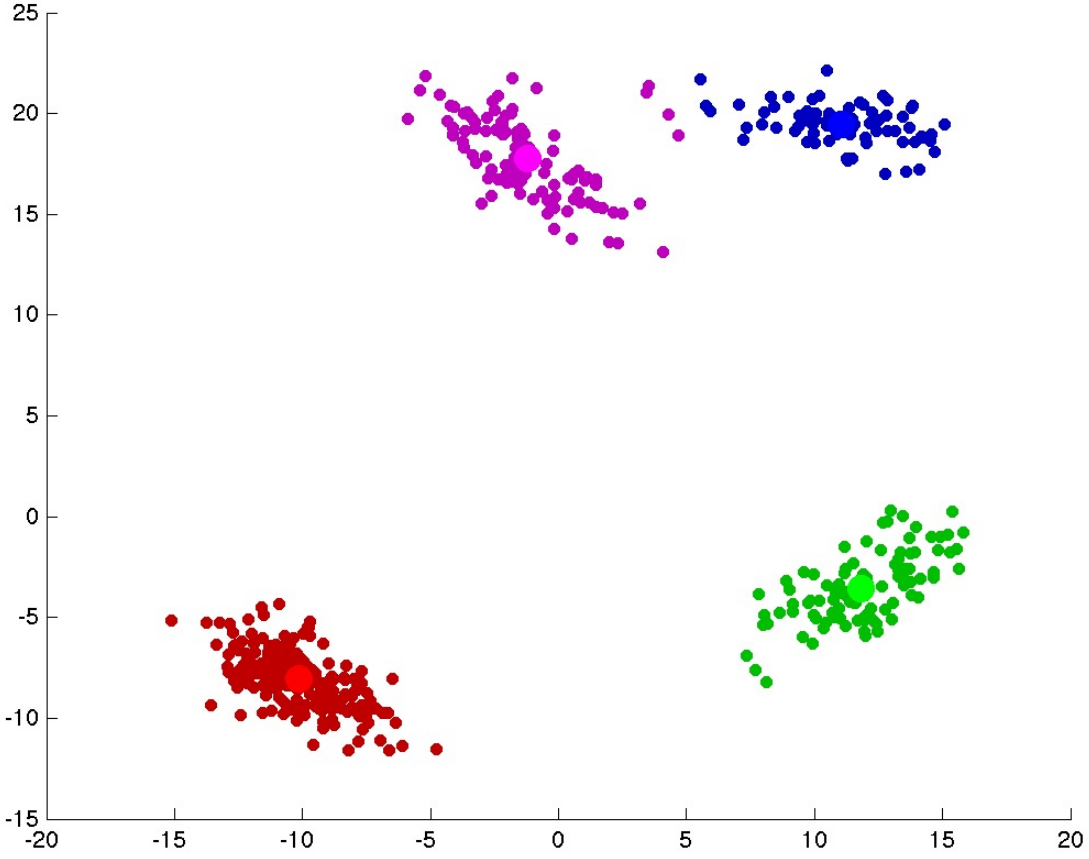
CPSC 340:
Machine Learning and Data Mining

More Clustering

Spring 2022 (2021W2)

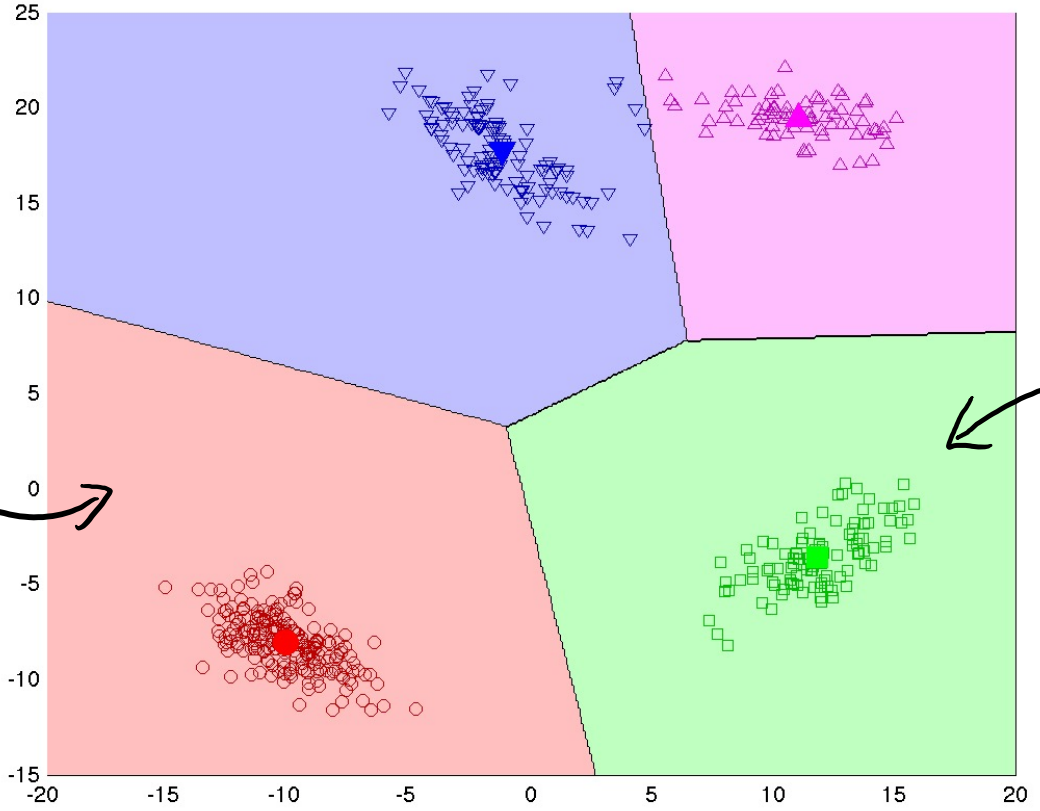
Last Time: K-Means Clustering

- We want to cluster data:
  - Assign examples to groups.
- K-means clustering:
  - Define groups by “means”
  - Assigns examples to nearest mean.
    (And updates means during training.)
- Also used for vector quantization:
  - Use means as “prototypes” of groups.
  - “Pick clothing sizes or spaghetti sauces”.
- Issues with k-means:
  - Fast but sensitive to initialization.
  - Choosing ‘k’ is annoying.



Shape of K-Means Clusters

- K-means partitions the space based on the "closest mean":



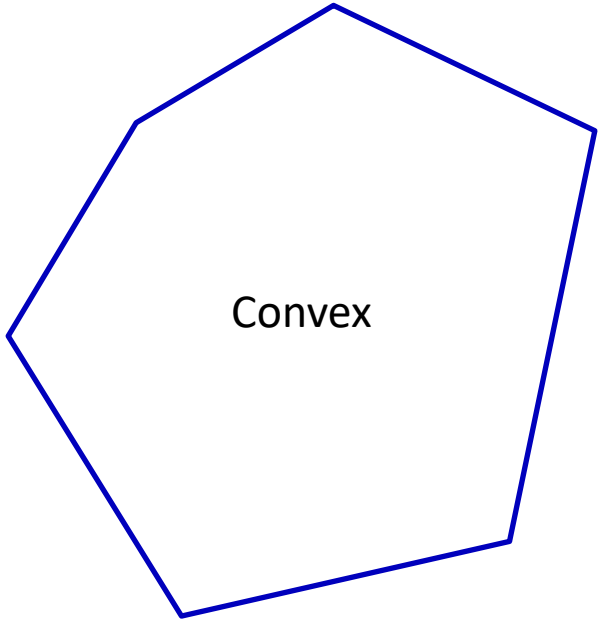
Points here are closest to red mean.

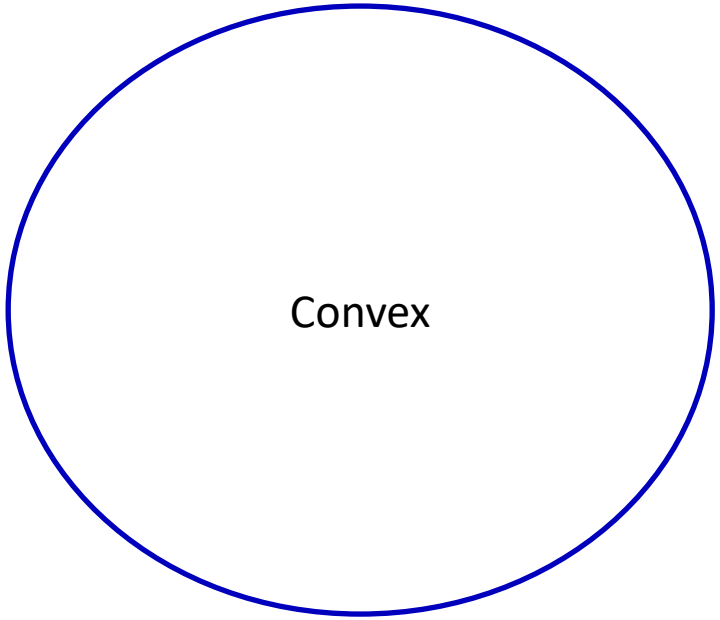
Points here are closest to green mean.

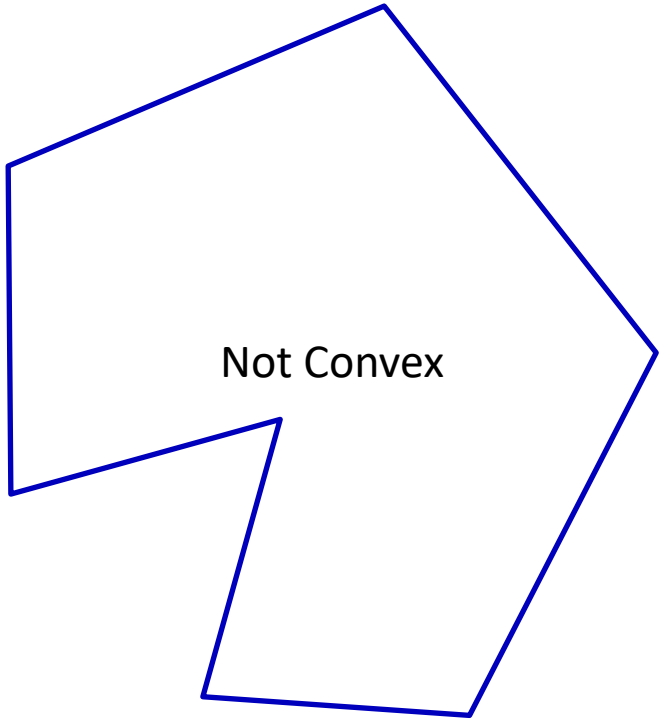
- Observe that the clusters are convex regions (proof in bonus).

Convex Sets

- A set is convex if line between two points in the set stays in the set.



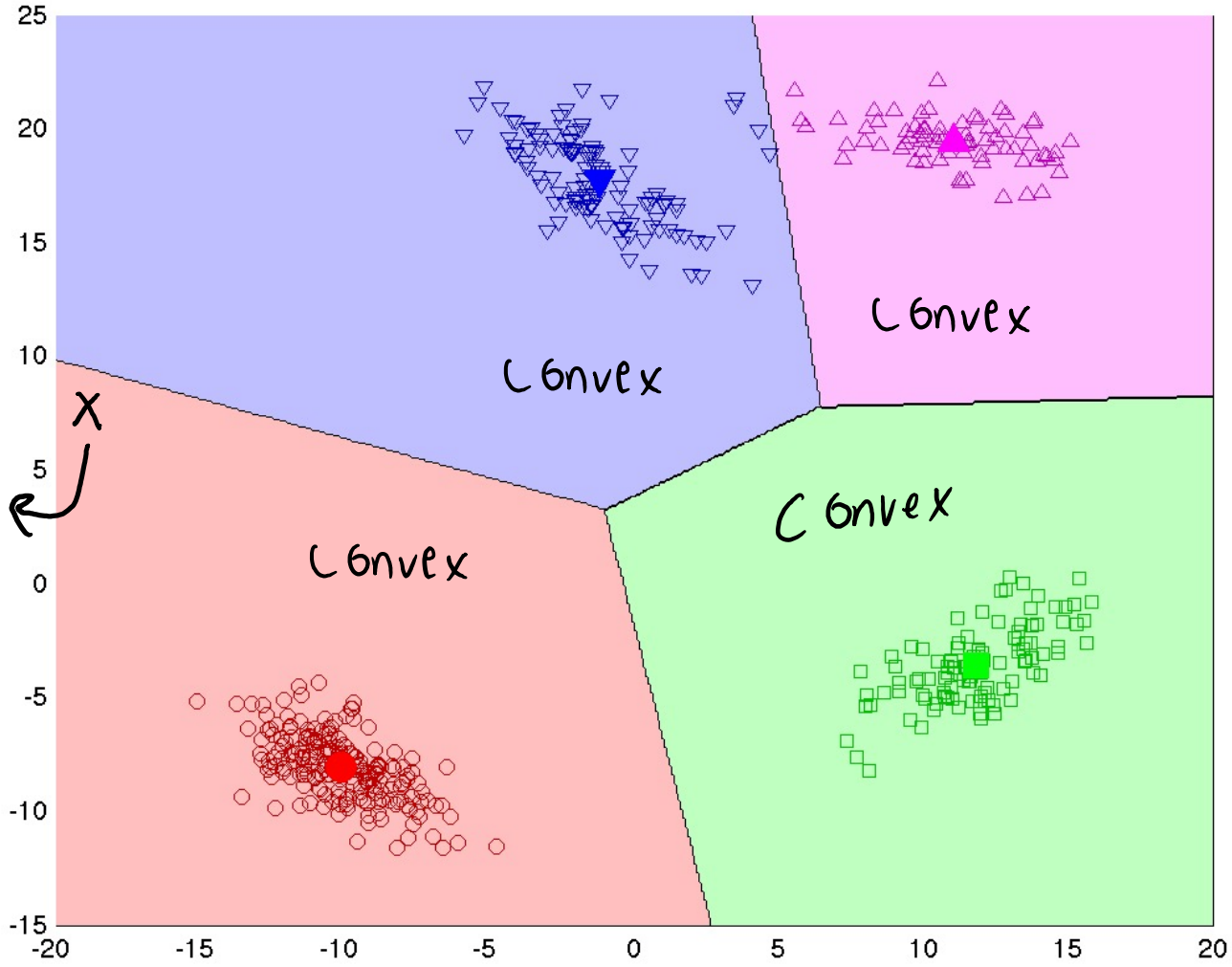




Shape of K-Means Clusters

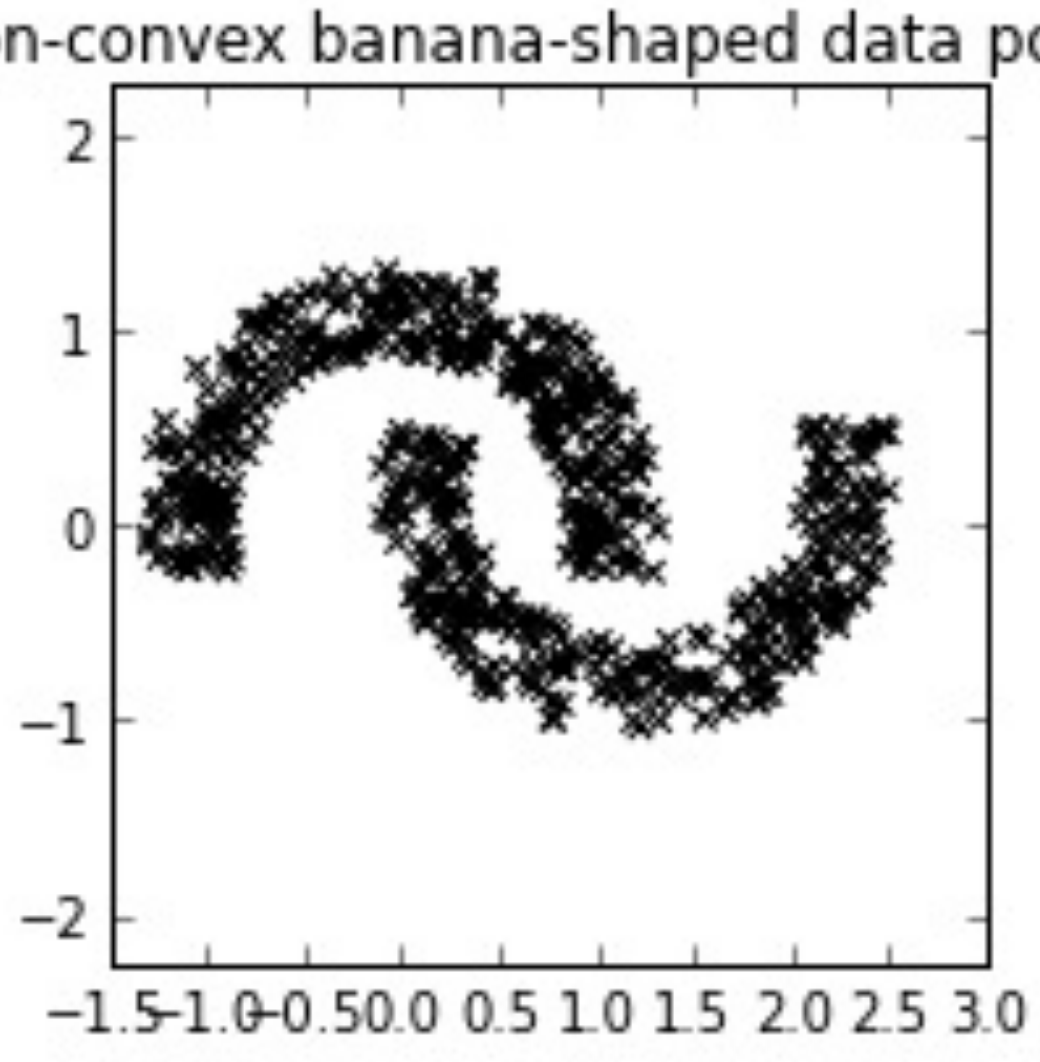
Issues with shape of K-means clusters:

1. Clusters in the data might not be convex.
2. Does this point really belong in red cluster



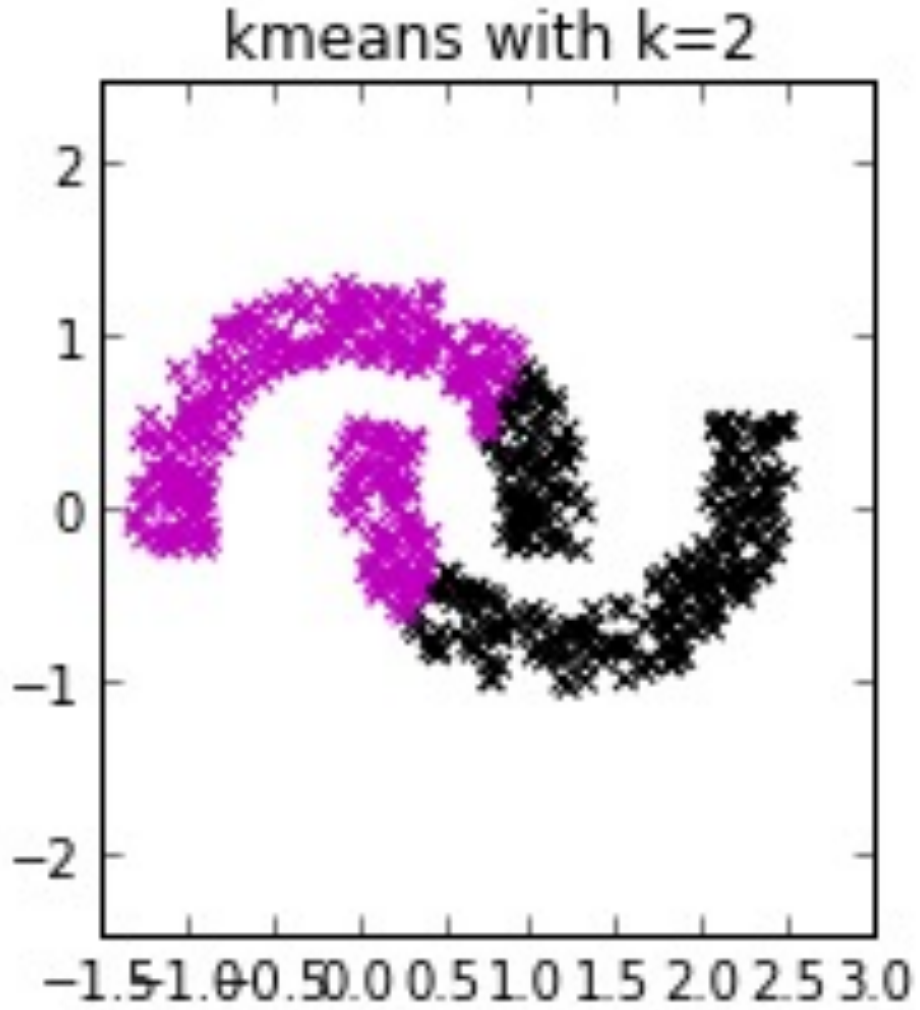
Animation

K-Means with Non-Convex Clusters



https://corelifesciences.com/human-long-non-coding-rna-expression-microarray-service.html

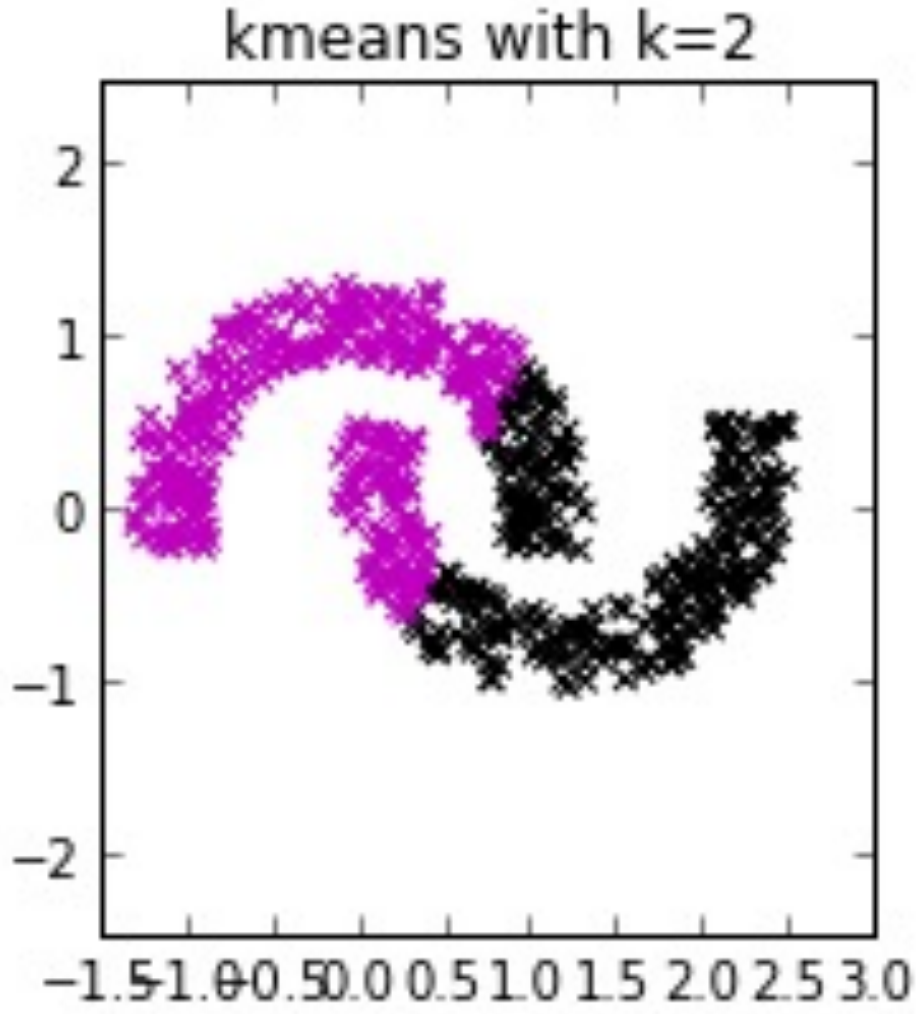
K-Means with Non-Convex Clusters



K-means cannot separate some non-convex clusters

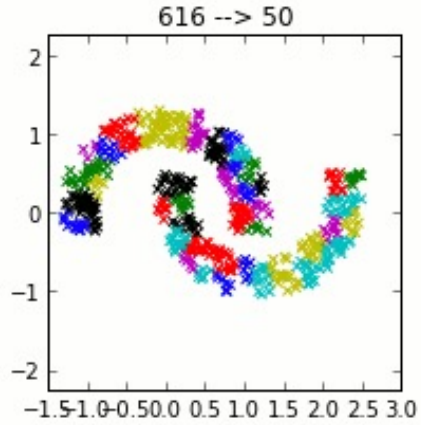
https://corelifesciences.com/human-long-non-coding-rna-expression-microarray-service.html

