CPSC 340: Machine Learning and Data Mining

More Clustering

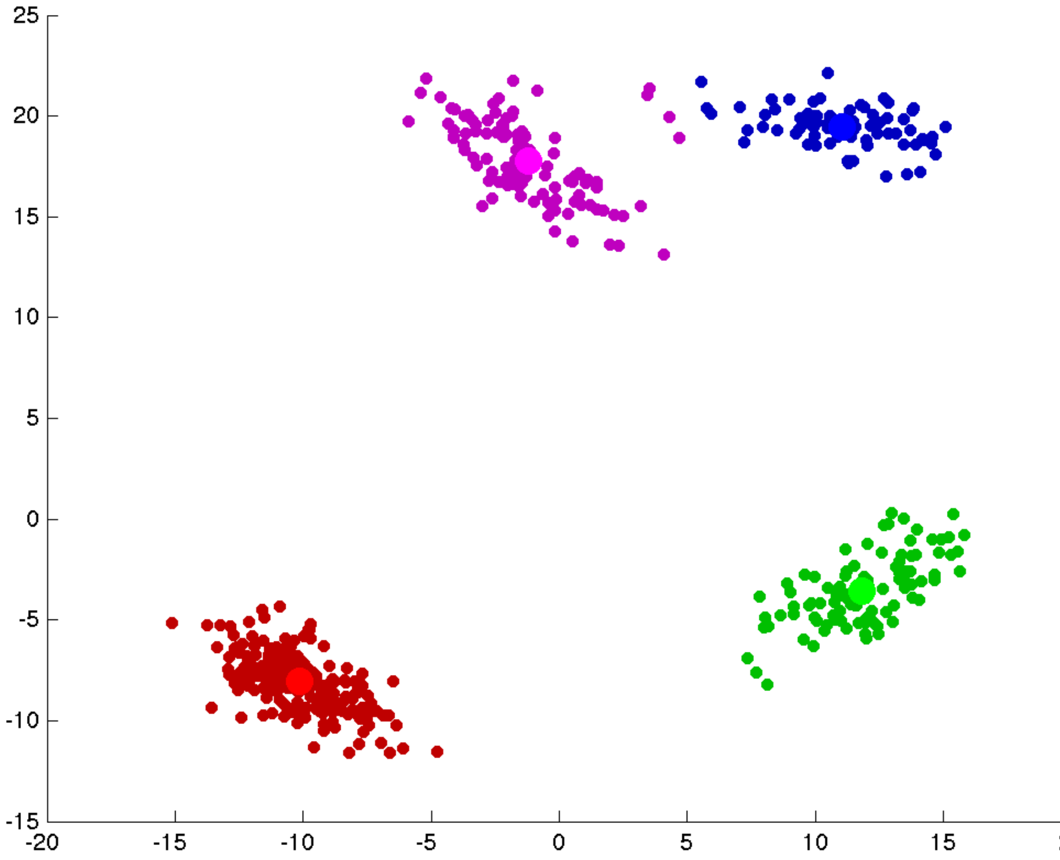
Fall 2018

Admin

- Assignment 2 is due Friday.
- Assignment 1 grades available?
- Midterm rooms are now booked.
  - October 18th at 6:30pm (BUCH A102 and A104).
- Mike and I will get a little out of sync over the next few lectures.
  - Keep this in mind if you alternating between our lectures.

Last Time: K-Means Clustering

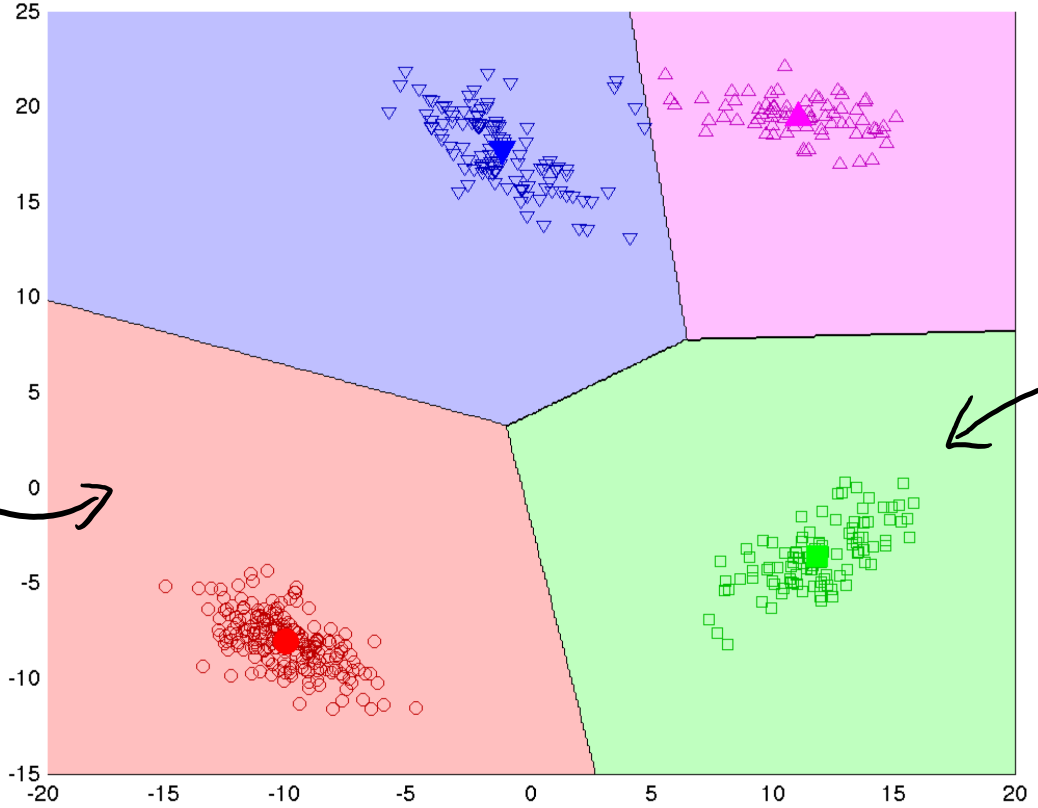
- We want to cluster data:
  – Assign examples to groups.
- K-means clustering:
  – Define groups by “means”
  – Assigns examples to nearest mean.
  (And updates means during training.)
- Also used for vector quantization:
  – Use means as “prototypes” of groups.
- Issues with k-means:
  – Fast but sensitive 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.



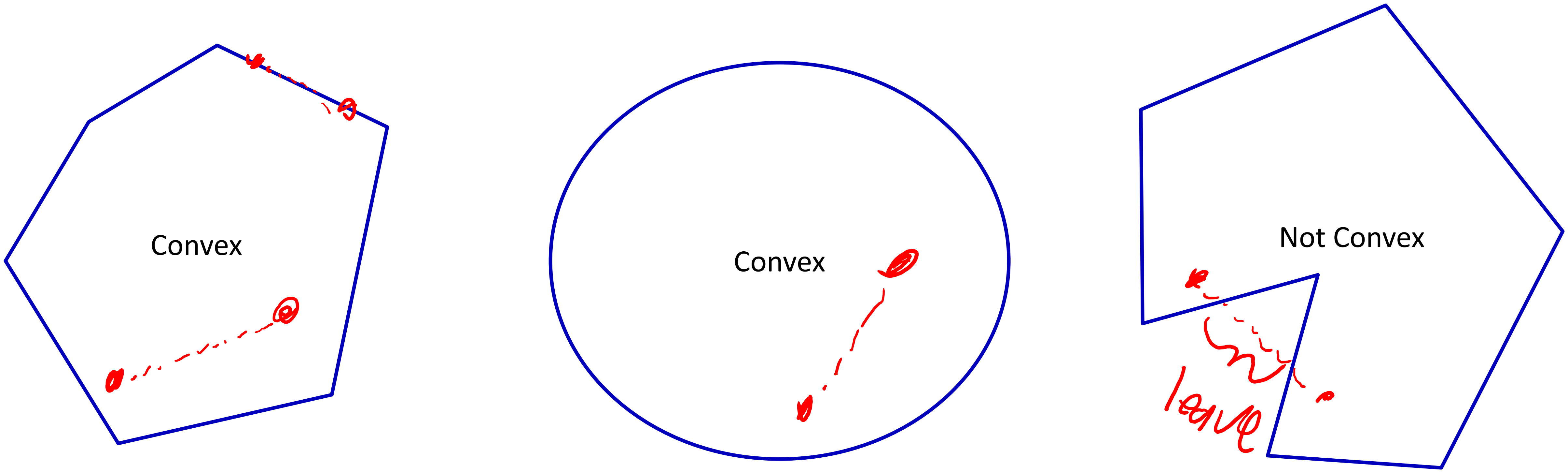
Points here are closest to green mean.

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

Animation

Convex Sets

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

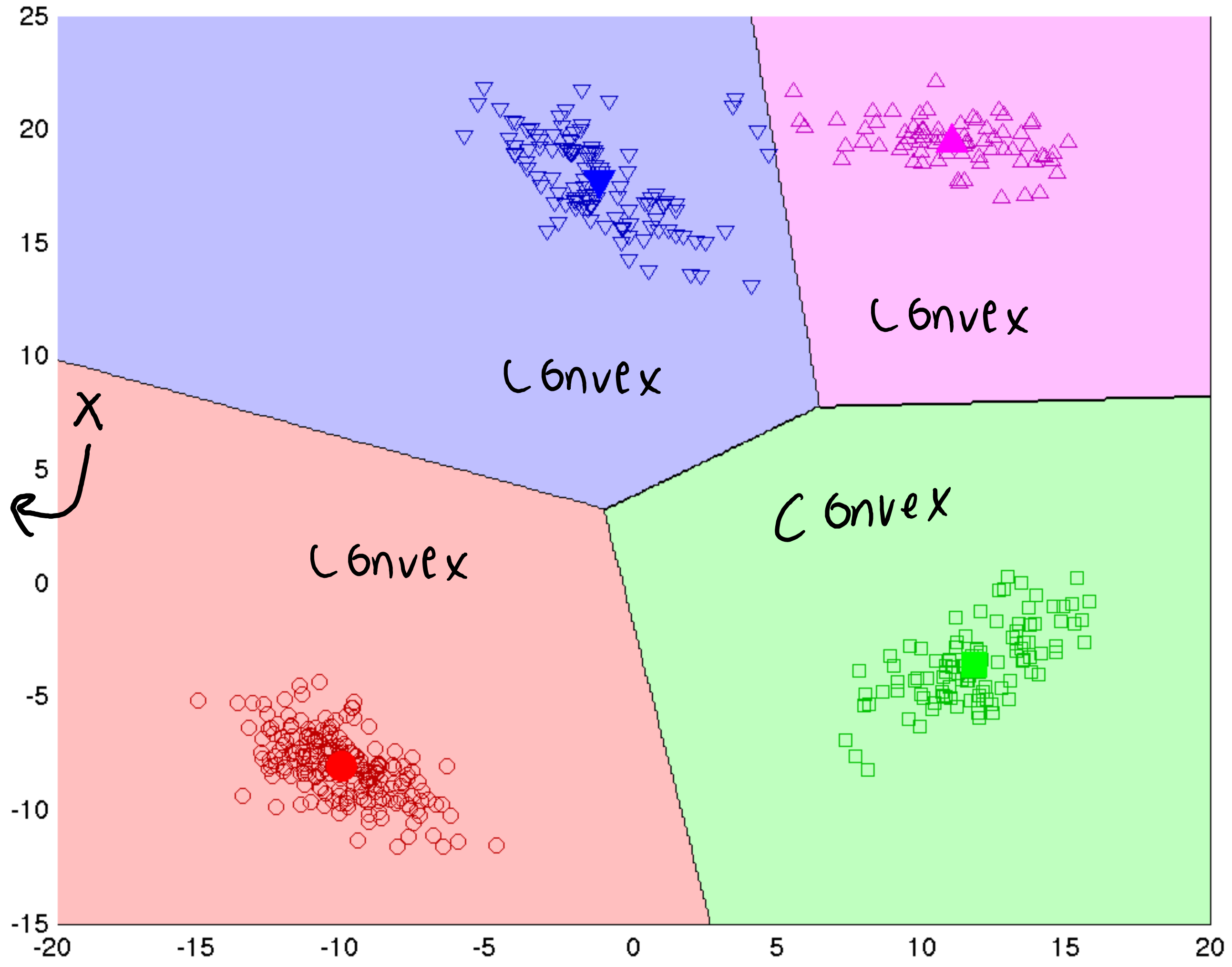


Shape of K-Means Clusters

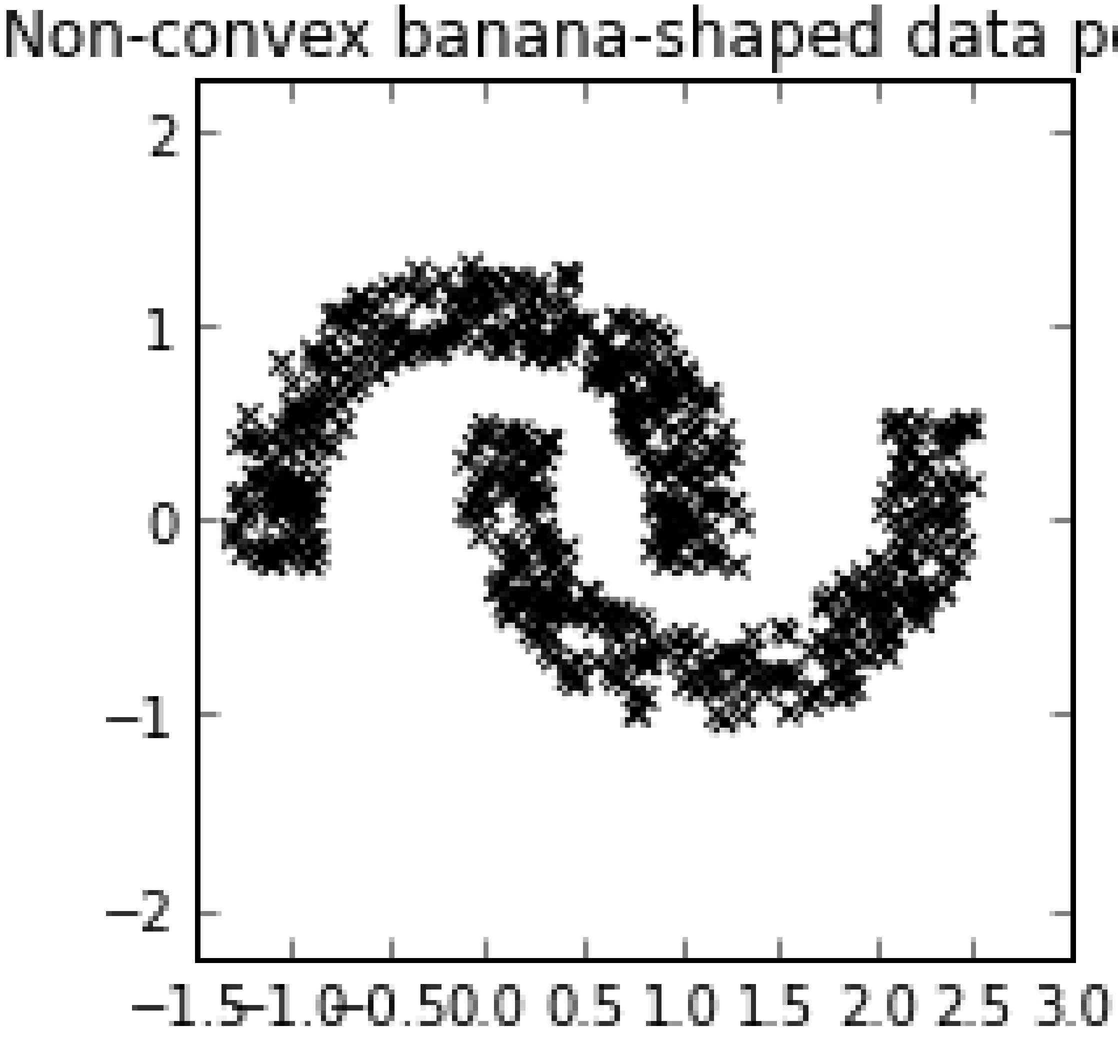
Issues with shape of K-means clusters:

