CSCI E-82
Advanced Machine Learning, Data Mining & Artificial Intelligence
Lecture 6

Dynamic Time Warping
Clustering

Peter V. Henstock
Fall 2018

© 2018 Peter V. Henstock

Dynamic Time Warping

© 2018 Peter V. Henstock

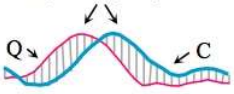
Problem with Distances

- What if we want to compare things that are different lengths?
- What if they are the same length but have intermediate states like phonemes within speech?
- What can we do?

© 2018 Peter V. Henstock

Dynamic Time Warping

- Stocks are easy to compare since start at same time and end at same time
- Speech is almost never aligned
  - Similar, but out of *phase peaks* ...



... *produce a large Euclidean distance.*

However this can be corrected by DTWs nonlinear alignment.

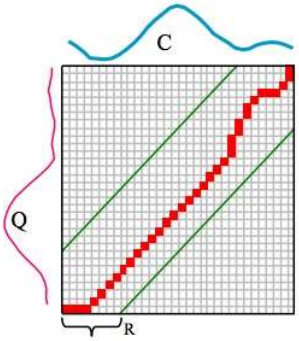


Figure 3: left) Two time series which are similar but out of phase. right) To align the sequences we construct a warping matrix, and search for the optimal warping path (red/solid squares). Note that Sakoe-Chiba Band with width R is used to constrain the warping path

http://4.bp.blogspot.com/-7QKyVmAJv2I/UIxFDTIcjMI/AAAAAAAACew/AUbn2u-rS8A/s1600/dtw2.jpg

© 2018 Peter V. Henstock

Dynamic Time Warping

- Stocks are easy to compare since start at same time and end at same time
- Bioinformatics pairwise sequence analysis is the same

Needleman-Wunsch

match = 1 mismatch = -1 gap = -1

		G	C	A	T	G	C	U
	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
T	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

https://en.wikipedia.org/wiki/Needleman%E2%80%93Wunsch_algorithm

© 2018 Peter V. Henstock

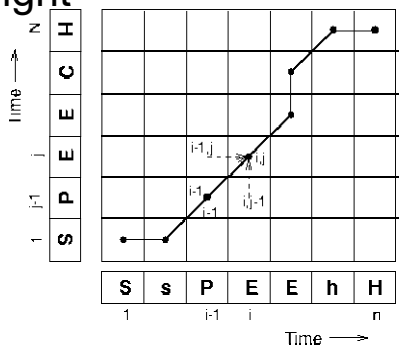
Dynamic Time Warping

- Sequences $x_1 \ldots x_n$ and $y_1 \ldots y_m$
- Construct warping matrix where i,j element is dist (x_i, y_j)
- Define cost: min {sqrt(sum w_k)}
  - w_k = matrix element of the warping path through the warping matrix
- Warping path
  - $z(i,j) = d(x_i, y_j) + \min\{z(i-1,j-1), z(i-1,j), z(i,j-1)\}$

© 2018 Peter V. Henstock

Dynamic Time Warping

- Construct matrix of distances
- Choose lowest cost path from bottom left to top right



$$z(i,j) = d(x_i,y_j) + \min\{ z(i-1,j-1), z(i-1,j), z(i,j-1) \}$$

http://www.cnel.ufl.edu/~kkale/dtw.html

© 2018 Peter V. Henstock

What are the problem with this?

1) How big will the distance matrix be for sequences of 5000 and 4000?

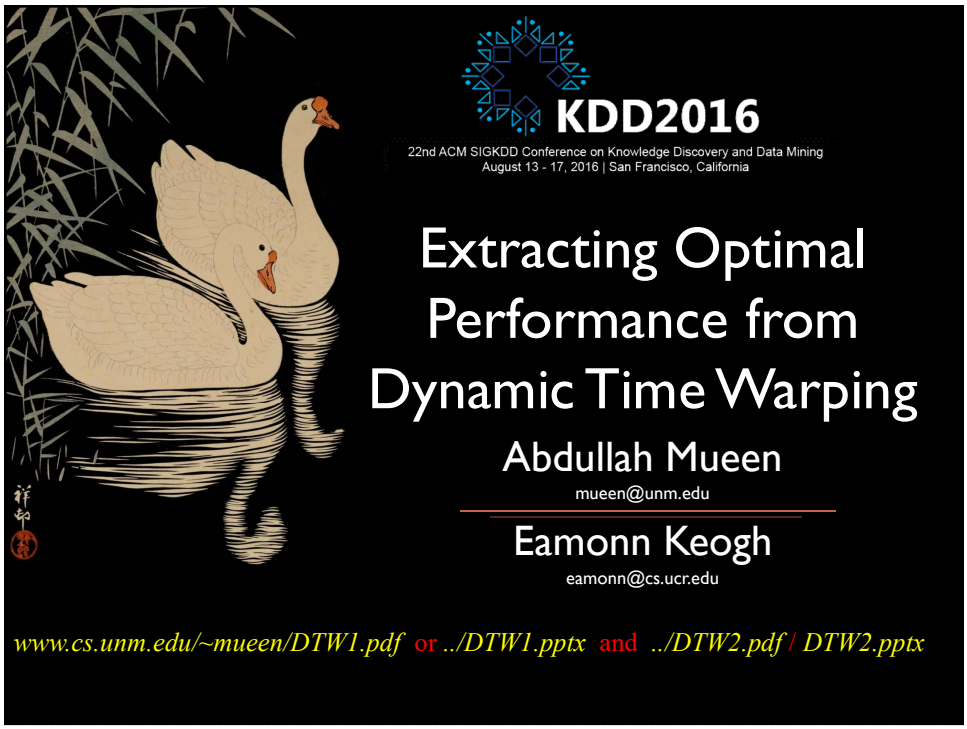
2) Assumed that they start and end at the same place. What if they don't?

© 2018 Peter V. Henstock

Dynamic Time Warping + KNN

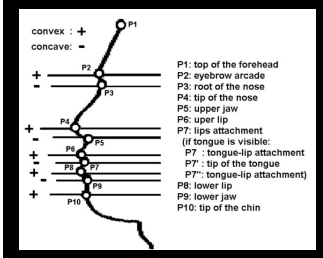
- Nearest Neighbor DTW is very hard to beat across fields
- When it loses, it barely loses but requires extreme complexity/coding
- DTW arises in many different problems

© 2018 Peter V. Henstock



An Unstated Assumption (alterative)

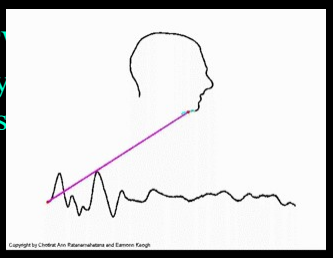
For all time series, heartbeats, gait, fish and faces, we could design and extract features, and just represent the objects as feature vectors. However, in the dozen of cases where we compared this type of approach to just using DTW on the raw data, DTW almost always wins!
So, at the very least, try the simplest ideas first, and DTW is very simple.



How many fiducials? Pantic suggests 10, Campos suggests 8, Dariush suggests 9, Liposcak suggests 12…

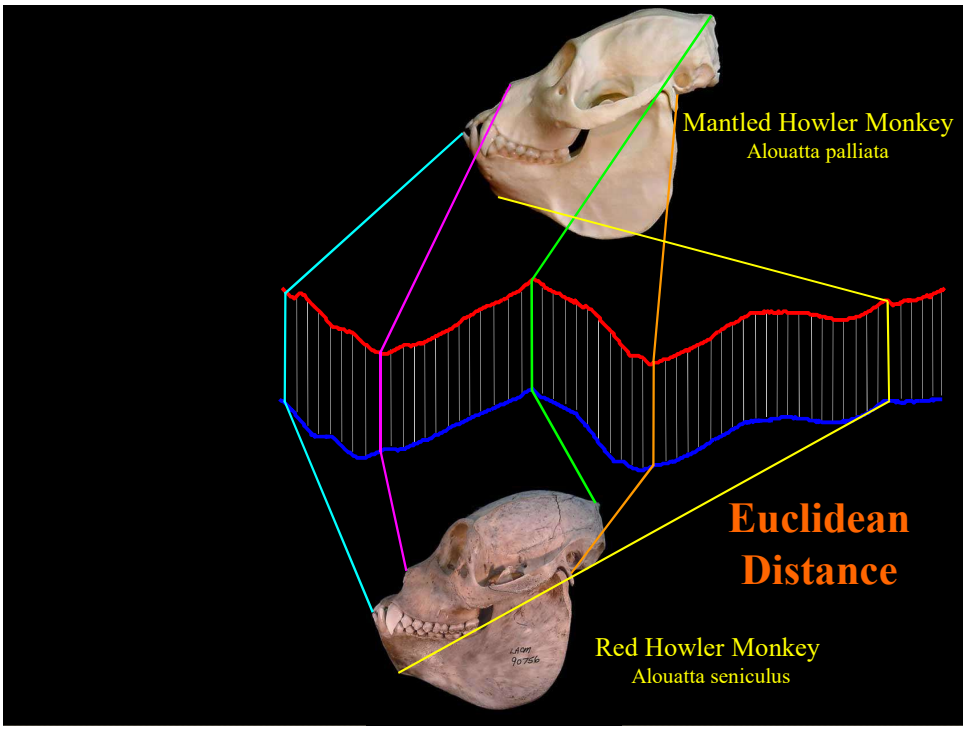
Accuracy
70 to 80%
(10 classes)

1NN DT
Accuracy
90% plus
(10 classes)



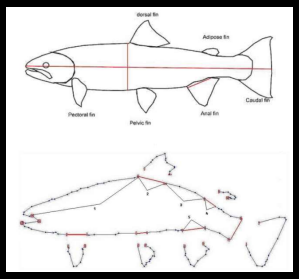
Parameter-free or parameter-lite, robust to changes of expression..

F1={7.4,1.3,2.1,1.2,4.6,



An Unstated Assumption (alterative)

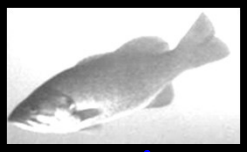
For all time series, heartbeats, gait, faces, and even fish!, we could design and extract features, and just represent the objects as feature vectors. However, in the dozen of cases where we compared this type of approach to just using DTW on the raw data, DTW almost always wins!
So, at the very least, try the simplest ideas first, and DTW is very simple.

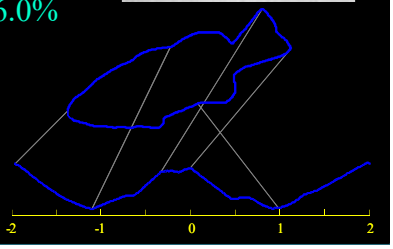


Accuracy 75.7%

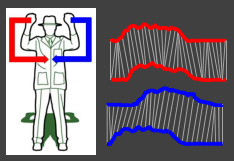
1. length between the nose and dorsal fin
2. width of the dorsal fin
3. distance between the dorsal fin and adipose fin
4. width of the adipose fin
5. width of the anal fin

1NN DTW Accuracy 86.0%



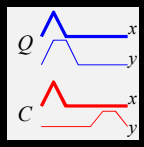


Generalizing to Multi-Dimensional Data 1 of 4



It is increasingly common to encounter Multi-Dimensional (MD) time series data. Here we measure the X-axis acceleration of both the left and right hand.

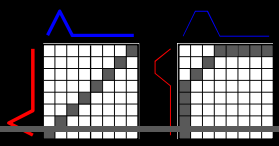
Given these pair of 2D objects…



There are two obvious ways to compute the MD DTW score.

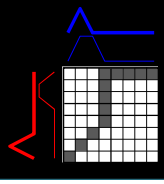
Independent: Just compute the DTW score for each dimension independently, and sum up each score.

$$DTW_I(Q,C) = DTW(Q_x,C_x) + DTW(Q_y,C_y) = \mathbf{2.4}$$



Dependent: Create a single distance matrix that reflect the distance between each corresponding pair of time series, then find the *single* warping path and distance as per normal.

$$DTW_D(Q,C) = DTW(\{Q_x,Q_y\},\{C_x,C_y\}) = \mathbf{3.2}$$



Generalizing to Multi-Dimensional Data 4 of 4

Critical Point: You *can* generalize DTW to 2,3,4,…1,000 dimensions.

However, it is *very* unlikely that more than 2 to 4 is useful, after that, you are almost certainly condemned to the curse of dimensionality.

Consider a physical activity dataset containing 36 axis synchronous measurements from three Inertial Measurement Units (IMUs) located on the wrist, chest and ankle. This dataset has eight subjects performing activities such as: `rope-jumping`, `running`, `folding laundry`, `ascending-stairs`.

PAMAP, Physical Activity Monitoring for Aging People www.pamap.org/

(we used DTW with $w = 0$ for simplicity here, we could do better by tuning w)

Which Dimensions?	Accuracy
All dimensions	0.19
A single random dimension (on average)	0.51
The best single dimension (as predicted by CV)	0.72
The best 3 dimensions (as predicted by CV)	0.89

Using *all* dimensions is a disaster!

Using the best three is a lot better than using the best one, so there is evidence that Multi-Dimensional DTW really does help.

Clustering

© 2018 Peter V. Henstock

Clustering Approaches

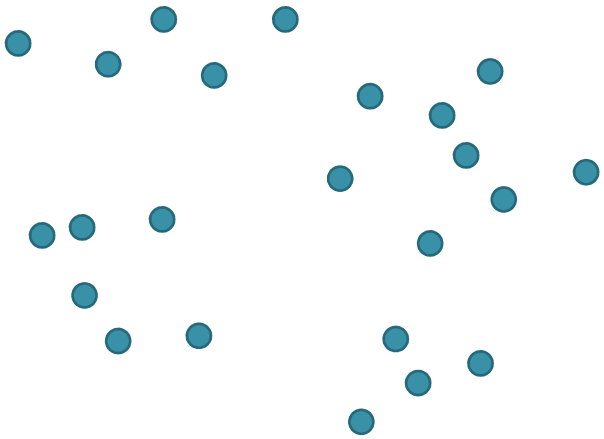
- Partitioning: non-overlapping subsets
- Hierarchical
- Density-based: DBSCAN
- Grid-based: CLIQUE
- (Correlation)
- Spectral
- (Gravitational)
- (Hard)

© 2018 Peter V. Henstock

K-Means

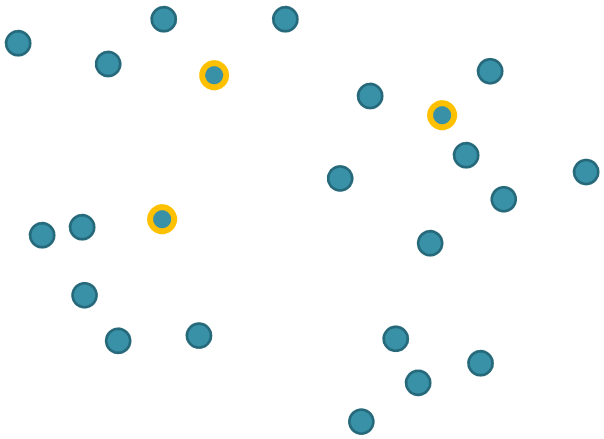
© 2018 Peter V. Henstock

Example: Do K=3 K-means



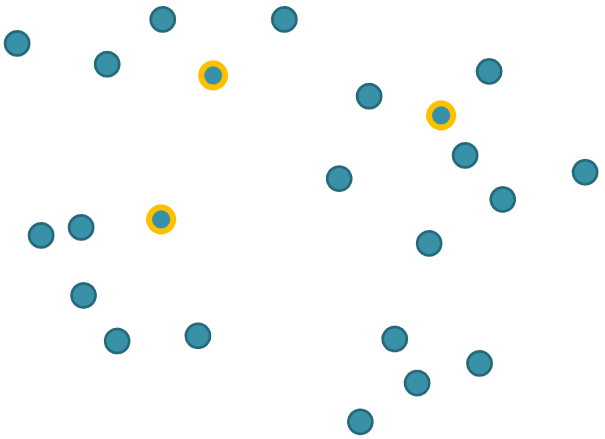
© 2018 Peter V. Henstock

Choose any 3 cluster centroids



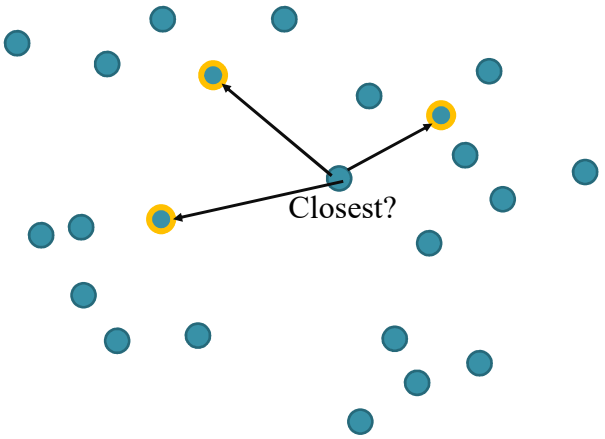
© 2018 Peter V. Henstock

Compute closest centroid for every blue data point



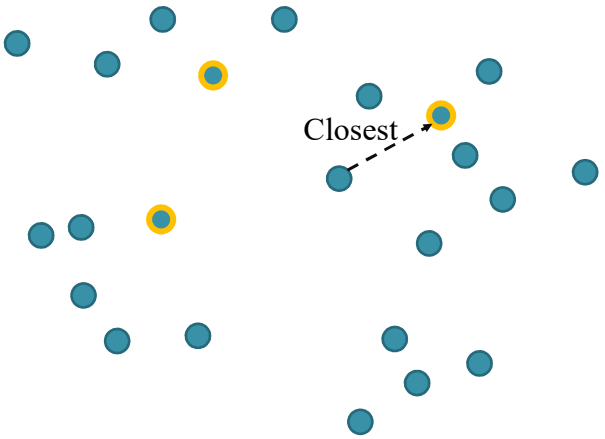
© 2018 Peter V. Henstock

Compute closest centroid for every blue data point



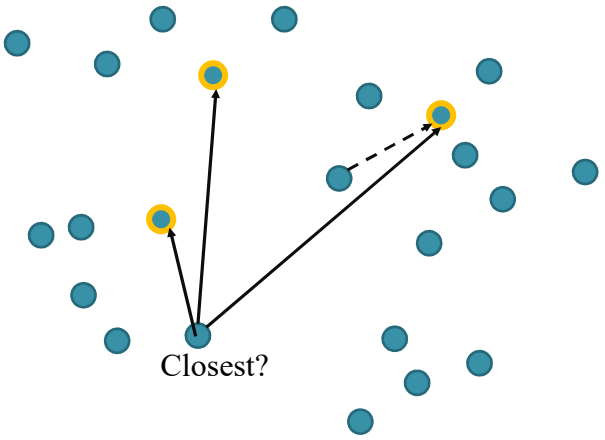
© 2018 Peter V. Henstock

Compute closest centroid for every blue data point



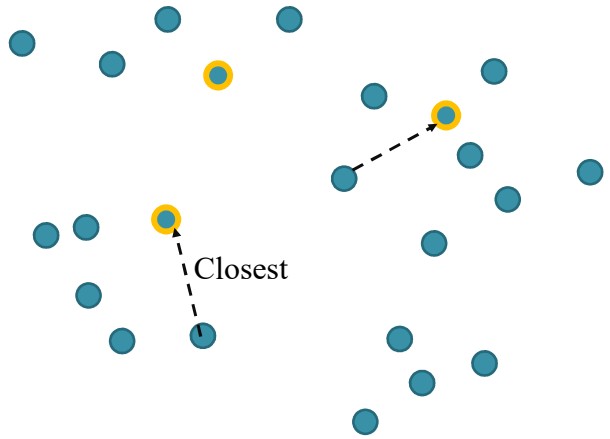
© 2018 Peter V. Henstock

Next point: which is closest centroid?



