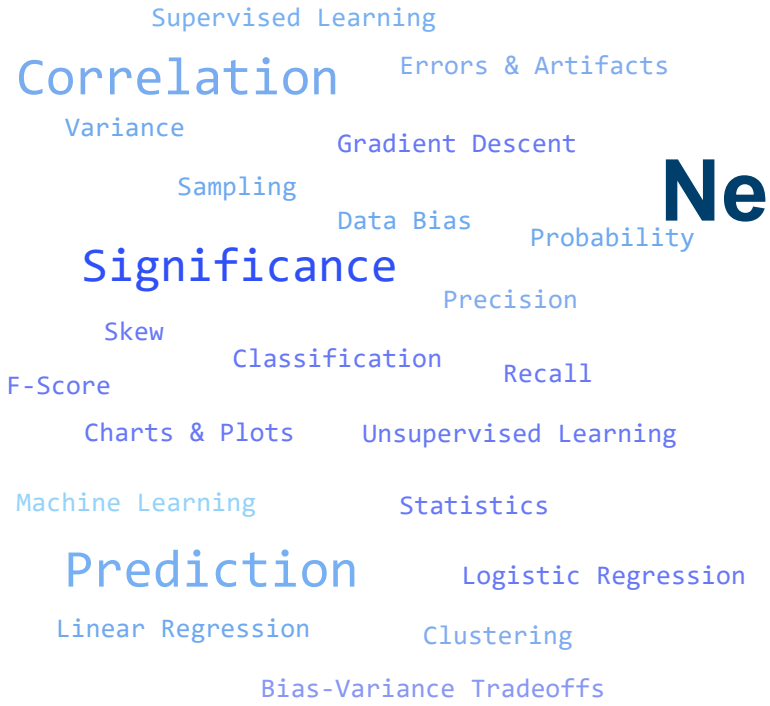


Data Science 1:
Introduction to Data Science

Nearest Neighbor Methods & Clustering

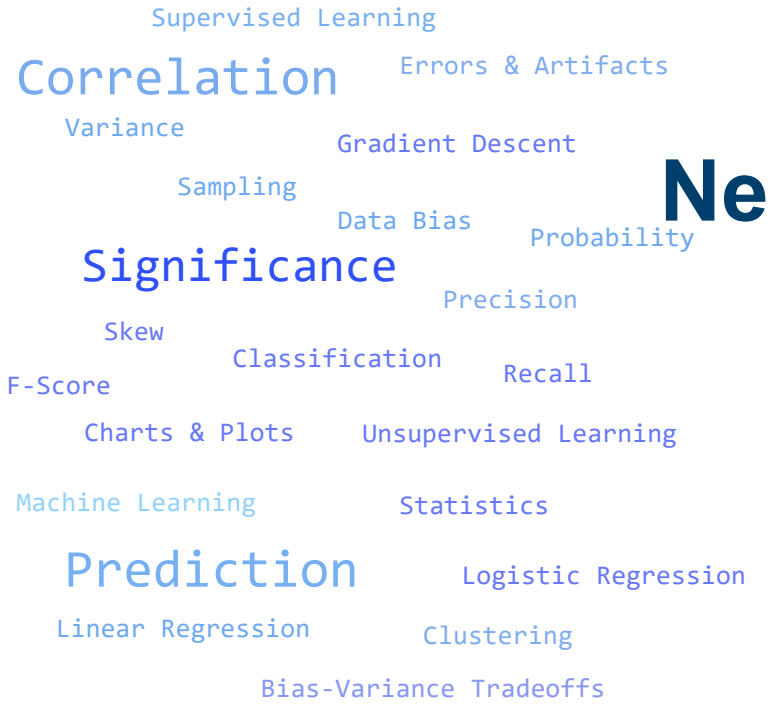
Winter 2025

Wolfram Wingerath, Jannik Schröder

Department for Computing Science
Data Science / Information Systems

Lecture slides based on content from “The Data Science Design Manual” (Steven Skiena, 2017) and associated course materials generously made available online by the author at https://www3.cs.stonybrook.edu/~skiena/data-manual/.

Special thanks to Professor Skiena for sharing these valuable teaching resources!



Data Science 1:
Introduction to Data Science

Nearest Neighbor Methods & Clustering

Winter 2025

Wolfram Wingerath, Jannik Schröder

Department for Computing Science
Data Science / Information Systems

Semester Schedule

CW 42	14. Oct	Lecture	1	Orga & Intro	1-26
CW 43	21. / 23. Oct	Lecture + Exercises	2	Probability, Statistics & Correlation	27-56
CW 44	28. Oct	Lecture	3	Data Munging, Cleaning & Bias	57-94 / “Invisible Women”
CW 45	04. / 06. Nov	Lecture + Exercises	4	Scores & Rankings	95-120
CW 46	11. Nov	Lecture	5	Statistical Distributions & Significance	121-154
CW 47	18. / 20. Nov	Lecture + Exercises	6	Building & Evaluating Models	201-236
CW 48	25. Nov	Guest Lecture	7	Data Visualization	155-200
CW 49	02. / 04. Dec	Lecture + Exercises	8	Intro to Machine Learning	351-390
CW 50	09. Dec	Lecture	9	Linear Algebra	237-266
CW 51	16. / 18. Dec	Lecture + Exercises	10	Linear Regression & Gradient Descent	267-288
CW 02	06. Jan	Lecture	11	Logistic Regression & Classification	289-302
CW 03	13. / 15. Jan	Lecture + Exercises	12	Nearest Neighbor Methods & Clustering	303-350
CW 04	20. Jan	Lecture	13	Data Science in the Wild	391-426
CW 05	27. / 29. Jan	Lecture + Exercises	14	Q&A / Feedback	
CW 06	03. / 04. Feb	Oral Exams (Block 1)		Preparation in our last session („Oral Exam Briefing“)	
CW 13	24. / 25. Mar	Oral Exams (Block 2)			

Oral Exam Schedule Open!

In Stud.IP, you can now schedule an oral exam:

- Tuesday session: 13:00-18:00
- Wednesday session: 10:00-18:00
- 20+10 min. per examination slot

Please *respect our time & commitment*, especially:

- Come well-prepared!
- Don’t ghost us! Don’t cancel last-minute!

Seeking Good Analogies

Many intellectual disciplines rest on analogies:

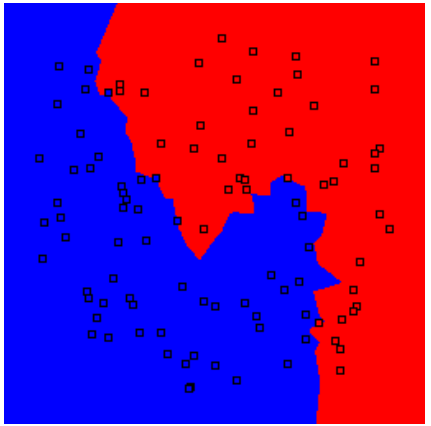
- Law: which legal precedent was most like this case?
- Medicine: how did I treat patients with similar symptoms, and did they survive?
- Real estate: what price did comparable properties sell for in the neighborhood?

Nearest Neighbor Classification

Identify which training example is most similar to the target, and take the class label from it.

The key issue here is devising the right **distance function** between rows/points.

Advantages: simplicity, interpretability, and non-linearity.



Which Representation is Better?

The nearest neighbors of an item in feature space “should” be like it.

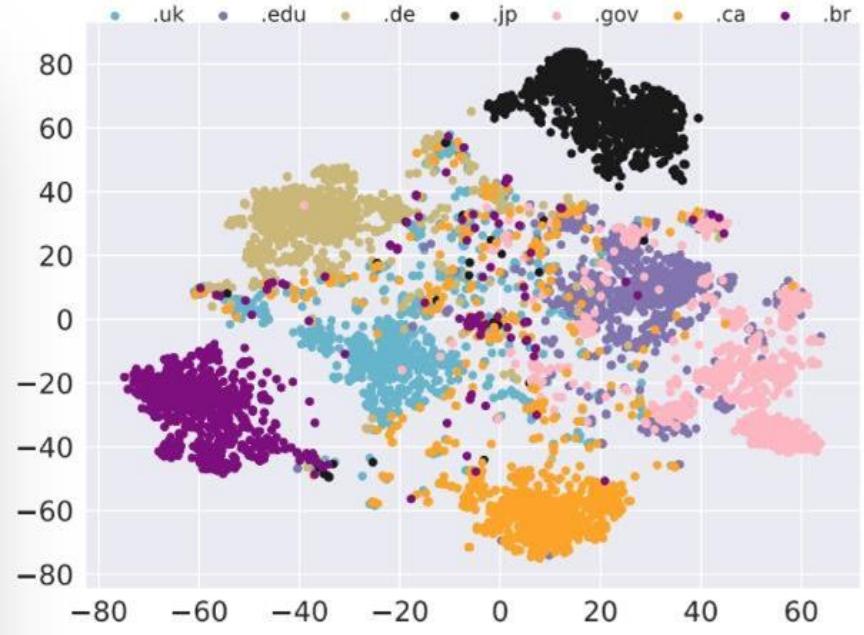
Assessing how well the nearest neighbors of points resemble targets is a great sniff test.

