DS501: Final Review!

Prof. Randy Paffenroth
rcpaffenroth@wpi.edu

Worcester Polytechnic Institute

Rules for final exam

The midterm exam and final exam will be in class, noncumulative, and open note, but **no collaboration will be allowed** and the exams be graded based upon demonstrated understanding of key concepts. For each exam, you are allowed to bring in up to **4 (four)** 8 ½ by 11 sheets of paper (either printed or handwritten) with whatever notes you want for the exam. You are also allowed a calculator that does not have a network connection (e.g., **no cell phones!**).

WPI

Note!

- You are responsible for all material covered in the class!
  - All slides.
  - All discussions.
  - All ideas from case studies.
- Of course, and stated in the syllabus, the final exam will focus on the second half of the class.
- These slides a subset of what you are responsible for (but perhaps an important subset).
  - You are definitely responsible for all of the lecture slides since the midterm.
- Keep an eye out for:

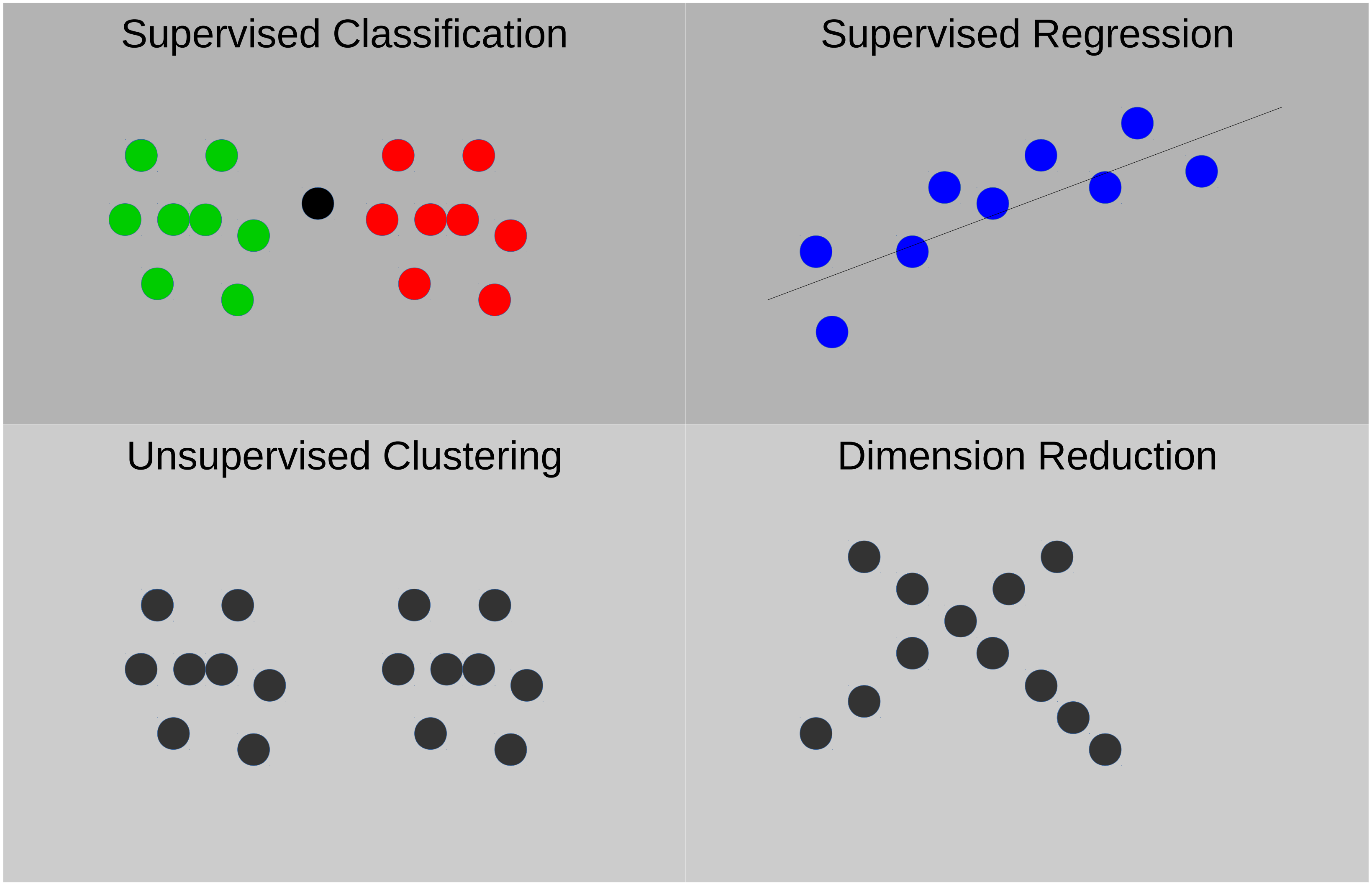
Important

WPI

Class 6
Machine Learning

We have four days we will cover four topics.

Important



What is PCA?

- Principle Component Analysis
  - Commonly used tool for visualization and data pre-processing.

What is Linear Support Vector Machine (SVM)?

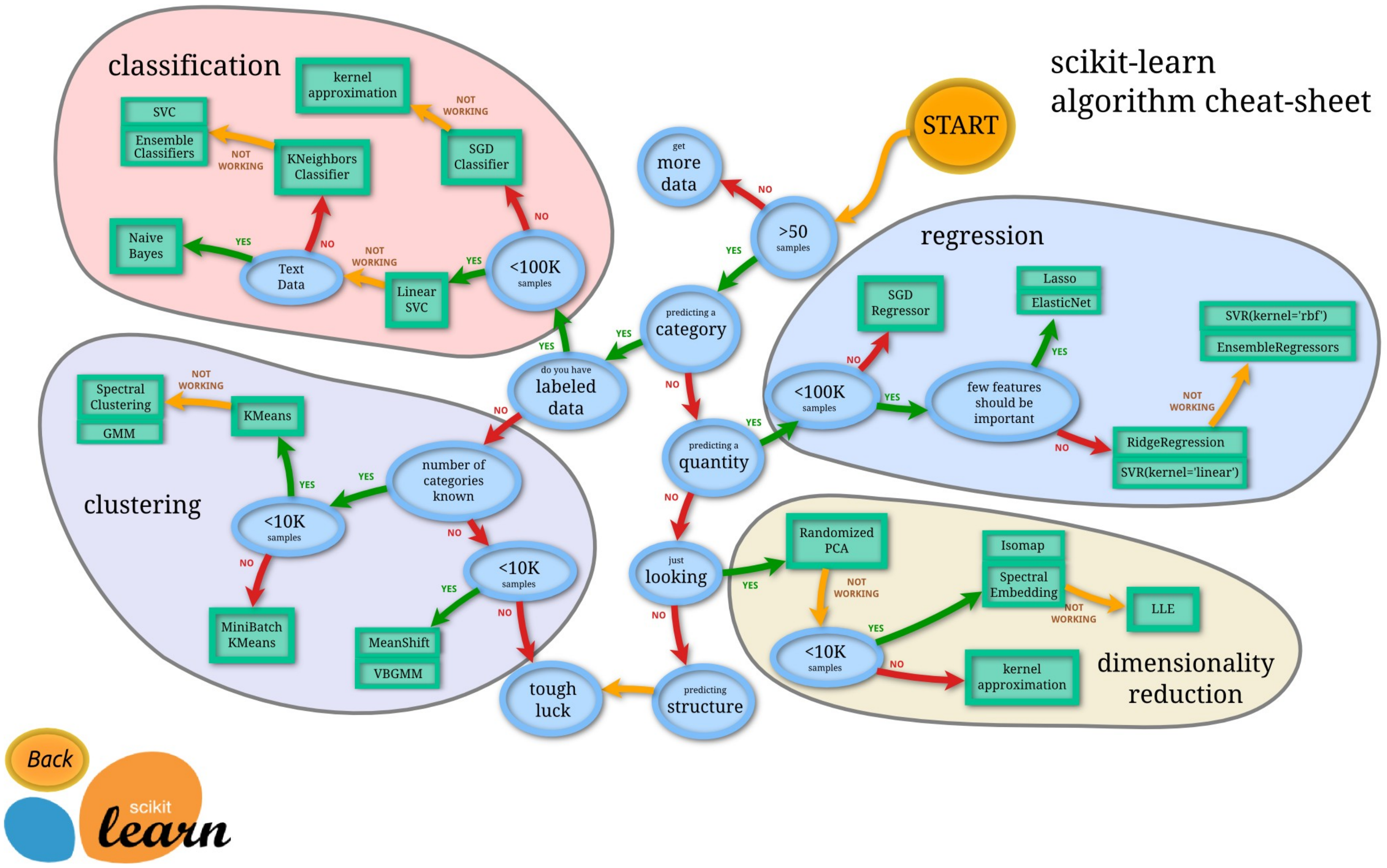
- Maximum margin classifier
  - Computes a linear “decision boundary” that splits the data into two regions.
  - Allows one to predict a classification of a point based upon which side of the decision boundary it lay on.

What is K-NN?

- K-nearest neighbors
- Another common classification algorithm
  - Perhaps the most common

Class 8
Machine Learning

scikit-learn



http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

WPI

Linear Regression flow chart

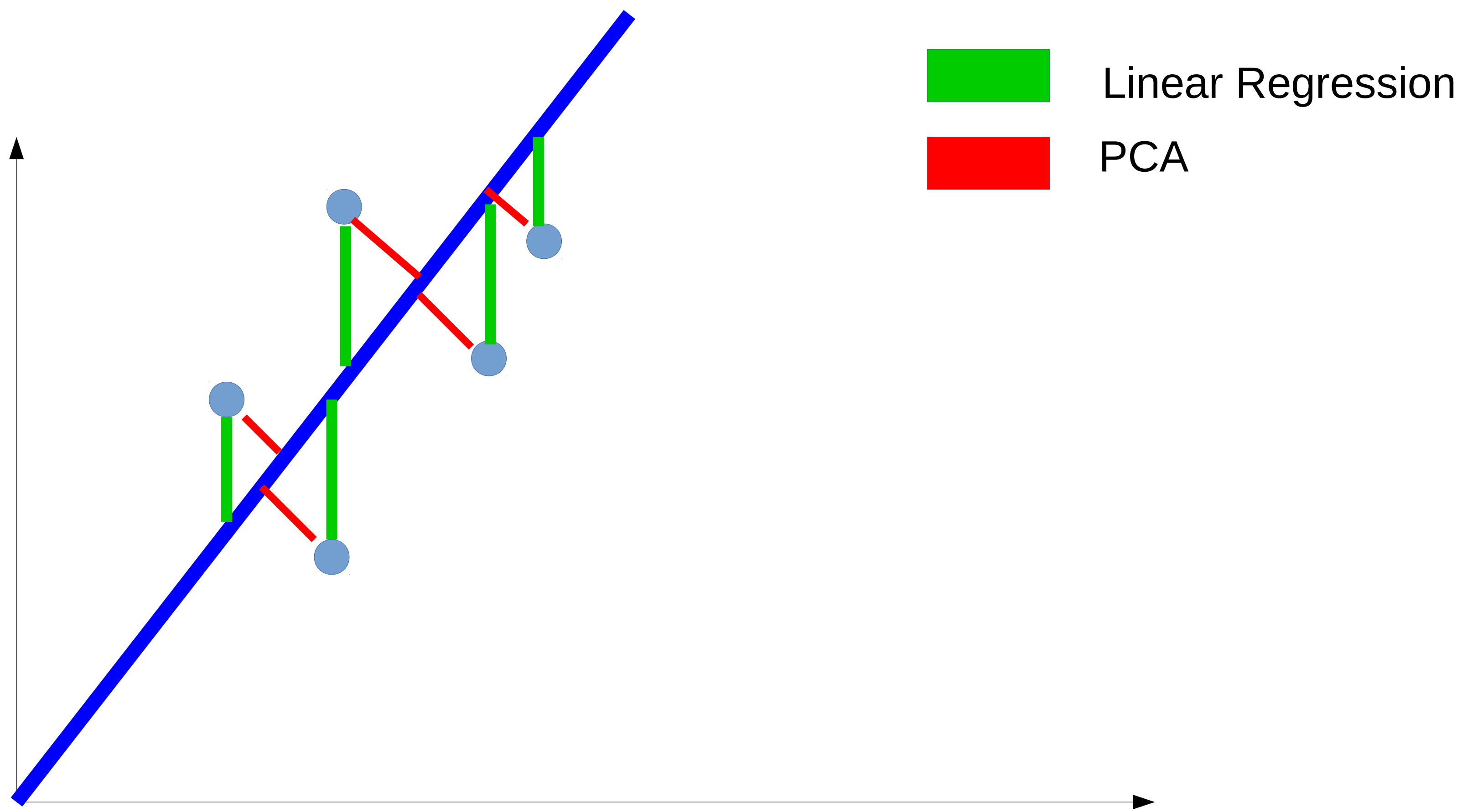
Fit → $RSS = (y_1 - \hat{y}_1)^2 + \cdots + (y_p - \hat{y}_p)^2$ → Compute minimum → $\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T y$

Predict → $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$ → Compute minimum

Error → $RSE = \sqrt{\dfrac{RSS}{n-(p+1)}}$ → Plug in data (testing and *training*) to see how we did.

Relationship between Linear Regression and PCA...

Important



Multiple-linear regression

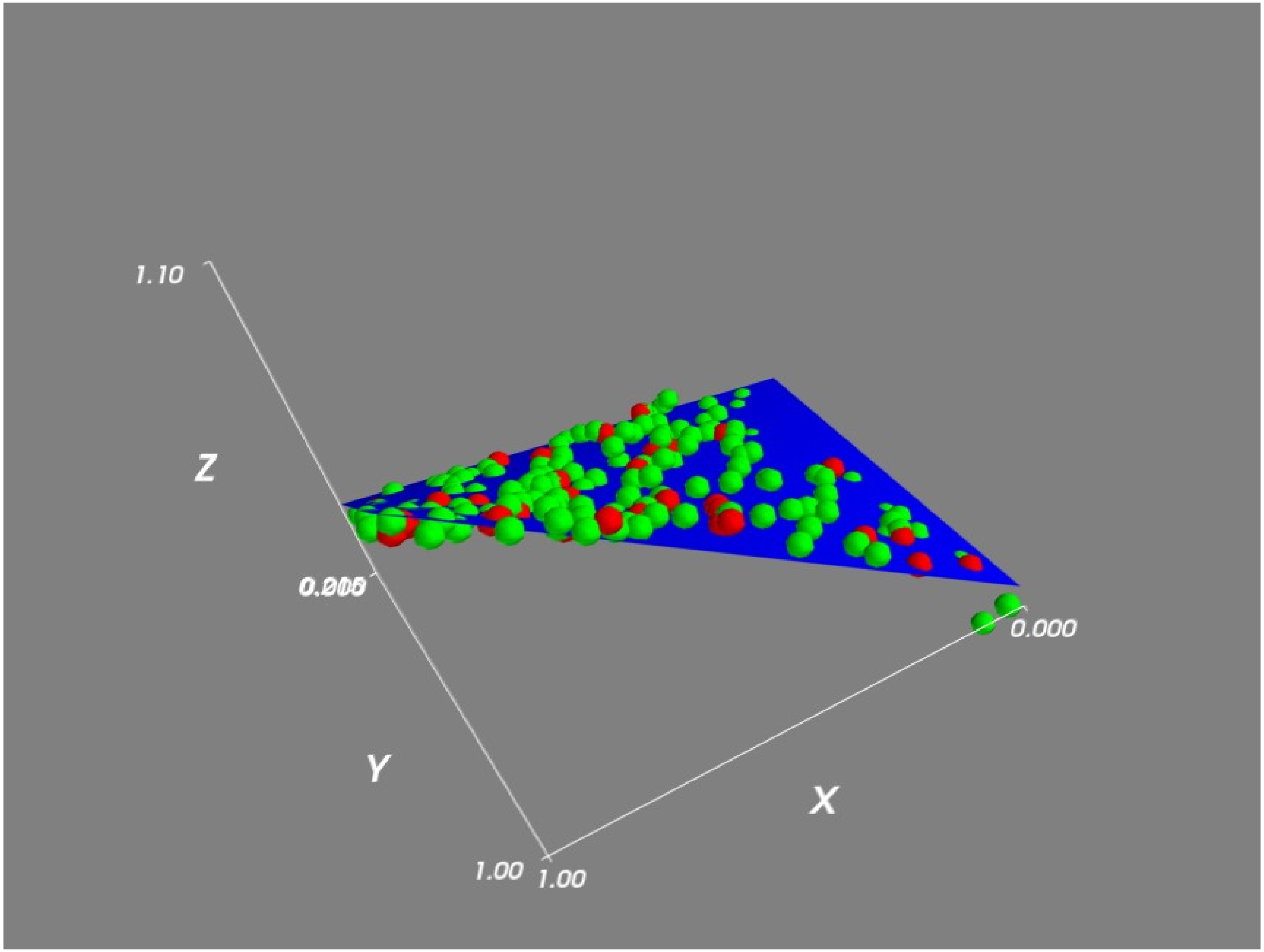
$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$$

$$y_i = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_1 x_{k,i} \quad \hat{y}_0 = \beta_0 + \beta_1 x_{1,0} + \cdots + \beta_1 x_{k,0}$$

“Non-linear” regression

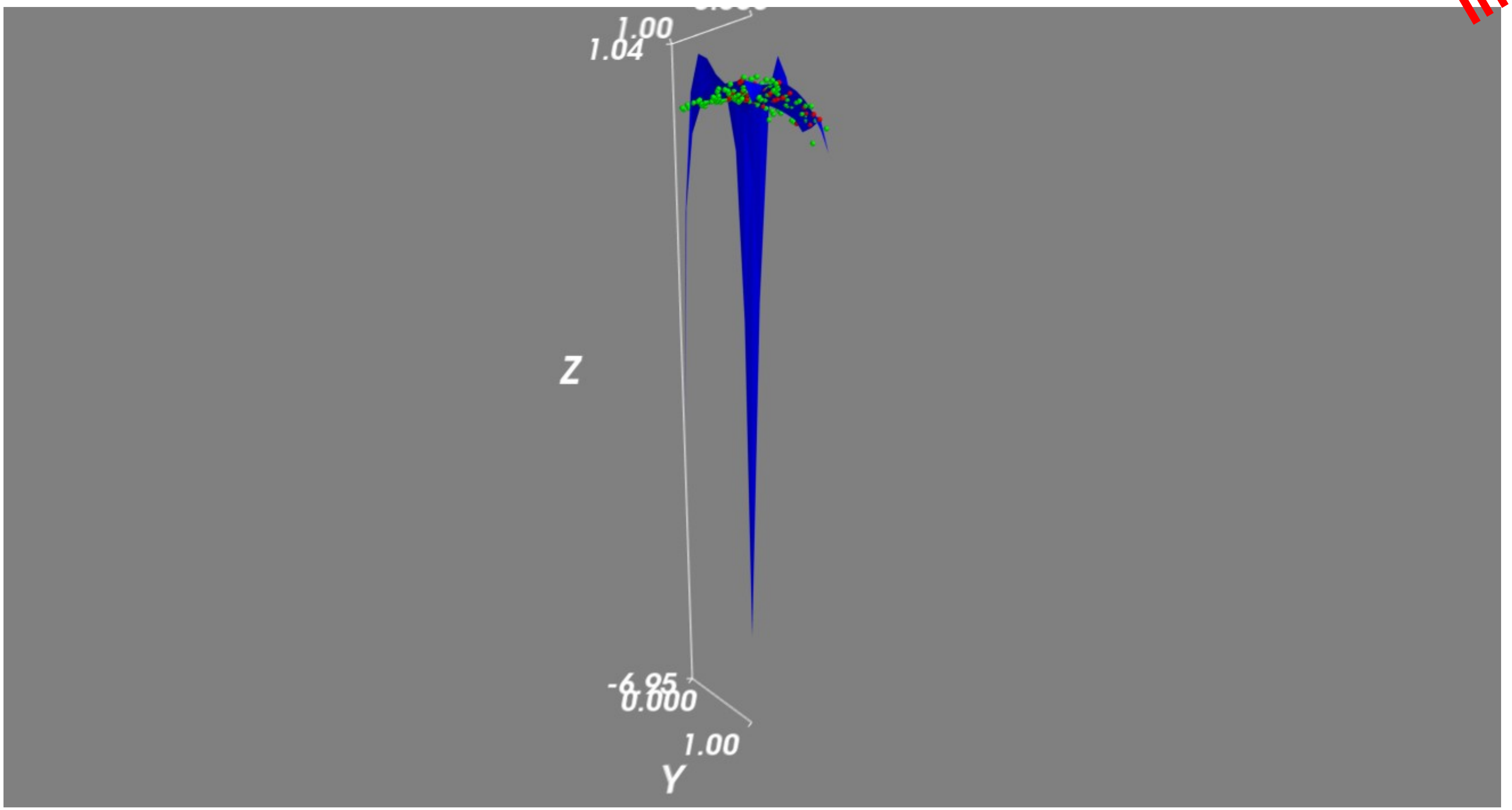
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$$

See it in Python



Overfitting!!!!

Important
Important
Important



Cross Validation

Important

- Validation set
- K-fold
- Leave-one-out cross validation (LOOCV)

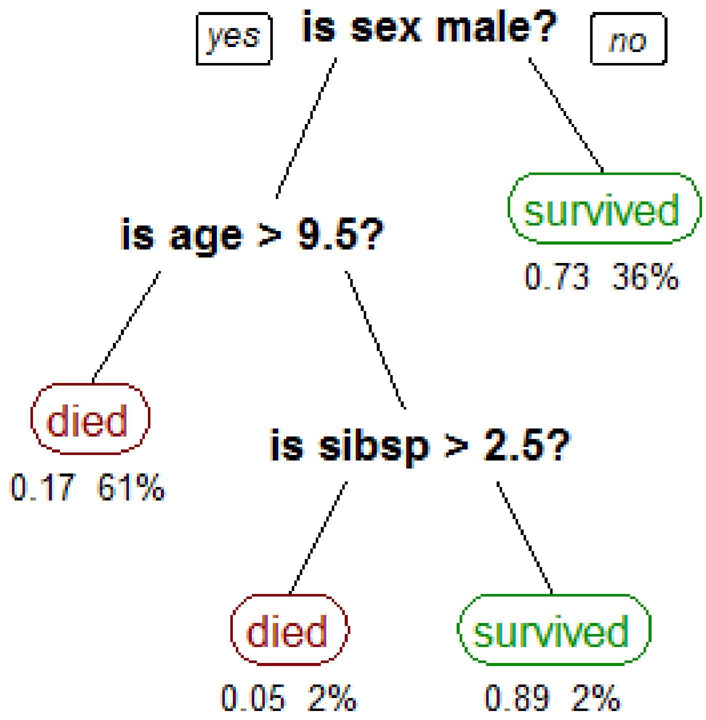
WPI

Decision tree

- Quite commonly used in data mining.
- Each node in the tree splits the data into (classically) two groups.
  - To make things easy (and fast) you classically perform each split on a single variable.
- Each leaf node then represents a value (or perhaps range of values) for the response based upon the input variables.

Titanic data

Probability of survival
Percentage in that leaf



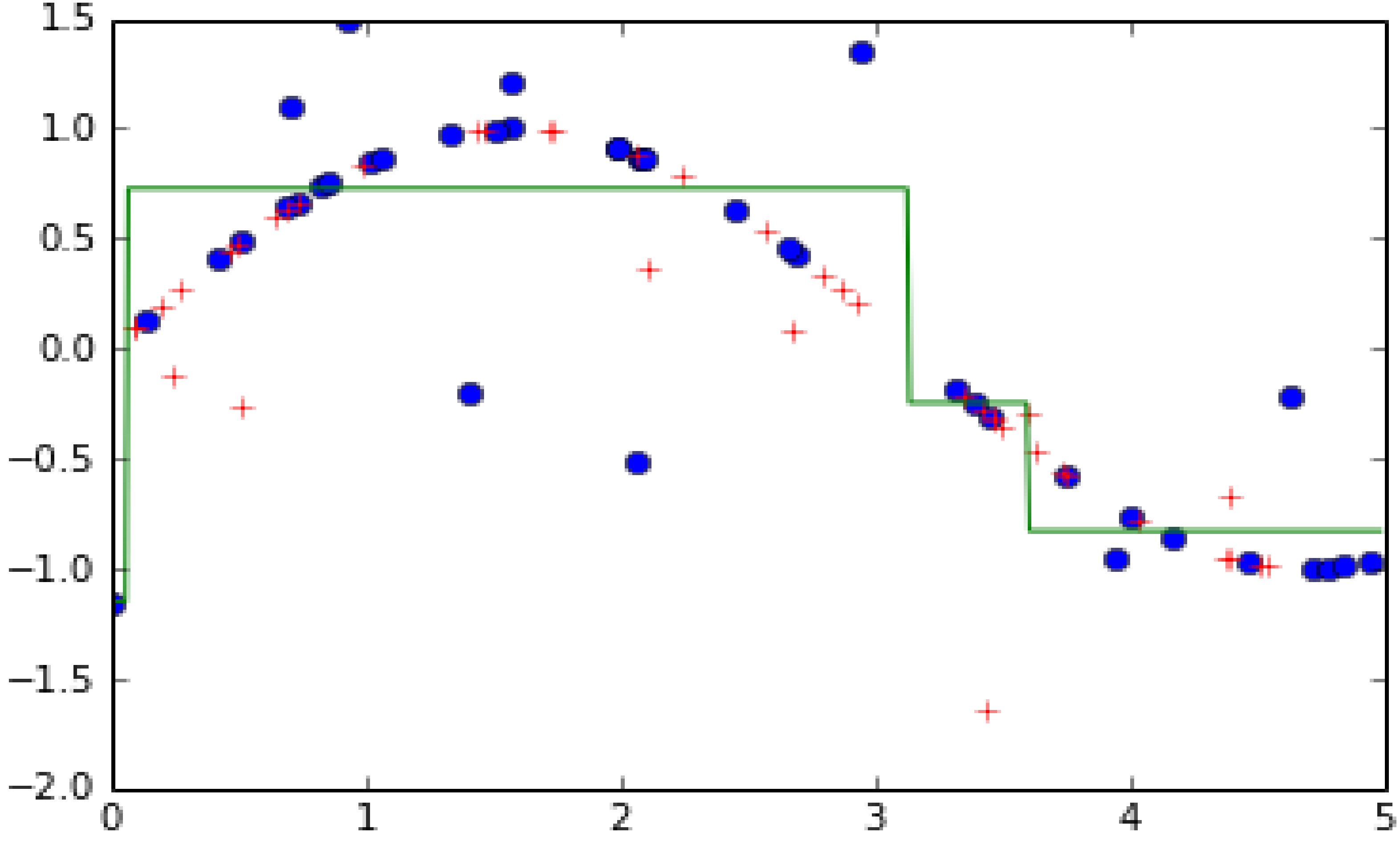
Decision tree: Making the splits

$$\hat{p}_{mk} = Pr(Y = k | X \text{ is in region } k)$$

$$E = \max_k \hat{p}_{mk}$$

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

See it in Python



WPI

Ensemble Learning: Random Forest

- As you can tell by the name, this idea revolves around having many trees.

WPI

Tree bagging: Bootstrap aggregation

Important

- Bootstrapping is one of my favorite algorithms in statistical learning.
- An extremely powerful idea for doing statistical learning with limited data.
- Generate many random samples of your data, with replacement, and train a tree on each...

Random forests

Important

- Add even more randomness by randomly selecting for each tree a subset of the features it is allowed to split on.
  - Reduces correlation between the trees!
  - Not all trees can pick the “obvious” best predictor to split on first.

K-Means

- Perhaps the single most used unsupervised classification algorithm.
- Given a number of classes k, divide the data into groups so that the distance within a group is “small” compared to the distances between groups.

K-Means

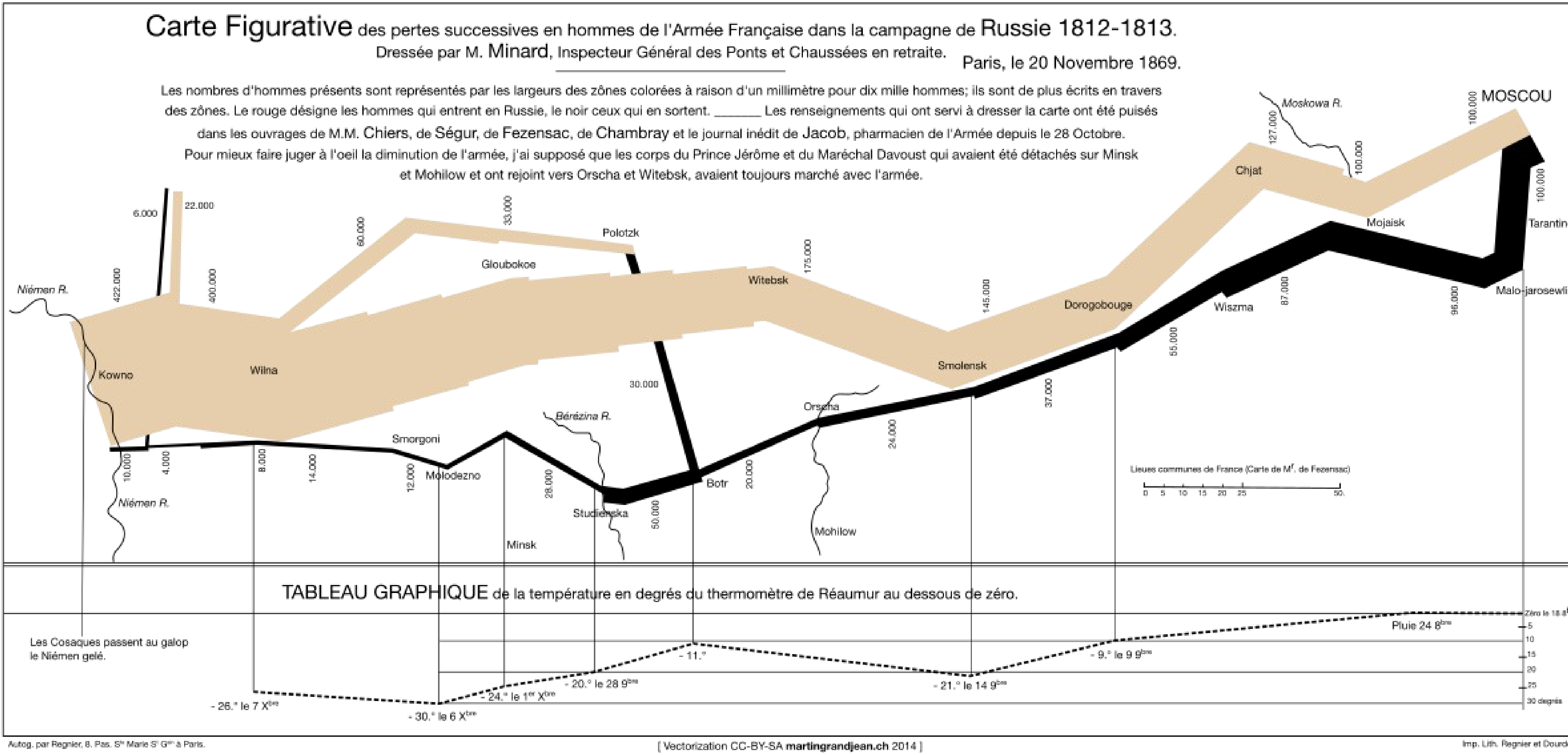
$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2$$

Class 9
Visualization

Visualization

Important

- Visualization: The activity of forming a mental model of something
  - R. Spence, Information Visualization, DOI 10.1007/978-3-319-07341-5_1



The numbers of men present are represented by the widths of the colored zones at a rate of one millimeter for every ten-thousand men; they are further written across the zones. The red [now brown] designates the men who enter into Russia, the black those who leave it.

WPI

7 Tasks

Important

- Overview: Gain an overview of the entire collection.
- Zoom : Zoom in on items of interest.
- Filter: filter out uninteresting items.
- Details-on-demand: Select an item or group and get details when needed.
- Relate: View relations hips among items.
- History: Keep a history of actions to support undo, replay, and progressive refinement.
- Extract: Allow extraction of sub-collections and of the query parameters.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. Proceedings of IEEE Symposium on Visual Languages - Boulder, CO (pp. 336-343).

Visualization Taxonomy

Important

- 1D/Linear
- 2D/Planar (incl. Geospatial)
- 3D/Volumetric
- Temporal
- nD/Multidimensional
- Tree/Hierarchical
- Network

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. Proceedings of IEEE Symposium on Visual Languages - Boulder, CO (pp. 336-343) and http://guides.library.duke.edu/vis_types.

WPI

Important

And then we had many examples that we analyzed...

WPI

1-dimensional

- Purely 1-dimensional data
  - Text documents
  - Program source code
  - Etc.
- Is there anything interesting here?

2-dimensional

- Maps
- Floorplans
- Etc.

Choropleth: Shaded or patterned map

http://bl.ocks.org/mbostock/4060606

WebGL: World population

- http://data-arts.appspot.com/globe/

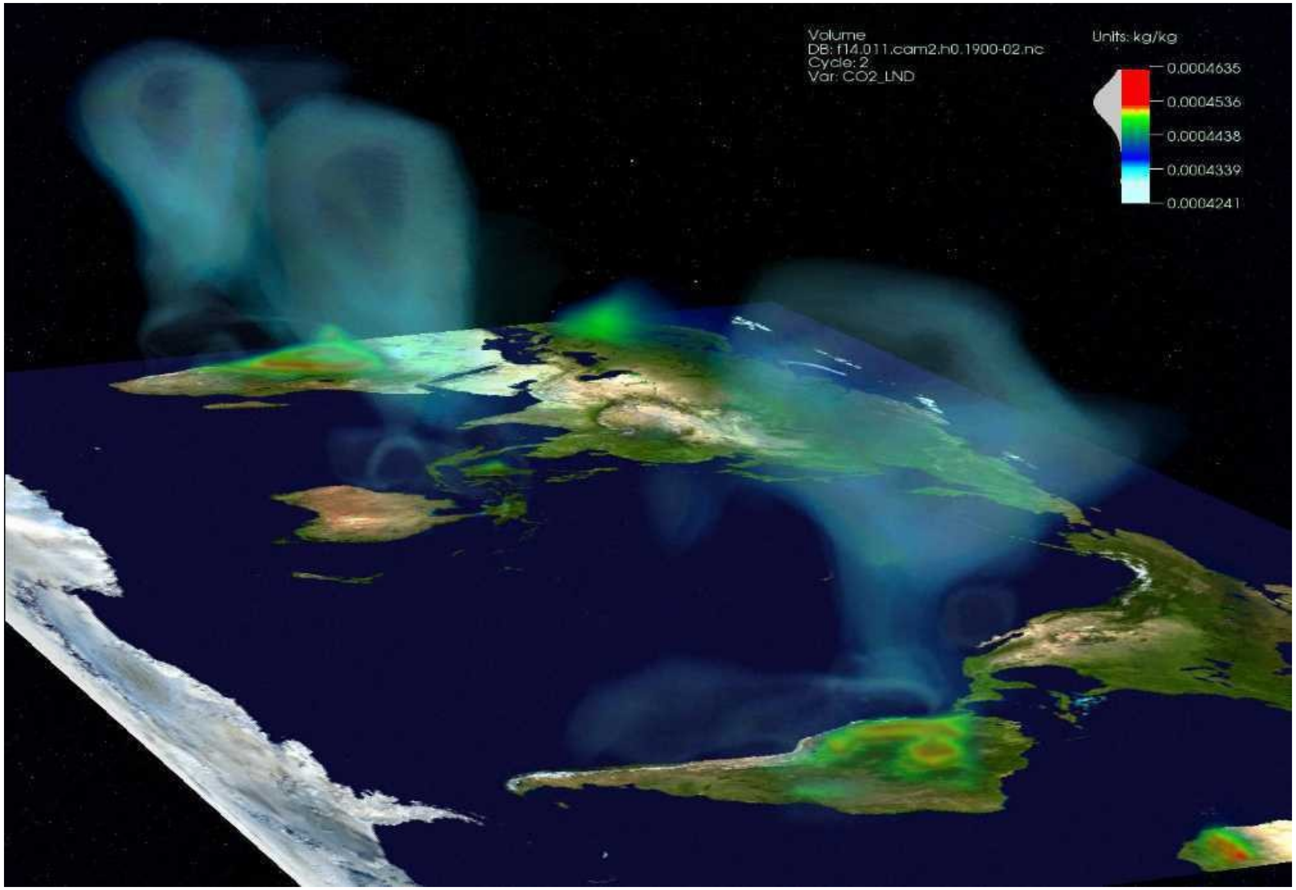
World Bank

http://d3.artzub.com/wbca/

3-dimensional

- Real world objects
  - Buildings
  - Molecules
  - Vehicles
  - Etc.