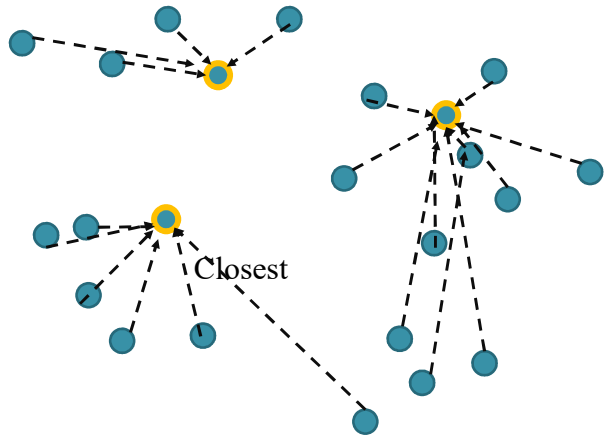
© 2018 Peter V. Henstock

Next point: which is closest centroid?



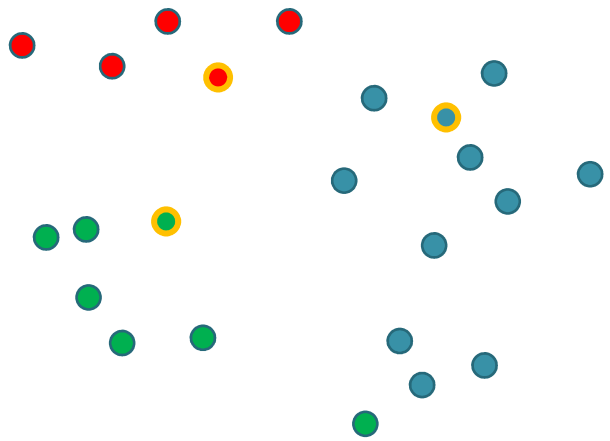
© 2018 Peter V. Henstock

Next point: Repeat for remaining points



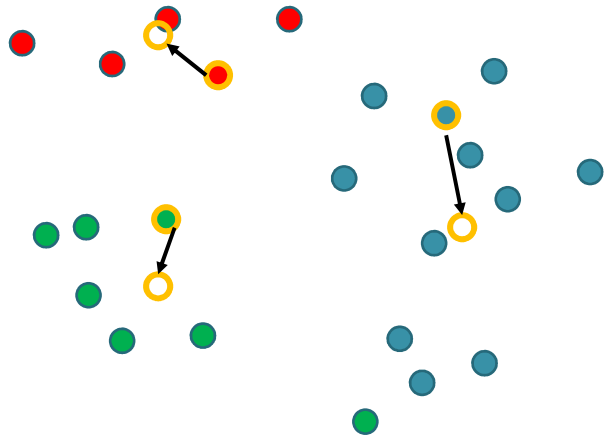
© 2018 Peter V. Henstock

We've assigned points to closest centroid



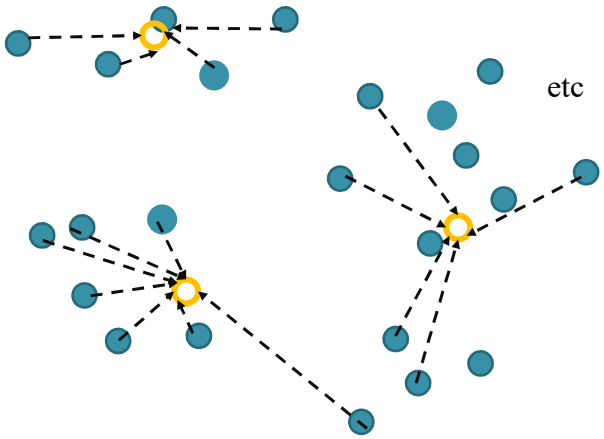
© 2018 Peter V. Henstock

K-means moves the centroids to the centroid (i.e. center) of the points they represent



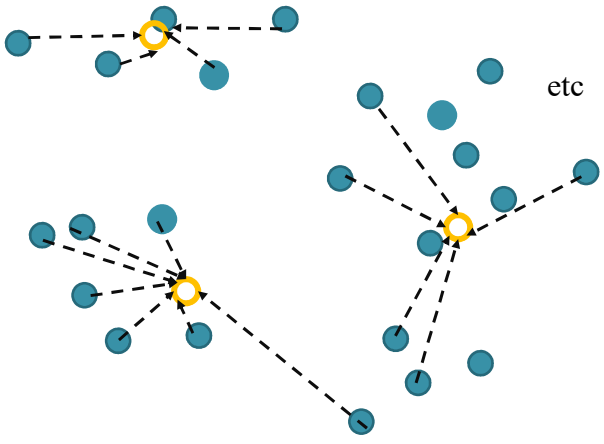
© 2018 Peter V. Henstock

Repeat the algorithm again assigning closes distances



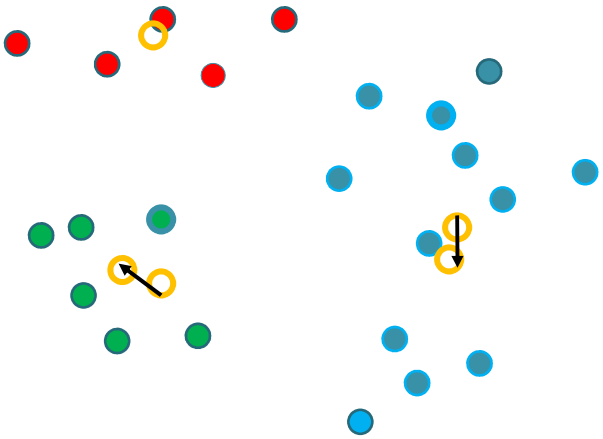
© 2018 Peter V. Henstock

Assign the clusters accordingly



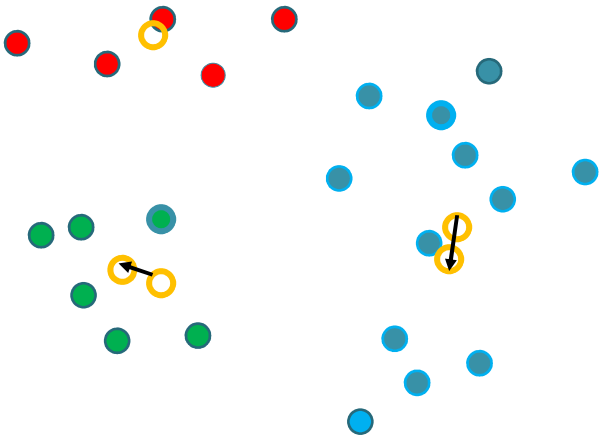
© 2018 Peter V. Henstock

Shift the centroids accordingly

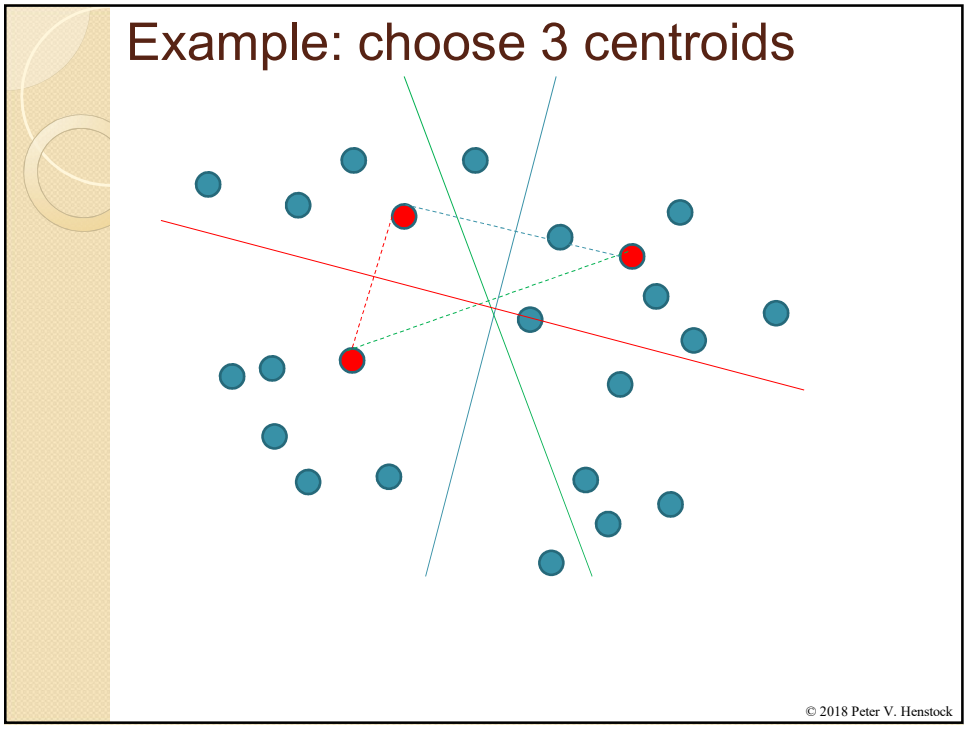


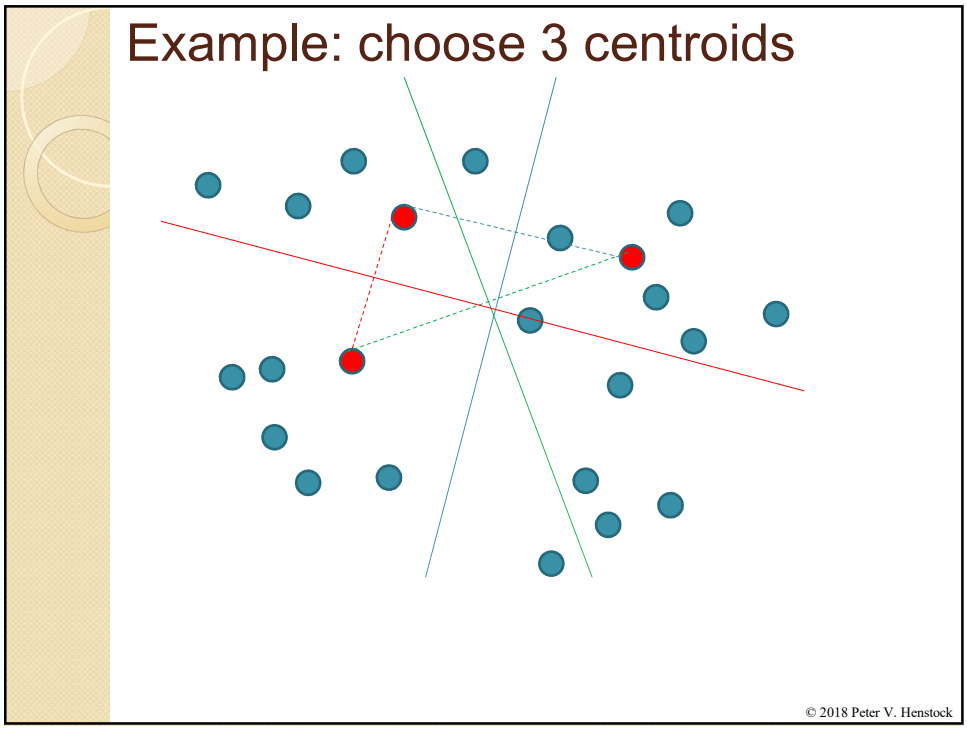
© 2018 Peter V. Henstock

We're probably done since centroids will probably not change further



© 2018 Peter V. Henstock


Example: choose 3 centroids
© 2018 Peter V. Henstock


Example: choose 3 centroids
© 2018 Peter V. Henstock

Voronoi Diagram (Wikipedia)

© 2018 Peter V. Henstock

Optimization

- Machine learning is all about optimization so…
- What is the optimization criteria for K-Means?

© 2018 Peter V. Henstock

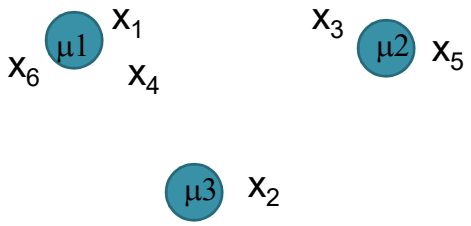
K-Means Clustering

- Data vector $x_1..x_N$ (with M features)
- Clusters $\{\mu_1 \ldots \mu_K\}$

- Which x map to which cluster?
  - $\{c_1 \ldots c_N\}$ is the cluster label for each $x_1..x_N$
    - Encode vector where 3 indicates cluster 3
  - Alternative encoding for cluster labels is "one hot coding" where
    - K vectors with 1 only when it applies
  - r_j = responsibility vector
    - r_{ij} = 1 if x_i maps to cluster j else 0

© 2018 Peter V. Henstock

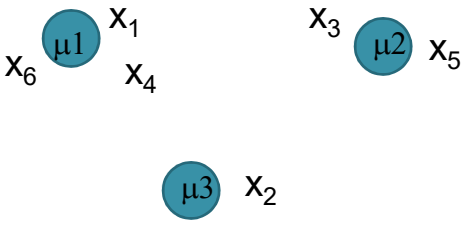
Coding Representation



- C coding: [? ? ? ? ? ?]

© 2018 Peter V. Henstock

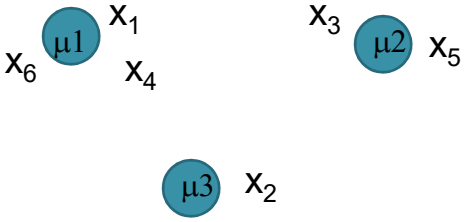
Coding Representation



- C coding: [1 3 2 1 2 1]
- One hot coding with r
  - r1: [?]
  - r2: [?]
  - r3: [?]

© 2018 Peter V. Henstock

Coding Representation



- C coding: [1 3 2 1 2 1]'
- One hot coding with r
  - r1: [1 0 0 1 0 1]'
  - r2: [0 0 1 0 1 0]'
  - r3: [0 1 0 0 0 0]'

© 2018 Peter V. Henstock

Cost (or Loss) Function

- $J_{xi}(r, \mu) = \Sigma_k \, r_{ik} \, [x_i - \mu_k]^2$
- This describes how far away point x_i is from the particular cluster to which is assigned
  - How many non-zero terms are in the sum?

© 2018 Peter V. Henstock

Cost (or Loss) Function

- $J_{xi}(r, \mu) = \Sigma_k \, r_{ik} \, [x_i - \mu_k]^2$
- This describes how far away point x_i is from the particular cluster to which is assigned
  - How many non-zero terms are in the sum?

© 2018 Peter V. Henstock

Cost (or Loss) Function

- $J_{xi}(r, \mu) = \Sigma_k \, r_{ik} \, [x_i - \mu_k]^2$
- This describes how far away point x_i is from the particular cluster to which is assigned
  - How many non-zero terms are in the sum?