Which of the two feature representations of the web on the next slide is better?

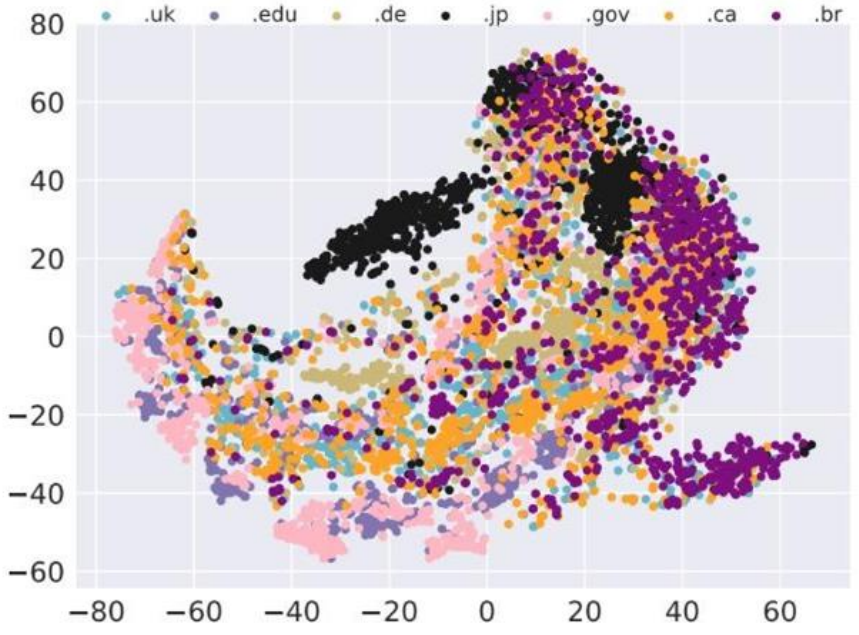
Websites	stonybrook.edu		baidu.com	
Methods	DeepWalk	RandNE	DeepWalk	RandNE
Neighbors	binghamton.edu	du.edu	sogou.com	alibaba.com
	www1.cuny.edu	wellesley.edu	qq.com	freepatentsonline.com
	qcpages.qc.cuny.edu	illinois.edu	youku.com	wolframalpha.com
	lehman.edu	amherst.edu	163.com	wikimediafoundation.org
	barnard.edu	ohio.edu	renren.com	news.softpedia.com
	esf.edu	binghamton.edu	tudou.com	worldwide.espacenet.com
	baruch.cuny.edu	smith.edu	naver.com	duckduckgo.com
	colgate.edu	vanderbilt.edu	taobao.com	whois.domaintools.com
	hunter.cuny.edu	macalester.edu	t.qq.com	images.apple.com
	hamilton.edu	lsa.umich.edu	baike.baidu.com	tineye.com

Websites	chase.com		wikipedia.org	
Methods	DeepWalk	RandNE	DeepWalk	RandNE
Neighbors	bankofamerica.com	fedex.com	wikipedia.com	timeanddate.com
	capitalone.com	travelocity.com	wikimediafoundation.org	groups.google.com
	citi.com	priceline.com	wikimedia.org	wolframalpha.com
	schwab.com	bankofamerica.com	openoffice.org	digg.com
	wellsfargo.com	capitalone.com	addons.mozillar.org	alexa.com
	discovercard.com	comcast.com	answers.com	spreadsheets.google.com
	creditcards.com	jdpower.com	wolframalpha.com	earth.google.com
	ameriprise.com	delta.com	dmoz.org	xkcd.com
	mastercard.us	ups.com	en.wiktionary.org	quora.com

Clusters by High Level Domain



(a) DeepWalk.



(b) RandNE.

Distance Metrics

Certain mathematical properties are expected of any distance measure, or *metric*:

- $d(x,y) \geq 0$ for all x, y (positivity)
- $d(x,y) = 0$ iff $x = y$ (identity)
- $d(x,y) = d(y,x)$ (symmetry)
- $d(x,y) \leq d(x,z) + d(z,y)$ (triangle inequality)

Not a Metric

Many natural similarity measures are not distance metrics:

- Correlation coefficient (-1 to 1)
- Cosine similarity / dot product (-1 to 1)
- Cheapest airfare (think triangle inequality)

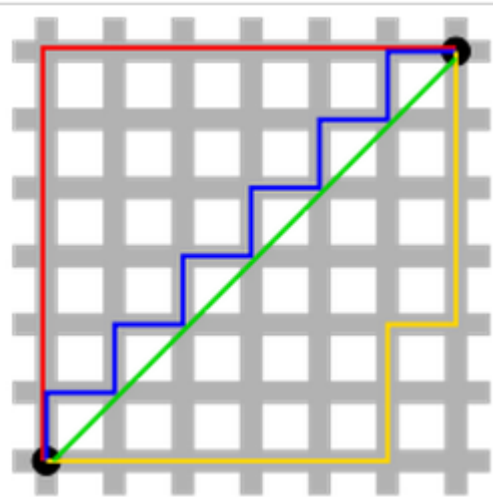
Euclidean Distance Metric

The traditional Euclidean distance metric weighs all dimensions equally:

$$d(x,y) = \sqrt{\sum_{i=1}^{d} |x_i - y_i|^2}$$

We might use a coefficient c_i to give a different weight to each dimension, but should *at least* normalize to make dimensions comparable.

L_k Distance Norms



To generalize Euclidean distance:

$$d(x,y) = (\sum_{i=1}^{d} |x_i - y_i|^k)^{1/k}$$

- $k = 1$ gives the Manhattan distance metric
- $k = 2$ gives the Euclidian distance metric
- $k = \infty$ gives the maximum component

k regulates the trade off between largest and total dimensional difference.

L_∞ Norm Example

Consider a vector v = [2, 2, 1]:

- $k = 1 : d_1(v, \bar{0}) = (2^1 + 2^1 + 1^1)^{1/k} = 5$
- $k = 2: d_2(v, \bar{0}) = (2^2 + 2^2 + 1^2)^{1/2} = 3$
- $k = 3: d_3(v, \bar{0}) = (2^3 + 2^3 + 1^3)^{1/3} \approx 2.571$
- $k = 10: d_{10}(v, \bar{0}) = (2^{10} + 2^{10} + 1^{10})^{1/10} \approx 2.144$
- …
- $k = \infty: d_k(v, \bar{0}) = (2^\infty + 2^\infty + 1^\infty)^{1/\infty} \approx \max(2,2,1)$

Which Point is Farther from (0,0)?

Is p_1=(2,0) or p_2=(1.5,1.5) farther from (0,0)?

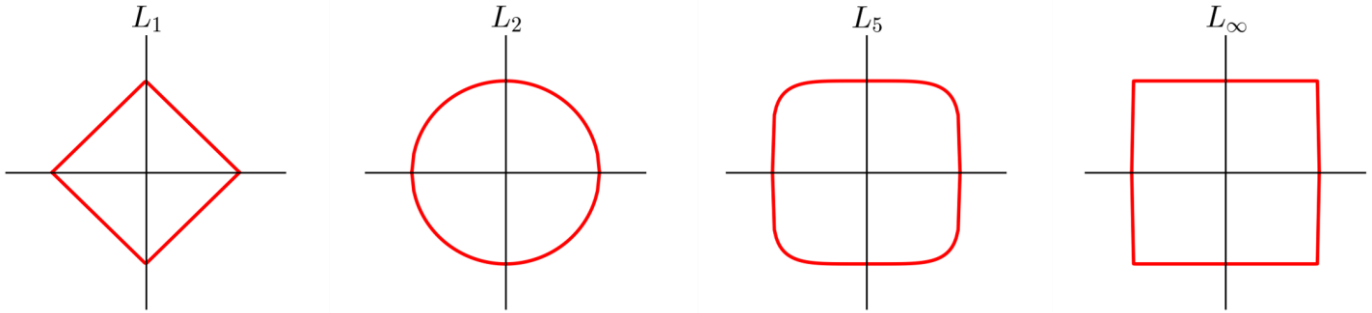
- For $k = 1$, the distances are 2 and 3, so p_2
- For $k = 2$, the distances are 2 and 2.12, so p_2
- For $k = \infty$, distances are 2 and 1.5, so p_1

The distance metric sets which point is closer.

