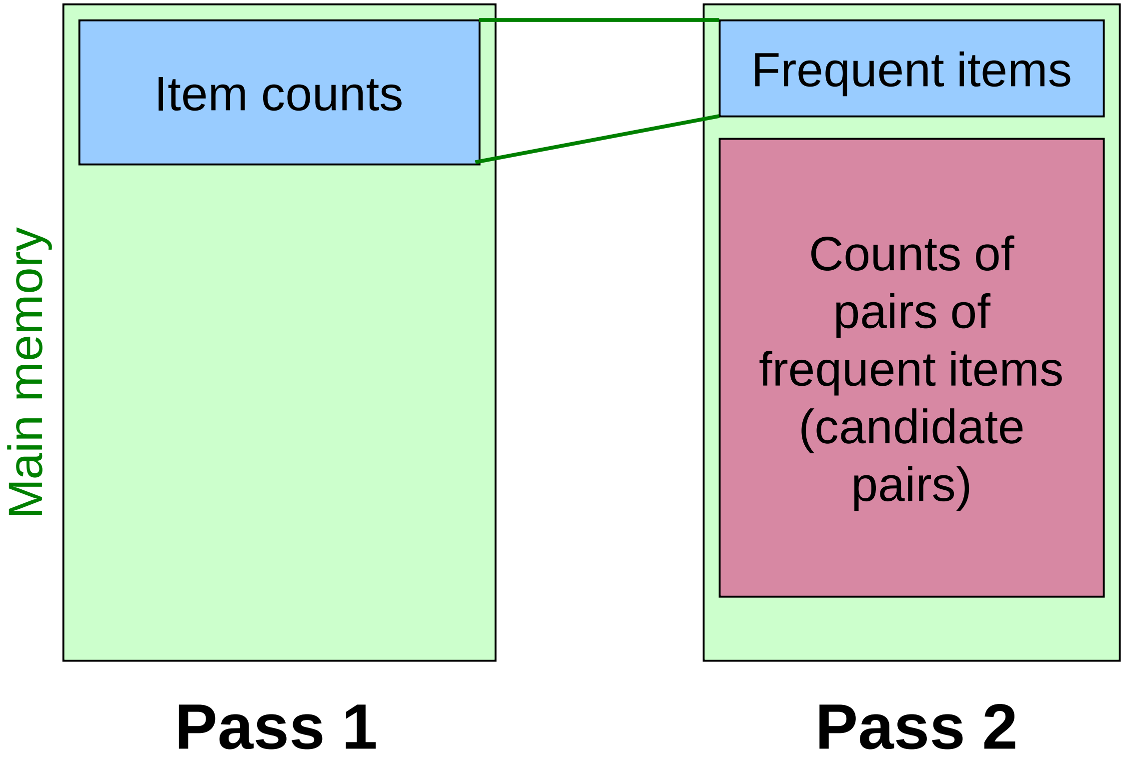
COMPUTATION MODEL

- The true cost of mining disk-resident data is usually the **number of disk I/Os**
- In practice, association-rule algorithms read the data in ***passes*** – all baskets read in turn
- Measure the cost by the **number of passes** an algorithm makes over the data
- Main Memory bottleneck. Why?