K-Means with Non-Convex Clusters



K-means cannot separate some non-convex clusters

Though over-clustering can help ("hierarchical")



https://corelifesciences.com/human-long-non-coding-rna-expression-microarray-service.html

bonus!

john snow - Google Sea

google.ca/search?safe=off&tbm=isch&q=john+snow&nfpr=1&sa=X&ved=0ahUKEwj9gsnWsLXWAhUJ8mMKHUAQBdQQvgUIkAIoAQ&biw=1440&bih=750&d

CPSC 540 | CPSC 340 | Old CPSC 340 | 340 TA | MLAPP | Calendar | My Page | 540 - Google Docs | Gmail - Inbox | Trips | Reviews | MLRG | Grants | Vball | Slack

Google

john snow

All | Images | News | Videos | Maps | More | Settings | Tools | View saved | SafeSearch

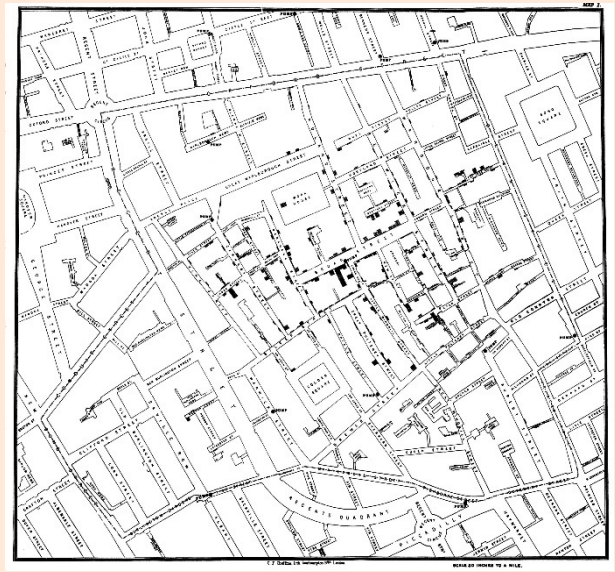
wallpaper | funny | season 6 | winter is coming meme | genderbent | let it snow | real life | ghost | wallpaper hd | the watch meme | his wolf | game throne | simpsons

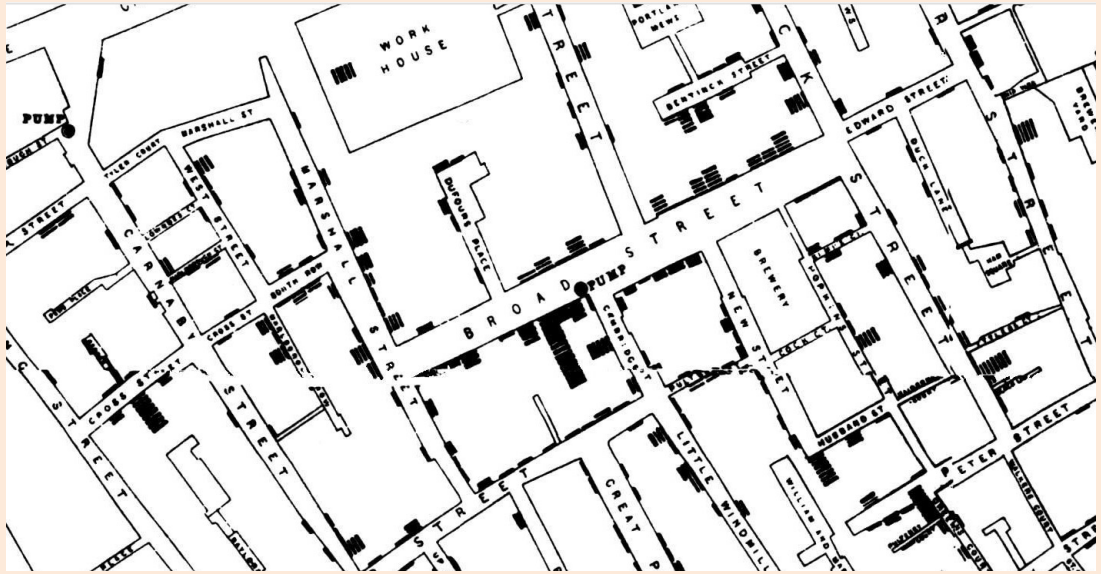
Did you mean: *jon* snow

bonus!

John Snow and Cholera Epidemic

- John Snow's 1854 spatial histogram of deaths from cholera:





- Found cluster of cholera deaths around a particular water pump.
  - Went against airborne theory, but pump later found to be contaminated.
  - "Father" of epidemiology.

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

Motivation for Density-Based Clustering

- Density-based clustering:
  - Clusters are defined by “dense” regions.
  - Examples in non-dense regions don’t get clustered.
    - Not trying to “partition” the space.

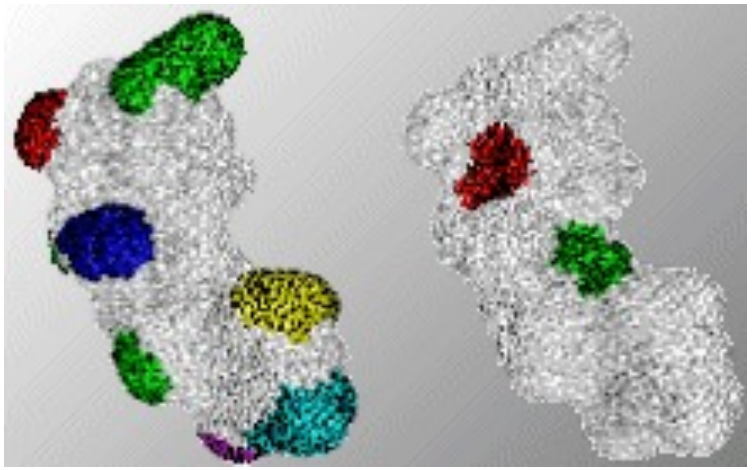
- Clusters can be non-convex:
  - Elephant clusters affected by vegetation, mountains, rivers, water access, etc.

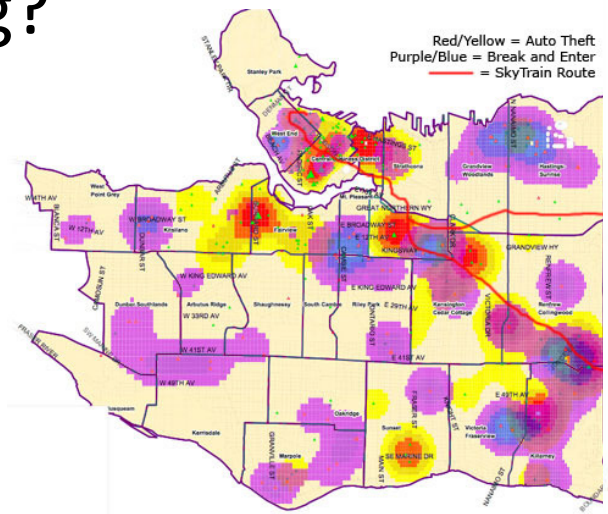
- It’s a non-parametric clustering method:
  - No fixed number of clusters ‘k’.
  - Clusters can become more complicated with more data.

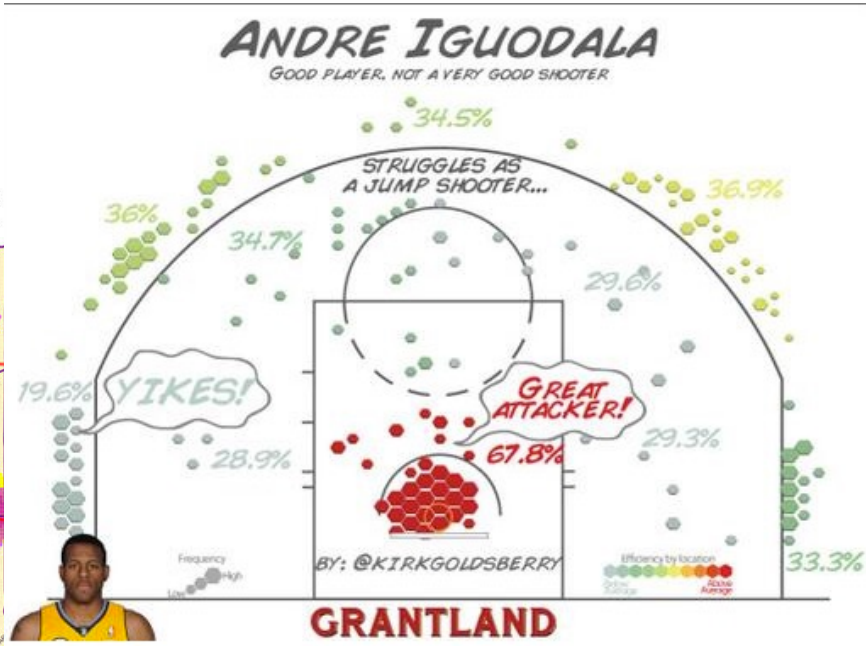
http://www.defenders.org/elephant/basic-facts

Other Potential Applications

- Where should taxis patrol?
- Where does Iguodala make/miss shots?
- Which products are similar to this one?
- Which pictures are in the same place?
- Where can proteins ‘dock’?
- Where are people tweeting?



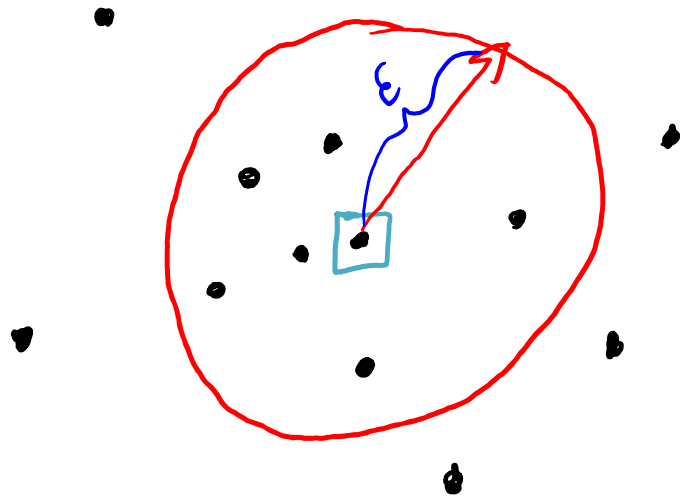




https://en.wikipedia.org/wiki/Cluster_analysis
https://www.flickr.com/photos/dbarefoot/420194128/
http://letsgowarriors.com/replacing-jarrett-jack/2013/10/04/
http://www.dbs.informatik.uni-muenchen.de/Forschung/KDD/Clustering/

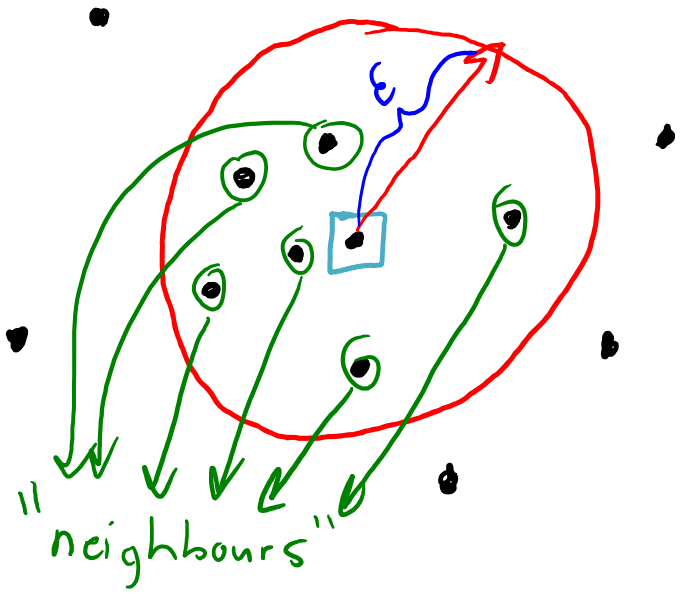
Density-Based Clustering

- Density-based clustering algorithm (DBSCAN) has two hyperparameters:
  - Epsilon (ε): distance we use to decide if another point is a "neighbour".



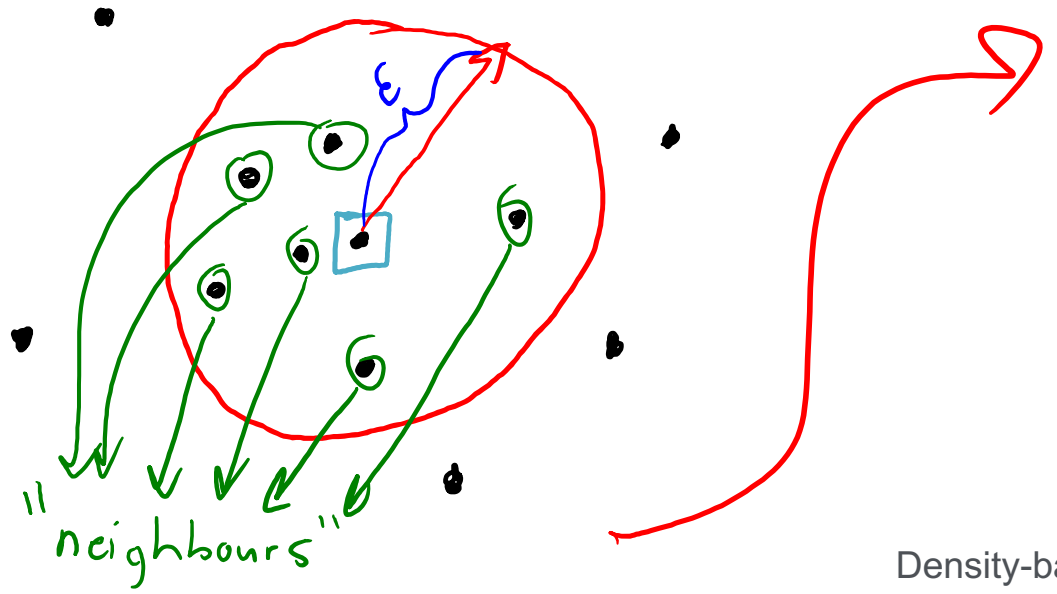
Density-Based Clustering

- Density-based clustering algorithm (DBSCAN) has two hyperparameters:
  - Epsilon (ε): distance we use to decide if another point is a "neighbour".



Density-Based Clustering

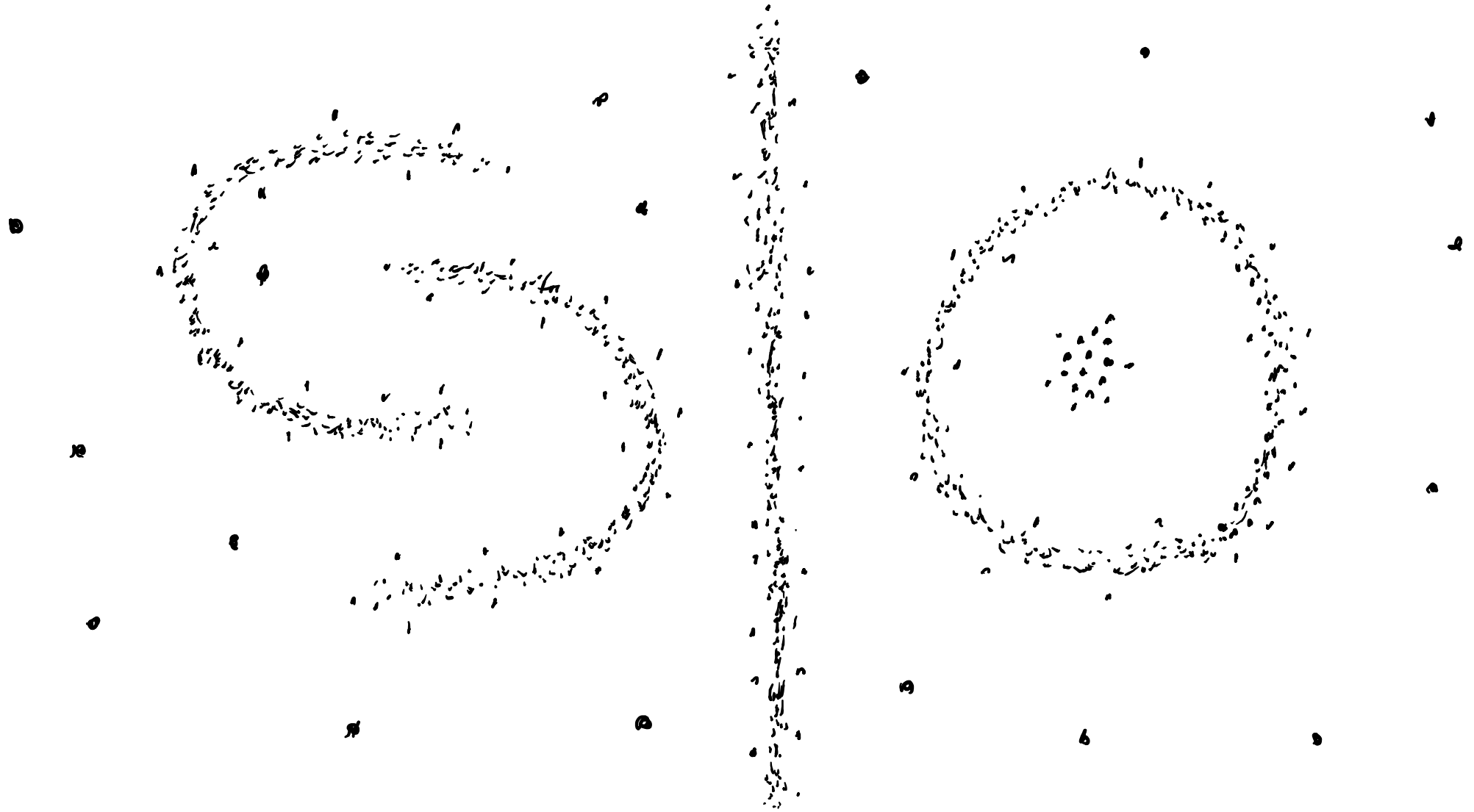
- Density-based clustering algorithm (DBSCAN) has two hyperparameters:
  - Epsilon (ε): distance we use to decide if another point is a "neighbour".
  - MinNeighbours: number of neighbours needed to say a region is "dense".
    - If you have at least minNeighbours "neighbours", you are called a "core" point.
- Main idea: merge all neighbouring core points to form clusters.



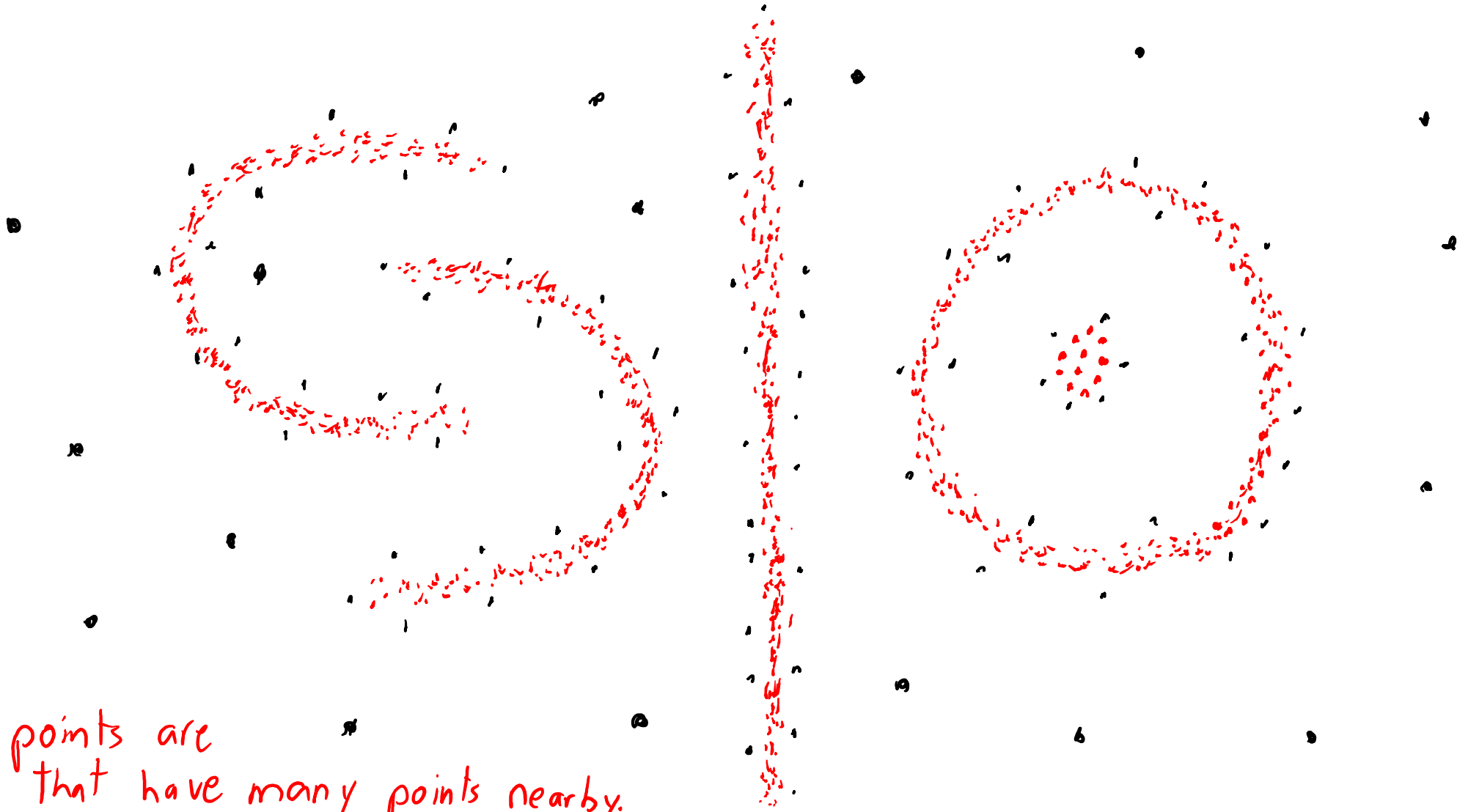
E.g., if minNeighbours = 3 then this is a "core" point since 6 points are "neighbours"

Density-based spatial clustering of applications with noise (DBSCAN)

Density-Based Clustering

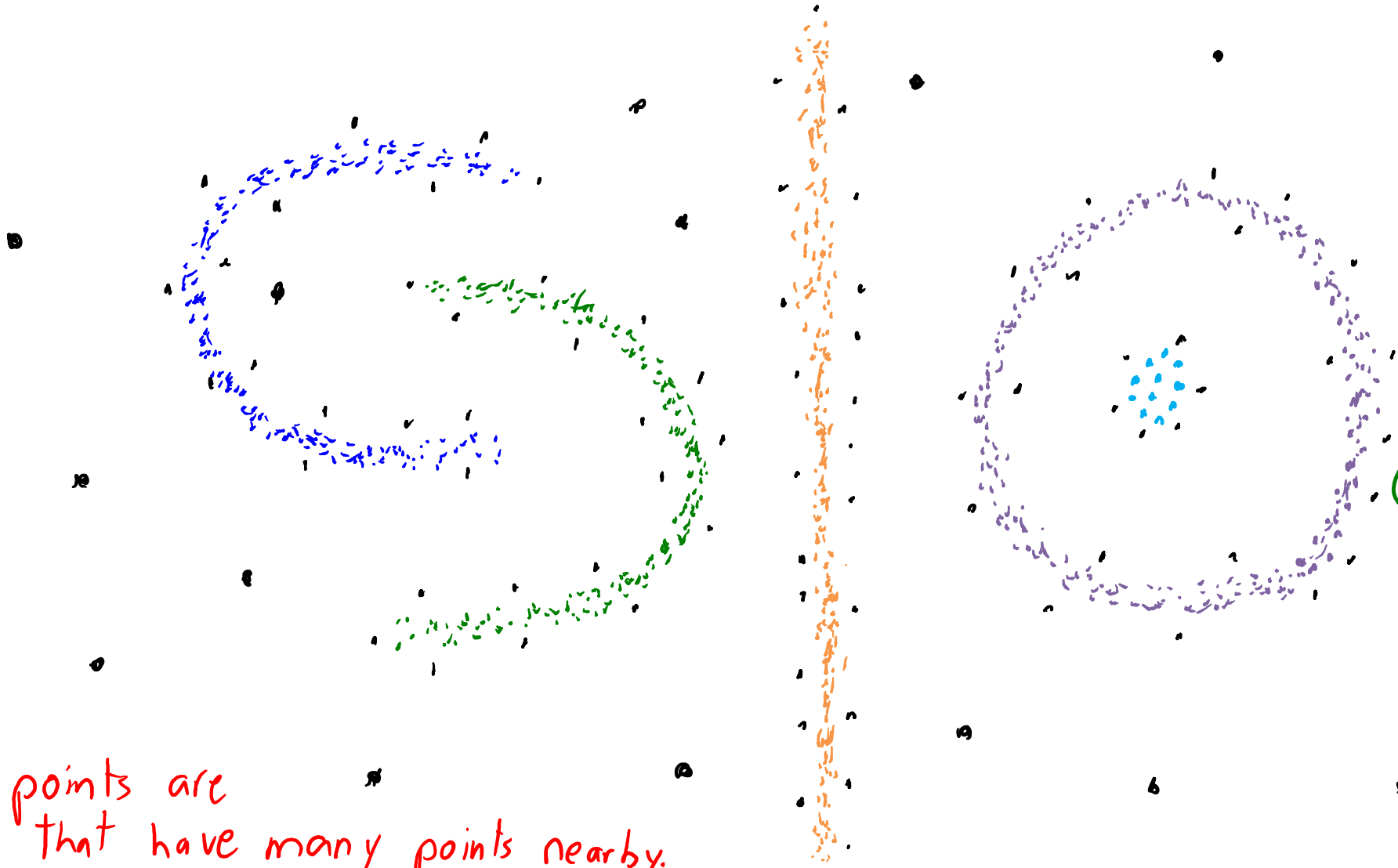


Density-Based Clustering



"Core" points are points that have many points nearby.