Are we more worried about random noise or dimensional outliers / artifacts?

Circles for Different *k*

The shape of the L_k "circle" governs which points are equal neighbors about the origin.



The distinction here become particularly important in higher dimensional spaces: do we care about deviations in all dimensions or primarily the biggest?

Projections from Higher Dimensions

Projection methods (like SVD) compress or ignore dimensions to reduce representation complexity.

Nearest neighbors in such spaces can be more robust than in the original space.

Dimensional Egalitarianism

Although L_k norms in principle weigh all dimensions equally, the scale matters.

But note that the real impact of height will differ depending upon whether it is measured in centimetres, meters, or kilometers.

This is why we use Z-scores for normalization!

Regression / Interpolation by NN

The idea of nearest neighbor classification can be generalized to function interpolation, by averaging the values of the *k* nearest points.

Weighted averaging schemes can value points differently according to (1) distance rank, (2) actual distances.

Similar ideas work for all classification methods.

K-Nearest Neighbors

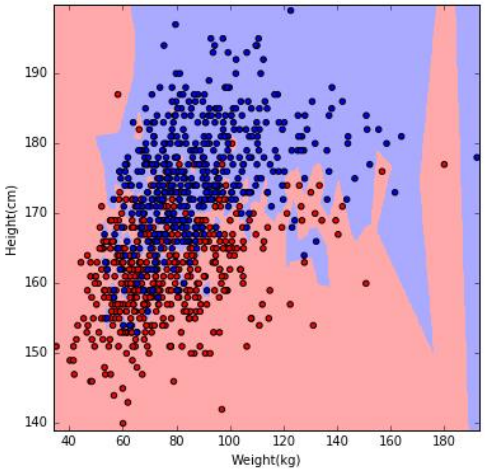
NN classification produce non-linear classifiers, because each training point changes the separating boundary.

More robust classification or interpolation follow from voting over the *k* closest neighbors for *k>1*.

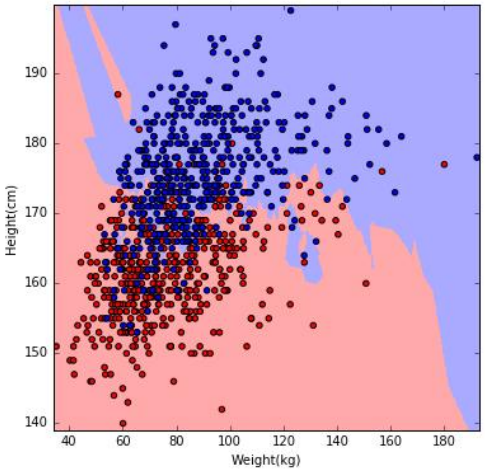
Gender Classification by Height/Weight

Smoother boundaries follow from larger k:

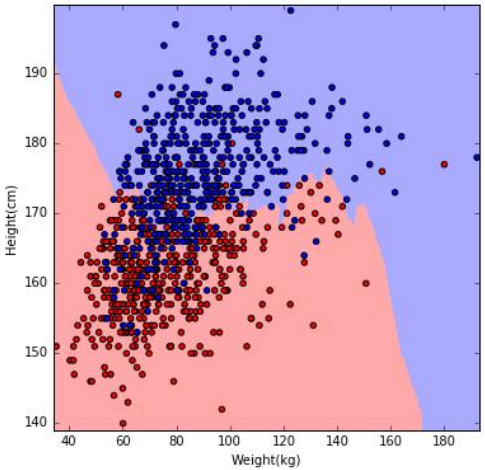
k=1



k=3



k=10



Finding Nearest Neighbors

Given *n* points in *d*-dimensions, it takes *O(nd)* time to find the NN using brute force search.

For large training sets or high dimensionality this becomes very expensive.

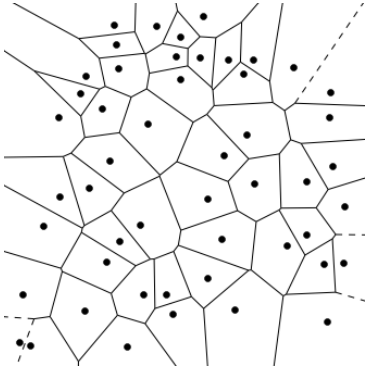
This motivates use of more sophisticated data structures: grid indices, kd-trees, Voronoi diagrams, and locality sensitive hashing.

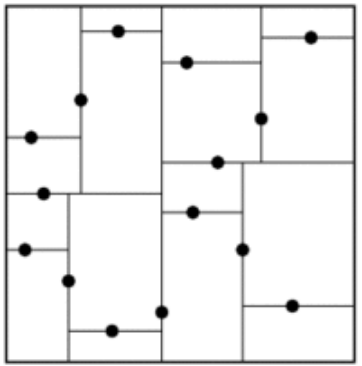
Voronoi Diagrams / kd-trees

Voronoi diagrams partition space into regions sharing nearest neighbors.

Efficient algorithms exist for finding nearest neighbors in low dimensions, such as kd-trees and ball trees.

But exact NN search is doomed to reduce to linear search for high-enough dimensionality data.

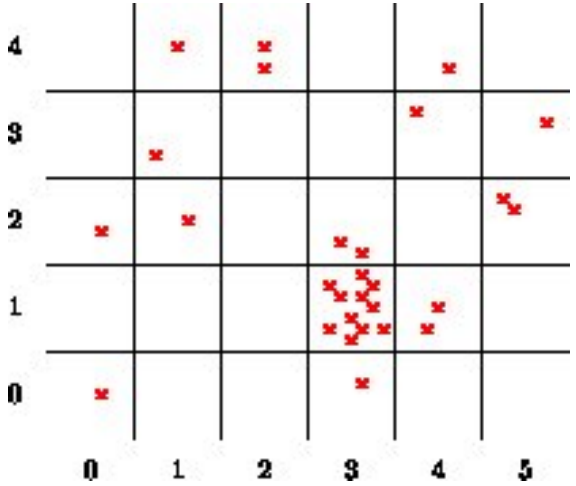




Grid Files

Bucketting points on a regularly-spaced grid provides a way to group points by similarity.

But such an index becomes expensive as the number of dimensions increases.



Locality Sensitive Hashing (LSH)

Hashing could speed nearest-neighbor search if nearby points got hashed to the same bucket.

But normal hashing uses hash functions that spreads similar items to distant buckets.

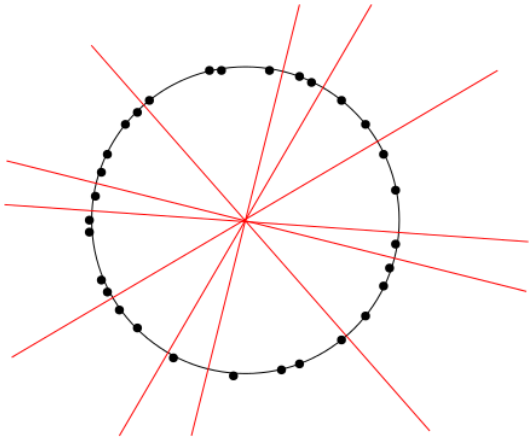
Locality sensitive hashing (LSH) takes points or vectors a and b such that likely h(a)=h(b) iff a is near b.

LSH for Points on a Sphere

Pick random planes cutting through the origin.

If near each other, two points are likely on the same side (left or right) of a given random plane.

L/R patterns for d random planes define a d-bit LSH hash code.



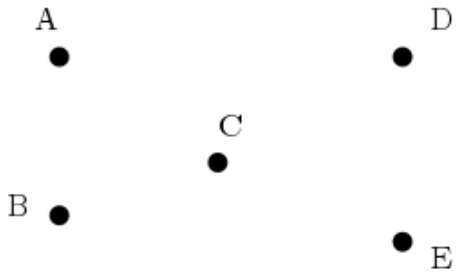
Network Data

Many datasets have natural interpretations as graphs/networks:

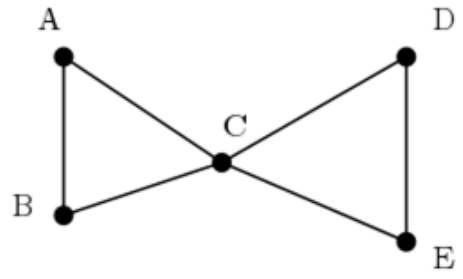
- Social networks: vertices are people, edges are friendships.
- WWW: vertices=pages, edges=hyperlinks.
- Product/customer networks: edges=sales.
- Genetic networks: v=genes, e=interactions.