1. Clusters in the data might not be convex.

2. Does this point really belong in red cluster

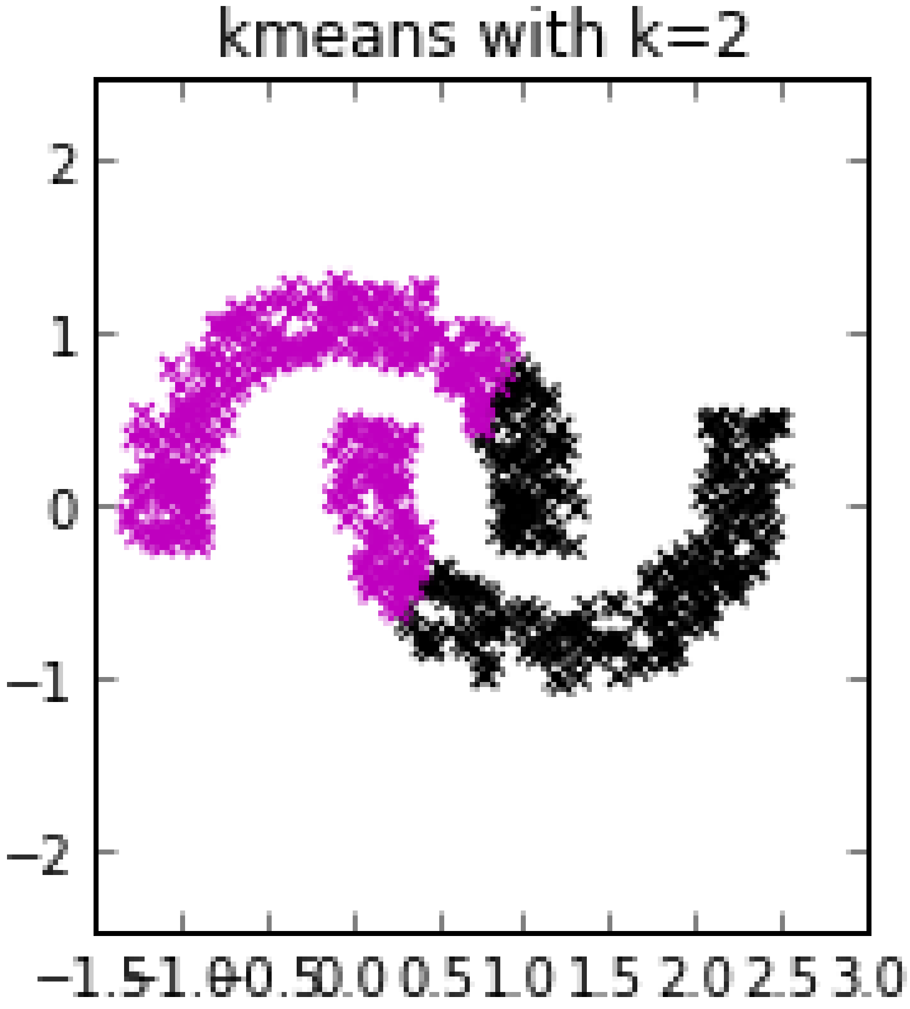


K-Means with Non-Convex Clusters



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

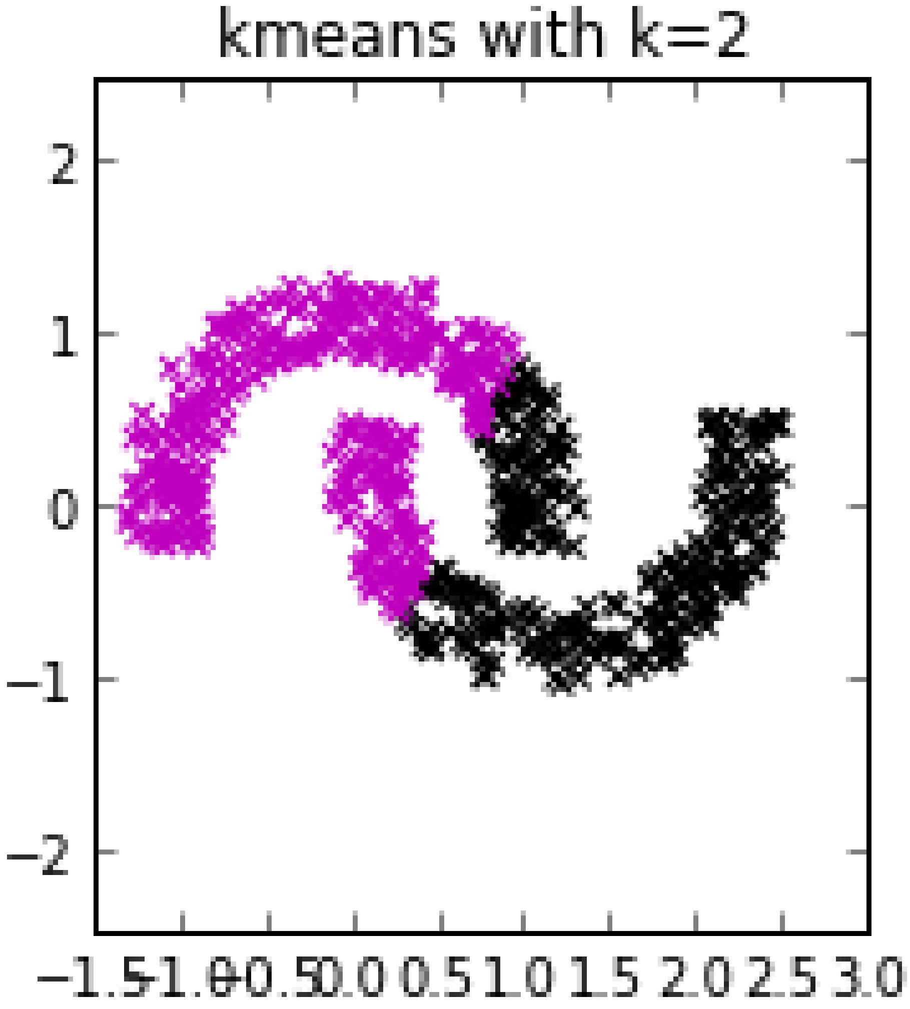
K-Means with Non-Convex Clusters



K-means cannot separate non-convex clusters

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

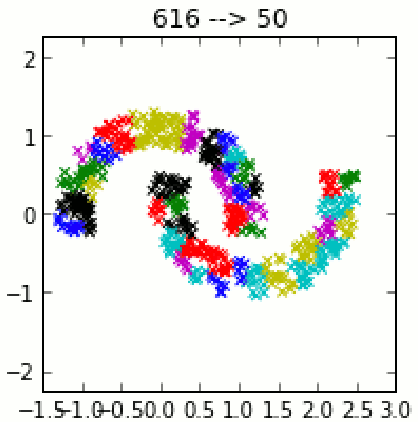
K-Means with Non-Convex Clusters



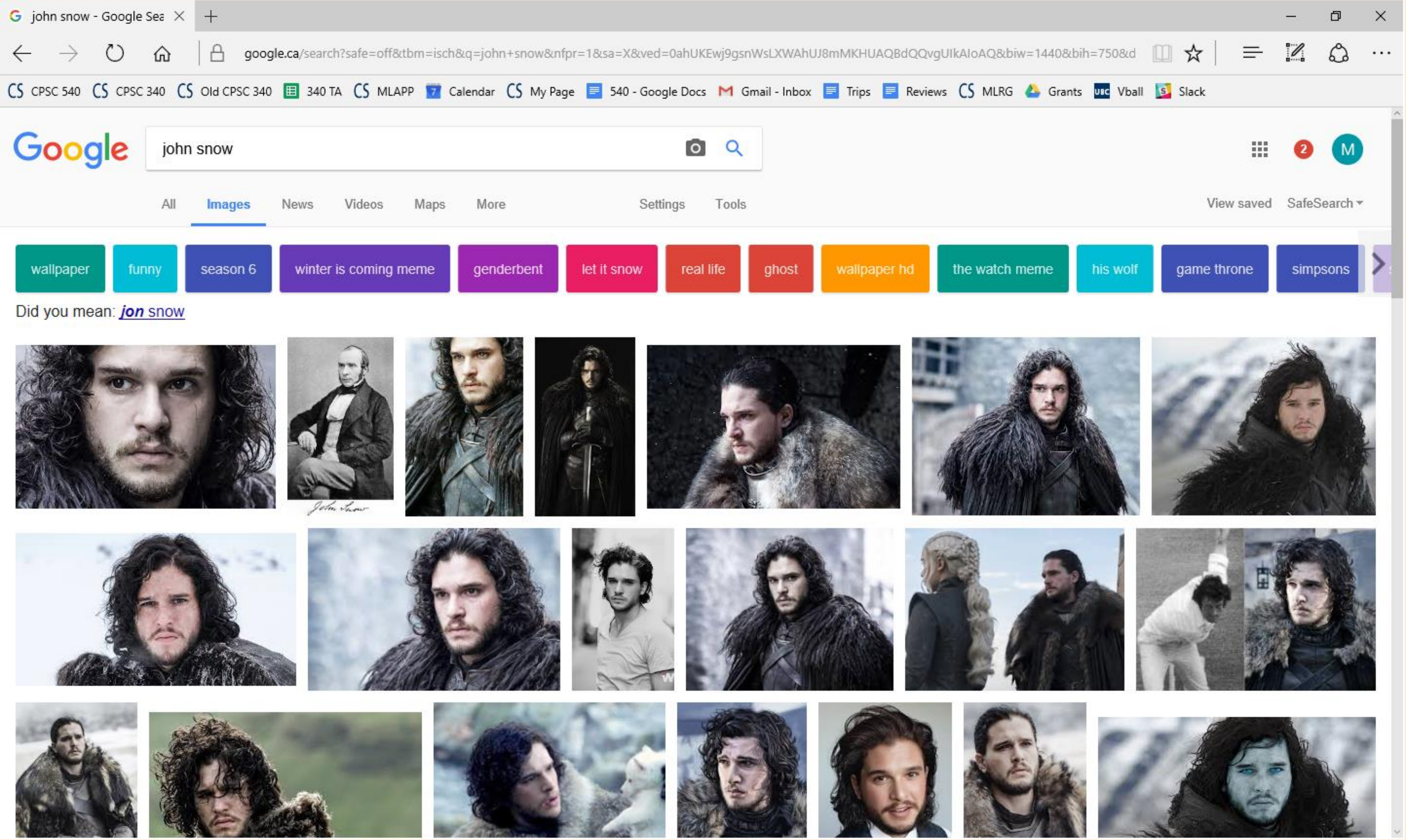
K-means cannot separate non-convex clusters

Though over-clustering can help

(next class)

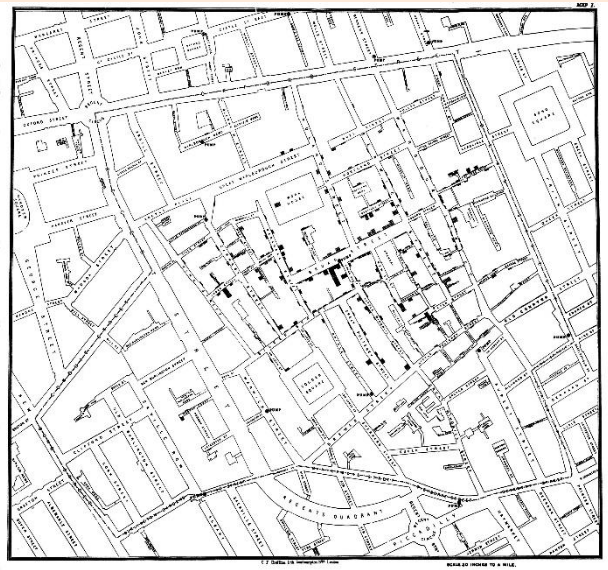


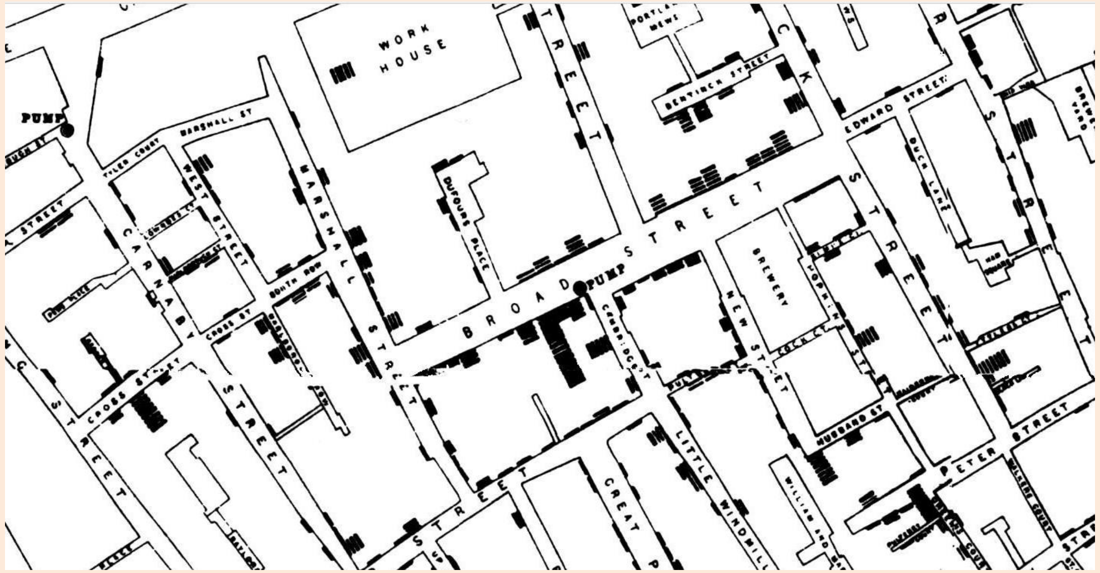
https://corelifesciences.com/human-long-non-coding-rna-expression-microarray-service.html


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

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

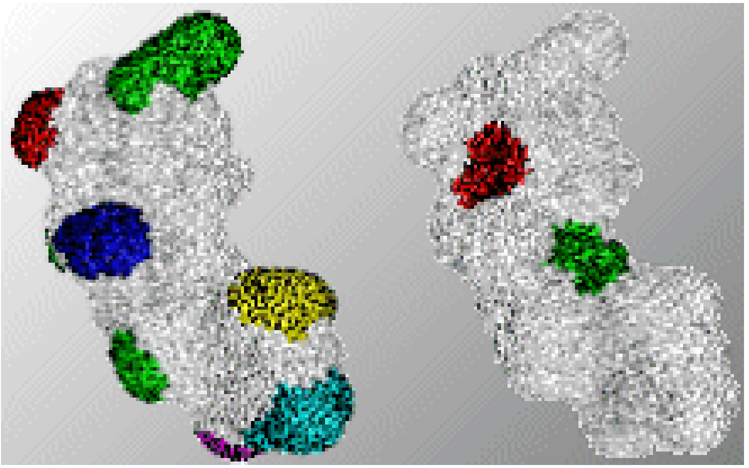
- Density-based clustering:
  - Clusters are defined by “dense” regions.
  - Examples in non-dense regions don’t get clustered.
    - Not trying to “partition” the space.

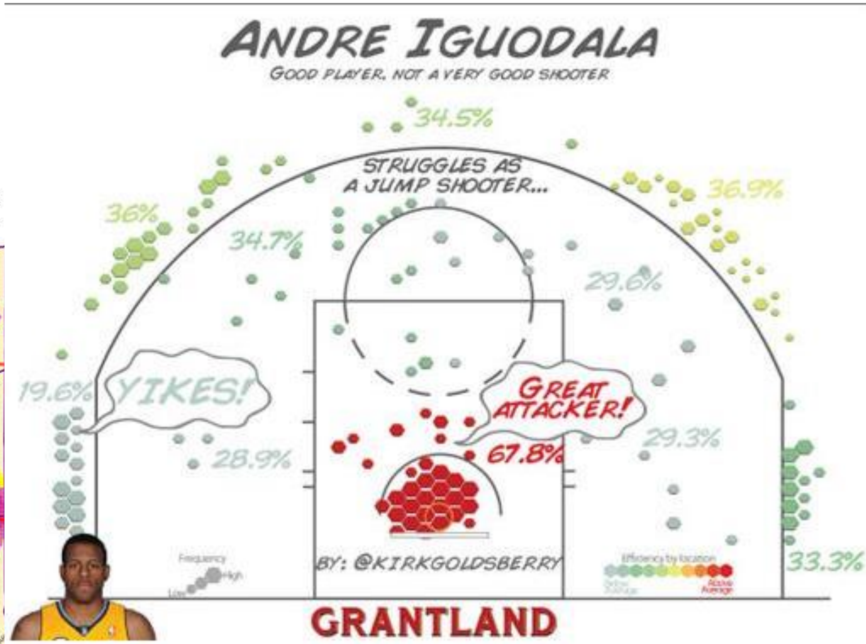
- Clusters can be non-convex:
  - Elephant clusters affected by vegetation, mountains, rivers, water access, etc.
- It’s a non-parametric clustering method:
  - No fixed number of clusters ‘k’.
  - Clusters can become more complicated with more data.

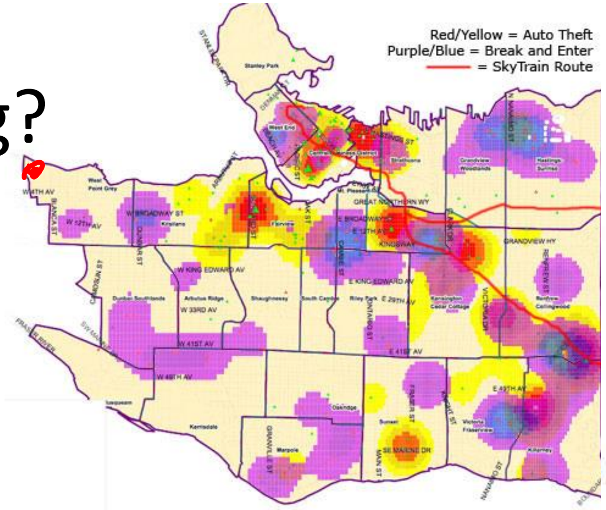
http://www.defenders.org/elephant/basic-facts

Other Potential Applications

- Where are high crime regions of a city?
- 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 protein ‘dock’?
- Where are people tweeting?



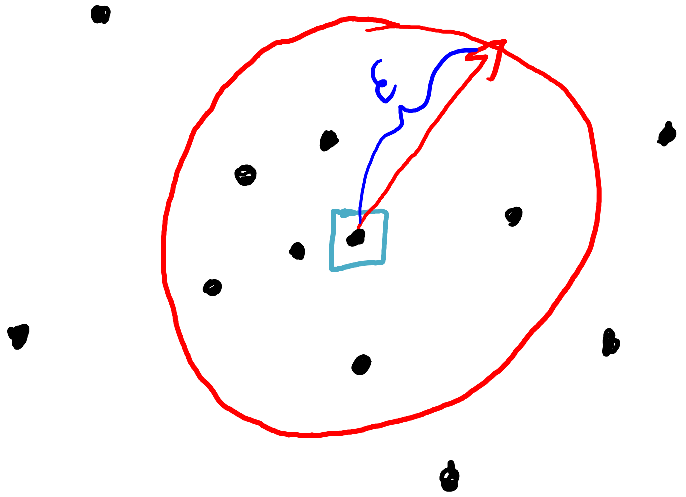




https://en.wikipedia.org/wiki/Cluster_analysis
https://www.flickr.com/photos/dbarefoot/420194128/
http://letsgowarriors.com/replacing-jarrett-jack/2013/10/04/
http://www.dbs.informatik.uni-muenchen.de/Forschung/KDD/Clustering/