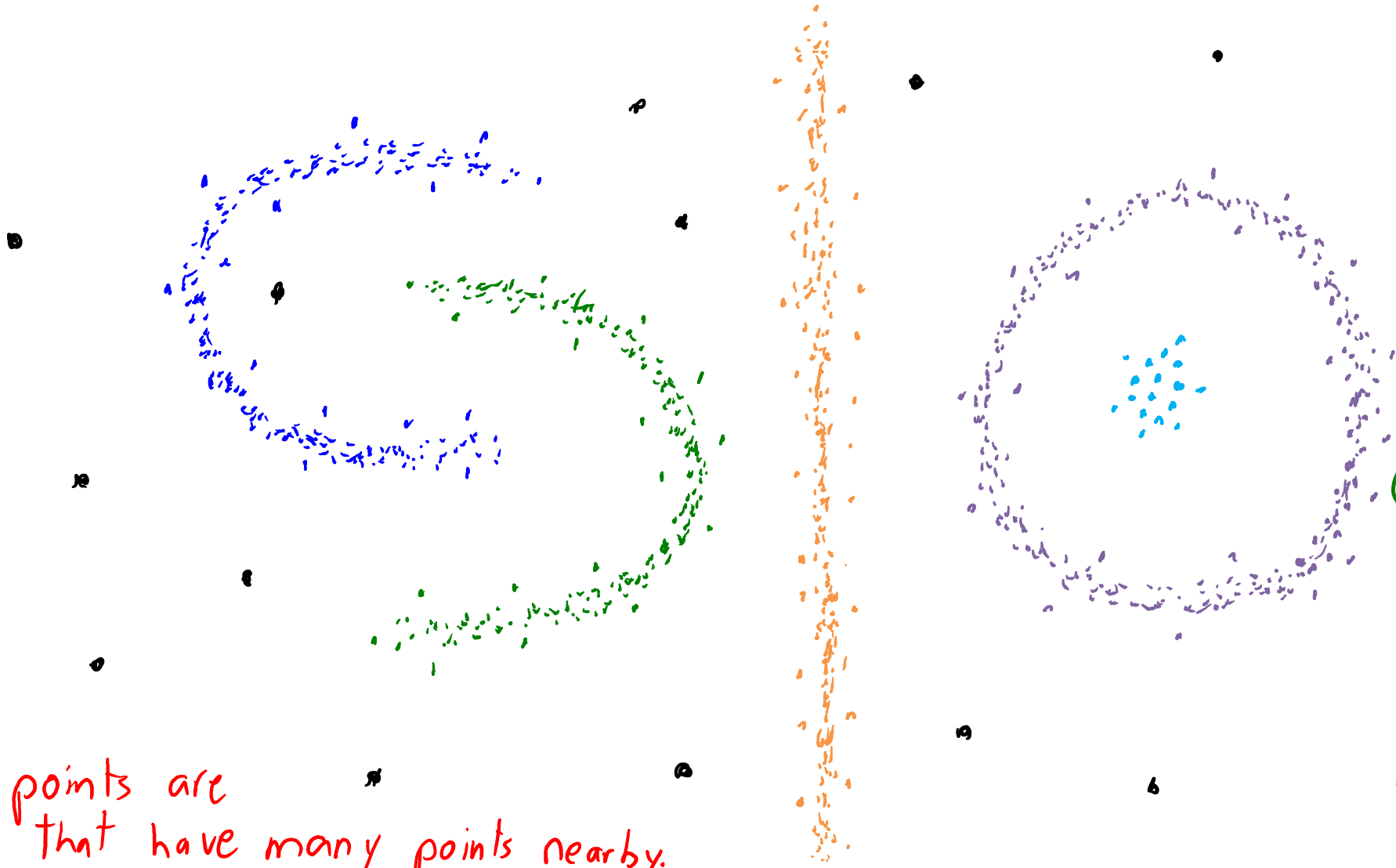
Density-Based Clustering



"Core" points are points that have many points nearby.

Clusters contain:
1. All "core" points that can be reached by following a sequence of core points.

Density-Based Clustering



"Core" points are points that have many points nearby.

Clusters contain:

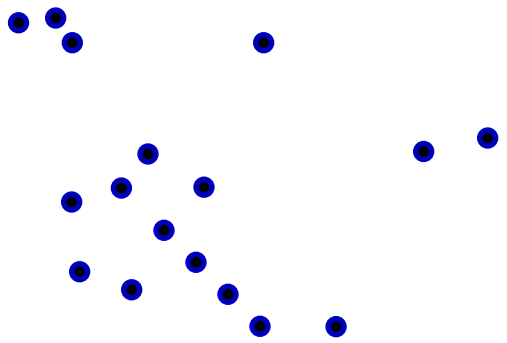
1. All "core" points that can be reached by following a sequence of core points.
2. All non-core neighbours of core points (boundary points)

Density-Based Clustering Pseudo-Code

- For each example x_i:
  - If x_i is already assigned to a cluster, do nothing.
  - Test whether x_i is a ‘core’ point (≥ minNeighbours examples within ‘ε’).
    - If x_i is not core point, do nothing (this could be an outlier).
    - If x_i is a core point, make a new cluster and call the “expand cluster” function.

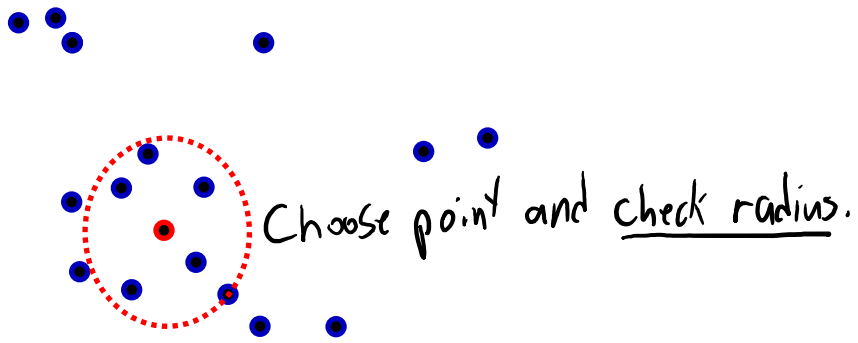
Density-Based Clustering Pseudo-Code

- "Expand cluster" function:
  - Assign to this cluster all x_j within distance 'ε' of core point x_i to this cluster.
  - For each new "core" point found, call "expand cluster" (recursively).



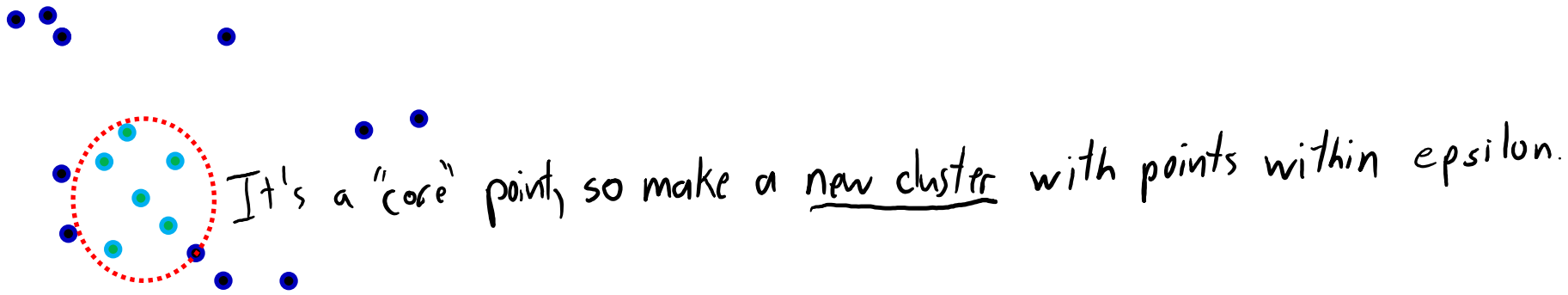
Density-Based Clustering Pseudo-Code

- "Expand cluster" function:
  - Assign to this cluster all x_j within distance 'ε' of core point x_i to this cluster.
  - For each new "core" point found, call "expand cluster" (recursively).



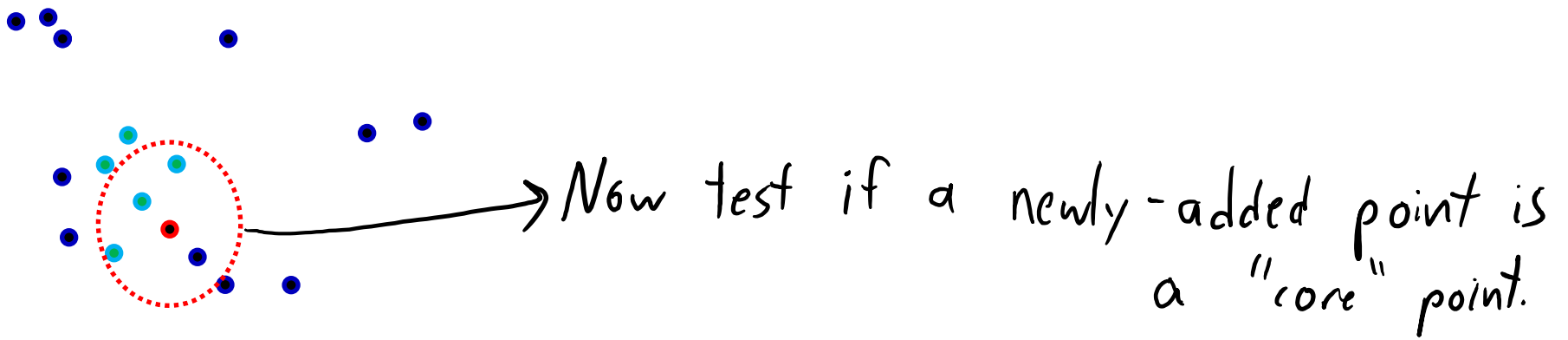
Density-Based Clustering Pseudo-Code

- "Expand cluster" function:
  - Assign to this cluster all x_j within distance 'ε' of core point x_i to this cluster.
  - For each new "core" point found, call "expand cluster" (recursively).



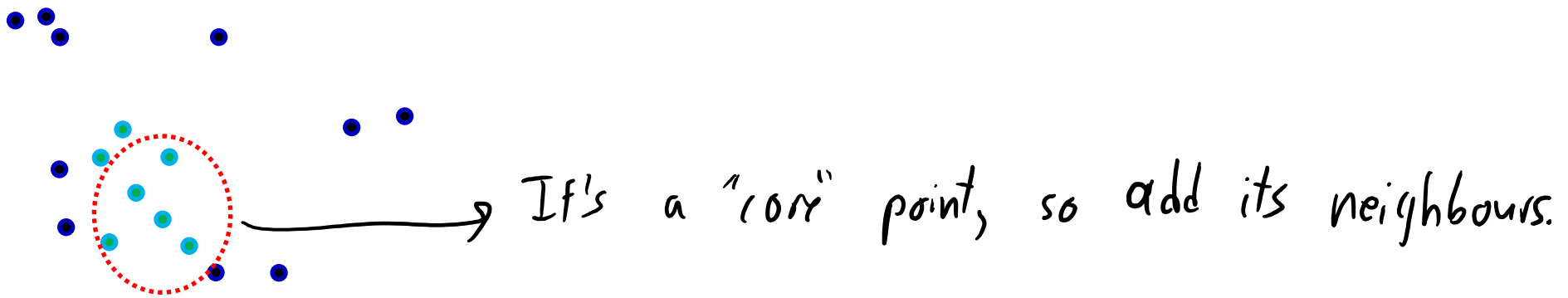
Density-Based Clustering Pseudo-Code

- "Expand cluster" function:
  - Assign to this cluster all x_j within distance 'ε' of core point x_i to this cluster.
  - For each new "core" point found, call "expand cluster" (recursively).



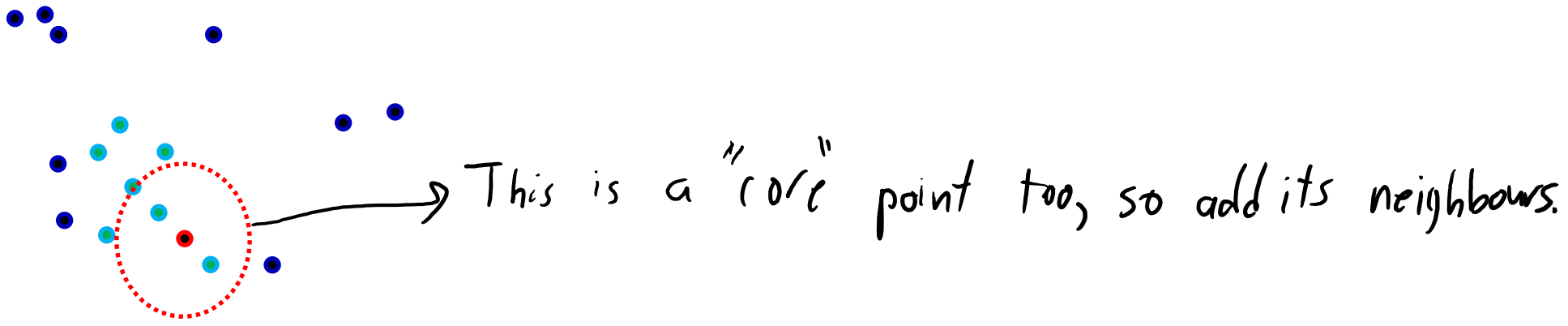
Density-Based Clustering Pseudo-Code

- "Expand cluster" function:
  - Assign to this cluster all x_j within distance 'ε' of core point x_i to this cluster.
  - For each new "core" point found, call "expand cluster" (recursively).



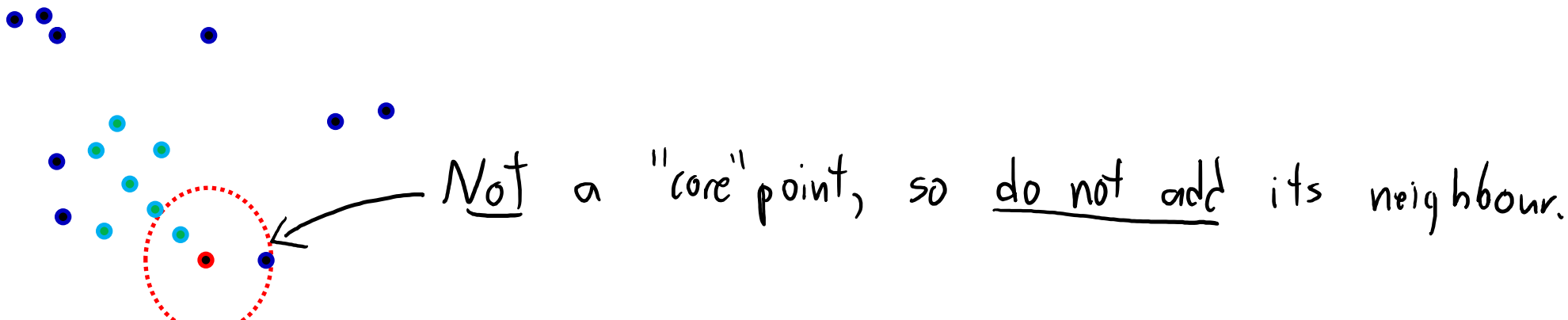
Density-Based Clustering Pseudo-Code

- "Expand cluster" function:
  - Assign to this cluster all x_j within distance 'ε' of core point x_i to this cluster.
  - For each new "core" point found, call "expand cluster" (recursively).



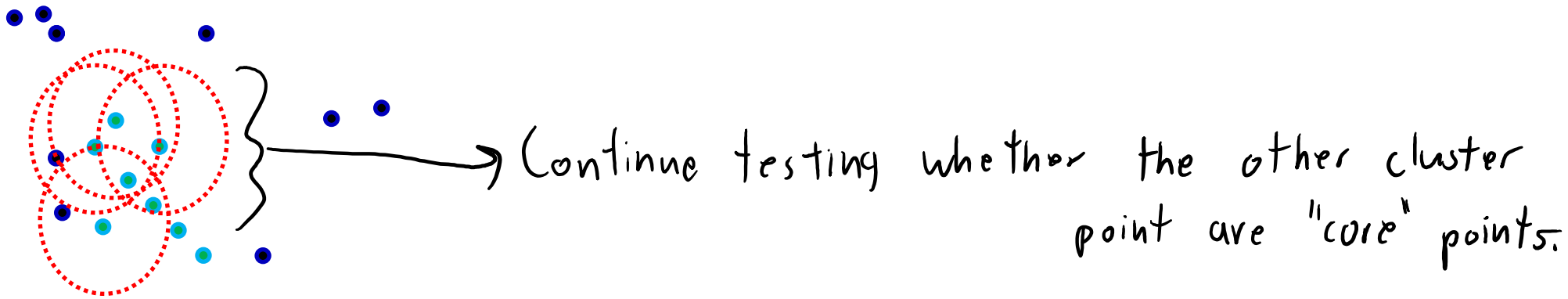
Density-Based Clustering Pseudo-Code

- "Expand cluster" function:
  - Assign to this cluster all x_j within distance 'ε' of core point x_i to this cluster.
  - For each new "core" point found, call "expand cluster" (recursively).



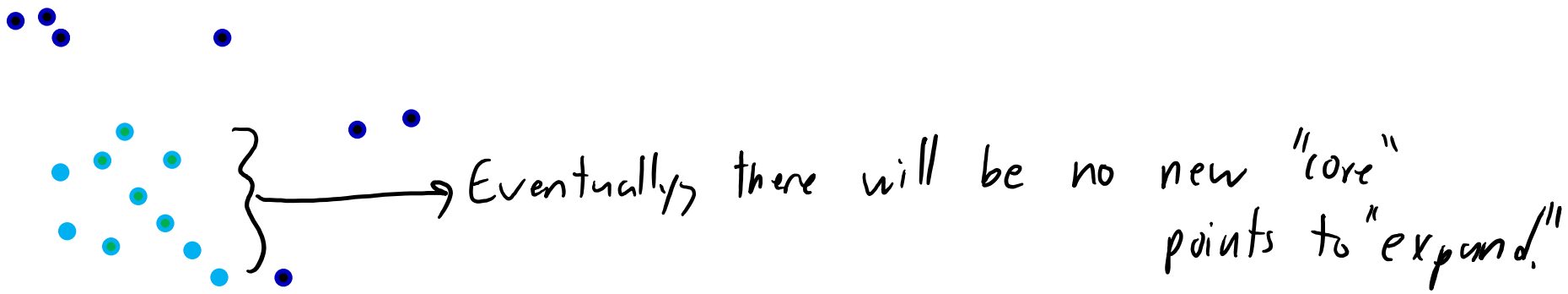
Density-Based Clustering Pseudo-Code

- "Expand cluster" function:
  - Assign to this cluster all x_j within distance 'ε' of core point x_i to this cluster.
  - For each new "core" point found, call "expand cluster" (recursively).



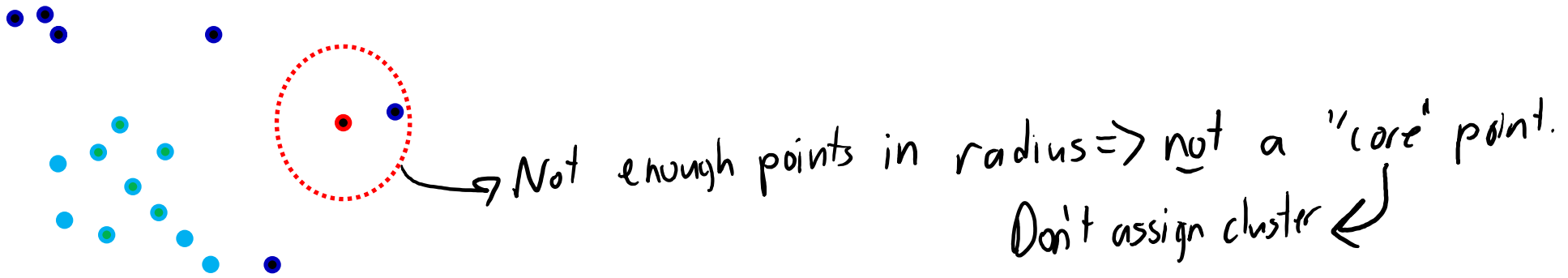
Density-Based Clustering Pseudo-Code

- "Expand cluster" function:
  - Assign to this cluster all x_j within distance 'ε' of core point x_i to this cluster.
  - For each new "core" point found, call "expand cluster" (recursively).