Point Sets and Graphs

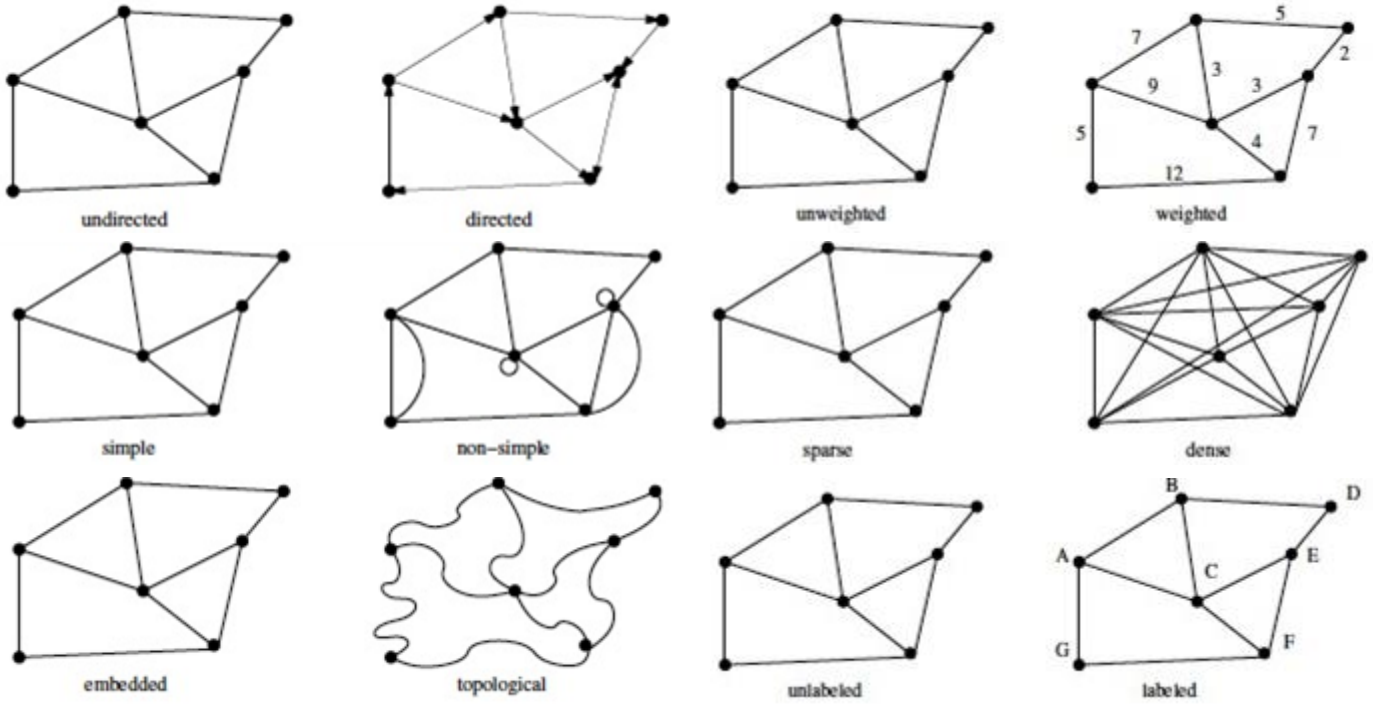
- Point sets naturally define graphs: add an edge *(x,y)* if *x* and *y* are close enough.
- Graphs naturally define point sets: perform an SVD of the adjacency matrix.



	A	B	C	D	E
A	0	6	7.2	13	14.8
B	6	0	6.3	14.3	13.0
C	7.2	6.3	0	8.1	7.6
D	13	14.3	8.1	0	4
E	14.8	13.0	7.6	4	0



Taxonomies of Graph Types



Talking the talk is part of walking the walk.

Classical Graph Algorithms

Distances in graphs are naturally defined in terms of the shortest path between vertices.

Classical algorithms for finding shortest paths, connected components, spanning trees, cuts, flows, matchings, topological sorting can be applied to any appropriate network.

PageRank

PageRank(v,G) corresponds to the probability that a random walk on *G* ends up at vertex *v*.

The basic formula is:

$$PR_j(v) = \sum_{(u,v)\in E} \frac{PR_{j-1}(u)}{out - degree(u)}$$

This recursive formula defines an iterative algorithm, which quickly converges in practice.

If All Roads Lead to Rome...

… then Rome must be an important place.

PageRank is a good feature to measure centrality or importance of vertices, like Wikipedia pages:

PageRank PR1 (all pages)	
1	Napoleon
2	George W. Bush
3	Carl Linnaeus
4	Jesus
5	Barack Obama
6	Aristotle
7	William Shakespeare
8	Elizabeth II
9	Adolf Hitler
10	Bill Clinton

PageRank PR2 (only people)	
1	George W. Bush
2	Bill Clinton
3	William Shakespeare
4	Ronald Reagan
5	Adolf Hitler
6	Barack Obama
7	Napoleon
8	Richard Nixon
9	Franklin D. Roosevelt
10	Elizabeth II

PageRank in Practice

Design decisions in PageRank include:

- Editing the graph to remove irrelevant vertices/edges (spam).
- Removing outdegree zero vertices.
- Tuning: the damping factor.

PageRank is less important to Google today than popularly supposed.

Graph Embeddings (DeepWalk)

Networks based on similarity or links define very sparse feature vectors. Techniques like SVD can compress the matrix, but are expensive.

Pairs of vertices which often appear near each other on random walks through a network may actually be similar.

Thus we can use data from random walk neighborhoods to train network representations!

Nearest Neighbors in Wikipedia

The links between pages defines the network.

Ludwig van Beethoven
- Franz Schubert (0.489)
- Johannes Brahms (0.532)
- Wolfgang Mozart (0.567)
- Robert Schumann (0.576)
- Gustav Mahler (0.635)

Barack Obama
- George W. Bush (0.474)
- Hillary Clinton (0.657)
- Bill Clinton (0.658)
- Joe Biden (0.750)
- Al Gore (0.791)

Scarlett Johansson
- Kirsten Dunst (0.784)
- Natalie Portman (0.786)
- Gwyneth Paltrow (0.796)
- Brad Pitt (0.858)
- Cameron Diaz (0.891)

Mick Jagger
- John Lennon (0.687)
- Keith Richards (0.687)
- Paul McCartney (0.796)
- Ronnie Wood (0.822)
- Eric Clapton (0.833)

Albert Einstein
- Richard Feynman (1.049)
- Max Planck (1.073)
- Freeman Dyson (1.107)
- Stephen Hawking (1.153)
- Robert Oppenheimer (1.156)

Steven Skiena
- Larry Page (1.597)
- Sergey Brin (1.598)
- Danny Hillis (1.644)
- Andrei Broder (1.652)
- Mark Weiser (1.653)

Random Projection Graph Embeddings

Random projection methods compute the dot product of each adjacency matrix row with random vectors.

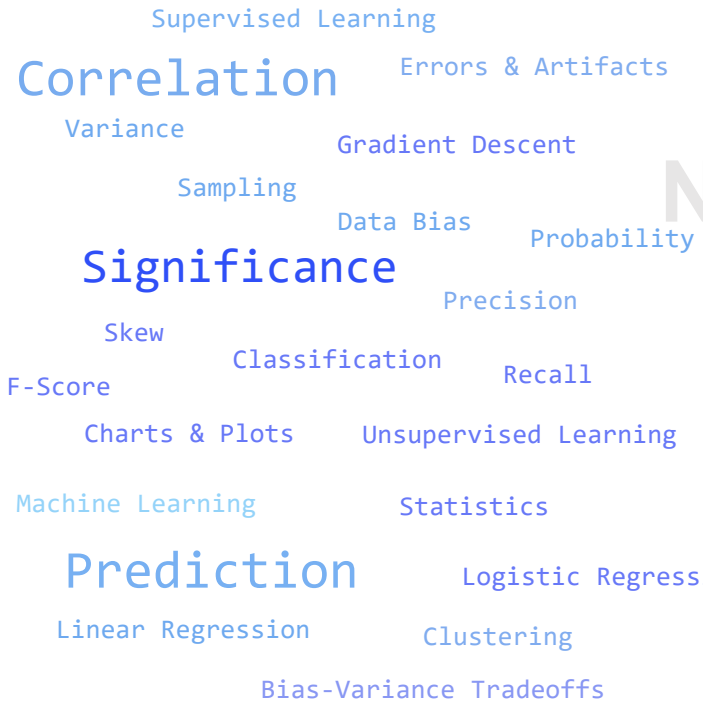
Similar rows should produce similar dot products:

[0,1,1,0,1,1,0] [5,-2,1,4,2,-3,-1] = -2+1+2-3 = -2

[0,1,0,0,1,1,0] [5,-2,1,4,2,-3,-1] = -2+2-3 = -1

[1,0,1,1,0,0,1] [5,-2,1,4,2,-3,-1] = 5+1+4-2 = 8

The Johnson-Lindenstrauss theorem says random projection in $O(\log n)^2$ dimensions preserves distances.