- $J(\text{all } r, \mu 1 .. \mu \kappa) = \Sigma_i \, \Sigma_k \, r_{ik} \, [x_i - \mu_k]^2$
  - Perform this for all points xi

© 2018 Peter V. Henstock

Optimization

- Similar case to regression where we set up a cost function
  - Solved it using a gradient search
  - Solved it analytically

- $J(\text{all } r, \mu 1 .. \mu \kappa) = \Sigma_i \, \Sigma_k \, r_{ik} \, [x_i - \mu_k]^2$

- Formally called the "K-Means Problem"
- Only problem is that it is NP-hard

© 2018 Peter V. Henstock

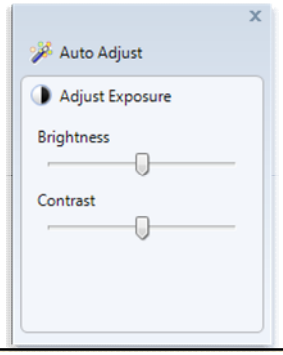
Lloyd's Algorithm

- Computationally feasible solution to solve the K-Means problem
- Coordinate descent

© 2018 Peter V. Henstock

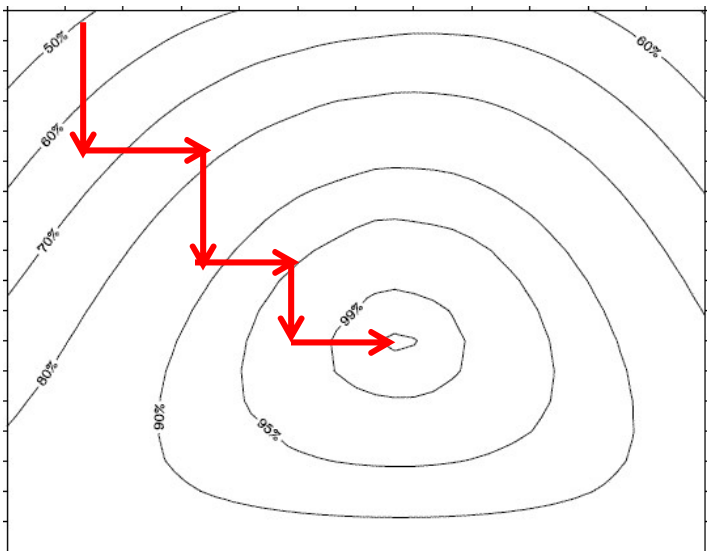
Coordinate Descent

- Gradient descent took the steepest descent across multiple variables at one time using the slope
- Coordinate descent optimizes one variable at a time



© 2018 Peter V. Henstock

Coordinate Descent



© 2018 Peter V. Henstock

Lloyd's Algorithm

- Computationally feasible solution to solve the K-Means problem
- Coordinate descent
- Optimize the r's and the μ's

© 2018 Peter V. Henstock

Two Steps of Lloyd's Algorithm

1) For each x_i, set $r_{ik} = \text{argmin}_k [x_i - \mu_k]^2$
   - Cluster assignment step

2) Optimize the centroids μ
   - Compute Gradient$(J_{xi}(r, \mu))_{\mu k}$ and set to 0
   - = Gradient $\Sigma_i r_{ik} [x_i - \mu_k]^2$
   - $-2 \Sigma_i r_{ik} [x_i - \mu_k] = 0$
   - $\Sigma_i r_{ik} x_i = \Sigma_k r_{ik} \mu_k$
   - $\Sigma_i r_{ik} x_i = N_k \mu_k$ where N_k = #pts in k
   - $\mu_k = \Sigma_i r_{ik} x_i / N_k$
   - μ_k = average of x's assigned to cluster k

© 2018 Peter V. Henstock

K-Means Clustering

- Requires the number of clusters k
- Takes a set of input vectors $x_1..x_N$
- Take k unique arbitrary input vectors as initial centroids $\mu_1..\mu_k$
- Loop {
  - Assign each input to the closest centroid
  - Shift centroid to center of points it represents

  }

© 2018 Peter V. Henstock

Critical aspects of K-Means?

- Choice of K
- Initial conditions
- Distance function (usually Euclidian)

© 2018 Peter V. Henstock

Computation & Initialization

- What is the big O( ) for K-Means?

© 2018 Peter V. Henstock

Computation & Initialization

- What is the big O( ) for K-Means?
  - O(#loops [kN dist calc + N re-centering])
  - O(#loops k N #features)
- Is it a fast method or a slow method?

© 2018 Peter V. Henstock

Computation & Initialization

- What is the big O( ) for K-Means?
  - O(#loops [kN dist calc + N recentering])
  - Assume k << N
  - O(loop kN)
- Is it a fast method or a slow method?

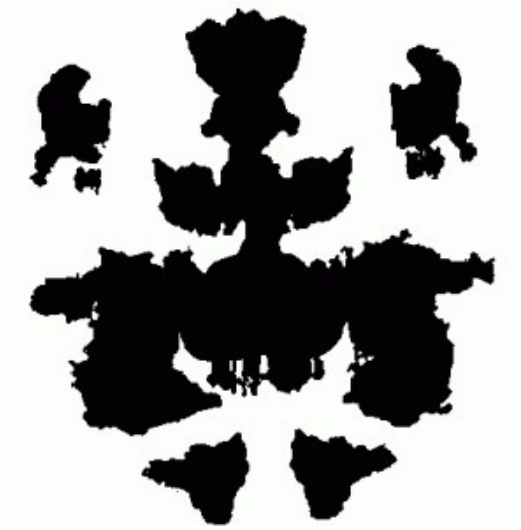
© 2018 Peter V. Henstock

Computation & Initialization

- What is the big O( ) for K-Means?
  - O(#loops [kN dist calc + N recentering])
  - Assume k << N
  - O(loop kN)
- Is it a fast method or a slow method?
  - Pretty fast
  - Can consider multiple initializations

© 2018 Peter V. Henstock

How many clusters?

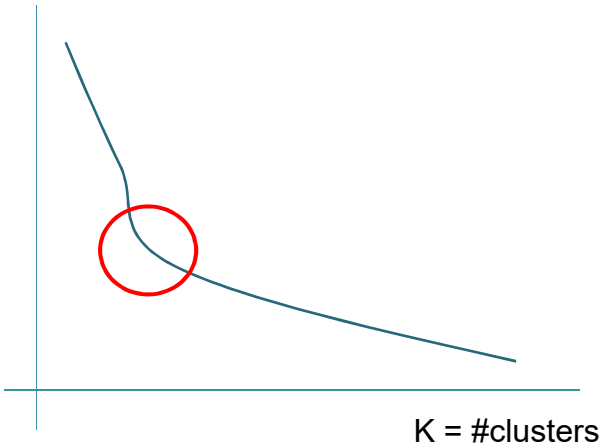


- Easy case when only 2D
- http://www.obeyclothing.com/blog/?p=3901

© 2018 Peter V. Henstock

Choosing Clusters

J = Cost Function



- Best choice is using prior knowledge

© 2018 Peter V. Henstock

Issues With K-Means

Entry	x1	x2	x3	x4	x5	x6
data1	2.3	1.5	3.5	4.9	6.6	9.4
data2	2	1	3	4	7	9
data3	1	7	3	5	6	10
data4	4	2	0	4	NA	8

© 2018 Peter V. Henstock

Issues With K-Means

Name	Height (feet)	Weight (lbs)	GPA
joe	5.7	160	3.5
fred	6.1	180	3.9
david	5.9	192	3.0
mike	5.95	210	4.0

© 2018 Peter V. Henstock

Issues With K-Means

Name	Race	Major	grade
joe	white	IT	A
fred	asian	IT	A
david	hispanic	Econ	B
mike	hispanic	IT	A

© 2018 Peter V. Henstock

K-Medoid Variation

- Medoid = represents object of a data set
- K-Means
  - Mean value of each parameter within cluster
- K-Mode
  - Mode value of each parameter
- K-Medoid
  - Best representative of a cluster that is also a data point

© 2018 Peter V. Henstock

K-Means

- Good
- Bad
- Ugly

© 2018 Peter V. Henstock

K-Means

- Good
  - Fast
  - Easy to interpret
- Bad
  - Need to predict the structure with K
  - Not optimal
- Ugly
  - Strongly depends on initial K
  - Varies with distance function approach
  - Creates spherical/circular clusters

© 2018 Peter V. Henstock

K-Means++ is which of the following?

a) K-means algorithm implemented in the C++ language for speed
b) K-means++ > K-means+ > K-means-- but you only use the latest: K-means+++
c) K-means for large data that increments values at each iteration
d) K-means with a probabilistic spacing of the initial centroids
e) K-means with improved cost function so it will reach the global minimum

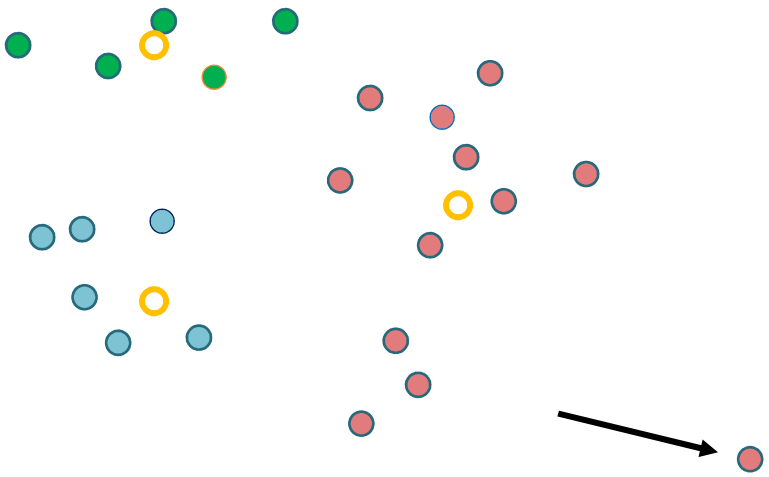
© 2018 Peter V. Henstock

K-Means++

- Initialization is different
  - Randomly choose one point as first centroid
  - Repeat until K centroids chosen
    - Compute D(x) for all points
    - $D(x_i)$ = distance of x_i to closest centroid so far
    - Choose new centroid with probability $\frac{D(x)^2}{\sum D(x)^2}$
- Use K-Means algorithm for the rest
- What points does "D^2 weighting" favor?

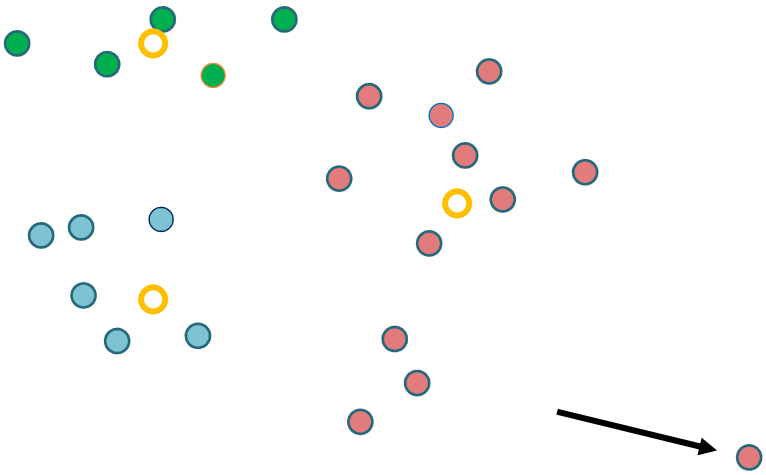
© 2018 Peter V. Henstock

K-Medians



© 2018 Peter V. Henstock

K-Means → what is K-Median?



- Which guarantees 'center' is a member of the data set?

© 2018 Peter V. Henstock

K-Medoid Variation

- Medoid
  - = representative object of a data set
  - = member of data set closest to middle
- K-Means
  - Mean value of each parameter within a cluster
- K-Mode
  - Mode value of each parameter
- K-Medoid
  - Best representative of a cluster that is also a data point

© 2018 Peter V. Henstock

Leader Clustering

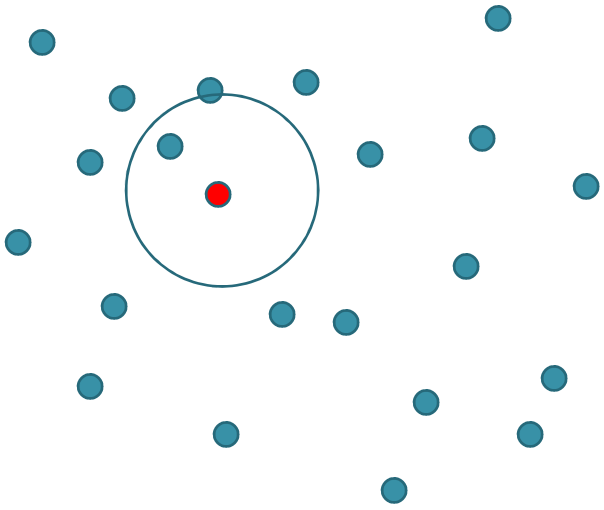
© 2018 Peter V. Henstock

Leader Clustering

- ClusterCenters = {}
- For each point x_i in randomly ordered data set
  - closestDist = dist(closestCluster, x_i)
  - If closestDist < threshold
    - Assign x_i as a member of closestCluster
  - Else {
    - Add x_i to ClusterCenters