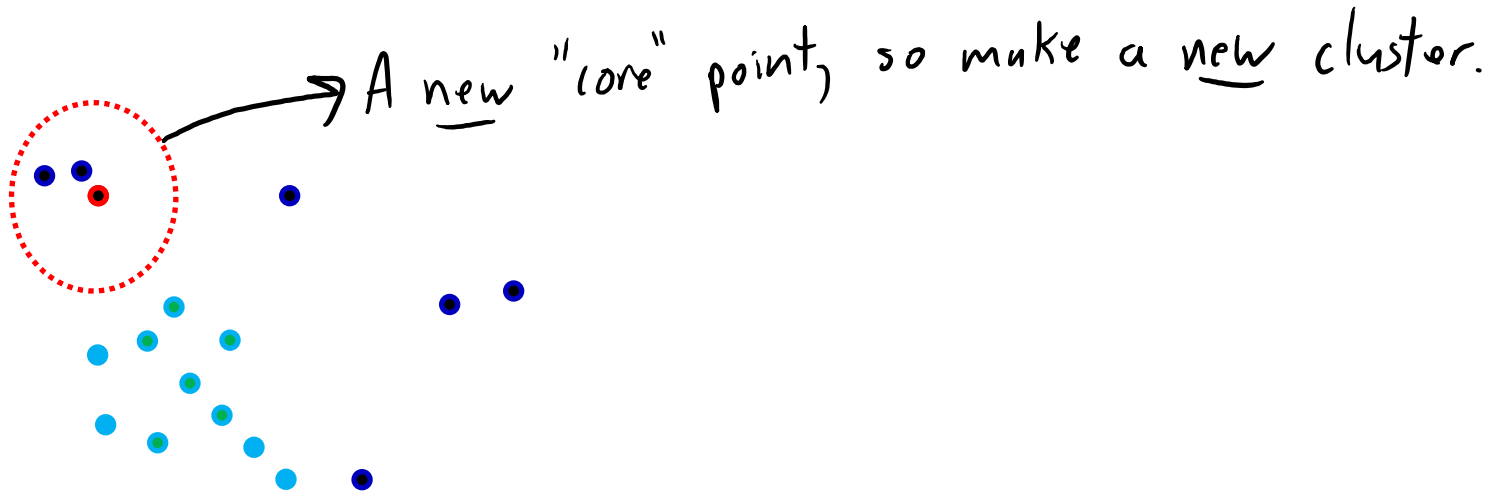
Density-Based Clustering Pseudo-Code

- "Expand cluster" function:
  - Assign to this cluster all x_j within distance 'ε' of core point x_i to this cluster.
  - For each new "core" point found, call "expand cluster" (recursively).



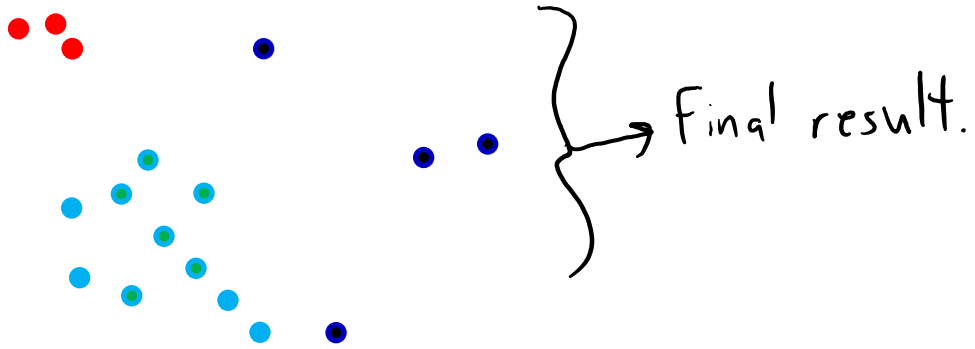
Density-Based Clustering Pseudo-Code

- "Expand cluster" function:
  - Assign to this cluster all x_j within distance 'ε' of core point x_i to this cluster.
  - For each new "core" point found, call "expand cluster" (recursively).

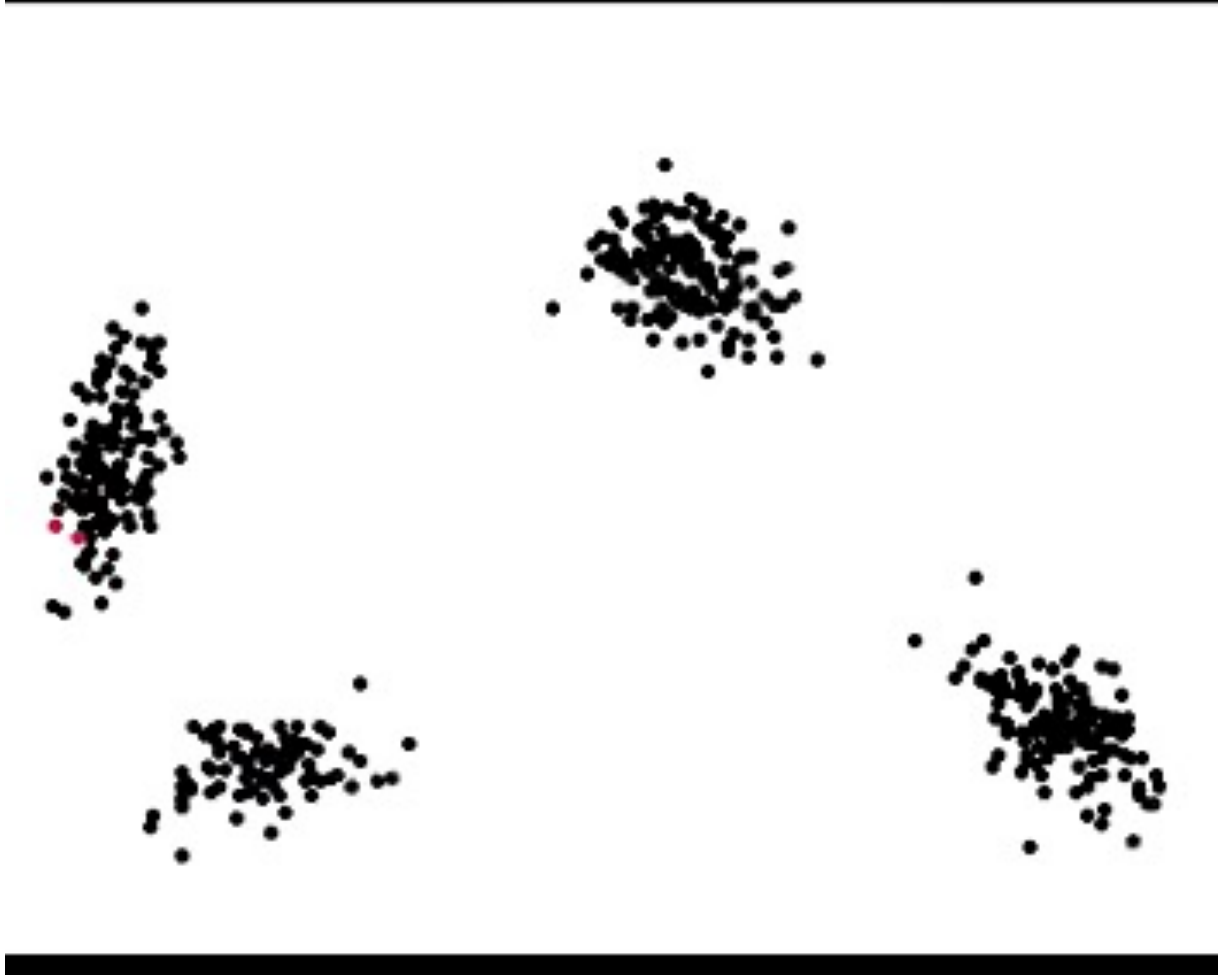


Density-Based Clustering Pseudo-Code

- "Expand cluster" function:
  - Assign to this cluster all x_j within distance 'ε' of core point x_i to this cluster.
  - For each new "core" point found, call "expand cluster" (recursively).



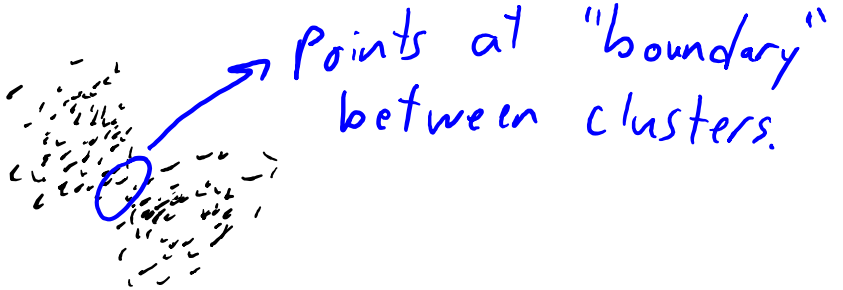
Density-Based Clustering in Action



[Interactive demo]

Density-Based Clustering Issues

- Some points are not assigned to a cluster.
  - Good or bad, depending on the application.
- Ambiguity of "non-core" (boundary) points:



- Sensitive to the choice of ε and minNeighbours.
  - Original paper proposed an "elbow" method (see bonus slide).
  - Otherwise, not sensitive to initialization (except for boundary points).

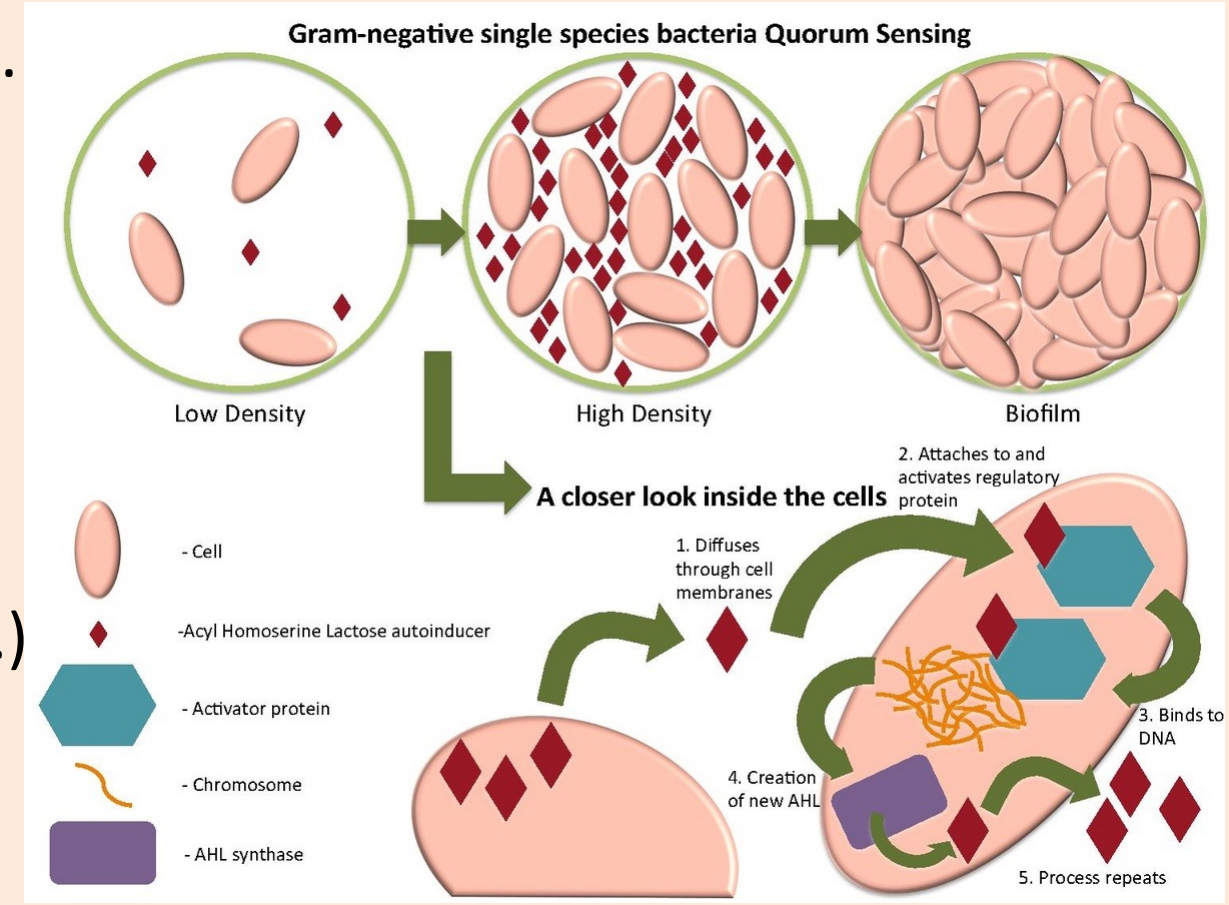
- If you get a new example, finding cluster is expensive.
  - Need to compute distances to core points (or maybe all training points).
- In high-dimensions, need a lot of points to 'fill' the space.

bonus!

Density-Based Clustering in Nature

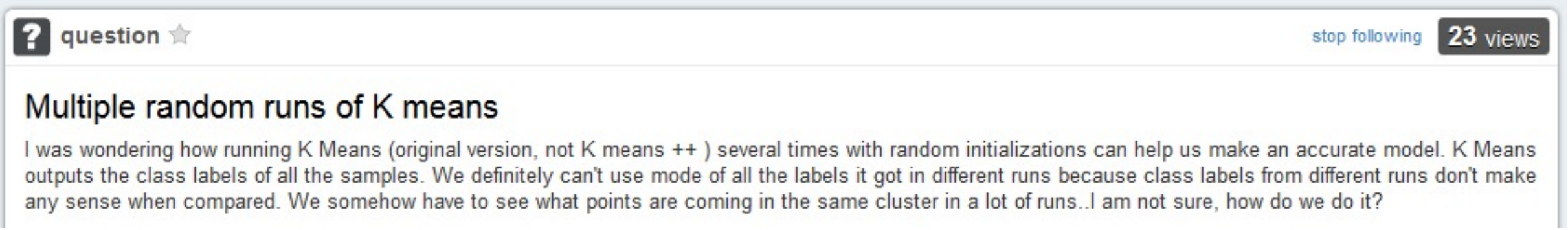
- Quorum sensing:
  - Bacteria continuously release a particular molecule.
  - They have sensors for this molecule.

- If sensors become very active:
  - It means cell density is high.
  - Causes cascade of changes in cells. (Some cells "stick together" to form a physical cluster via "biofilm".)



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

Ensemble Clustering


question

stop following 23 views

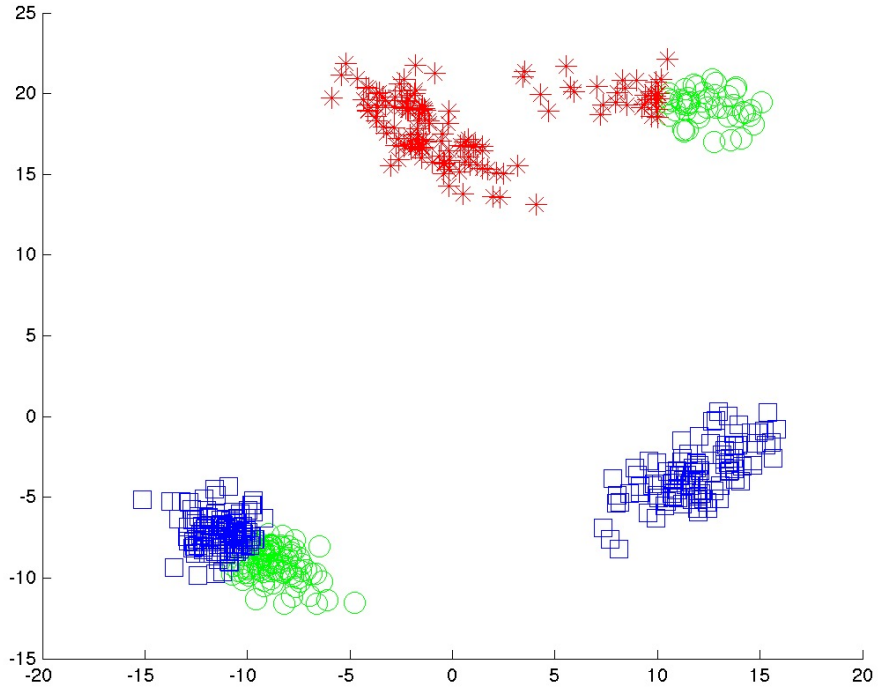
Multiple random runs of K means

I was wondering how running K Means (original version, not K means ++) several times with random initializations can help us make an accurate model. K Means outputs the class labels of all the samples. We definitely can't use mode of all the labels it got in different runs because class labels from different runs don't make any sense when compared. We somehow have to see what points are coming in the same cluster in a lot of runs..I am not sure, how do we do it?

- We can consider ensemble methods for clustering.
  - "Consensus clustering"
- It's a good/important idea:
  - Bootstrapping is widely-used.
  - "Do clusters change if the data was slightly different?"
- But we need to be careful about how we combine models.

Ensemble Clustering

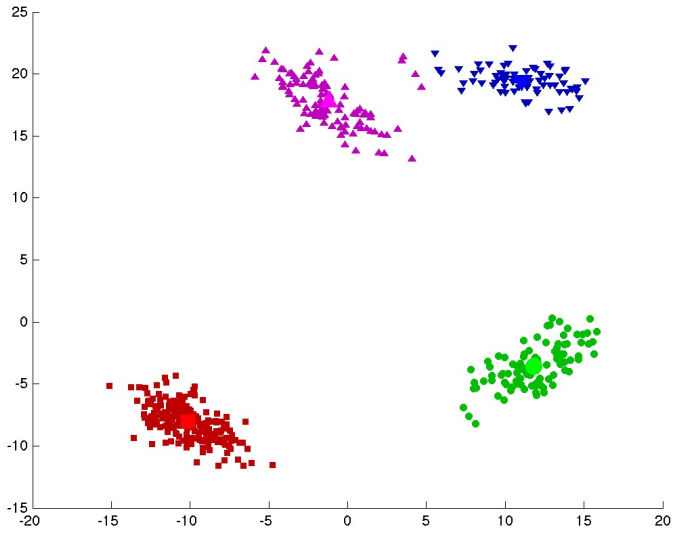
- E.g., run k-means 20 times and then cluster using the mode of each $\hat{y}_i$.
- Normally, averaging across models doing different things is good.

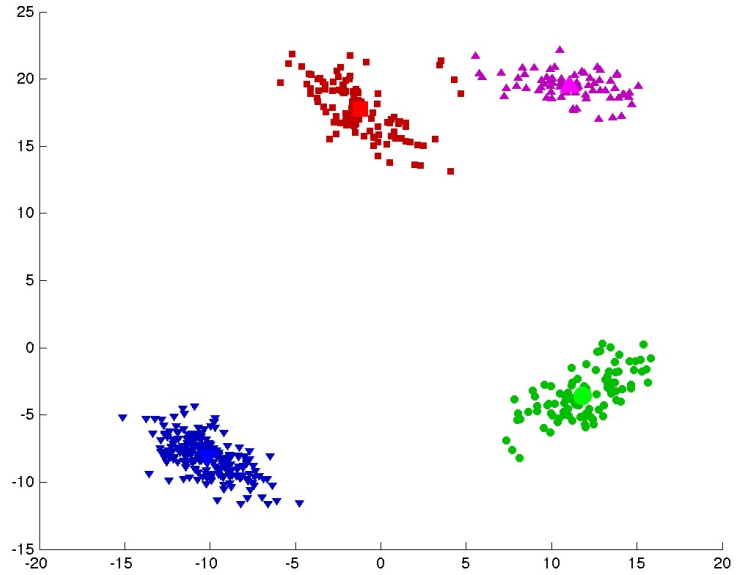


- But this is a bad ensemble method: worse than k-means on its own.

Label Switching Problem

- This doesn't work because of "label switching" problem:
  - The cluster labels $\hat{y}_i$ are meaningless.
  - We could get same clustering with permuted labels ("exchangeable"):

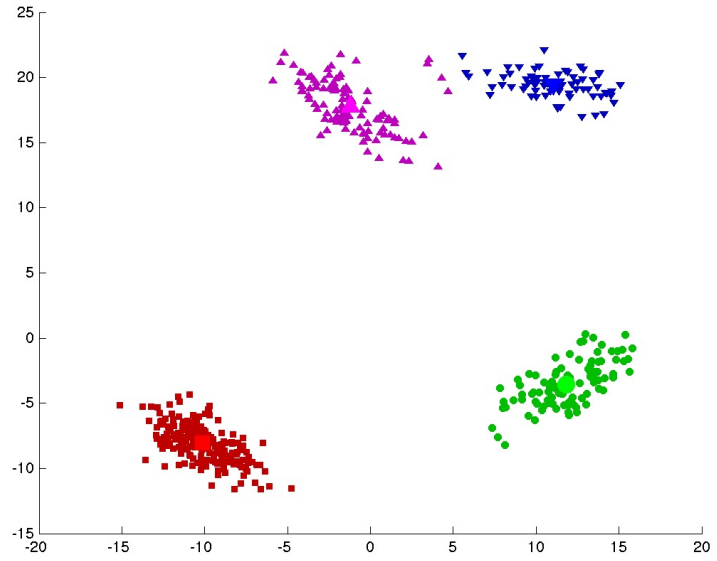


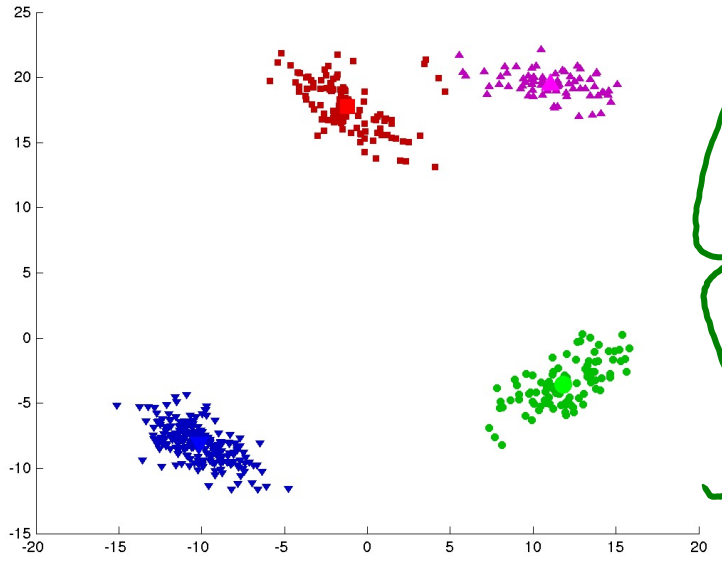


  - All $\hat{y}_i$ become equally likely as number of initializations increases.

Addressing Label Switching Problem

- Ensembles can't depend on label "meaning":
  - Don't ask "is point x_i in red square cluster?", which is meaningless.
  - Ask "is point x_i in the same cluster as x_j?", which is meaningful.



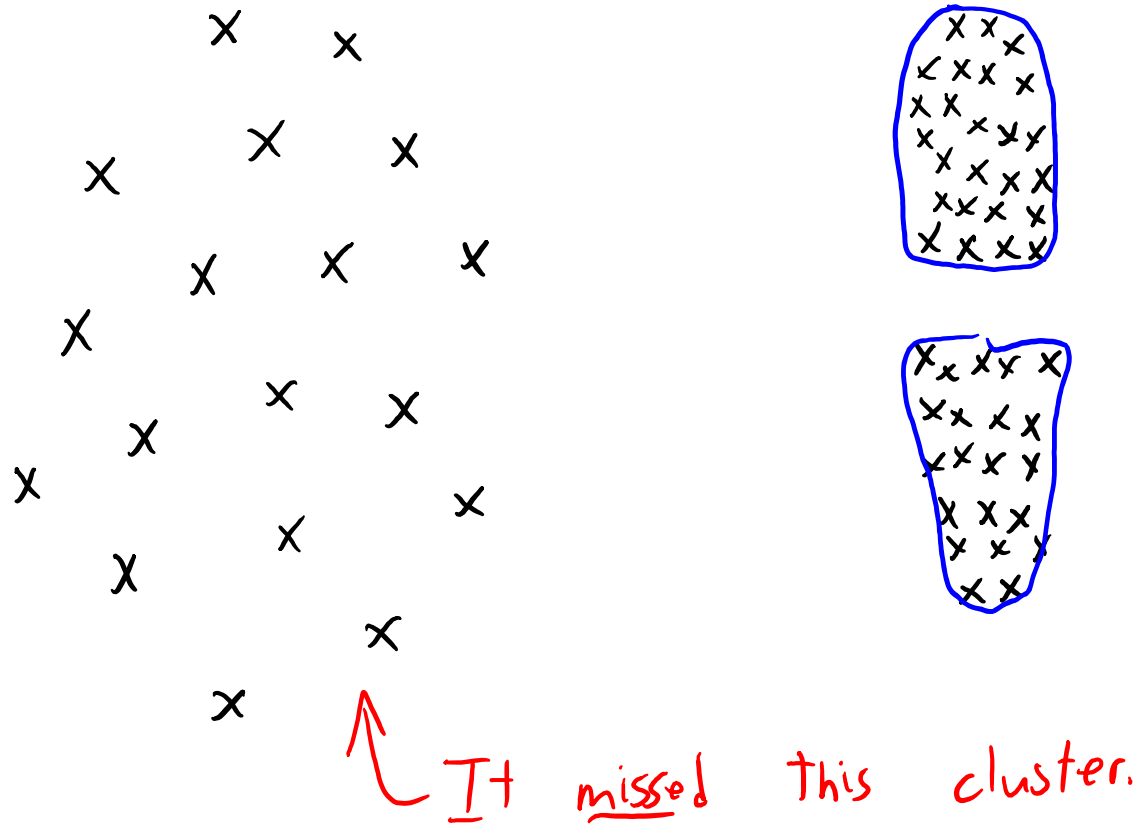


Different permutation of labels but same groups of points.