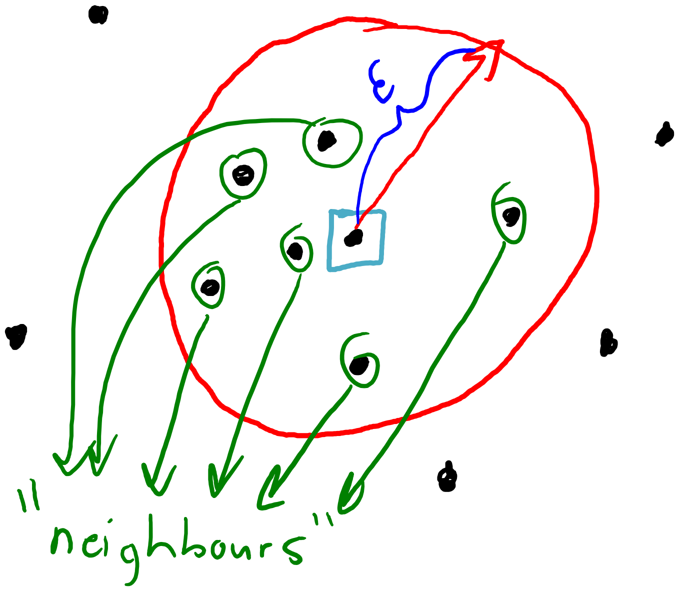
Density-Based Clustering

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



Density-Based Clustering

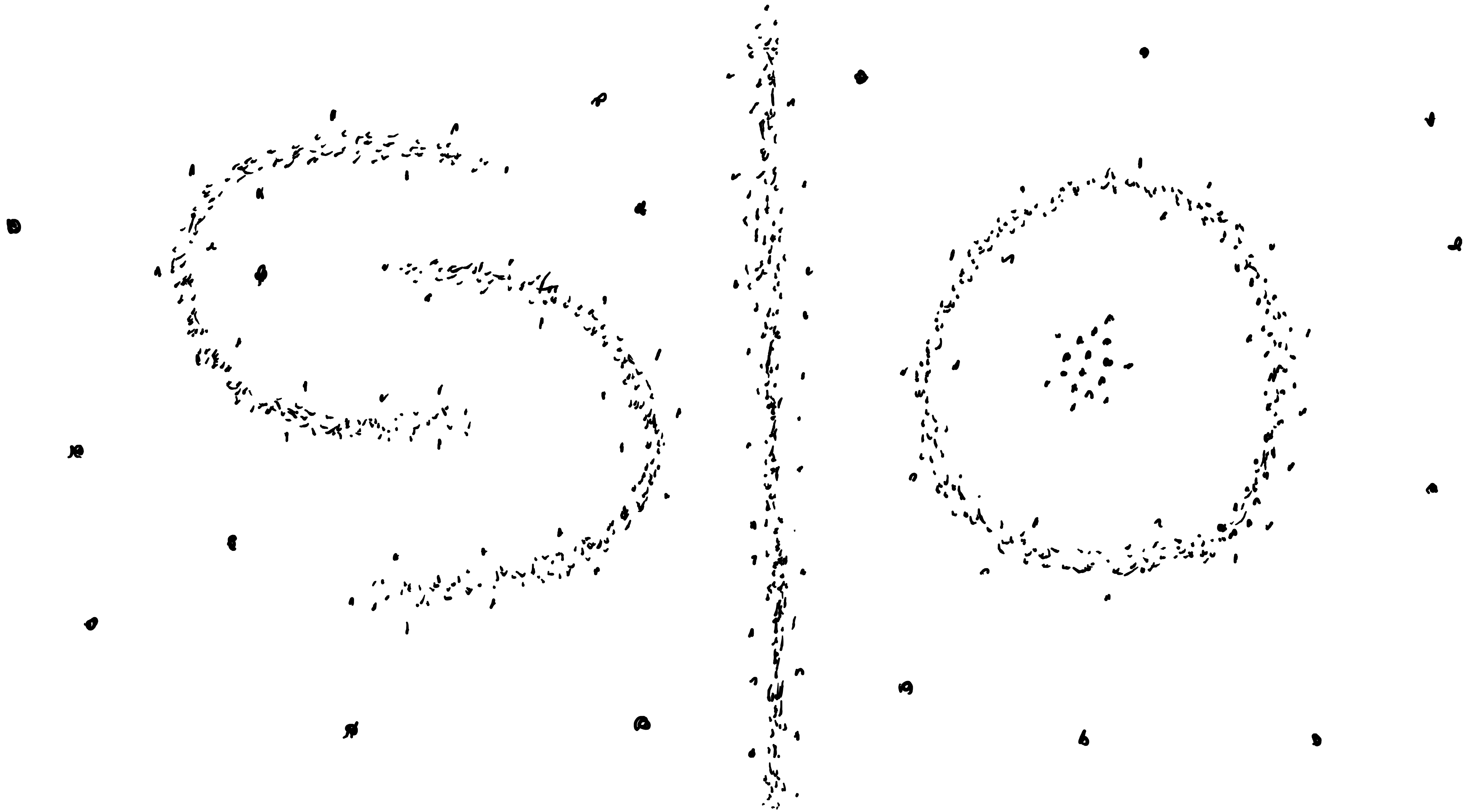
- Density-based clustering algorithm (DBSCAN) has two hyperparameters:
  - Epsilon (ε): distance we use to decide if another point is a "neighbour".



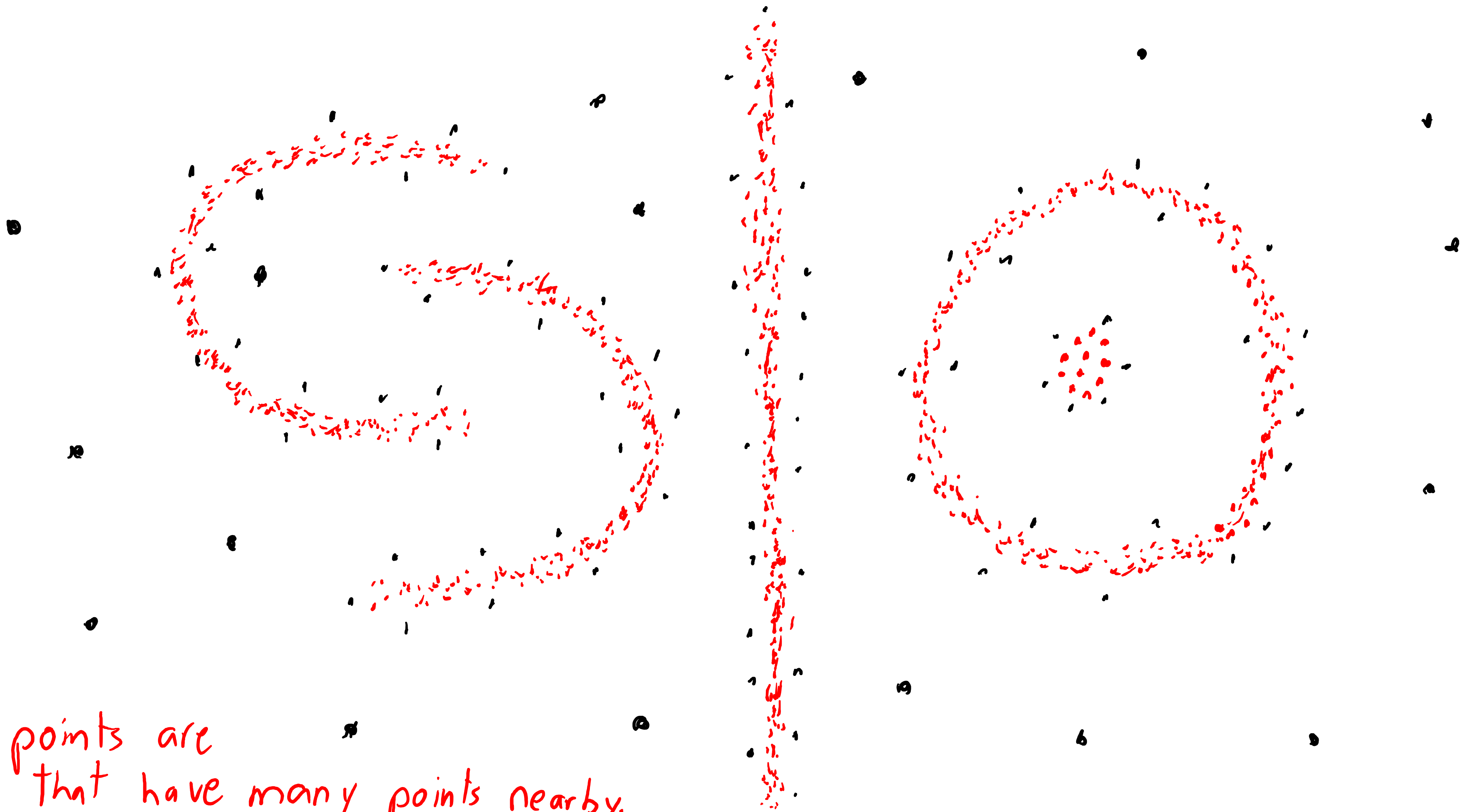
Density-Based Clustering

- 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.

Density-Based Clustering

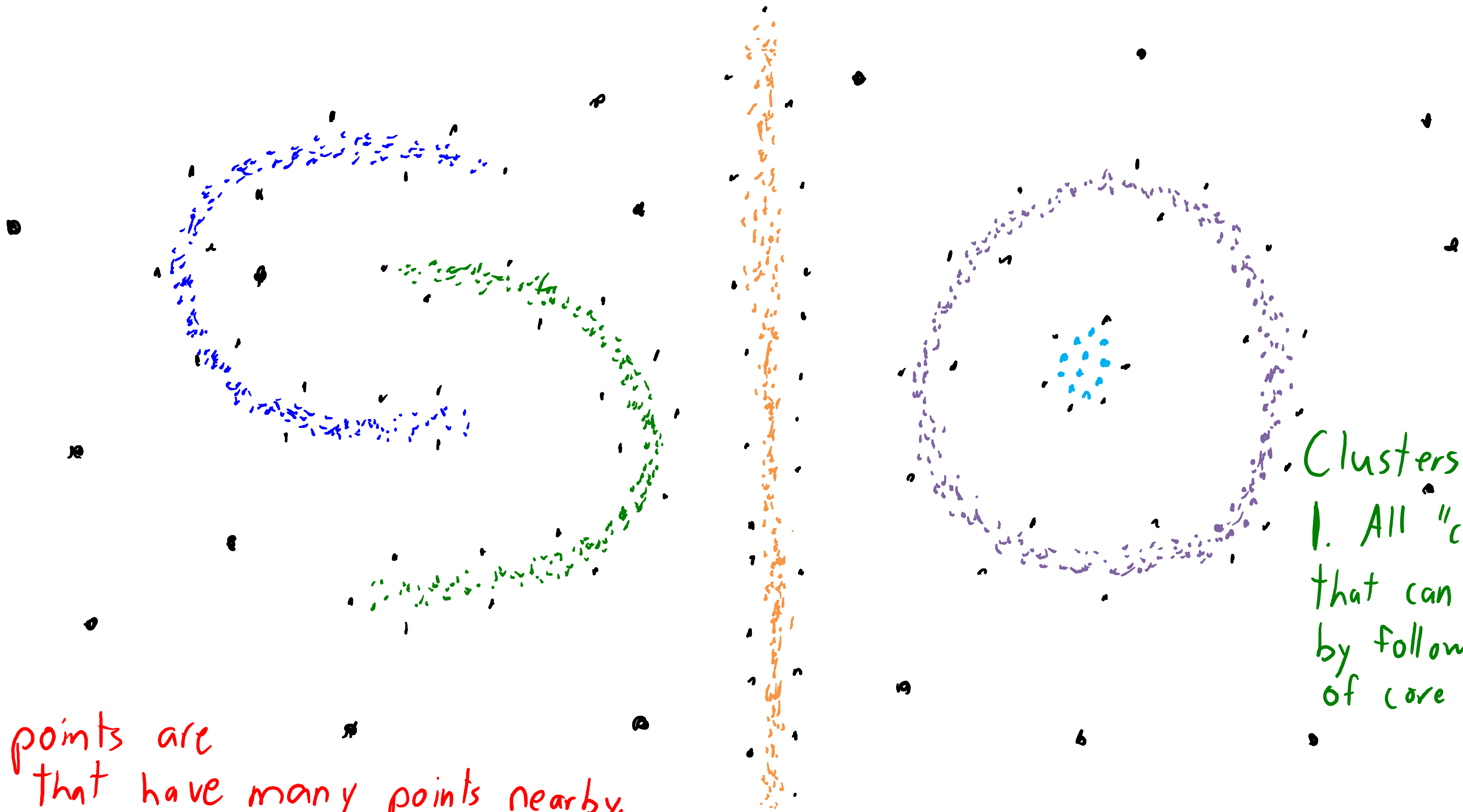


Density-Based Clustering



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

Density-Based Clustering

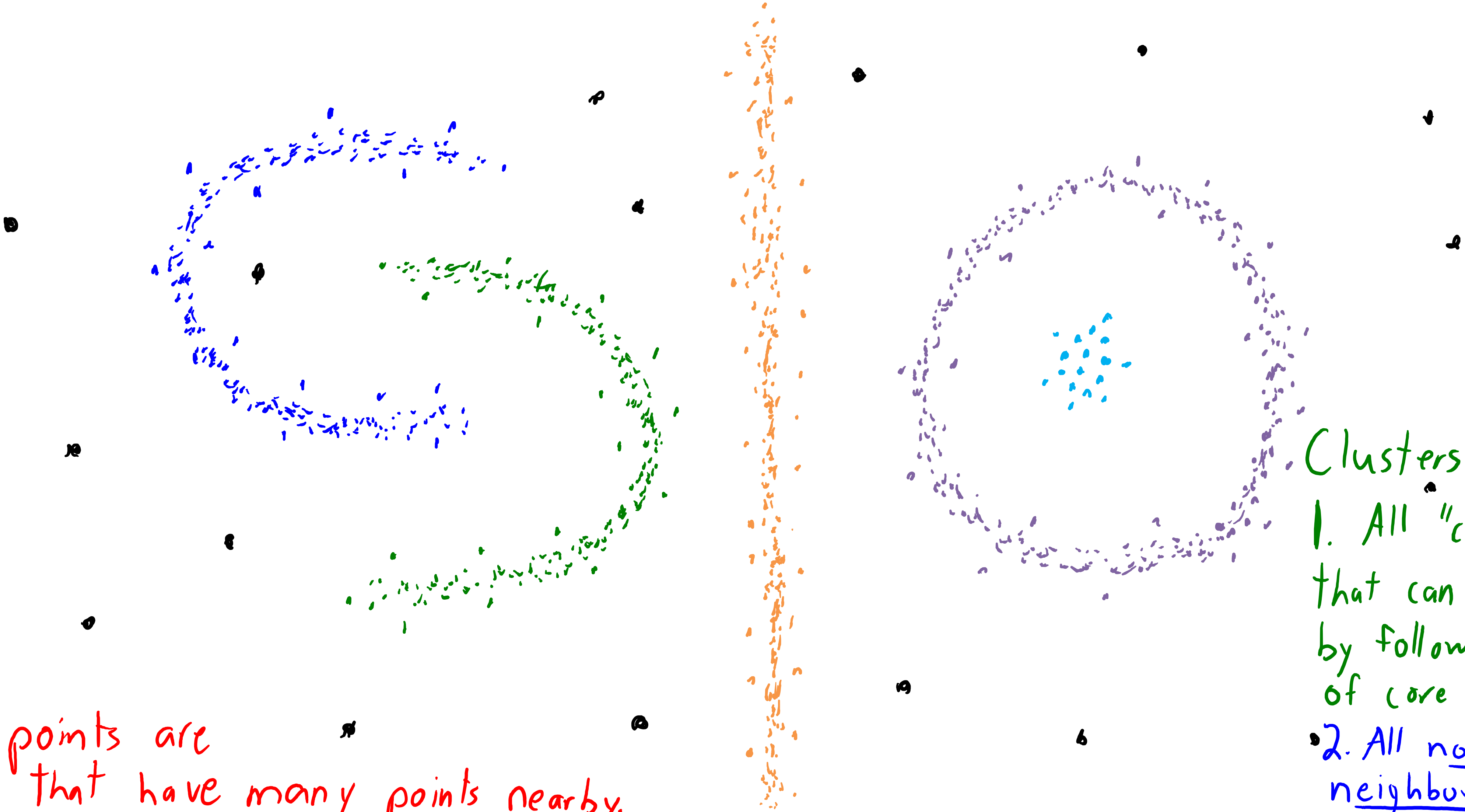


Clusters contain:

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

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

Density-Based Clustering



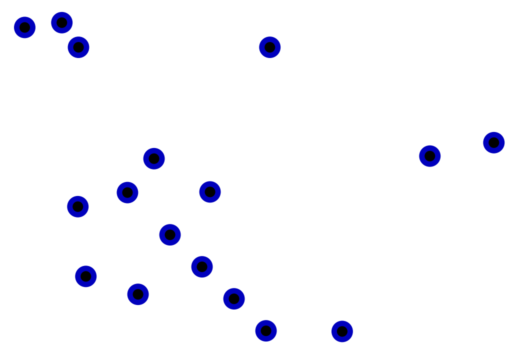
"Core" points are points that have many points nearby.

Clusters contain:

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

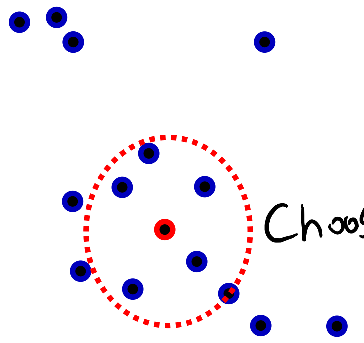
Density-Based Clustering (Example)

- Each “core” point defines a cluster:
  - Consisting of “core” point and all its “neighbours”.
- Merge clusters if “core” points are “neighbours” of each other.



Density-Based Clustering (Example)

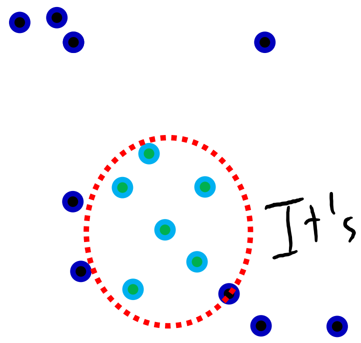
- Each "core" point defines a cluster:
  - Consisting of "core" point and all its "neighbours".
- Merge clusters if "core" points are "neighbours" of each other.



Choose point and check radius.

Density-Based Clustering (Example)

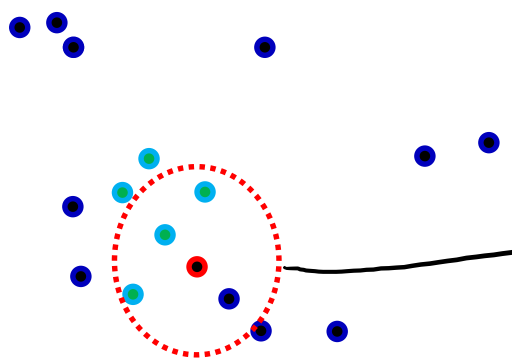
- Each “core” point defines a cluster:
  - Consisting of “core” point and all its “neighbours”.
- Merge clusters if “core” points are “neighbours” of each other.



It's a "core" point, so make a new cluster with points within epsilon.

Density-Based Clustering (Example)

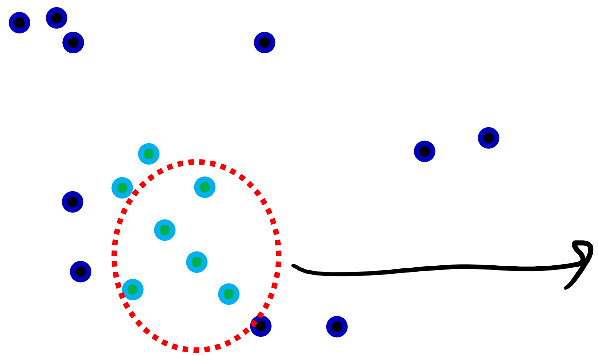
- Each “core” point defines a cluster:
  - Consisting of “core” point and all its “neighbours”.
- Merge clusters if “core” points are “neighbours” of each other.



Now test if a newly-added point is a “core” point.