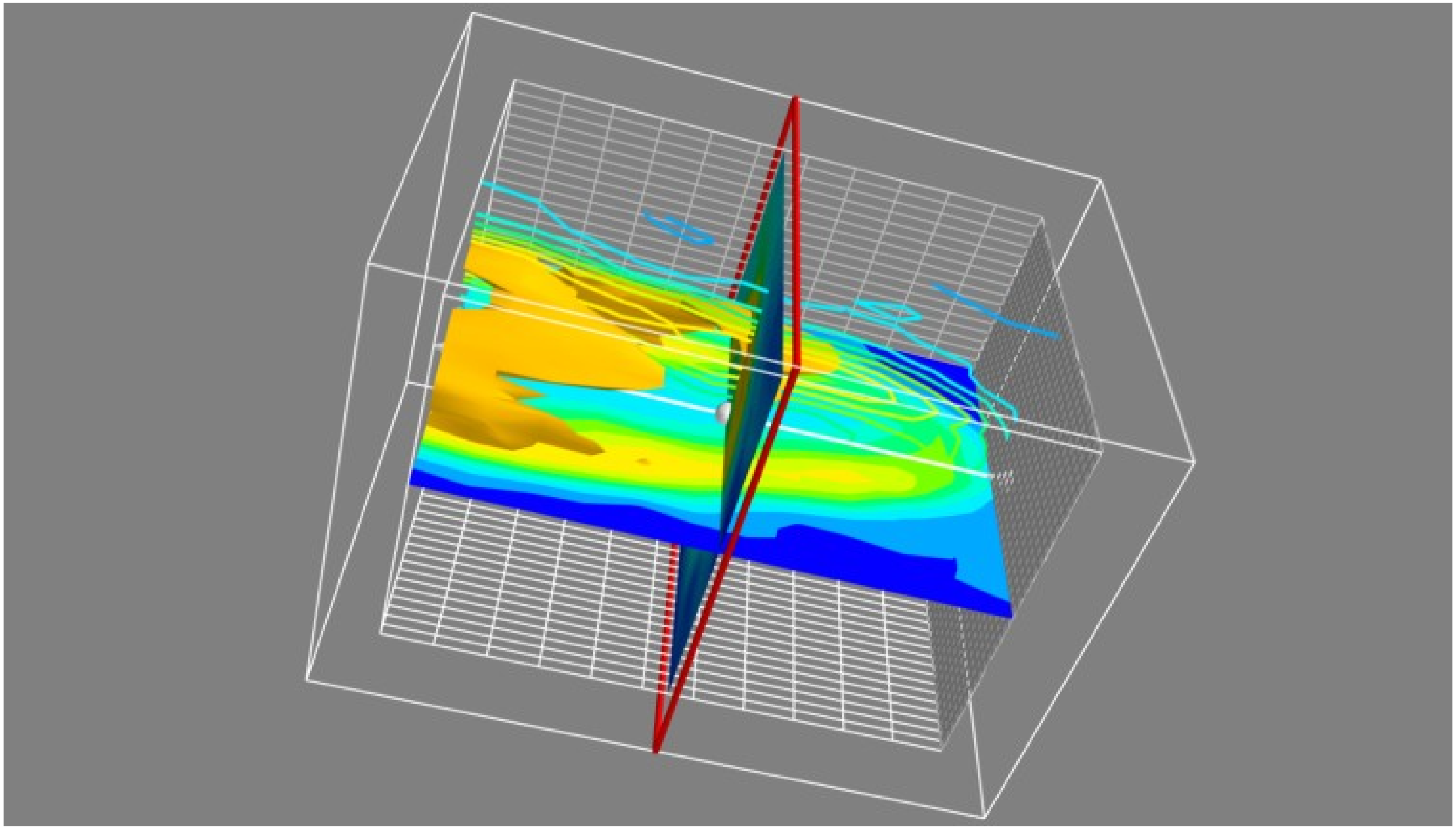
Climate



CO2 from ocean plumes.

WPI

Contours

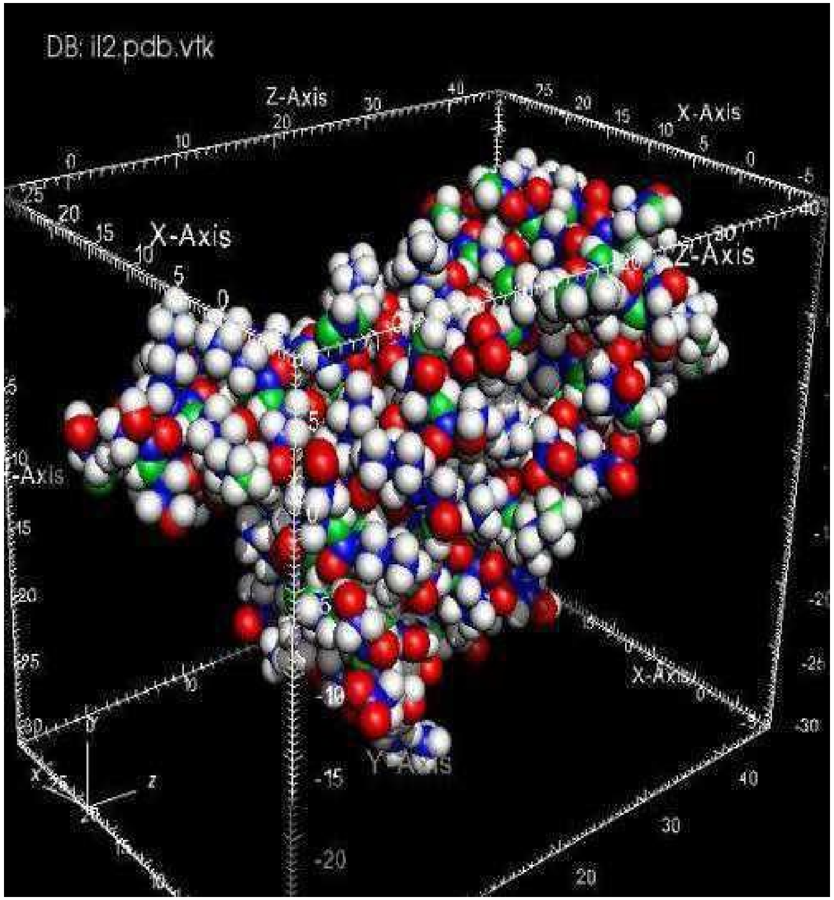


Python 2

Python 3

WPI

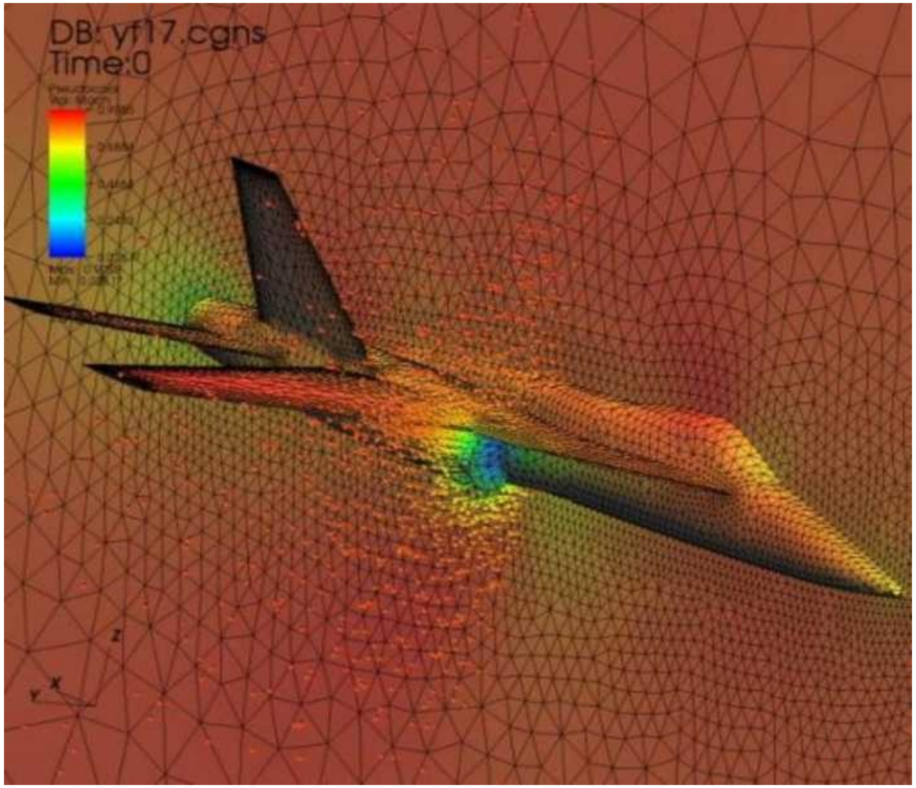
Molecules



Protein

"Molecular rendering" by UCRL-WEB - visualizations that have been created with VisIt. at wci.llnl.gov. Licensed under Public domain via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Molecular_rendering.jpg#mediaviewer/File:Molecular_rendering.jpg

Finite Element Simulation



Mach number

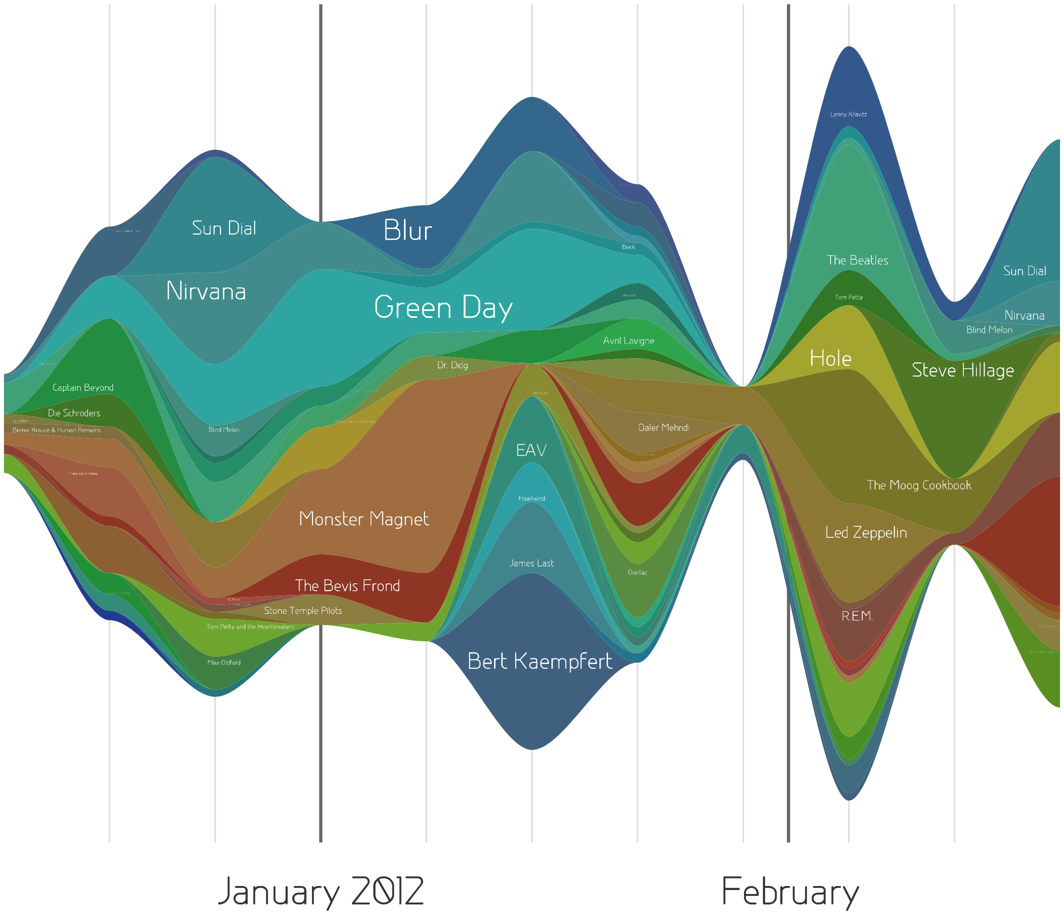
"YF-17 aircraft Plot" by The dataset was provided by the CGNS user community's - Visualizations that have been created with VisIt. at wci.llnl.gov. Licensed under Public domain via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:YF-17_aircraft_Plot.jpg#mediaviewer/File:YF-17_aircraft_Plot.jpg

WPI

Temporal

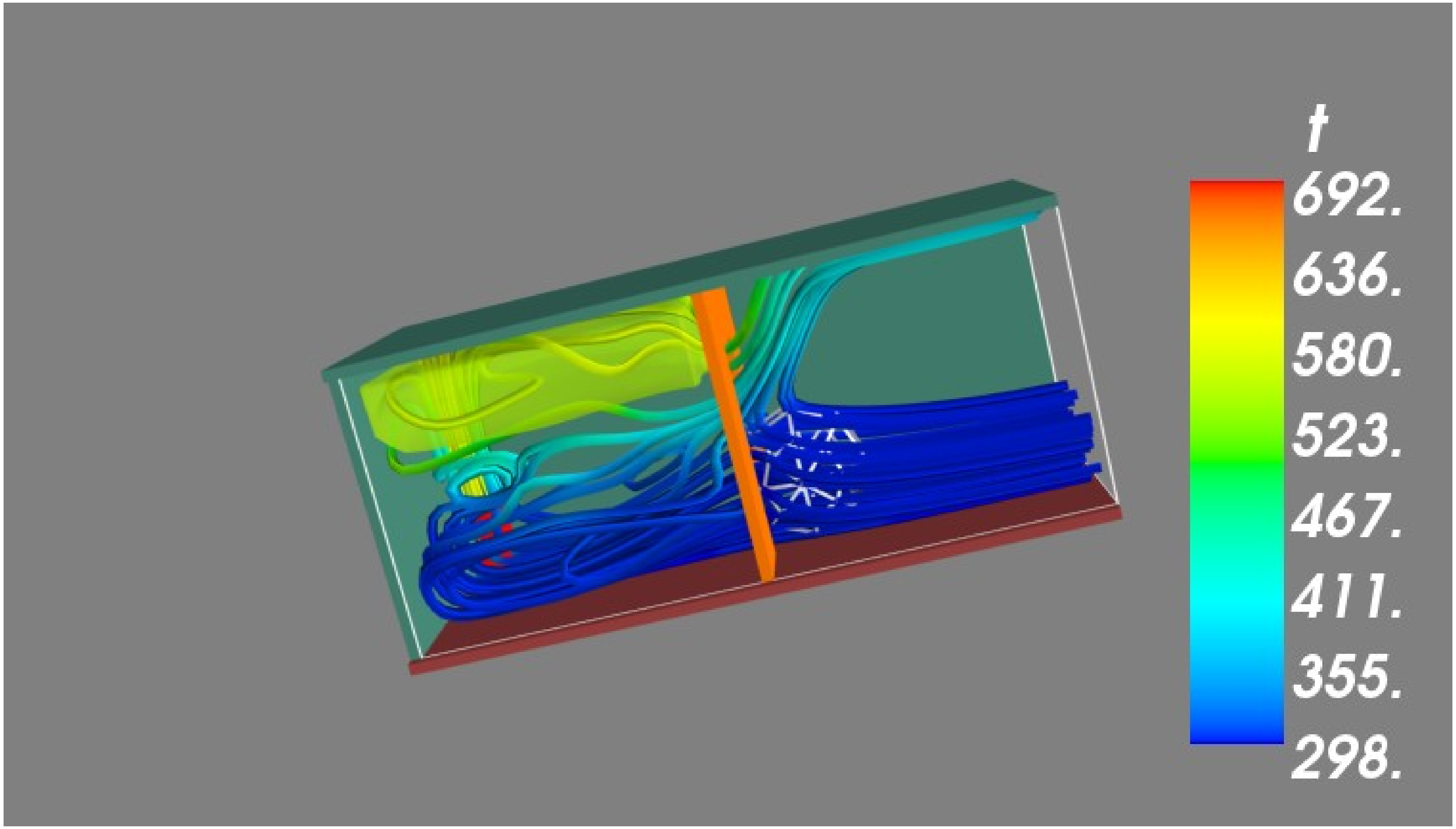
- Data with start and finish times
  - Historical events
  - Medical records
  - Marketing
  - Etc.

Streamgraph



Last.fm person's listening habits.

Streams



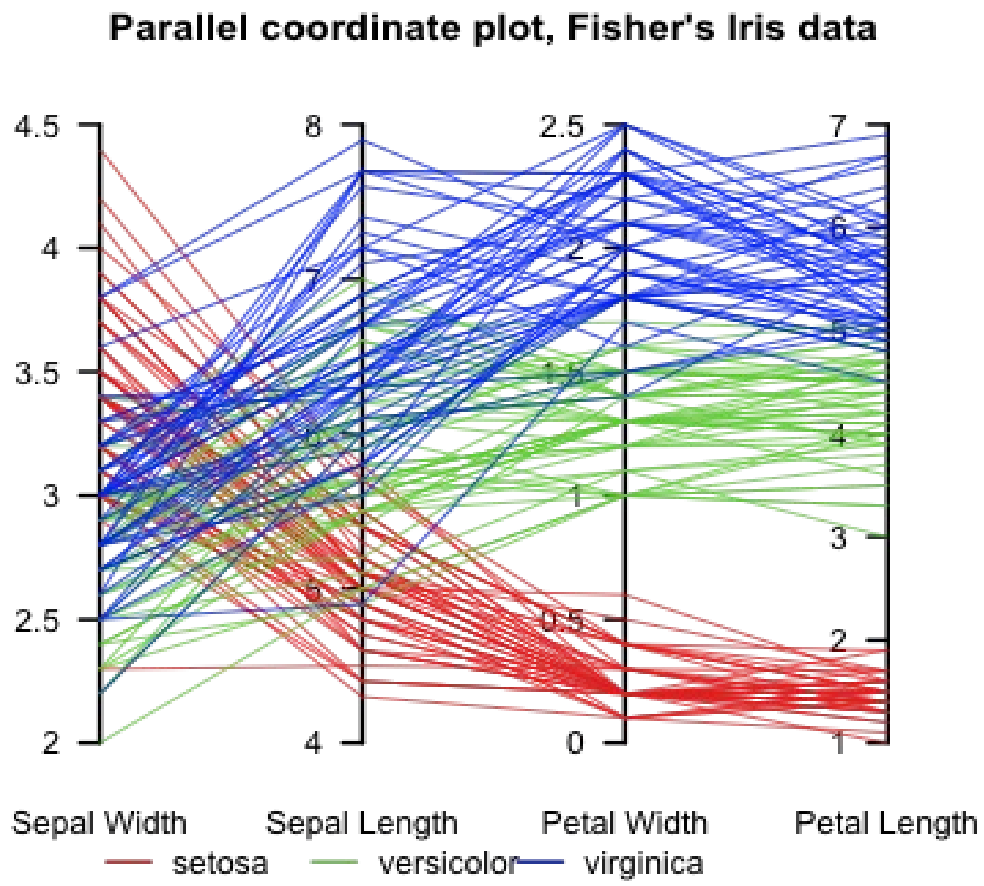
WebGL: Temperature anomalies

- http://www.chromeexperiments.com/detail/temperature-anomalies-200-years/?f=webgl

Multi-dimensional

- As far as Data Science is concerned, this is really where the things get very interesting!
  - Can you give me some examples?

Parallel coordinates

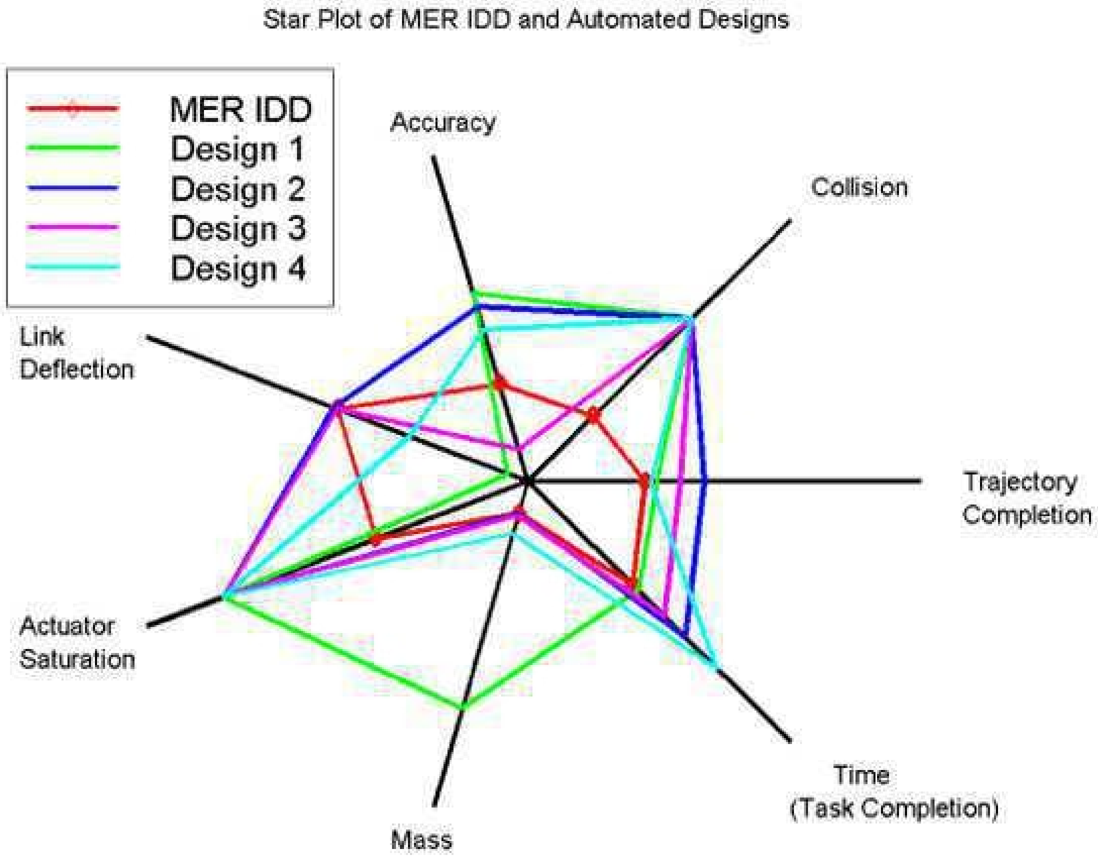


Iris data set!

Parallel coordinates

- http://exposedata.com/parallel/

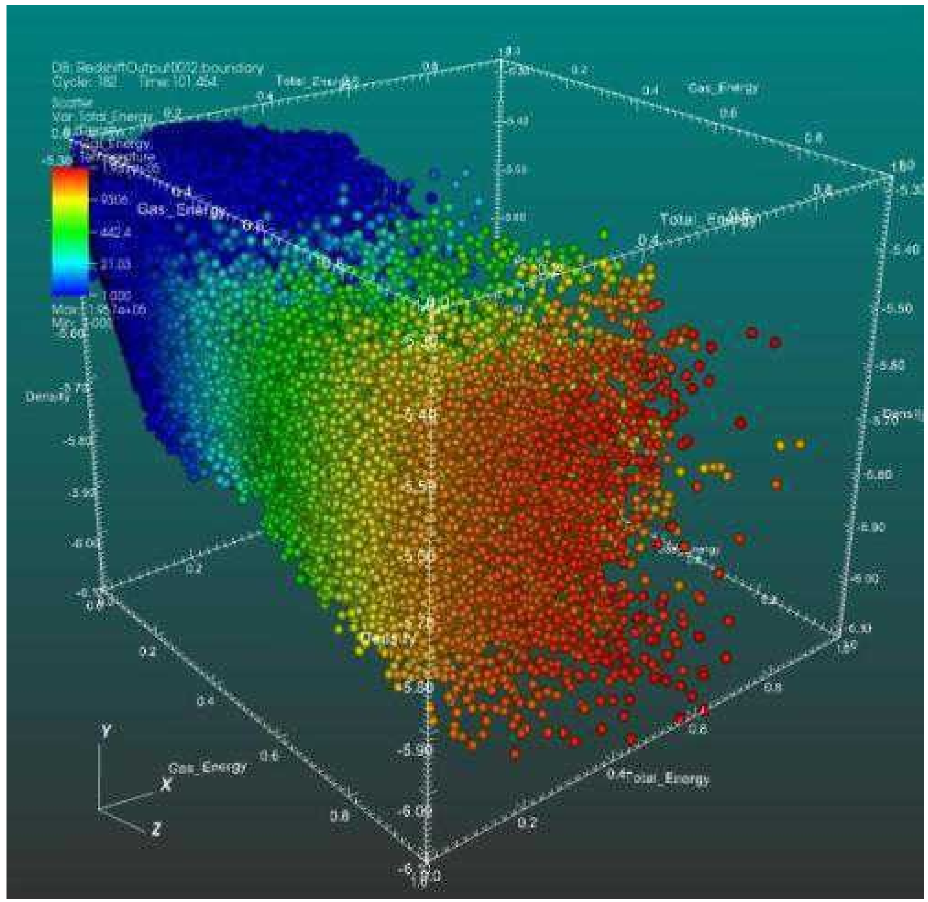
Radar Chart



NASA Space system design: Desirable is in the middle.

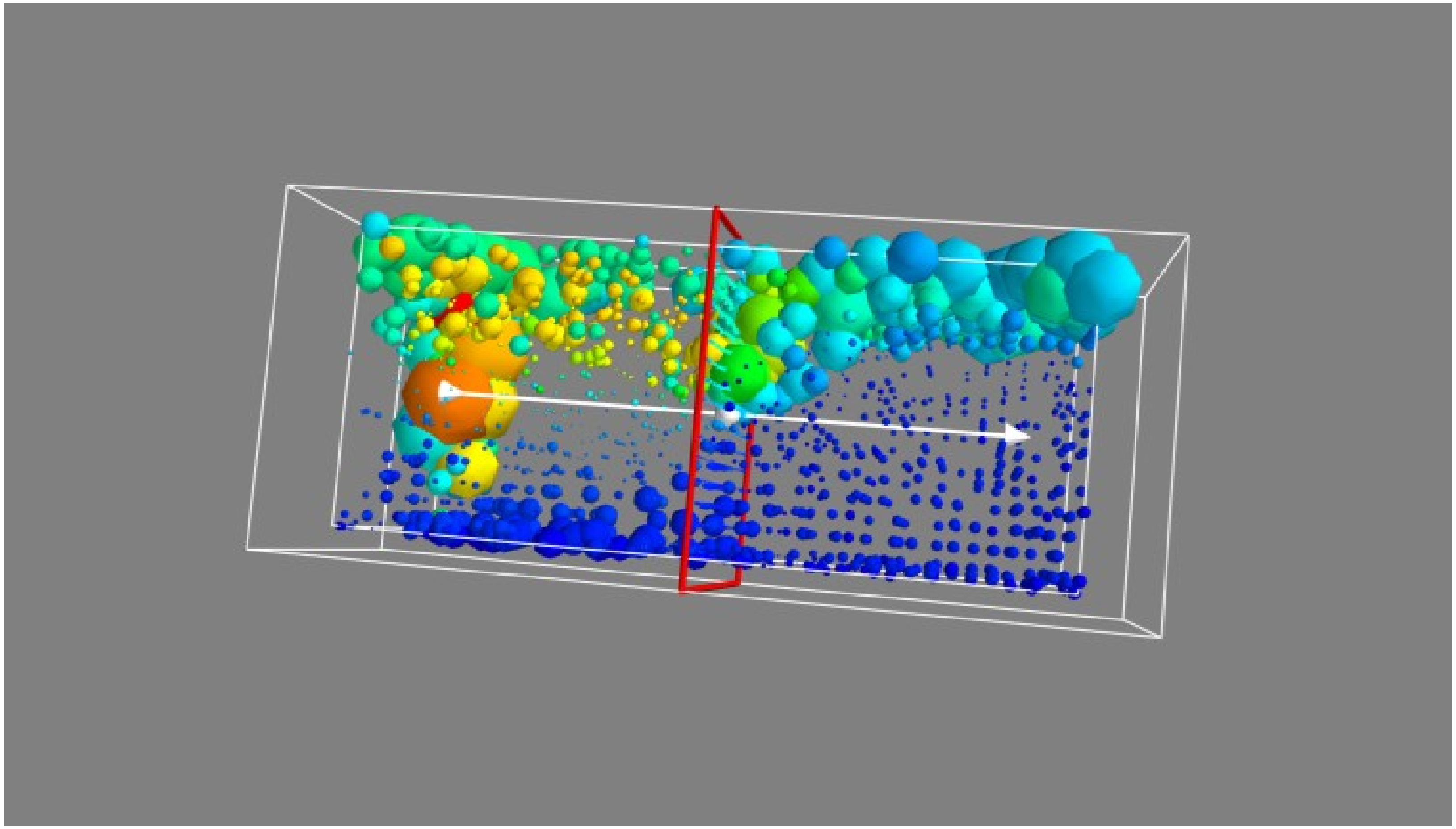
WPI

Scatter plot

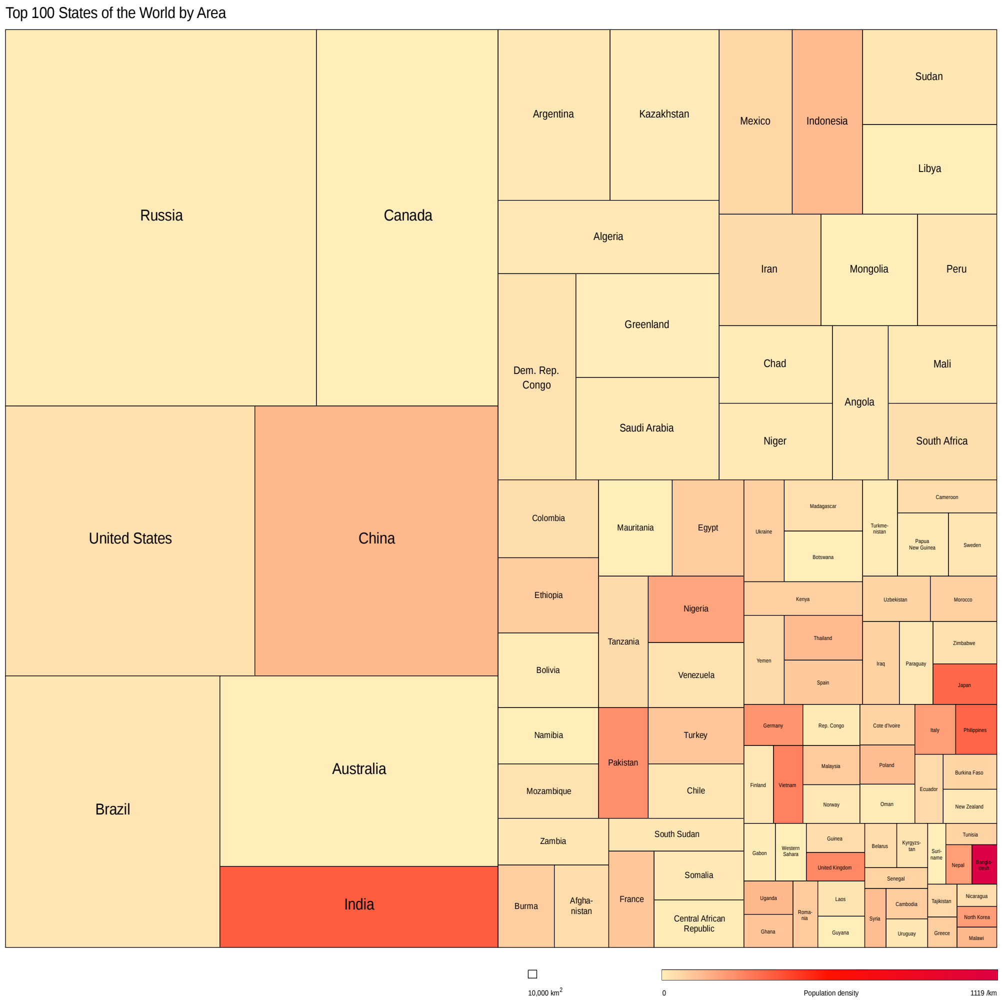


4 dimensions per point

Glyphs



Tree chart



Size is area, color is density.

"Top100 states area treemap pop-density" by Own work - Data: CIA World Factbook (visited 2012-09-12);Treemapping: Google Chart Tools visualization API. Licensed under Public domain via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Top100_states_area_treemap_pop-density.svg#mediaviewer/File:Top100_states_area_treemap_pop-density.svg

Tree or hierarchical

- Items that fall into groups that can be further sub-divided.

Federal budget

http://www.brightpointinc.com/interactive/budget/index.html?source=d3js

Network

- Graphs
- Social networks
- Computer networks

Social Network

League of nations personel

"Social Network Analysis Visualization" by Calvinius - Own work : http://www.martingrandjean.ch/wp-content/uploads/2013/10/Graphe3.png. Licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Social_Network_Analysis_Visualization.png#mediaviewer/File:Social_Network_Analysis_Visualization.png

Matrix representation

http://bokeh.pydata.org/docs/gallery/les_mis.html

Network

- http://fatiherikli.github.io/programming-language-network/

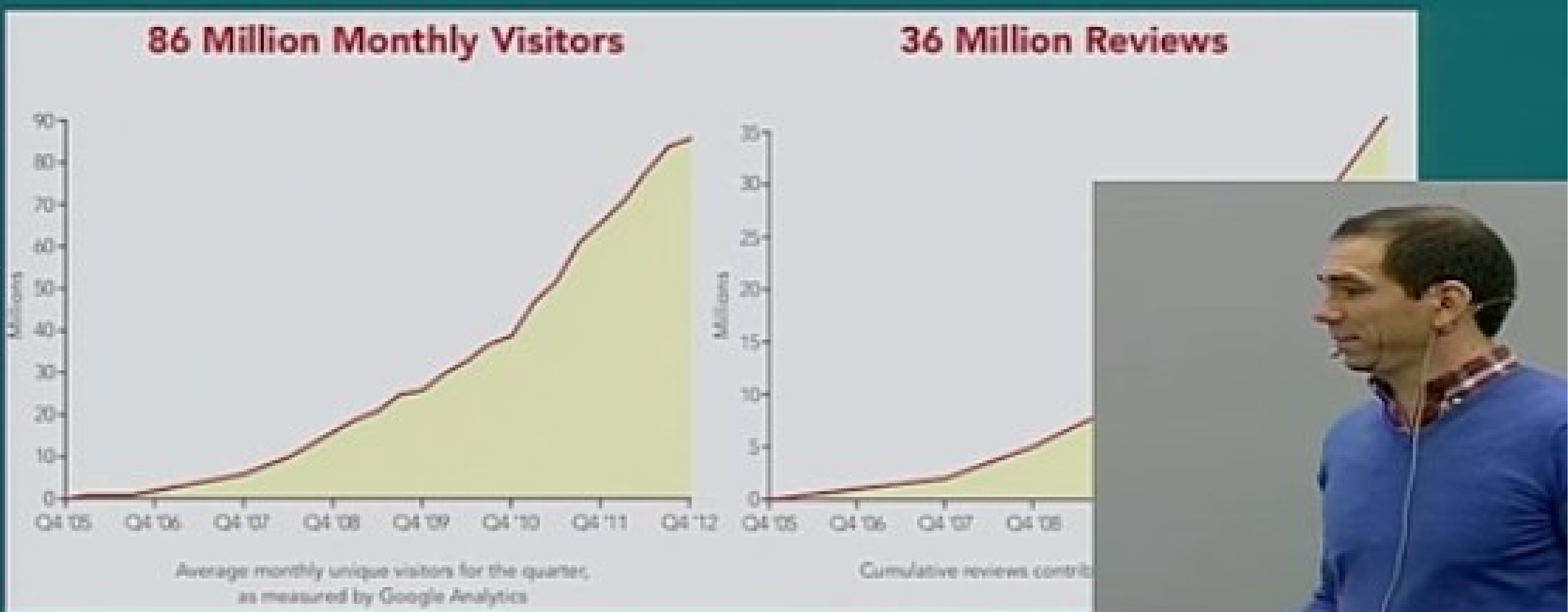
Chord Diagram

http://bost.ocks.org/mike/uberdata/

Class 10
Map-reduce and large scale data

Yelp has a problem

- 250+ GB of logs per day
- Each GB takes 10 minutes to process
- How long to handle a day's logs?



Oops...

- 250/(6*24) = 1.73 days of work (per day!)
  - You never catch up!
  - What do you do?

What is the answer?

More specifically...

Important

- What is the core issue:
  - CPU?
  - Memory?
  - Disk space?
  - Network Access?

Modern trend: GPUs aren't just for gaming....

https://www.flickr.com/photos/juanpol/8232592/in/photostream/

They are for linear algebra!

Nvidia Tesla K40

"NvidiaTesla" by Mahogny - CameraTransferred from en.wikipedia. Licensed under Public domain via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:NvidiaTesla.jpg#mediaviewer/File:NvidiaTesla.jpg

WPI

Distributed computing is hard...

Important

- I mean really hard.
  - Parallelization
  - Synchronization
  - Resource contention
    - Deadlock
    - Dining Philosophers...
  - Fault Tolerance
  - Distributed I/O
  - Etc.

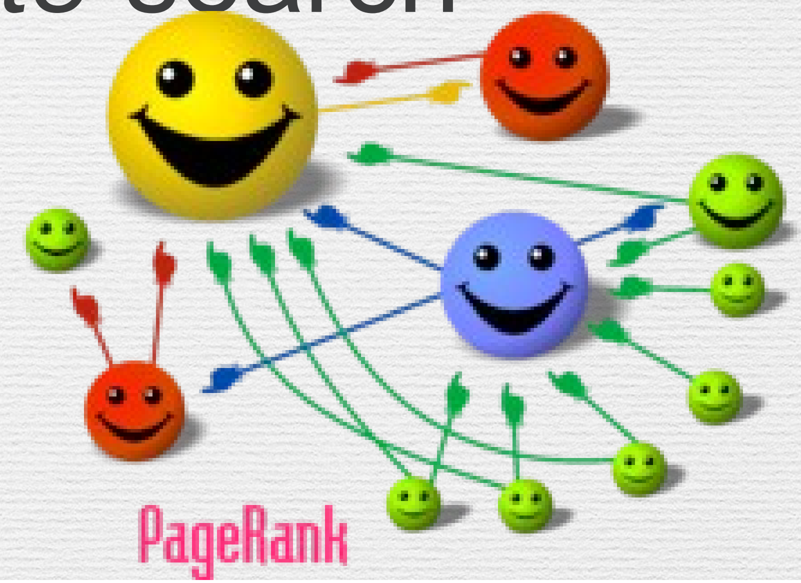
WPI

But can’t you automate it?