Data Science 1:
Introduction to Data Science

Nearest Neighbor Methods &

Clustering

Winter 2025

Wolfram Wingerath, Jannik Schröder

Department for Computing Science
Data Science / Information Systems

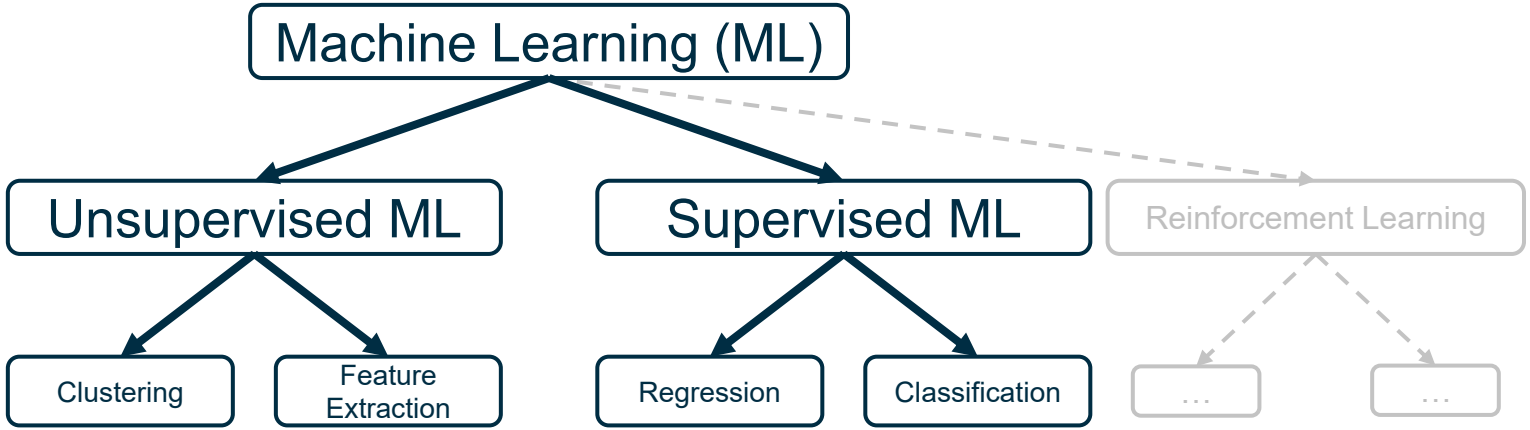
Supervised vs. Unsupervised Learning

The methods discussed so far assume class labels or target variables in the training data.

Unsupervised methods try to find structure in the data, by providing labels (clusters) or values (rankings) without a trusted standard.

Semi-supervised methods amplify small amounts of labeled data into more.

Supervised vs. Unsupervised Learning



ML approaches can be classified by the way that the procedures work and how they use labeled (supervised) or unlabeled (unsupervised) data

Clustering

Clustering is the problem of grouping points by similarity.

Often elements come from a small number of “sources” or “explanations”, and clustering is a good way to reveal these origins.

Similarity is defined by some underlying distance function / metric.

How Many Clusters Do You See?

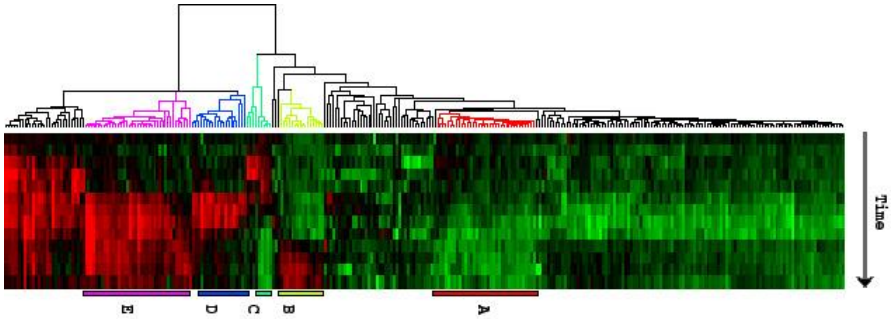
Clustering is an inherently ill-defined problem since they upon context and the eye of the beholder.

How many do you see?

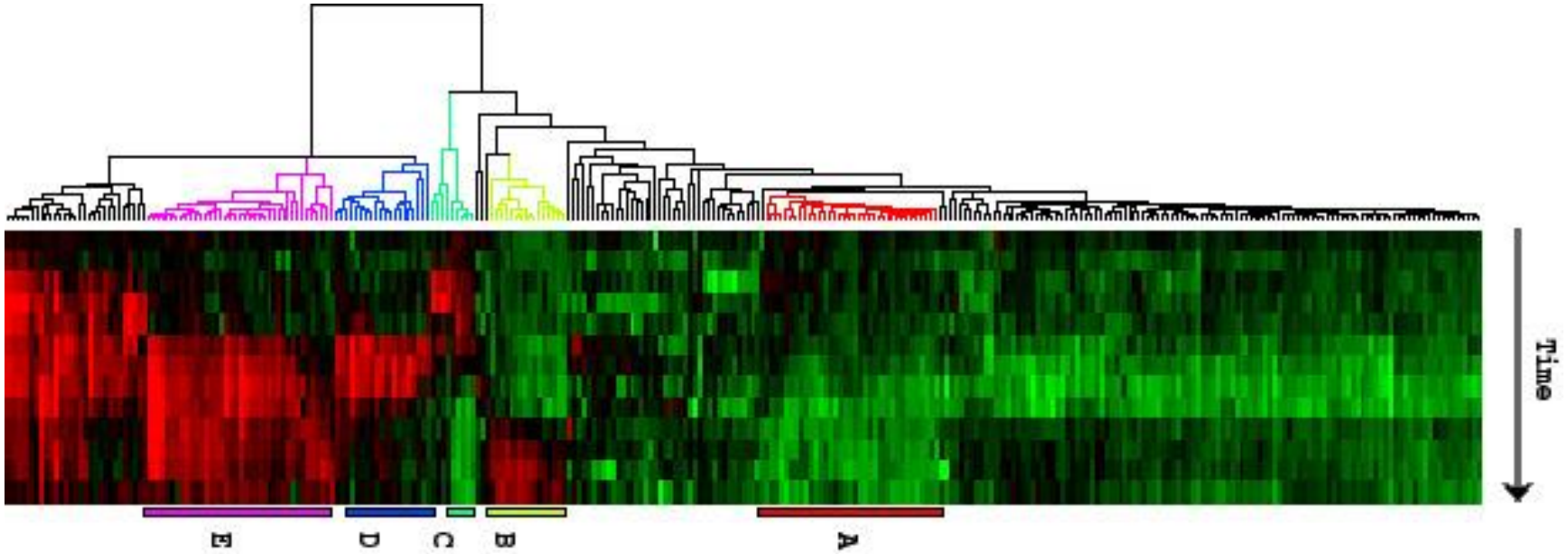
Compact, circular clusters are natural but not universal.

Clustering Gene Expression Data

- Clustering the columns groups genes active in the same phases of the cell cycle.
- Biological clusterings are often associated with dendograms or phylogenic trees.