  - Bonus slides give an example method ("UBClustering").

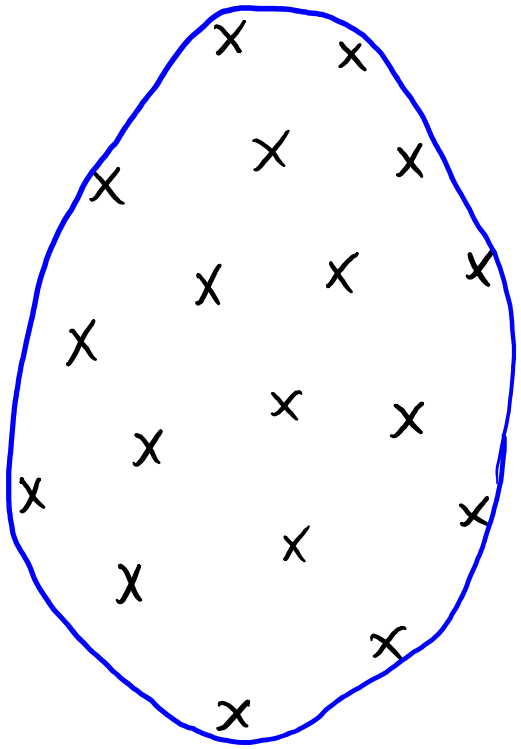
Differing Densities

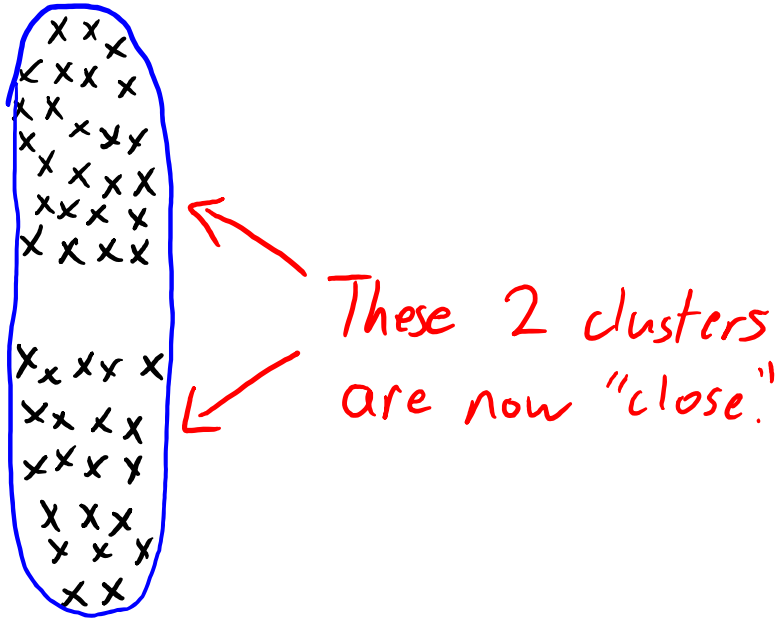
- Consider density-based clustering on this data:



Differing Densities

- Increase epsilon and run it again:

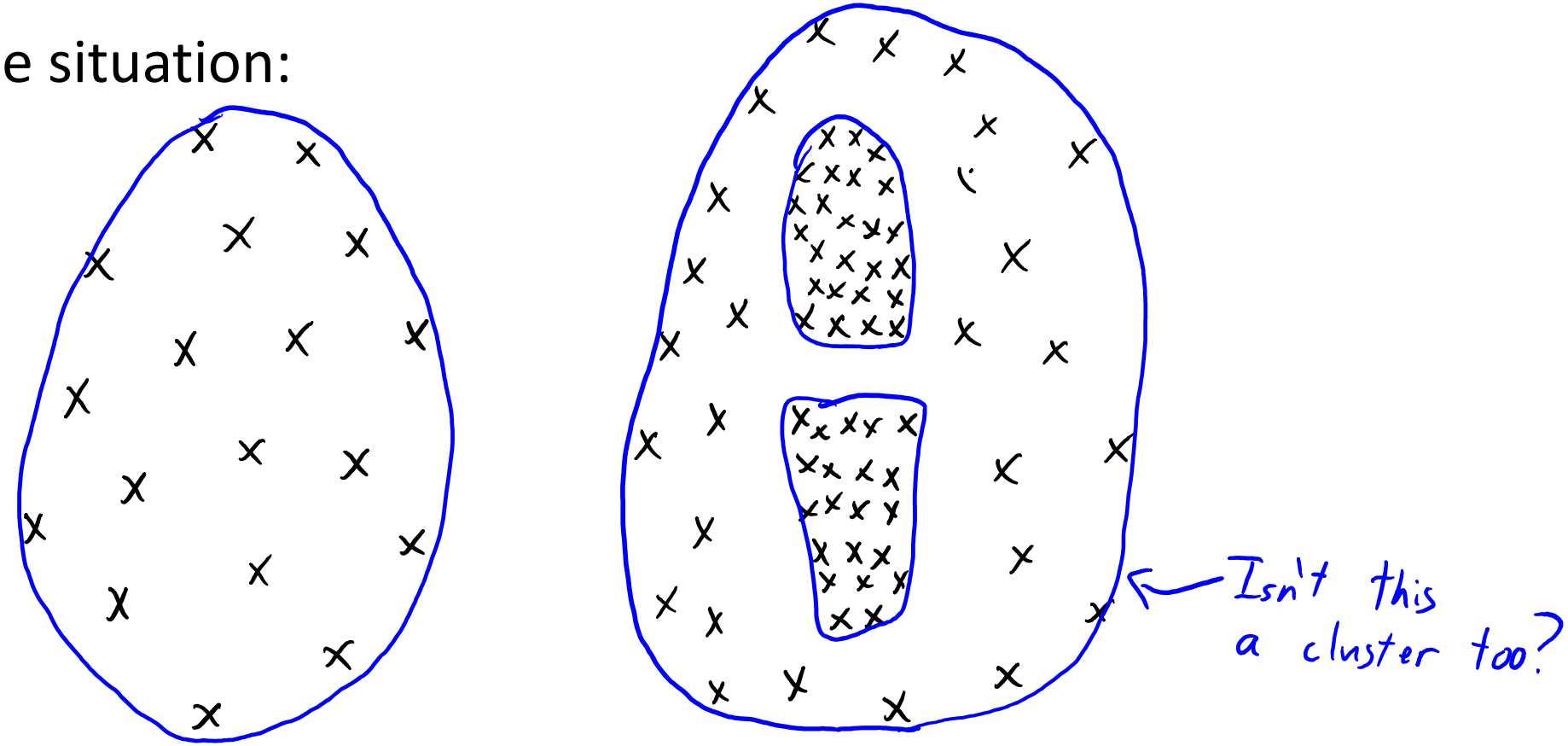




- There may be no density level that gives you 3 clusters.

Differing Densities

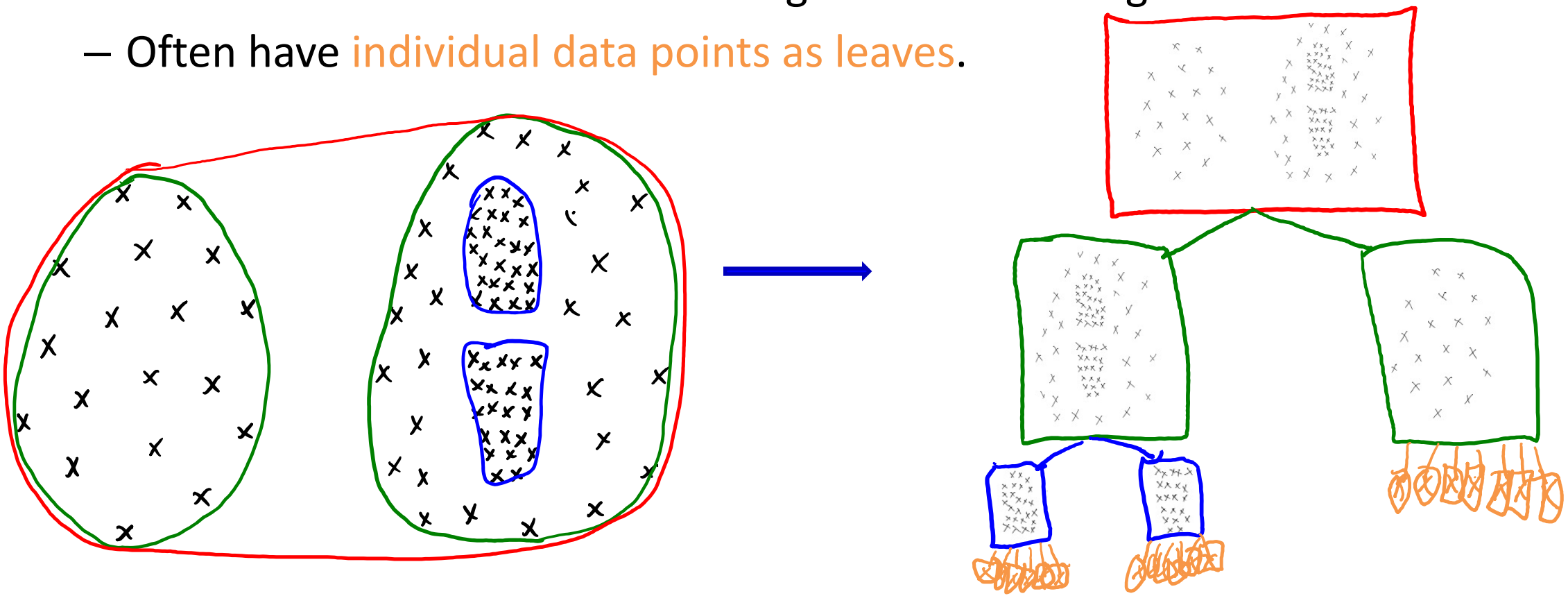
- Here is a worse situation:



- Now you need to choose between coarse/fine clusters.
- Instead of fixed clustering, we often want hierarchical clustering.

Hierarchical Clustering

- Hierarchical clustering produces a tree of clusterings.
  - Each node in the tree splits the data into 2 or more clusters.
  - Much more information than using a fixed clustering.
  - Often have individual data points as leaves.



GIF

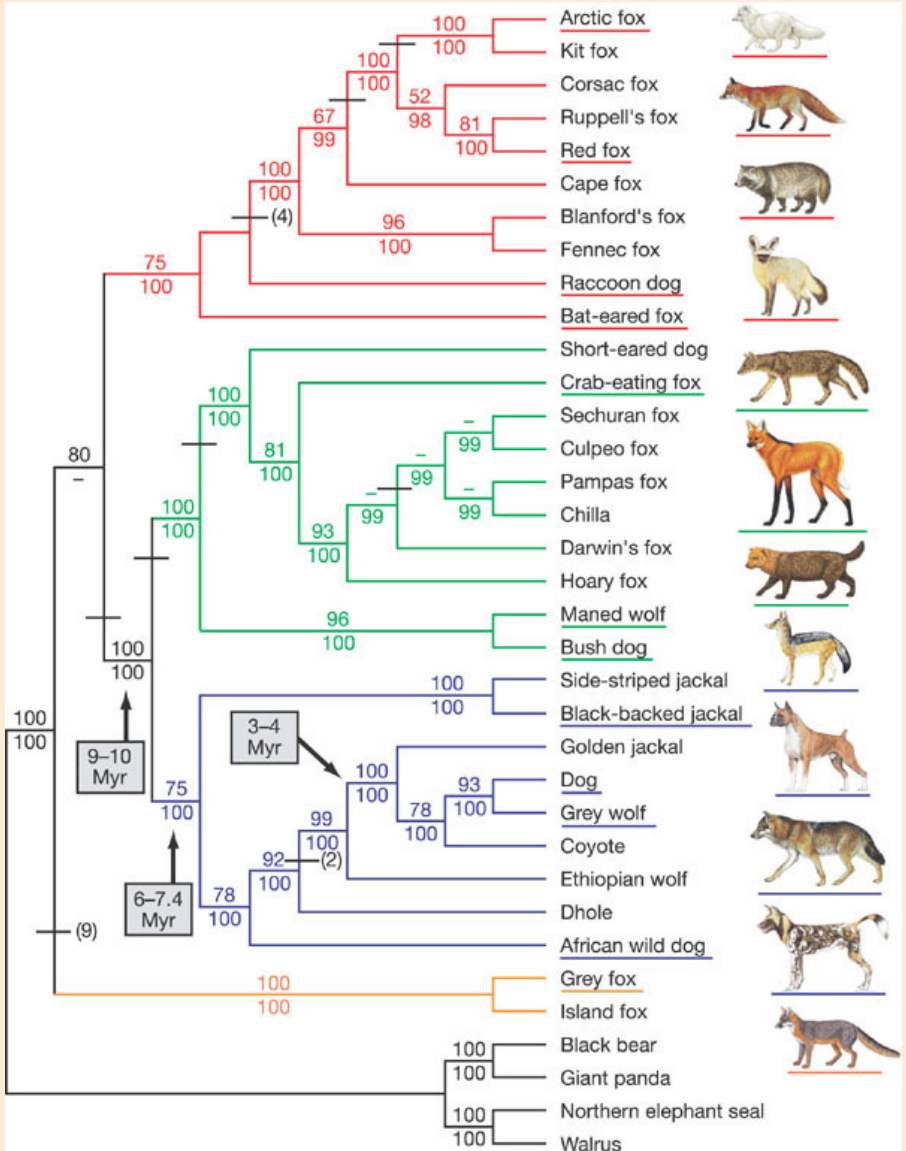
bonus!

Application: Phylogenetics

- We sequence genomes of a set of organisms.
- Can we construct the “tree of life”?

- Comments on this application:
  - On the right are individuals.
  - As you go left, clusters merge.
  - Merges are ‘common ancestors’.

- More useful information in the plot:
  - Line lengths: chosen here to approximate time.
  - Numbers: #clusterings across bootstrap samples.
  - ‘Outgroups’ (walrus, panda) are a sanity check.

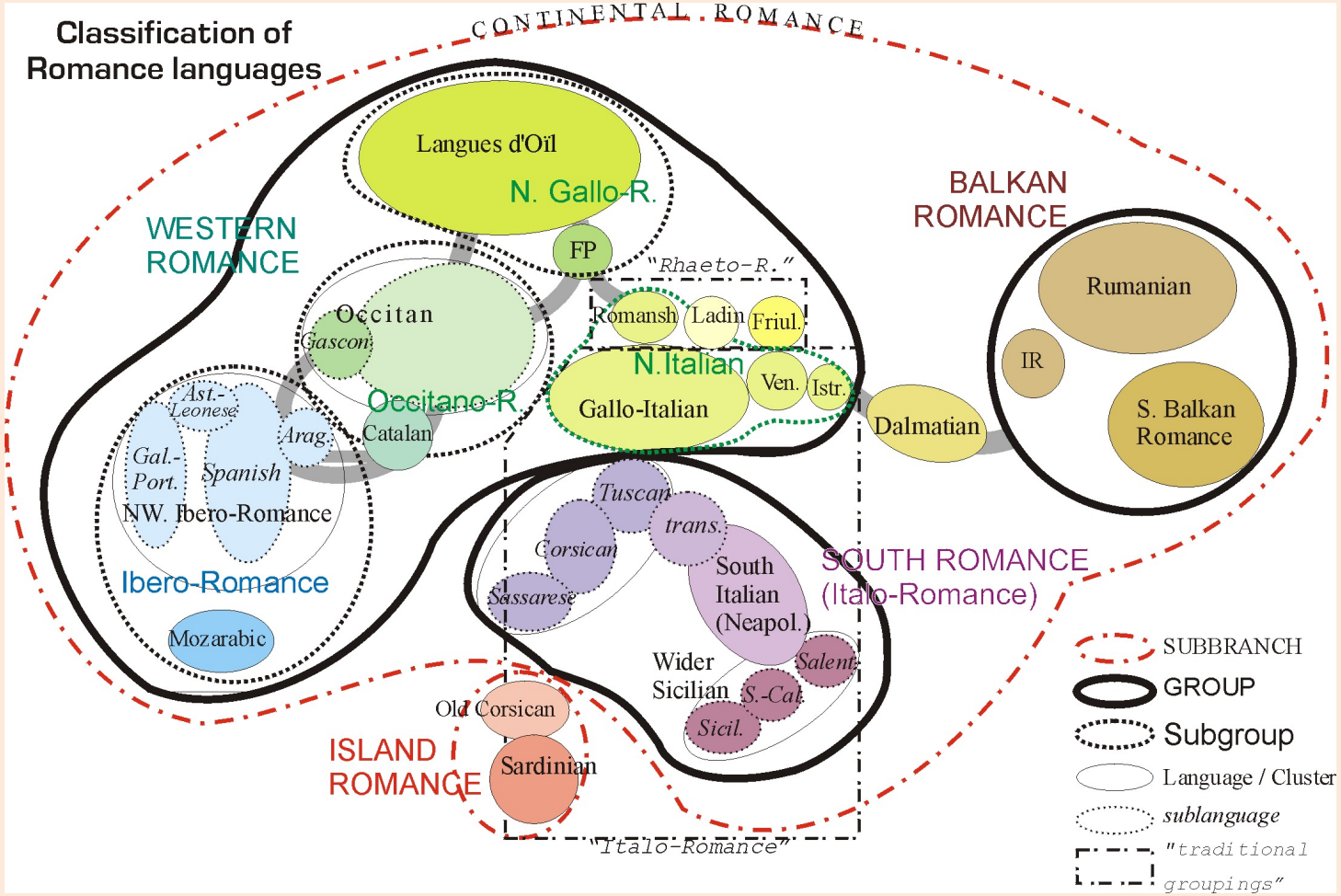


http://www.nature.com/nature/journal/v438/n7069/fig_tab/nature04338_F10.html

bonus!

Application: Phylogenetics

- Comparative method in linguistics studies evolution of languages:

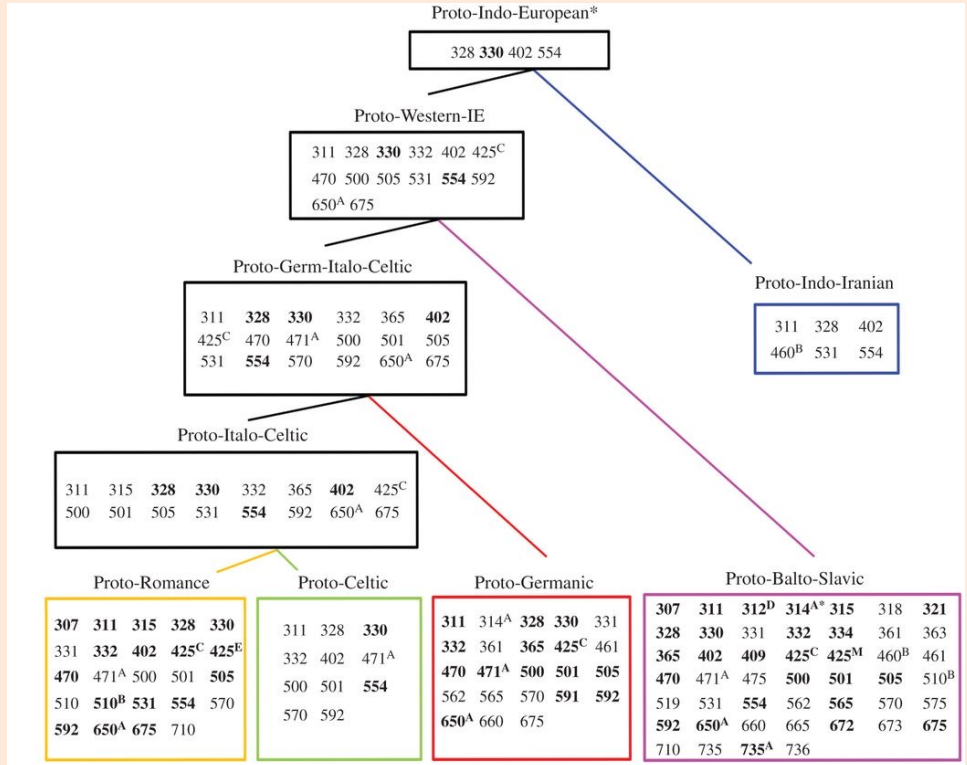


https://en.wikipedia.org/wiki/Comparative_method_(linguistics)

bonus!

Application: Phylogenetics

- January 2016: evolution of fairy tales.
  - Evidence that “Devil and the Smith” goes back to bronze age.
  - “Beauty and the Beast” published in 1740, but might be 2500-6000 years old.



International tale types

307	The Princess in the Coffin	**409**	The Girl as Wolf	**562**	The Spirit in the Blue Light
311	Rescue by Sister	**425C**	Beauty and the Beast	**565**	The Magic Mill
312D	Rescue by the Brother	**425E**	The Enchanted Husband	**570**	The Rabbit-Herd
314A	The Shepherd and the Giants	**425M**	The Snake Bridegroom	**575**	The Prince's Wings
314A*	Animal Helper in the Flight	**460B**	The Journey	**591**	The Thieving Pot
315	The Faithless Sister	**461**	Three Hairs	**592**	The Dance Among Thorns
318	The Faithless Wife	**470**	Friends in Life and Death	**650A**	Strong John
321	Eyes Recovered from Witch	**471A**	The Monk and the Bird	**660**	The Three Doctors
328	The Boy Steals Ogre's Treasure	**475**	The Man as the Heater	**665**	The Man who Flew and Swam
330	The Smith and the Devil	**500**	Supernatural Helper	**672**	The Serpent's Crown
331	The Spirit in the Bottle	**501**	The Three Old Spinning Women	**673**	The White Serpent's Flesh
332	Godfather Death	**505**	The Grateful Dead	**675**	The Lazy Boy
334	Household of the Witch	**510**	Cinderella and Peau d'Âne	**710**	Our Lady's Child
361	Bear Skin	**510B**	Peau d'Asne	**735**	The Rich and the Poor Man
363	The Corpse-Eater	**519**	The Strong Woman as Bride	**735A**	Bad Luck Imprisoned
365	The Dead Bridegroom	**531**	The Clever Horse	**736**	Luck and Wealth
402	The Animal Bride	**554**	The Grateful Animals		

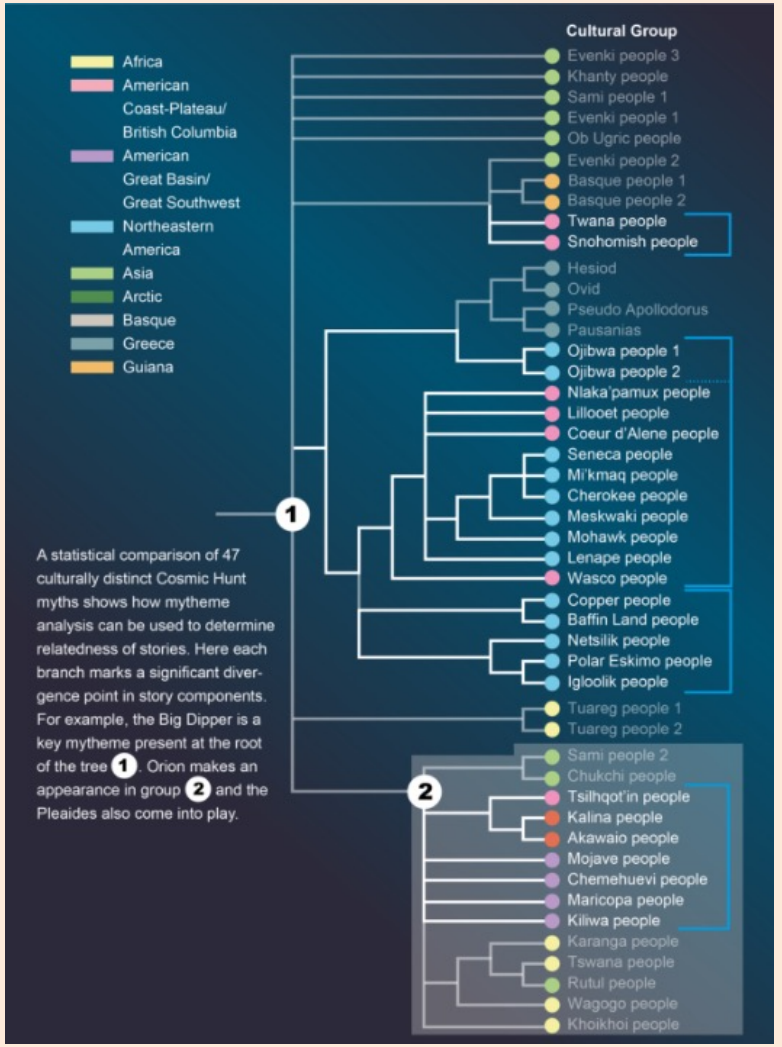
http://rsos.royalsocietypublishing.org/content/3/1/150645

bonus!