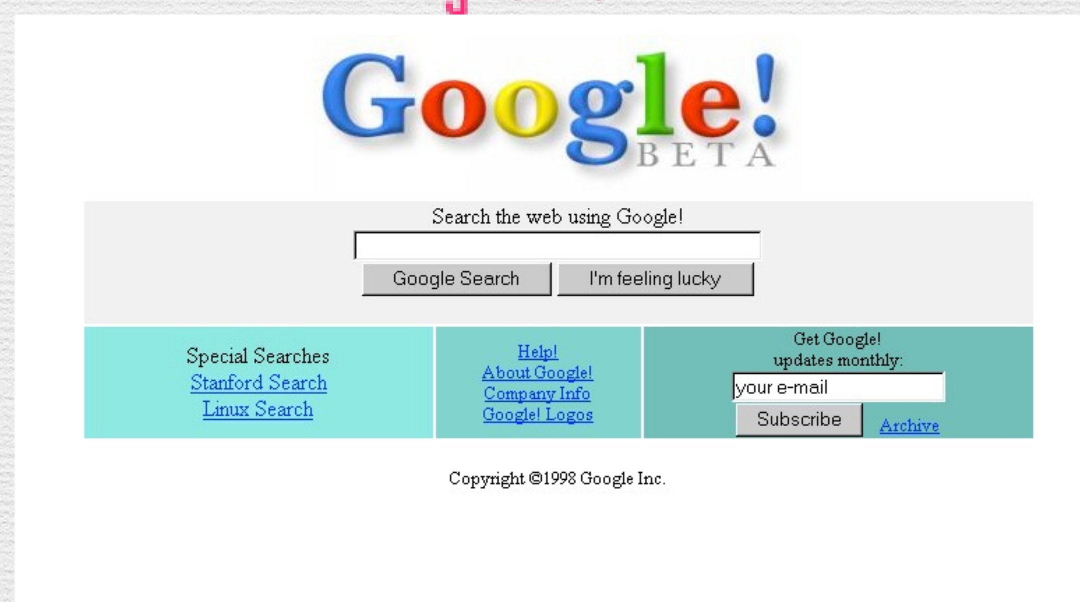
- There have been many tries.
  - For example, many extensions based on Fortran 90.
- Doing anything like this in **general** is very hard.
  - I mean, you can’t even solve the **Halting Problem,** much less more general problems.
- However, specific **subsets** of the problem have shown great progress.

1997

The desire to *automate* search



Larry Page and Sergey Brin



The PageRank Citation Ranking: Bringing Order to the Web

January 29, 1998

Abstract

The importance of a Web page is an inherently subjective matter, which depends on the readers interests, knowledge and attitudes. But there is still much that can be said objectively about the relative importance of Web pages. This paper describes PageRank, a method for rating Web pages objectively and mechanically, effectively measuring the human interest and attention devoted to them.

We compare PageRank to an idealized random Web surfer. We show how to efficiently compute PageRank for large numbers of pages. And, we show how to apply PageRank to search and to user navigation.

1 Introduction and Motivation

The World Wide Web creates many new challenges for information retrieval. It is very large and heterogeneous. Current estimates are that there are over 150 million web pages with a doubling life of less than one year. More importantly, the web pages are extremely diverse, ranging from "What is Joe having for lunch today?" to journals about information retrieval. In addition to these major challenges, search engines on the Web must also contend with inexperienced users and pages engineered to manipulate search engine ranking functions.

However, unlike "flat" document collections, the World Wide Web is hypertext and provides considerable auxiliary information on top of the text of the web pages, such as link structure and

The PageRank Paper

Important

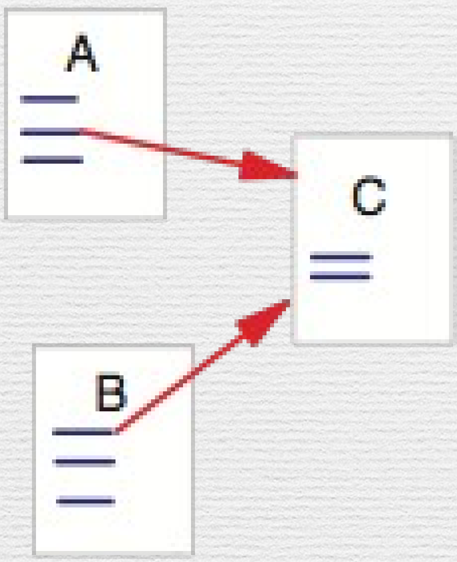


Figure 1: A and B are Backlinks of C

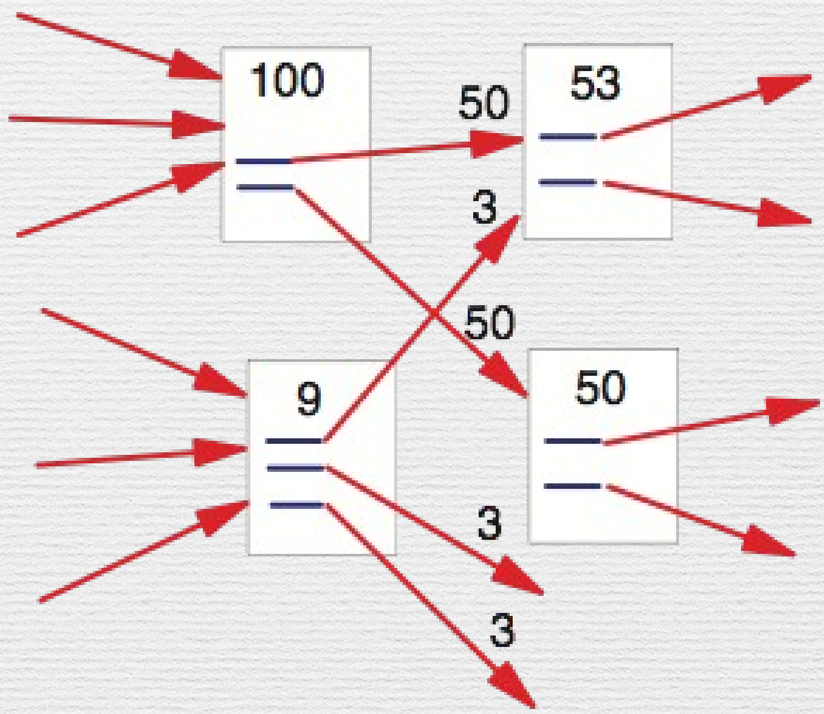


Figure 2: Simplified PageRank Calculation

MapReduce:
History

Some slides based on:

cecs.wright.edu/~tkprasad/courses/cs707/L06MapReduce.ppt

www.eecg.toronto.edu/~amza/ece1747h/slides/MapReduce.1.4.pptx

2003

MapReduce

Sanjay Ghemawat and Jeffrey Dean

How do you make large scale, data centric parallelism accessible for the masses?

Any ideas?

Google

MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

Google, Inc.

Abstract

MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a *map* function that processes a key/value pair to generate a set of intermediate key/value pairs, and a *reduce* function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown in the paper.

Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The run-time system takes care of the details of partitioning the input data, scheduling the pro-

given day, etc. Most such computations are conceptually straightforward. However, the input data is usually large and the computations have to be distributed across hundreds or thousands of machines in order to finish in a reasonable amount of time. The issues of how to parallelize the computation, distribute the data, and handle failures conspire to obscure the original simple computation with large amounts of complex code to deal with these issues.

As a reaction to this complexity, we designed a new abstraction that allows us to express the simple computations we were trying to perform but hides the messy details of parallelization, fault-tolerance, data distribution and load balancing in a library. Our abstraction is in-

MapReduce Vs. Hadoop

Important

- MapReduce is an **idea**
  - **A way to organize code so that it is easy to parallelize.**
- Hadoop is (one) implementation of the MapReduce idea.
  - hadoop.apache.org

MapReduce/Hadoop: Diving more deeply

Some slides based on:

cecs.wright.edu/~tkprasad/courses/cs707/L06MapReduce.ppt

www.eecg.toronto.edu/~amza/ece1747h/slides/MapReduce.1.4.pptx

MapReduce

- Programming model for distributed computations
- Software framework for clusters
- Massive data processing
- No hassle with low level programming
  - Partitioning input data
  - Scheduling execution
  - Handling failures
  - Intermachine communication

- Open source implementation
- MRJob: Python class for Hadoop Streaming

Part 1: Distributed Filesystem

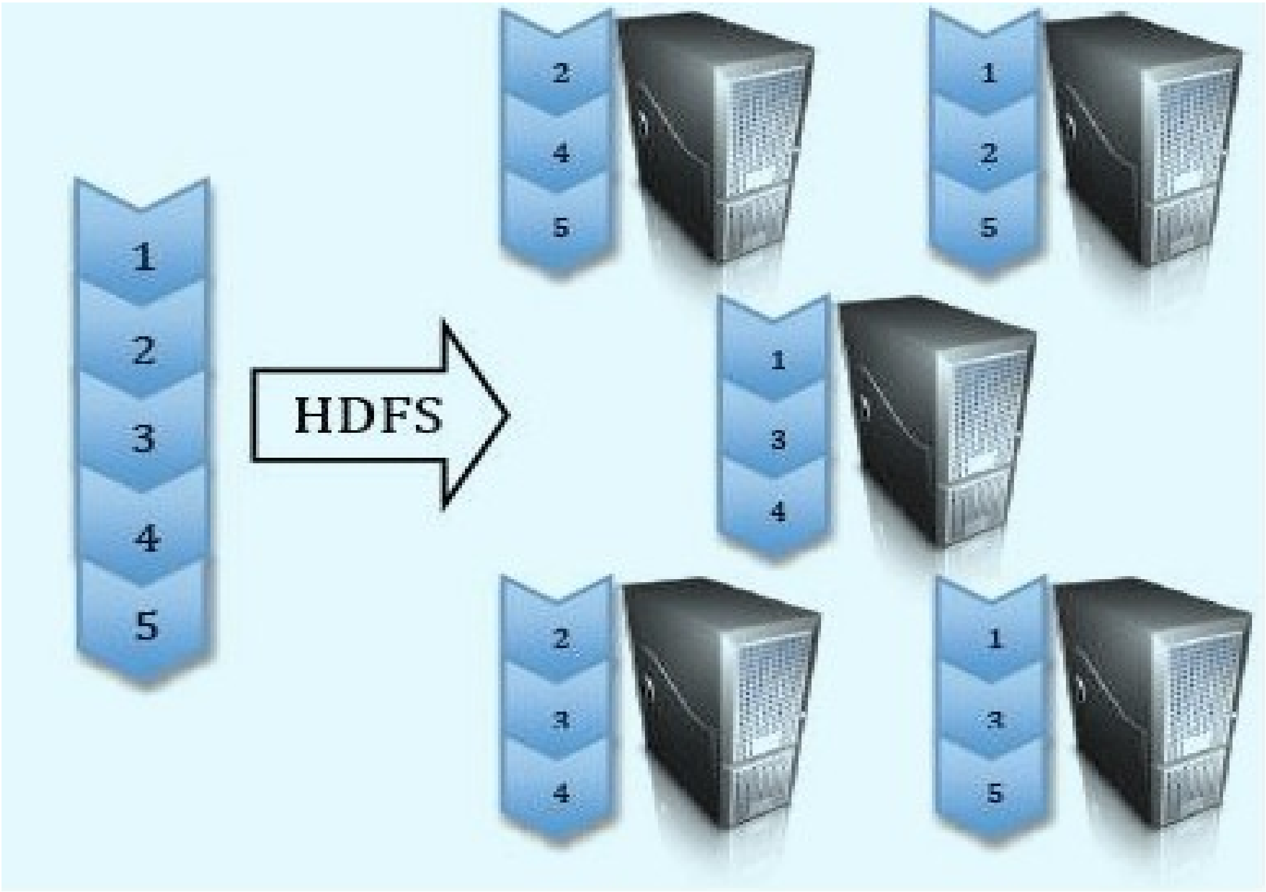
Stable storage

- First order problem: if nodes can fail, how can we store data persistently?
- Answer: Distributed File System
  - Provides global file namespace
  - Google GFS; Hadoop HDFS; Kosmix KFS
- Typical usage pattern
  - Huge files (100s of GB to TB)
  - Data is rarely updated in place
  - Reads and appends are common

cecs.wright.edu/~tkprasad/courses/cs707/L06MapReduce.ppt

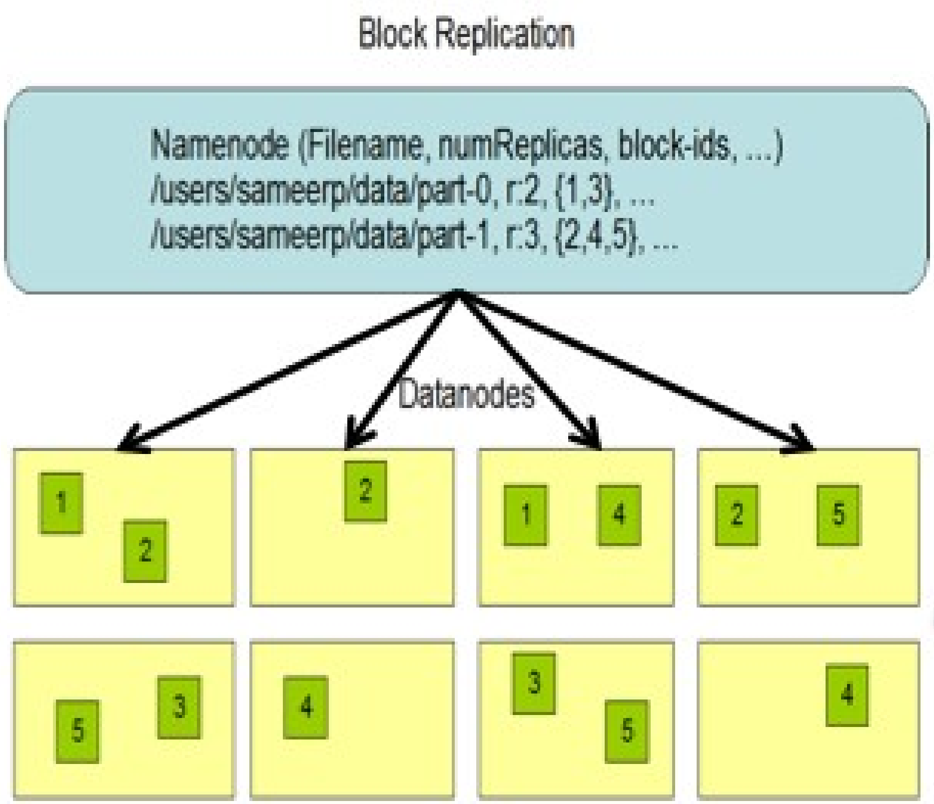
Google File System (GFS)
Hadoop Distributed File System (HDFS)

- Split data and store 3 replica on commodity servers

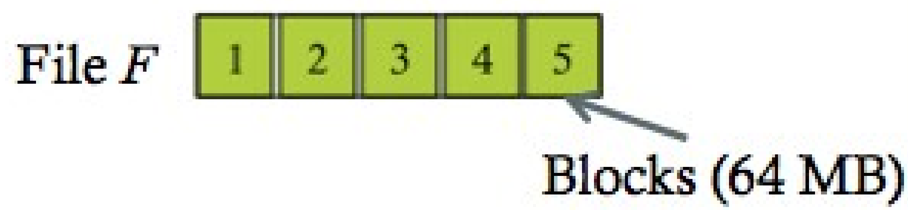


www.eecg.toronto.edu/~amza/ece1747h/slides/MapReduce.1.4.pptx

Hadoop Distributed File System (HDFS)



Centralized namenode
- Maintains metadata info about files



Many datanode (1000s)
- Store the actual data
- Files are divided into blocks
- Each block is replicated *N* times (Default = 3)

Main Properties of HDFS

Important

- ***Large:*** A HDFS instance may consist of thousands of server machines, each storing part of the file system's data
- ***Replication:*** Each data block is replicated many times (default is 3)
- ***Failure:*** Failure is the norm rather than exception
- ***Fault Tolerance:*** Detection of faults and quick, automatic recovery from them is a core architectural goal of HDFS
  - Namenode is consistently checking Datanodes

Part 1: The “map and reduce” part

MapReduce

Important

- Map
  - Grab the relevant data from the source
  - User function gets called for each chunk of input

- Reduce
  - Aggregate the results
  - User function gets called for each unique key

Map example

Important

- (map ***f list [list$_2$ list$_3$ …]***)

Unary operator

- (map square ‘(1 2 3 4))
  - (1 4 9 16)

cecs.wright.edu/~tkprasad/courses/cs707/L06MapReduce.ppt

Reduce Example

Important

- (reduce ***f id list***)

Binary operator

- (reduce + 0 ‘(1 4 9 16))
  - (+ 16 (+ 9 (+ 4 (+ 1 0))))
  - 30

cecs.wright.edu/~tkprasad/courses/cs707/L06MapReduce.ppt

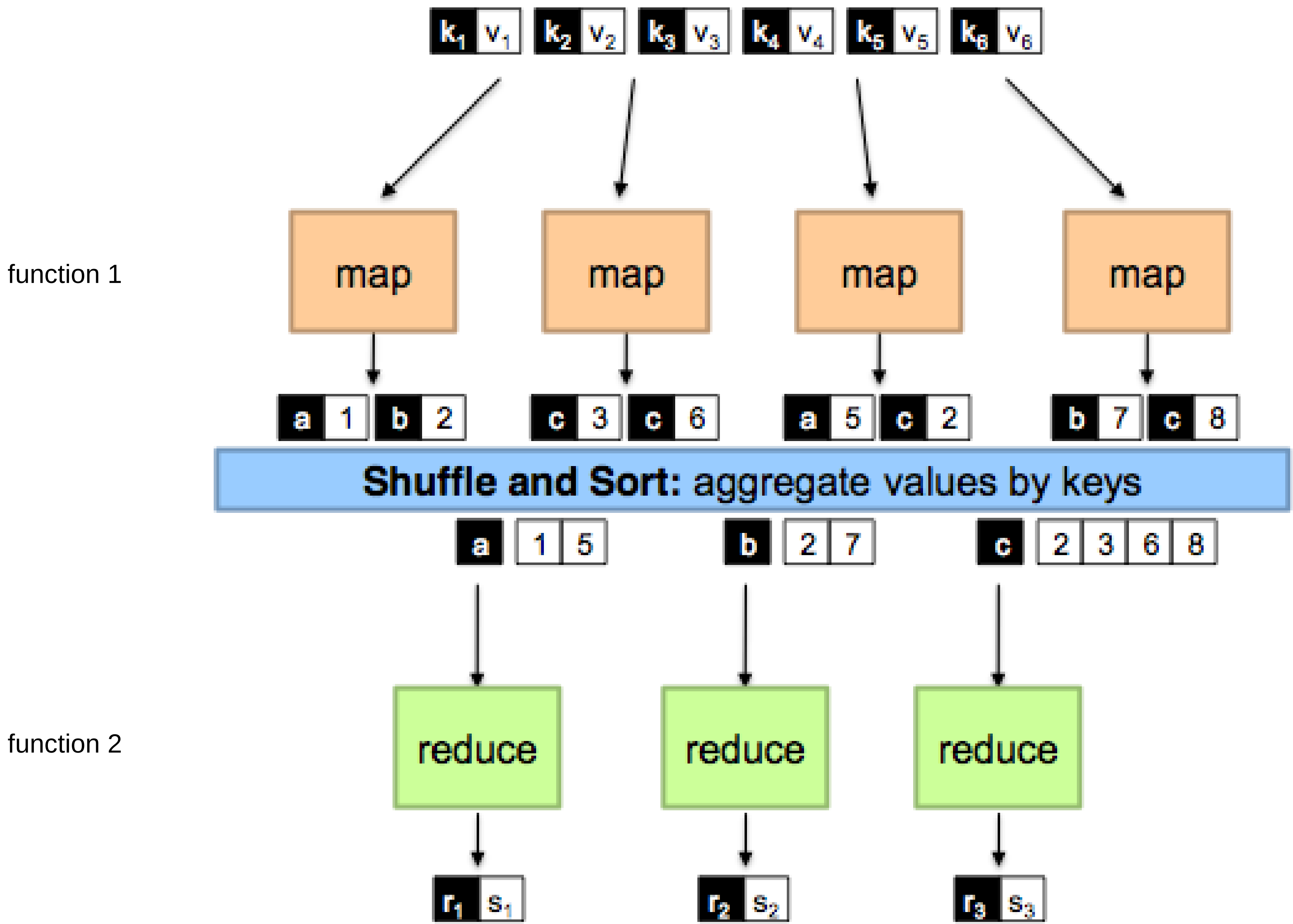
Key- Value Pairs

Important

- Mappers and Reducers are users' code (provided functions)
- Just need to obey the Key-Value pairs interface
- **Mappers:**
  - Consume <key, value> pairs
  - Produce <key, value> pairs
- **Reducers:**
  - Consume <key, <list of values>>
  - Produce <key, value>
- **Shuffling and Sorting:**
  - Hidden phase between mappers and reducers
  - Groups all similar keys from all mappers, sorts and passes them to a certain reducer in the form of <key, <list of values>>

{ "name" => "Alex Tamoykin" }

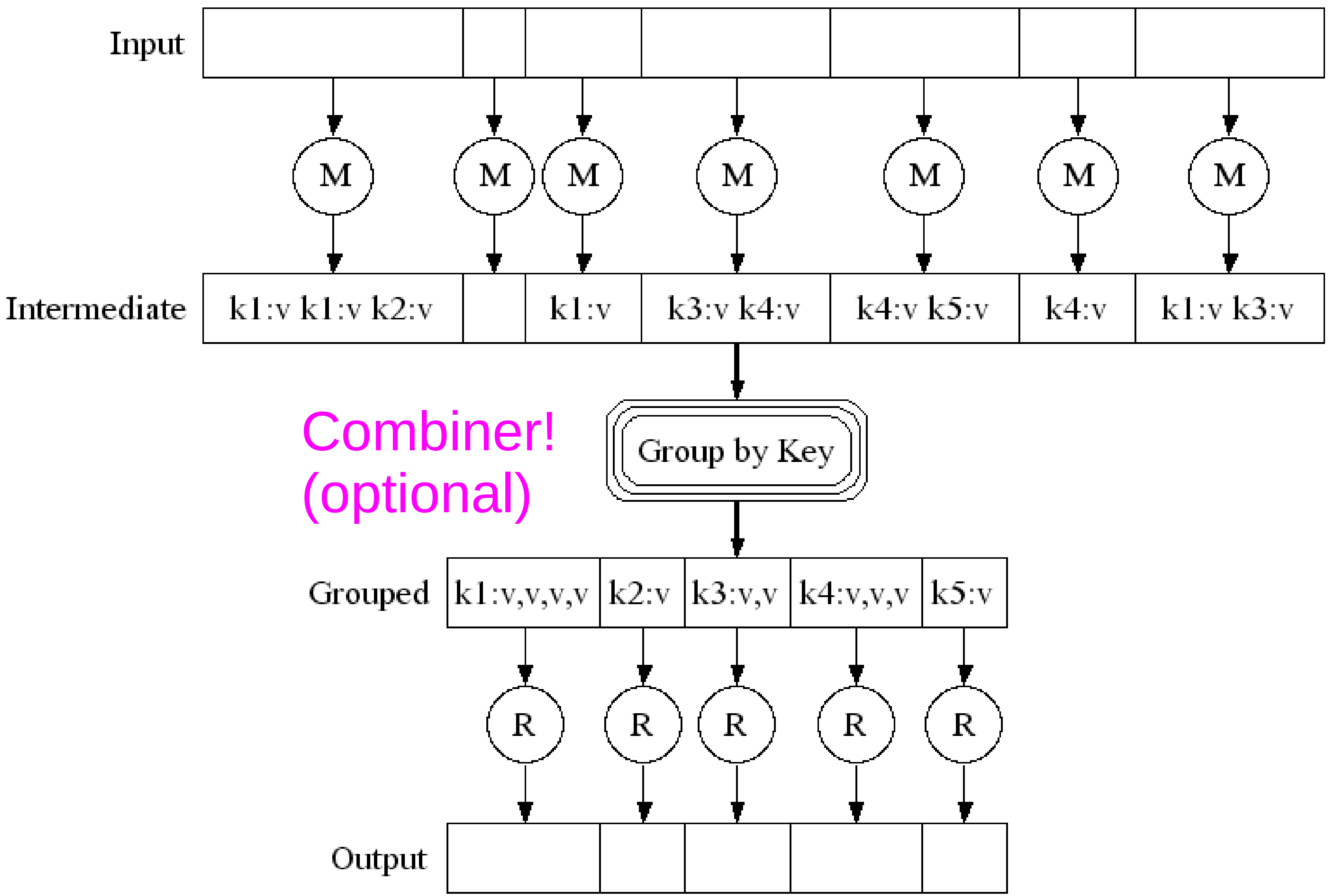
Key **Value**



k_1 v_1 k_2 v_2 k_3 v_3 k_4 v_4 k_5 v_5 k_6 v_6
function 1
map
map
map
map
a 1 b 2
c 3 c 6
a 5 c 2
b 7 c 8
Shuffle and Sort: aggregate values by keys
a 1 5
b 2 7
c 2 3 6 8
function 2
reduce
reduce
reduce
r_1 s_1
r_2 s_2
r_3 s_3

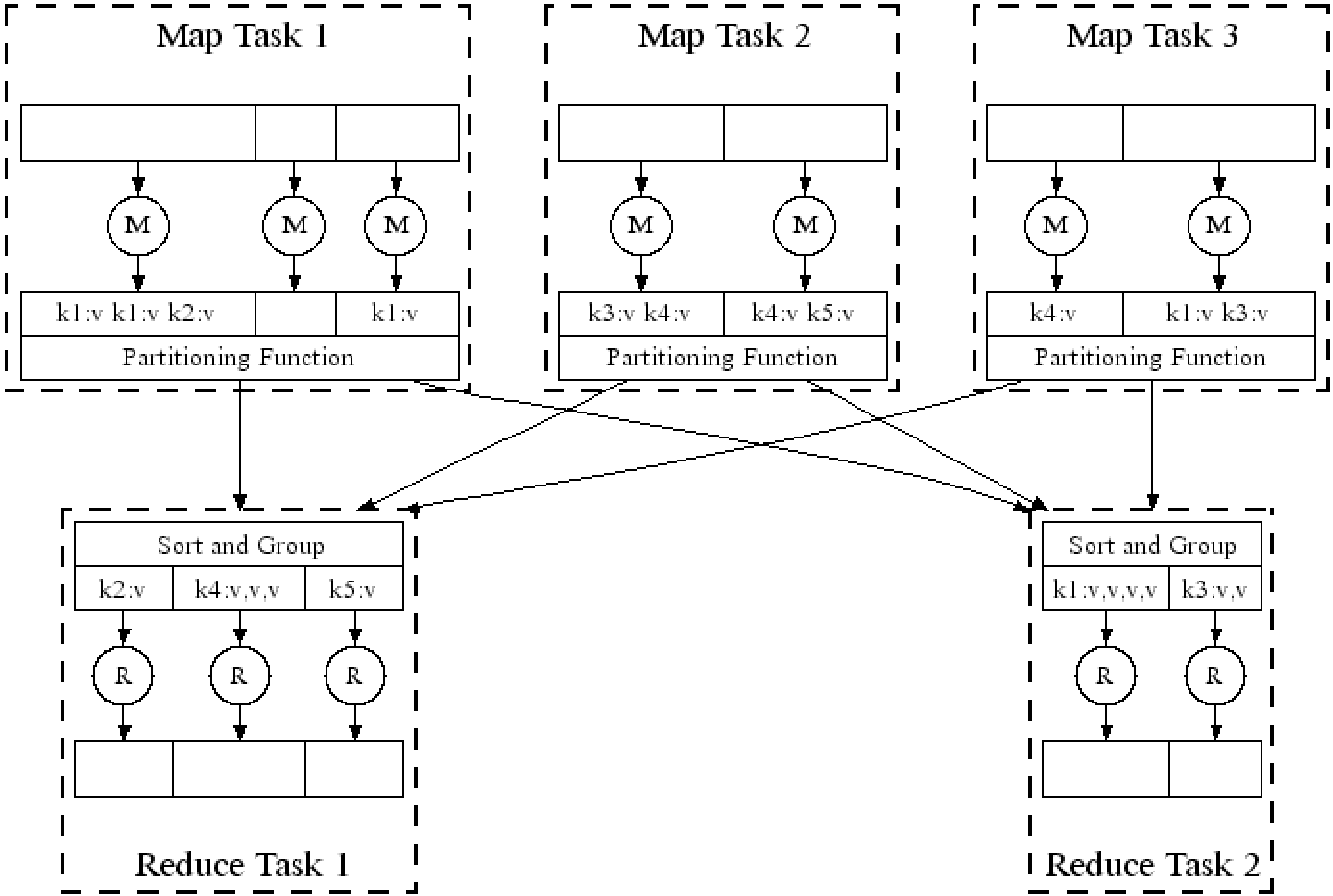
What it really should be called...

- Not “MapReduce”
- But “MapShuffleReduce”


Input
M
M
M
M
M
M
M
Intermediate
k1:v k1:v k2:v
k1:v
k3:v k4:v
k4:v k5:v
k4:v
k1:v k3:v
Combiner!
(optional)
Group by Key
Grouped
k1:v,v,v,v
k2:v
k3:v,v
k4:v,v,v
k5:v
R
R
R
R
R
Output

What it really should be called...

- Not “MapReduce”
- But “MapShuffleCombineReduce”



Map Task 1
Map Task 2
Map Task 3
M
k1:v k1:v k2:v
k1:v
k3:v k4:v
k4:v k5:v
k4:v
k1:v k3:v
Partitioning Function
Sort and Group
k2:v
k4:v,v,v
k5:v
k1:v,v,v,v
k3:v,v
R
Reduce Task 1
Reduce Task 2

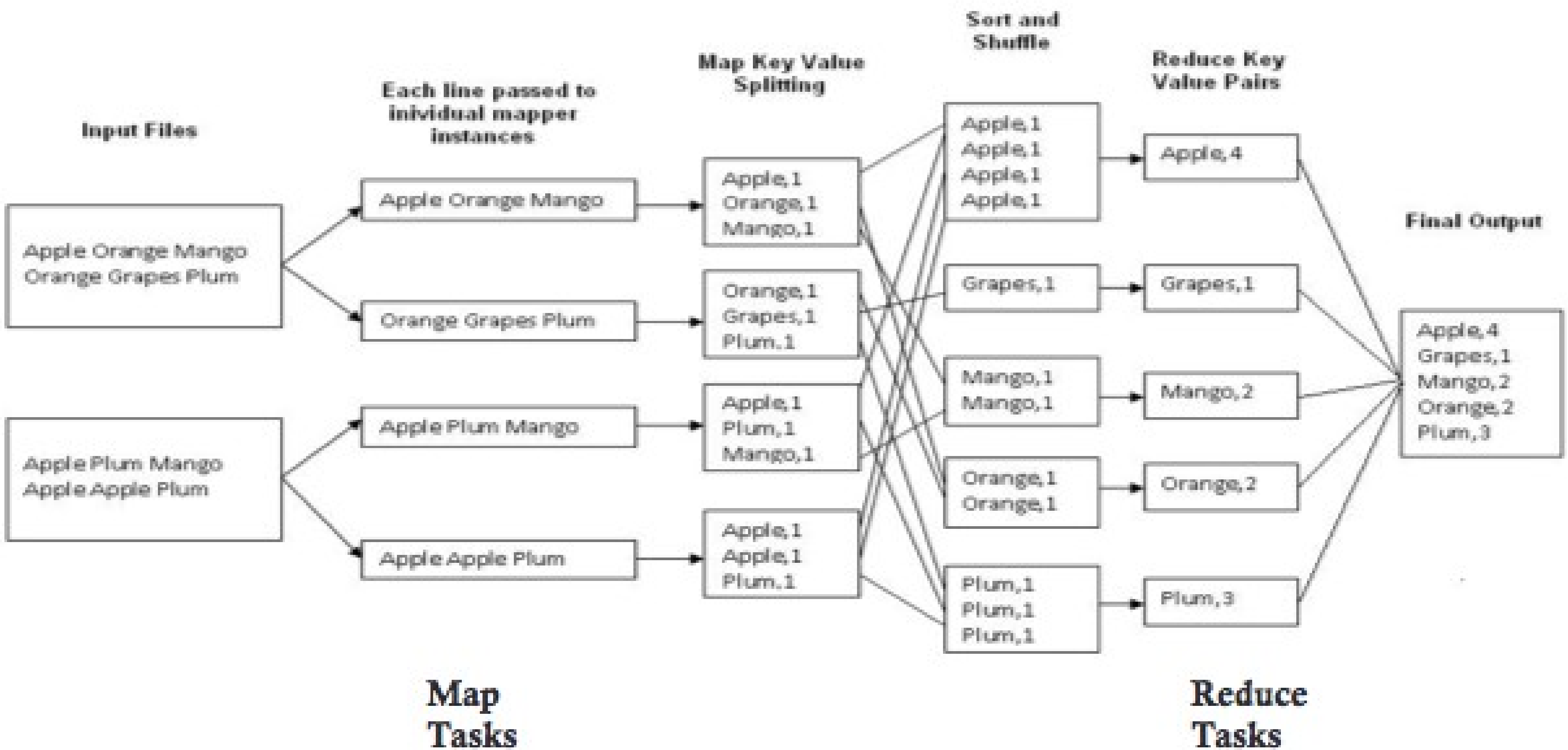
What it really should be called...

- Not “MapReduce”
- But “MapShuffleCombinePartitionReduce”

Example 1: Word Count

Important

- **Job: Count the occurrences of each word in a data set**

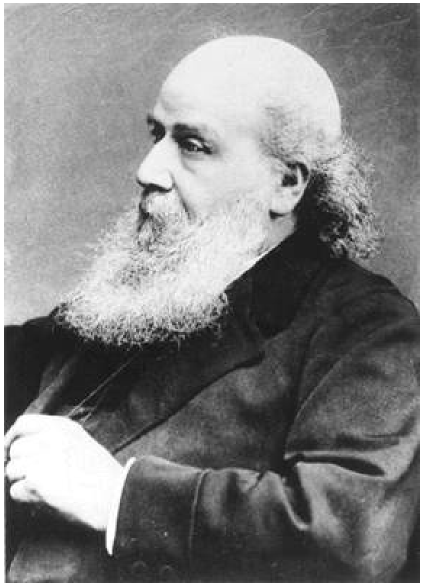


Class 11
Graph data

History

- The paper written by Leonhard Euler on the Seven Bridges of Königsberg and published in 1736 is regarded as the first paper in the history of graph theory.
- The term "graph" was introduced by Sylvester in a paper published in 1878 in Nature, where he draws an analogy between invariants in algebra and molecular diagrams.

Leonhard Euler



James Joseph Sylvester

https://en.wikipedia.org/wiki/Graph_theory

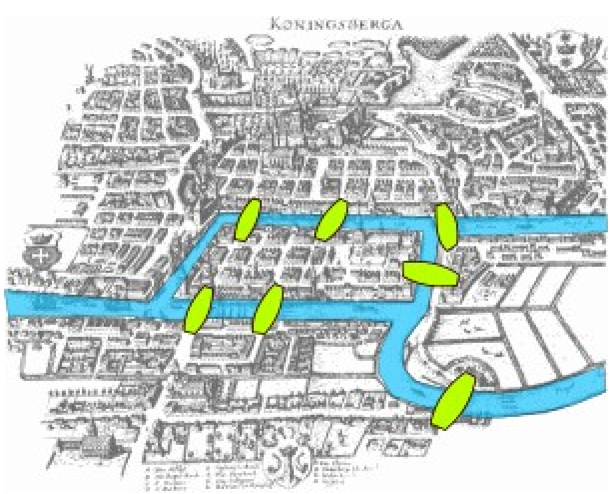
WPI

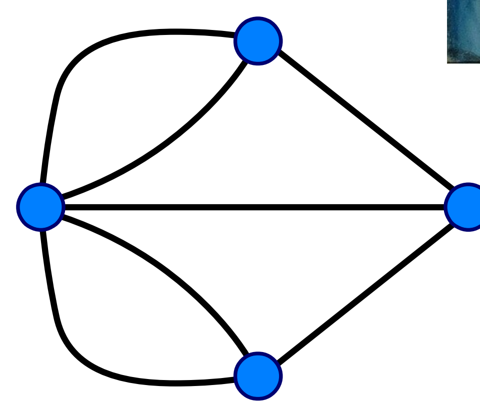
That seems simple, so why all the interest?

- Graphs lead to many deep and beautiful mathematical concepts!

Bridges of Konigsberg

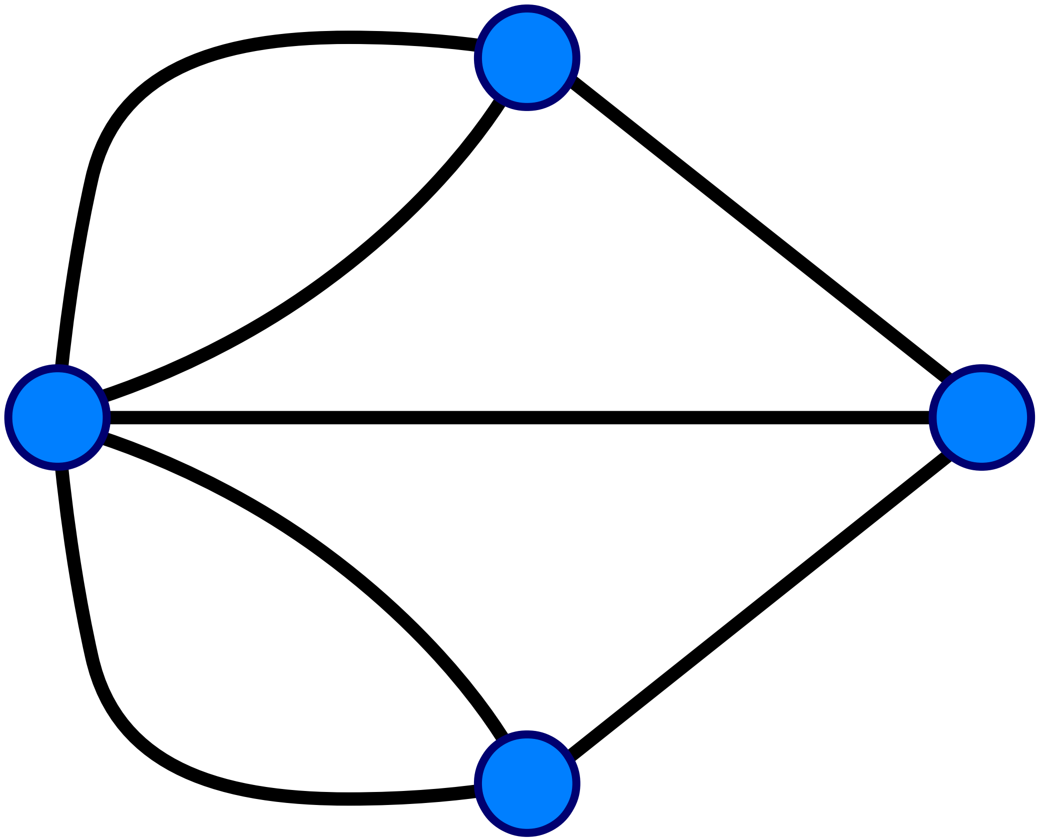
Leonhard Euler





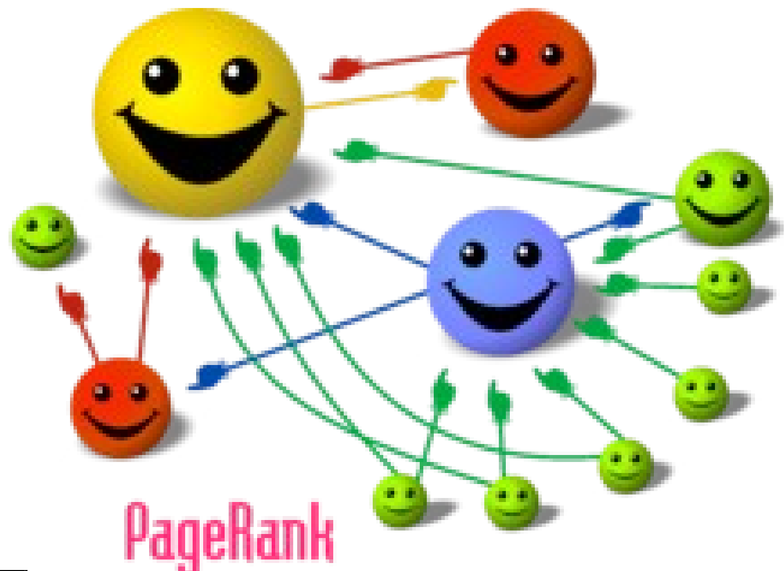
Example

- Can you cross every bridge exactly once?