Density-Based Clustering (Example)

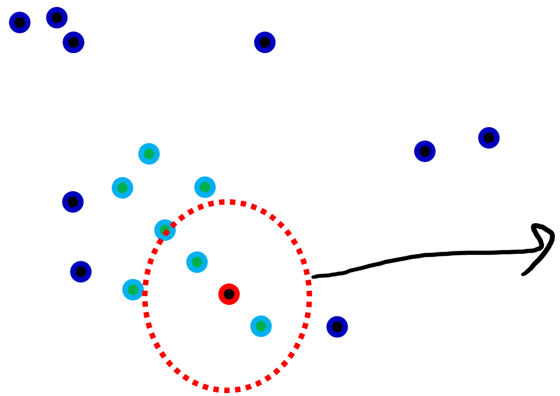
- Each “core” point defines a cluster:
  - Consisting of “core” point and all its “neighbours”.
- Merge clusters if “core” points are “neighbours” of each other.



It's a “core” point, so add its neighbours.

Density-Based Clustering (Example)

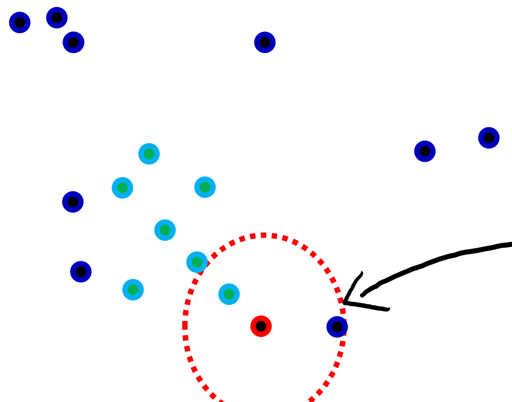
- Each "core" point defines a cluster:
  - Consisting of "core" point and all its "neighbours".
- Merge clusters if "core" points are "neighbours" of each other.



This is a "core" point too, so add its neighbours.

Density-Based Clustering (Example)

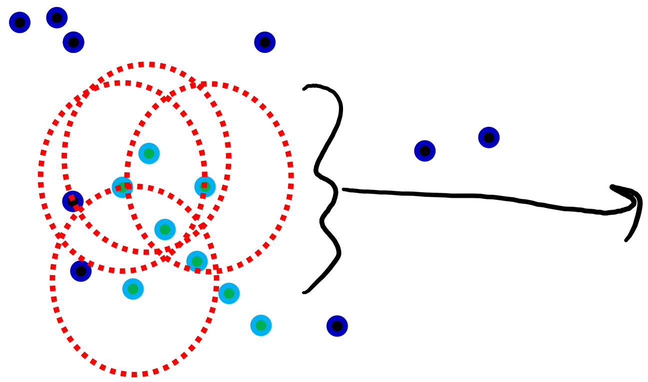
- Each "core" point defines a cluster:
  - Consisting of "core" point and all its "neighbours".
- Merge clusters if "core" points are "neighbours" of each other.



Not a "core" point, so do not add its neighbour.

Density-Based Clustering (Example)

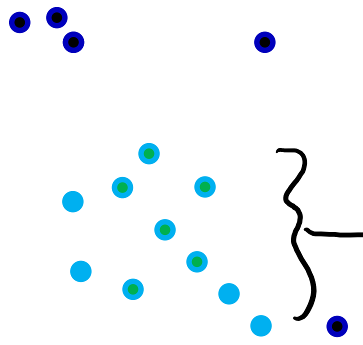
- Each "core" point defines a cluster:
  - Consisting of "core" point and all its "neighbours".
- Merge clusters if "core" points are "neighbours" of each other.



Continue testing whether the other cluster point are "core" points.

Density-Based Clustering (Example)

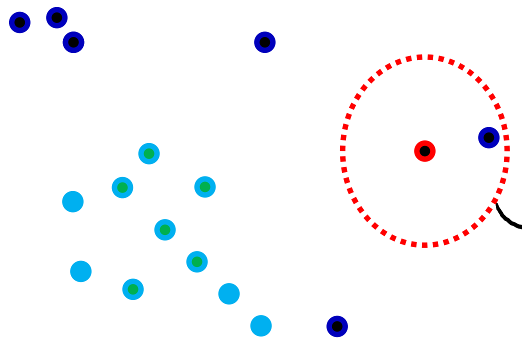
- Each "core" point defines a cluster:
  - Consisting of "core" point and all its "neighbours".
- Merge clusters if "core" points are "neighbours" of each other.



Eventually, there will be no new "core" points to "expand."

Density-Based Clustering (Example)

- Each "core" point defines a cluster:
  - Consisting of "core" point and all its "neighbours".
- Merge clusters if "core" points are "neighbours" of each other.



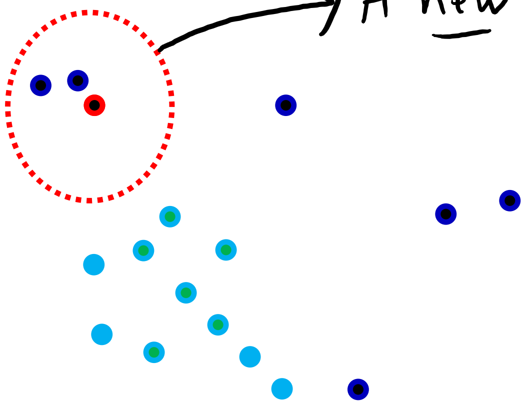
Not enough points in radius => not a "core" point.

Don't assign cluster

Density-Based Clustering (Example)

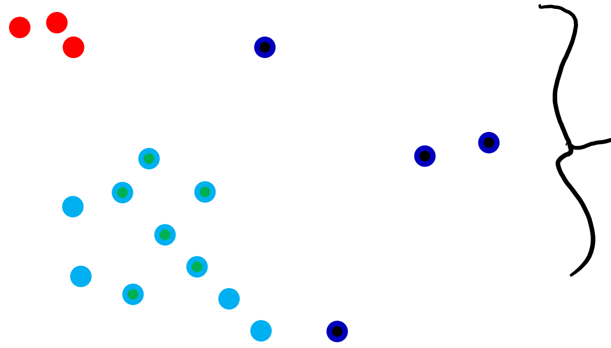
- Each “core” point defines a cluster:
  - Consisting of “core” point and all its “neighbours”.
- Merge clusters if “core” points are “neighbours” of each other.

A new "core" point, so make a new cluster.



Density-Based Clustering (Example)

- Each "core" point defines a cluster:
  - Consisting of "core" point and all its "neighbours".
- Merge clusters if "core" points are "neighbours" of each other.



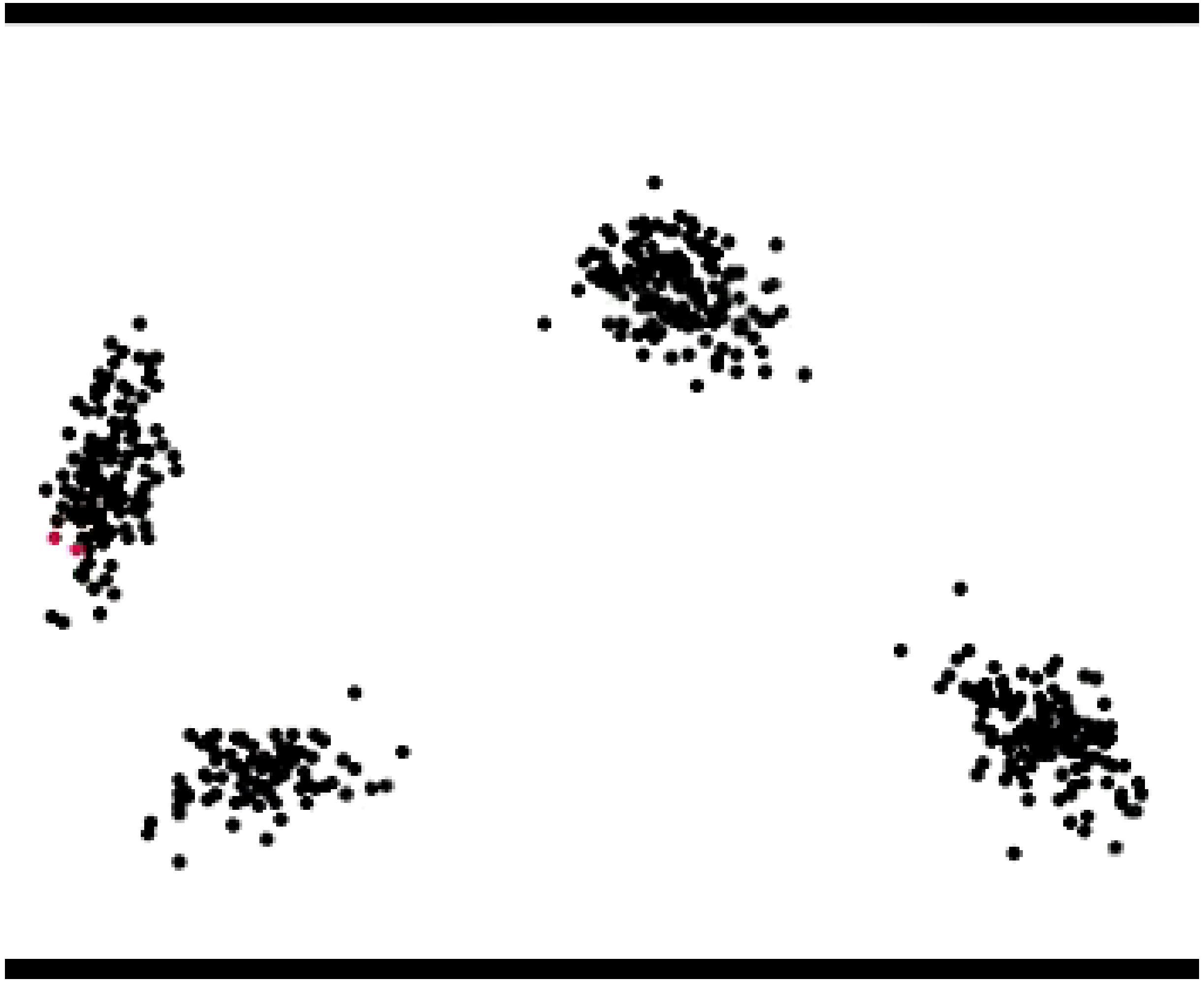
Final result.

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 "expand" the cluster.

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

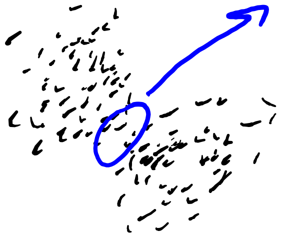
Density-Based Clustering in Action



[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:



Points at "boundary" between clusters.

- Sensitive to the choice of ε and minNeighbours.
  - Otherwise, not sensitive to initialization (except for boundary points).
- If you get a new example, finding cluster is expensive.
  - Need to compute distances to training points.
- In high-dimensions, need a lot of points to 'fill' the space.

(pause)

Ensemble Clustering

question — stop following — 23 views

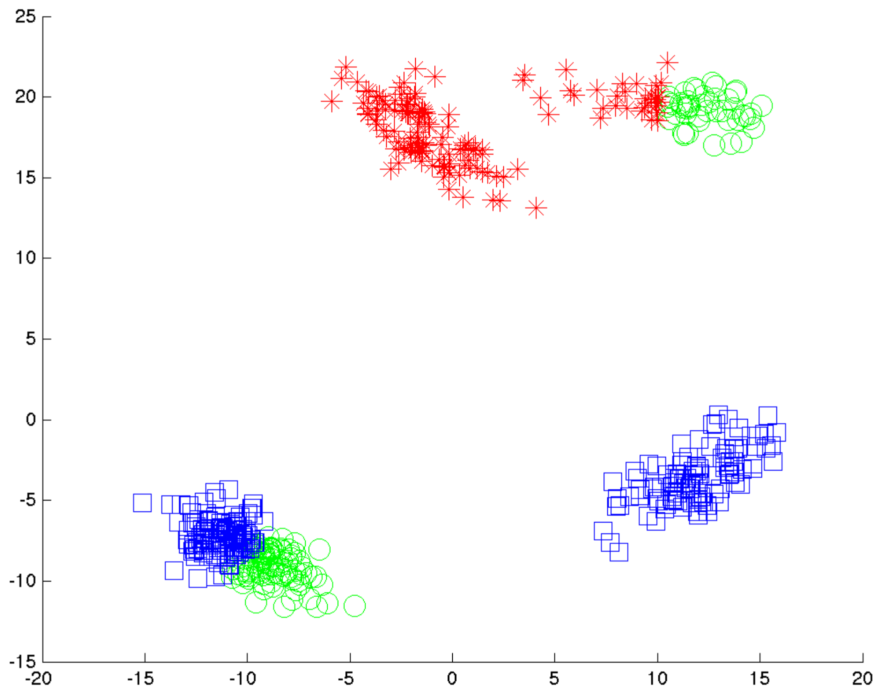
Multiple random runs of K means

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

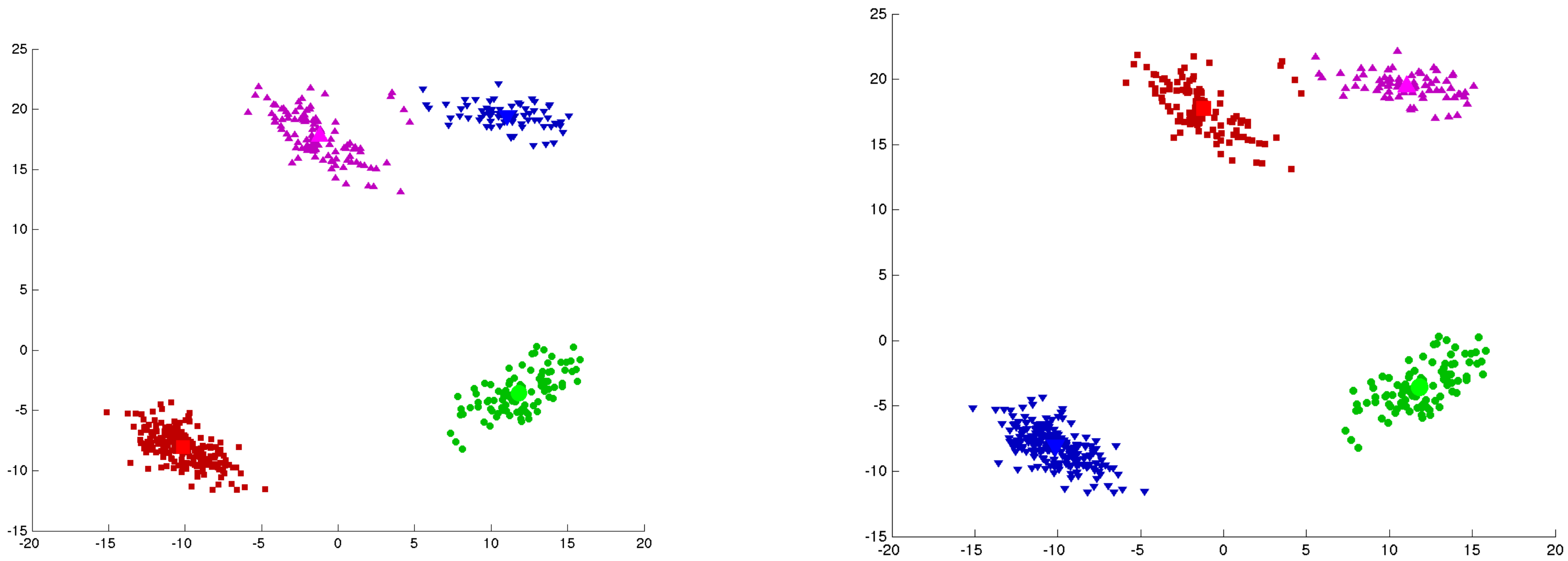
- E.g., run k-means 20 times and then cluster using the mode of each $\hat{y}_i$.
- Normally, averaging across models doing different things is good.



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

Label Switching Problem

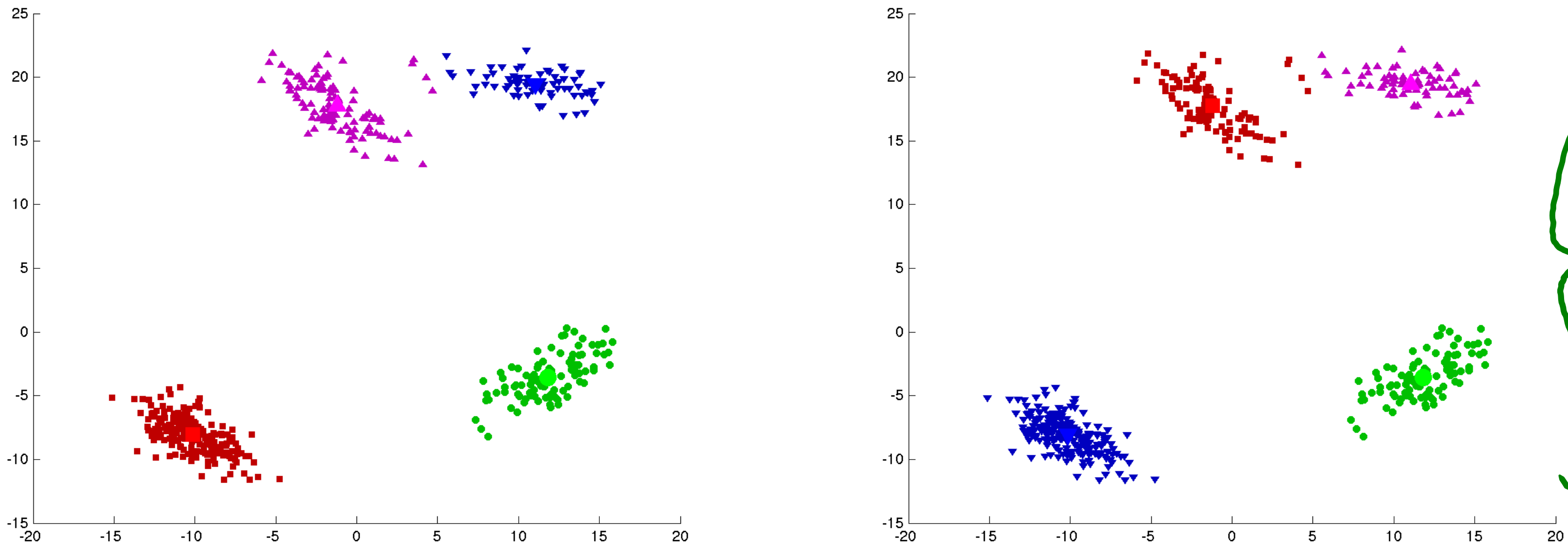
- This doesn't work because of "label switching" problem:
  - The cluster labels $\hat{y}_i$ are meaningless.
  - We could get same clustering with permuted labels ("exchangeable"):



  - 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.