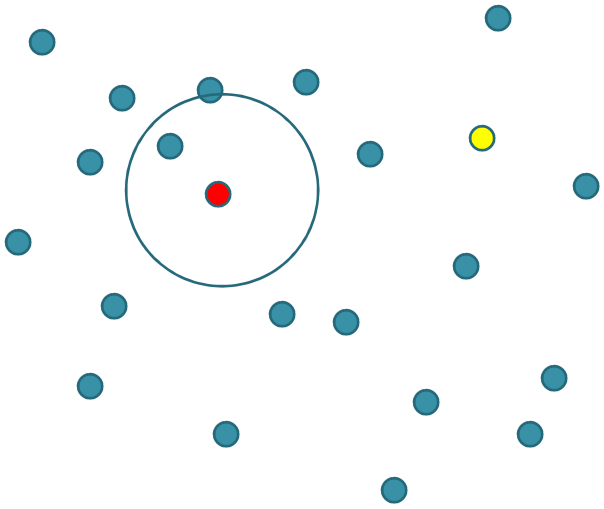
© 2018 Peter V. Henstock

Example Leader



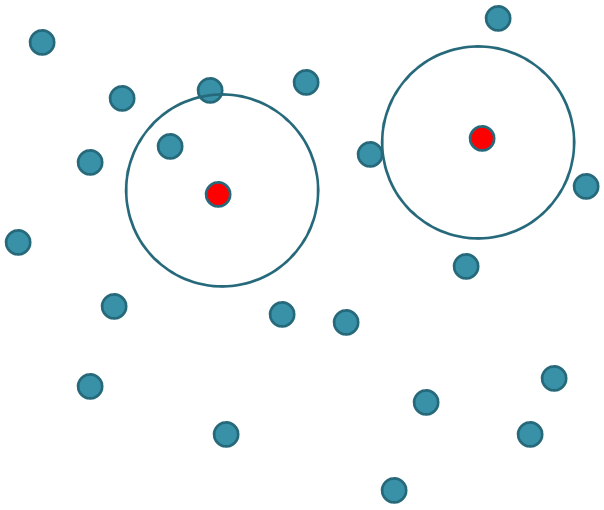
© 2018 Peter V. Henstock

Example Leader



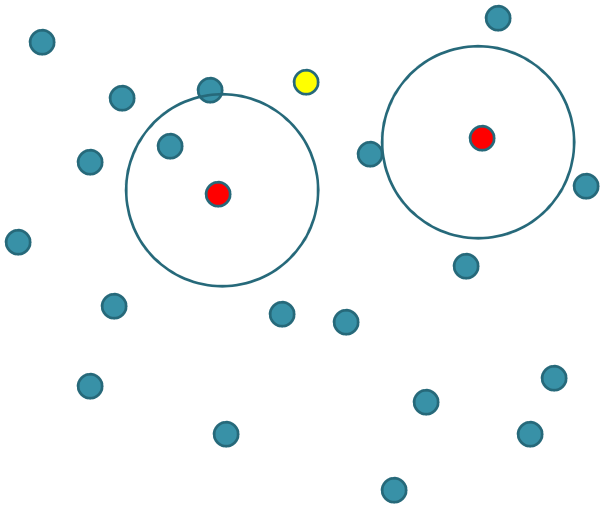
© 2018 Peter V. Henstock

Example Leader



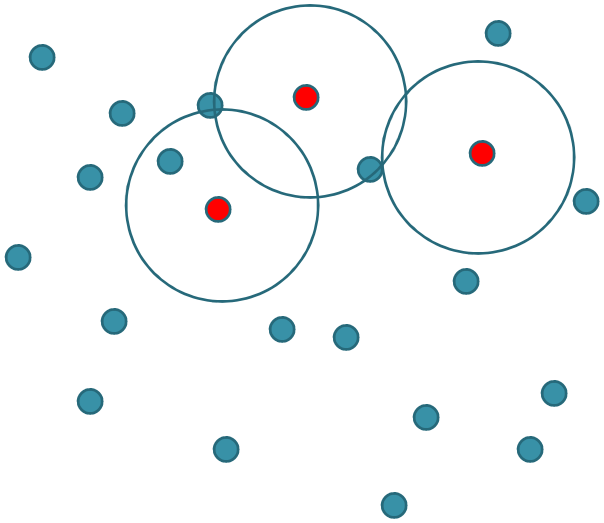
© 2018 Peter V. Henstock

Example Leader



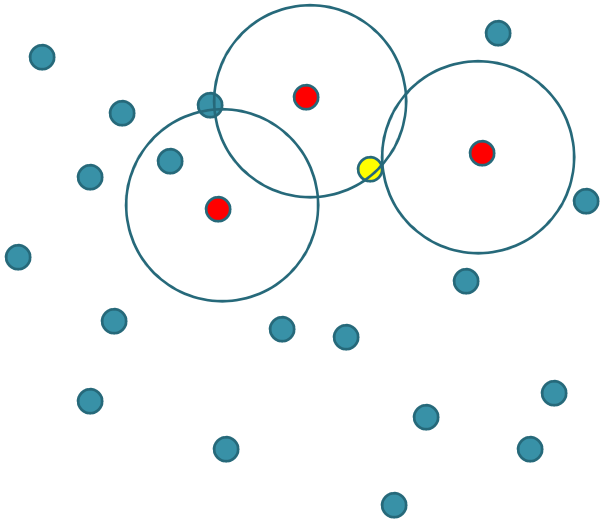
© 2018 Peter V. Henstock

Example Leader



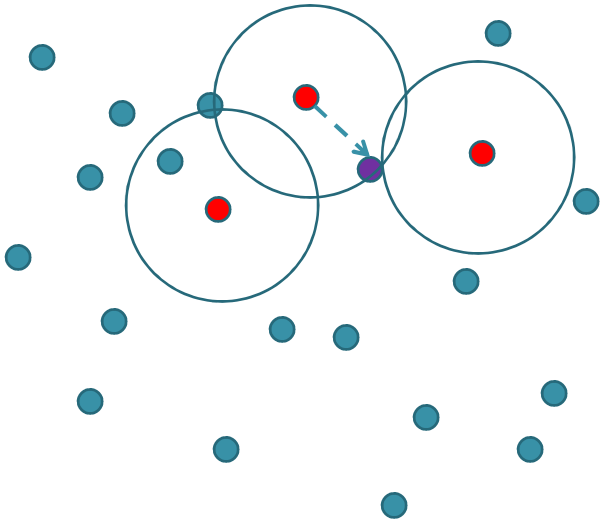
© 2018 Peter V. Henstock

Example Leader



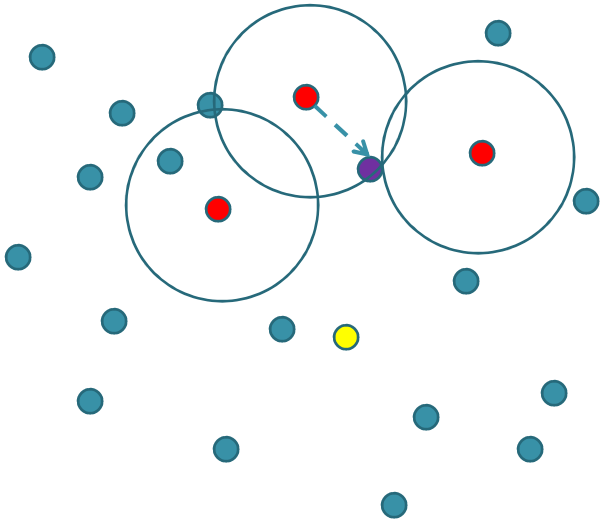
© 2018 Peter V. Henstock

Example Leader



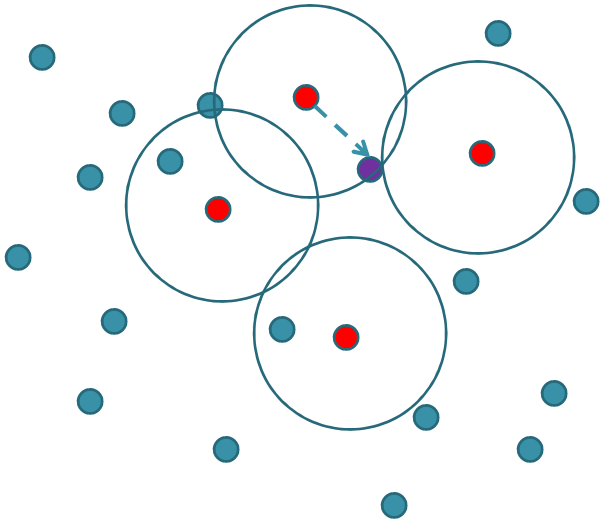
© 2018 Peter V. Henstock

Example Leader



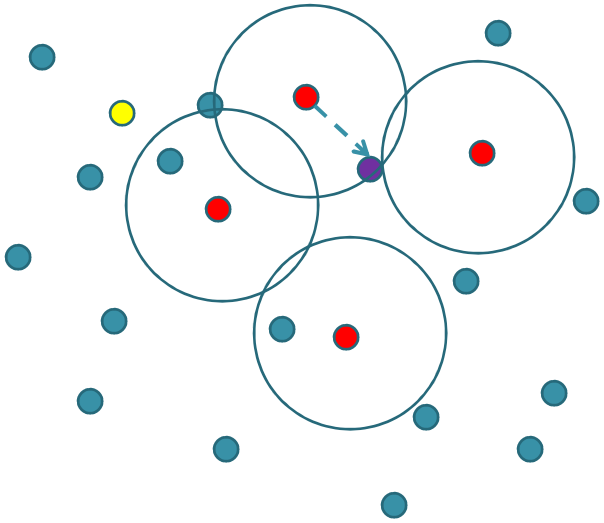
© 2018 Peter V. Henstock

Example Leader



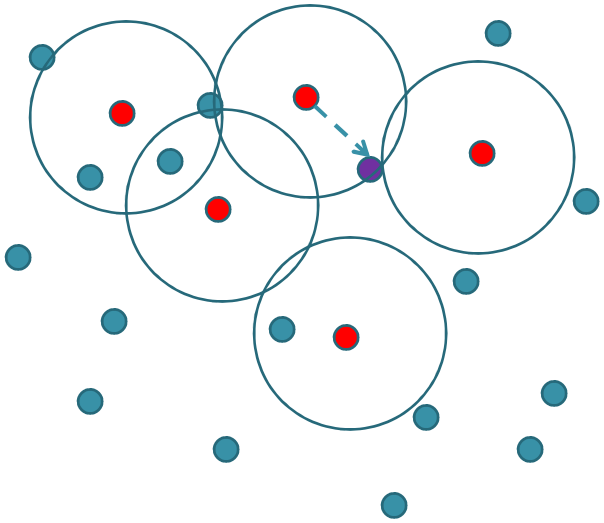
© 2018 Peter V. Henstock

Example Leader



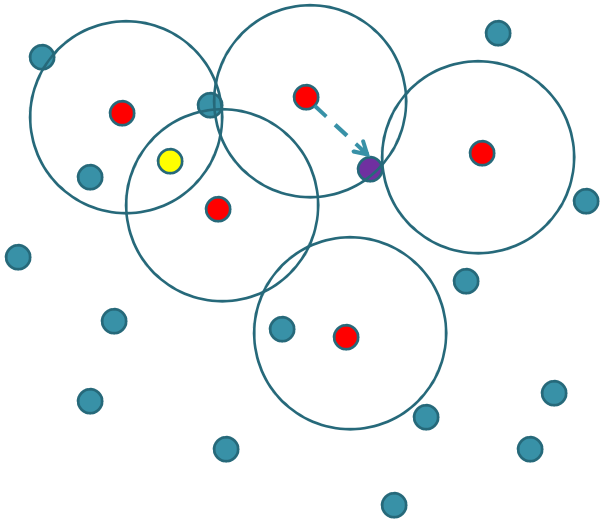
© 2018 Peter V. Henstock

Example Leader



© 2018 Peter V. Henstock

Example Leader



© 2018 Peter V. Henstock

K-Means & Leader

- What do they have in common?
- Which is better?
- Which is faster?

© 2018 Peter V. Henstock

Space & Time

- How much time?
- How much memory?

© 2018 Peter V. Henstock

OptiSim

© 2018 Peter V. Henstock

How to get a diverse set

- K-means works poorly if the initial values are too close together
- How would you go about approaching the problem?

© 2018 Peter V. Henstock

OptiSim

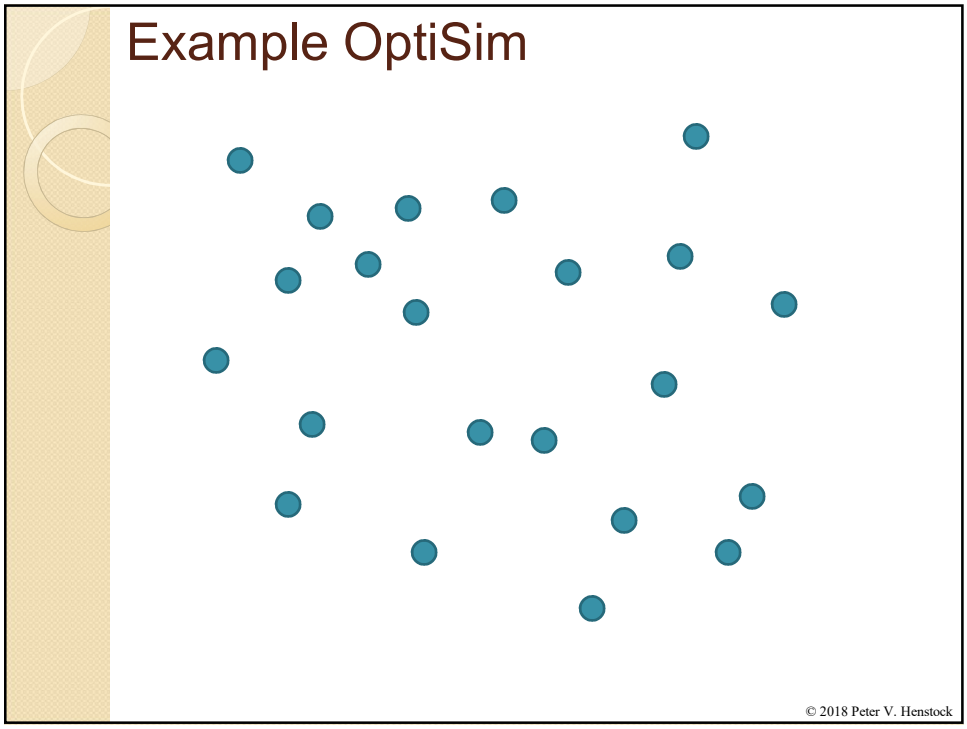
- Clark 1997, Clark & Langton 1998
- Randomly order data points
- Take 1st data point → {selection}
- SampleSet ← {}
- Repeat until obtained "enough" samples
  - dataPoint ← next data point
  - If dist(dataPoint, sample in selection) > Thr
    - Add dataPoint to SampleSet
    - if |SampleSet| == K
      - Add most dissimilar point to {selection}
      - SampleSet ← {}

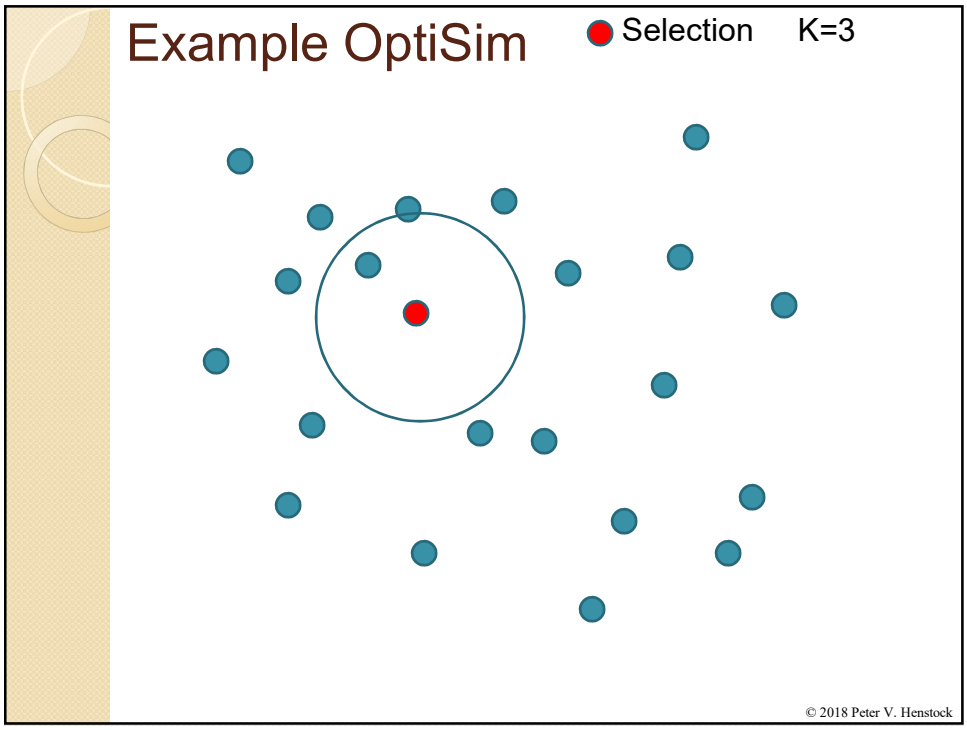
© 2018 Peter V. Henstock

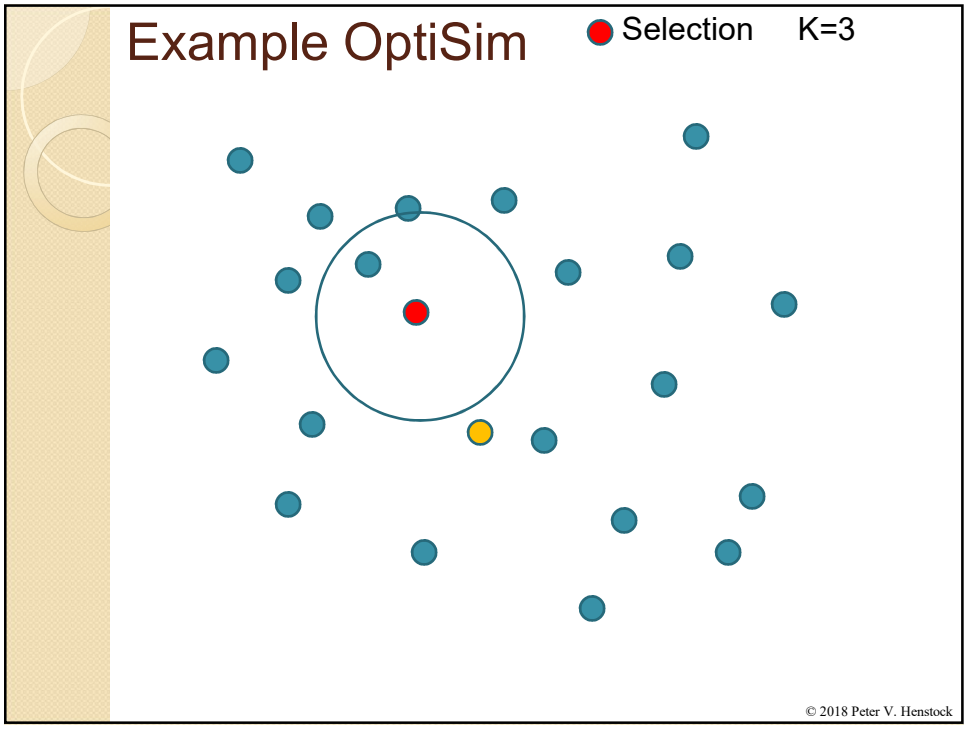
OptiSim Explained

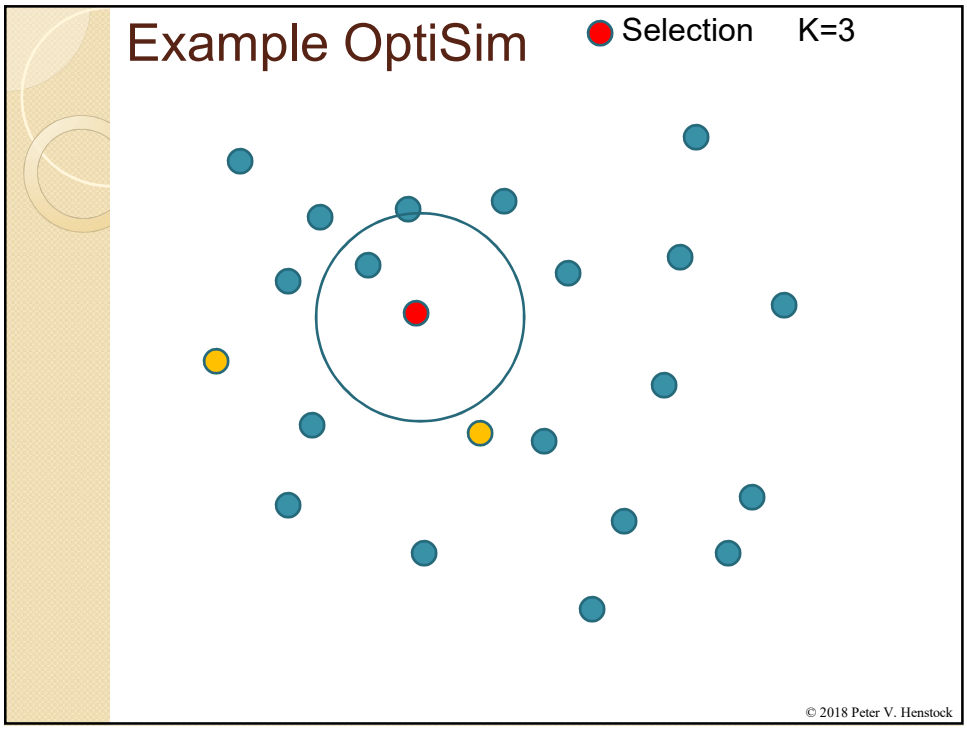
- All samples are ≥ threshold apart
  - Throw out all the close samples
- Sample set is essentially a random sample from the data set
  - Avoids extreme values that are rare
  - Chooses data representative of the data set
- Optimizing from sample set by choosing most distant point