1997

The desire to *automate* search



Larry Page and Sergey Brin

The PageRank Citation Ranking: Bringing Order to the Web

January 29, 1998

Abstract

The importance of a Web page is an inherently subjective matter, which depends on the readers interests, knowledge and attitudes. But there is still much that can be said objectively about the relative importance of Web pages. This paper describes PageRank, a method for rating Web pages objectively and mechanically, effectively measuring the human interest and attention devoted to them.

We compare PageRank to an idealized random Web surfer. We show how to efficiently compute PageRank for large numbers of pages. And, we show how to apply PageRank to search and to user navigation.

1 Introduction and Motivation

The World Wide Web creates many new challenges for information retrieval. It is very large and heterogeneous. Current estimates are that there are over 150 million web pages with a doubling life of less than one year. More importantly, the web pages are extremely diverse, ranging from "What is Joe having for lunch today?" to journals about information retrieval. In addition to these major challenges, search engines on the Web must also contend with inexperienced users and pages engineered to manipulate search engine ranking functions.

However, unlike "flat" document collections, the World Wide Web is hypertext and provides considerable auxiliary information on top of the text of the web pages, such as link structure and

WPI

The PageRank Paper

Important

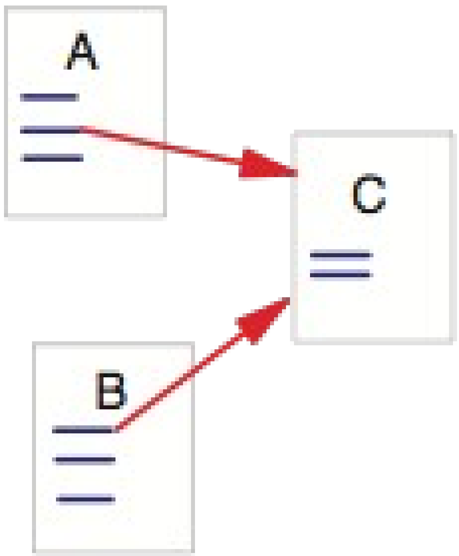


Figure 1: A and B are Backlinks of C

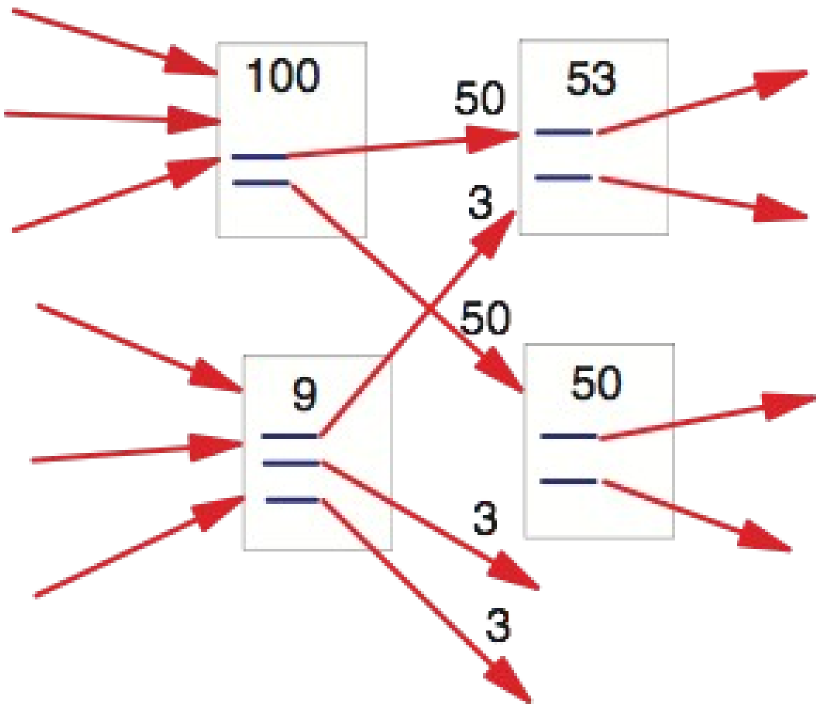


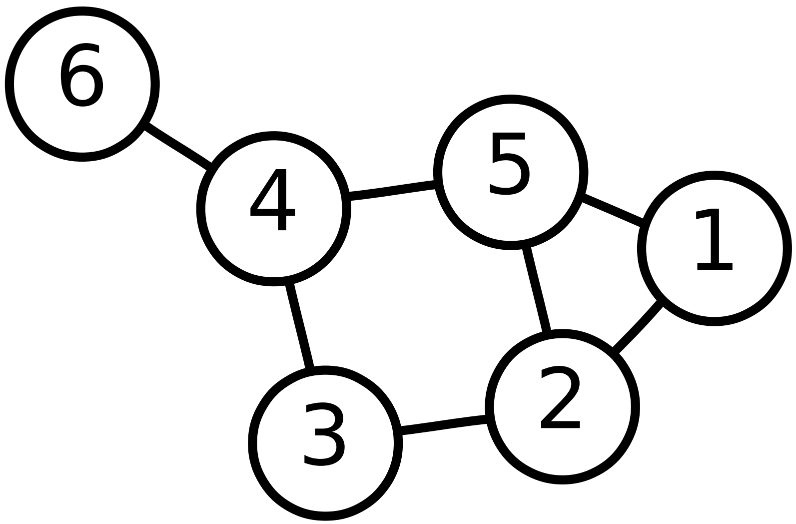
Figure 2: Simplified PageRank Calculation

Graph Representation

Important

- Mathematical object consisting of
  - V : A set of nodes/vertices/point
  - E : A set of connection/links between pairs of nodes
- G(V, E)

Example



V =

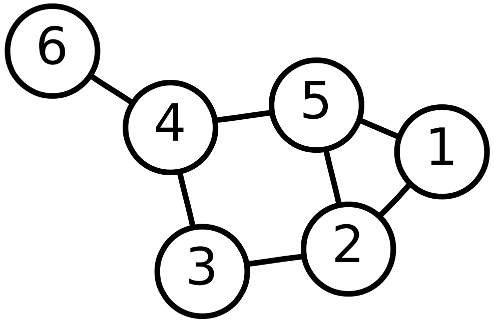
E =

Types of graphs

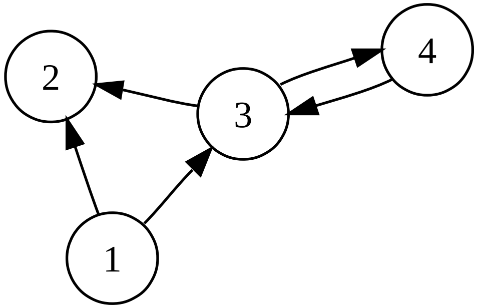
- Edge types:
  - Undirected
    - Example?
  - Directed
    - Example?
  - Loops

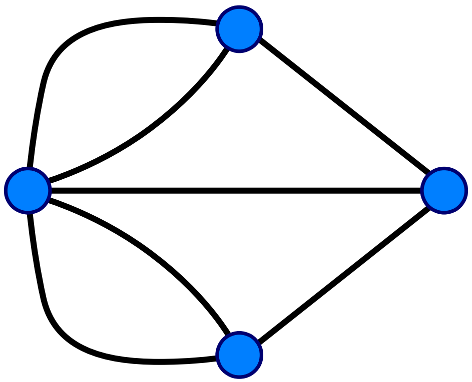
Types of graphs

Important



- Graph types
  - Undirected
  - Directed
  - Mixed
  - Multi-graphs

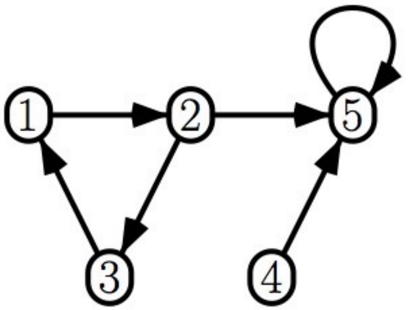


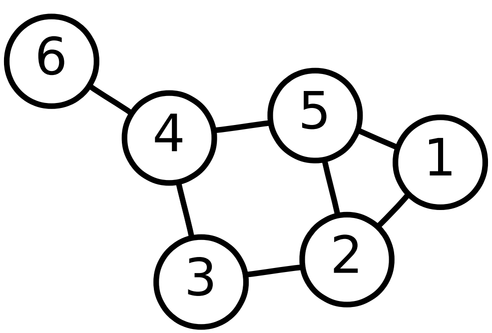


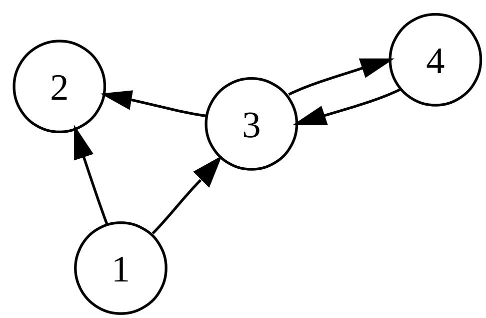
Types of graphs

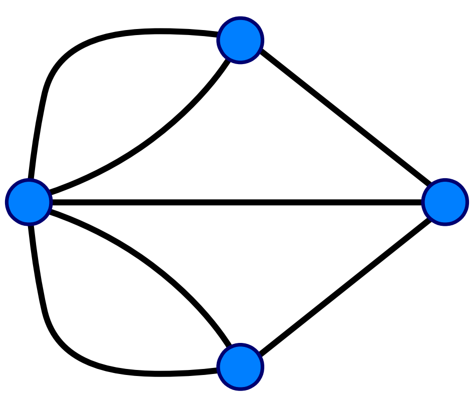
Important

- Simple graphs
  - Undirected
  - No loops or multiple edges

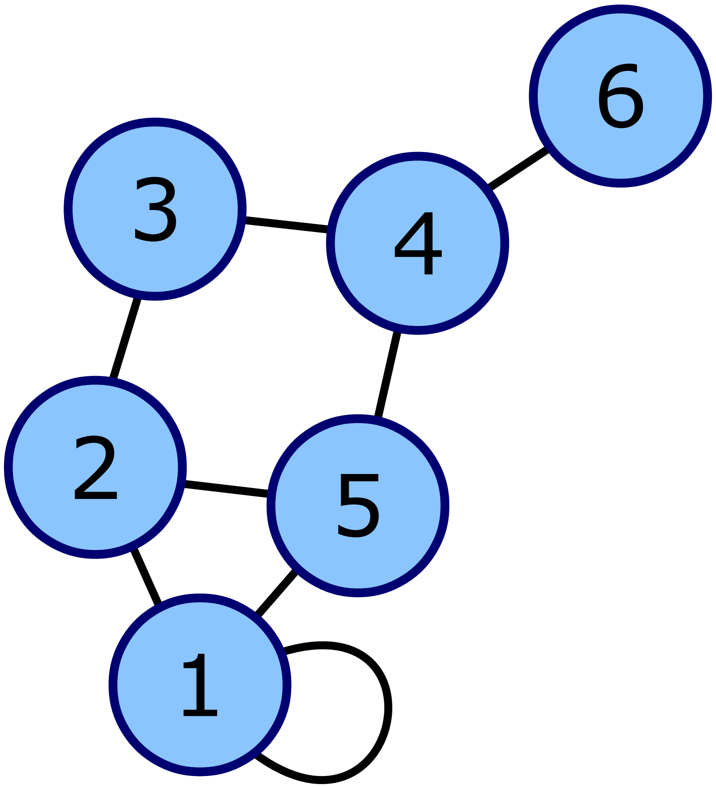








Two views of a graph



Adjacency matrix

$$\begin{pmatrix} 2 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

“Large Graph”

- Of course, in Data Science we are interested in “Large Graphs”
  - What makes a graph large?
  - Can you give some examples?

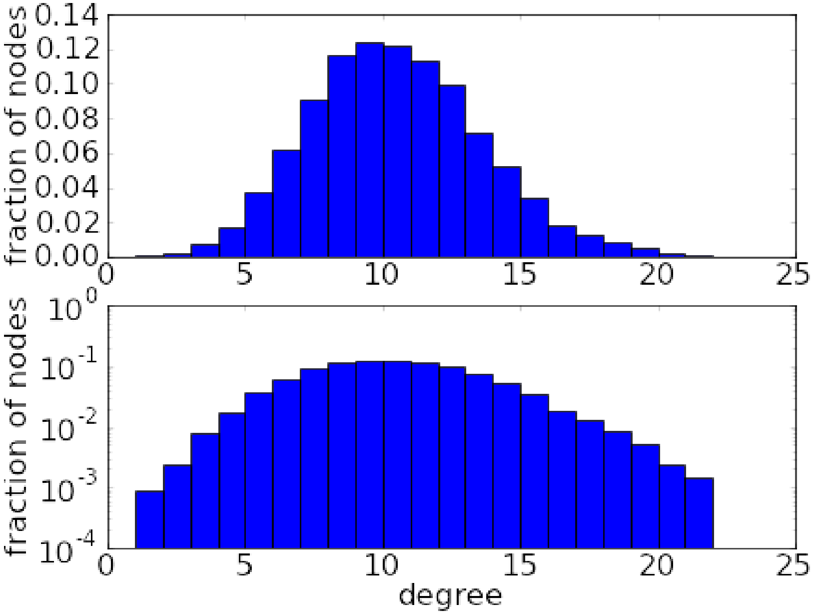
Warm up problem

- Given a **simple** graph with 10 vertices, how many edges can you have?

$\frac{10 \cdot 9}{2} = 45$

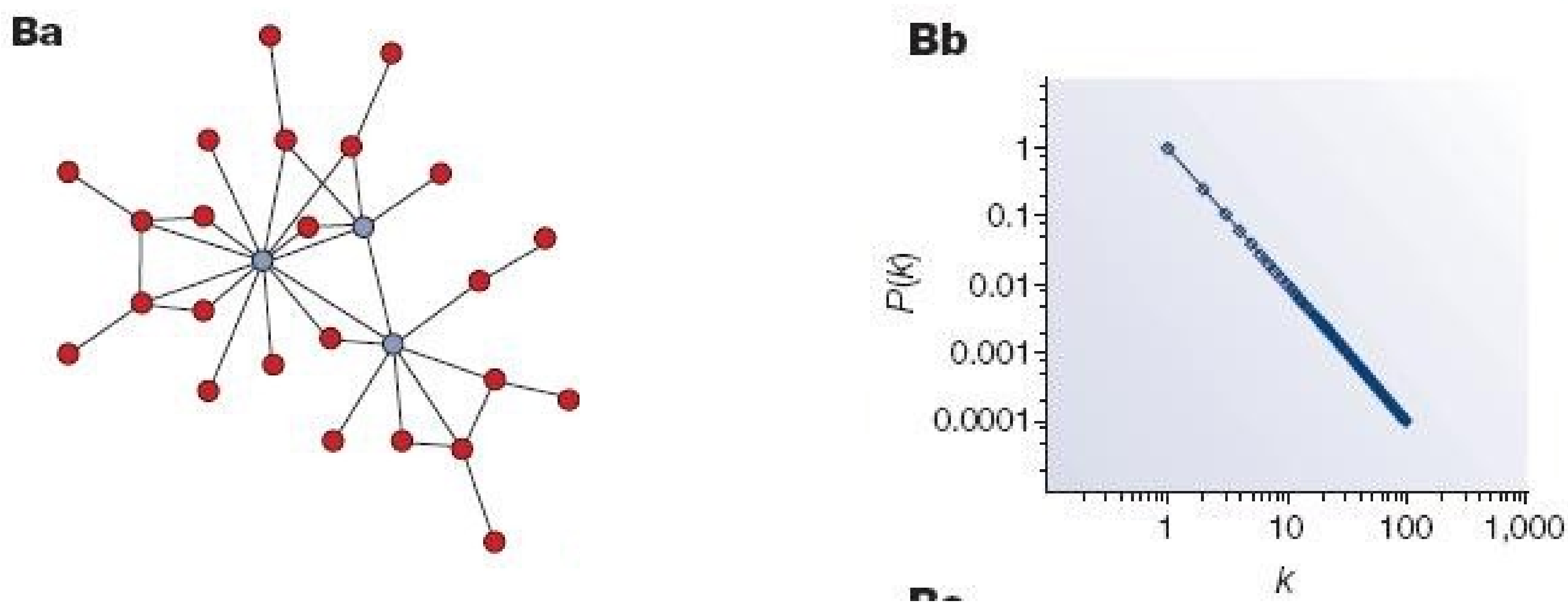
Degree Histogram

- Outdegree of a vertex = # outgoing edges
- For each number d, let n(d) = # vertices with outdegree d



Zipf Distribution (power law)

- n(d) ~ 1/ d^x for some value x > 0
- Human-generated data has Zipf distribution: letters in alphabet, words in vocabulary.
- In log-log scale



Distribution of the Web

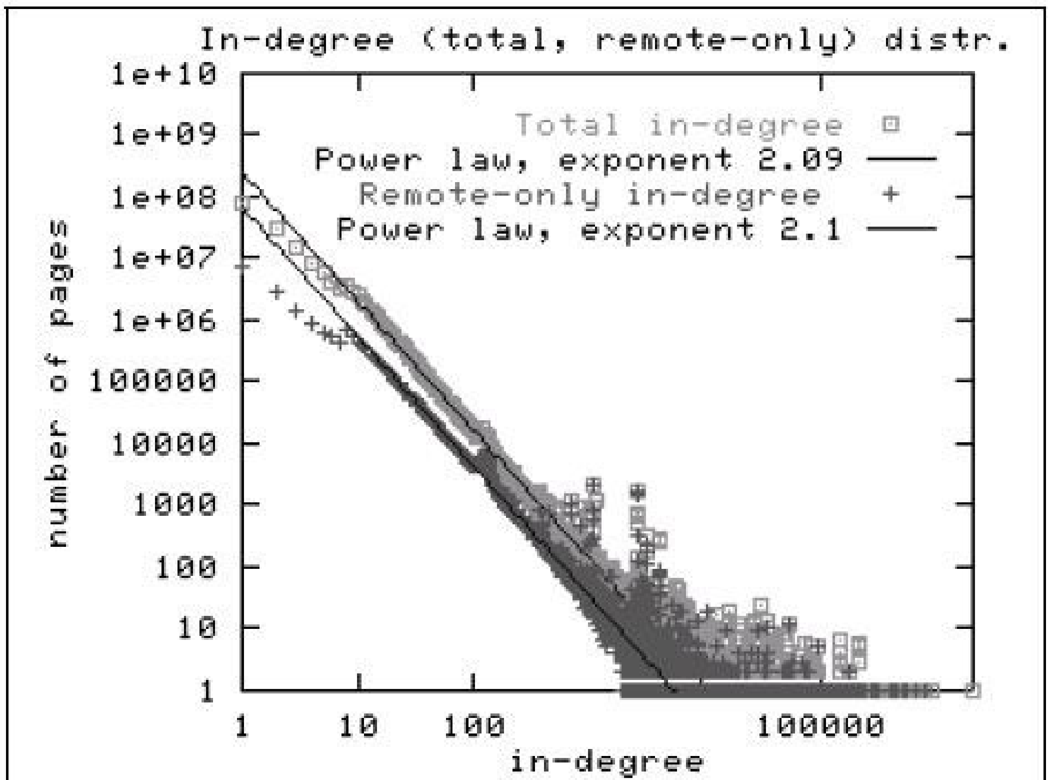


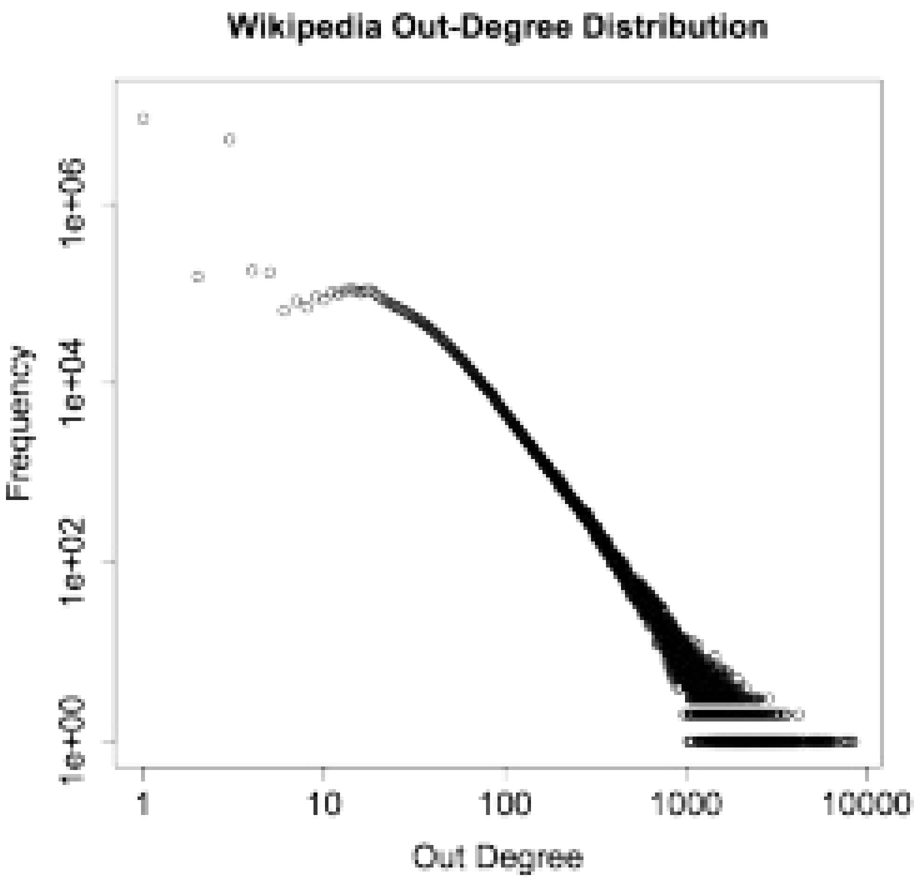
Figure 2: In-degree distribution.

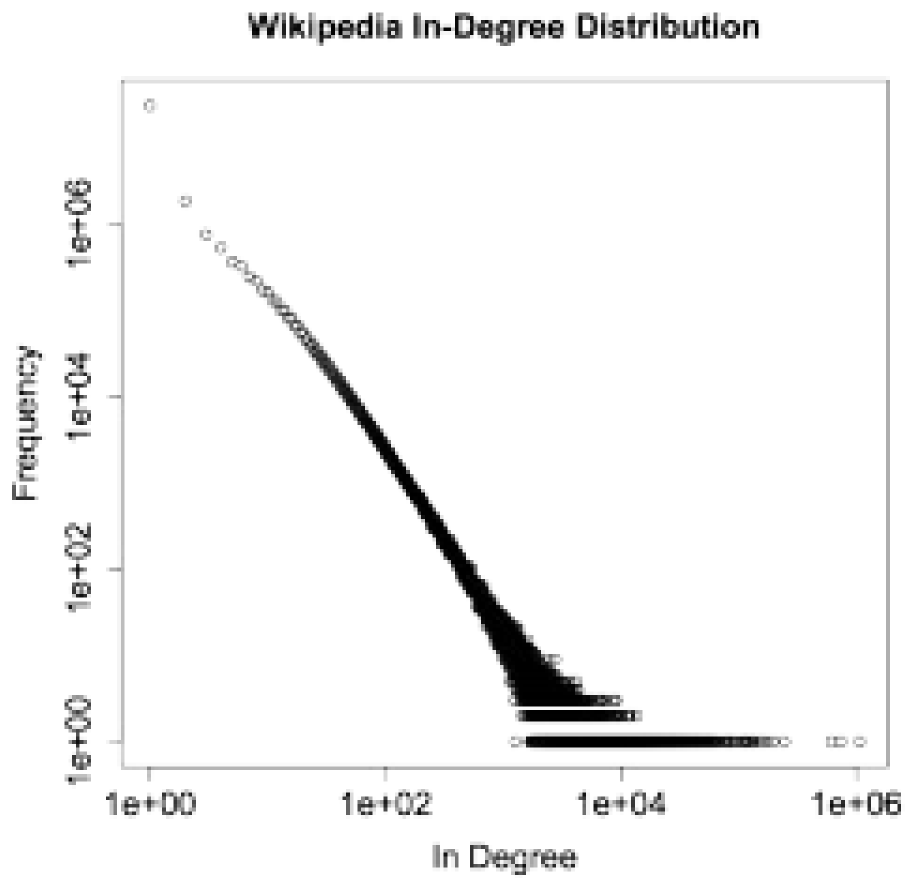
Late 1990's
200M Webpages

Exponential ?

Power Law?

Distribution of Wikipedia





Class 12
Graph Data and High Dimensional Data

WPI

Evaluating the Nodes

- How would you define the important nodes in a graph?

Evaluating the Nodes

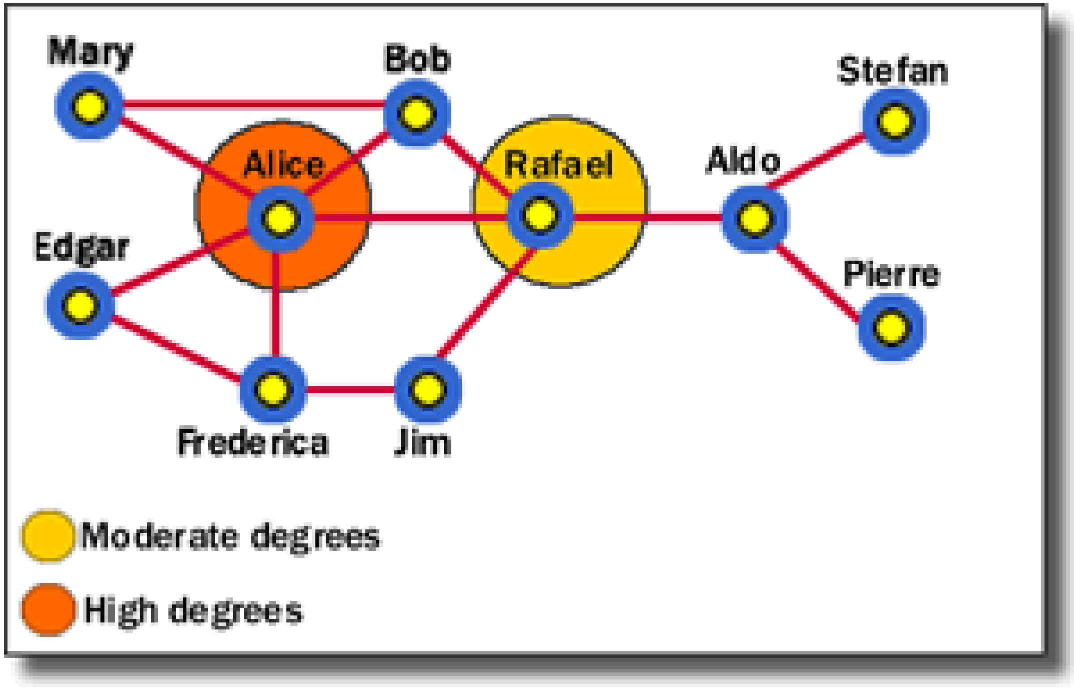
Important

- Degree Centrality
- Betweenness : Number paths
- Eigenvector Centrality (i.e, PageRank)

Node: **Degree** Centrality

Important

- **Degree** Centrality: number of edges



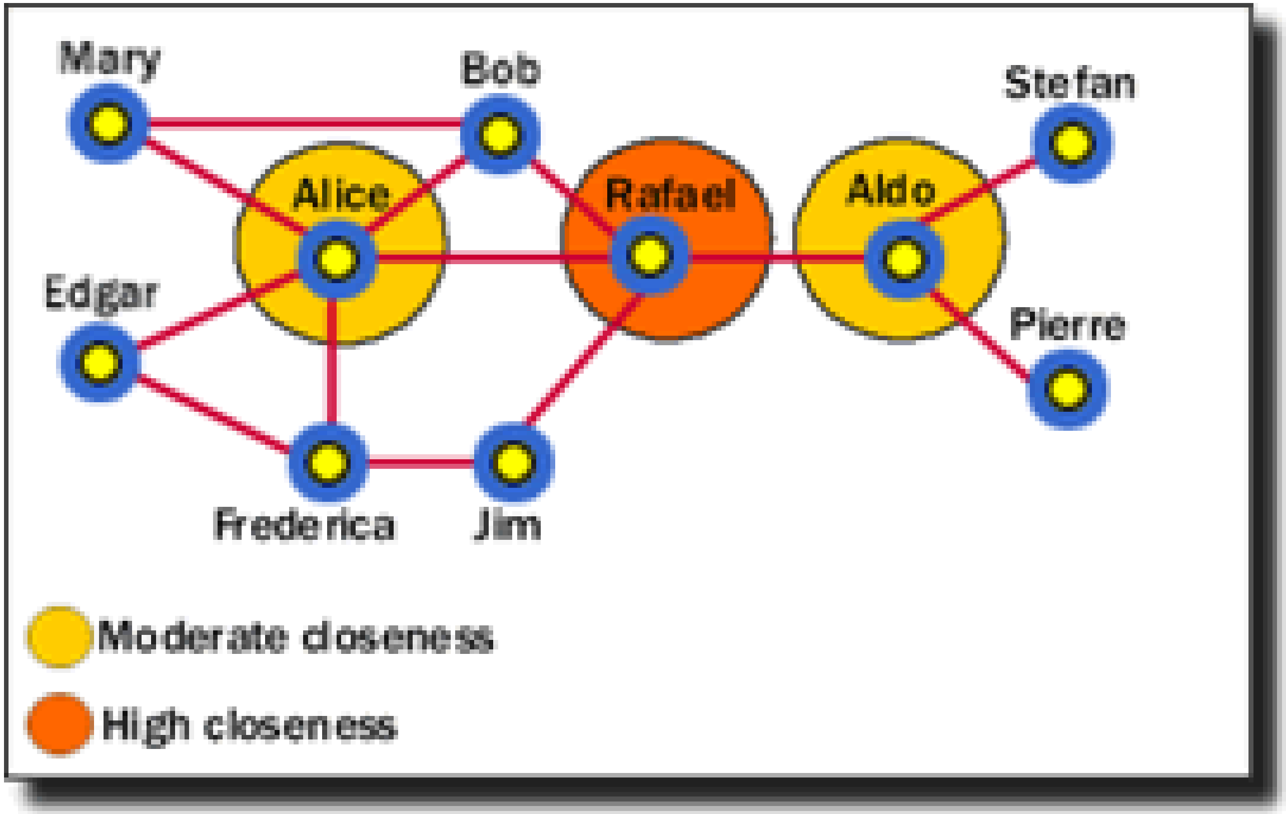
Pros: Is generally an active player in the network

Cons: ignore the whole graph

Node: **Closeness** Centrality

Important

- **Closeness** of a vertex: the inverse of the average length of all its shortest paths



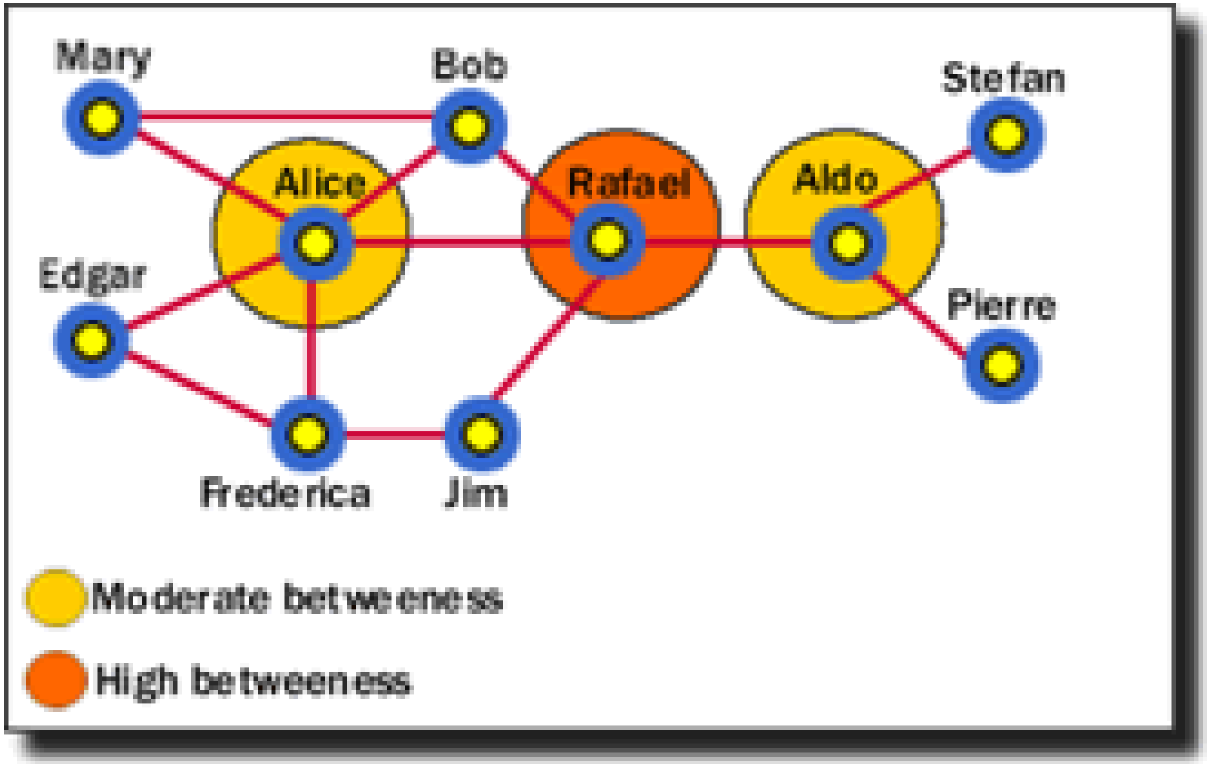
Pros: Is close to other entities

Cons: ignore the importance of different nodes

Node: **Betweenness** Centrality

Important

- **Betweeness** of a vertex *v*: the fraction of all shortest paths that pass through *v*.