Application: Phylogenetics

- January 2016: evolution of fairy tales.
  - Evidence that “Devil and the Smith” goes back to bronze age.
  - “Beauty and the Beast” published in 1740, but might be 2500-6000 years old.

- September 2016: evolution of myths.
  - “Cosmic hunt” story:
    - Person hunts animal that becomes constellation.
      - Previously known to be at least 15,000 years old.
    - May go back to paleololithic period.

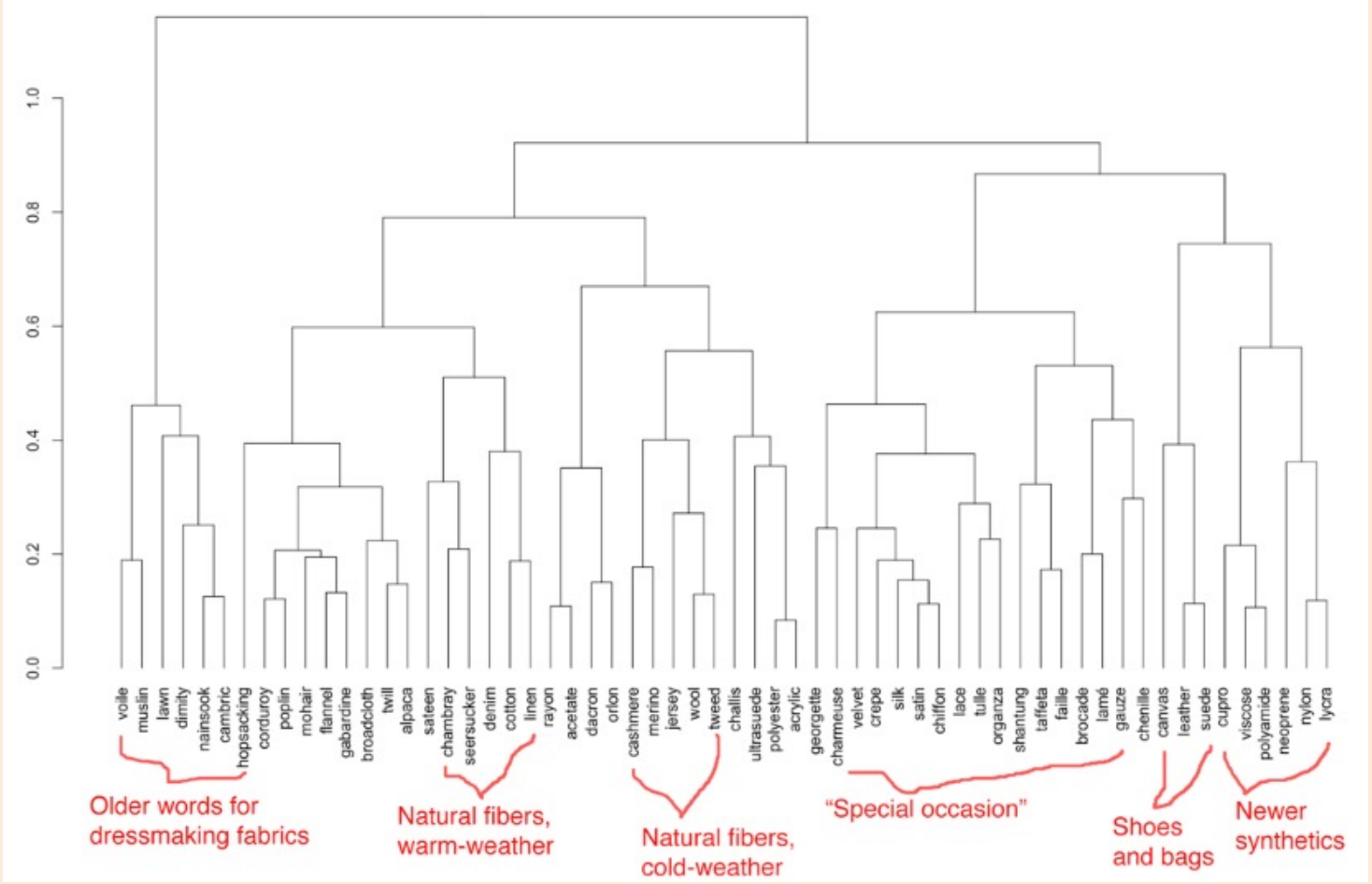


http://www.nature.com/nature/journal/v438/n7069/fig_tab/nature04338_F10.html

bonus!

Application: Fashion?

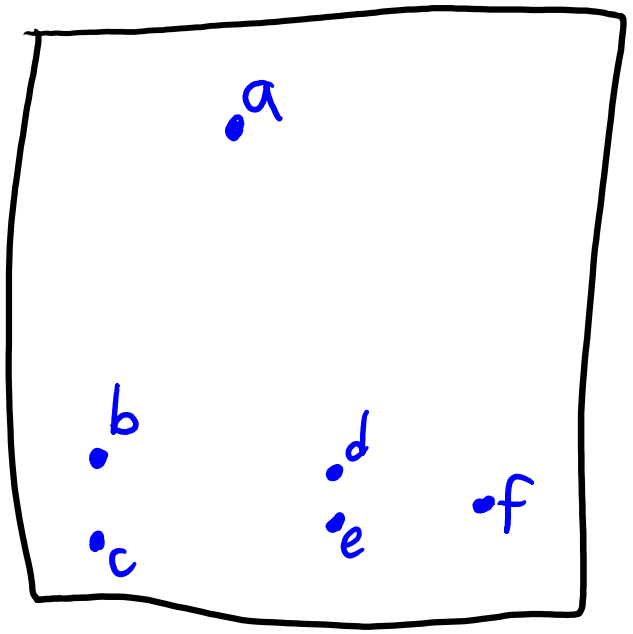
- Hierarchical clustering of clothing material words in Vogue:



http://dh.library.yale.edu/projects/vogue/fabricspace/

Agglomerative (Bottom-Up) Clustering

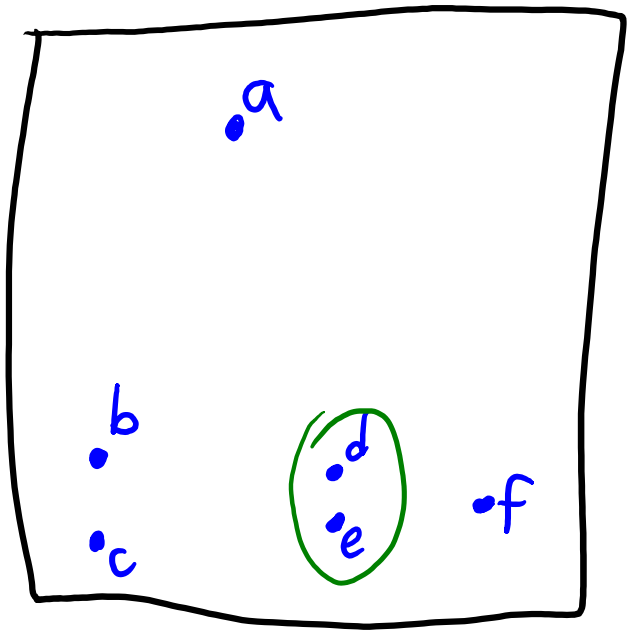
- Most common hierarchical method: agglomerative clustering.
  1. Starts with each point in its own cluster.

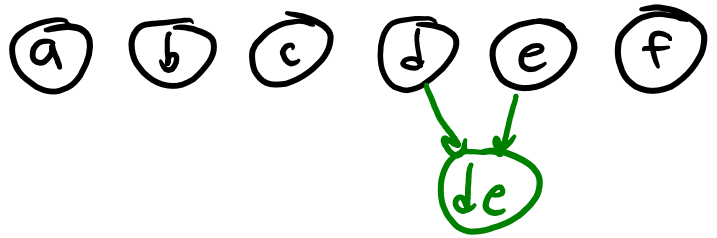


https://en.wikipedia.org/wiki/Hierarchical_clustering

Agglomerative (Bottom-Up) Clustering

- Most common hierarchical method: agglomerative clustering.
  1. Starts with each point in its own cluster.
  2. Each step merges the two “closest” clusters.

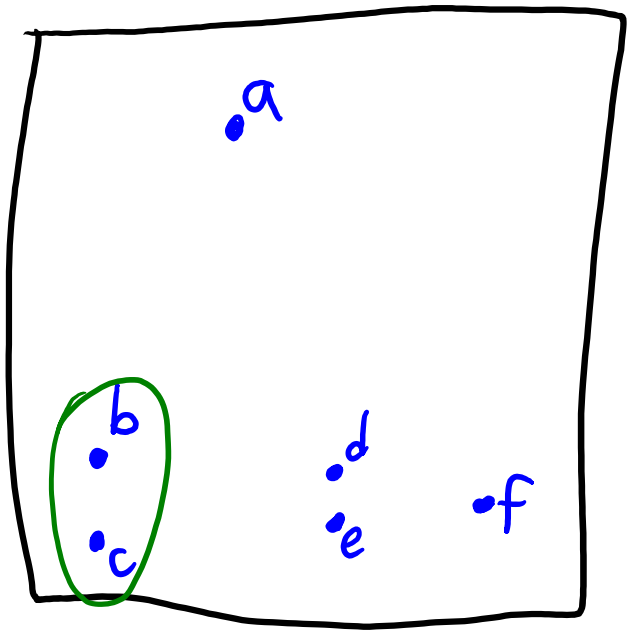


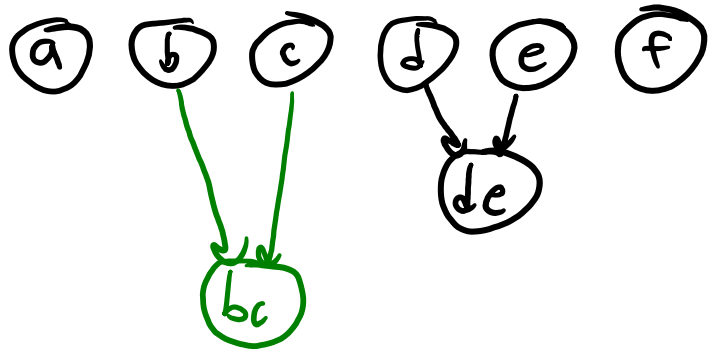


https://en.wikipedia.org/wiki/Hierarchical_clustering

Agglomerative (Bottom-Up) Clustering

- Most common hierarchical method: agglomerative clustering.
  1. Starts with each point in its own cluster.
  2. Each step merges the two “closest” clusters.

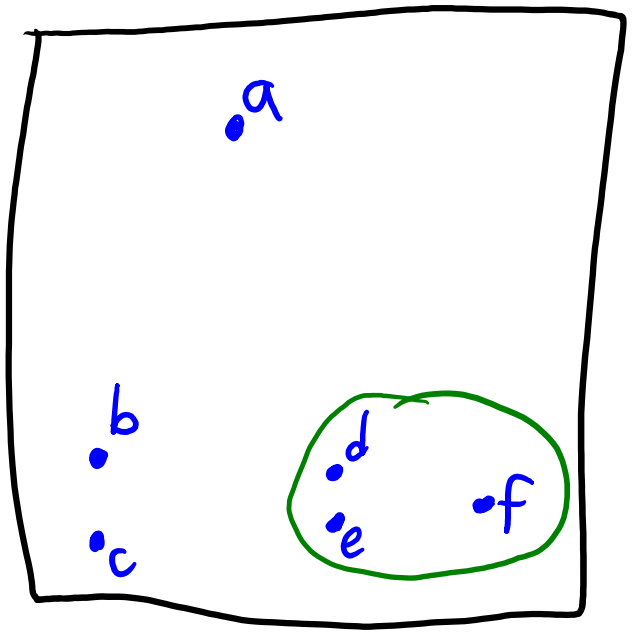


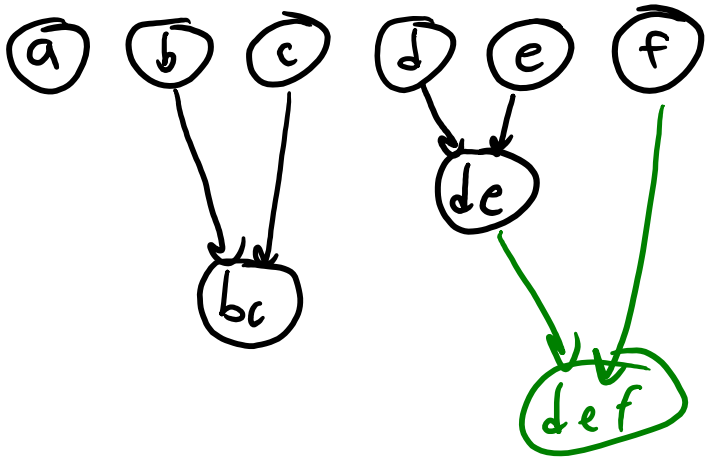


https://en.wikipedia.org/wiki/Hierarchical_clustering

Agglomerative (Bottom-Up) Clustering

- Most common hierarchical method: agglomerative clustering.
  1. Starts with each point in its own cluster.
  2. Each step merges the two “closest” clusters.

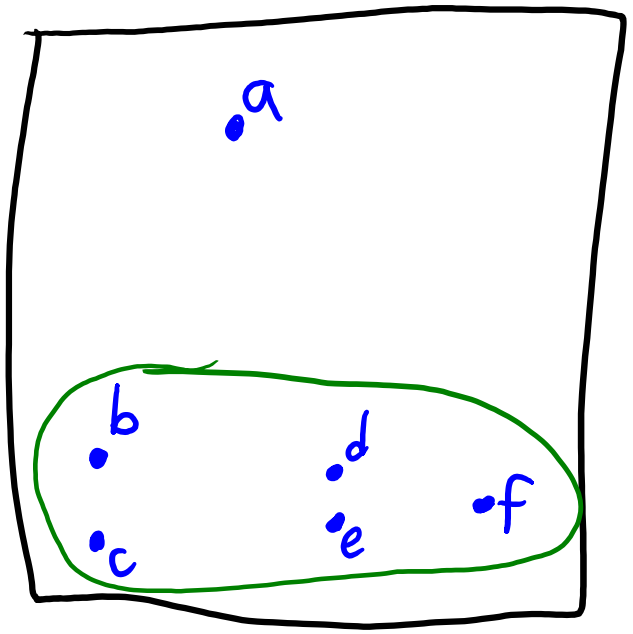


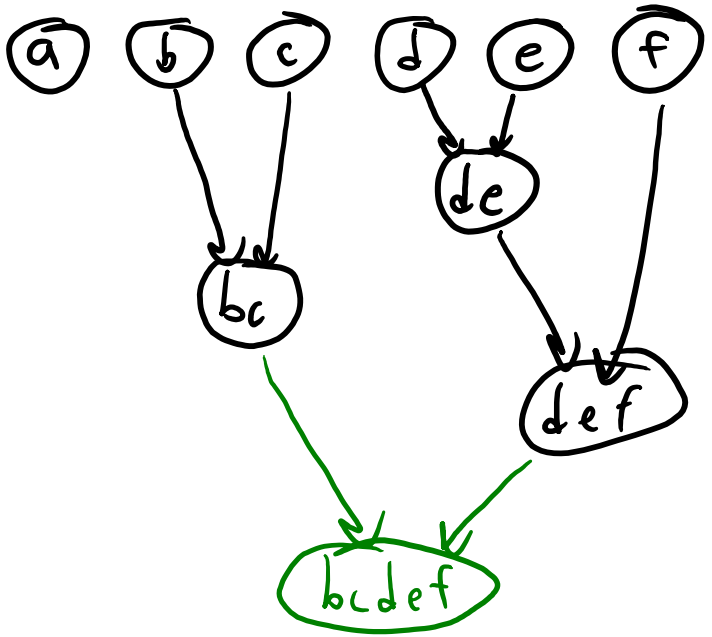


https://en.wikipedia.org/wiki/Hierarchical_clustering

Agglomerative (Bottom-Up) Clustering

- Most common hierarchical method: agglomerative clustering.
  1. Starts with each point in its own cluster.
  2. Each step merges the two “closest” clusters.

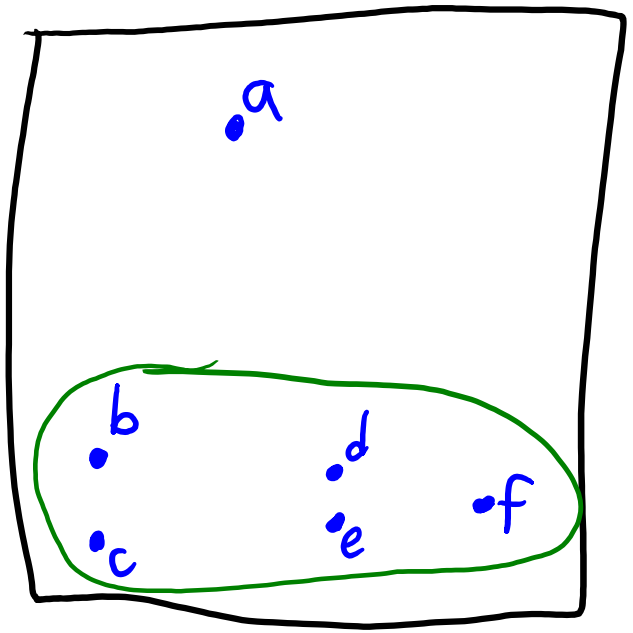


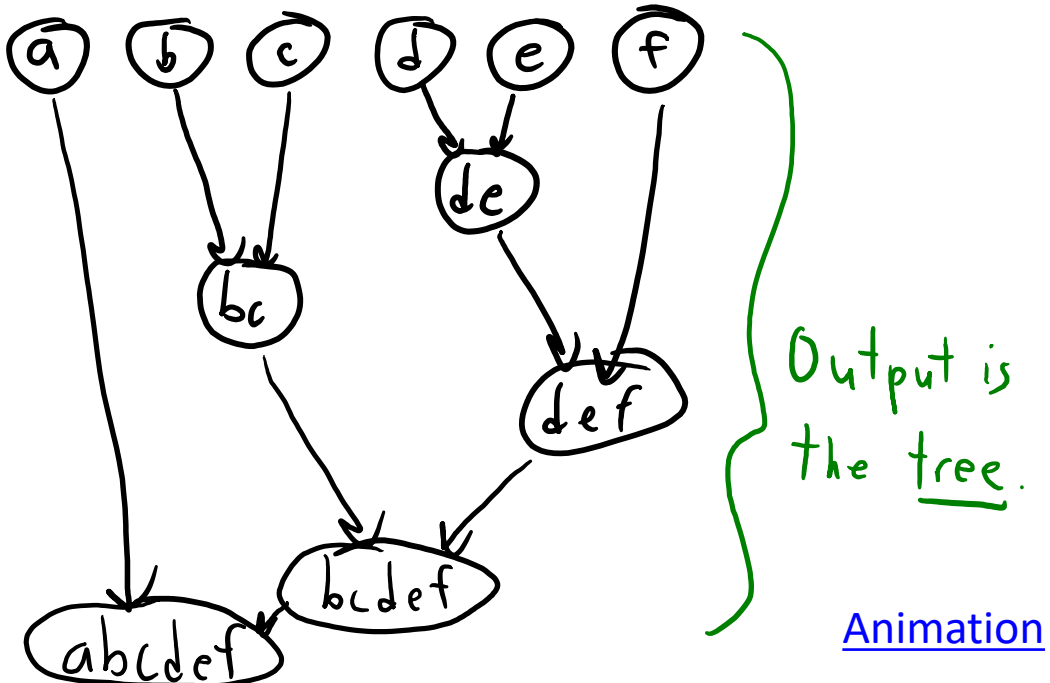


https://en.wikipedia.org/wiki/Hierarchical_clustering

Agglomerative (Bottom-Up) Clustering

- Most common hierarchical method: agglomerative clustering.
  1. Starts with each point in its own cluster.
  2. Each step merges the two “closest” clusters.
  3. Stop with one big cluster that has all points.





Animation

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

Agglomerative (Bottom-Up) Clustering

- Reinvented by different fields under different names ("UPGMA").
- Needs a "distance" between two clusters.
- A standard choice: distance between means of the clusters.
  - Not necessarily the best, many choices exist (bonus slide).
- Cost is $O(n^3 d)$ for basic implementation.
  - Each step costs $O(n^2 d)$, and each step might only cluster 1 new point.

Summary

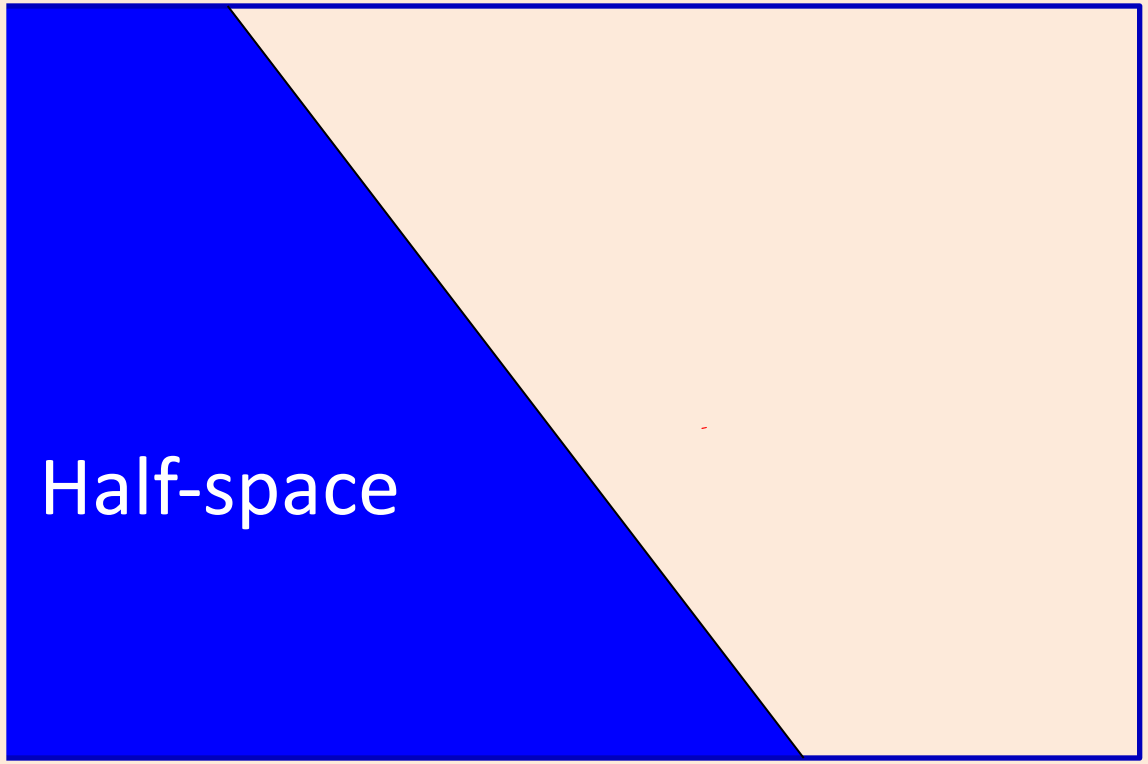
- Shape of K-means clusters:
  - Partitions space into convex sets.
- Density-based clustering:
  - “Expand” and “merge” dense regions of points to find clusters.
  - Not (too) sensitive to initialization or outliers.
  - Useful for finding non-convex connected clusters.
- Ensemble clustering: combines multiple clusterings.
  - Can work well but need to account for label switching.
- Hierarchical clustering: more informative than fixed clustering.
- Agglomerative clustering: standard hierarchical clustering method.
  - Each point starts as a cluster, sequentially merge clusters.

bonus!