Clustering Gene Expression Data



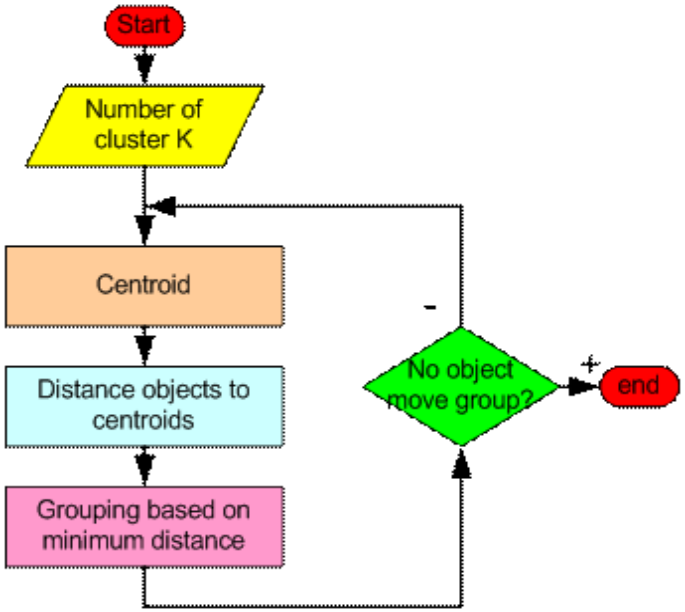
Why Clustering?

- Hypothesis development – how many distinct populations are there in your data?
- Modeling over smaller groups – build separate predictive models for each cluster.
- Data reduction – replace / represent each cluster of items by its centroid.
- Outlier detection – which items are far from cluster centers, or stuck in tiny clusters?

K-Means Clustering

Pick *k* points as centers, then assign all examples to the nearest center.

Recalculate the center, and repeat until sufficiently stable.



Running Time

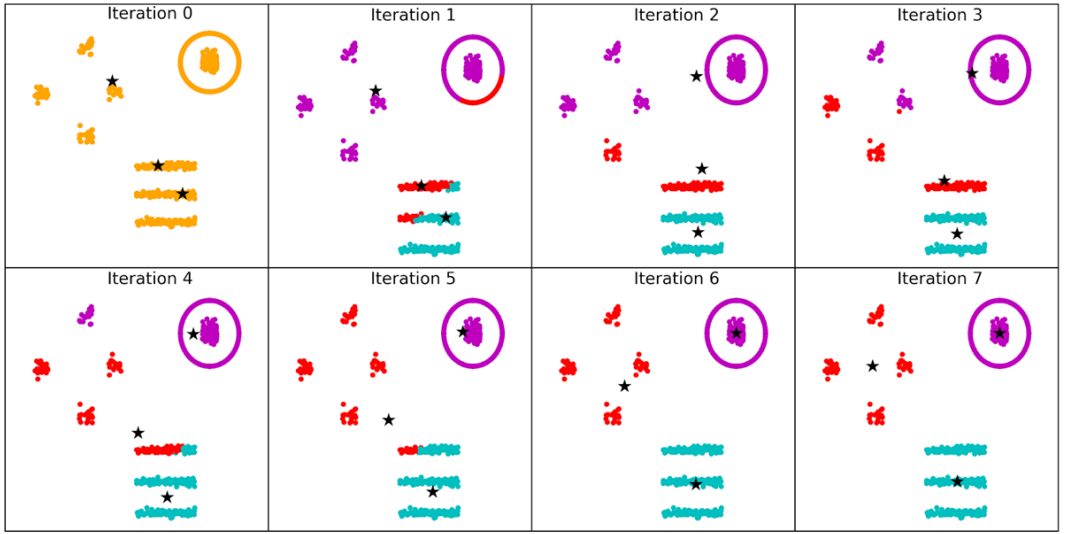
Finding the nearest neighbor among *k* cluster centers takes *O(kd)* per point, or *O(nkd)*.

Finding the centroids of each of k clusters takes *O(nd)* per cluster, or *O(nkd)*.

The number of iterations is usually small, but can be exponential, bounded by the number of partitions since we end when a partition repeats.

K-Means Clustering Example

It can get stuck in local optima, but generally does pretty well.

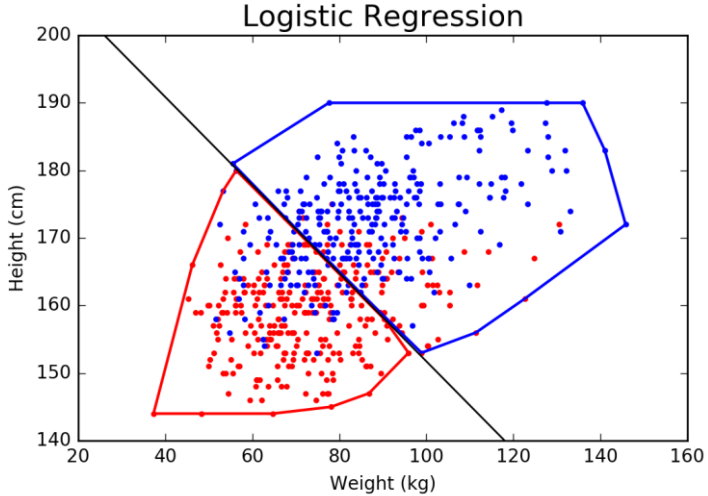


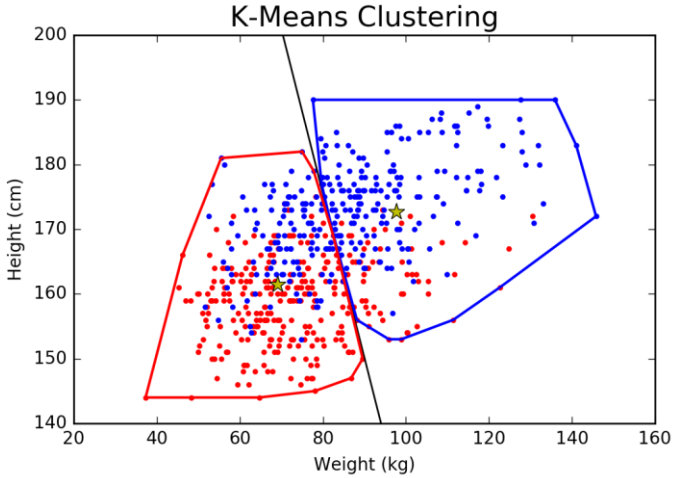
K-Means vs. Logistic Regression

K-means: 240w / 112m red, 174m / 54w blue

Logistic: 229w / 63m red, 223m / 65w blue

But K-means was unsupervised!





Centroids or Center Points?

Centroids are not well defined in clustering non-numerical attributes such as color or gender.

$$C_d = \frac{1}{|S'|} \sum_{p \in S'} p[d]$$

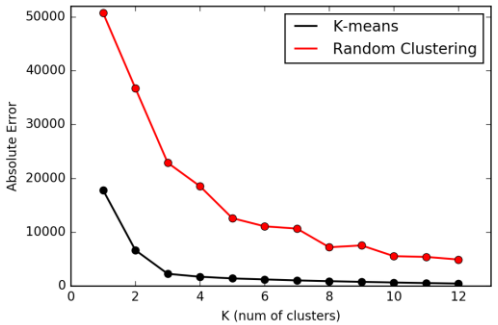
Using the centermost input example as center means we can run k-means so long as we have a meaningful distance function.

How Many Clusters?

The “right” number of clusters is usually unknown prior to clustering.

The SQE of points from their center should generally decrease when adding clusters.

But the SQE should decrease slowly once exceeding the right number of clusters

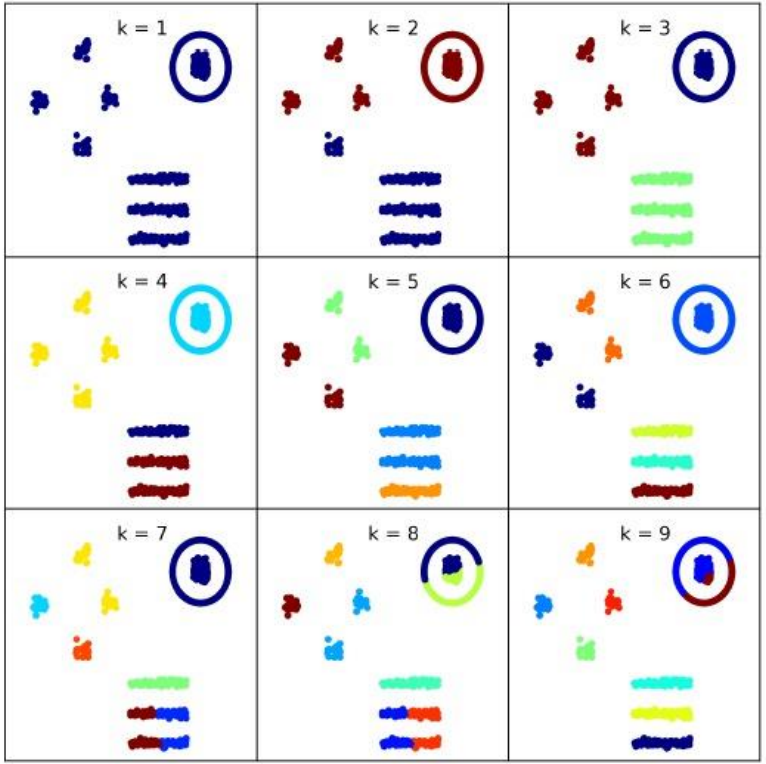


Limitations of K-means

K-means wants round clusters, so it has trouble with:

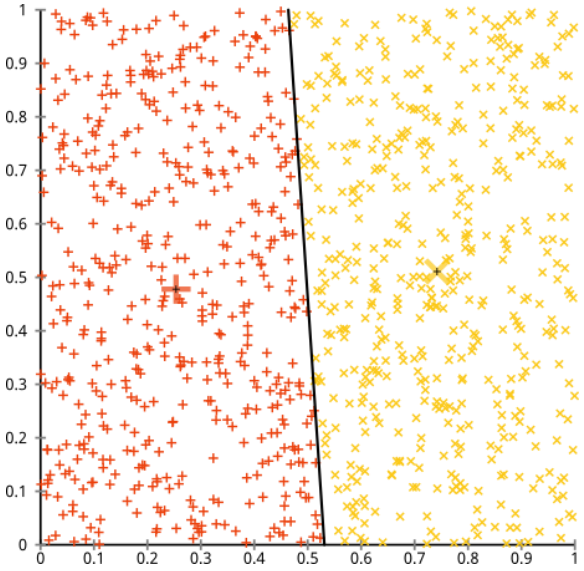
- nested clusters, and
- long thin clusters.

Repeated runs help avoid local optima.

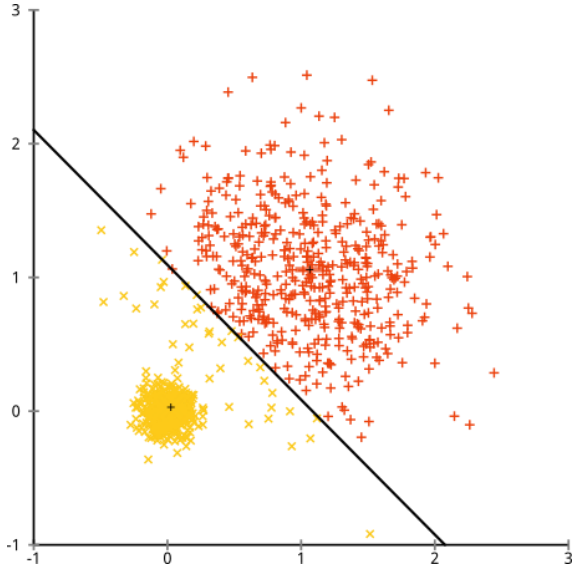


Bad Cases for K-Means

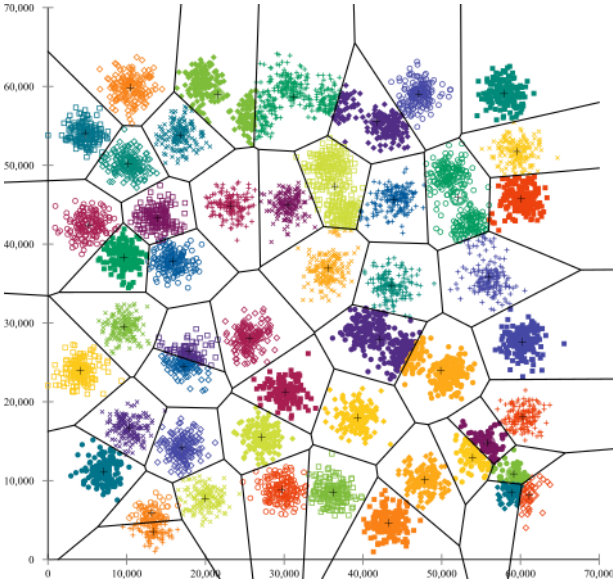
Uniform points



disparate variances

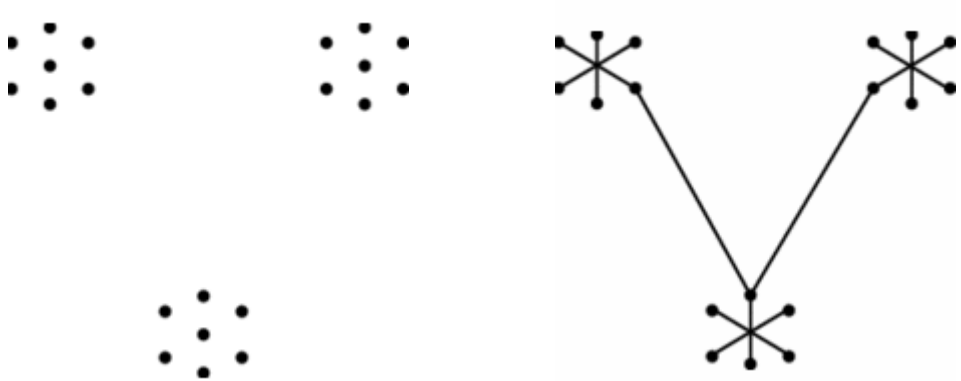


local minima (k=50)



Agglomerative Clustering

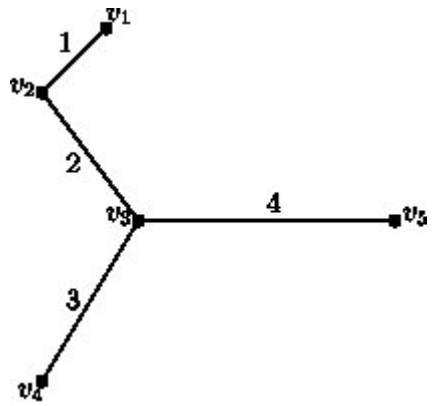
These bottom-up methods merge repeatedly merge the two nearest clusters.

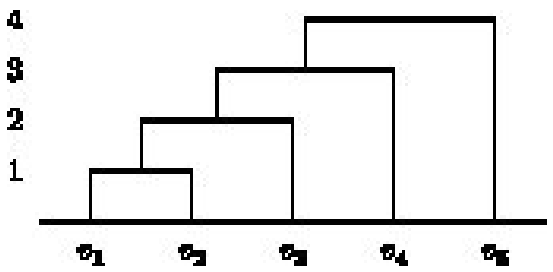


Minimum Spanning tree = single-link clustering

Kruskal's Algorithm and Dendograms

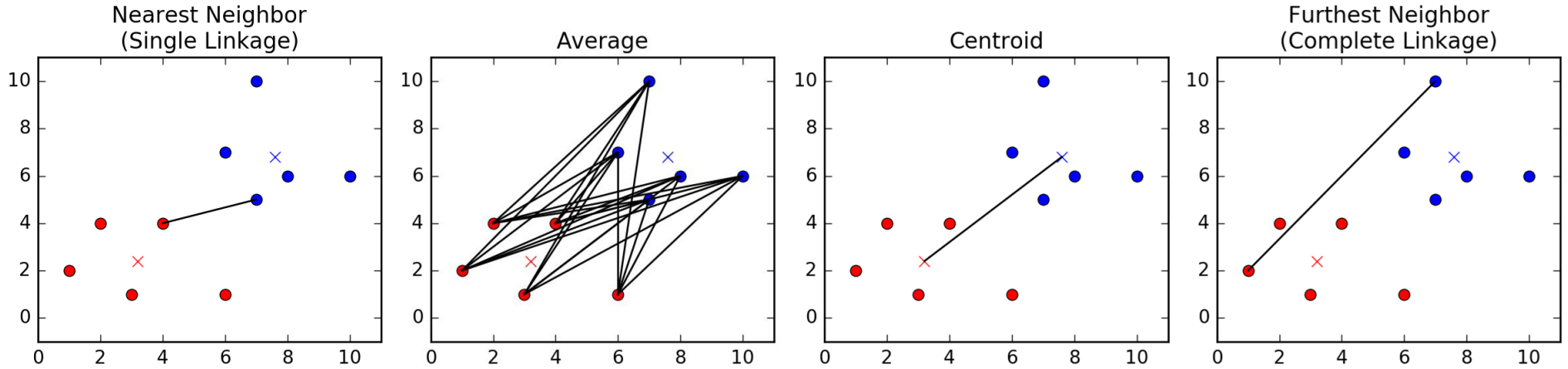
Dendograms are constructed by reflecting the height of the merge as the edge in question, and permuting the vertices so merges take place between neighboring clusters.