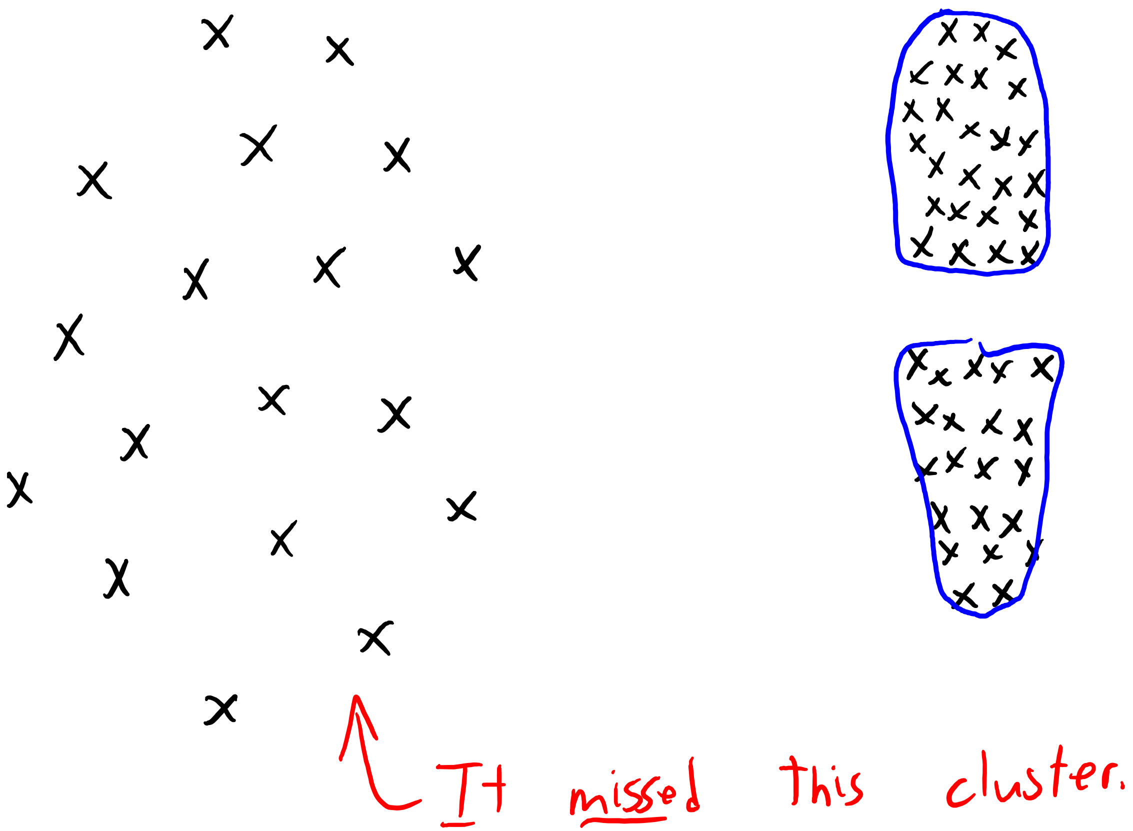
Different permutation of labels but same groups of points.

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

(pause)

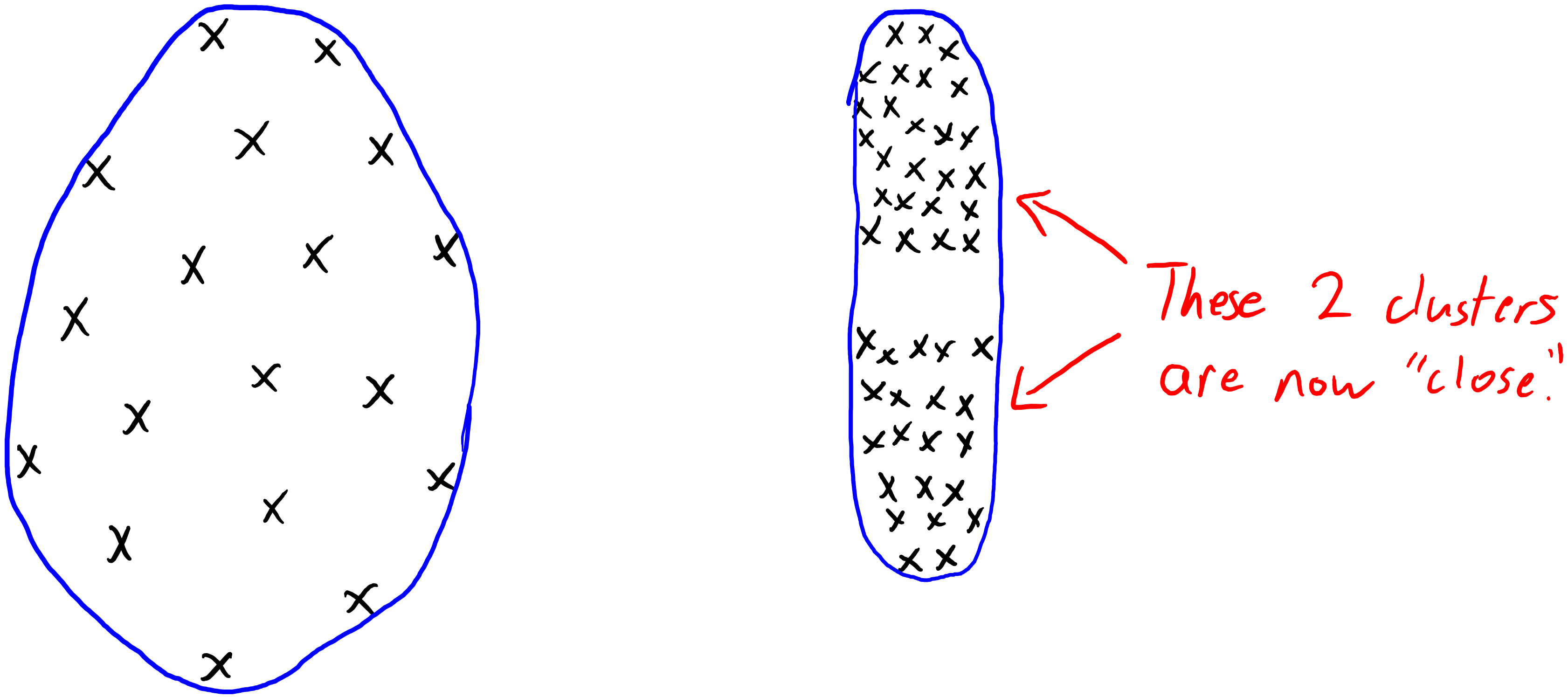
Differing Densities

- Consider density-based clustering on this data:



Differing Densities

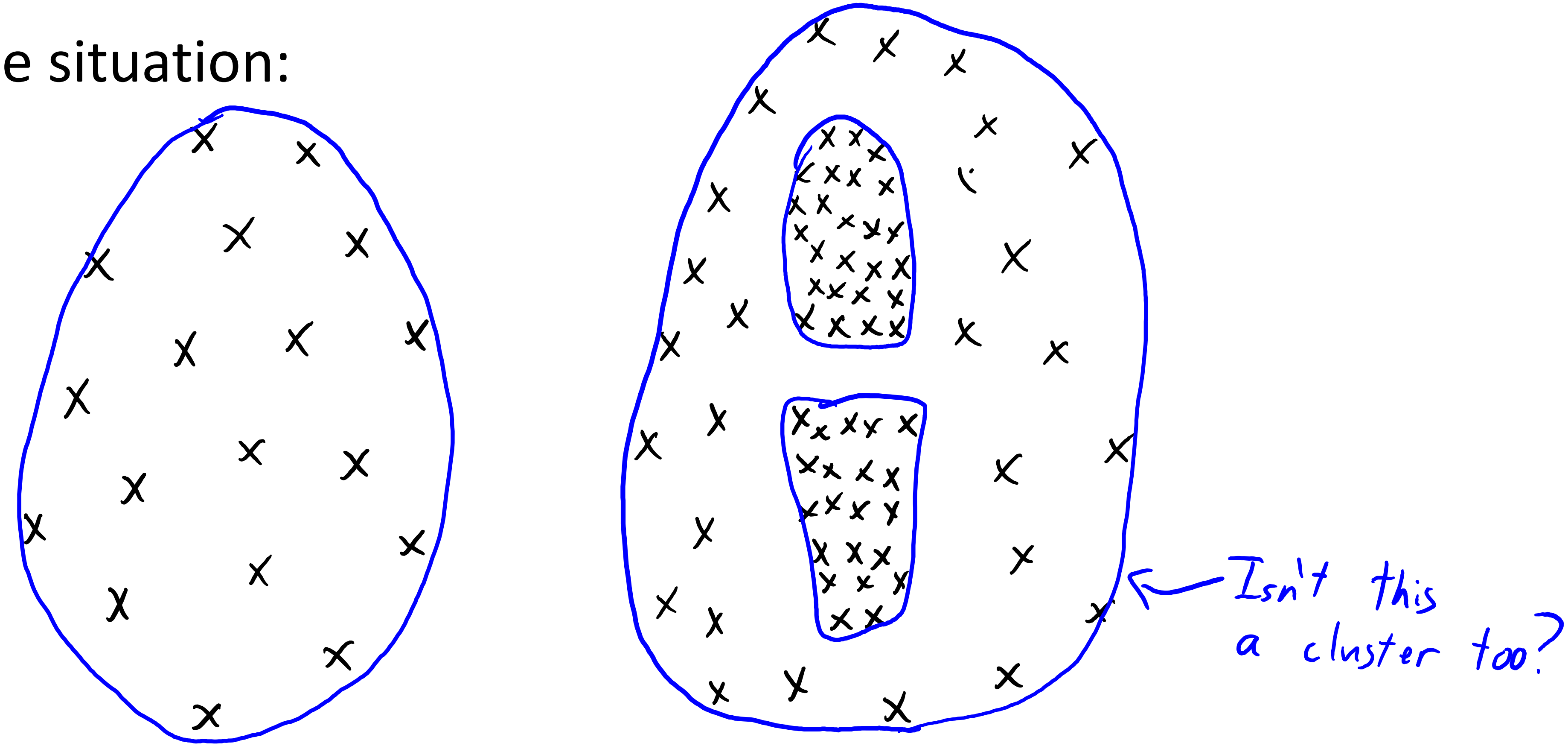
- Increase epsilon and run it again:



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

Differing Densities

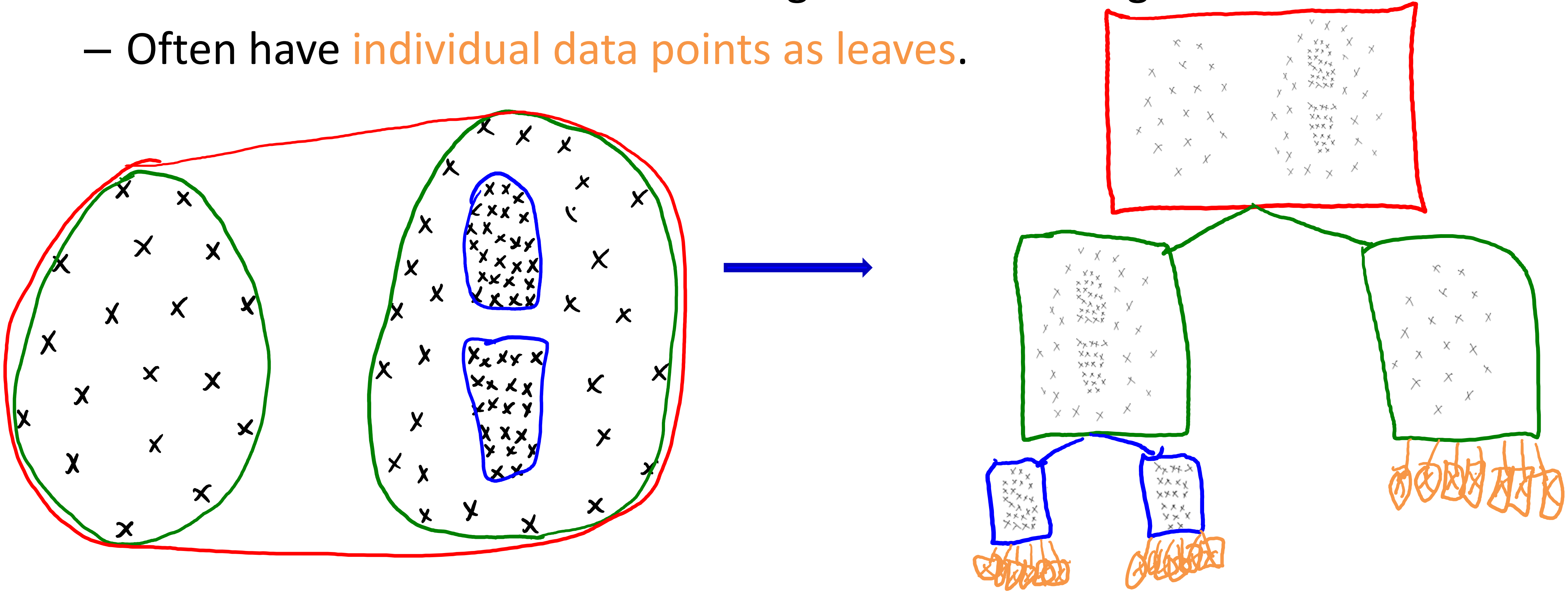
- Here is a worse situation:



- 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.

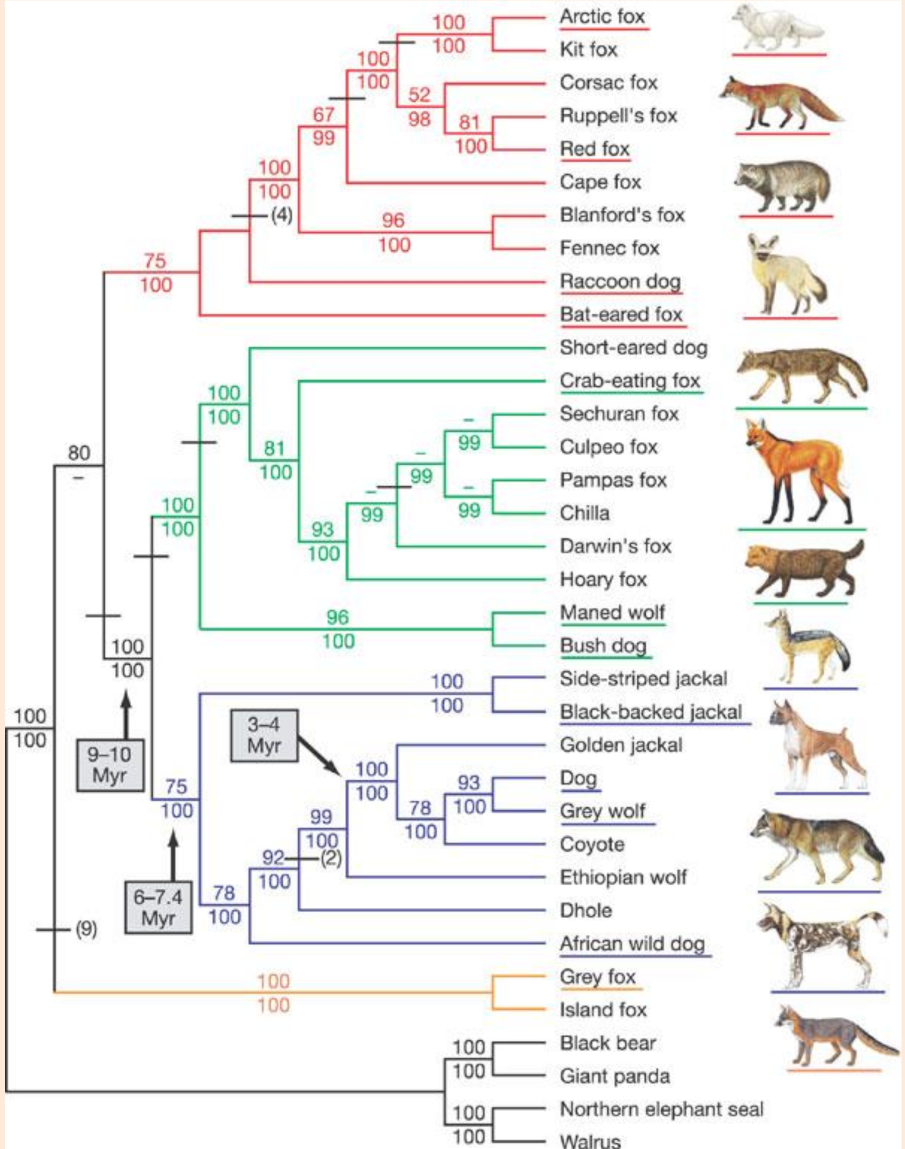


Application: Phylogenetics

- 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’.