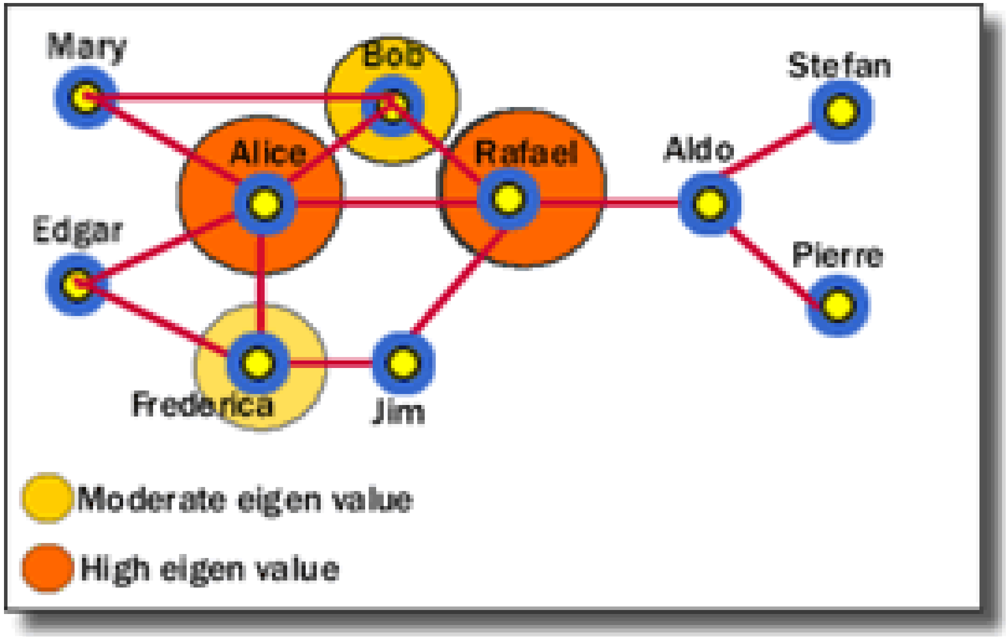


Pros: represents a single point of failure, has a greater amount of influence over what happens in a network.
Cons: ignore the importance of different nodes

Node: **Eigenvalue** (PageRank)

Important

- Eigenvalue measures how ***close*** an entity is to other highly ***close*** entities within a network



Idea of PageRank

Important

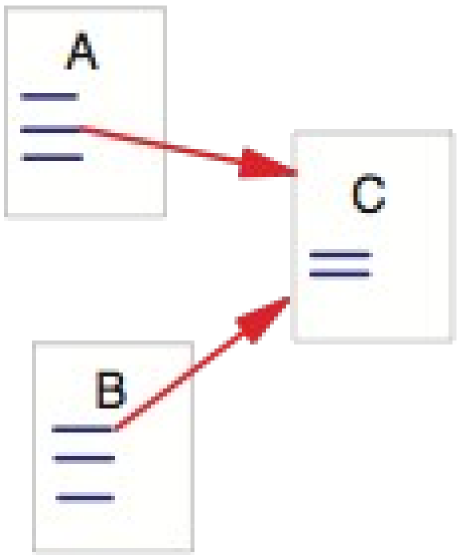


Figure 1: A and B are Backlinks of C

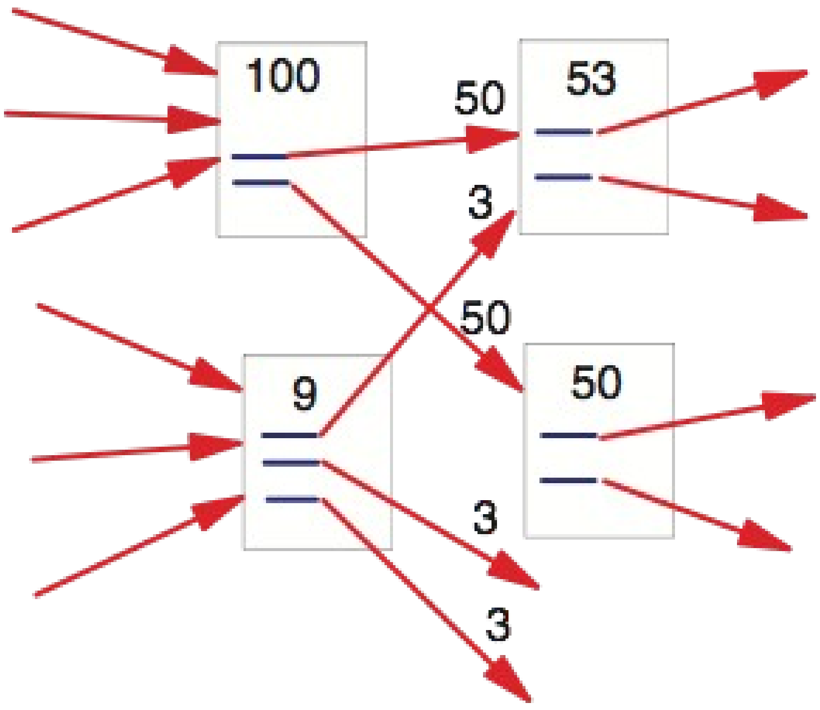
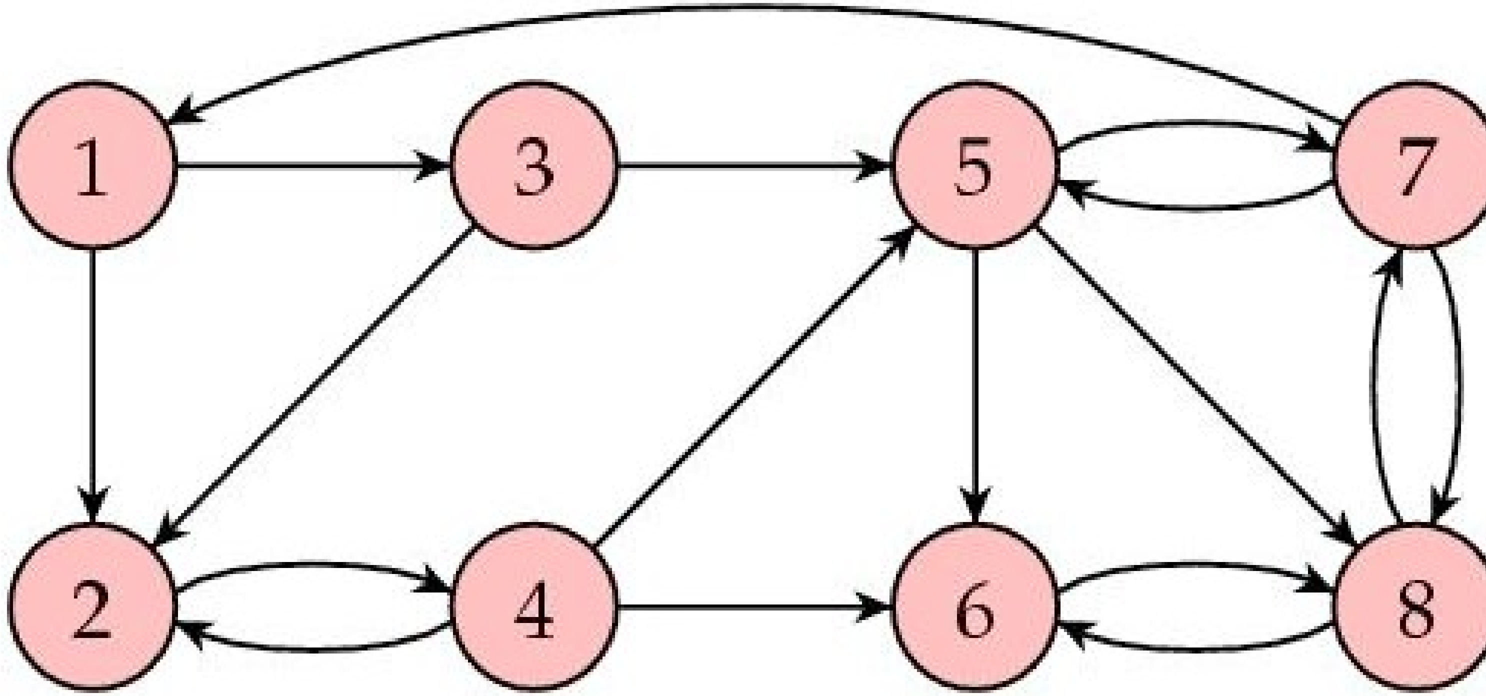


Figure 2: Simplified PageRank Calculation



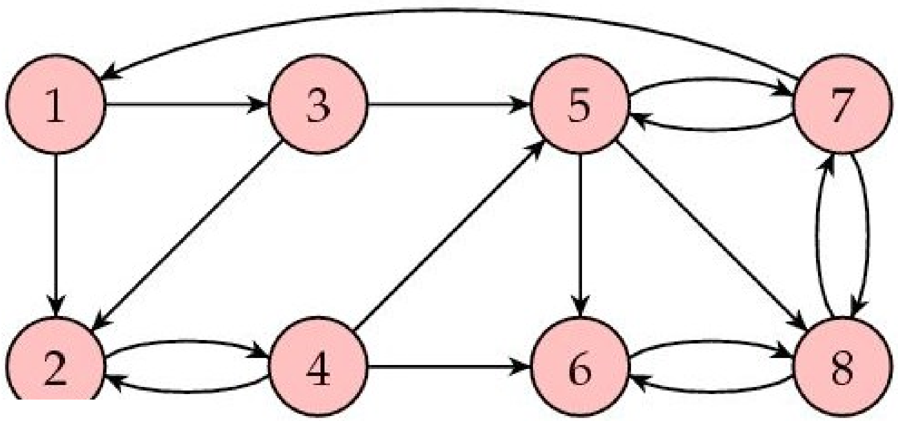
The corresponding matrix is

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 \\ 1/2 & 0 & 1/2 & 1/3 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 1 & 1/3 & 0 \end{bmatrix}$$

Probabilistic Interpretation of H

Important

Random Surfing



$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 \\ 1/2 & 0 & 1/2 & 1/3 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/3 & 0 & 0 & 1/3 & \\ 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 1 & 1/3 & 0 \end{bmatrix}$$

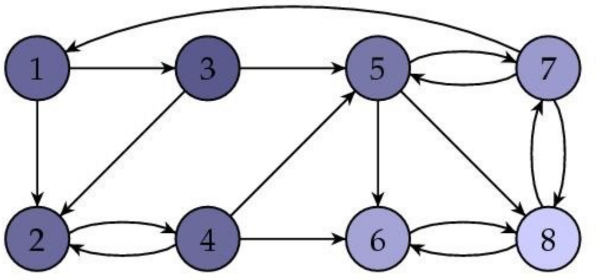
How to Compute!?
Power method

Important

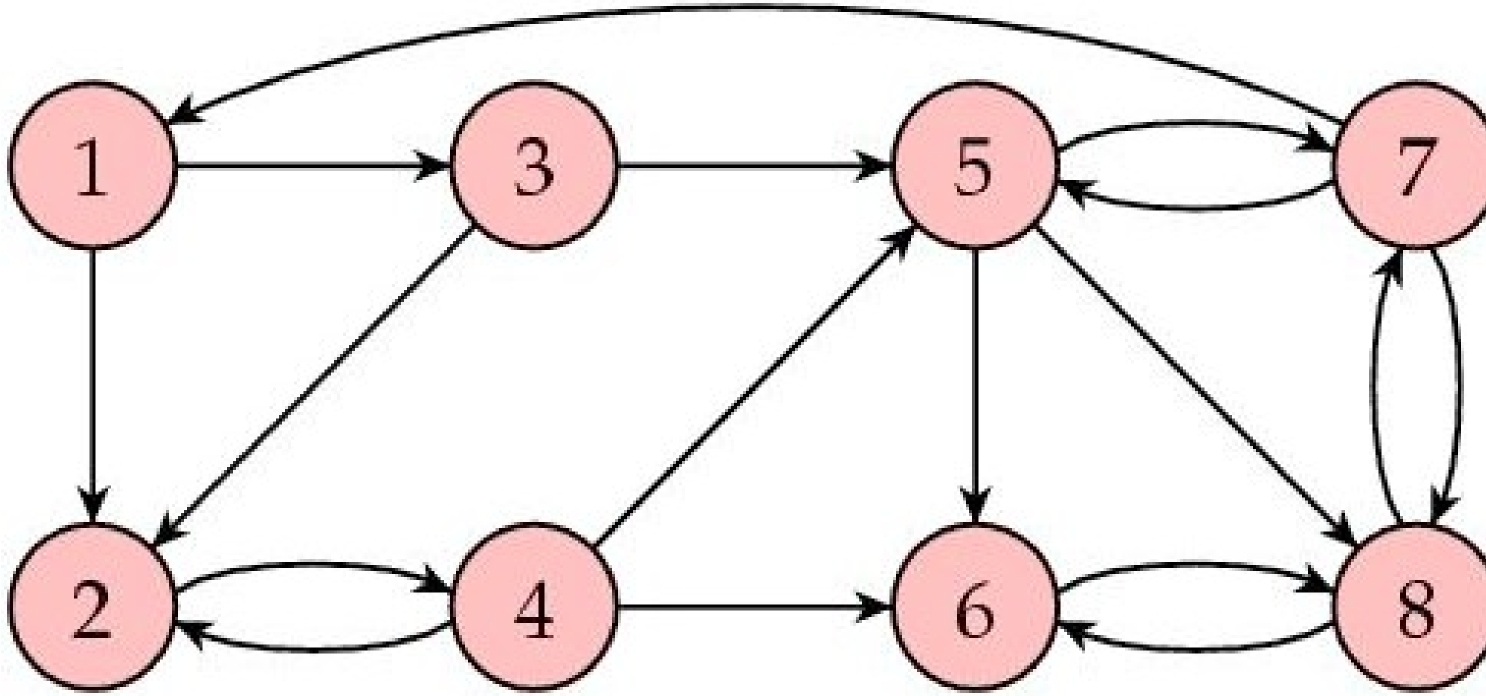
$$I^{k+1} = HI^k$$

How to Compute!? Power method

$$I^{k+1} = HI^k$$



I^0	I^1	I^2	I^3	I^4	...	I^{60}	I^{61}
1	0	0	0	0.0278	...	0.06	0.06
0	0.5	0.25	0.1667	0.0833	...	0.0675	0.0675
0	0.5	0	0	0	...	0.03	0.03
0	0	0.5	0.25	0.1667	...	0.0675	0.0675
0	0	0.25	0.1667	0.1111	...	0.0975	0.0975
0	0	0	0.25	0.1806	...	0.2025	0.2025
0	0	0	0.0833	0.0972	...	0.18	0.18
0	0	0	0.0833	0.3333	...	0.295	0.295



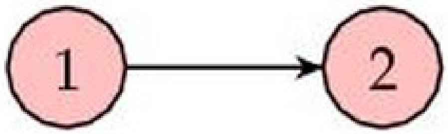
The corresponding matrix is

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 \\ 1/2 & 0 & 1/2 & 1/3 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 1 & 1/3 & 0 \end{bmatrix} \quad \text{with stationary vector} \quad I = \begin{bmatrix} 0.0600 \\ 0.0675 \\ 0.0300 \\ 0.0675 \\ 0.0975 \\ 0.2025 \\ 0.1800 \\ 0.2950 \end{bmatrix}$$

Does it always converge!?

Problem with this solution

- Does the sequence I^k always converge?
- Is the vector to which it converges independent of the initial vector I^0?
- Do the importance rankings contain the information that we want?



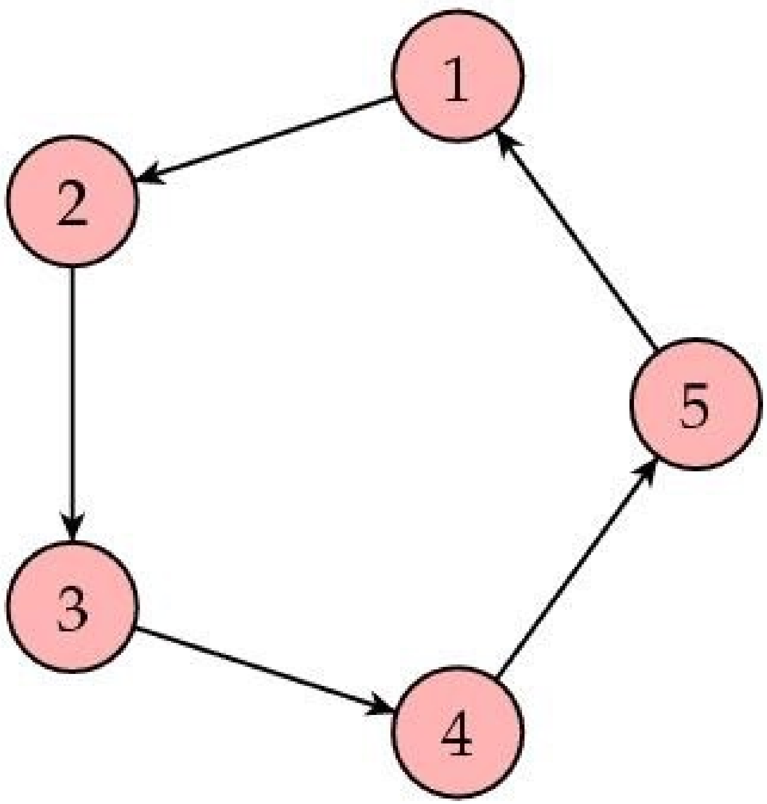
with matrix $$H = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

Here is one way in which our algorithm could proceed:

I^0	I^1	I^2	$I^3 = I$
1	0	0	0
0	1	0	0

WPI

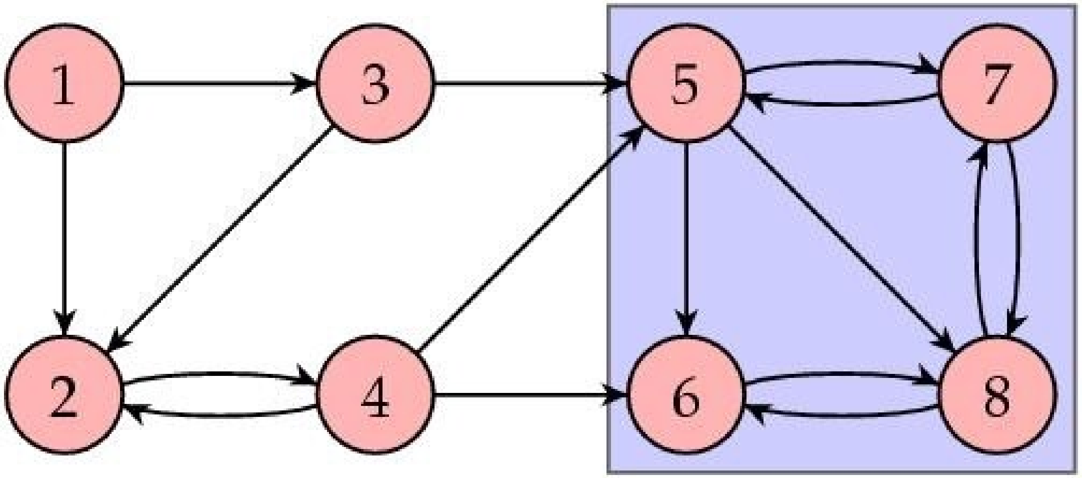
Problem 2



$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

I^0	I^1	I^2	I^3	I^4	I^5
1	0	0	0	0	1
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0

Problem 3: No Outlinks



$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 1/3 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/3 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 1 & 1/2 & 0 \end{bmatrix}$$

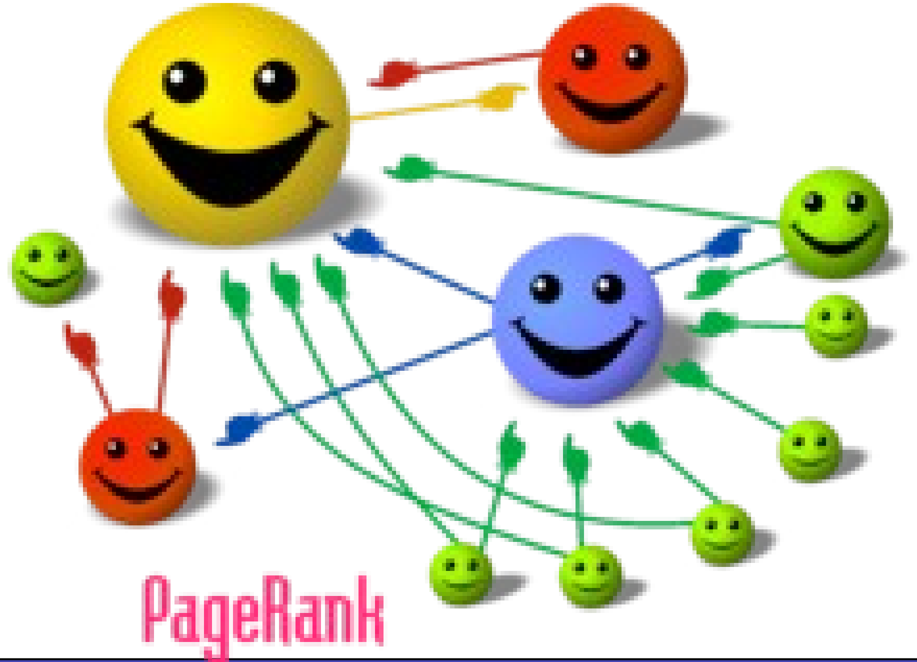
Better Solution

Important

$$G = \alpha H + (1-\alpha)O \qquad O = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

Teleportation!

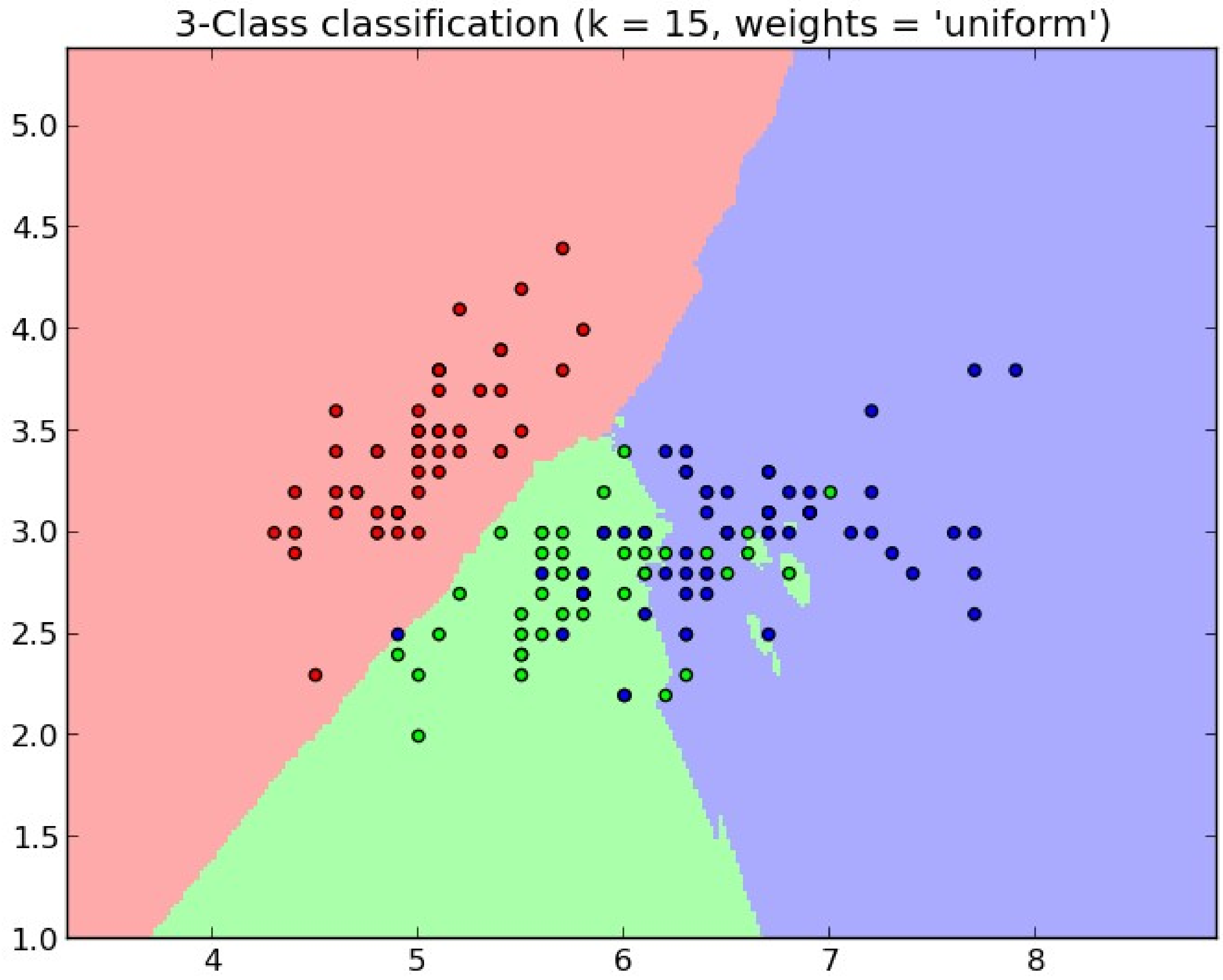
Adding (very unlikely) outlinks for every node



WPI

High Dimensional Data

“Perfect World” for Data Analysis



Real World Data Analysis

First Name	Last Name	Address	City	Age
Mickey	Mouse	123 Fantasy Way	Anaheim	73
Bat	Man	321 Cavern Ave	Gotham	54
Wonder	Woman	987 Truth Way	Paradise	39
Donald	Duck	555 Quack Street	Mallard	65
Bugs	Bunny	567 Carrot Street	Rascal	58
Wiley	Coyote	999 Acme Way	Canyon	61
Cat	Woman	234 Purrfect Street	Hairball	32
Tweety	Bird	543	Itotltaw	28

High Dimension?

Important

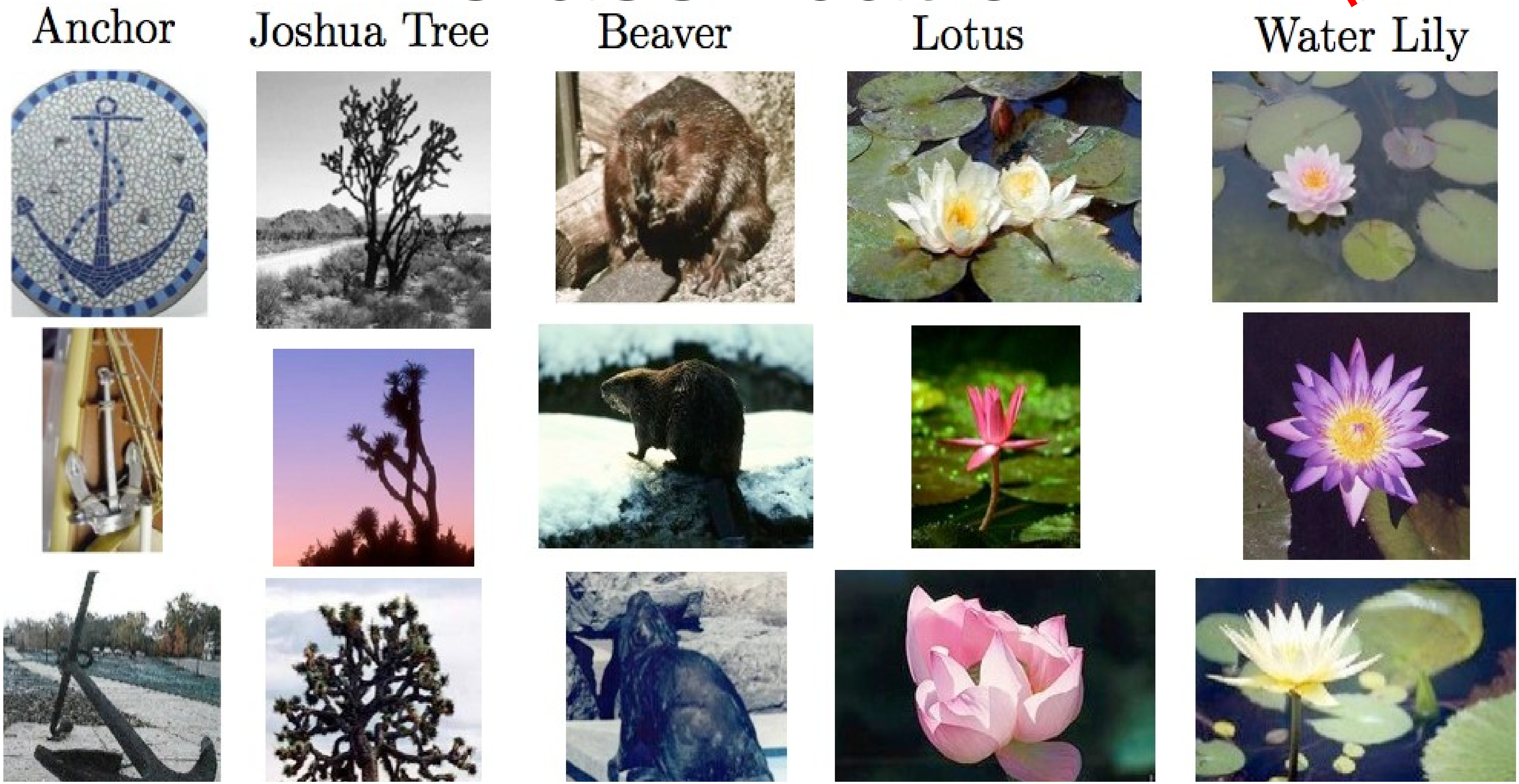
P

N

First Name	Last Name	Address	City	Age
Mickey	Mouse	123 Fantasy Way	Anaheim	73
Bat	Man	321 Cavern Ave	Gotham	54
Wonder	Woman	987 Truth Way	Paradise	39
Donald	Duck	555 Quack Street	Mallard	65
Bugs	Bunny	567 Carrot Street	Rascal	58
Wiley	Coyote	999 Acme Way	Canyon	61
Cat	Woman	234 Purrfect Street	Hairball	32
Tweety	Bird	543	Itotltaw	28

High Dimensional Data Classification

Important

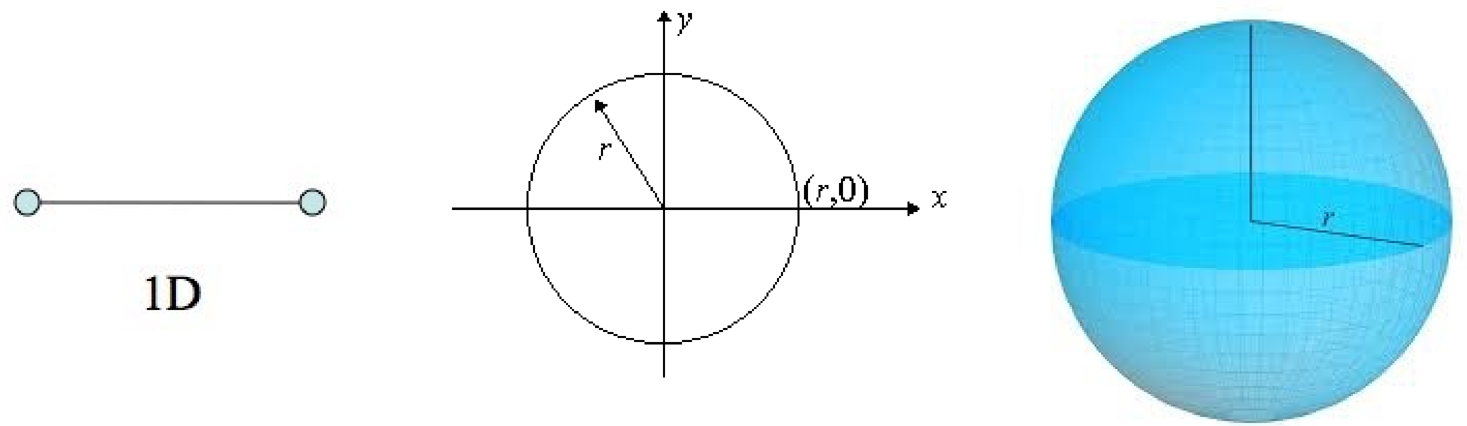


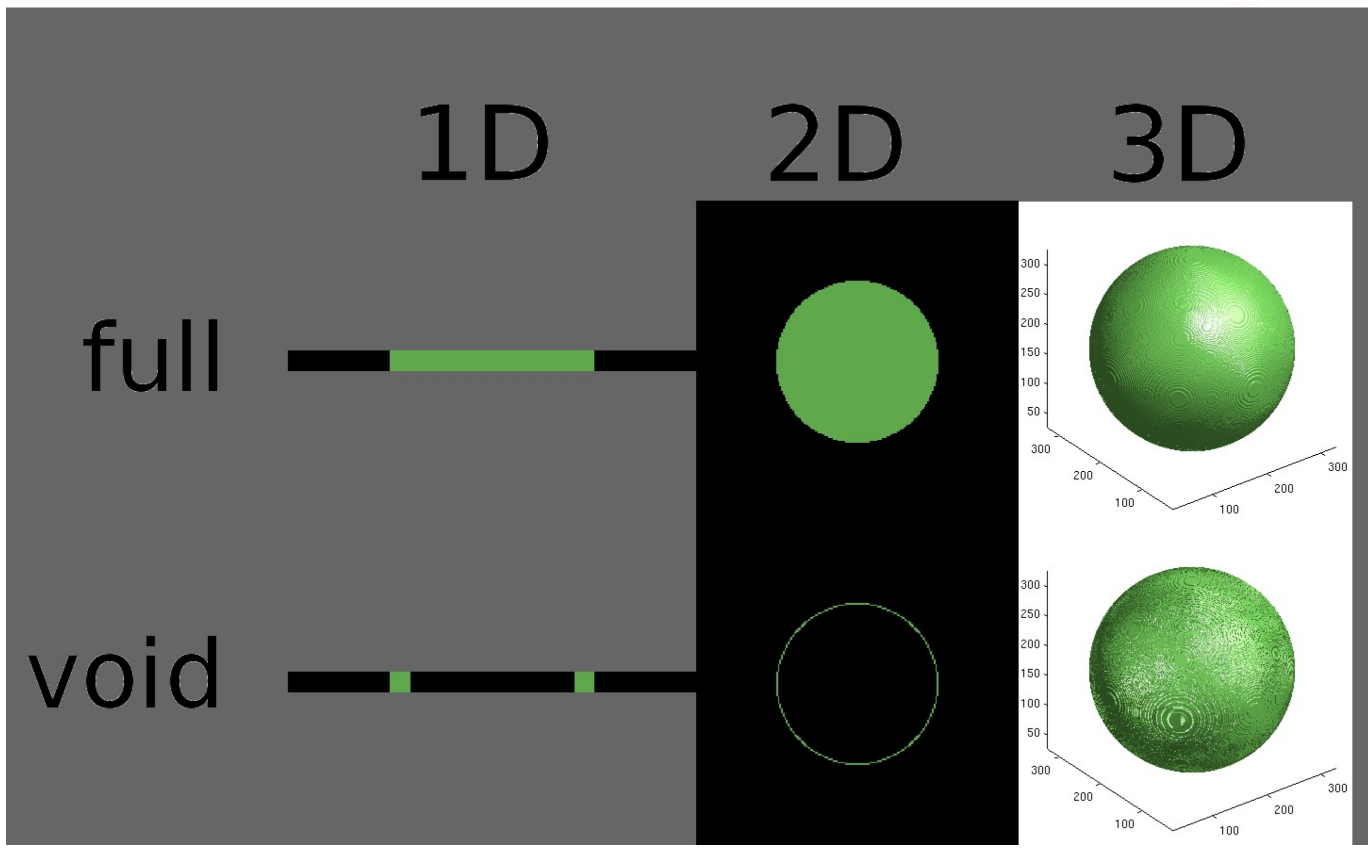
Outline

- What are spheres and cubes in high dimensions?
- What does distance mean in high dimensions?
- What does random mean in high dimensions?
- What does data mean in high dimensions?

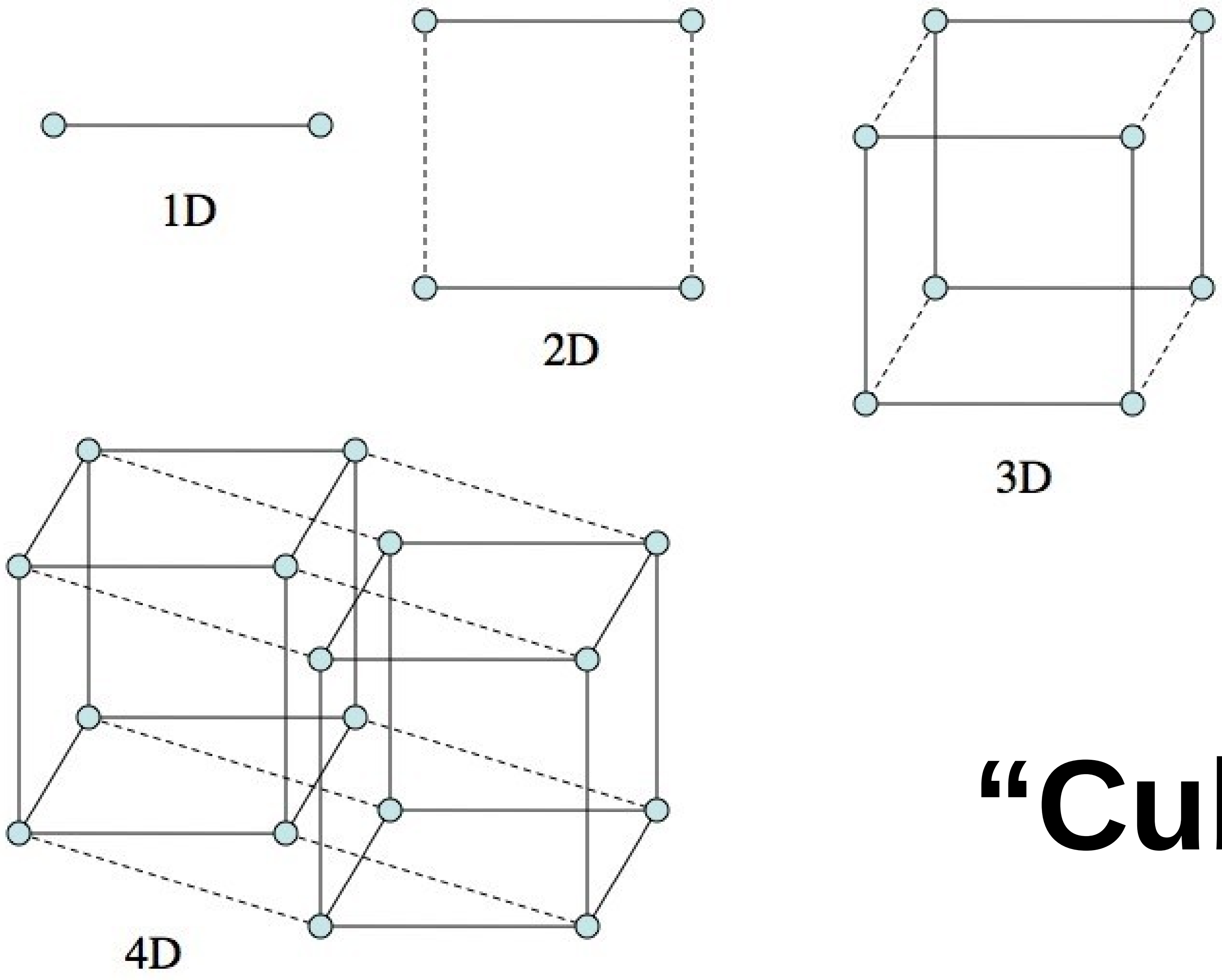
Who is this guy?