What Does Closest Cluster Mean?

The pointwise distance metric is not enough to define distance between clusters:



Linkage Criteria

Nearest neighbor (single link, MST)

$$d(C_1, C_2) = \min_{x \in C_1, y \in C_2} ||x - y||$$

Average link (more robust but expensive)

$$d(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{x \in C_1} \sum_{y \in C_2} ||x - y||$$

Nearest centroid (faster but still robust)

Furthest link (keeps clusters round)

$$d(C_1, C_2) = \max_{x \in C_1, y \in C_2} ||x - y||$$

Agglomerative Clustering Complexity

The run time for clustering is the number of merge steps (n-1) times the cost per merge.

The linkage criteria trades off between speed (O(n) to O(n²) per iteration) and robustness.

Each merge changes the cost for only 2n of up to n² cluster pairs, so we can avoid recomputation.

Advantages of Cluster Hierarchies

- Organization of clusters and subclusters
- Visualization of the clustering process
- Natural measure of distance between clusters.
- Efficient classification of new items: compare against centroids as we march down the tree, in time proportional to height.

Which Clustering Algorithm To Use?

There are an enormous number of possible clustering algorithms, but much more important decisions are:

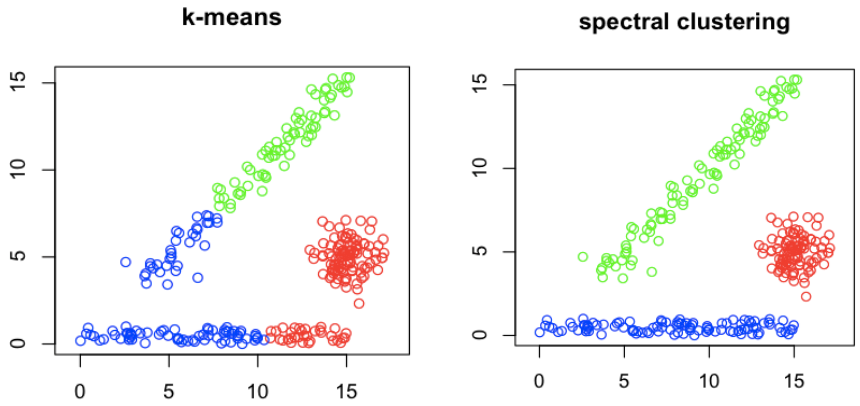
- using the right distance function
- properly normalizing your variables
- appropriately visualizing the final clusters to know whether they are good.

Seeking Connected Clusters

K-means finds centroids and spherical clusters, but not skinny or nested ones.

Single-link agglomerative clustering finds skinny clusters.

But it is easily fooled into merging two clusters by a single close point pair.



Similarity Graphs

Each entry S[i,j] in a similarity matrix S scores how much alike elements i and j are.

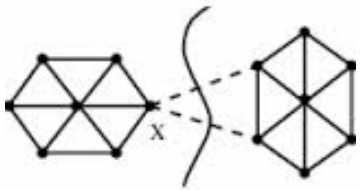
It is essentially an inverse of a distance matrix, and can be computed: $S_{ij} = \exp(-\beta||x_i - x_j||)$

Thus similarity ranges from 0 to 1.

This weighted graph could be made sparse by setting all small terms to zero.

Cuts in Graphs

Clusters in similarity graphs should have small / light edges spanning them.



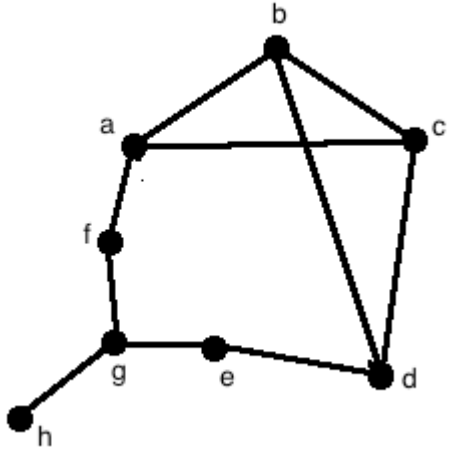
Ideally clusters will have a high weight (ie. sum of internal edges) and a small cut.

Finding Cuts in Graphs

Network flow methods can find the minimum cut in a graph, but not one whose internal weight is large.

The minimum cut will naturally tend to separate isolated vertices, not clusters.

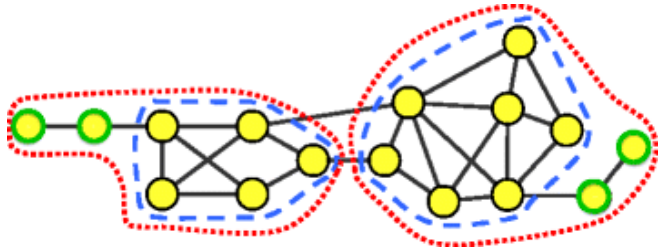
The problem of graph partitioning which seeks balanced clusters is NP-complete, motivating heuristics and other approaches.



Conductance and Eigenvectors

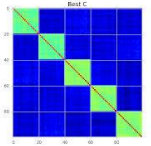
The *conductance* of a cluster *C* is defined as the weight of the cut edges over the weight of the internal edges: $\frac{W\prime(C)}{W(C)}$

Low conductance clusters are desirable.

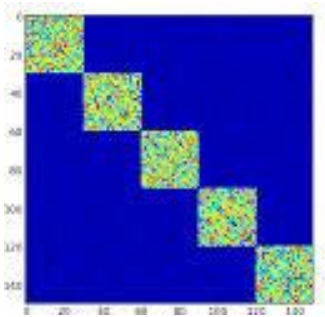


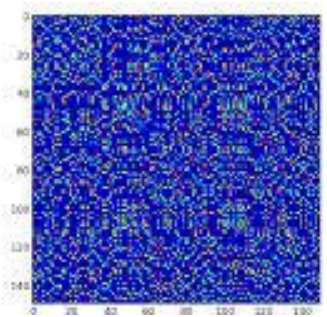
Blocky Matrices and Eigenvectors

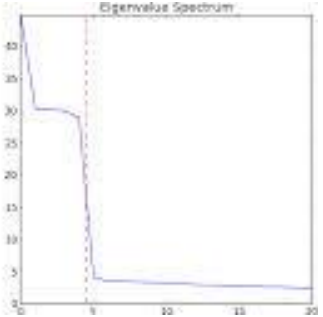
Low conductance clusters correspond to blocky similarity matrices.

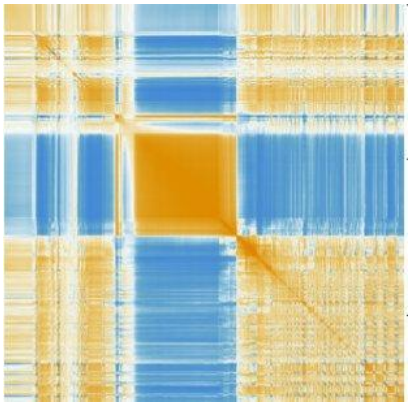


Recall that we created blocky matrices using Eigenvector decomposition.









Recall: Singular Value Decomposition

The covariance matrix can be represented by summing the spectrum of Eigenvalues / vectors.

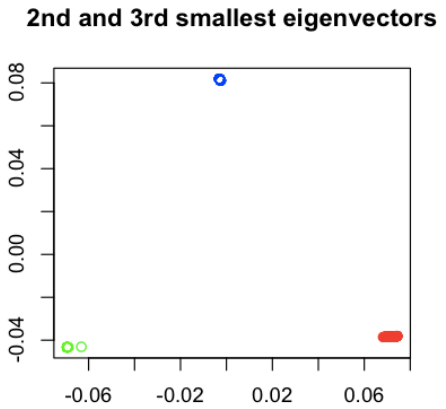
However, just using the vectors associated with the largest Eigenvalues gives a good approximation.

This dimension reduction method is very useful to produce smaller, more effective feature sets.

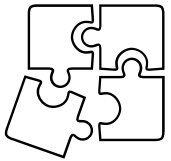
Spectral Clustering

- From similarity matrix *W* computes *L=D-W* (Laplacian) where D is the degree matrix.
- Small Eigenvectors of L define features for each element.

Performing k-means clustering on this transformed feature space often yields good clusters.



Wrapup:
Near. Neighb. Methods & Clustering



- Nearest-neighbor methods rely on the assumption that close points play similar roles
- Clustering is an unsupervised ML approach for grouping data points that are close in a way that you gain some kind of useful abstraction
- Choosing the right approach is just as important as choosing the right distance metric