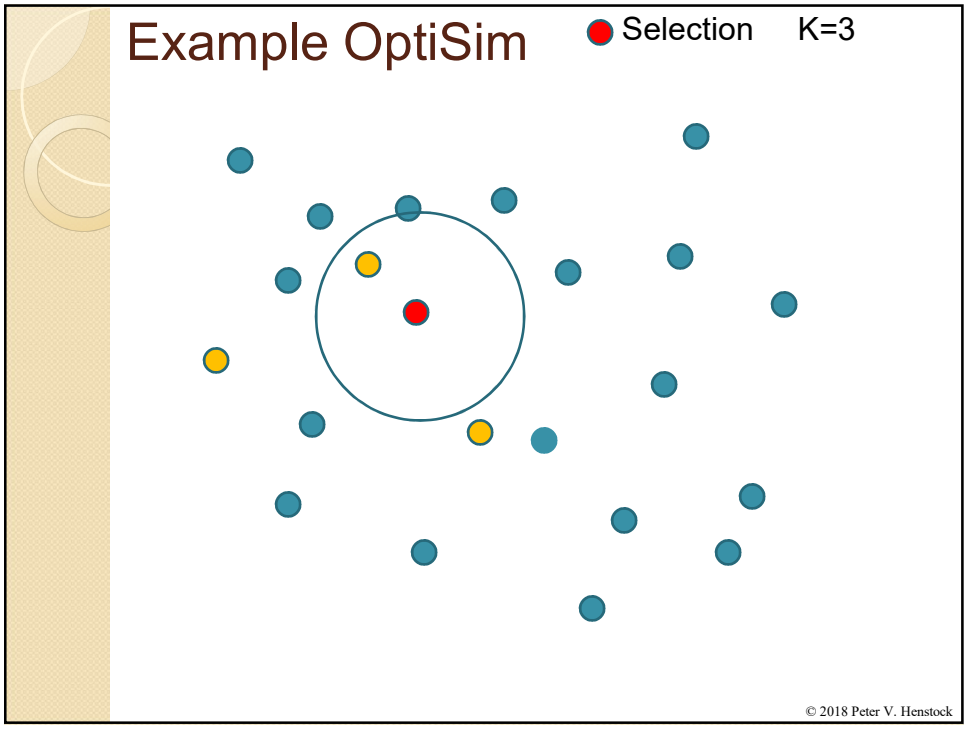
© 2018 Peter V. Henstock

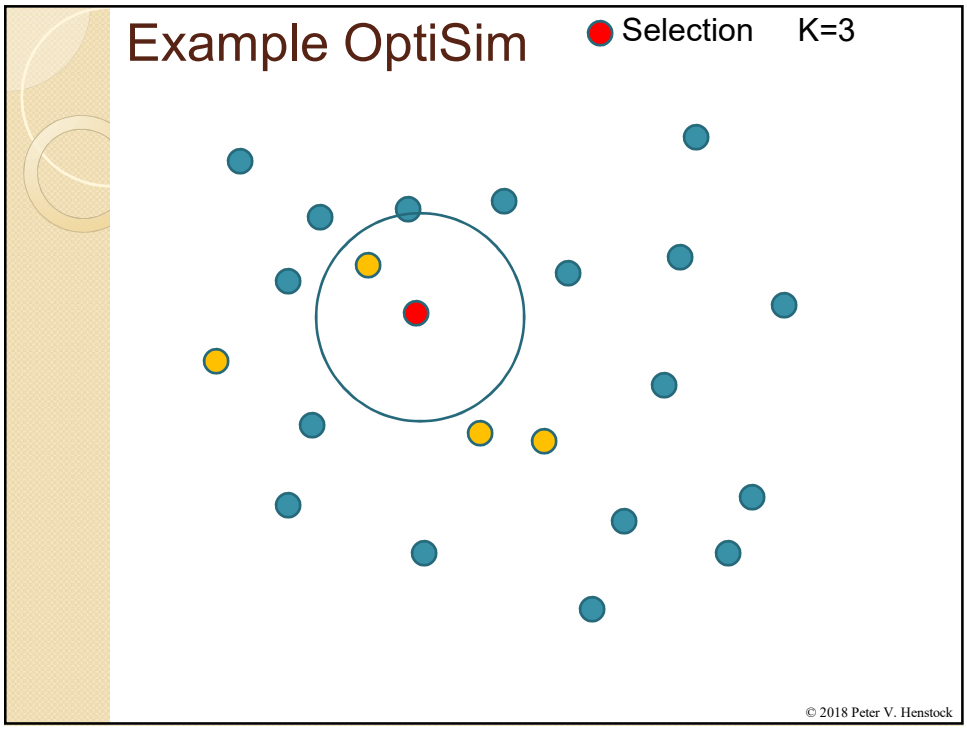

Example OptiSim
© 2018 Peter V. Henstock

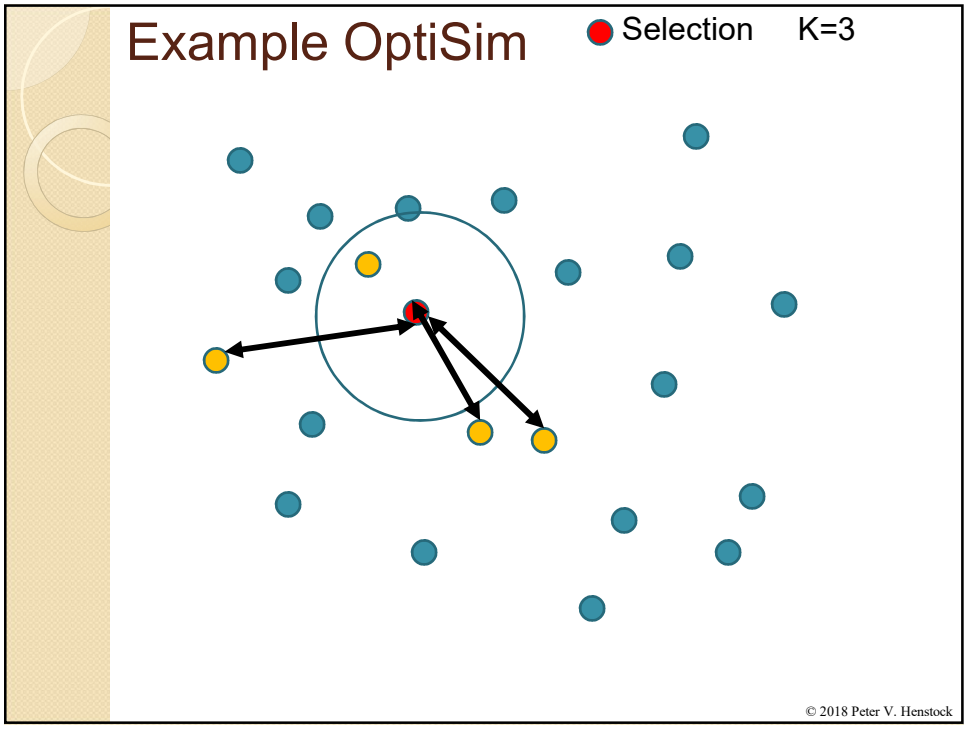

Example OptiSim
Selection
K=3
© 2018 Peter V. Henstock

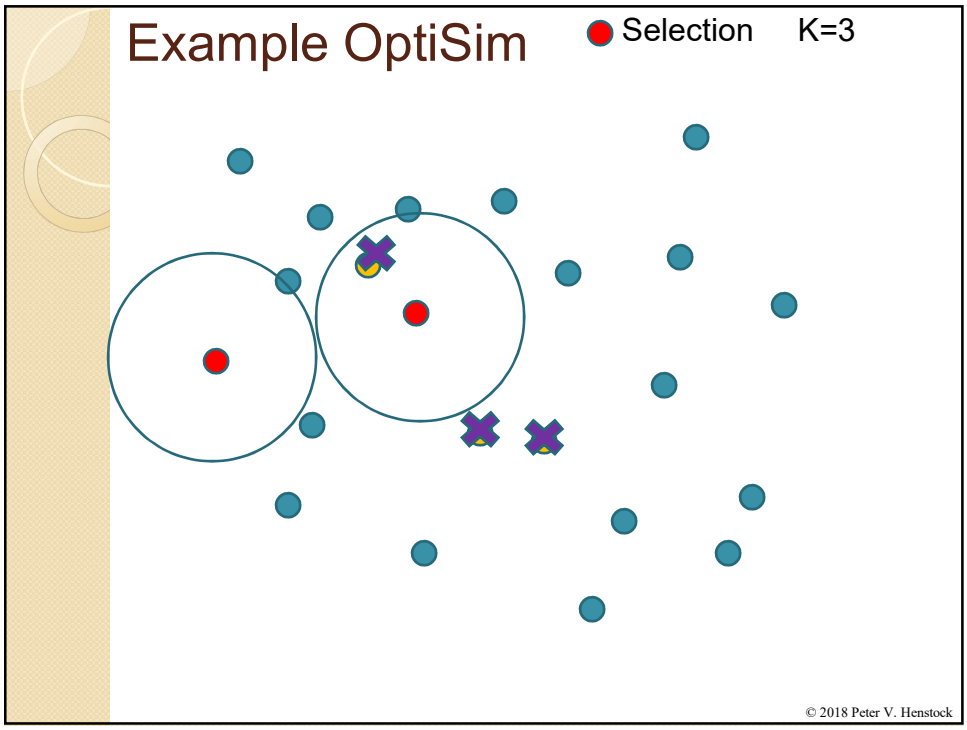

Example OptiSim
Selection
K=3
© 2018 Peter V. Henstock

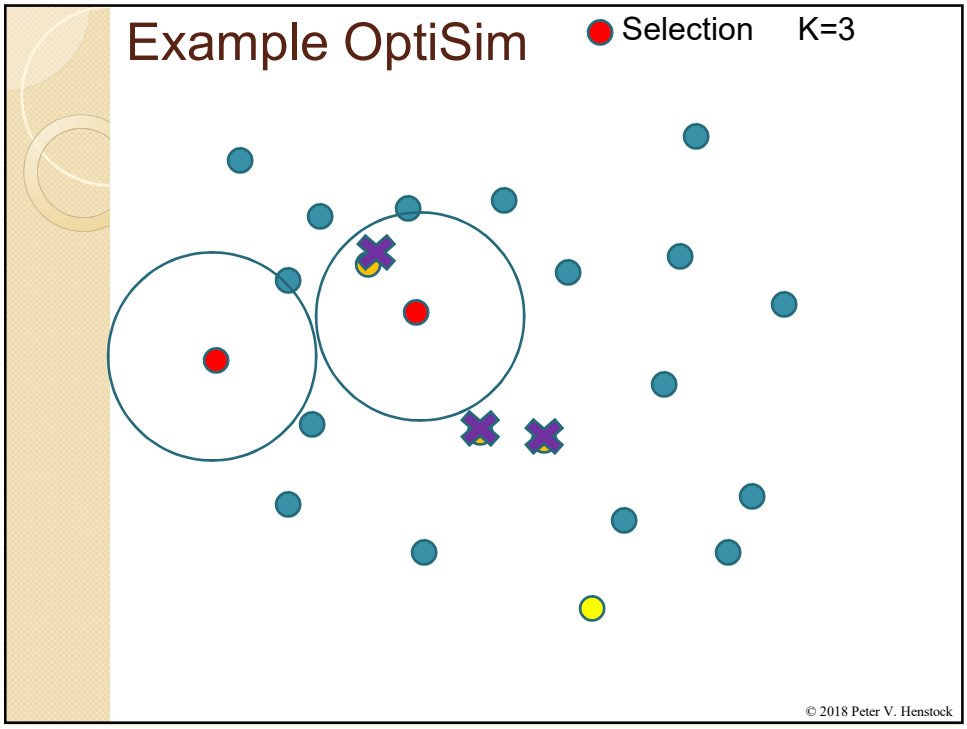

Example OptiSim
Selection
K=3
© 2018 Peter V. Henstock

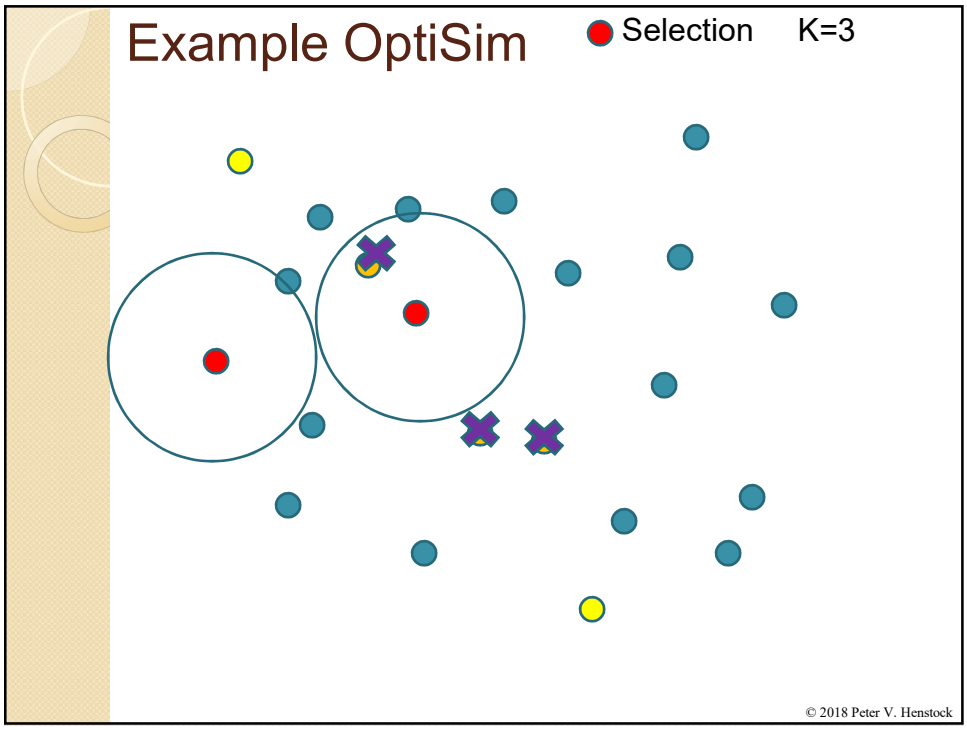

Example OptiSim
Selection
K=3
© 2018 Peter V. Henstock

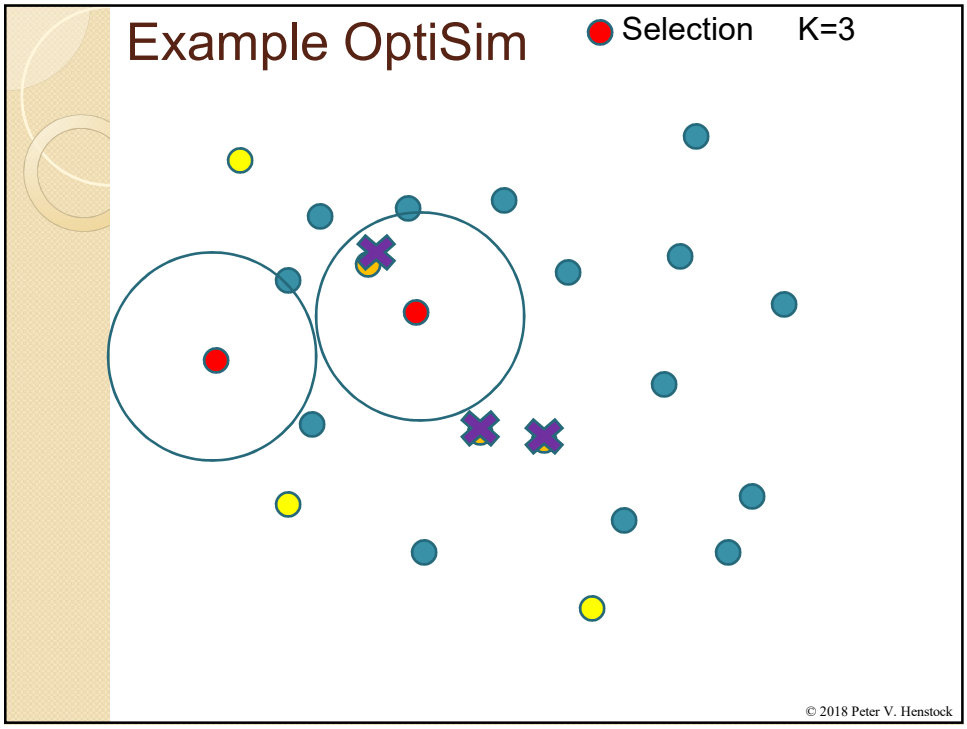

Example OptiSim
Selection
K=3
© 2018 Peter V. Henstock

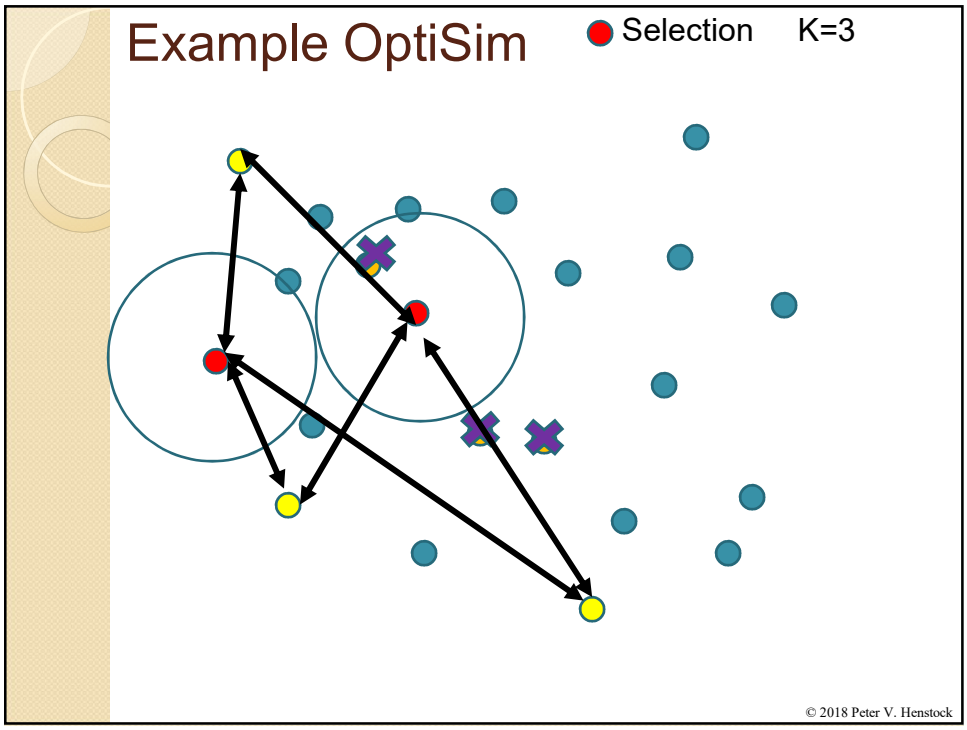

Example OptiSim
Selection
K=3
© 2018 Peter V. Henstock

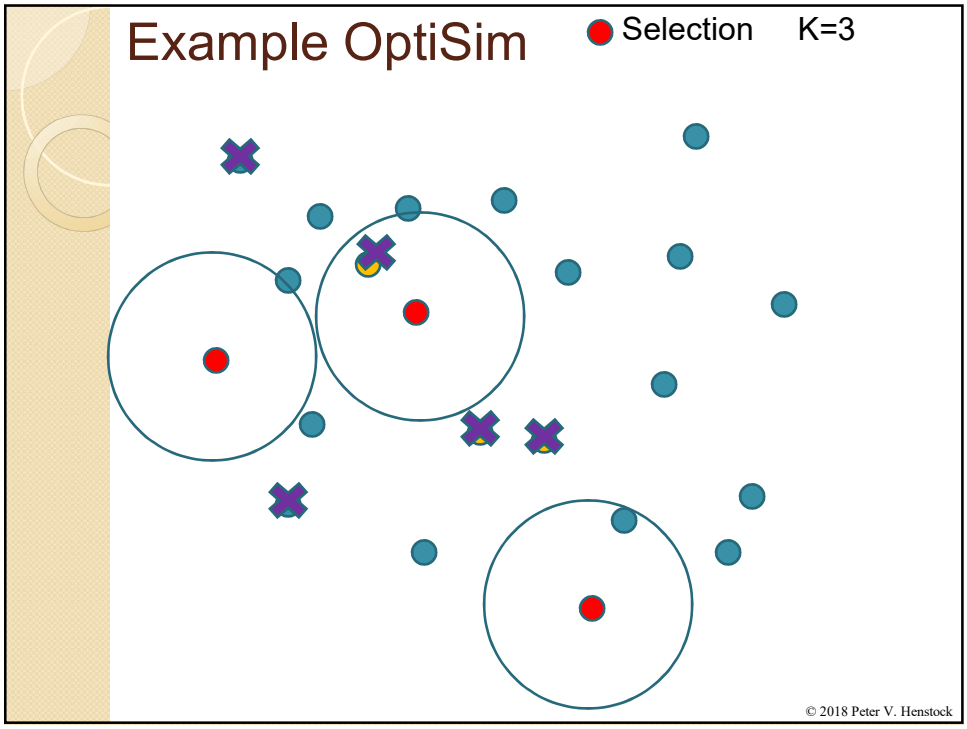

Example OptiSim
Selection
K=3
© 2018 Peter V. Henstock


Example OptiSim
Selection
K=3
© 2018 Peter V. Henstock


Example OptiSim
Selection
K=3
© 2018 Peter V. Henstock


Example OptiSim
Selection
K=3
© 2018 Peter V. Henstock


Example OptiSim
Selection
K=3
© 2018 Peter V. Henstock

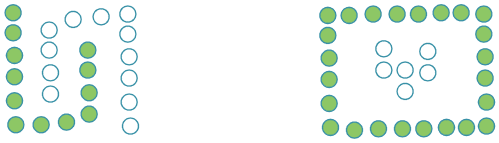


Jarvis Patrick

© 2018 Peter V. Henstock

Irregularly Shaped Clustering

- Jarvis-Patrick and many more in literature
- Tend to be more application-dependent and less useful for statistical analysis
- Good to know they exist

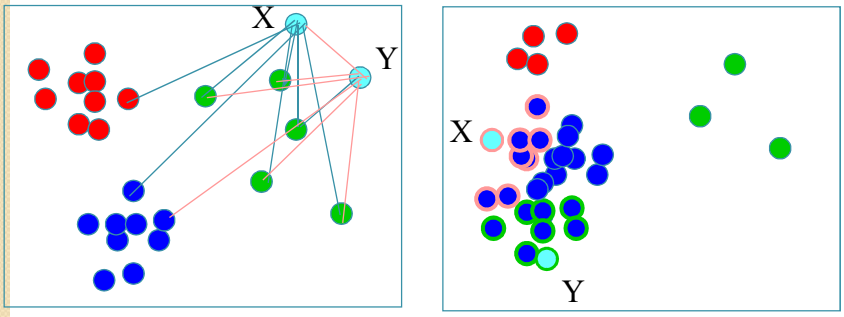


- Jarvis-Patrick example definition:
  - X and Y are in the same cluster if they share N% of their nearest M neighbors

© 2018 Peter V. Henstock

Jarvis-Patrick

- X and Y are in the same cluster if they share > N% of their nearest M neighbors (Left)
- But, there is a gradual merging of clusters through transitive property (Right)



Clustering Using a Similarity Measure Based on Shared Near Neighbors. Jarvis & Patrick, IEEE Trans. Computers Vol. C-22 No. 11, November 1973

© 2018 Peter V. Henstock

Good, Bad & Ugly of Jarvis-Patrick

- Good?
- Bad?
- Ugly?

© 2018 Peter V. Henstock

Good, Bad & Ugly of Jarvis-Patrick

- Good
  - Allows for interesting shapes that may be useful
  - Exploits peer-pressure relationships
  - Found to be useful for many applications
  - Provides a definition for a cluster
- Bad
  - Similar in complexity to hierarchical since have to find the closest compounds that is similar to all pairwise calculation
  - Nearest neighbor in a sparse region is may be far away
    - Some implementations include distance threshold as well
  - Simple cluster definition evolves into complex relationship as clusters grow
- Ugly
  - Quite complex to understand how distant points are related, particularly in > 2 dimensions

© 2018 Peter V. Henstock

Hierarchical Clustering

© 2018 Peter V. Henstock

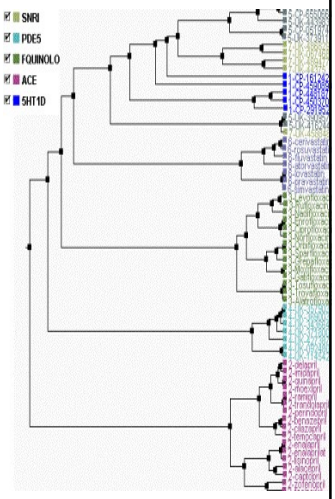
Approaches to Hierarchical

- Agglomerative
  - Start: each data point is a separate cluster
  - Merge clusters until ‘done’
- Divisive
  - Start: all data points are in single cluster
  - Divide and conquer until ‘done’
- ~80% clustering algorithms are bottom-up since there are efficient methods
  - Almost all hierarchical are agglomerative

© 2018 Peter V. Henstock

Hierarchical Clustering

- Produces “dendrogram”
  - Tree-representation
- 2 children per parent
  - (almost always)
- Represented in .nhx files
  - New Hampshire string format:
  - (((a,b),(c,(d,e)),(f,g))



© 2018 Peter V. Henstock

Two Approaches to Hierarchical

- Distance matrix approach
  - Pre-compute a distance matrix
  - Store distance matrix in memory
  - Efficient for smallish data < 30K data rows
- Reciprocal Nearest Neighbor (RNN)
  - Computes distances on the fly
  - Efficient for large data sets

© 2018 Peter V. Henstock

Distance Matrix Method

- Compute all pairwise distances
- Using an appropriate "distance metric"
  - Triangle inequality, dist >= 0, dist(x,y)=dist(y,x)
  - Only need to compute upper/lower triangle

	A	B	C	D	E
A	0	0.9	0.8	0.4	0.5
B		0	0.7	0.3	0.4
C			0	0.2	0.3
D				0	0.8
E					0

© 2018 Peter V. Henstock

Distance Matrix Method

- Repeat until there is only one tree:
  - Find the two "closest" sub-trees
  - Merge these "closest" sub-trees
  - Update the distances to all other nodes

	A	B	C	D	E
A	0	0.9	0.8	0.4	0.5
B		0	0.7	0.3	0.4
C			0	0.2	0.3
D				0	0.8
E					0

- Most similar entries are C & D
- Merge C and D; C' = (C,D)

© 2018 Peter V. Henstock

Distance Matrix Method

- Repeat until there is only one tree:
  - Find the two "closest" sub-trees
  - Merge these "closest" sub-trees
  - Update the distances to all other nodes

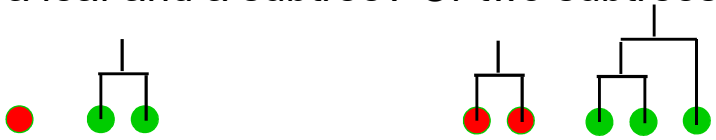
	A	B	C	D	E
A	0	0.9	0.8	0.4	0.5
B		0	0.7	0.3	0.4
C			0	0.2	0.3
D				0	0.8
E					0

- Most similar entries are C & D
- Merge C and D; C' = (C,D)
- Now what do we do?

© 2018 Peter V. Henstock

Updating distances

- How to calculate the new distance between a leaf and a subtree? Or two subtrees?

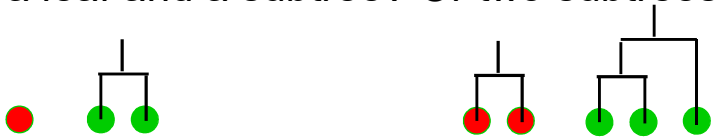


- How would you approach the problem?
- What would you do?

© 2018 Peter V. Henstock

Updating distances

- How to calculate the new distance between a leaf and a subtree? Or two subtrees?

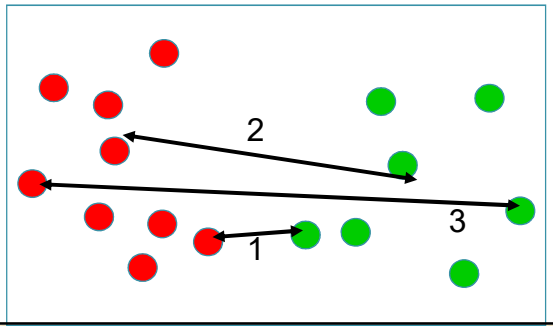


- Answer is defined as the "linkage" and there are many choices:
  - Min
  - Max
  - Median
  - Centroid
  - Average pairwise
  - Minimum variance

© 2018 Peter V. Henstock

Linkage Criteria

- Linkage is the criteria that is used as the basis of calculating distance between subtrees and/or data
- What is the distance between red and green cluster?



© 2018 Peter V. Henstock

Linkage Criteria

- Linkage is the criteria that is used as the basis of calculati distance between subtrees and/or data