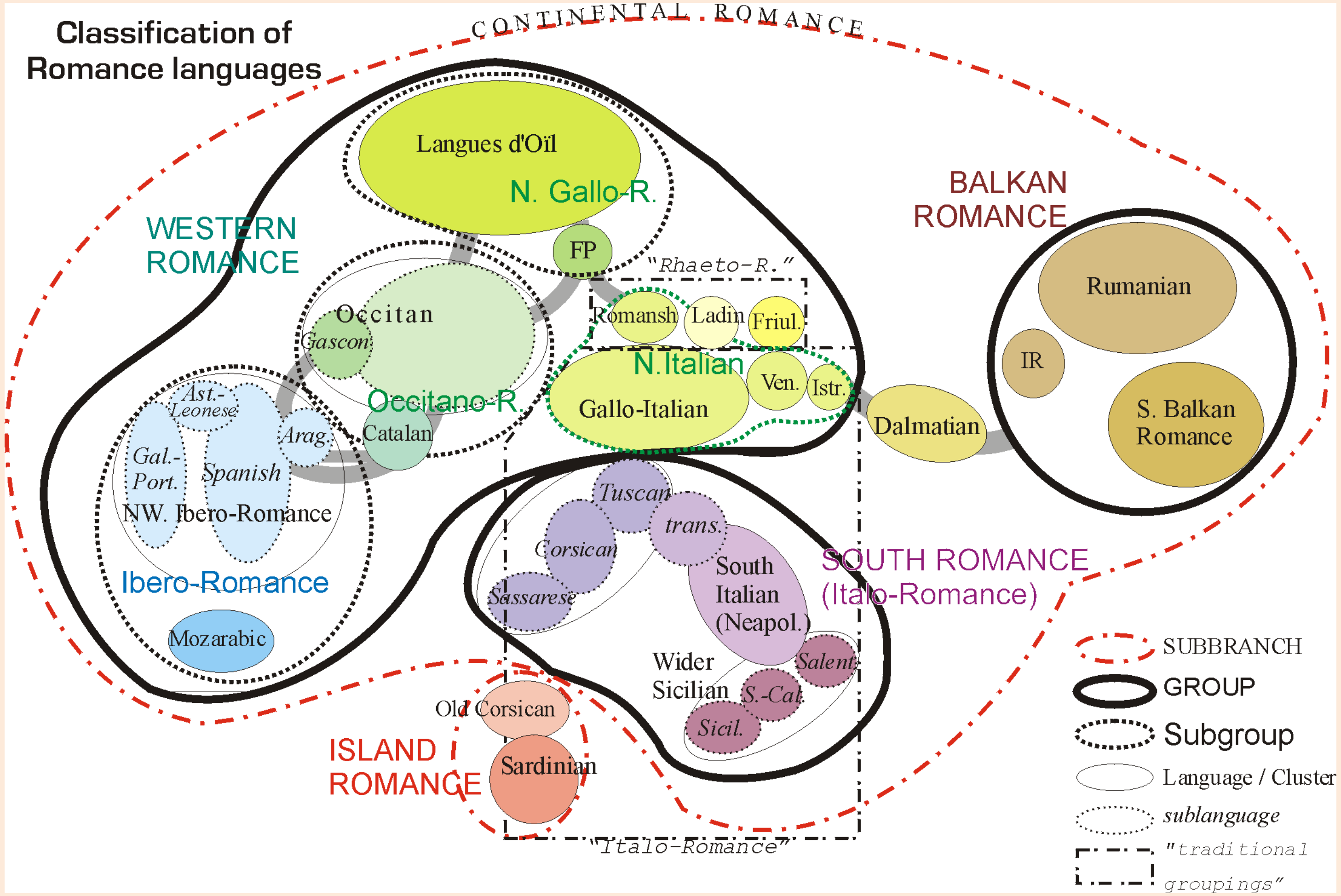
- More useful information in the plot:
  - Line lengths: chose here to approximate time.
  - Numbers: #clusterings across bootstrap samples.
  - ‘Outgroups’ (walrus, panda) are a sanity check.



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

Application: Phylogenetics

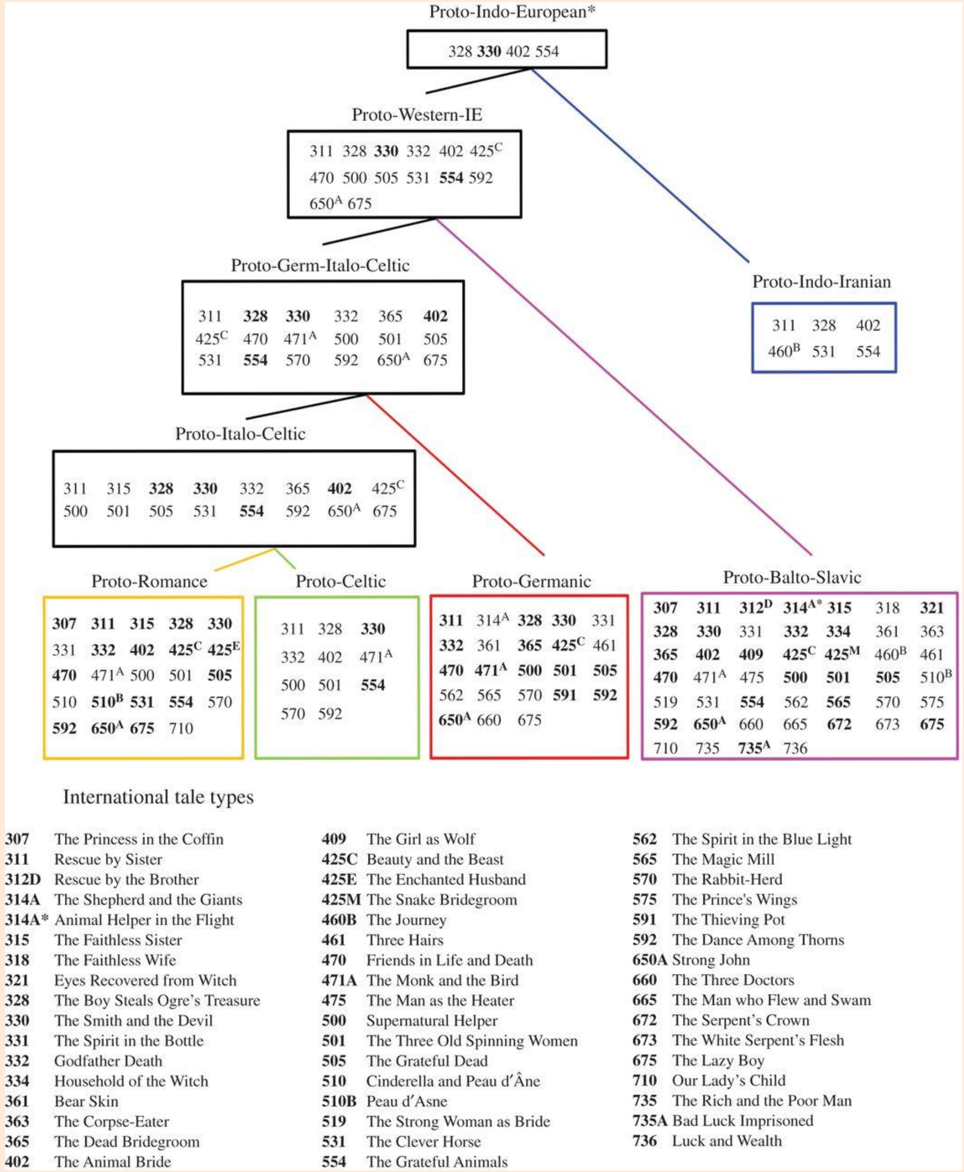
- Comparative method in linguistics studies evolution of languages:



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

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.

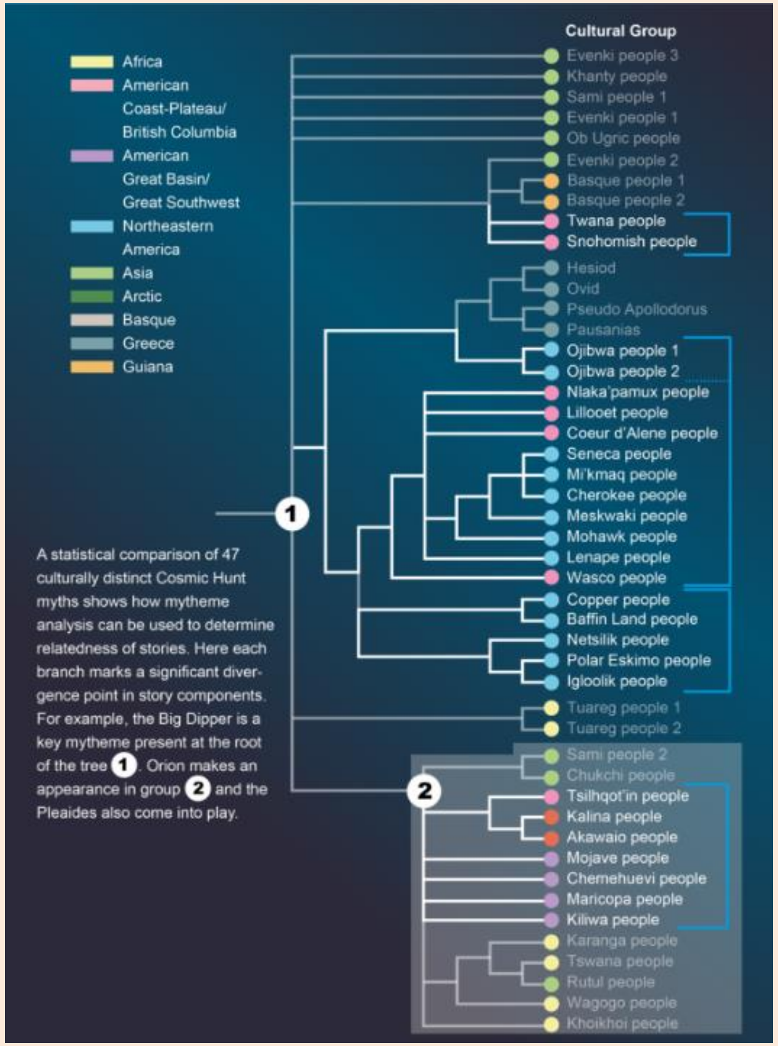


http://rsos.royalsocietypublishing.org/content/3/1/150645

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.

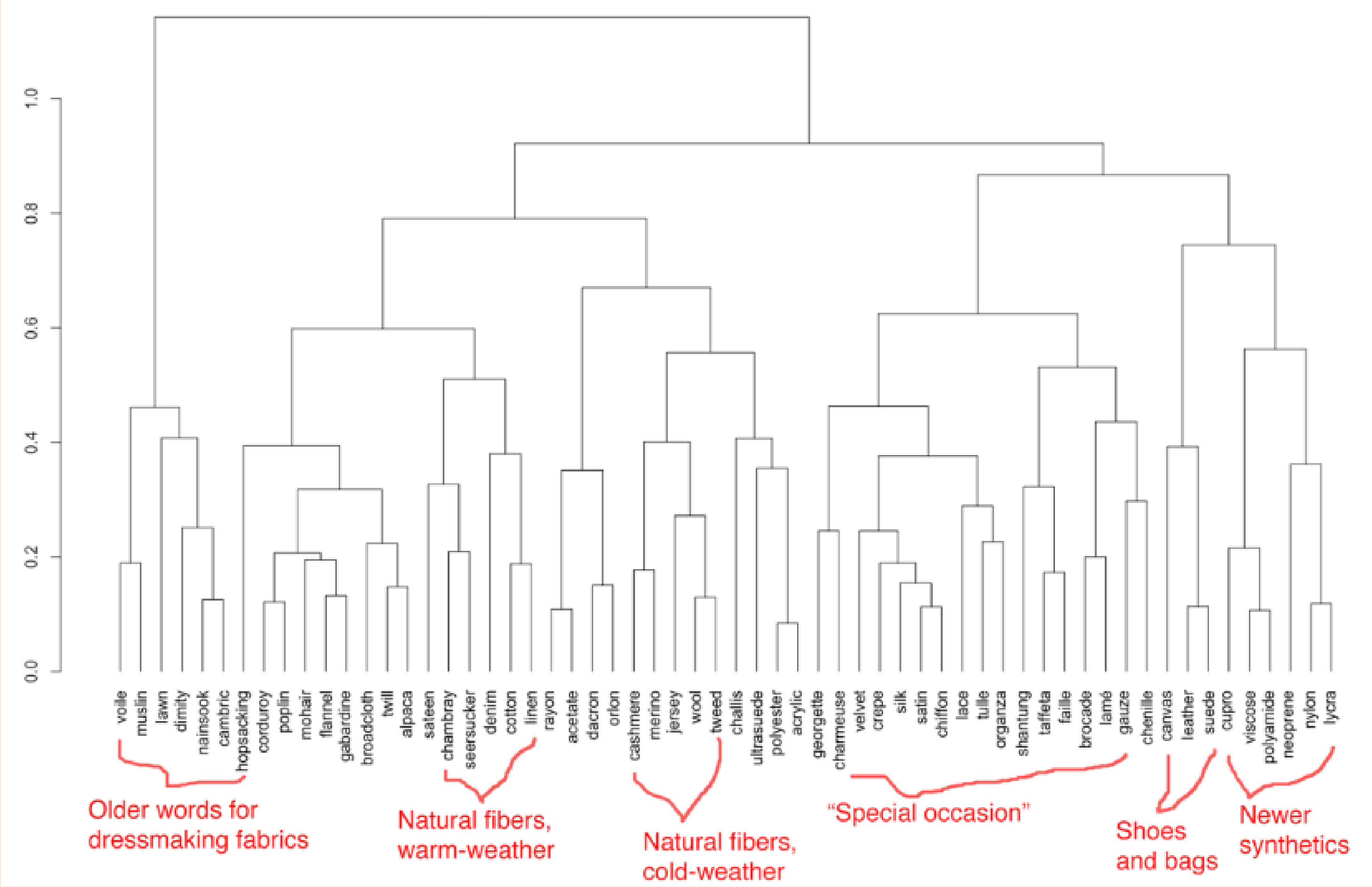
- September 2016: evolution of myths.
  - “Comic hunt” story:
    - Person hunts animal that becomes constellation.
      - Previously known to be at least 15,000 years old.
    - May go back to paleololithic period.



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

Application: Fashion?

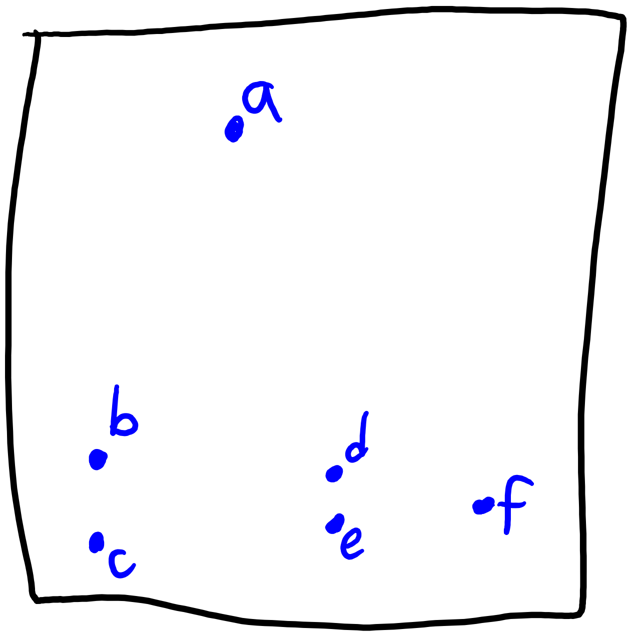
- Hierarchical clustering of clothing material words in Vogue:

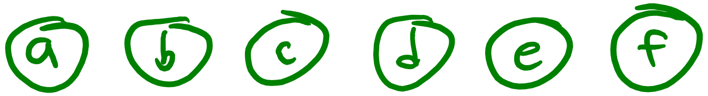


http://dh.library.yale.edu/projects/vogue/fabricspace/

Agglomerative (Bottom-Up) Clustering

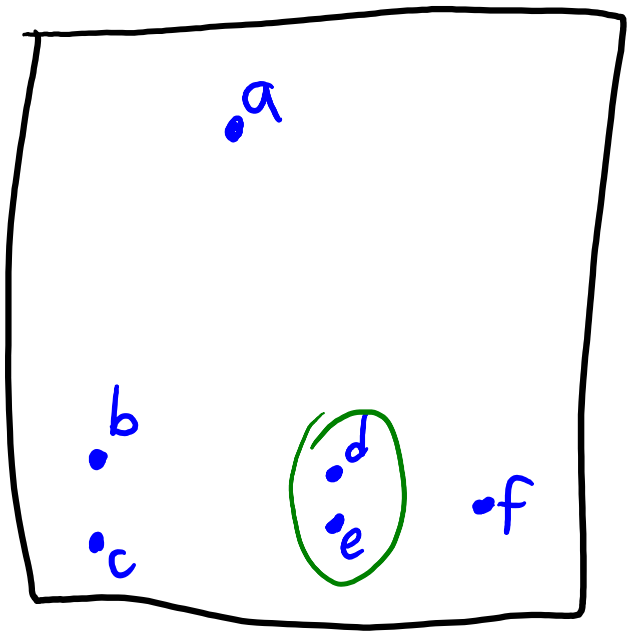
- More common hierarchical method: agglomerative clustering.
  1. Starts with each point in its own cluster.