Sphere





Line - Square - Cube - HyperCube

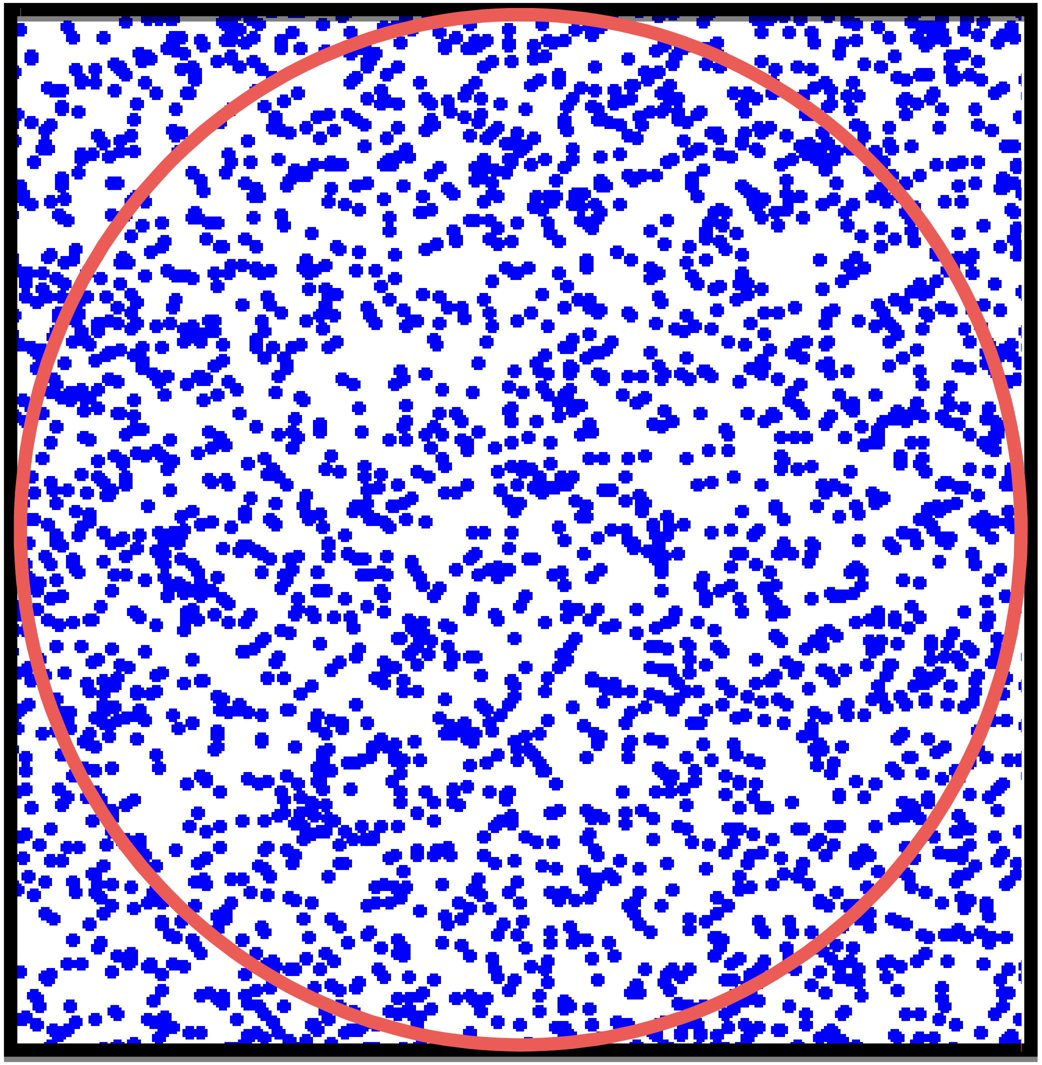


“Cube”

How do spheres and cubes relate in high-dimensions?

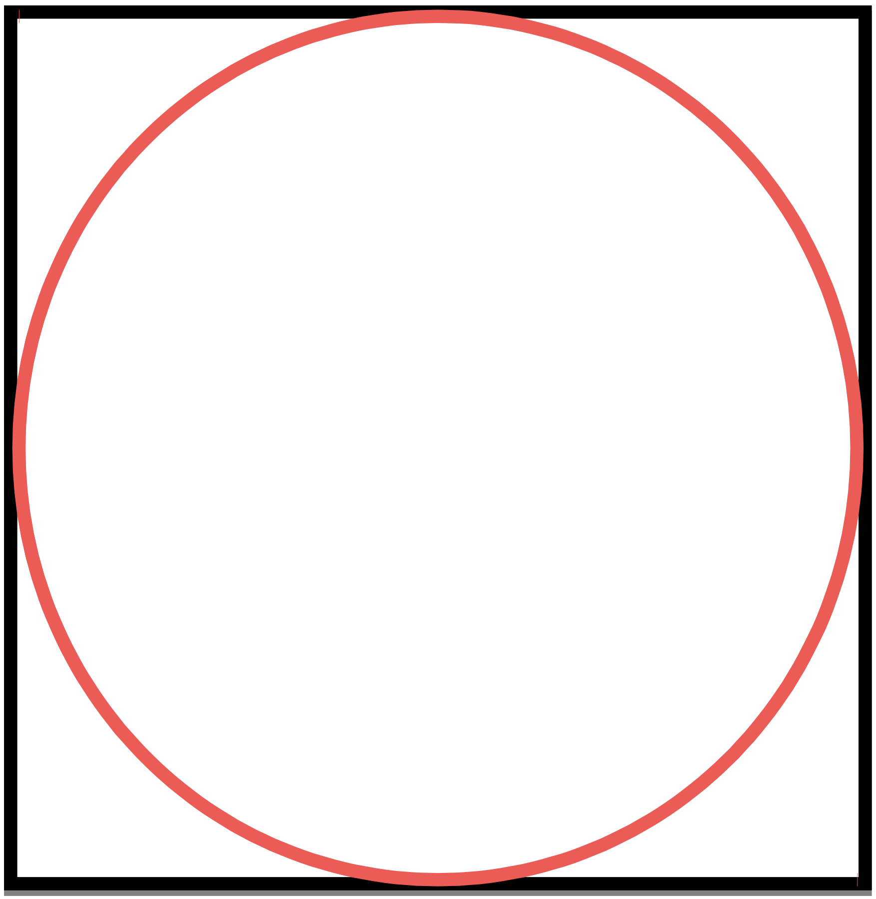
Question #1:

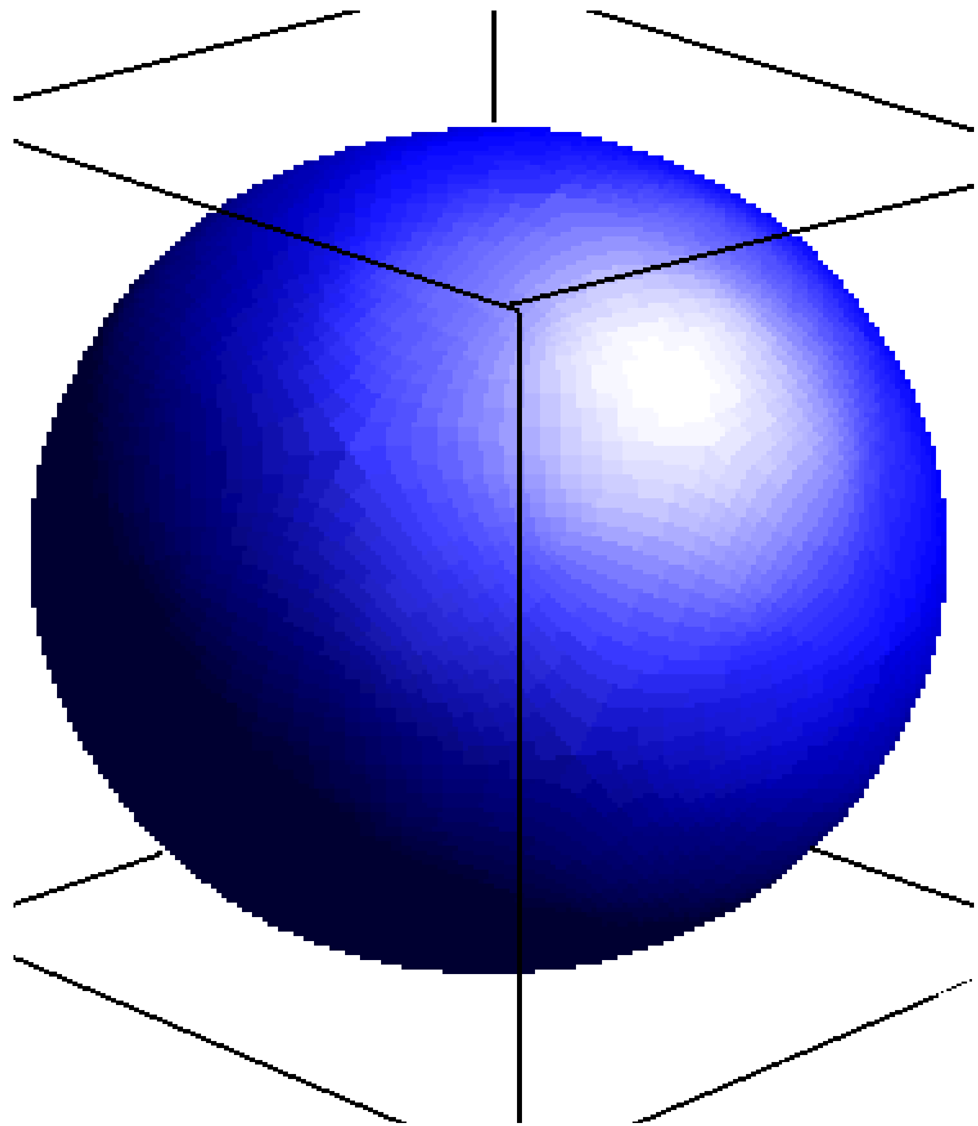
Uniform Distribution (Center vs Corner)



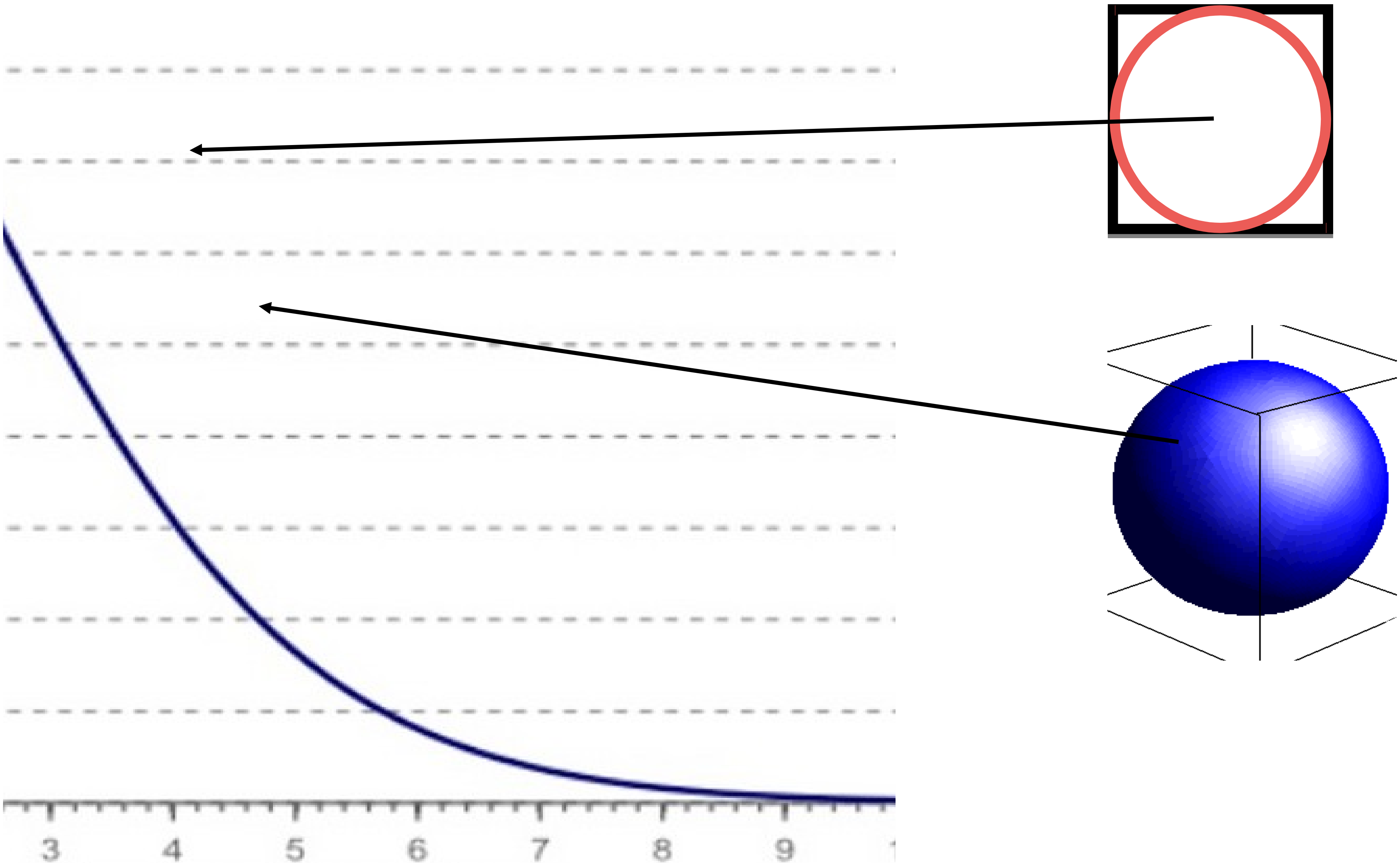
Center vs Corner

Important



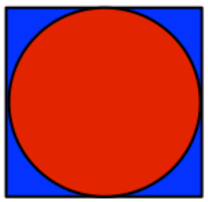


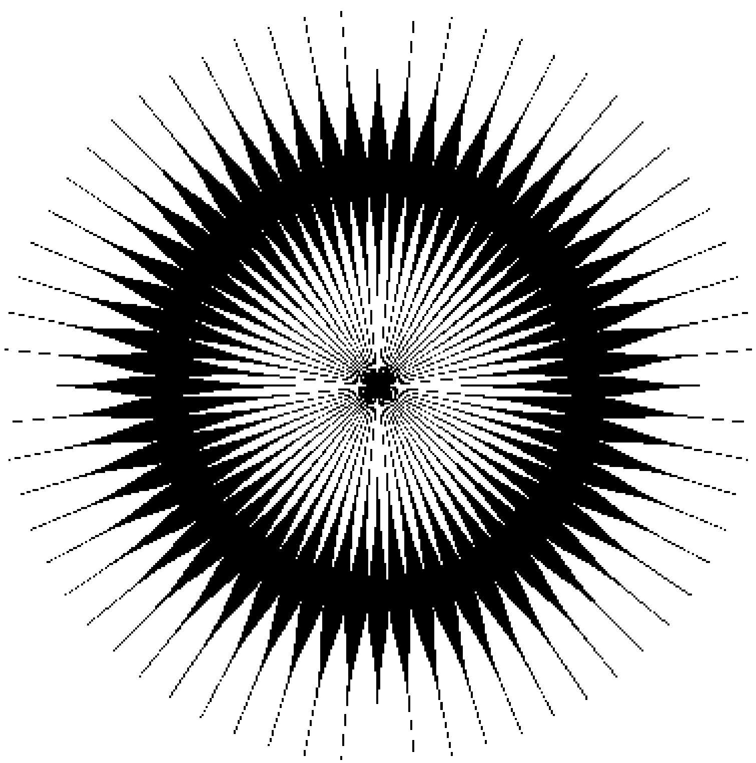
Volume of the hypersphere



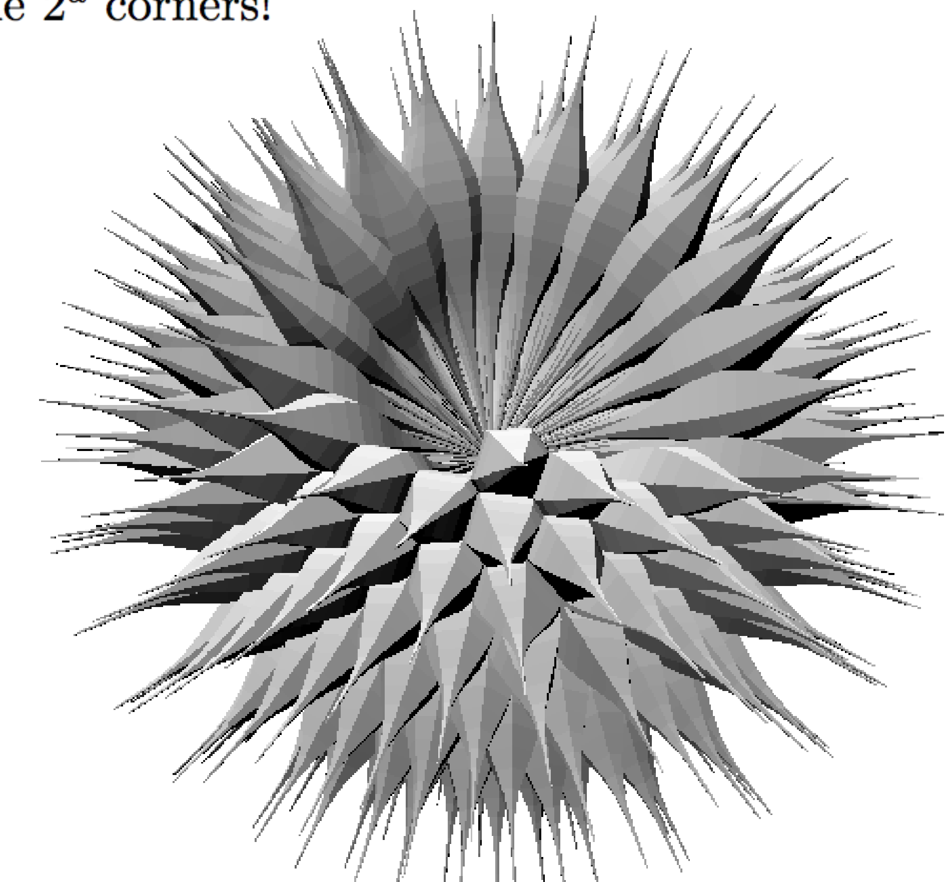
High Dimensional Cube

$$\lim_{d\to\infty} \frac{\text{volume sphere of radius r}}{\text{volume } [0,r]^d} = 0$$

 : nearly all points are in the 2^d corners!



6 dimensional cube projection



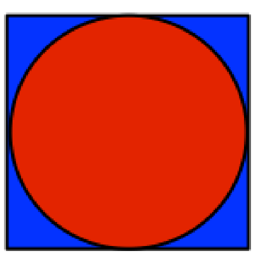
8 dimensional cube projection

In High Dimensional Space:
Rule # 1

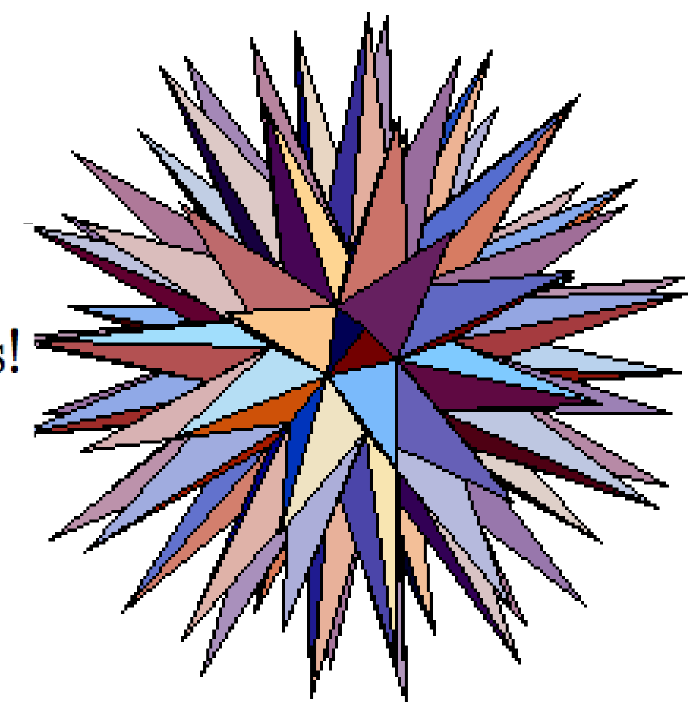
Important

- **Center v.s. Corners :**

    - **Corners win!**

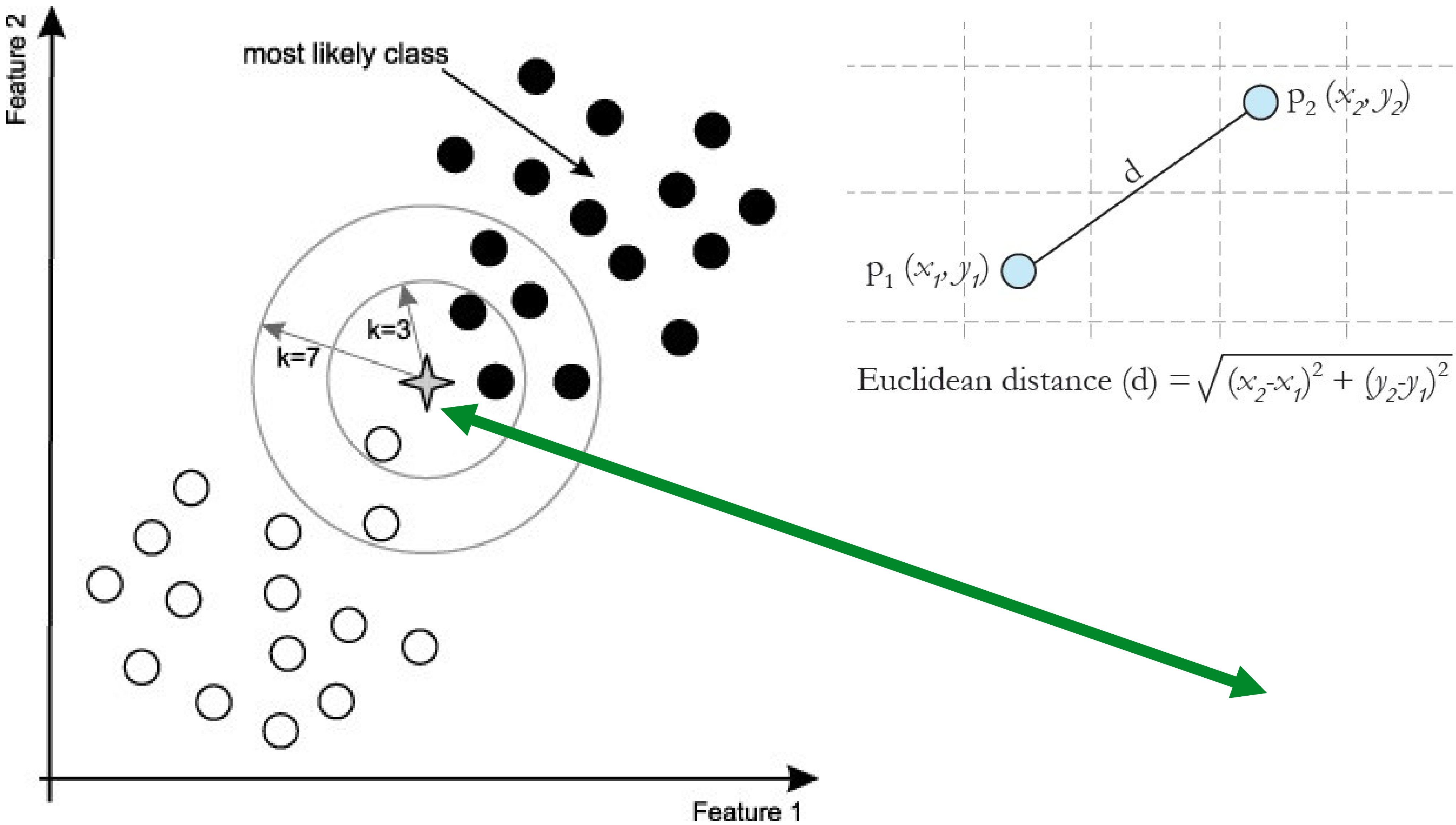


: nearly all points are in the 2^d corners!



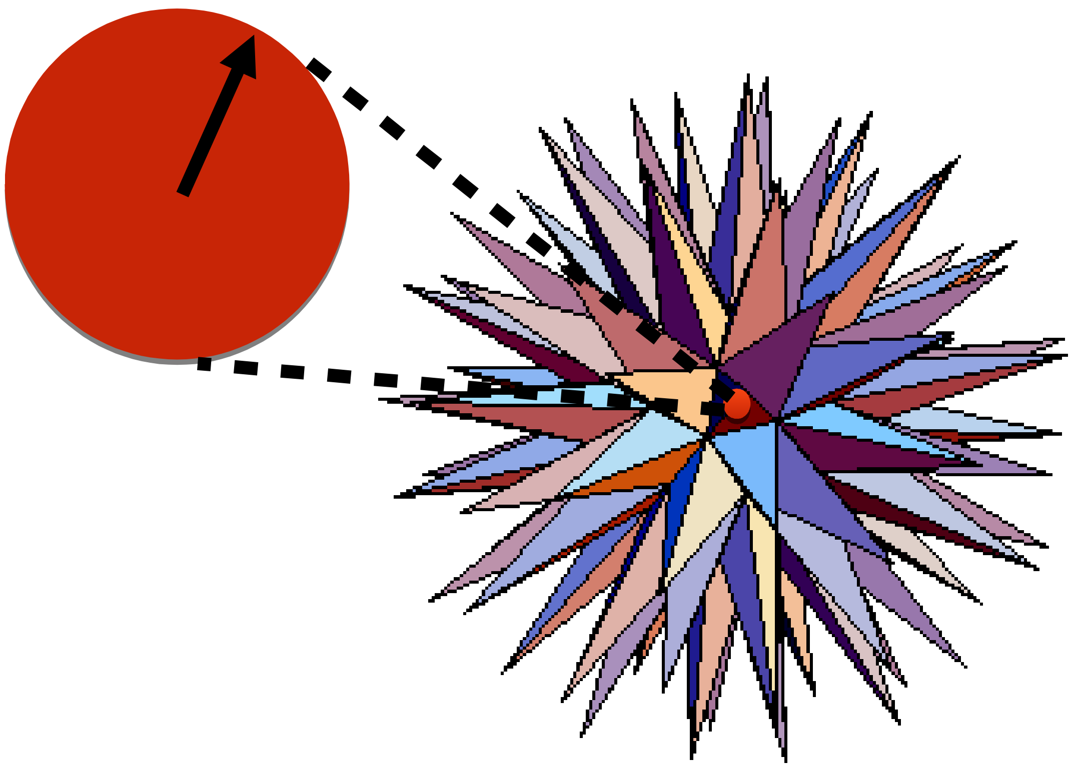
What does distance mean in high-dimensions?