- Distance between two trees:
  - Closest pair = single linkage
    - Great for identifying fully separable clusters
    - Tends to suffer from "chaining problem"
  - Furthest pair = complete linkage
    - Often creates very small clusters
  - Ward's pair = minimize variance of merge
    - Tends to work well for compounds
    - www.chemaxon.com/conf/Eurocombi_poster_Ltr.pdf
  - Average linkage: average all pairwise distances
  - Centroid linkage: centroids of clusters
  - …

© 2018 Peter V. Henstock

Update Formula (Euclidian)

Figure 2 Agglomerative clustering schemes.

Name	Distance update formula FORMULA for $d(I \cup J, K)$	Cluster dissimilarity between clusters A and B
single	$\min(d(I,K), d(J,K))$	$\min_{a \in A, b \in B} d[a,b]$
complete	$\max(d(I,K), d(J,K))$	$\max_{a \in A, b \in B} d[a,b]$
average	$\frac{n_I d(I,K) + n_J d(J,K)}{n_I + n_J}$	$\frac{1}{\lvert A \rvert \lvert B \rvert} \sum_{a \in A} \sum_{b \in B} d[a,b]$
weighted	$\frac{d(I,K) + d(J,K)}{2}$	
Ward	$\sqrt{\frac{(n_I + n_K)d(I,K) + (n_J + n_K)d(J,K) - n_K d(I,J)}{n_I + n_J + n_K}}$	$\sqrt{\frac{2\lvert A \rvert \lvert B \rvert}{\lvert A \rvert + \lvert B \rvert}} \cdot \lVert \vec{c}_A - \vec{c}_B \rVert_2$
centroid	$\sqrt{\frac{n_I d(I,K) + n_J d(J,K)}{n_I + n_J} - \frac{n_I n_J d(I,J)}{(n_I + n_J)^2}}$	$\lVert \vec{c}_A - \vec{c}_B \rVert_2$
median	$\sqrt{\frac{d(I,K)}{2} + \frac{d(J,K)}{2} - \frac{d(I,J)}{4}}$	$\lVert \vec{w}_A - \vec{w}_B \rVert_2$

Legend: Let I, J be two clusters joined into a new cluster, and let K be any other cluster. Denote by n_I, n_J and n_K the sizes of (i.e. number of elements in) clusters I, J, K, respectively.

- Modern hierarchical, agglomerative clustering methods by Daniel Műllner

© 2018 Peter V. Henstock

Lance-Williams Update Formula

dist(i ∪ j, k) = α_i dist(i, k) + α_j dist(j, k) +
β dist(i, j) + γ |dist(i, k) − dist(j, k)|

- Different linkage functions provide different values of α_i α_j β γ
- Idea: merge two clusters i and j
- Update distance to k based on the distances between (i,j), (i,k) and (j,k)
- Assume Euclidian distance

© 2018 Peter V. Henstock

Lance-Williams Update Formula

dist(i ∪ j, k) = α(i) dist(i, k) + α (j) dist(j, k)
+ βdist(i, j) + γ|dist(i, k) − dist(j, k)|

Hierarchical clustering methods (and aliases)	Lance and Williams dissimilarity update formula
Single link (nearest neighbor)	$\alpha_i = 0.5$ $\beta = 0$ $\gamma = -0.5$ (More simply: $min\{d_{ik}, d_{jk}\}$)
Complete link (diameter)	$\alpha_i = 0.5$ $\beta = 0$ $\gamma = 0.5$ (More simply: $max\{d_{ik}, d_{jk}\}$)
Group average (average link, UPGMA)	$\alpha_i = \frac{\lvert i\rvert}{\lvert i\rvert+\lvert j\rvert}$ $\beta = 0$ $\gamma = 0$
McQuitty's method (WPGMA)	$\alpha_i = 0.5$ $\beta = 0$ $\gamma = 0$
Median method (Gower's, WPGMC)	$\alpha_i = 0.5$ $\beta = -0.25$ $\gamma = 0$
Centroid (UPGMC)	$\alpha_i = \frac{\lvert i\rvert}{\lvert i\rvert+\lvert j\rvert}$ $\beta = -\frac{\lvert i\rvert\lvert j\rvert}{(\lvert i\rvert+\lvert j\rvert)^2}$ $\gamma = 0$
Ward's method (minimum variance, error sum of squares)	$\alpha_i = \frac{\lvert i\rvert+\lvert k\rvert}{\lvert i\rvert+\lvert j\rvert+\lvert k\rvert}$ $\beta = -\frac{\lvert k\rvert}{\lvert i\rvert+\lvert j\rvert+\lvert k\rvert}$ $\gamma = 0$

http://arxiv.org/pdf/1105.0121.pdf

© 2018 Peter V. Henstock

Hierarchical Clustering Algorithm

- Find all pairwise distances
- Repeat until there is only one tree:
  - Find the two "closest" sub-trees
  - Merge these "closest" sub-trees
  - Update the distances to all other nodes

	A	B	C	D	E
A	0	0.9	0.8	0.4	0.5
B		0	0.7	0.3	0.4
C			0	0.2	0.3
D				0	0.8
E					0

- Most similar entries are C & D
- Merge C and D; C' = (C,D)
- Recalculate the distance between C' and {A,B,E}

© 2018 Peter V. Henstock

Example update for Centroid

	A	B	C	D	E
A	0	0.9	0.8	0.4	0.5
B		0	0.7	0.3	0.4
C			0	0.2	0.3
D				0	0.8
E					0

Centroid (UPGMC)

$$\alpha_i = \frac{|i|}{|i|+|j|}$$
$$\beta = -\frac{|i||j|}{(|i|+|j|)^2}$$
$$\gamma = 0$$

- $dist(i \cup j, k) = \alpha_i\, dist(i, k) + \alpha_j\, dist(j, k) + \beta dist(i, j) + \gamma|dist(i, k) - dist(j, k)|$
- $dist(C \cup D, A) = \alpha dist(C, A) + \alpha dist(D, A) + \beta dist(C, D) + \gamma|dist(C, A) - dist(D, A)|$
- ½(0.8) + ½(0.4) - ¼(0.2) + 0 = 0.55
- Repeat for dist(C∪D, B) & dist(C∪D, E)

© 2018 Peter V. Henstock

Updating distances

- Potentially extremely computationally complex
  - Average pairwise would require recomputing distance between all nodes in the subtree for each candidate
- There are fast-update equations that update distance calculations every merge quickly
  - But…Only work where triangle equality holds!
- Example: Centroid:
  - d(p+q,l) = np/(np+nq)*d(p,l) + (nq/(np+nq)*d(q,l) – ni*d(p,q)
  - www.soziologie.wiso.uni-erlangen.de/ koeln/script/chap4.pdf
- Notion is you can quickly update your distance to all other objects after every merge from pre-merge distances

© 2018 Peter V. Henstock

Practical Data Structures

- Distance matrix upper triangle: ~O($N^2/2$)

	A	B	C	D	E
A	0	0.9	0.8	0.4	0.5
B		0	0.7	0.3	0.4
C			0	0.2	0.3
D				0	0.8
E					0

- Sorted minimum value of each column
  - Replace N^2 search for minimum with N
  - [0.9, 0.7 0.2 0.3] needs updating each merge
- Table: ordered merges x, y (and distance)
- Indexed array of cluster members and #members in each

© 2018 Peter V. Henstock

Practical Data Structures

- Distance matrix upper triangle
- Sorted min of columns
  - [0.9, 0.7 0.2 0.3]
  - needs updating each merge
- Table of merges
- Table of clusters (D active?)