Why are k-means clusters convex?

- K-means clusters are formed by the intersection of half-spaces.

Half-space is Set of points (satisfying 'x') a linear inequality, like $\sum_{j=1}^{d} a_j x_j \leq b$

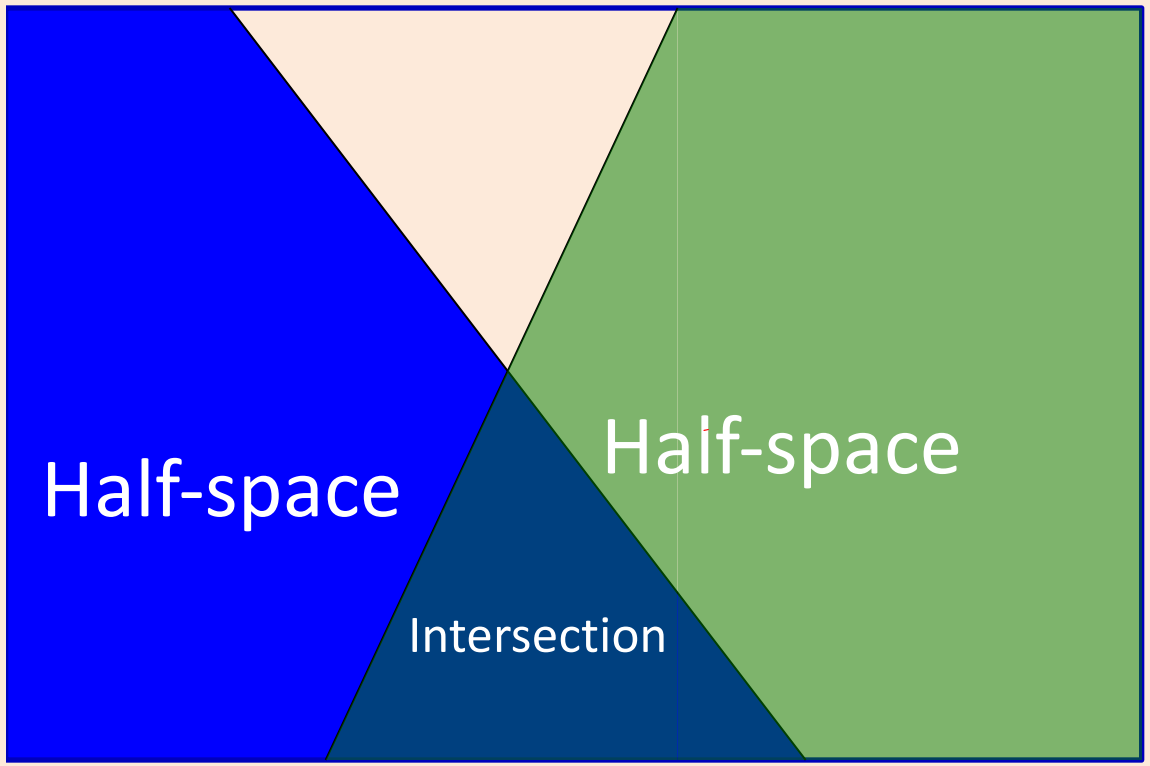


bonus!

Why are k-means clusters convex?

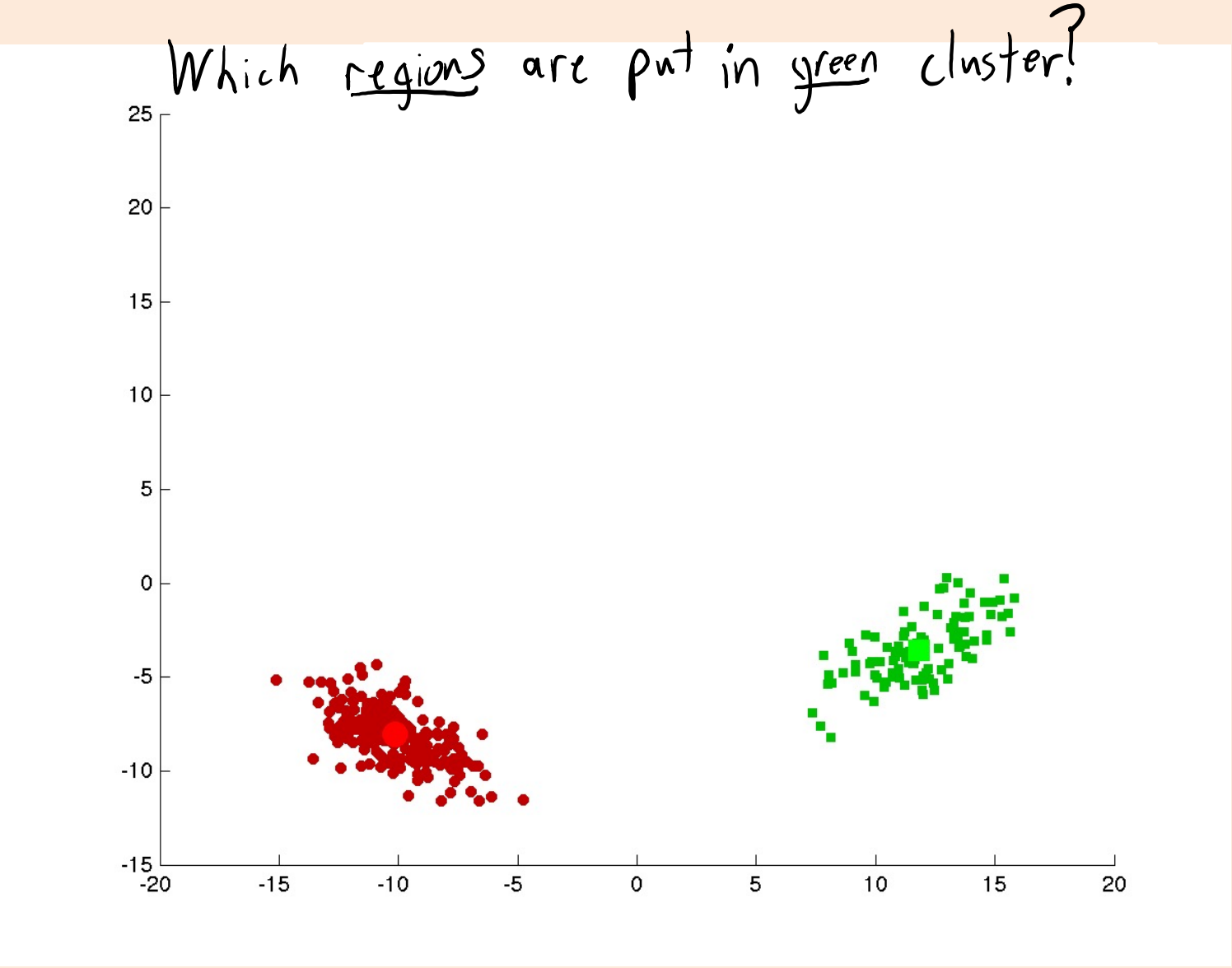
- K-means clusters are formed by the intersection of half-spaces.

Half-space is set of points (satisfying 'x') a linear inequality, like $\sum_{j=1}^{d} a_j x_j \leq b$



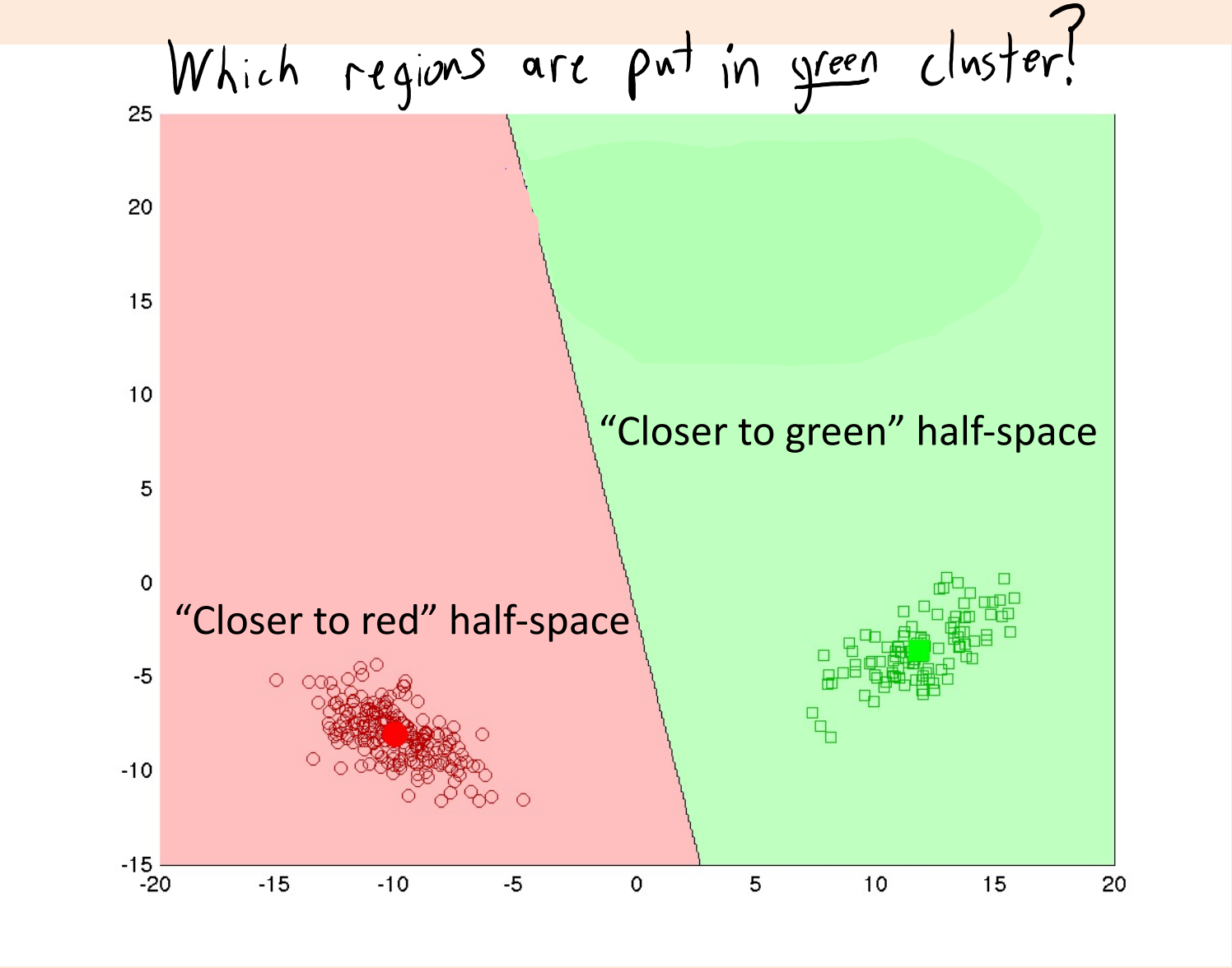
bonus!

Why are k-means clusters convex?



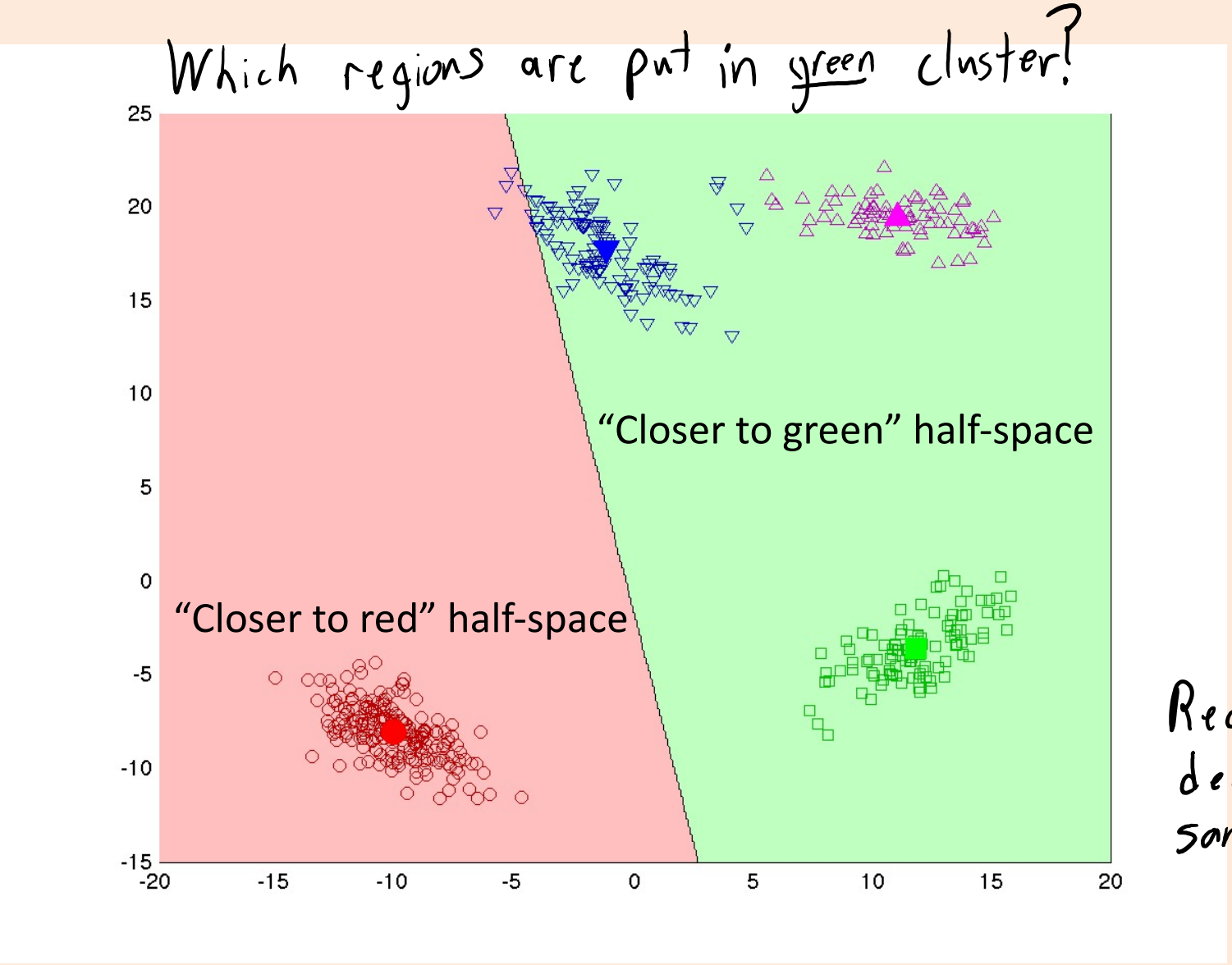
bonus!

Why are k-means clusters convex?



bonus!

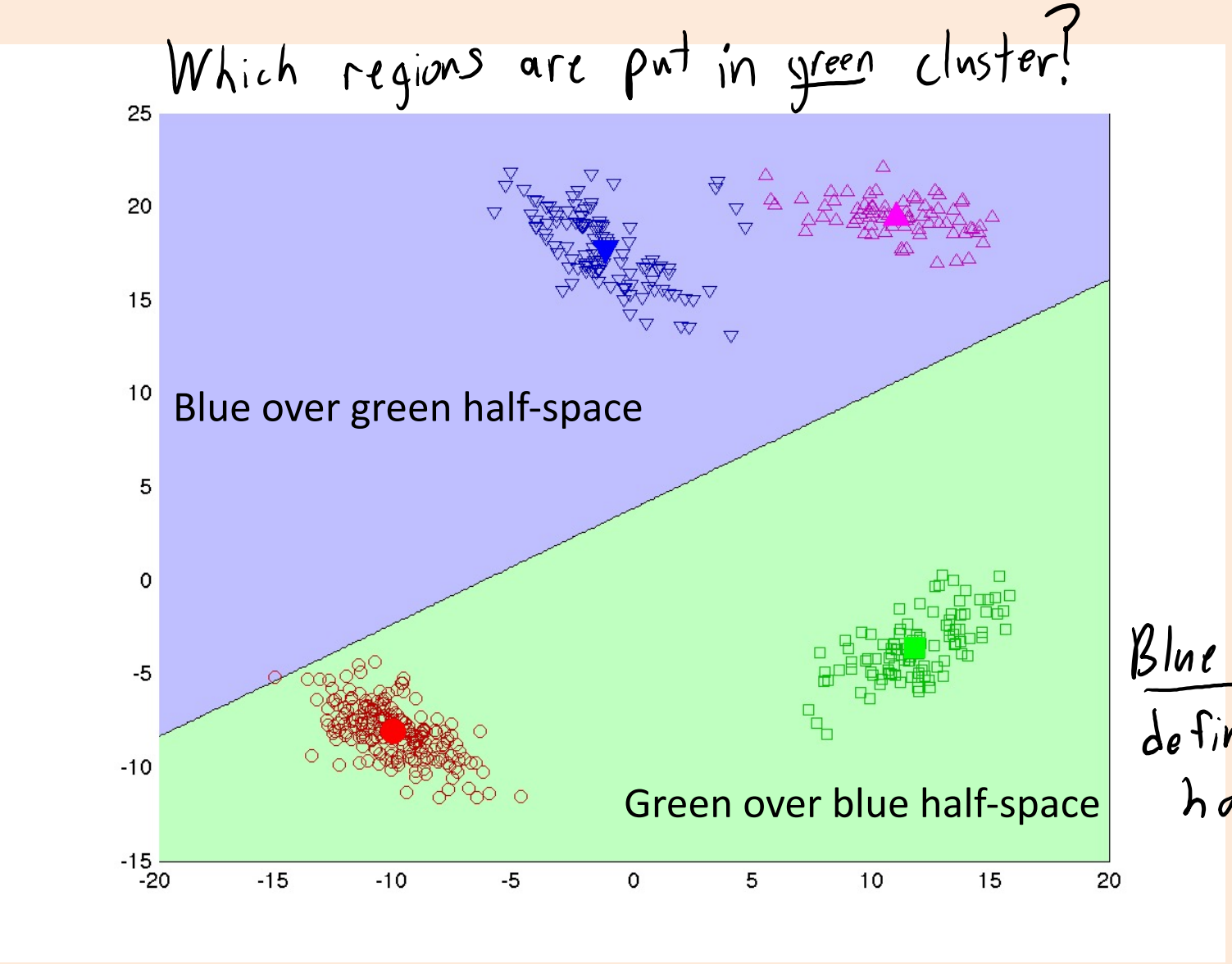
Why are k-means clusters convex?



Red vs. green decision stays the same with more clusters.

bonus!

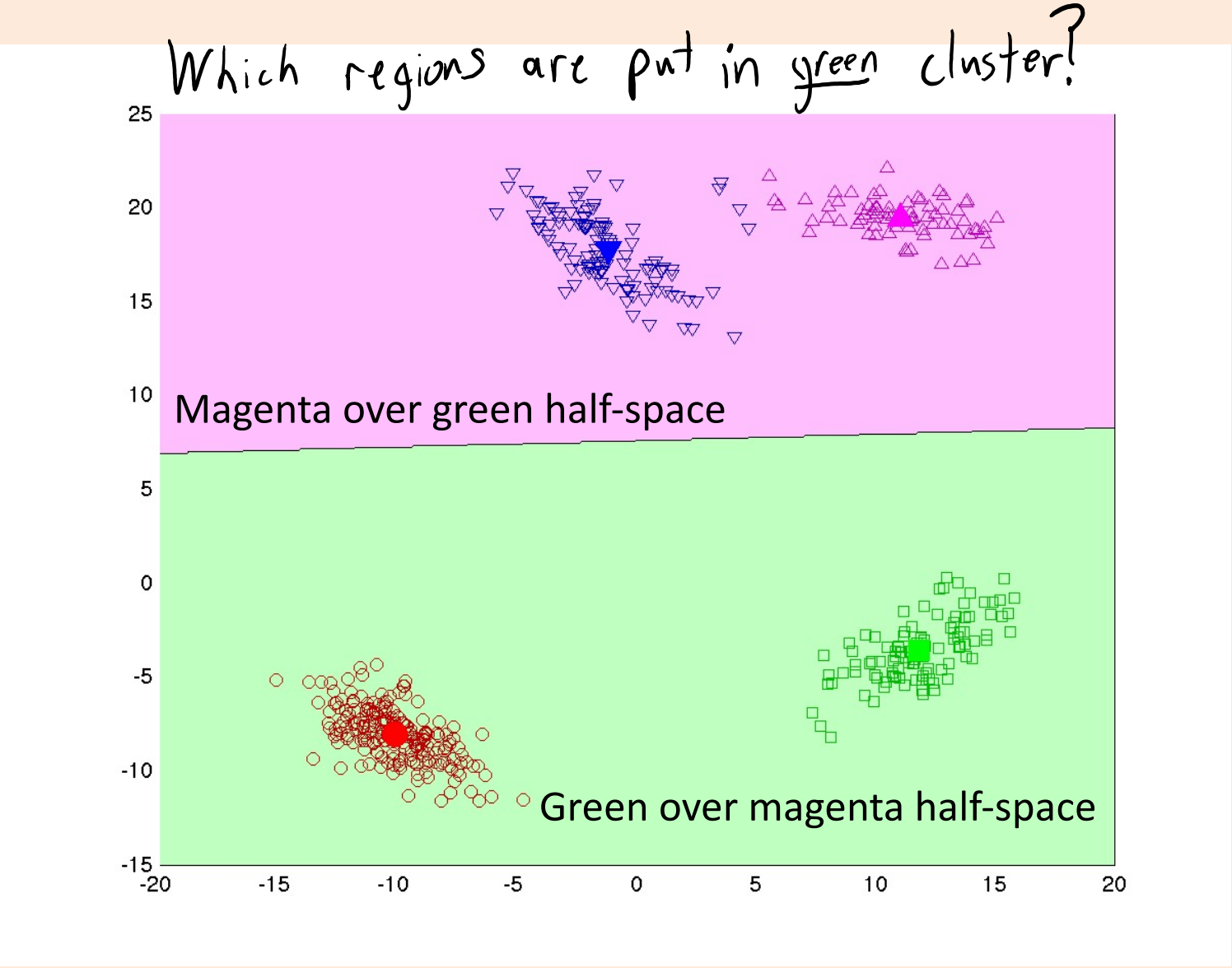
Why are k-means clusters convex?



Blue vs. green decision defines different half-spaces.