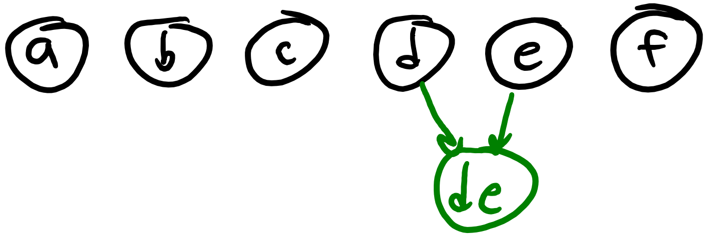




Agglomerative (Bottom-Up) Clustering

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

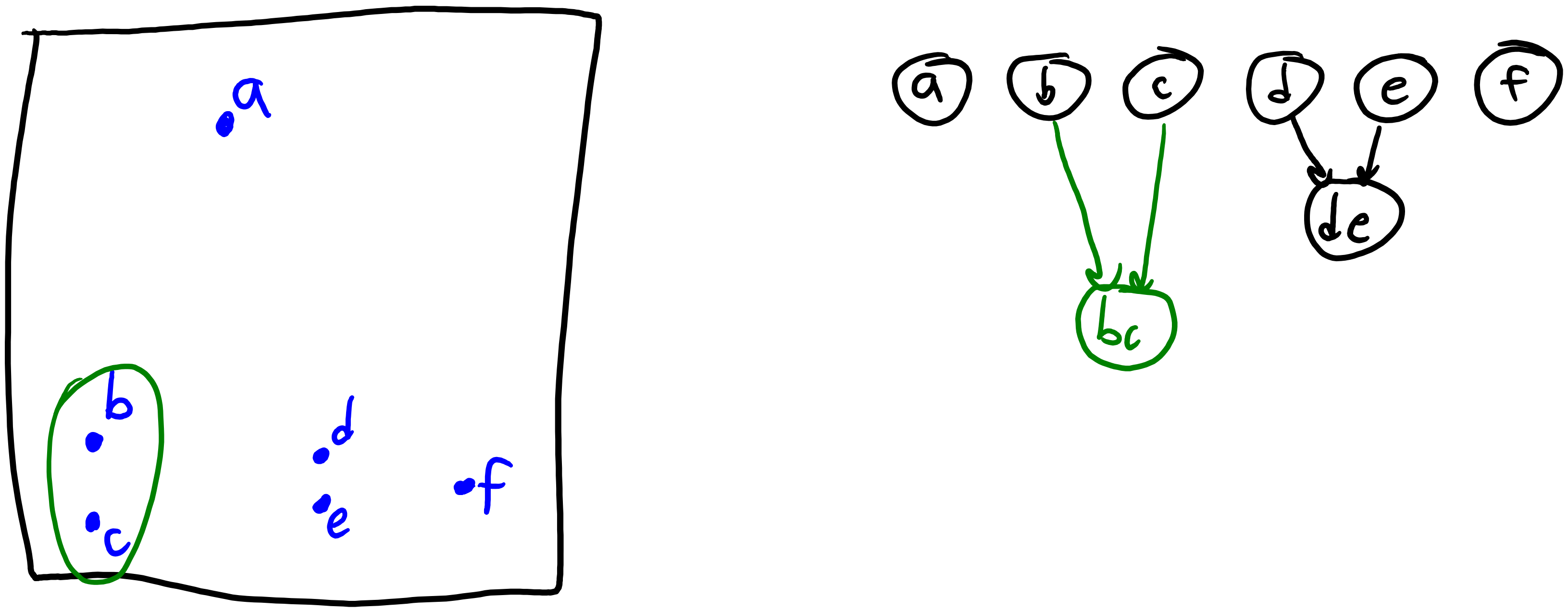




https://en.wikipedia.org/wiki/Hierarchical_clustering

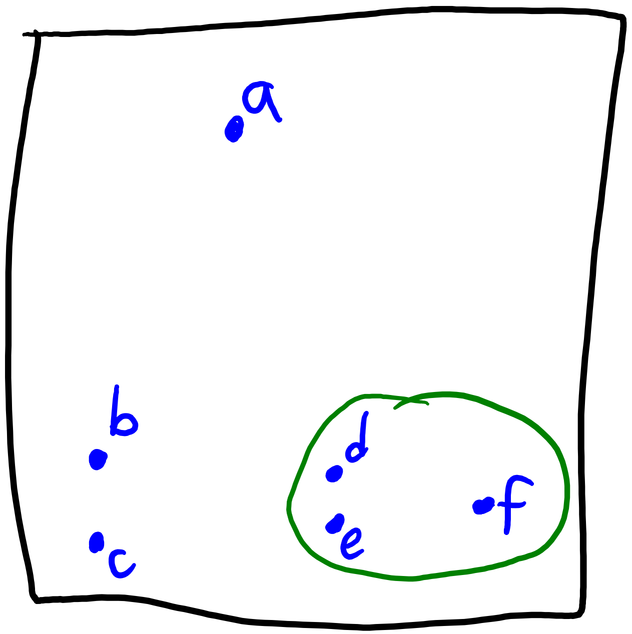
Agglomerative (Bottom-Up) Clustering

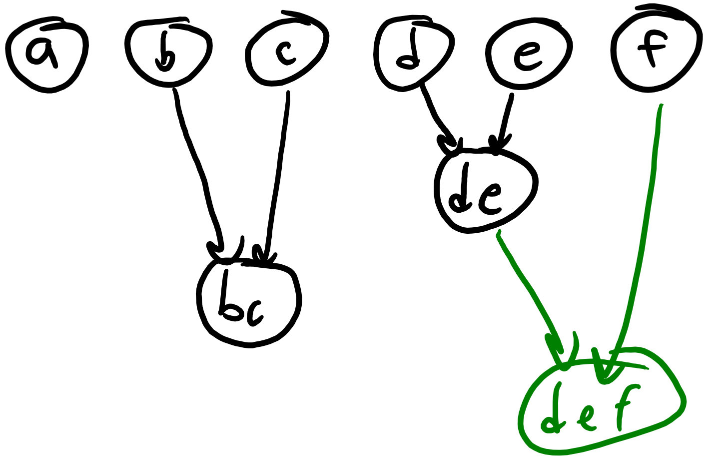
- More common hierarchical method: agglomerative clustering.
  1. Starts with each point in its own cluster.
  2. Each step merges the two “closest” clusters.



Agglomerative (Bottom-Up) Clustering

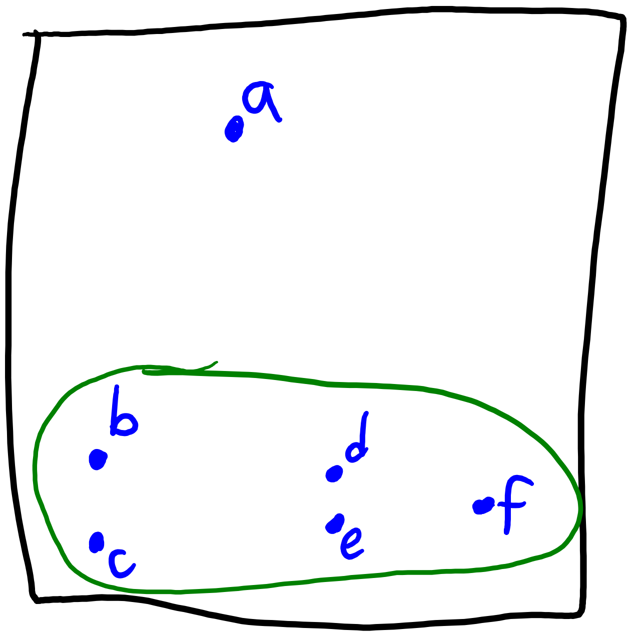
- More common hierarchical method: agglomerative clustering.
  1. Starts with each point in its own cluster.
  2. Each step merges the two "closest" clusters.

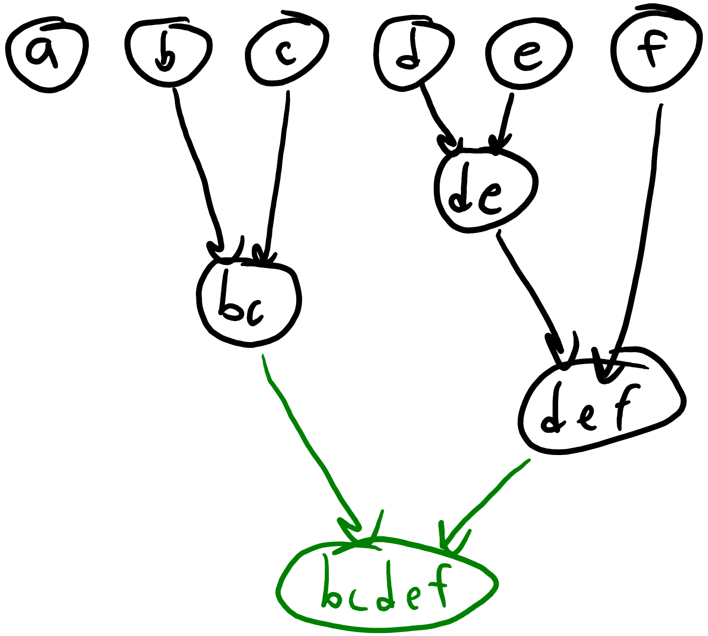




Agglomerative (Bottom-Up) Clustering

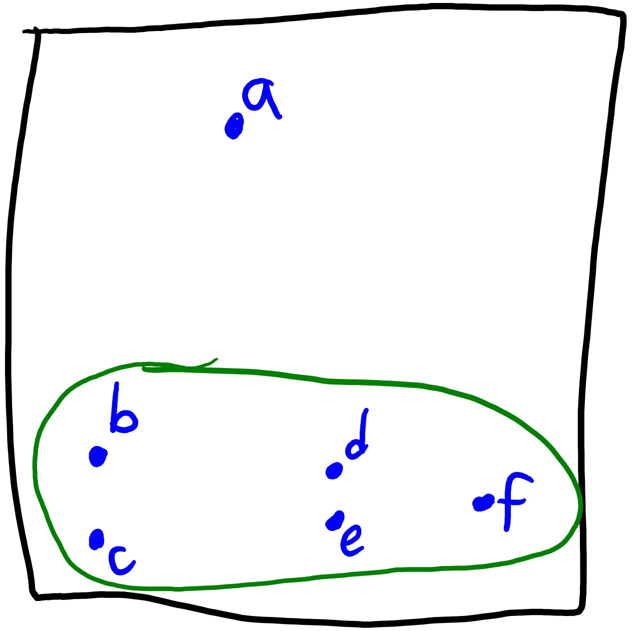
- More common hierarchical method: agglomerative clustering.
  1. Starts with each point in its own cluster.
  2. Each step merges the two "closest" clusters.

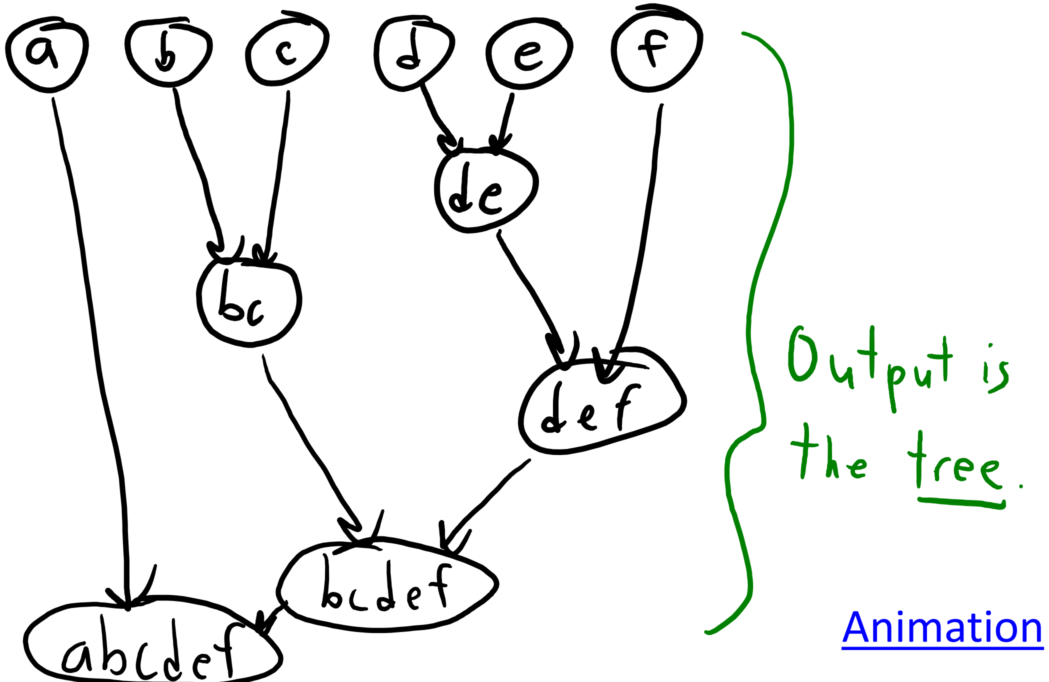




Agglomerative (Bottom-Up) Clustering

- More 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

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 $O(n^3d)$ for basic implementation.
  - Each step costs $O(n^2d)$, and each step might only cluster 1 new point.

Summary

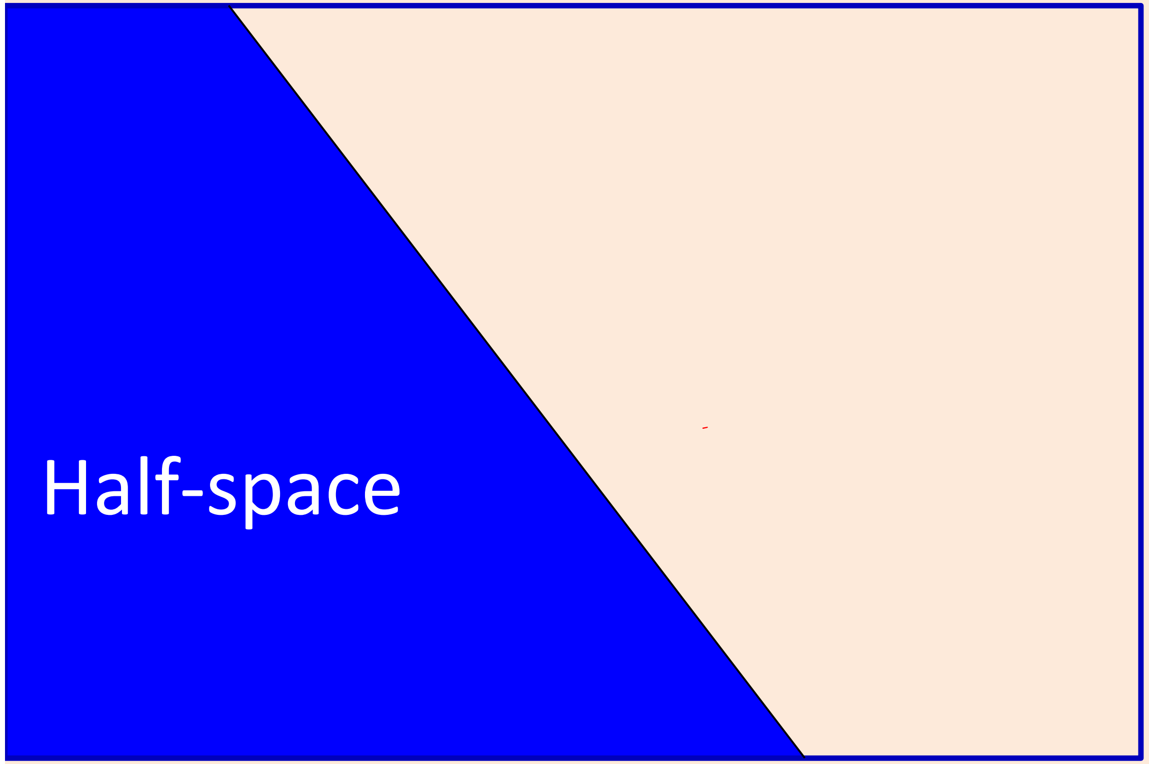
- Shape of K-means clusters:
  - Partitions space into convex sets.
- Density-based clustering:
  - “Expand” and “merge” dense regions of points to find clusters.
  - Not sensitive to initialization or outliers.
  - Useful for finding non-convex connected clusters.
- Ensemble clustering: combines multiple 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.

- Next time:
  - Discovering (and then ignoring) a hole in the ozone layer.

Why are k-means clusters convex?

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

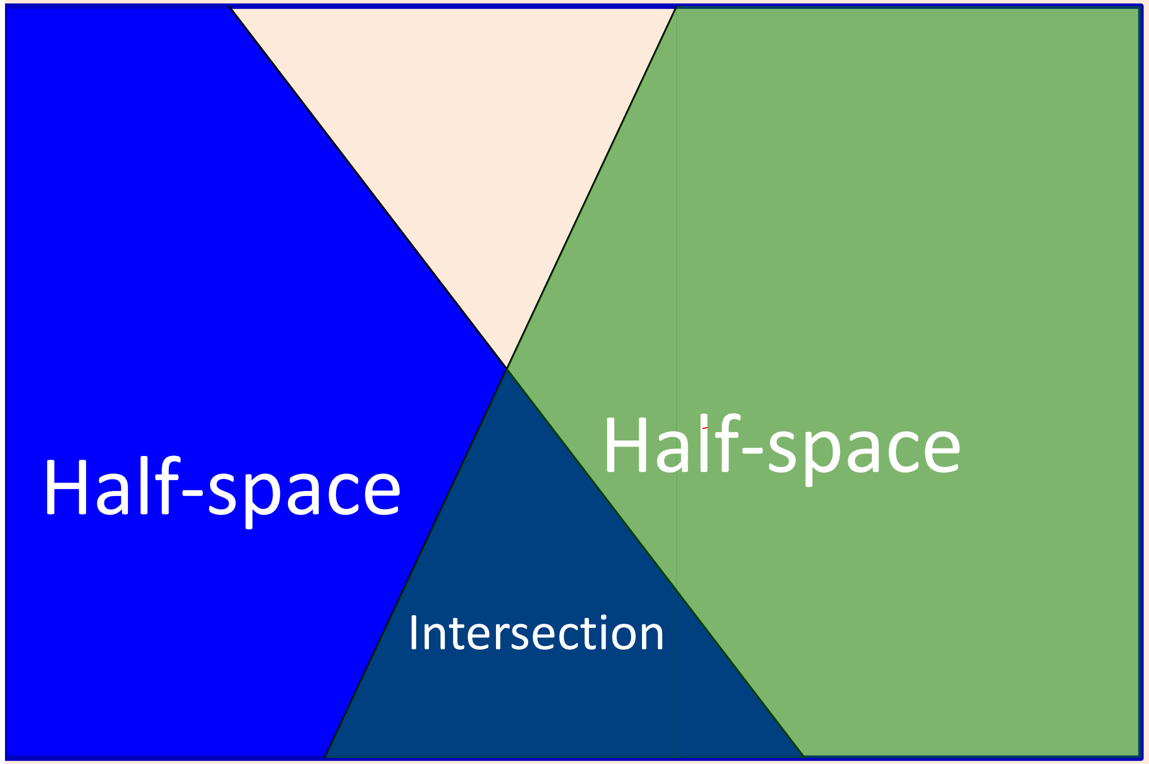
Half-space is Set of points 'x' satifying a linear inequality, like $\sum_{j=1}^{d} a_j x_j \leq b$



Why are k-means clusters convex?

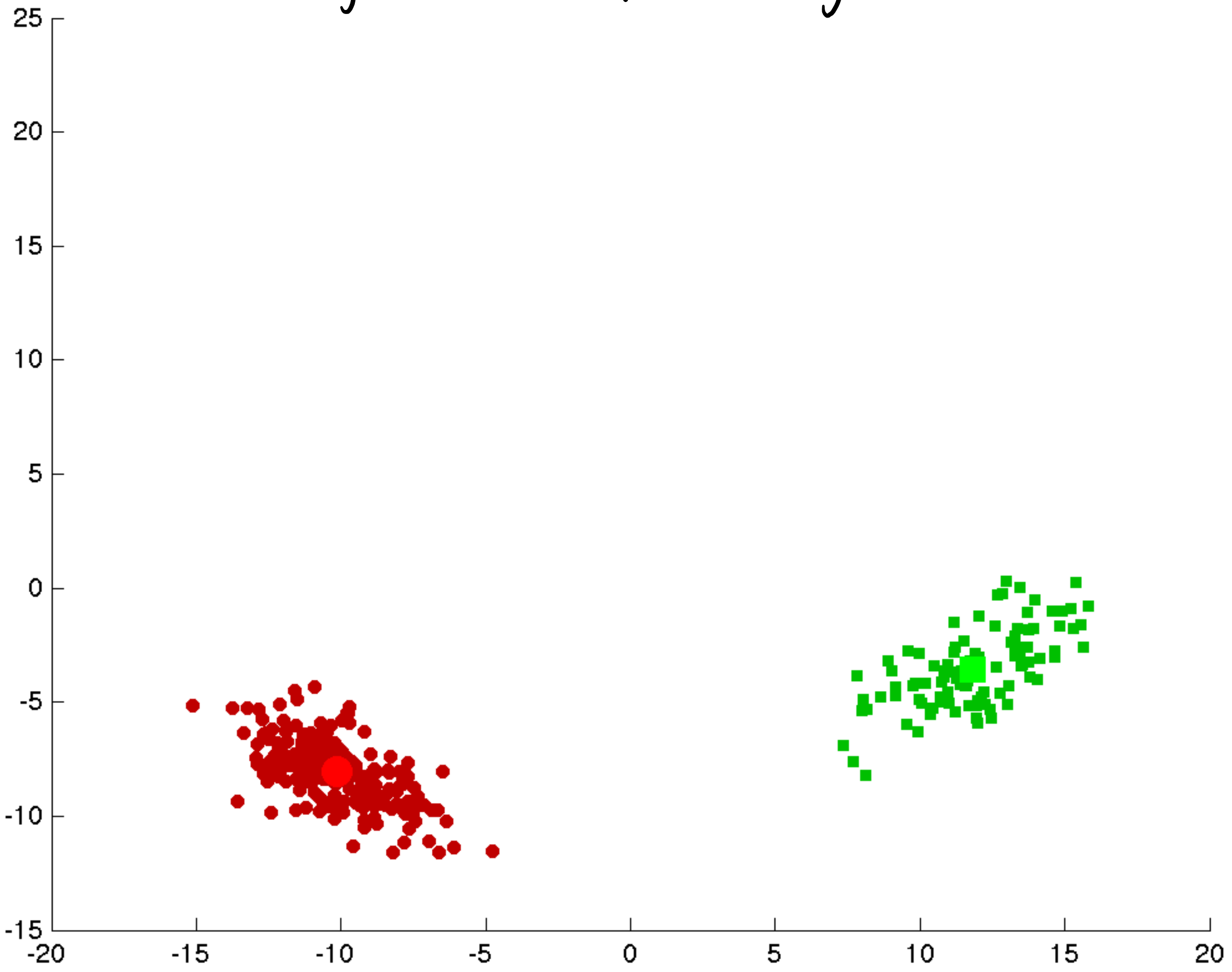
- K-means clusters are formed by the intersection of half-spaces.