FINDING FREQUENT PAIRS

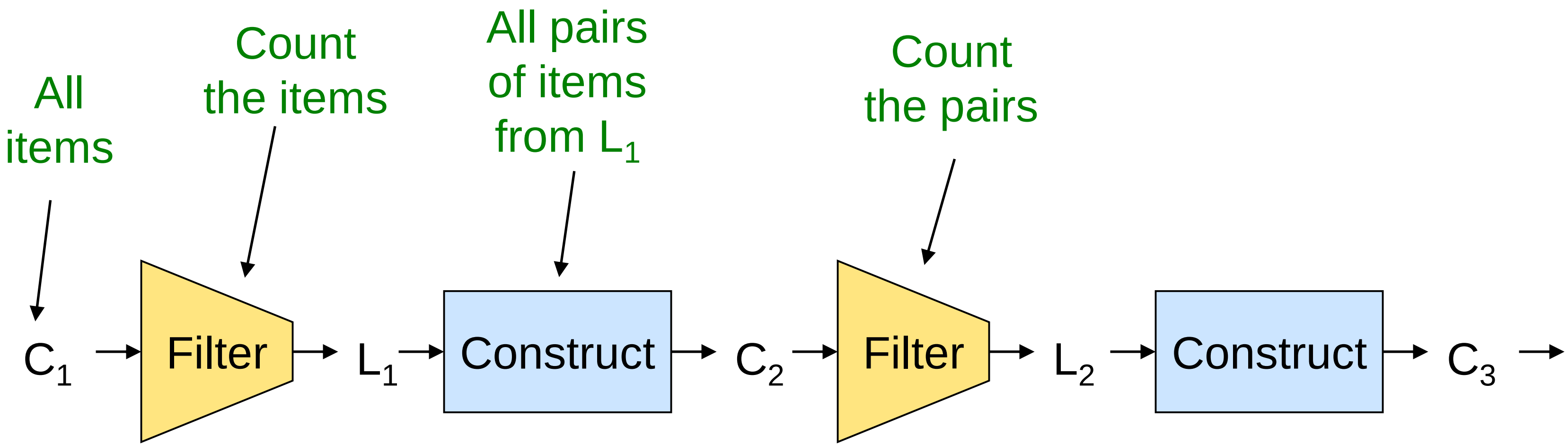
- **The hardest problem often turns out to be finding the frequent pairs of items {*i_1*, *i_2*}**
- **So, first concentrate on pairs, then extend to larger sets**
- **The approach:**
  - One always need to generate all the itemsets
  - But only count (keep track) of those itemsets that in the end turn out to be frequent

MAIN-MEMORY: PICTURE OF A-PRIORI

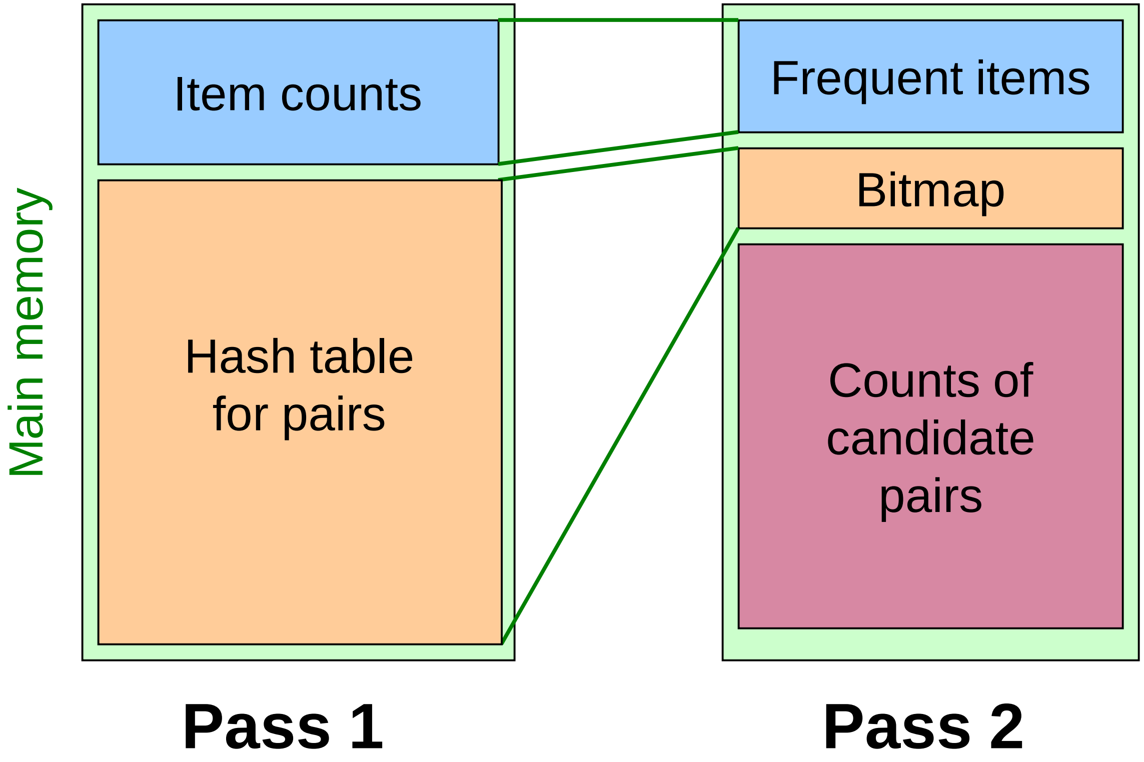


FREQUENT TRIPLES, ETC.