Question #2: Euclidean Distance



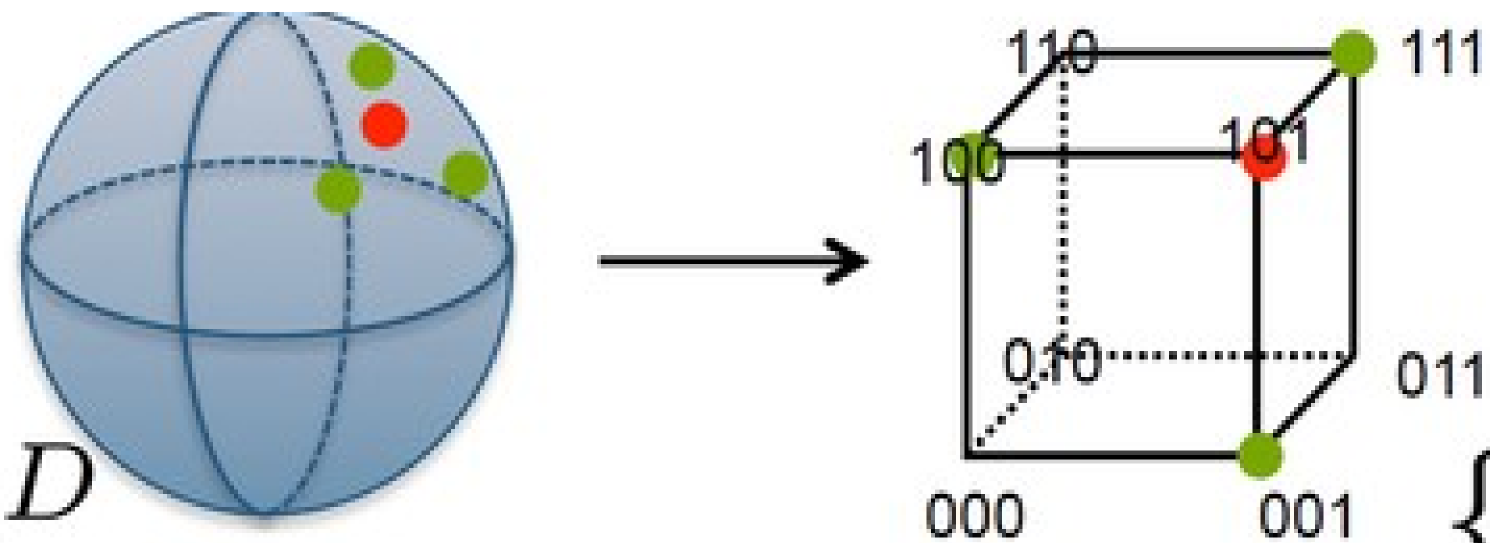
In High Dimensional Space: **Rule # 2**

- **Don't Use Euclidean Distance**



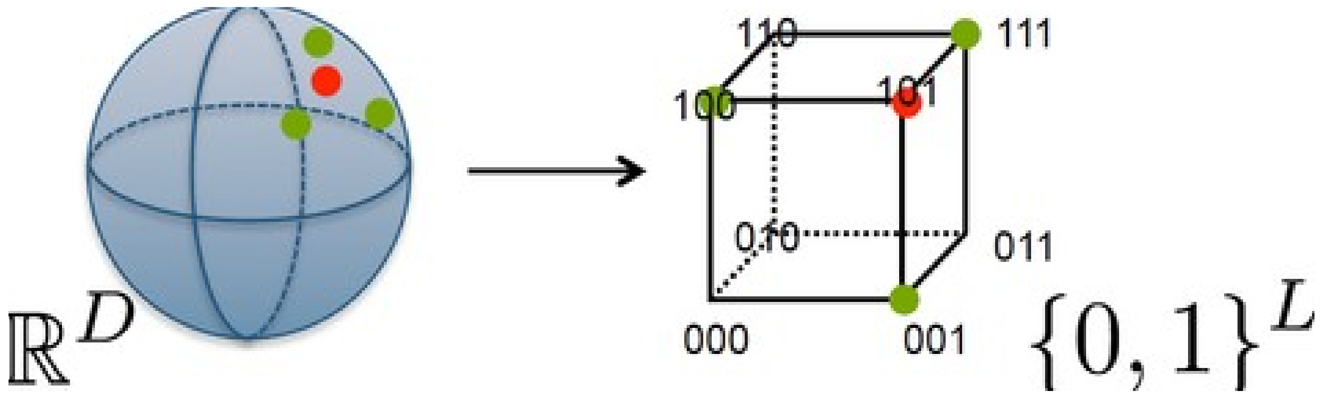
all your "neighbors" are far away

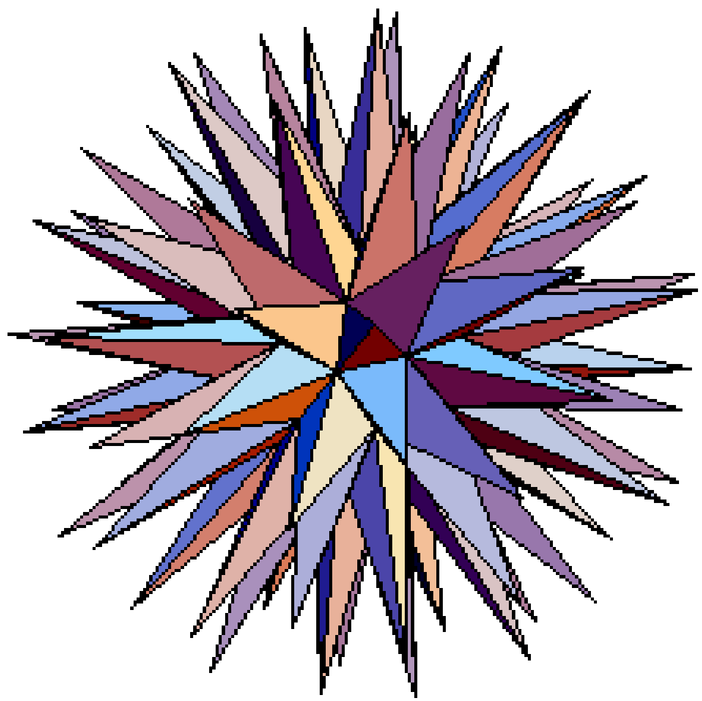
All “neighbors” are “equally” far away



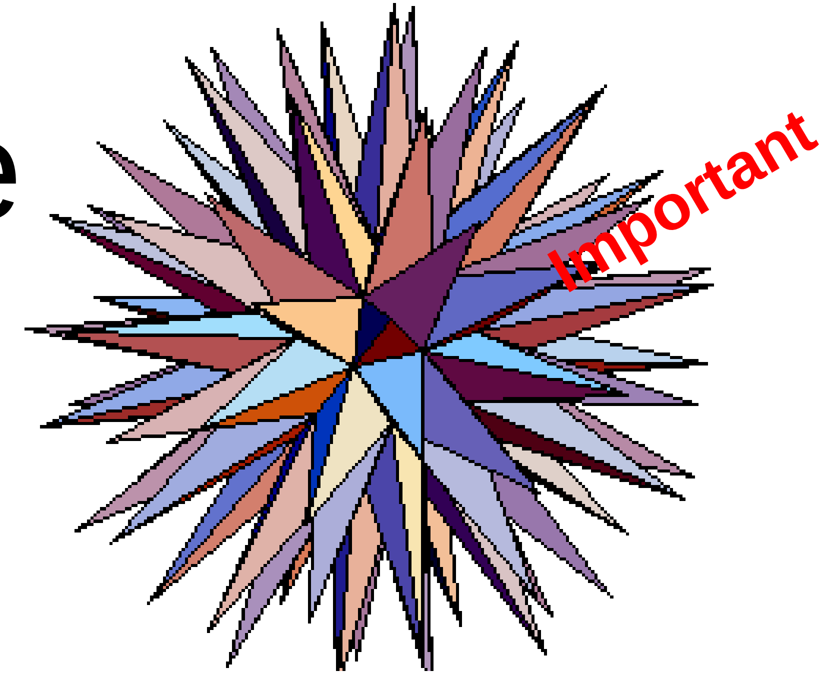
$\mathbb{R}^D$ $\{0,1\}^L$

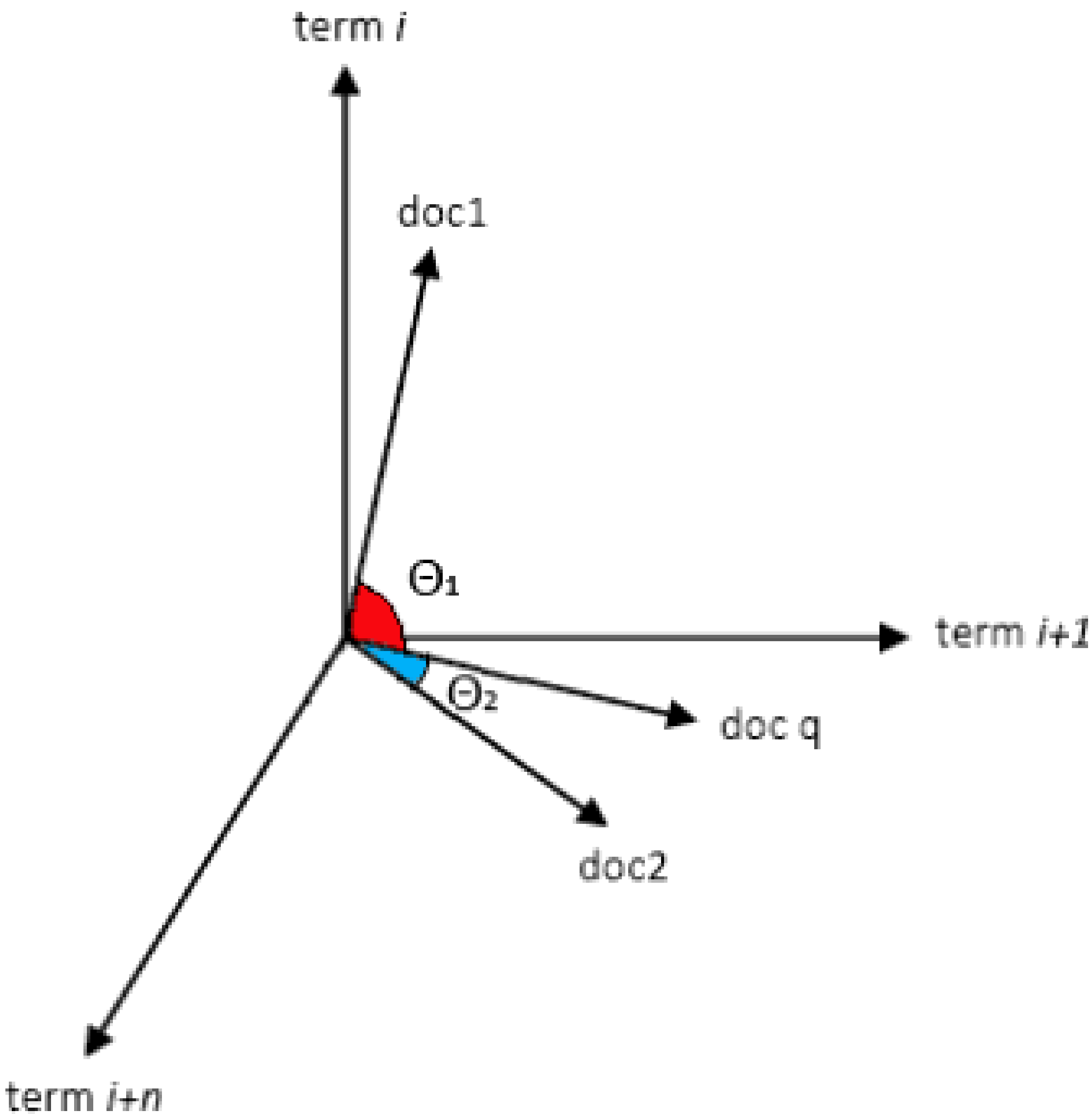
All “neighbors” are many very far away

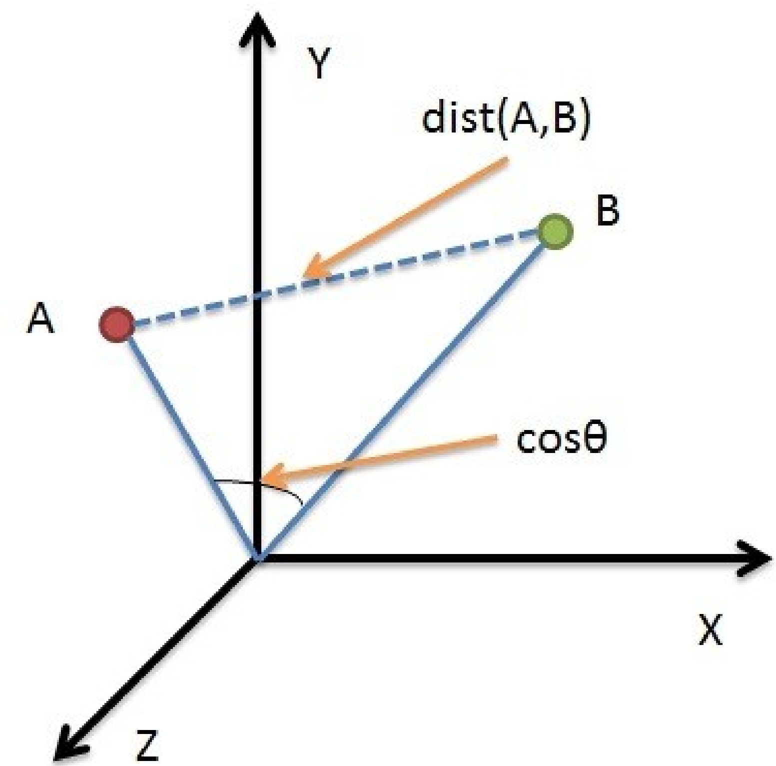




Cosine Distance



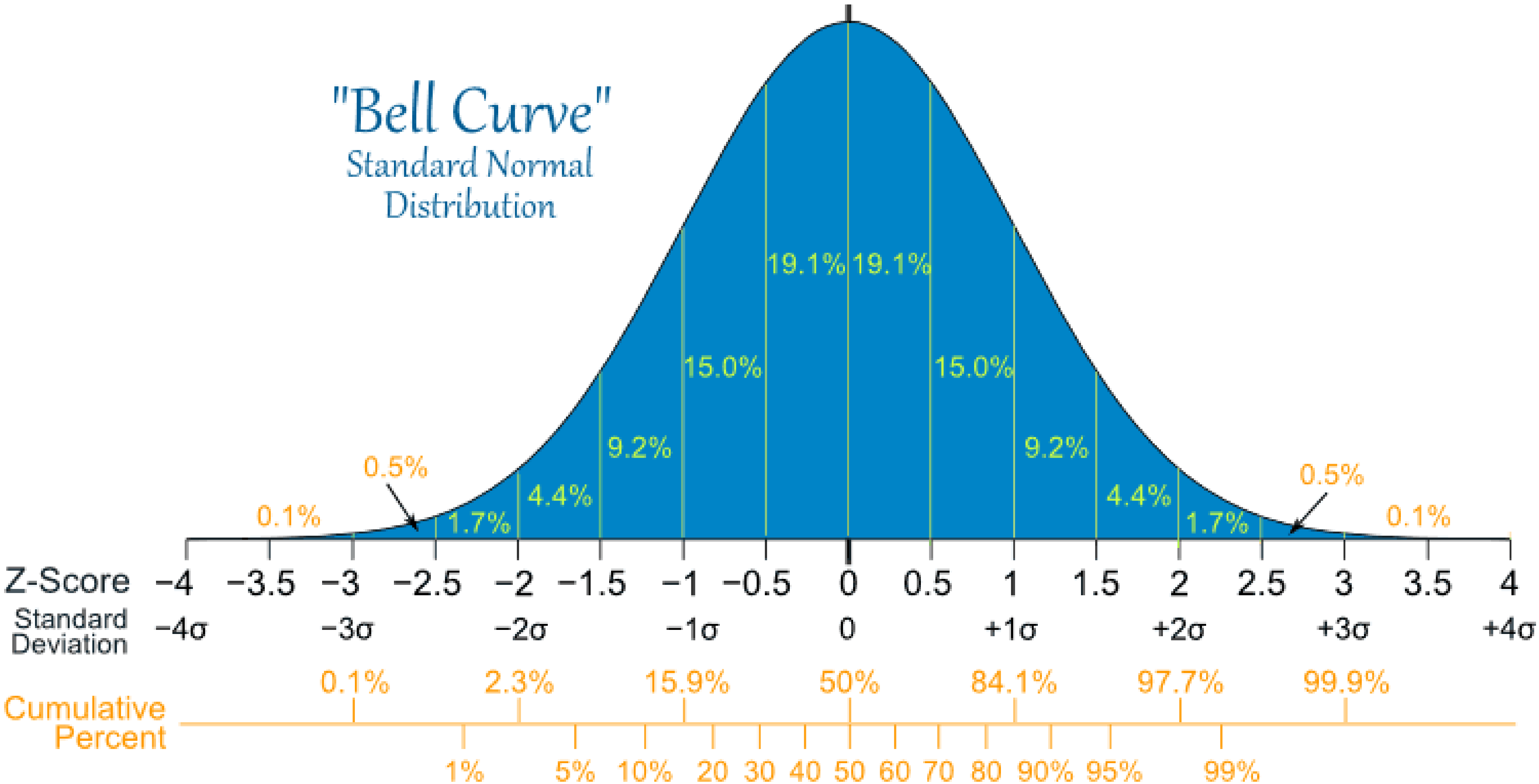


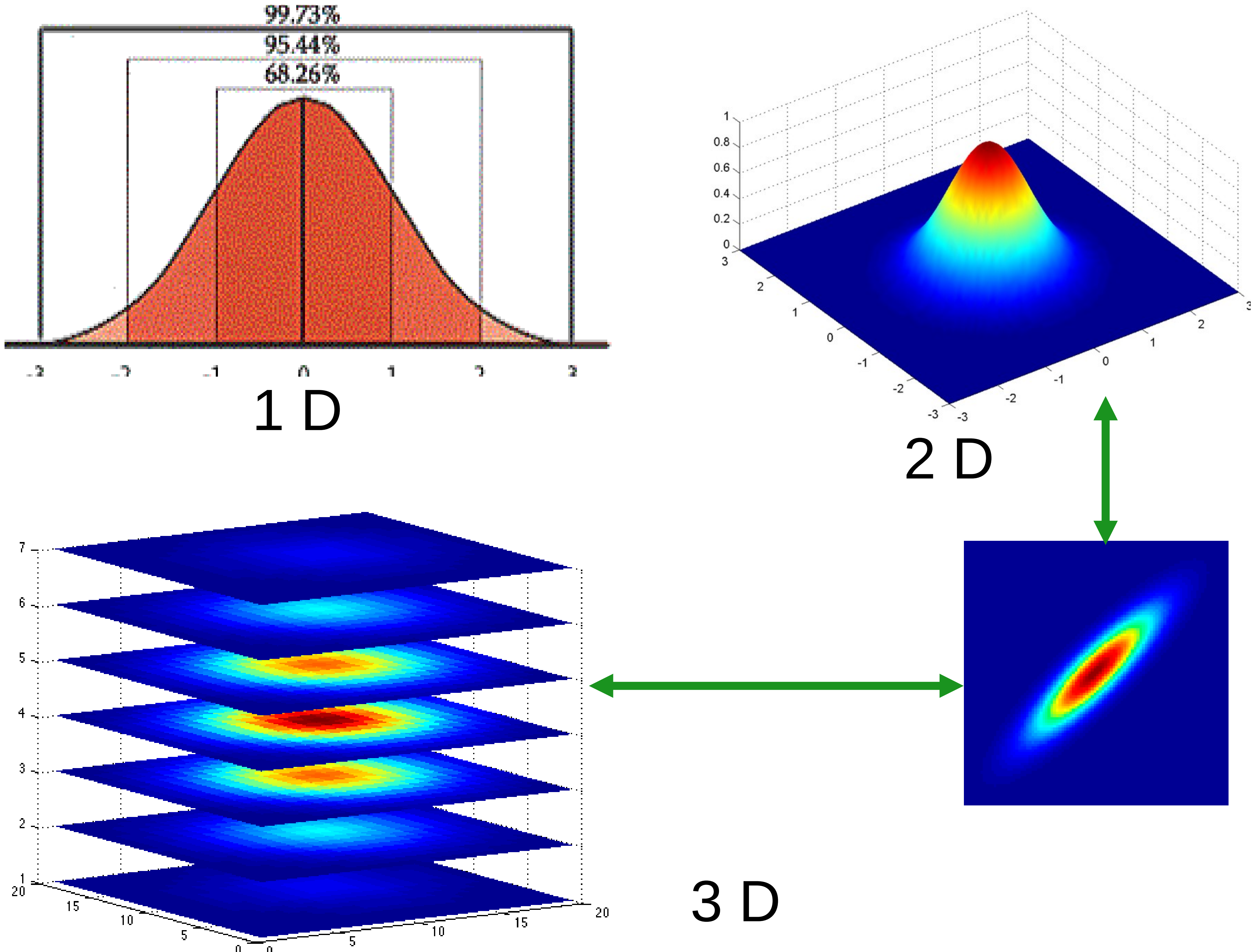


What does random mean in high-dimensions?

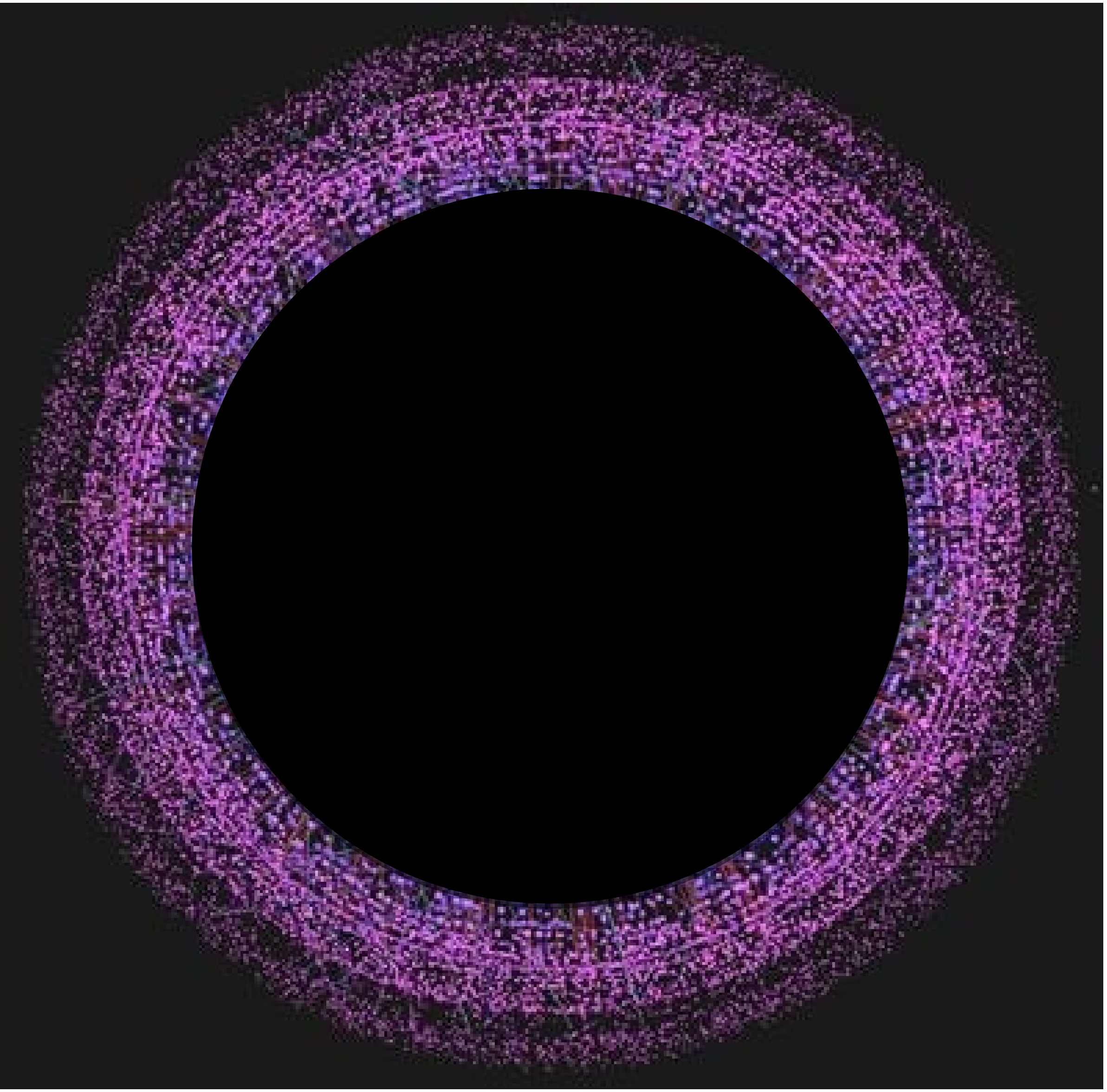
Question #3:

Gaussian Distribution (Center v.s. Surface)

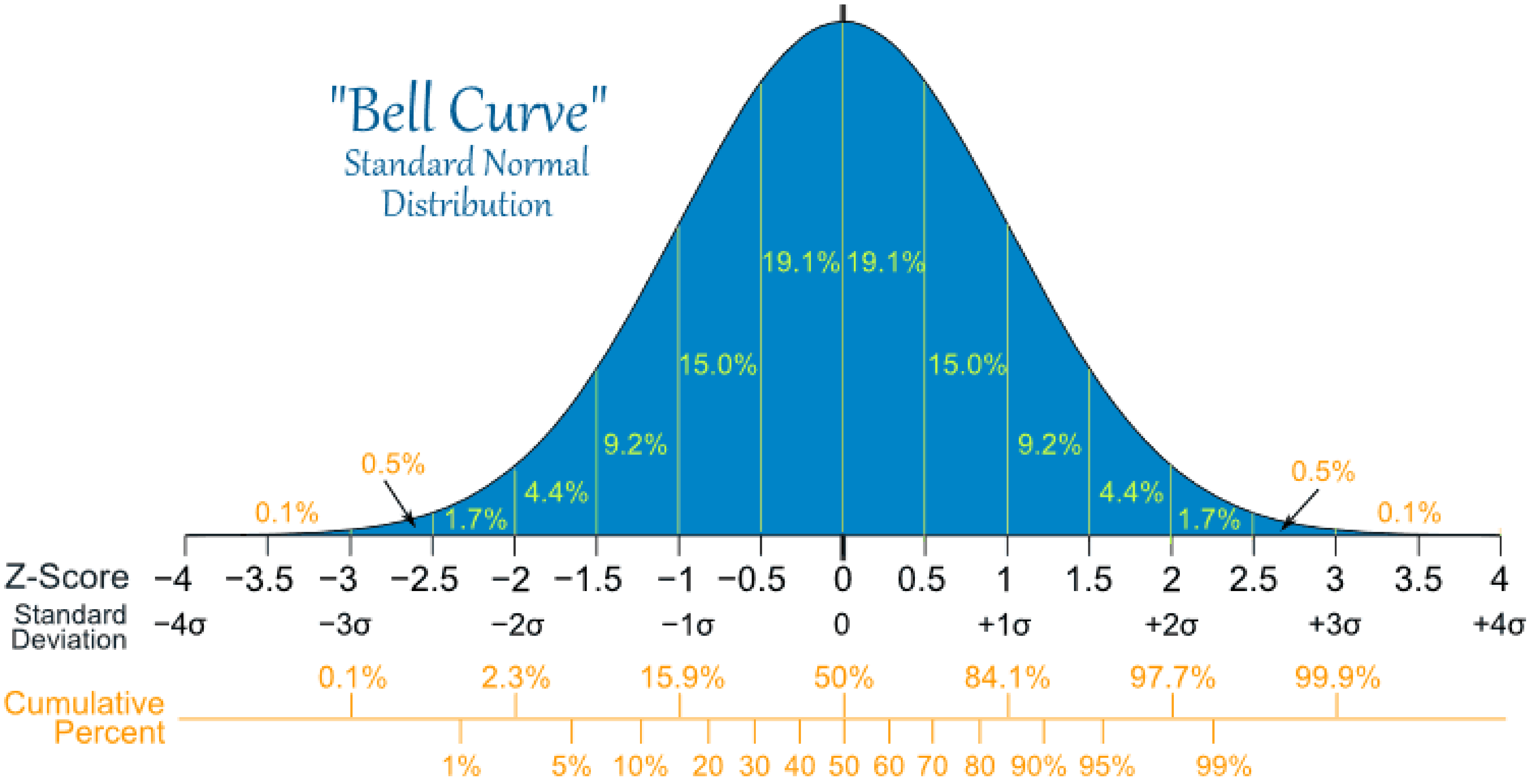



99.73%
95.44%
68.26%
1 D
2 D
3 D

High Dimensional Gaussian Distribution



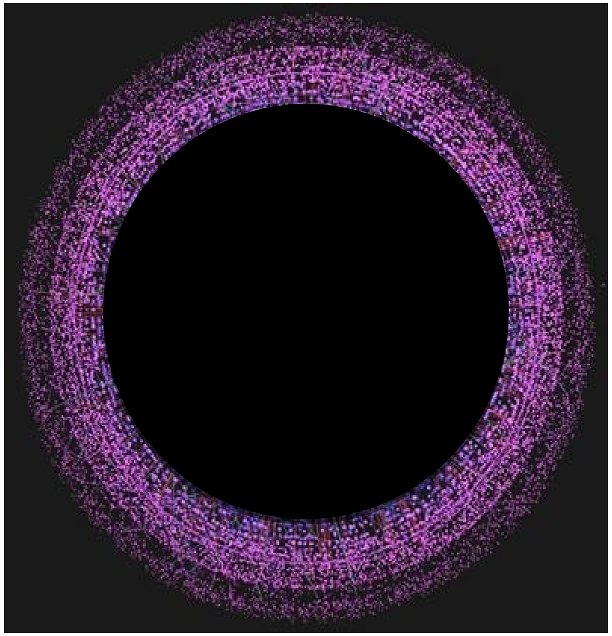
Question #3:
Gaussian Distribution (Center v.s. Surface)



In High Dimensional Space: **Rule # 3**

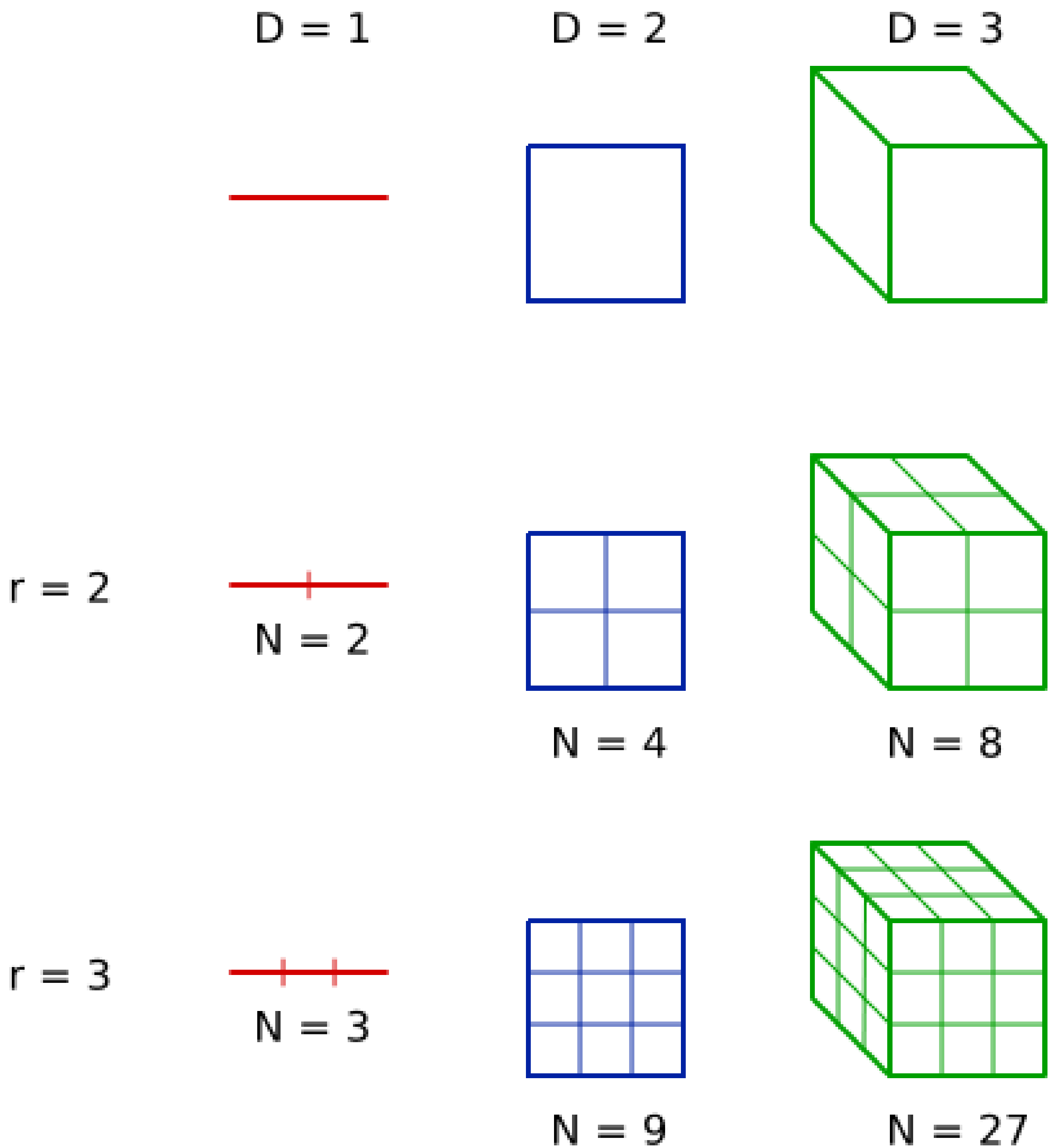
Important

- **All data points are on the surface**
  - By chance, data will never reach the inner space of the ball.



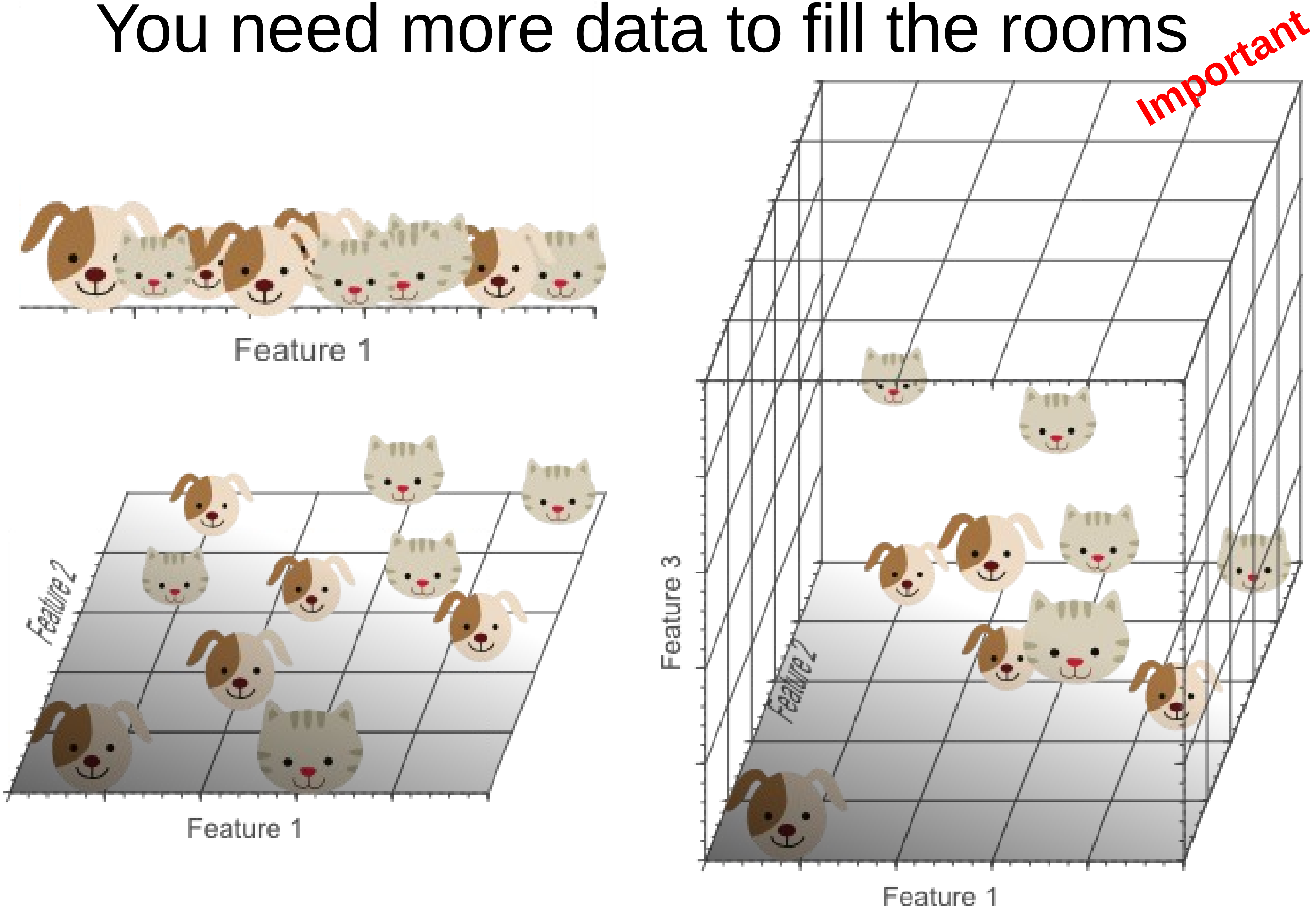
What does data mean in high-dimensions?

Question # 4: do we have enough data



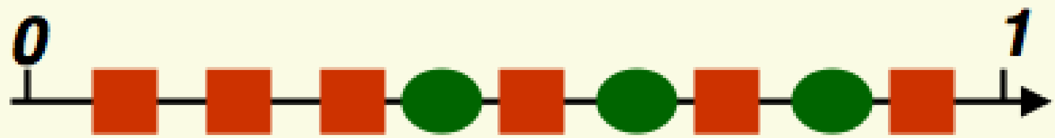
Number
of
“Rooms”

You need more data to fill the rooms



Data become sparser

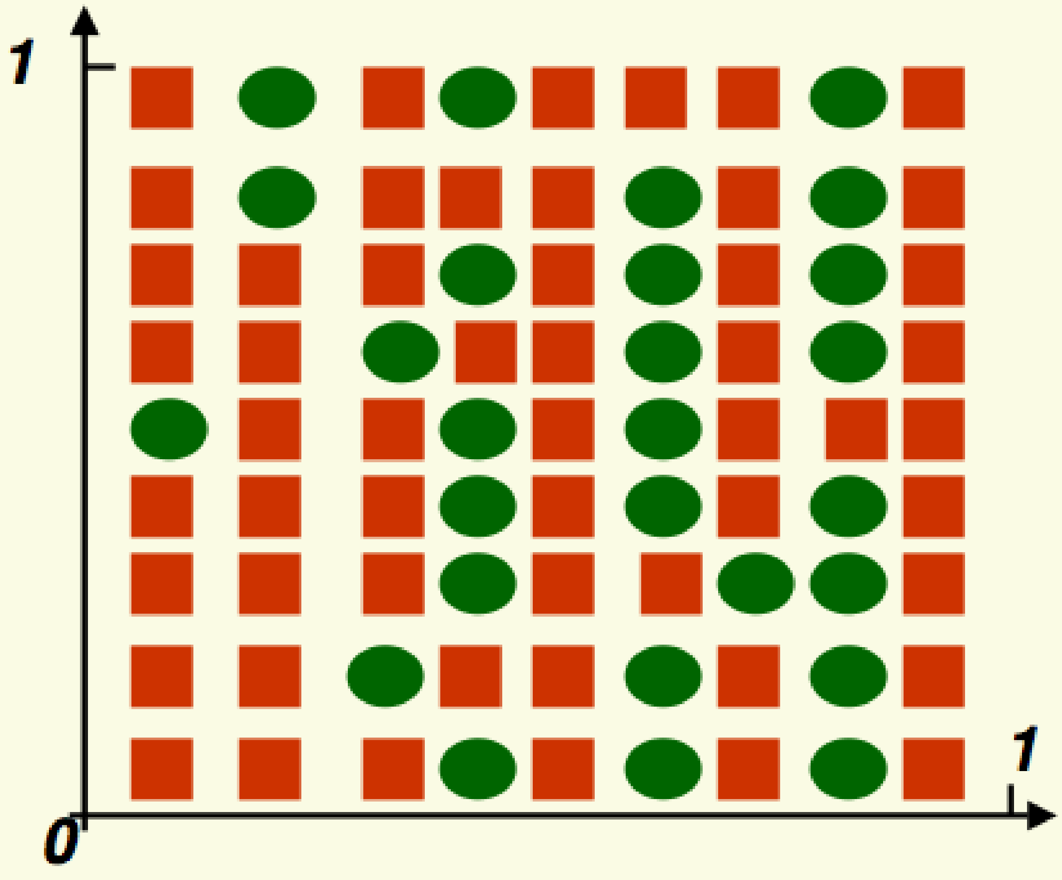
- Suppose we want to use the nearest neighbor approach with ***k*** = 1 (***1NN***)
- Suppose we start with only one feature



- This feature is not discriminative, i.e. it does not separate the classes well
- We decide to use 2 features. For the 1NN method to work well, need a lot of samples, i.e. samples have to be dense
- To maintain the same density as in 1D (9 samples per unit length), how many samples do we need?

Data become Sparser: Number of Samples

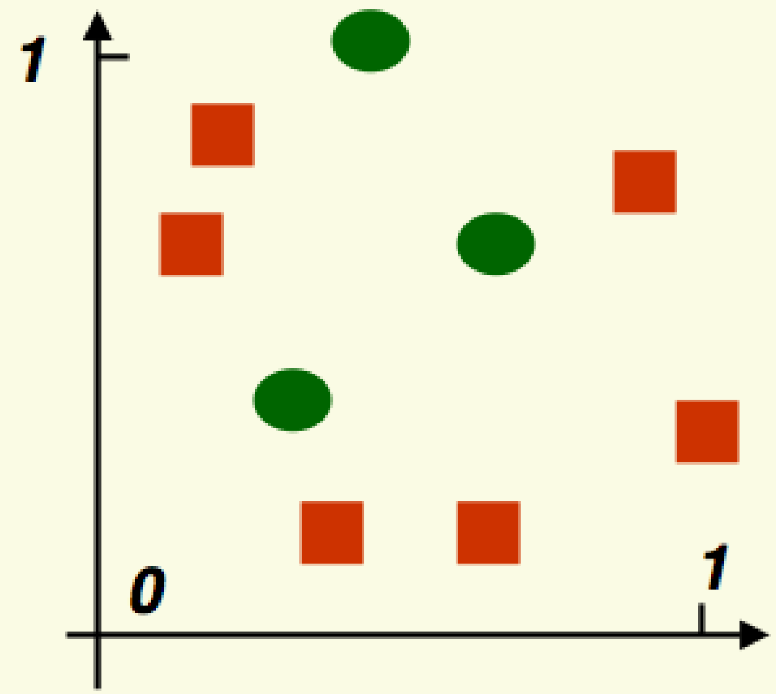
- We need 9^2 samples to maintain the same density as in *1D*



Curse of Dimensionality: Number of Samples

Important

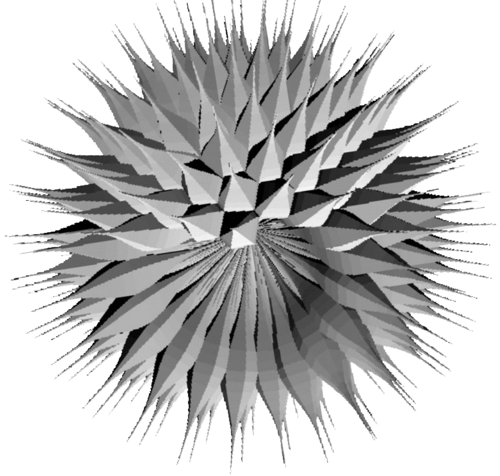
- Of course, when we go from 1 feature to 2, no one gives us more samples, we still have 9

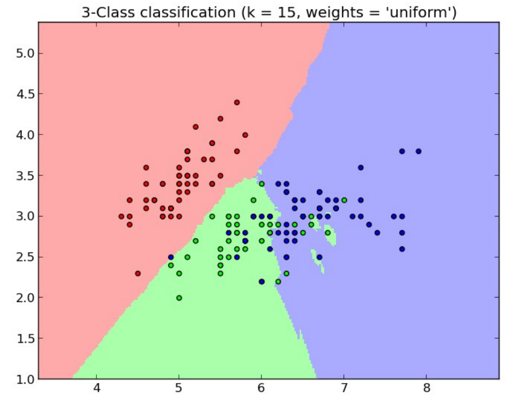


- This is way too sparse for ***1NN*** to work well

Now you know what is really happening

in High Dimensional Space

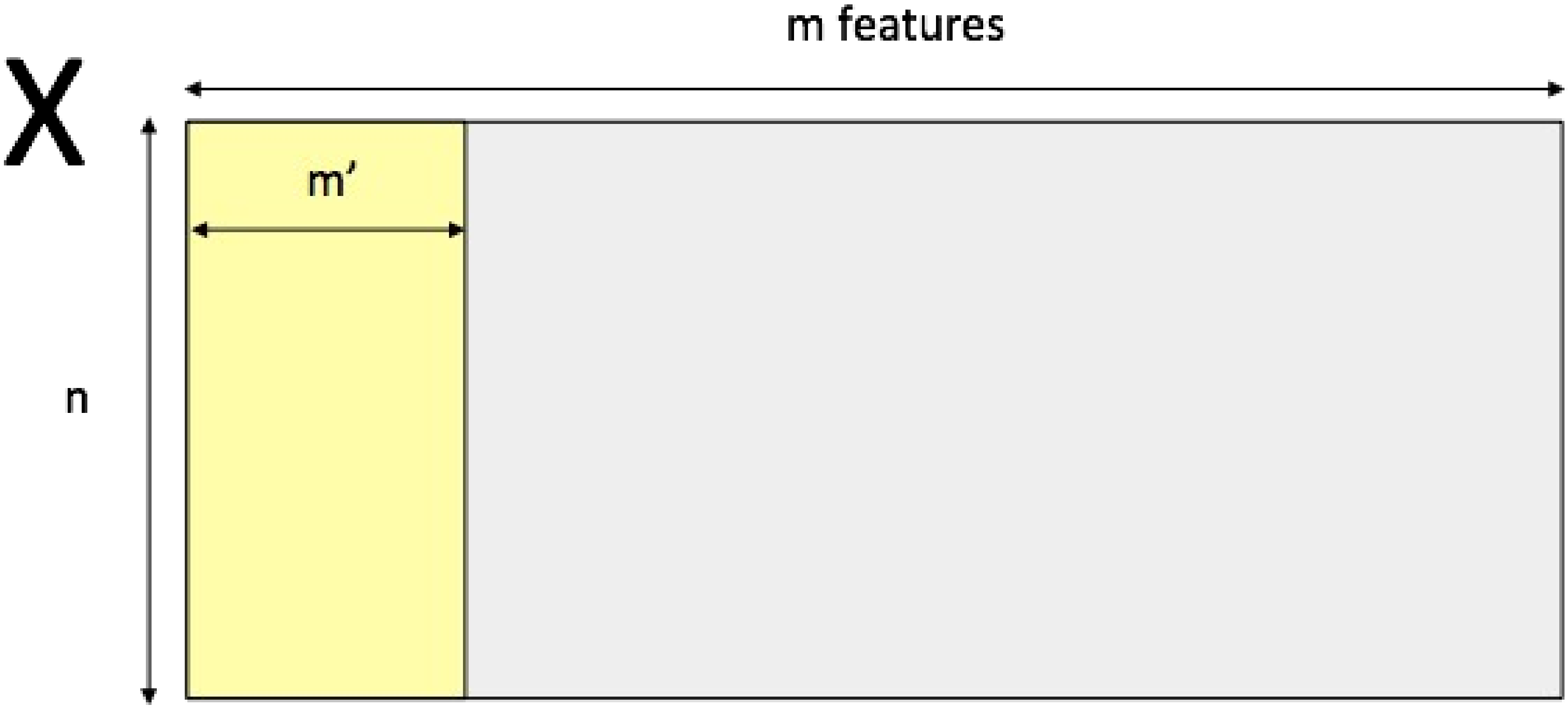




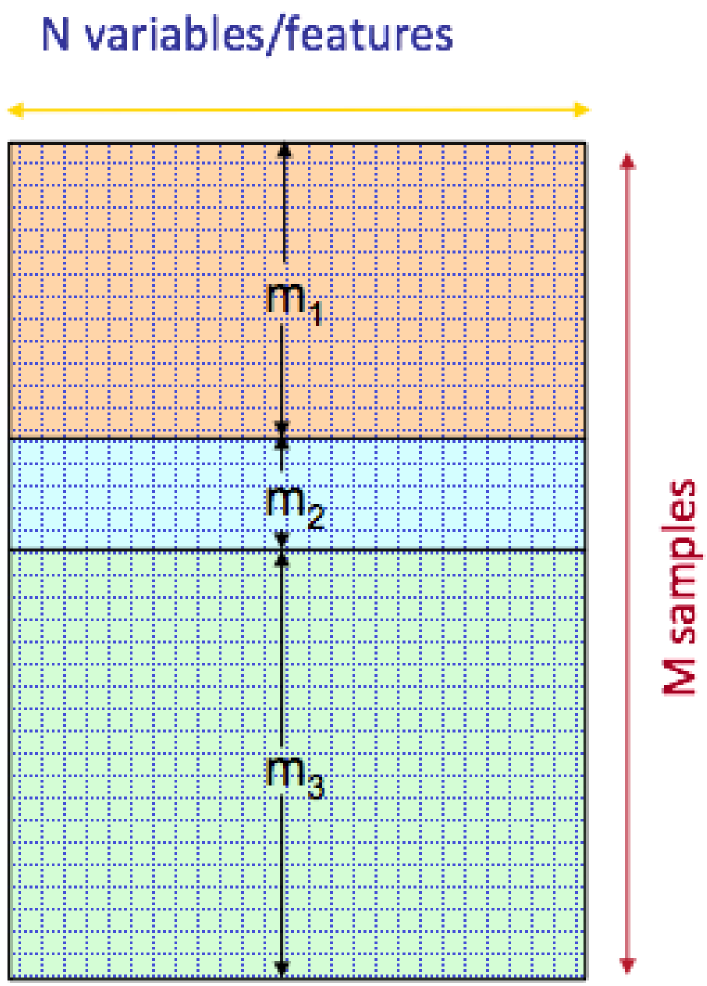
- Feature Selection
  - Filtering approach
  - Wrapper approach
  - Embedded methods

- Dimensionality Reduction
  - Principal Components Analysis (PCA)
  - Nonlinear PCA (Kernel PCA, CatPCA)
  - Multi-Dimensional Scaling (MDS)
  - Homogeneity Analysis

Feature Selection



Wrapper methods



Split data into 3 sets:

training, validation, and test set.