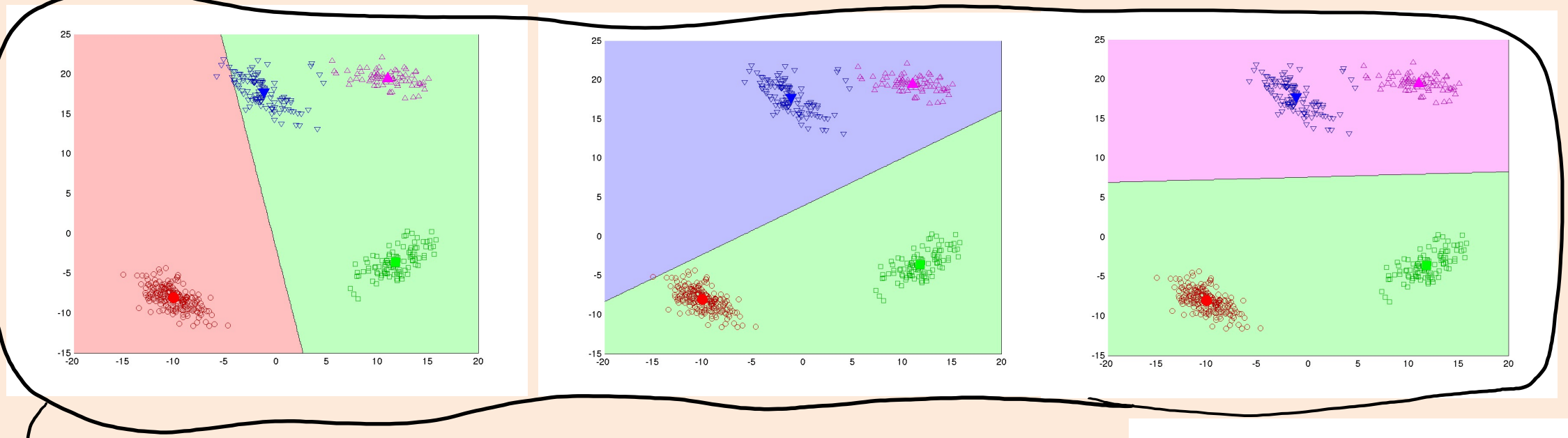
bonus!

Why are k-means clusters convex?



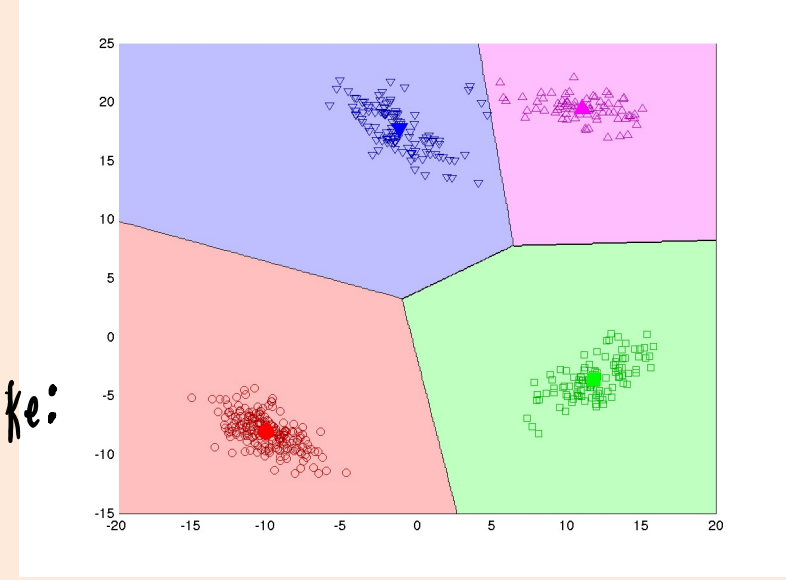
bonus!

Why are k-means clusters convex?



Green "cluster" is the intersection of these three half-spaces.

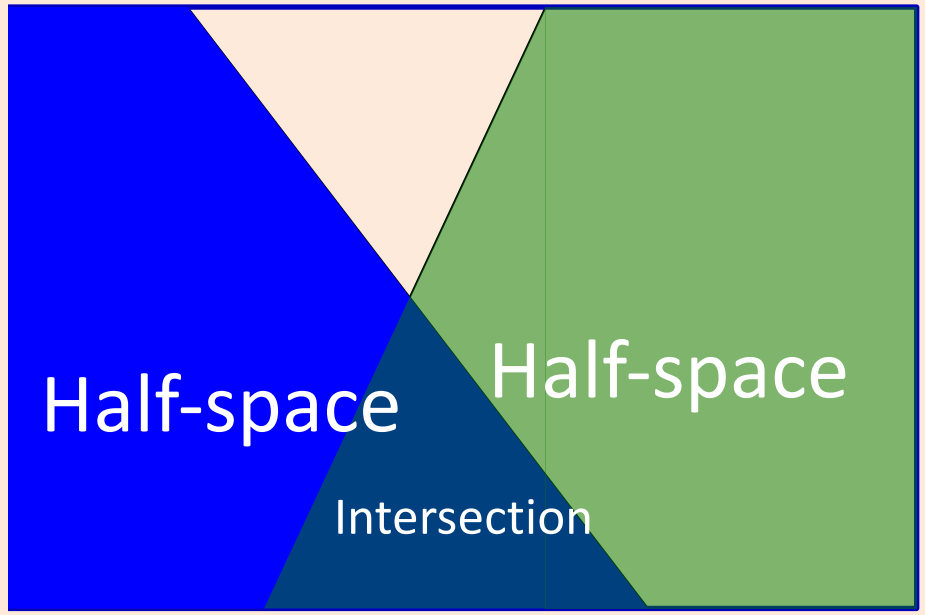
Here is what the four clusters look like:



bonus!

Why are k-means clusters convex?

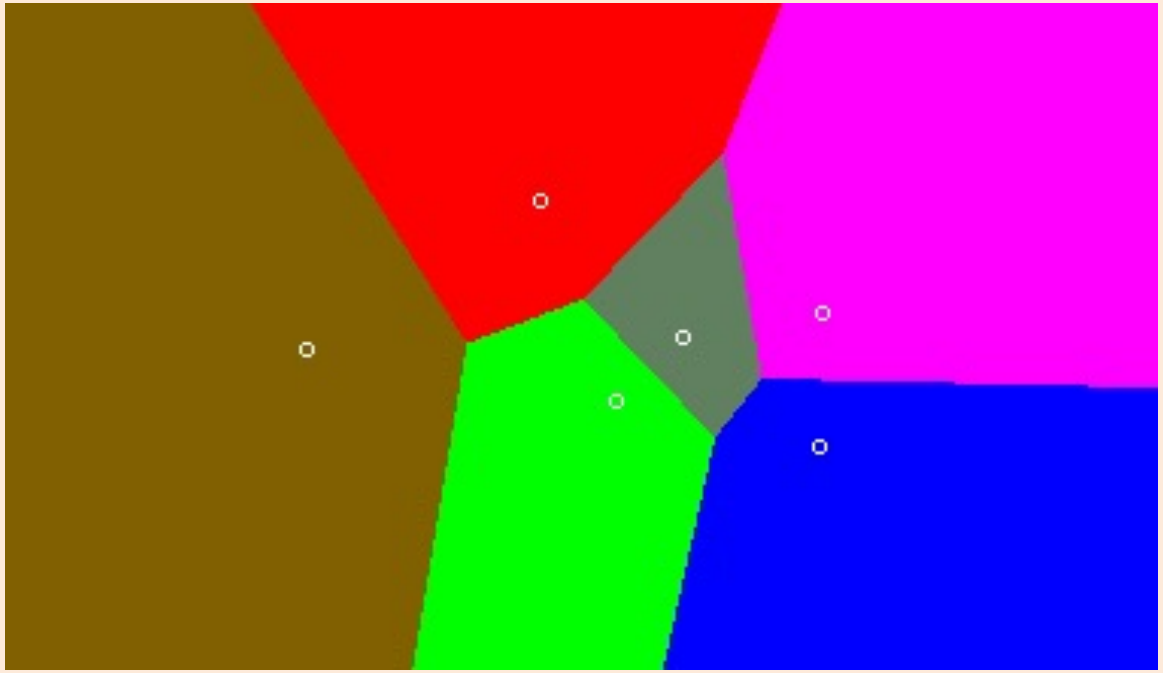
- Half-spaces are convex sets.
- Intersection of convex sets is a convex set.
  - Line segment between points in each set are still in each set.
- So intersection of half-spaces is convex.



bonus!

Voronoi Diagrams

- The k-means partition can be visualized as a Voronoi diagram:



- Can be a useful visualization of “nearest available” problems.
  - E.g., nearest tube station in London.

http://datagenetics.com/blog/may12017/index.html

Density-Based Clustering Runtime

bonus!

question

stop following 72 views

Actions

DBSCAN Training time & Testing time

This is a follow-up inquiry post with Mike about the DBScan, would like to know:

1. Training runtime of DBScan, under k iterations (training set X has n examples and d features)
2. Testing runtime for a single example in DBScan; Testing runtime for test set of size t in DBScan,

the instructors' answer, *where instructors collectively construct a single answer*

For training, you'll check that each point is a core point exactly once. This check costs $O(nd)$ since you measure the distance to each other point, leading to a total training cost of $O(n^2d)$.

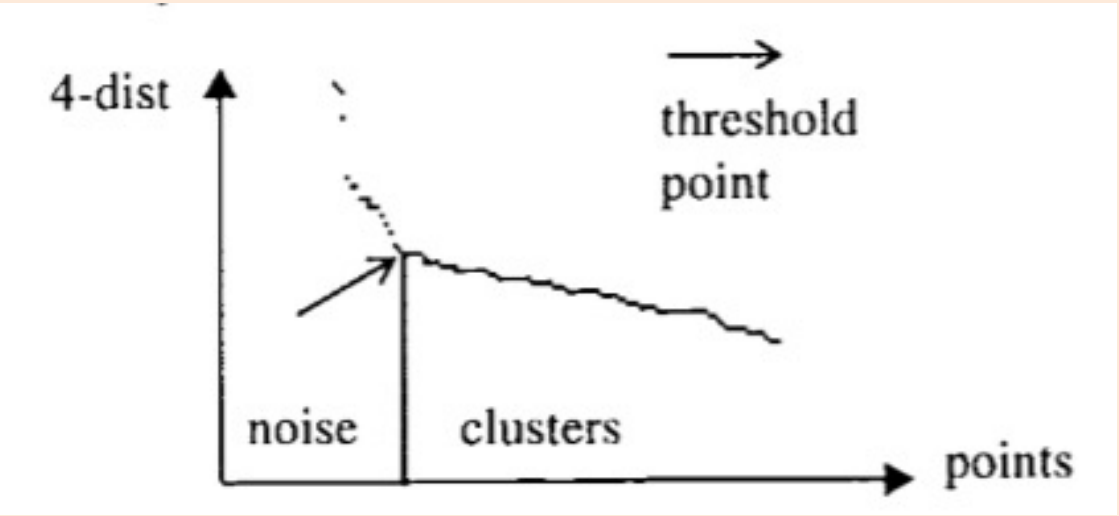
(There are ways to speed this up, like grid-based pruning.)

We didn't define how to apply the DBSCAN model to test data. But a plausible way is to test if the new point is a neighbor of any existing core points. If you have m core points, you would be able to do this in $O(md)$.

“Elbow” Method for Density-Based Clustering

bonus!

- From the original DBSCAN paper:
  - Choose some ‘k’ (they suggest 4) and set minNeighbours=k.
  - Compute distance of each points to its ‘k’ nearest neighbours.
  - Sort the points based on these distances and plot the distances:



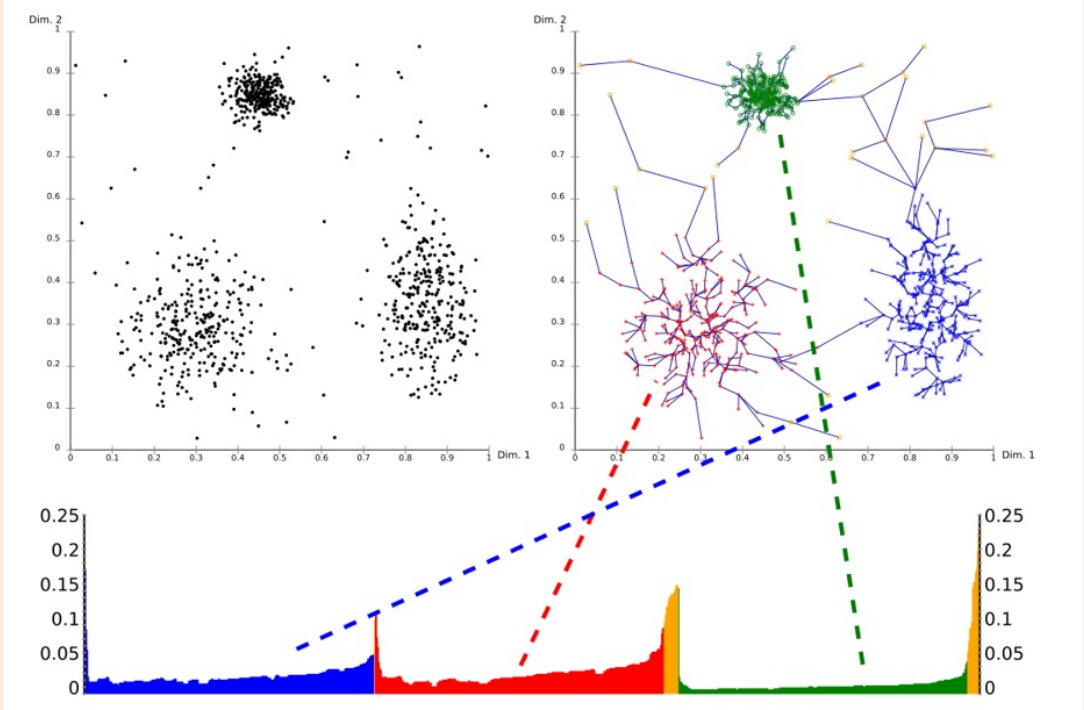
  - Look for an “elbow” to choose ϵ.

https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf

bonus!

OPTICS

- Related to the DBSCAN “elbow” is “OPTICS”.
  - Sort the points so that neighbours are close to each other in the ordering.
  - Plot the distance from each point to the next point.
  - Clusters should correspond to sequencers with low distance.



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

bonus!

UBClustering Algorithm

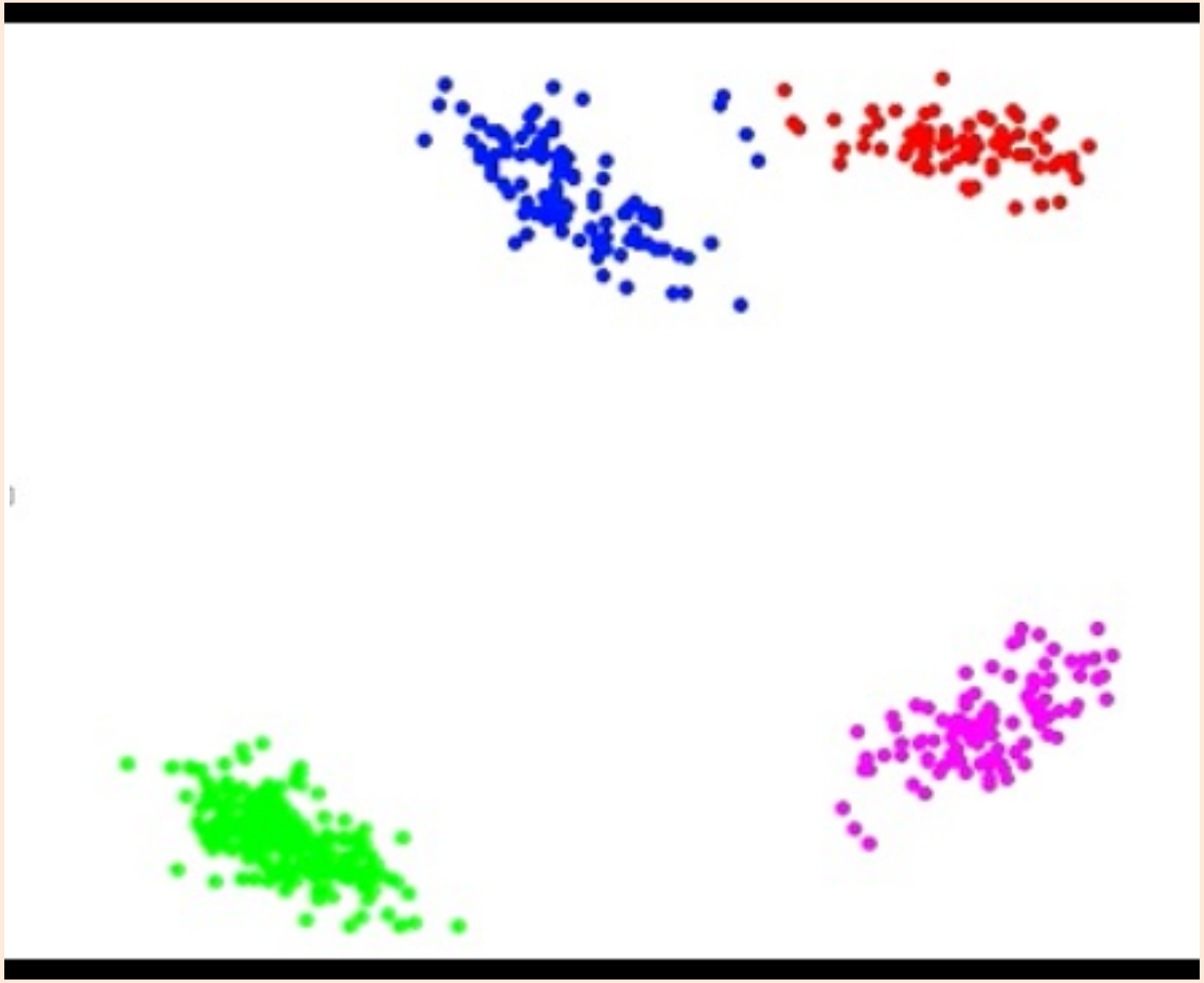
- Let's define a new ensemble clustering method: UBClustering.

1. Run k-means with 'm' different random initializations.
2. For each example i and j:
   - Count the number of times x_i and x_j are in the same cluster.
   - Define p(i,j) = count(x_i in same cluster as x_j)/m.
3. Put x_i and x_j in the same cluster if p(i,j) > 0.5.

- Like DBSCAN merge clusters in step 3 if i or j are already assigned.
  - You can implement this with a DBSCAN code (just changes "distance").
  - Each x_i has an x_j in its cluster with p(i,j) > 0.5.
  - Some points are not assigned to any cluster.

bonus!

UBClustering Algorithm



It looks like DBSCAN, but far-away points will be assigned to a cluster if they always appear in same cluster as other points.

bonus!

Distances between Clusters

- Other choices of the distance between two clusters:
  - "Single-link": minimum distance between points in clusters.
  - "Average-link": average distance between points in clusters.
  - "Complete-link": maximum distance between points in clusters.
  - Ward's method: minimize within-cluster variance.
  - "Centroid-link": distance between a representative point in the cluster.
    - Useful for distance measures on non-Euclidean spaces (like Jaccard similarity).
    - "Centroid" often defined as point in cluster minimizing average distance to other points.

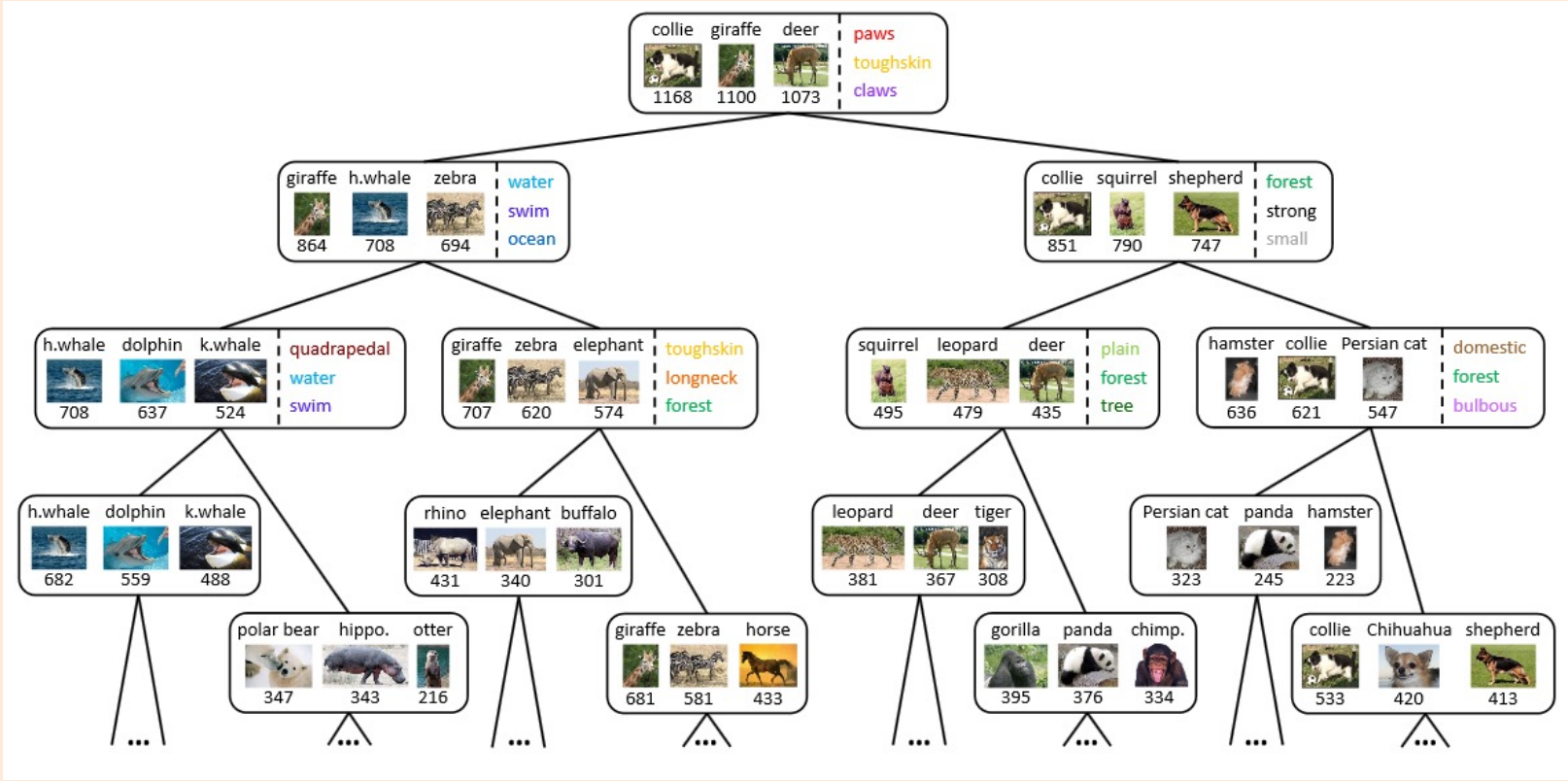
bonus!

Cost of Agglomerative Clustering

- One step of agglomerative clustering costs $O(n^2d)$:
  - We need to do the O(d) distance calculation between up to $O(n^2)$ points.
  - This is assuming the standard distance functions.
- We do at most O(n) steps:
  - Starting with 'n' clusters and merging 2 clusters on each step, after O(n) steps we'll only have 1 cluster left (though typically it will be much smaller).
- This gives a total cost of $O(n^3d)$.

- This can be reduced to $O(n^2d \log n)$ with a priority queue:
  - Store distances in a sorted order, only update the distances that change.
- For single- and complete-linkage, you can get it down to $O(n^2d)$.
  - "SLINK" and "CLINK" algorithms.

Bonus Slide: Divisive (Top-Down) Clustering

bonus!

- Start with all examples in one cluster, then start dividing.
- E.g., run k-means on a cluster, then run again on resulting clusters.
  - A clustering analogue of decision tree learning.