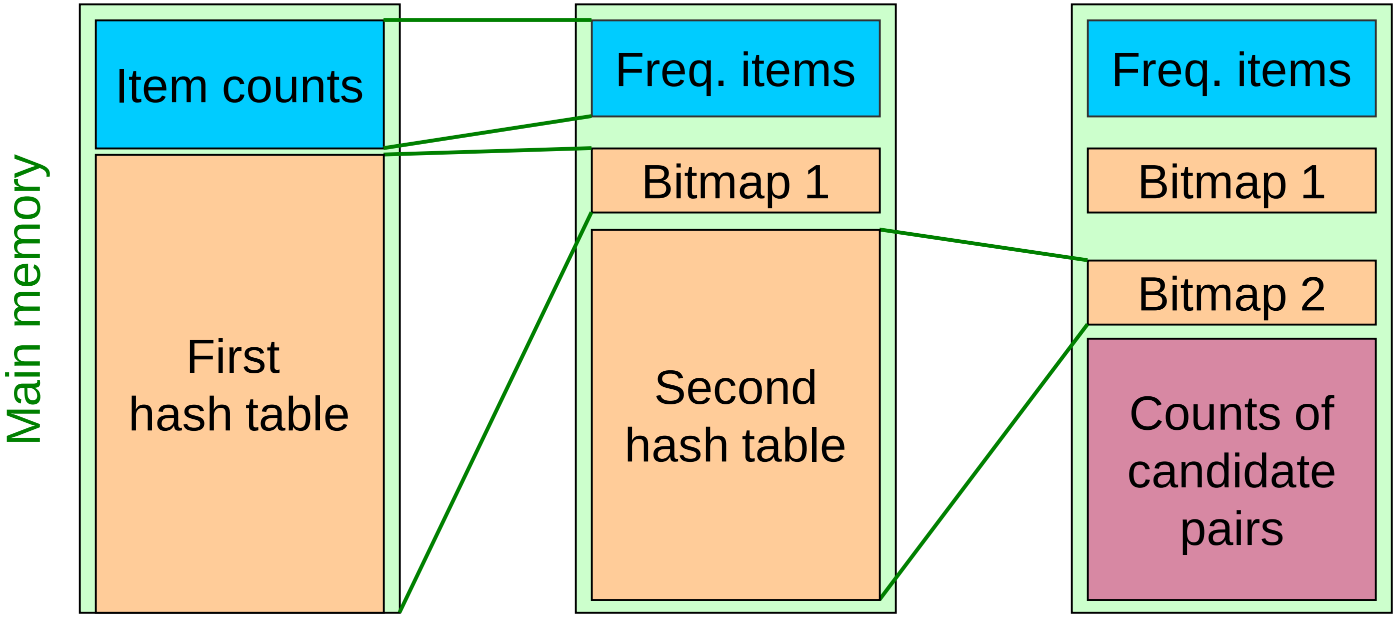
- **For each *k*, we construct two sets of *k-tuples*** (sets of size *k*):
  - C_k = ***candidate k-tuples*** = those that might be frequent sets (support ≥ **s**) based on information from the pass for ***k*–1**
  - L_k = the set of truly frequent ***k***-tuples



Main-Memory: Picture of PCY



MAIN-MEMORY: MULTISTAGE



Pass 1

Count items
Hash pairs {i,j}

Pass 2

Hash pairs {i,j}
into Hash2 iff:
i,j are frequent,
{i,j} hashes to
freq. bucket in B1

Pass 3

Count pairs {i,j} iff:
i,j are frequent,
{i,j} hashes to
freq. bucket in B1
{i,j} hashes to
freq. bucket in B2