1) For each feature subset, train predictor on training data.
2) Select the feature subset, which performs best on validation data.
   - Repeat and average if you want to reduce variance (cross-validation).
3) Test on test data.

Danger of over-fitting with intensive search!

Dimensionality Reduction

Important

- High dimensionality is challenging and redundant
- It is natural to try to reduce dimensionality
- Reduce dimensionality by feature combination: combine old features ***x*** to create new features ***y***

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \rightarrow f\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}\right) = \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} = \mathbf{y} \quad \text{with } k < d$$

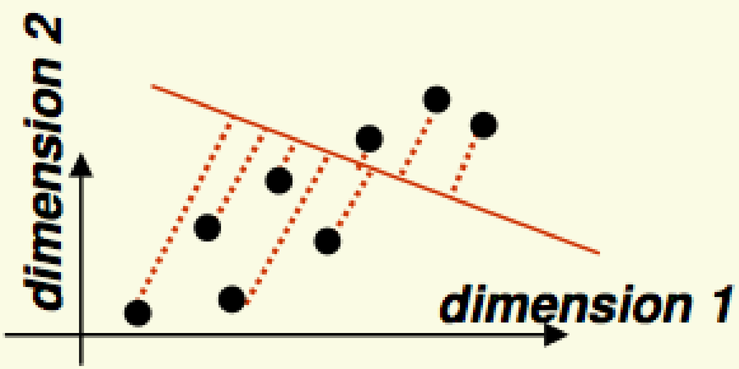
- For example,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 + x_2 \\ x_3 + x_4 \end{bmatrix} = \mathbf{y}$$

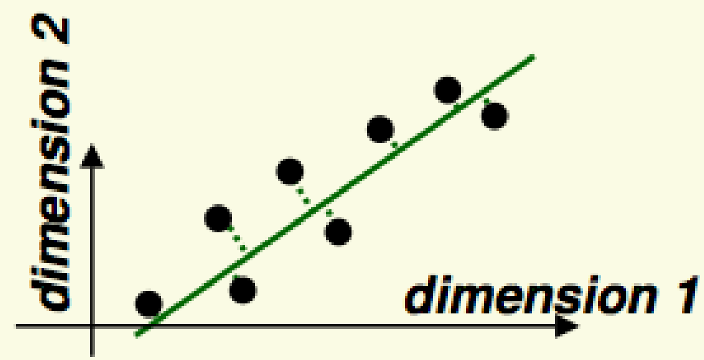
- Ideally, the new vector ***y*** should retain from ***x*** all information important for classification

PCA:Principle Component Analysis

- Main idea: seek most accurate data representation in a lower dimensional space
- Example in 2-D
  - Project data to 1-D subspace (a line) which minimize the projection error



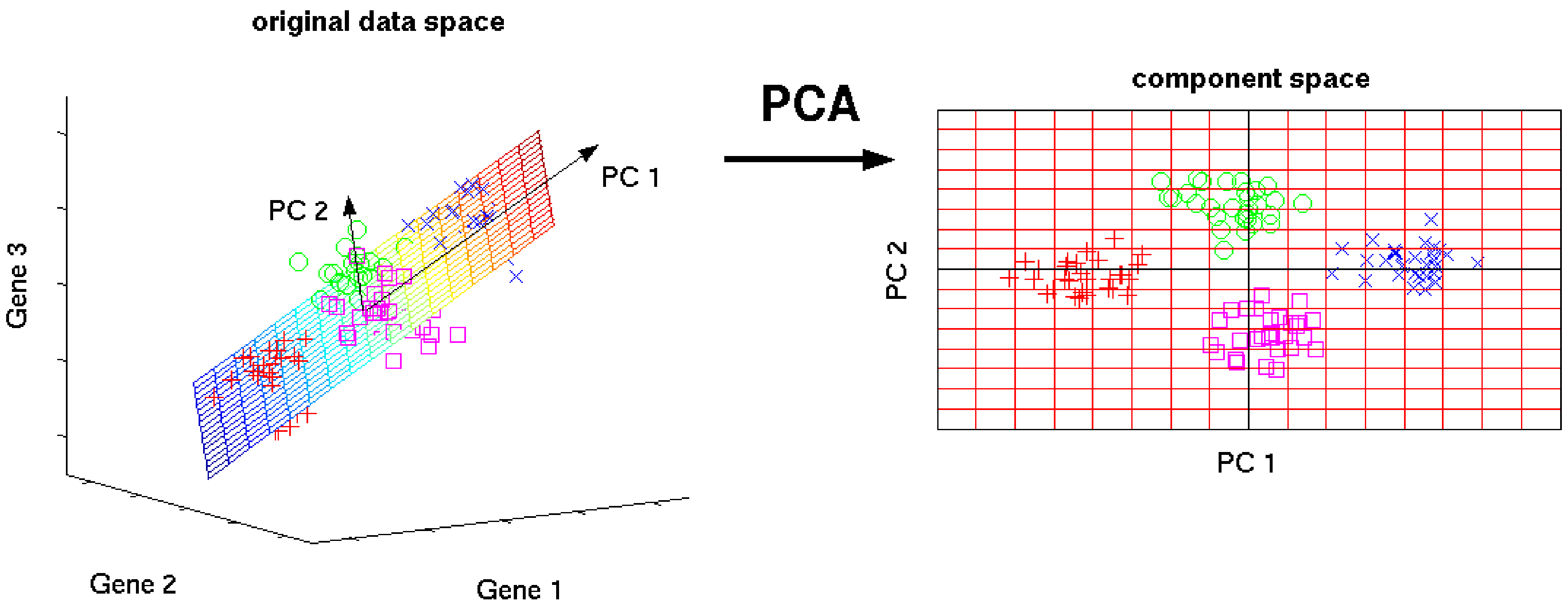
large projection errors, bad line to project to



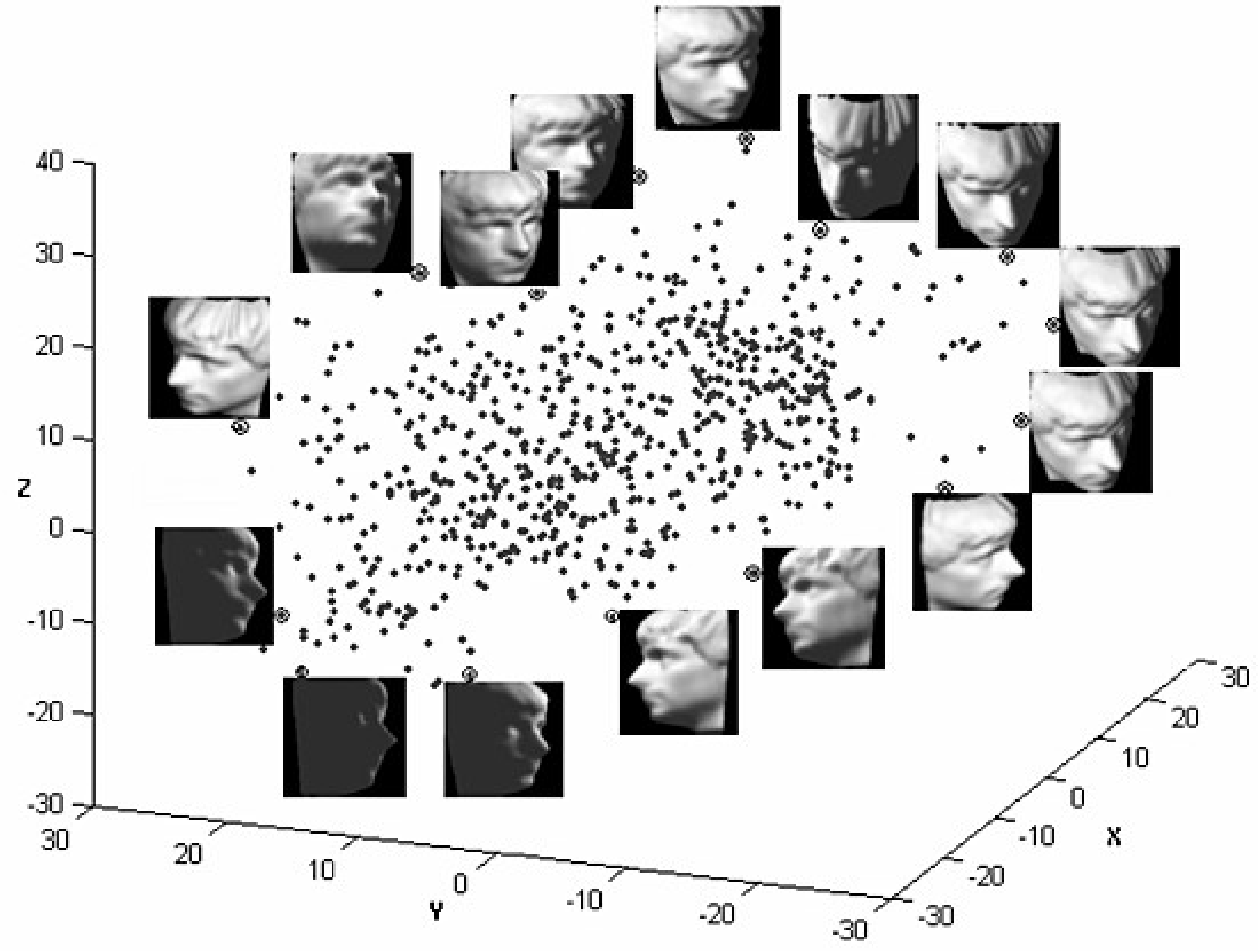
small projection errors, good line to project to

- Notice that the the good line to use for projection lies in the direction of largest variance

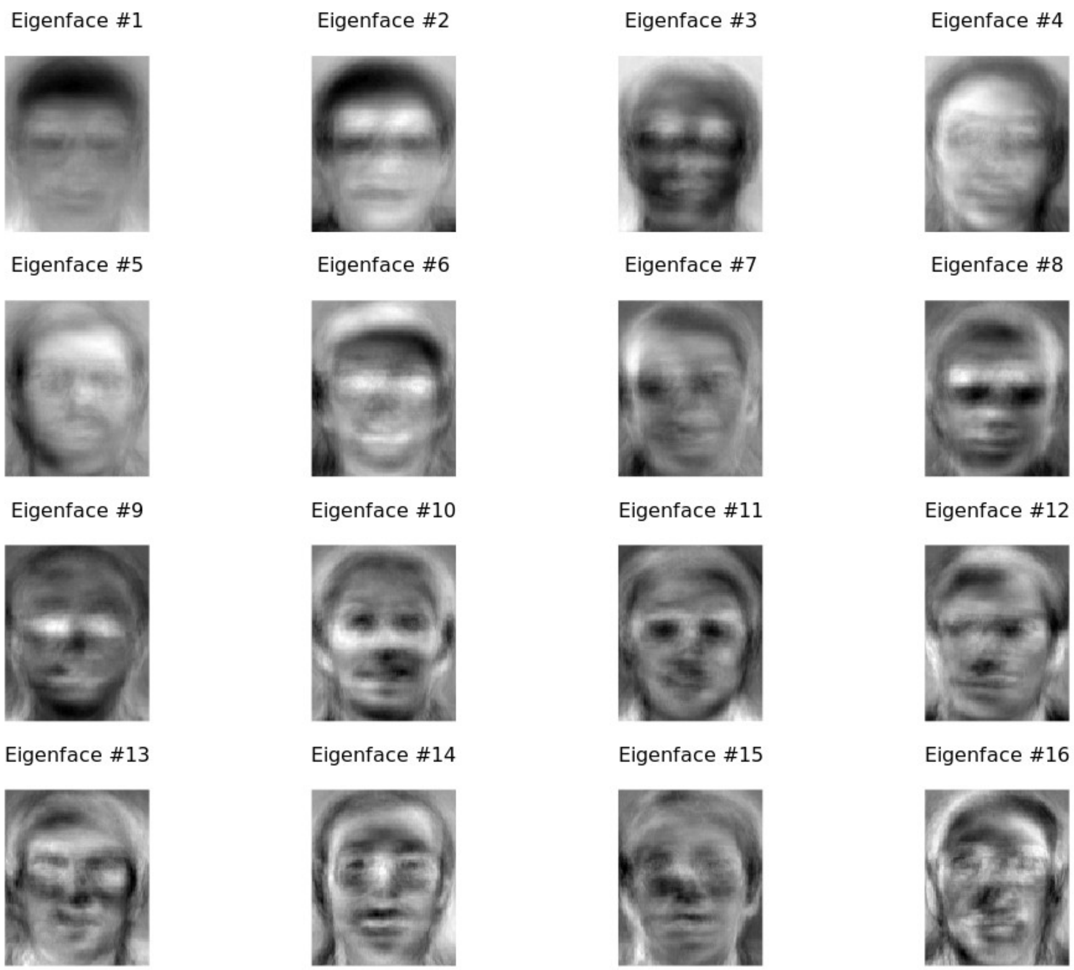
PCA in 3D


original data space
PC 2
PC 1
Gene 3
Gene 2
Gene 1
PCA
component space
PC 2
PC 1

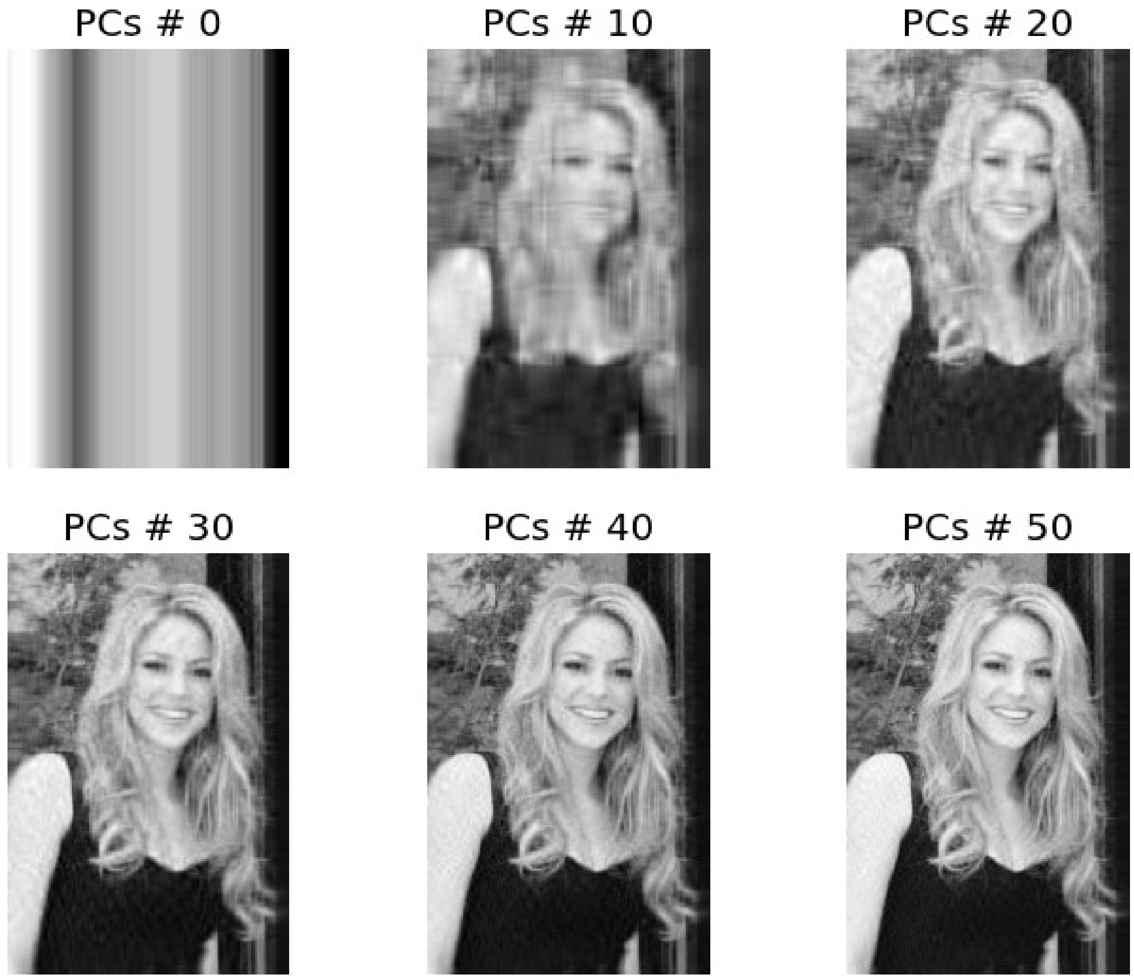
Neat application: Eigenfaces!



http://doi.ieeecomputersociety.org/cms/Computer.org/dl/trans/tp/2008/05/figures/ttp200805079619.gif



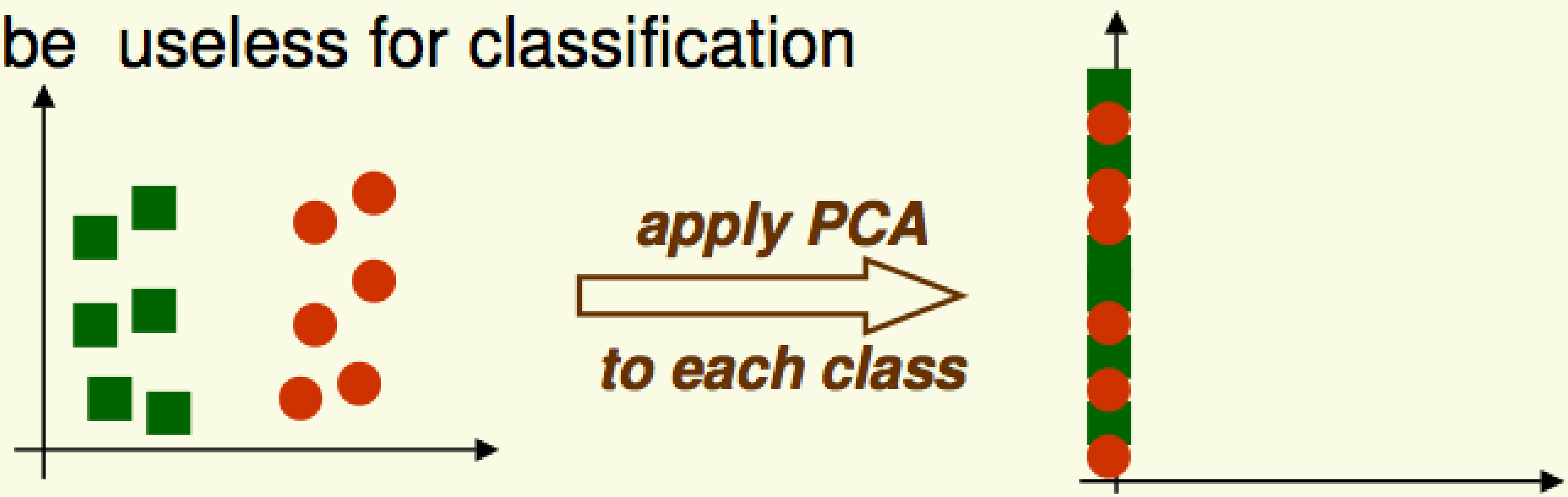
http://bytefish.de/static/images/blog/eigenfaces/subplot_eigenfaces.png



http://3.bp.blogspot.com/-eTWWNr-CyeI/Ti1l2h9ldnI/AAAAAAAAAE0/fUF2hl-hTMQ/s1600/pca_shakira.png

Drawbacks of PCA

- PCA was designed for accurate *data representation*, not for data classification
  - Preserves as much variance in data as possible
  - If directions of maximum variance is important for classification, will work
- However the directions of maximum variance may be useless for classification



Class 13
Deep Learning

Feature Engineering

- We have talked about this a few times.
  - Perhaps the data you are given in not the best data for making the predictions you want to make.
- The last question on Case Study 3 was a prime example.
  - How do you find features that make prediction easier?

Is this problem ubiquitous?

Important

Coming up with features is difficult, time-consuming, requires expert knowledge. "Applied machine learning" is basically feature engineering.

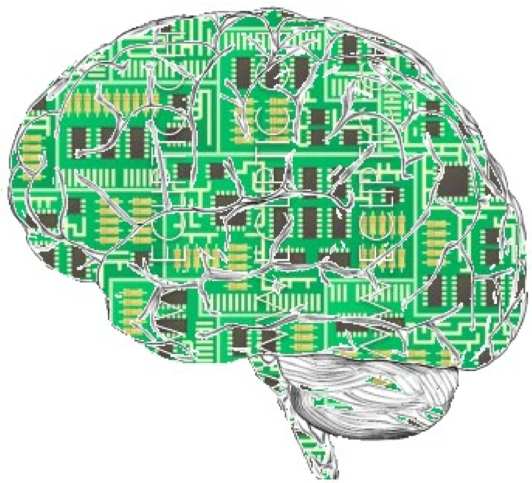
— Andrew Ng, Machine Learning and AI via Brain simulations

https://en.wikipedia.org/wiki/Feature_engineering

Deep learning

Important

- Deep learning (deep structured learning, hierarchical learning or deep machine learning) is a branch of machine learning based on a set of algorithms that attempt to **model high-level abstractions** in data by using **multiple processing layers**, with complex structures or otherwise, composed of **multiple non-linear transformations.**



https://en.wikipedia.org/wiki/Deep_learnin
g

WPI

Deep learning: More specifically

Important

1) Deep learning is a **collection** of different techniques, not a single technique.

2) Deep learning aims to automatically learn feature hierarchies from data.

- As opposed to “by hand” feature engineering.

3) Generally, deep learning is based on artificial neural networks.

- Though that is not always true, depending on what an “artificial neural network” really is.

WPI

Is deep learning supervised or unsupervised?

Depends!

Both!

Not quite PCA...

Linear/PCA

Input layer

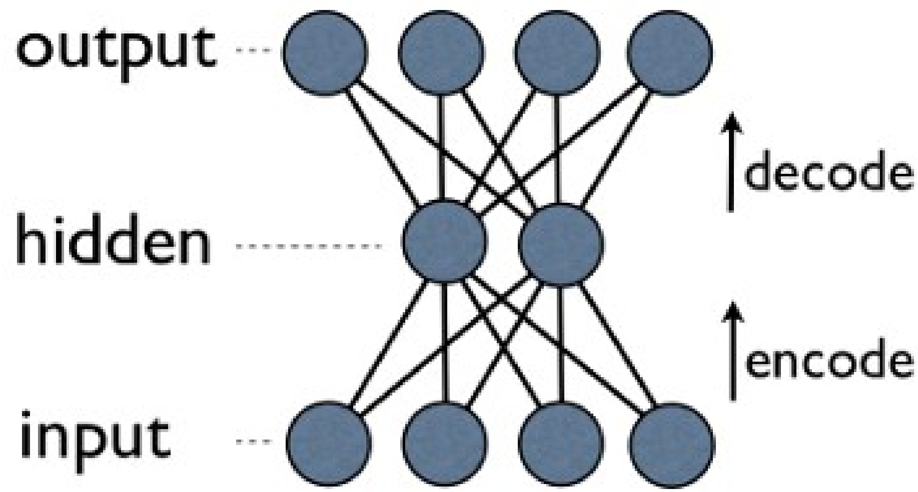
$$Wx = h$$

$$W^{+}h = \hat{x}$$

Weights

Hidden layer

Output layer



Non-linearity

Non-linear layer

Input layer

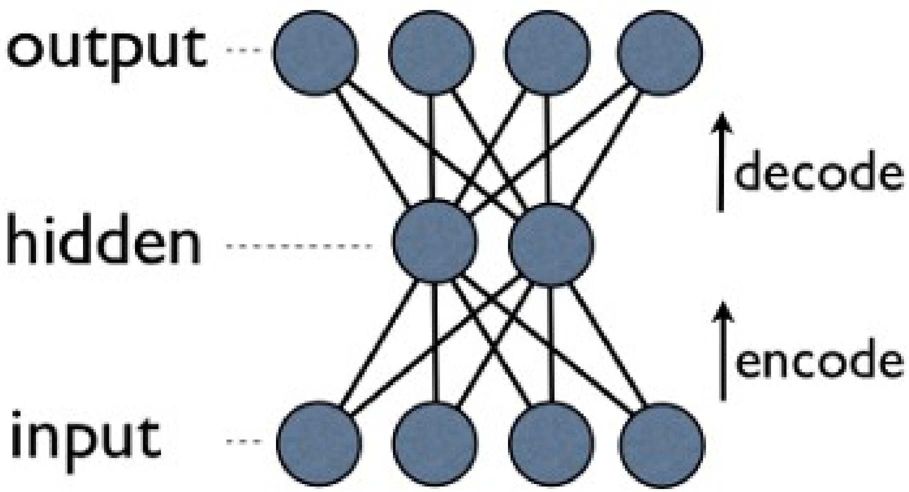
$$f(Wx + b) = h$$

Weights

Hidden layer

$$f(W^T h + b) = \hat{x}$$

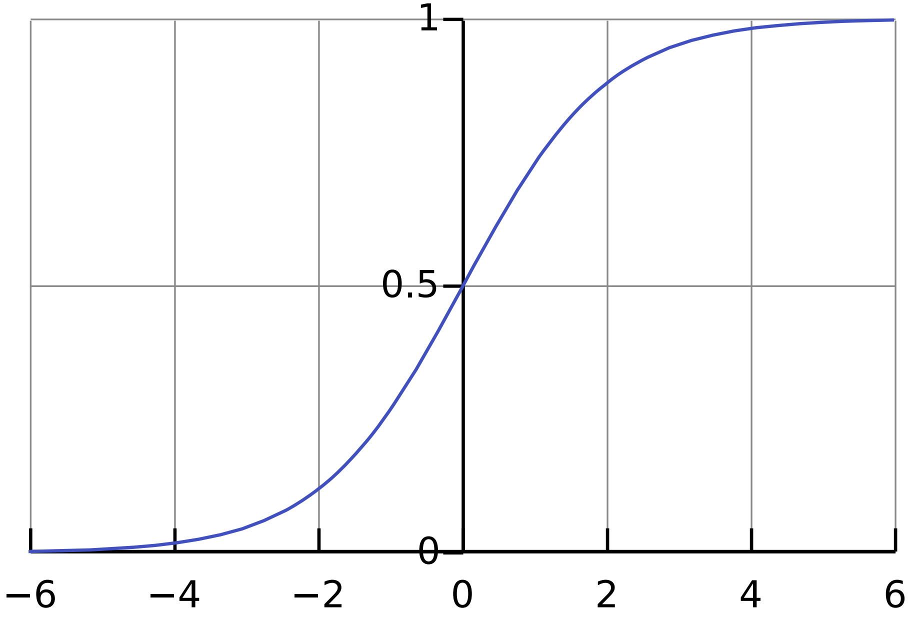
Output layer



What is f ?

Sigmoid (is one popular choice)!

$$f(x) = \frac{1}{1 + e^{-x}}$$



Not on exam

Huh!? Why sigmoid?

- One reason – The Universal Approximation Theorem

Let $\varphi(\cdot)$ be a nonconstant, bounded, and monotonically-increasing continuous function. Let I_m denote the *m*-dimensional unit hypercube $[0,1]^m$. The space of continuous functions on I_m is denoted by $C(I_m)$. Then, given any function $f \in C(I_m)$ and $\varepsilon > 0$, there exists an integer N, real constants $v_i, b_i \in \mathbb{R}$ and real vectors $w_i \in \mathbb{R}^m$, where $i = 1, \cdots, N$, such that we may define:

$$F(x) = \sum_{i=1}^{N} v_i \varphi\left(w_i^T x + b_i\right)$$

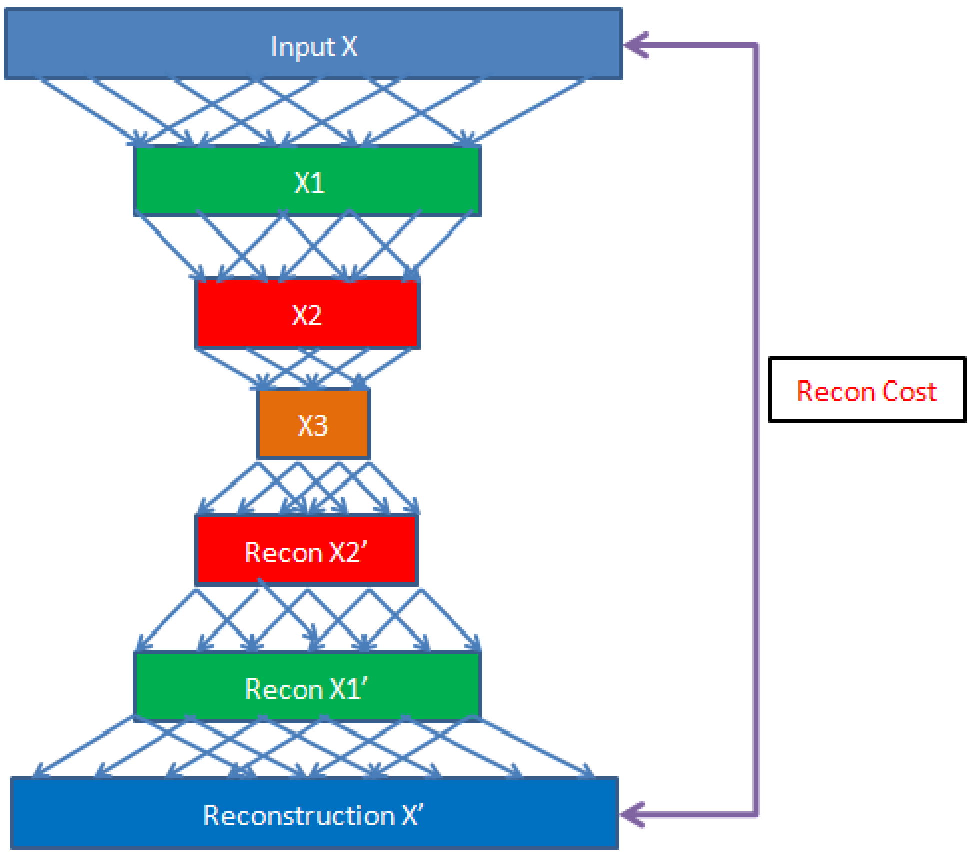
as an approximate realization of the function f where f is independent of φ; that is,

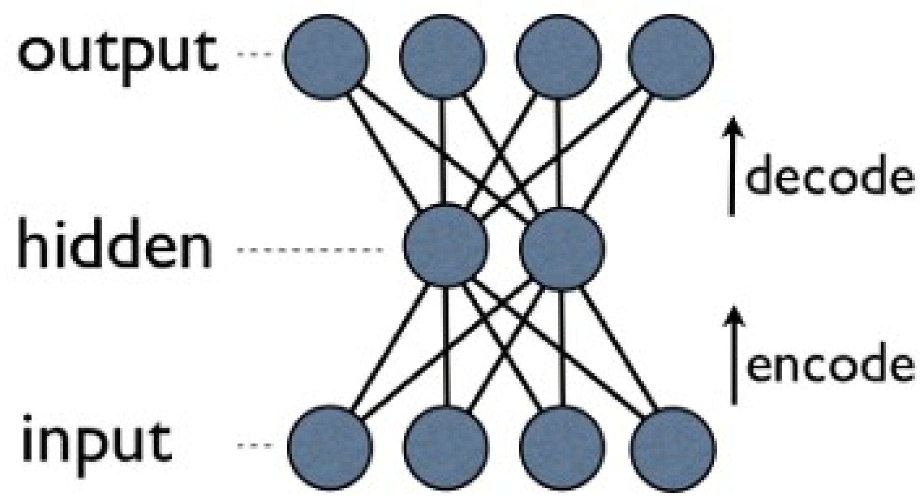
$$|F(x) - f(x)| < \varepsilon$$

for all $x \in I_m$. In other words, functions of the form $F(x)$ are dense in $C(I_m)$.

https://en.wikipedia.org/wiki/Universal_approximation_theorem

What makes it “deep” learning deep?



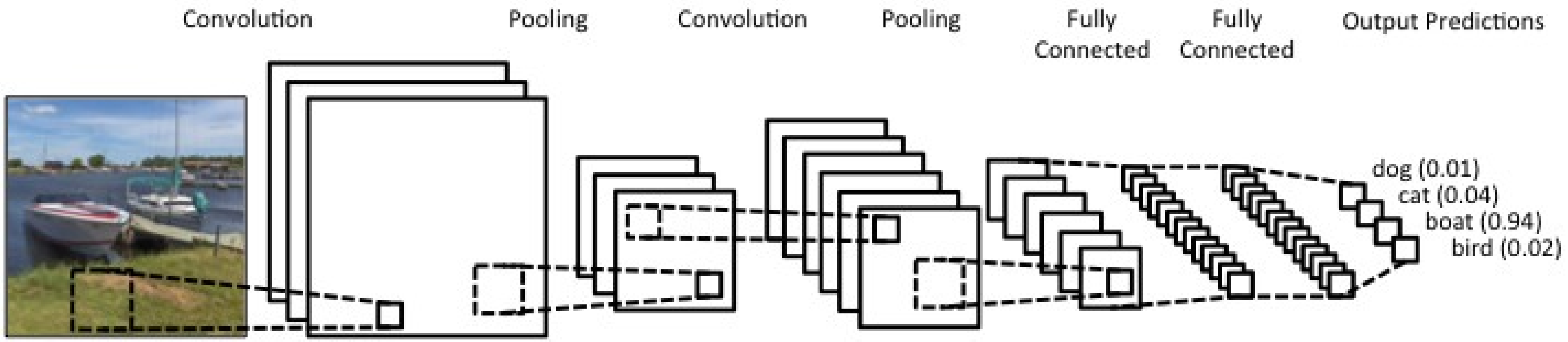


Different types of Deep Learning architectures (there are many :-)

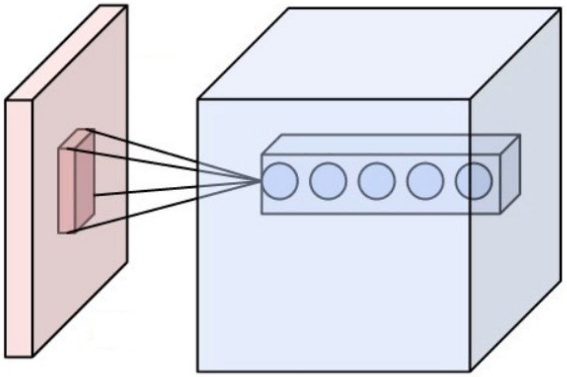
Important

- Convolutional Neural Network (CNN)
- Recurrent Neural Network
- Deep auto-encoders

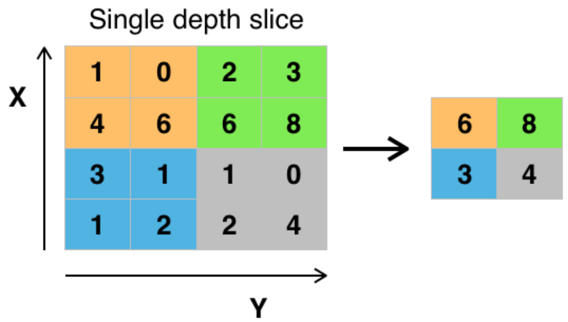
Convolutional Neural Network



https://www.clarifai.com/technology

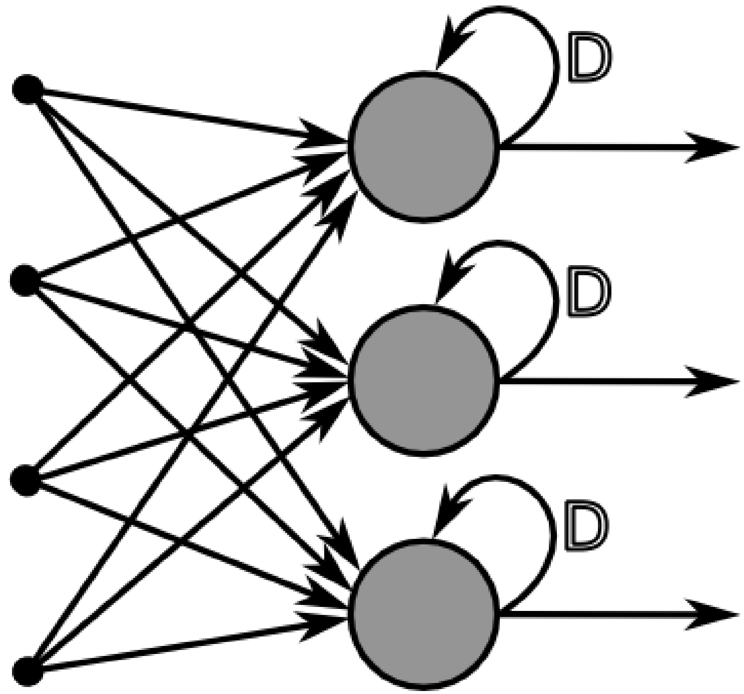


By Aphex34 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=45659236



By Aphex34 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=45673581

Recurrent Neural Network



Auto-encoders… the one I am most interested in.

- Did you ever run a compression algorithm on an already compressed file?
  - That is the idea :-)

WPI

Auto-encoders… the one I am most interested in.