	A	B	C	D	E
A	0	0.9	0.8	0.4	0.5
B		0	0.7	0.3	0.4
C			0	0.2	0.3
D				0	0.8
E					0

X	Y	Dist
C	D	0.20
C	E	0.24
B	C	0.56
A	B	0.62

A → A 1
B → B 1
C → C 1
D → D 1
E → E 1

A → A 1
B → B 1
C → C,D 2
D → null 0
E → E 1

© 2018 Peter V. Henstock

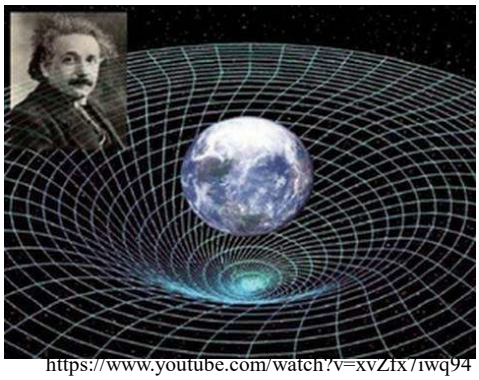
Big O applied to Hierarchical

- Find all pairwise distances
  - $O(n^2)$ in space; $O(n^2)$ in time

  Repeat until there is only one tree:
  - Find the two "closest" sub-trees
    - $O(n^2)$ in time but can do in $O(n)$ = $O(n^2)$
  - Merge these "closest" sub-trees
    - $O(1)$ $O(n)$
  - Update the distances to all other nodes
    - $O(n) \times n = O(n^2)$
- Conclusion $O(n^2)$ in space and time
- 10,000 data points →~1E8 time/space units
- 1,000,000 data points→ ~1E12 time/space units

© 2018 Peter V. Henstock

Time and Space



https://www.youtube.com/watch?v=xvZIx7iwq94

- How much time?
- How much memory?

© 2018 Peter V. Henstock

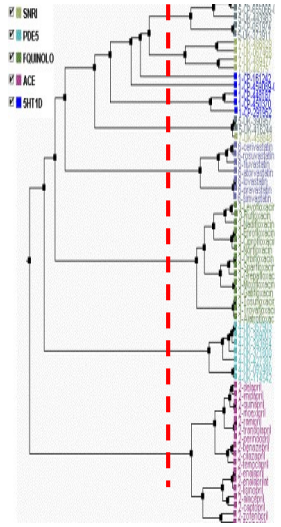
Time and Space

- How much time?
  - Compute distance matrix $O(N^2)$
  - For each data: N times
    - Find min distance → O(N)
    - Update N distances → O(N)
- How much memory?
  - Distance matrix $O(N-1)^2/2) \rightarrow O(N^2)$

© 2018 Peter V. Henstock

Good, Bad and Ugly of Hierarchical

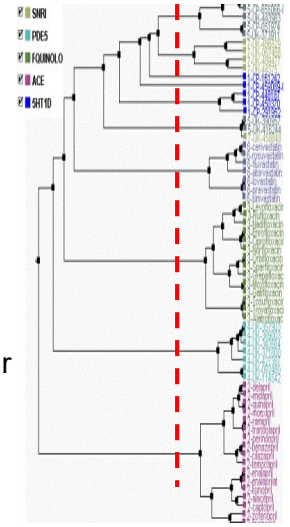
- Good?
- Bad?
- Ugly?



© 2018 Peter V. Henstock

Good, Bad and Ugly of Hierarchical

- Good
  - Very intuitive for end users
  - Very few parameters required
    - distance function and linking function
    - Cut the tree into “clusters” afterwards
  - Define clusters after clustering
- Bad
  - Takes a lot of memory and speed
  - Do about 25,000 on a laptop in an hour
  - Challenging to do significantly more
- Ugly
  - Can’t understand how clusters relate
  - No definition of a cluster within the hierarchy
  - Spatial arrangement is meaningless between clusters



© 2018 Peter V. Henstock

Reciprocal Nearest Neighbor

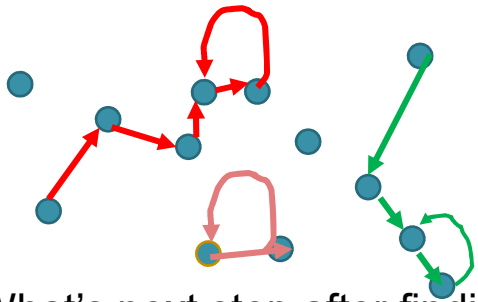
- Only works for a subset of hierarchical clustering methods
- On-the-fly clustering
- Avoids pre-computing the distance matrix

- Repeat until done
  - Take a data point or cluster u1
  - Find closest point of u1 → u2
  - Find closest point of u2 → u3
  - …
  - If closest(ui) == u(i-1) merge them

© 2018 Peter V. Henstock

Reciprocal Nearest Neighbor

- Form a RNN-chain that identifies the leaf of the dendrogram
- Lots of mathematical/geometric proofs that it works

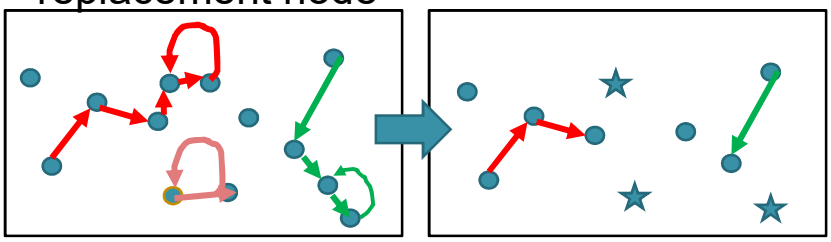


- What's next step after finding each RNN?

© 2018 Peter V. Henstock

Update formula

- Previously computed the distance to all the other nodes d(aUb,*)
- However, we haven't computed the distances of d(a,*) or d(b,*)
- Replace the merged cluster with a replacement node



© 2018 Peter V. Henstock

Update Formulae

Median method (Gower's, WPGMC)	$\alpha_i = 0.5$ $\beta = -0.25$ $\gamma = 0$	$\mathbf{g} = \frac{\mathbf{g}_i + \mathbf{g}_j}{2}$	$\|\mathbf{g}_i - \mathbf{g}_j\|^2$
Centroid (UPGMC)	$\alpha_i = \frac{\|i\|}{\|i\|+\|j\|}$ $\beta = -\frac{\|i\|\|j\|}{(\|i\|+\|j\|)^2}$ $\gamma = 0$	$\mathbf{g} = \frac{\|i\|\mathbf{g}_i + \|j\|\mathbf{g}_j}{\|i\|+\|j\|}$	$\|\mathbf{g}_i - \mathbf{g}_j\|^2$
Ward's method (minimum variance, error sum of squares)	$\alpha_i = \frac{\|i\|+\|k\|}{\|i\|+\|j\|+\|k\|}$ $\beta = -\frac{\|k\|}{\|i\|+\|j\|+\|k\|}$ $\gamma = 0$	$\mathbf{g} = \frac{\|i\|\mathbf{g}_i + \|j\|\mathbf{g}_j}{\|i\|+\|j\|}$	$\frac{\|i\|\|j\|}{\|i\|+\|j\|}\|\mathbf{g}_i - \mathbf{g}_j\|^2$

- Use the g's to compute the new centroid from those merged

Recommended paper on RNN
Methods of Hierarchical Clustering by Murtagh & Contreras 2011 http://arxiv.org/pdf/1105.0121.pdf

© 2018 Peter V. Henstock

Recent Work

- If you assume you will cut into clusters anyway
  - Pre-cluster using a fast technique (I.e. leader)
  - Perform hierarchical clustering on clusters with RNN
  - Takes $kO(m^2)$ where m<<n so avoid full n x n matrix
- Reciprocal Nearest Neighbor (RNN)
  - Only works for some distance/linkage combos
  - Take entry (j) and find nearest neighbor of j: k
  - Repeat to find NN(k) until NN(k)=NN(j) which is RNN
  - NN(x1)=x2; NN(x2)=x3; NN(x3)=x4; …NN(x14)=x13
  - RNN chain will form an independent subtree

© 2018 Peter V. Henstock

RNN Movie Link & References

- http://www.juergenwiki.de/work/wiki/doku.php?id=public:reciprocal_nearest_neighbour_clustering_rnn
- Recommended paper on RNN
- Methods of Hierarchical Clustering by Murtagh & Contreras 2011 http://arxiv.org/pdf/1105.0121.pdf

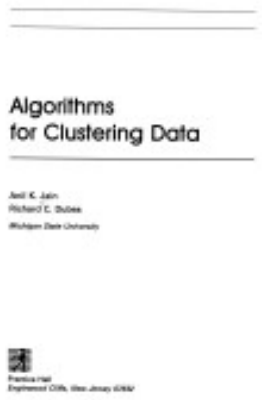
© 2018 Peter V. Henstock

Assessing Clustering

© 2018 Peter V. Henstock

Cluster Validation

- "The validation of clustering structures is the most difficult and frustrating part of cluster analysis."
- "Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."



© 2018 Peter V. Henstock

General Assessment

- Do we know the correct clustering?
  - "Extrinsic" = Supervised
  - Classification rates
  - Information theoretic or Entropy
  - Purity
  - Rand Index
- No ground truth: "Intrinsic"
  - Assess goodness of clustering properties
  - Separation of data
  - Compact/density of clusters

© 2018 Peter V. Henstock

Extrinsic Measures

- Properties of good clusters
  - Homogeneity / Purity → clusters same truth
  - Completeness: same truth → same clusters
  - Rag bag
    - Tradeoff of spoiling pure cluster vs. "other" cluster
  - Small cluster preservation
- Set based
- Entropy
- Contingency Table
- B-Cubed

© 2018 Peter V. Henstock

Purity Measures (Set measures)

- C = #clusters L = #truth categories
- N = #clustered data points

- Purity = $\sum_i \frac{|C_i|}{N} max_j Precision(C_i, L_j)$
- $Precision(C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|}$

© 2018 Peter V. Henstock

Contingency Tables

- Set of truth labels A,B,C,D
- Set of cluster labels a,b,c,d
  - Might have a,b→A c→C,D d→D

Counts of Pairs	Same Cluster	Different Cluster
Same Truth	SS	DS
Different Truth	SD	DD

- Rand statistic: $\frac{SS+DD}{SS+SD+DS+D}$
- Jaccard Coefficient J = $\frac{SS}{SS+SD+DS}$
- Folkes & Mallows FM = $\sqrt{\frac{SS}{SS+SD}\frac{SS}{SS+DS}}$

© 2018 Peter V. Henstock

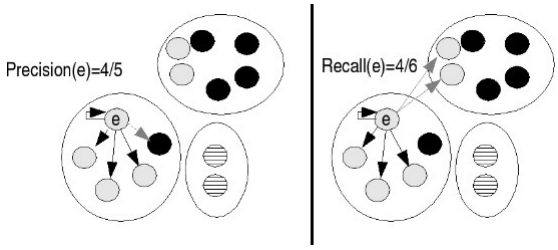
Entropy

$$\text{Entropy} = -\sum_j \frac{n_j}{n} \sum_i P(i,j) \times \log_2 P(i,j)$$

- P(i,j) = probability of finding element of category I in cluster j
- n_j = # elements in cluster j
- N = # elements overall

© 2018 Peter V. Henstock

B-Cubed



- Compute precision & recall at every point
- Precision ~ purity
  - Fraction of items in with same truth in cluster
- Recall ~ distribution
  - Fraction of total items with same truth in that particular cluster

© 2018 Peter V. Henstock

For more information

- A comparison of extrinsic clustering evaluation metrics on formal constraints
- Amigo et al. 2009
- http://nlp.uned.es/docs/amigo2007a.pdf

© 2018 Peter V. Henstock

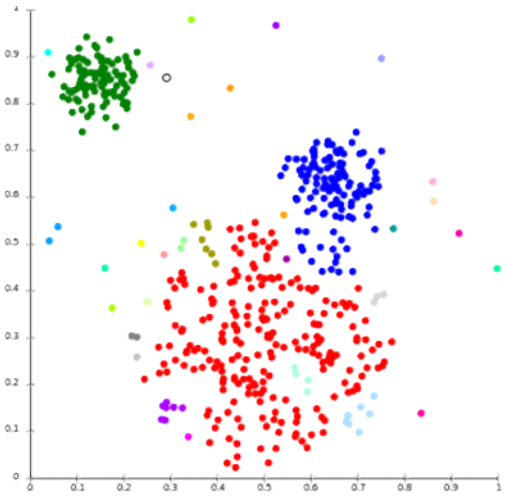
Unsupervised: Silhouette Measure

- Cluster the data first
- a(i) = average dissimilarity of data i with all other data in the same cluster
  - ~Intra cluster dissimilarity
- b(i) = lowest average dissimilarity of i to any other cluster to which is not a member
  - Dissimilarity to nearest neighbor cluster
- $silhouette(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$
  - Near 1 if tight cluster with high separation
  - Near -1 loose cluster with others closer

© 2018 Peter V. Henstock

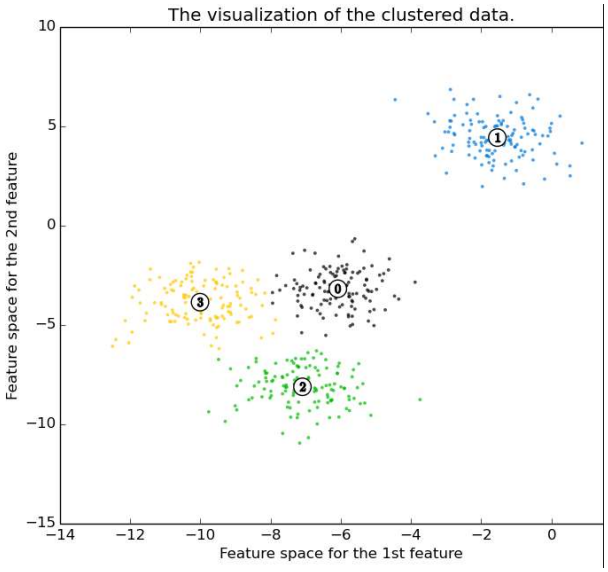
Silhouette for green, blue, red

$$silhouette(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$



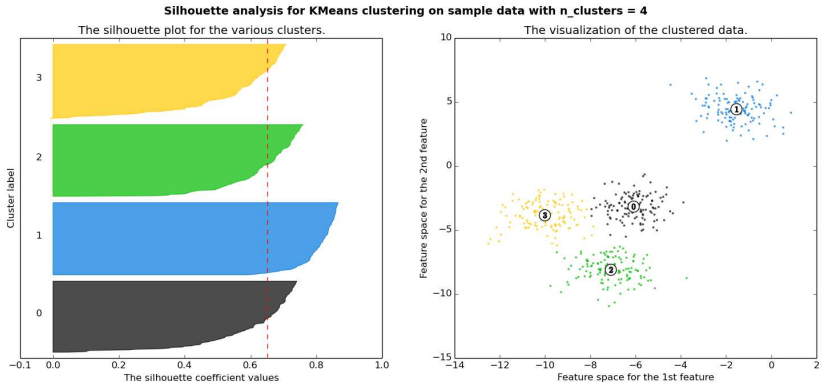
© 2018 Peter V. Henstock

Which will have largest silhouette?



© 2018 Peter V. Henstock

Silhouette Plots



- http://scikit-learn.org/stable/_images/plot_kmeans_silhouette_analysis_003.png

© 2018 Peter V. Henstock

Assessing Number of Clusters

- Empirical → k = ~sqrt(N) N=data pts.
- Elbow/Knee method

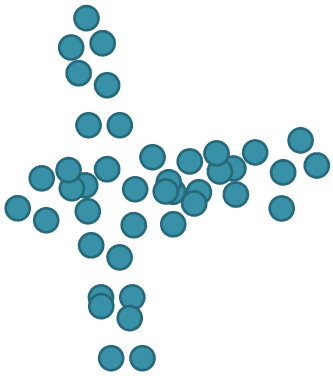
- Cross-validation
  - Discuss this in classification
  - Divide data up into pieces and estimate
  -

© 2018 Peter V. Henstock

Mixture Models GMM

© 2018 Peter V. Henstock

How would K-means handle this?



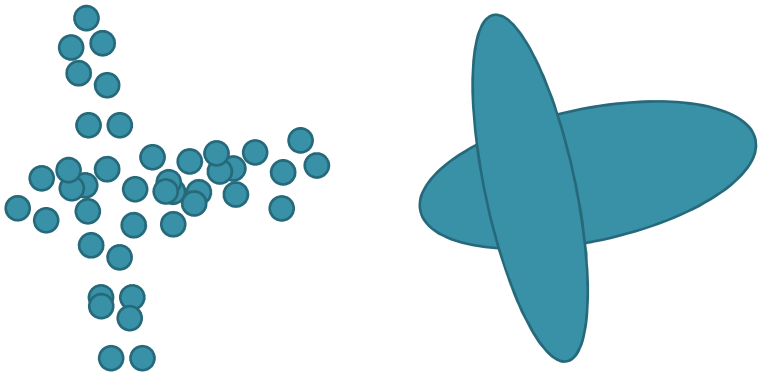
© 2018 Peter V. Henstock

K-Means Issues

- Likes round/spherical clusters
  - Based on the distance metric
- Hard-limiter:
  - Points are in only one cluster
  - Probability = {0, 1} of membership in cluster
- Statistically, it a centroid and a non-statistical ad hoc radius
- Many fields prefer Gaussian distributions over round 'blob' clusters

© 2018 Peter V. Henstock

How would K-means handle this?