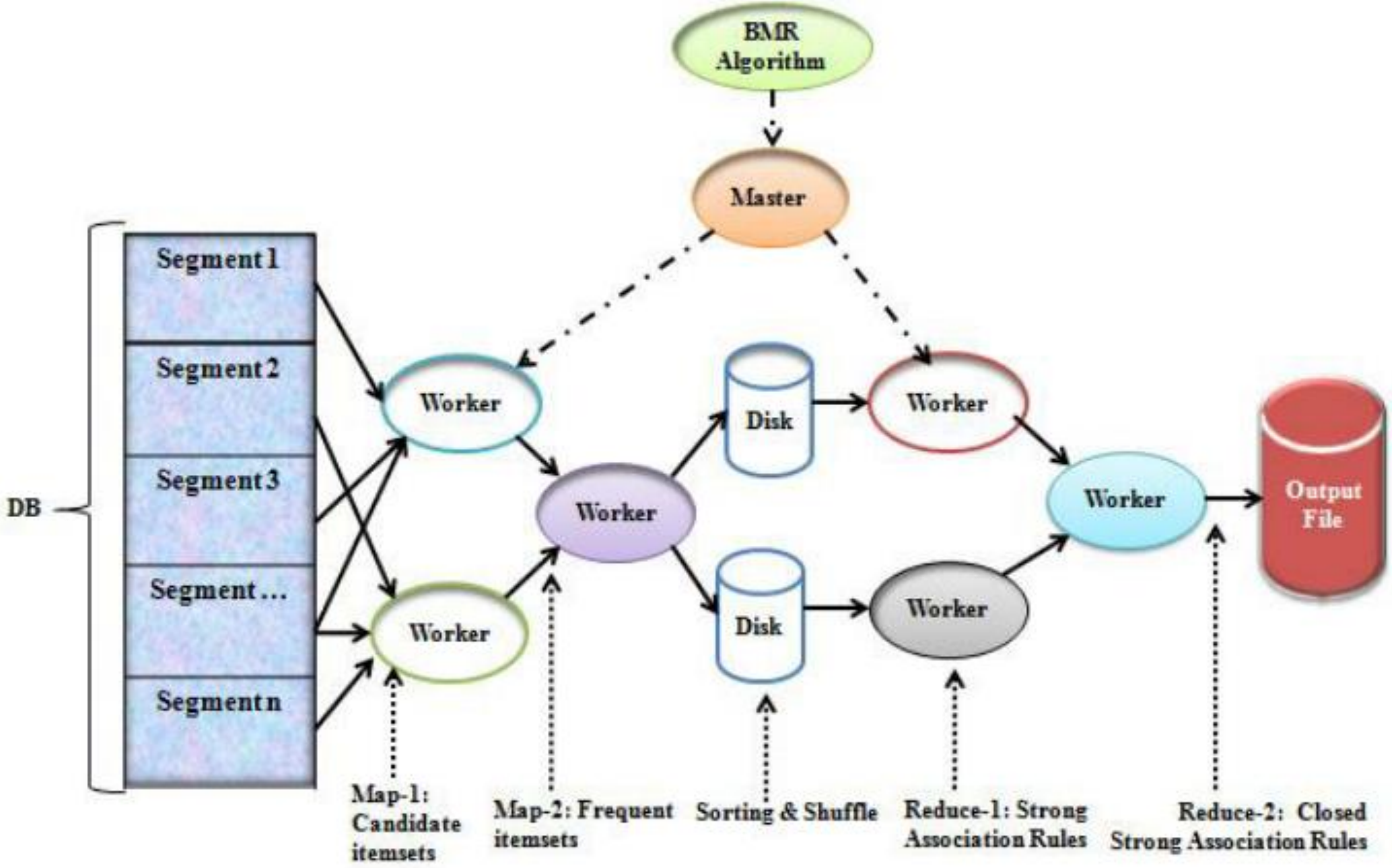
FREQUENT ITEMSETS IN $\leq$ 2 PASSES

- A-Priori, PCY, etc., take k passes to find frequent itemsets of size k

- **Is it possible to use fewer passes?**

- Use 2 or fewer passes for all sizes, **but may miss some frequent itemsets**
  - Random sampling
  - SON (Savasere, Omiecinski, and Navathe)
  - Toivonen

ARM-: MAPREDUCE VERSION



EXAMPLE 3:

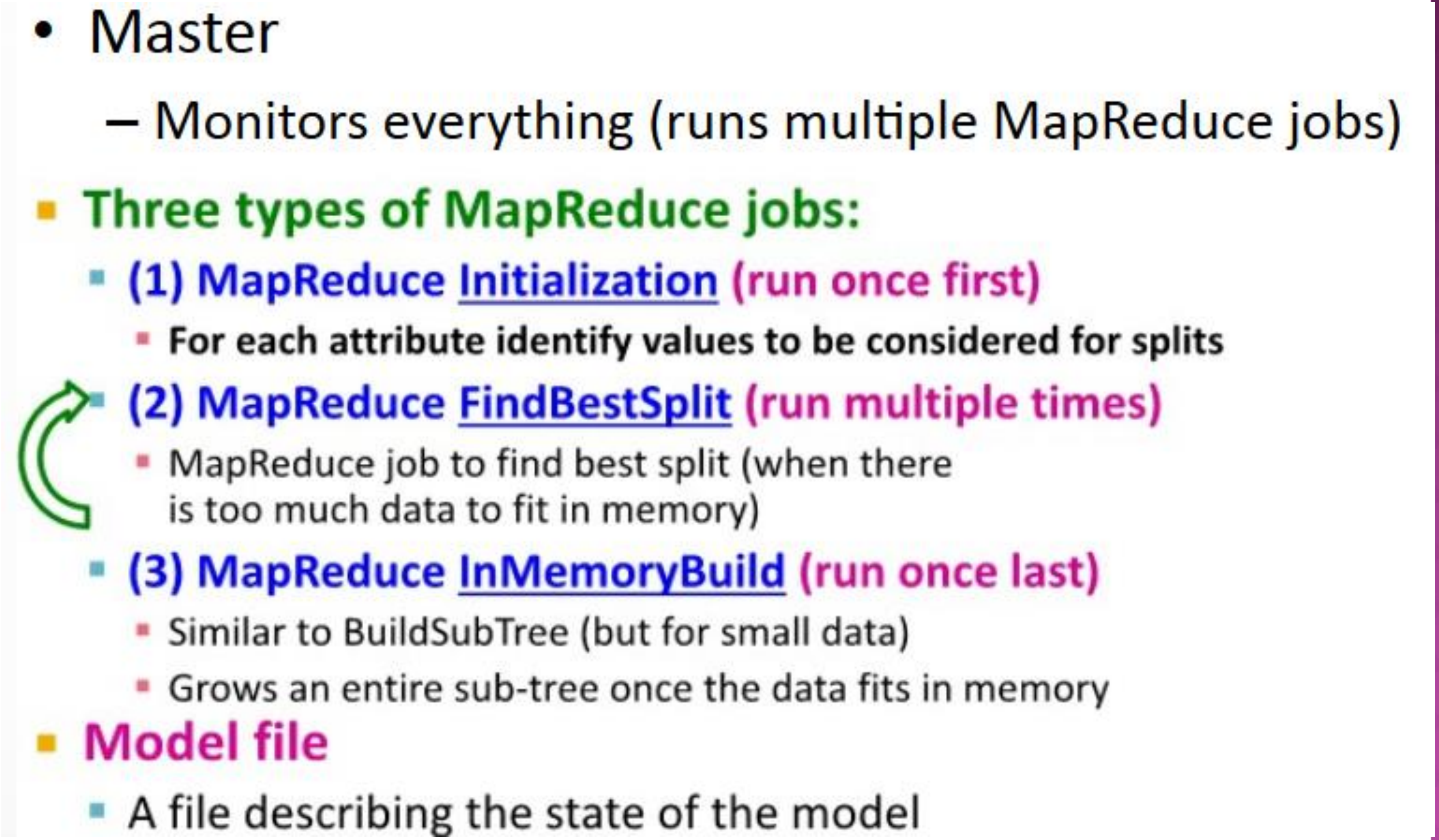
- Classification
- Linear Regression

A sequential method for estimating a regression model is to "build up" an array of product-moments by sequentially reading in segments of a file. Store the cumulative results for

$\sum X$, $\sum Y$, $\sum XY$, $\sum X^2$ and $\sum Y^2$.

$$\overset{\Lambda}{\beta}_1 = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - \left(\sum x\right)^2} \qquad \hat{\beta}_0 = \bar{Y} - \bar{X}\hat{\beta}_1.$$

DECISION TREE: MR VERSION



- Master
  - – Monitors everything (runs multiple MapReduce jobs)
- **Three types of MapReduce jobs:**
  - **(1) MapReduce Initialization (run once first)**
    - **For each attribute identify values to be considered for splits**
  - **(2) MapReduce FindBestSplit (run multiple times)**
    - MapReduce job to find best split (when there is too much data to fit in memory)
  - **(3) MapReduce InMemoryBuild (run once last)**
    - Similar to BuildSubTree (but for small data)
    - Grows an entire sub-tree once the data fits in memory
- **Model file**
  - A file describing the state of the model

RANDOM FOREST

- Use Random Forest, a supervised learning ensemble model
- The Random Forest algorithm randomly selects observations and features (several samples that can create trees in parallel in Map phase) to build several decision trees and then averages the results (In Reduce phase).
- Another difference is that „deep“ decision trees might suffer from overfitting. Random Forest prevents overfitting most of the time, by creating random subsets of the features and building smaller trees using these subsets.

THANK YOU

QUESTIONS?