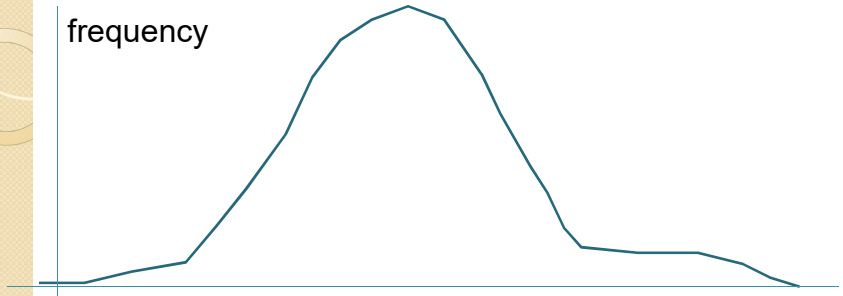
- Gaussian distribution is more satisfying

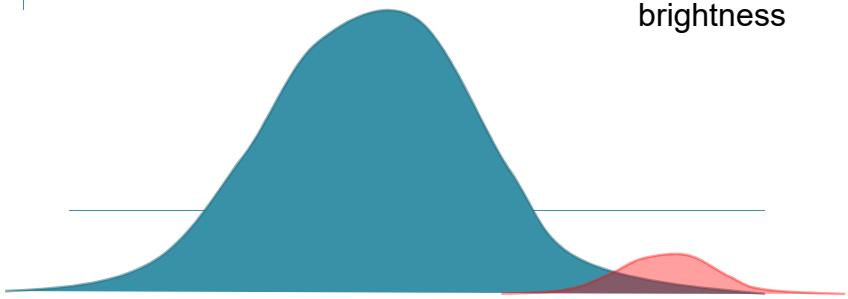
© 2018 Peter V. Henstock

Lena: Edge Detection

© 2018 Peter V. Henstock

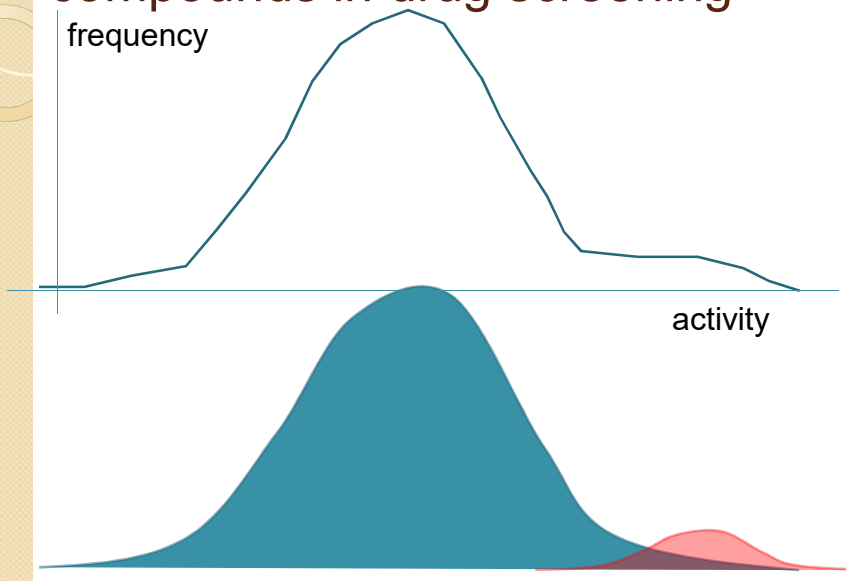
How to threshold?





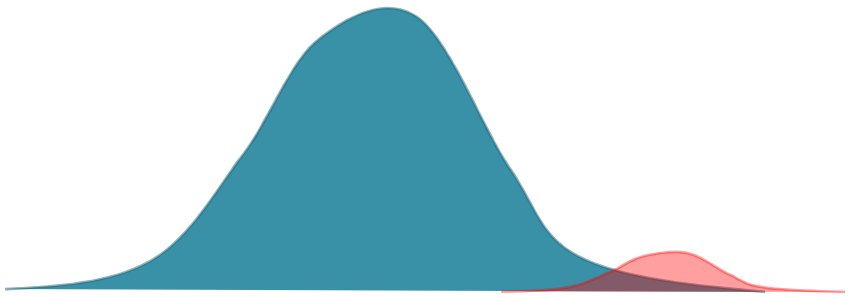
© 2018 Peter V. Henstock

Same problem: active vs. inert compounds in drug screening



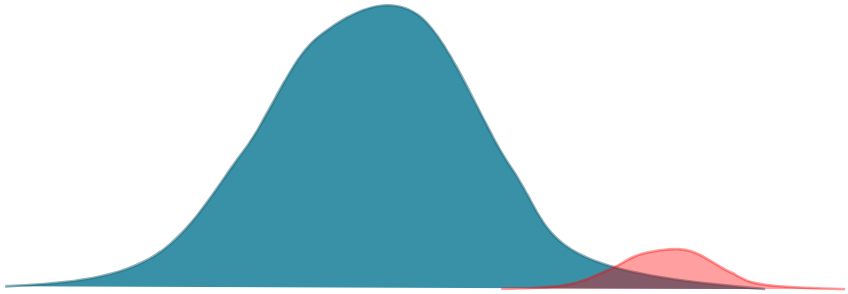
© 2018 Peter V. Henstock

What are the unknowns?



© 2018 Peter V. Henstock

What are the unknowns?

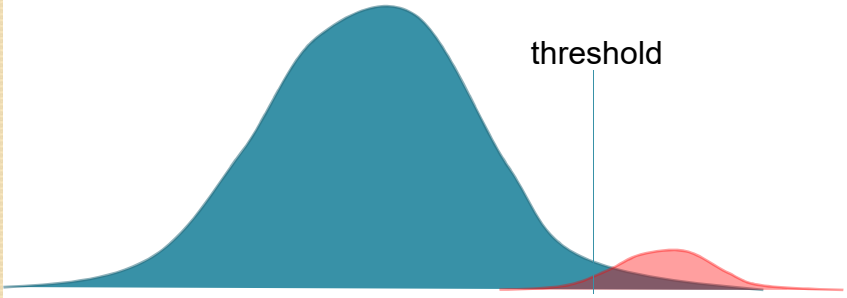


- Blue: mean and standard deviation
- Red: mean and standard deviation
- Probability of blue or red

© 2018 Peter V. Henstock

Hard vs. Soft Threshold

- K-means uses hard threshold: black/white
  - In one cluster or not: r is {0, 1}



- Soft threshold: shades of gray
  - x has 35% chance of being blue, and
  - X has 65% chance of being red

© 2018 Peter V. Henstock

Gaussian Mixture Model

- Actually not mixing anything
- Using multiple Gaussians to represent data instead of just one

https://clipartxtras.com/categories/view/90a208683ac530bb5e65d28d8636d2a5baf35606/mix-drawing.html

© 2018 Peter V. Henstock

Mixture Models

- $P(x) = \sum_{c=0}^{\#clusters} P(cluster\_c)P(x|cluster\_c)$
- Generative model for P(x) = one point
  - Pick a cluster at random P(cluster_c)
    - Assume each cluster has a probability
      - Gaussian prior
  - Generate a value X from the Gaussian distribution of cluster_c
  - Sum over values to get P(x) above
  - Also called a "latent" model
- Max likelihood solution:
  - Choose clusters so they are most likely given the data → maximize P(x)

© 2018 Peter V. Henstock

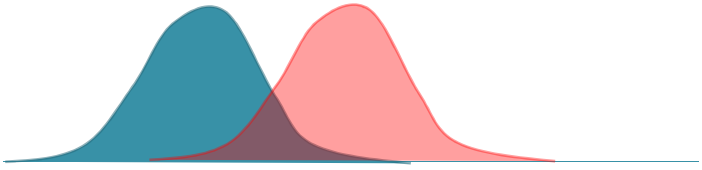
Mixture Models

- Models for p(x|c_i)
  - Normal distributions is standard approach
  - Assume iid so can use a product
- $P(\text{x}|\text{cluster\_c}) = \prod_{j=1}^{N} P(xj|cluster\_c)$
- What are we trying to find?

© 2018 Peter V. Henstock

Mixture of Gaussians

$$P(x|\mu_i, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp[\frac{-(x - \mu_i)^2}{2\sigma^2}]$$



$$\mathcal{N}(\underline{x}\,;\,\underline{\mu},\Sigma) = \frac{1}{(2\pi)^{d/2}}|\Sigma|^{-1/2}\exp\left\{-\frac{1}{2}(\underline{x}-\underline{\mu})^T\Sigma^{-1}(\underline{x}-\underline{\mu})\right\}$$

© 2018 Peter V. Henstock

How to learn mixture models?

- "Estimation Maximization" algorithm
- = EM algorithm
- Maximizes likelihood in presence of 'missing data' i.e. the clusters
- Defines solution as f(g) and g(f) and iterates the optimization between each
- 2 step algorithm

© 2018 Peter V. Henstock

EM Algorithm

- Initialize parameters
- Loop:
  - E-step:
    - Use current parameters to compute estimates of unobserved variables
  - M-step
    - Update parameter values to maximize the probability of the observed and unobserved data

© 2018 Peter V. Henstock

EM Algorithm

- Initialize parameters (perhaps K-means)
- Loop:
  - E-step—estimating the assignments
    - Use current parameters to compute estimates of unobserved variables
    - Compute probability that point xi maps to cluster_c
  - M-step—maximizing the values
    - Update parameter values to maximize the probability of the observed and unobserved data
    - Compute probability of cluster_c
    - Compute centroid μ_c
    - Compute covariance Σ or stdev σ of cluster c

© 2018 Peter V. Henstock

E-step

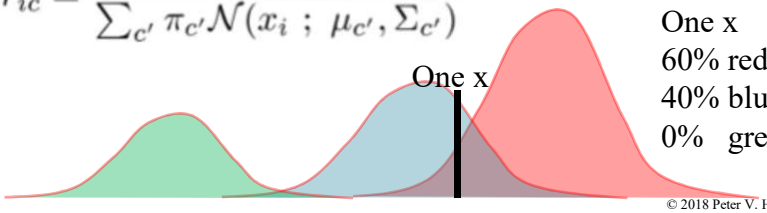
- Initialize means (perhaps using K-means)
- For each point x_j: compute
- $P(\mu_c|x_j) = \frac{P(x_j|\mu_c)P(\mu_c)}{P(x_j)} = \frac{P(x_j|\mu_c)P(\mu_c)}{\sum_{k=1}^{\#clust} P(\mu_k)P(x_j|\mu_k)}$
- Recall $P(x|\mu_c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp[\frac{-(x-\mu_c)^2}{2\sigma_c^2}]$
- Changing notation, we are computing r just as we did the K-means

© 2018 Peter V. Henstock

E-Step Another way

- $P(\mu_c|x_j) = \frac{P(x_j|\mu_c)P(\mu_c)}{P(x_j)} = \frac{P(\mu_c)P(x_j|\mu_c)}{\sum_{k=1}^{\#clust} P(\mu_k)P(x_j|\mu_k)}$
- In K-means, we computed which x's were assigned to which cluster centroids
- Here, we are using the likelihood values to figure out which cluster it belongs to
- Normalizing in denominator so sums to 1

$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i \,;\, \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \,;\, \mu_{c'}, \Sigma_{c'})}$$



One x
60% red
40% blue
0% green

© 2018 Peter V. Henstock

E-Step Yet another way

- Removing the hard-limit "closest" assignment r of each x_i to cluster c_j
- Adding a probabilistic soft-limit that assigns x_i to each cluster c_j with r in [0,1]
- Each x_i has to belong to some cluster with a probability so we scale them so that require $\Sigma r = 1$
- Sum of assignments of point xi to all clusters sums to 1 over all clusters

© 2018 Peter V. Henstock

M-Step

- M step compute parameters: π_c, μ_c, σ_c
- P(cluster_c) = $\pi_c = \frac{1}{N}\sum_{j=1}^{N} P(\mu_c | x_j)$
  - $P(\mu_c | x_i)$ = probability x_i belongs to μ_c
- $\mu_c = \frac{\sum_{j=1}^{N} x_j P(\mu_c|x_j)}{\sum_{j=1}^{N} P(\mu_c|x_j)}$
  - Weighted average of the values in cluster
- $\sigma_c = \sqrt{\frac{\sum_{j=1}^{N}(x_j-\mu_c)^2 P(\mu_c|x_j)}{\sum_{j=1}^{N} P(\mu_c|x_j)}}$
  - Weighted average of variance

© 2018 Peter V. Henstock

M-Step again: work with 'soft' limit

- What % of data are in cluster c?
  - r_{ic} = assignment of xi to cluster c
  - $\Sigma_i r_{ic}$ = sum of all xi mapping to cluster c = m_c
  - Need to normalize to [0, 1] so divide by $\Sigma_c m_c$
  - $\pi_c = m_c / \Sigma_k m_k$ = probability of cluster c
- Mean is defined as Σ xp(x) in general
  - Centroid of cluster c = $\Sigma_i r_{ic}$ xi / m_c
  - If xi is strongly in the cluster r large and impacts the centroid
- Covar cluster c= $\frac{1}{m_c}\sum_i r_{ic}[x_i - \mu_c]^T [x_i - \mu_c]$

© 2018 Peter V. Henstock

K-Means vs. EM

- K-Means is a special case of EM
  - Cluster priors are equal
  - Covariance matrix: identity matrix
  - Means = means
- Assignments in K-Means are not a probability but a hard limit

© 2018 Peter V. Henstock

EM Algorithm Summary of Updates

E.M. for General GMMs

$p_i(t)$ is shorthand for estimate of $P(\omega_i)$ on t'th iteration

Iterate. On the t'th iteration let our estimates be

$$\lambda_t = \{ \mu_1(t), \mu_2(t) \ldots \mu_c(t), \Sigma_1(t), \Sigma_2(t) \ldots \Sigma_c(t), p_1(t), p_2(t) \ldots p_c(t) \}$$

E-step

Compute "expected" classes of all datapoints for each class

Just evaluate a Gaussian at x_k

$$P(w_i | x_k, \lambda_t) = \frac{p(x_k | w_i, \lambda_t) P(w_i | \lambda_t)}{p(x_k | \lambda_t)} = \frac{p(x_k | w_i, \mu_i(t), \Sigma_i(t)) p_i(t)}{\sum_{j=1}^{c} p(x_k | w_j, \mu_j(t), \Sigma_j(t)) p_j(t)}$$

M-step.

Compute Max. like **μ** given our data's class membership distributions

$$\mu_i(t+1) = \frac{\sum_k P(w_i | x_k, \lambda_t) x_k}{\sum_k P(w_i | x_k, \lambda_t)} \qquad \Sigma_i(t+1) = \frac{\sum_k P(w_i | x_k, \lambda_t)[x_k - \mu_i(t+1)][x_k - \mu_i(t+1)]^T}{\sum_k P(w_i | x_k, \lambda_t)}$$

$$p_i(t+1) = \frac{\sum_k P(w_i | x_k, \lambda_t)}{R}$$

R = #records

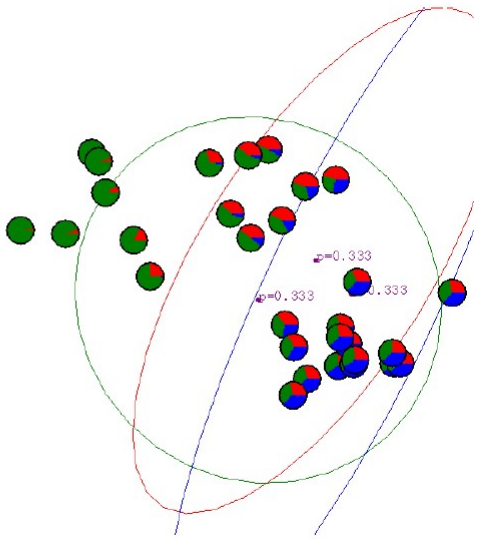
Copyright © 2001, Andrew W. Moore

Clustering with Gaussian Mixtures: Slide 39

©2017 Peter V. Henstock

EM Algorithm Initialized

Gaussian Mixture Example: Start



Advance apologies: in Black and White this example will be incomprehensible

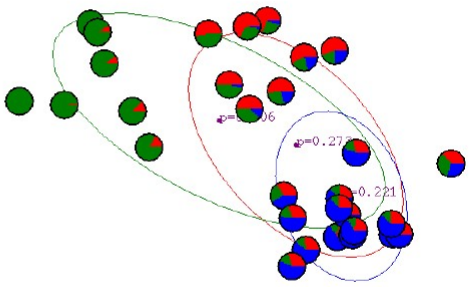
Copyright © 2001, Andrew W. Moore

Clustering with Gaussian Mixtures: Slide 40

©2017 Peter V. Henstock

EM Algorithm After First Iteration

After first iteration



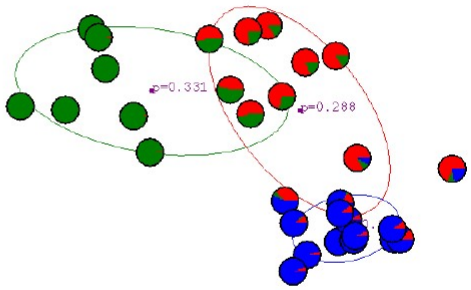
Copyright © 2001, Andrew W. Moore

Clustering with Gaussian Mixtures: Slide 41

©2017 Peter V. Henstock

EM Algorithm After 4 Iterations

After 4th iteration



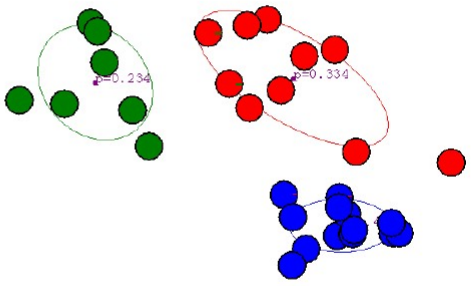
Copyright © 2001, Andrew W. Moore

Clustering with Gaussian Mixtures: Slide 44

©2017 Peter V. Henstock

EM Algorithm Near End

After 20th iteration



Copyright © 2001, Andrew W. Moore

Clustering with Gaussian Mixtures: Slide 47

©2017 Peter V. Henstock

Good, Bad & Ugly of EM

- Good
- Bad
- Ugly

© 2018 Peter V. Henstock

Good, Bad & Ugly of EM

- Good
  - Statistically accurate and robust to various start points
  - Reasonably fast
  - Ideal for medium to large clusters
  - “Soft” cluster memberships
- Bad
  - Assumes have Gaussian distributions
    - Small isolated clusters work poorly
  - Need to know how many clusters you have
  - Can sometimes be fussy about initialization
- Ugly
  - Quite beautiful and elegant when assumptions apply

© 2018 Peter V. Henstock

Density Based Techniques

- Mean Shift Clustering
- DBSCAN
- OPTICS

© 2018 Peter V. Henstock