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



Why are k-means clusters convex?

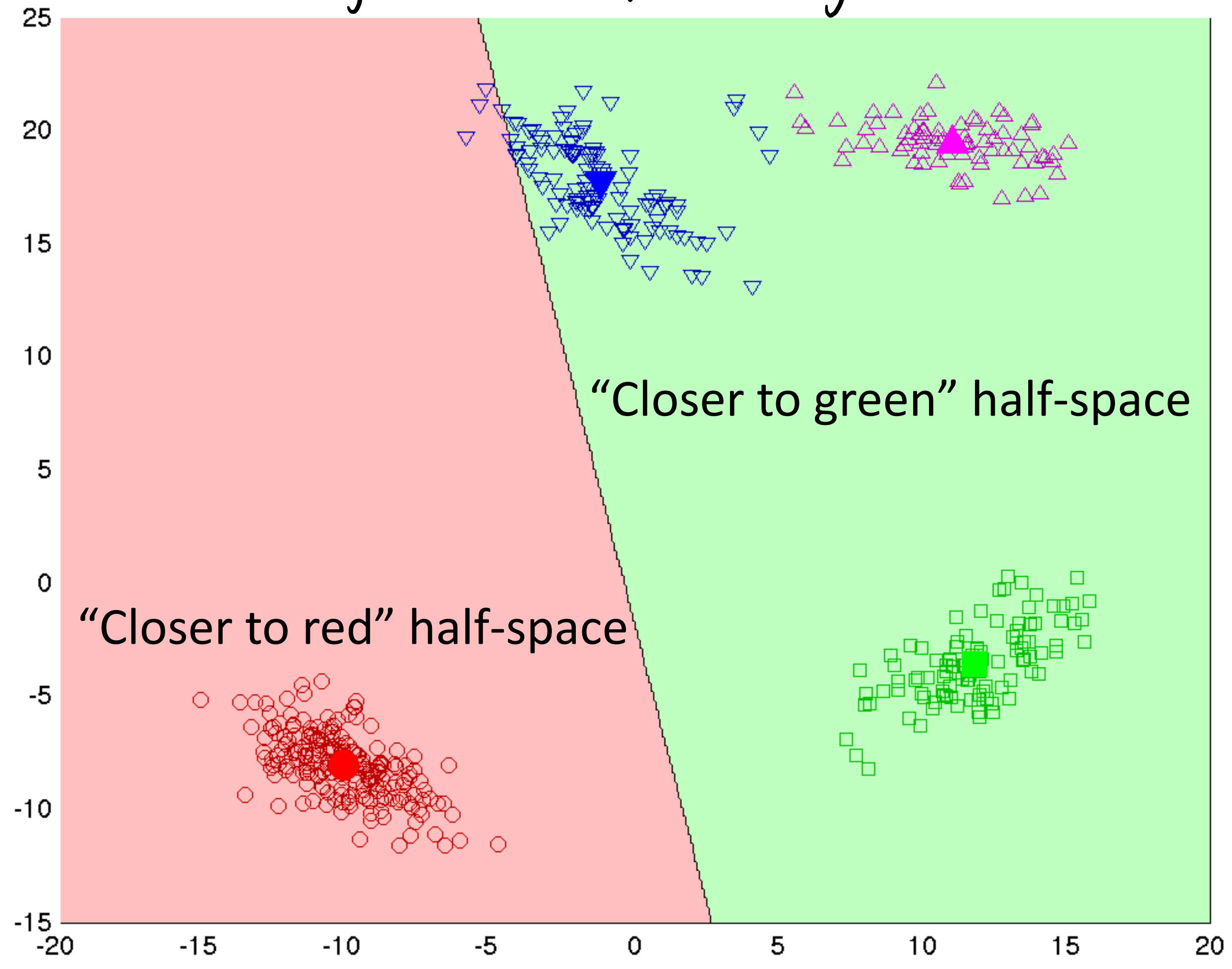
Which regions are put in green cluster?



Why are k-means clusters convex?

Why are k-means clusters convex?

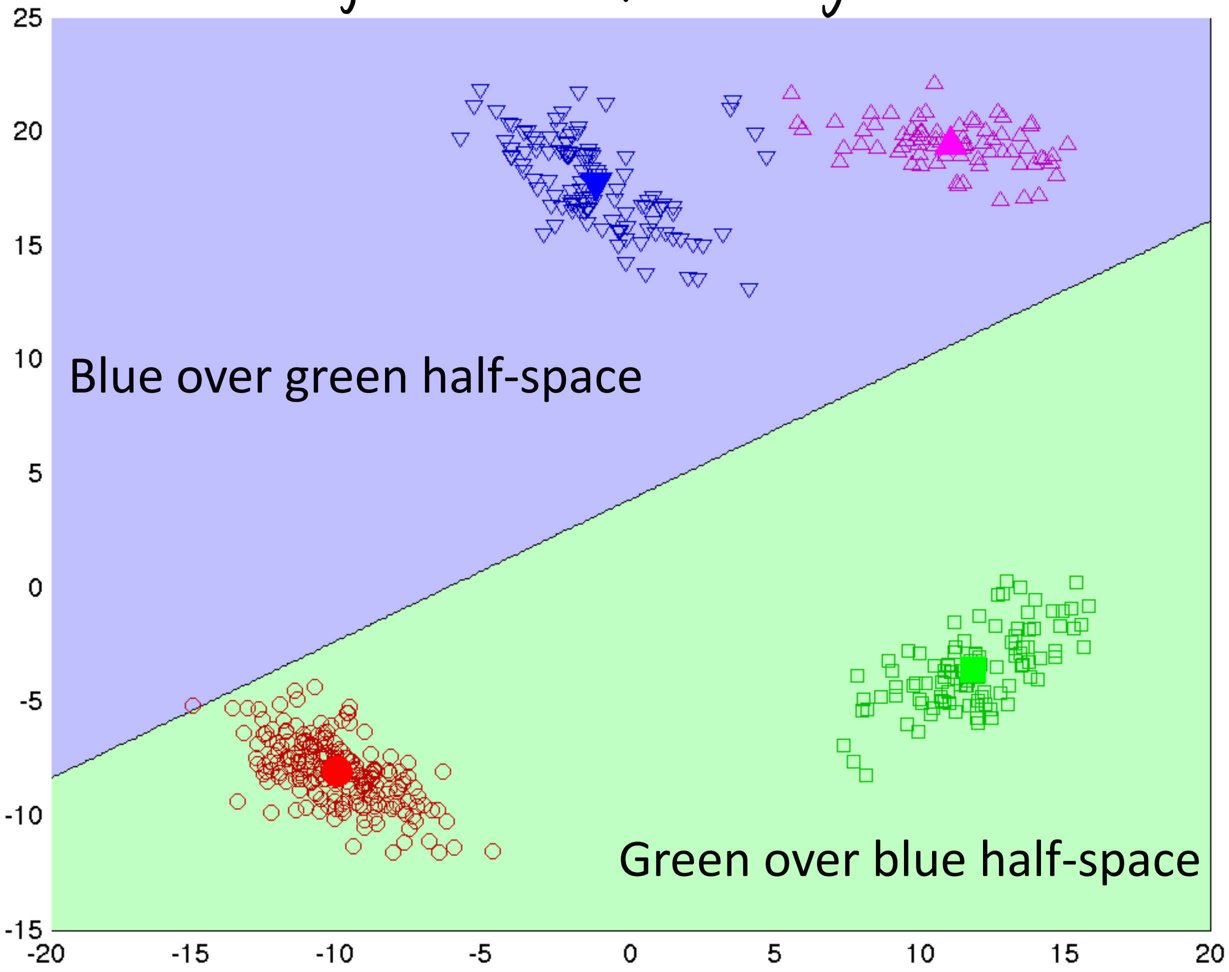
Which regions are put in green cluster?



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

Why are k-means clusters convex?

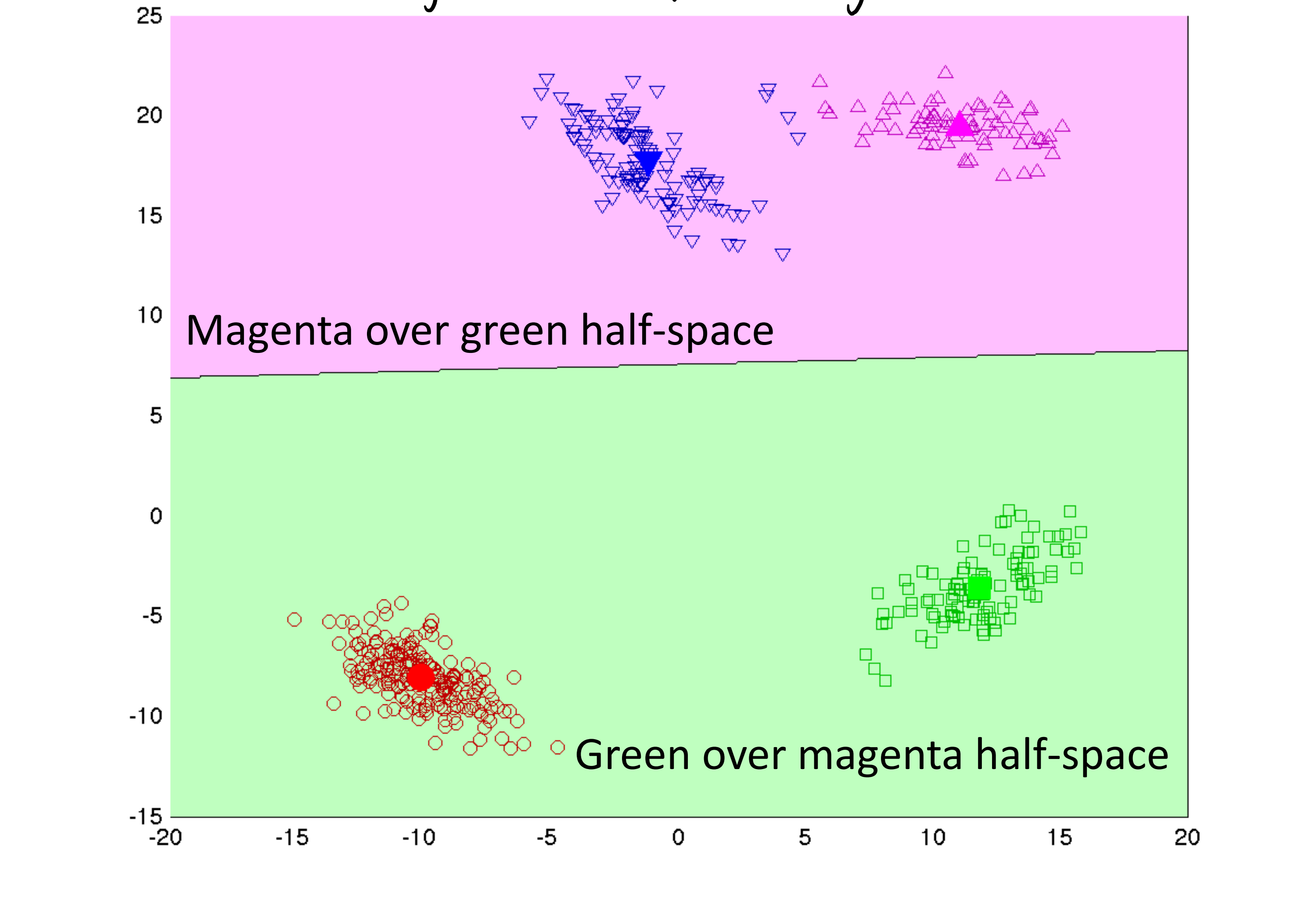
Which regions are put in green cluster?



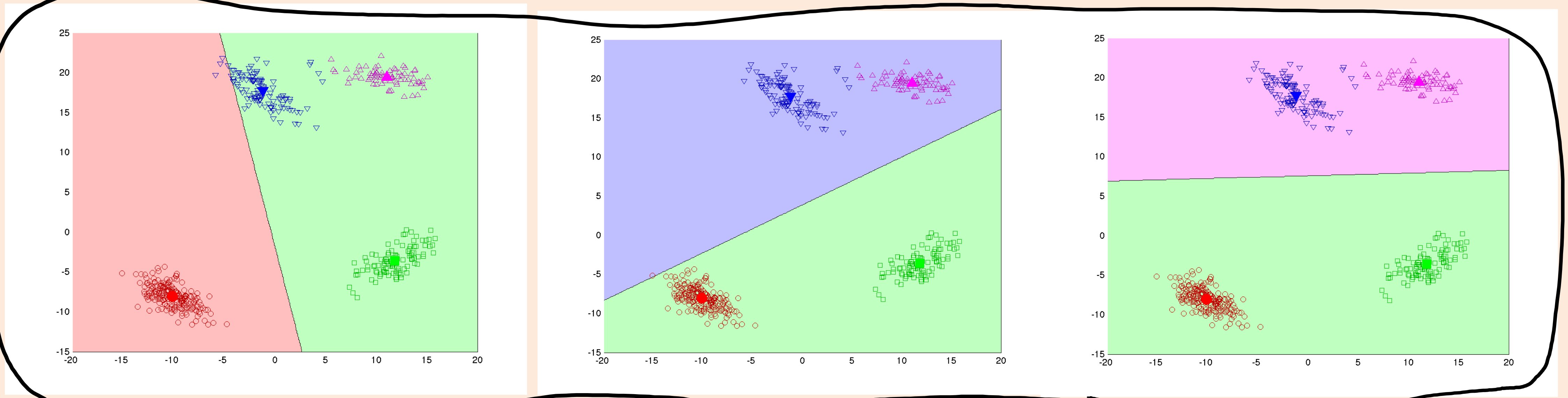
Blue vs. green decision defines different half-spaces.

Why are k-means clusters convex?

Which regions are put in green cluster?

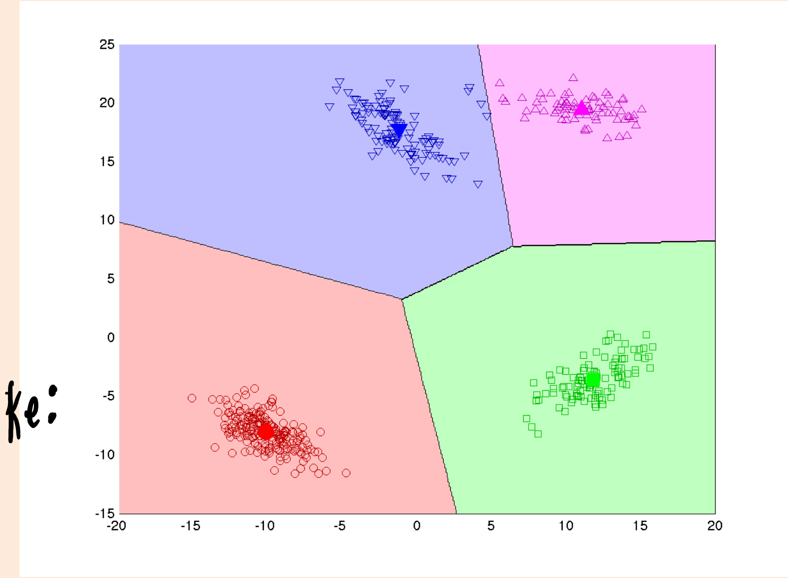


Why are k-means clusters convex?



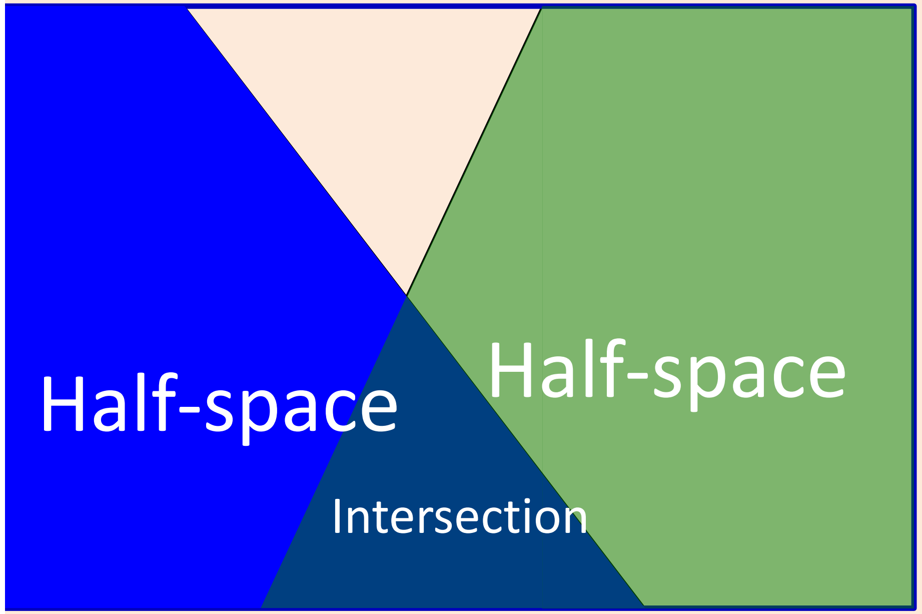
Green "cluster" is the intersection of these three half-spaces.

Here is what the four clusters look like:



Why are k-means clusters convex?

- Half-spaces are convex sets.
- Intersection of convex sets is a convex set.
- So intersection of half-spaces is convex.



Why are k-means clusters convex?

- Formal proof that "cluster 1" is convex (so all clusters are convex):

Let x_i and x_j be arbitrariy points in cluster 1.

→ By def'n of cluster 1, $\|x_i - w_1\| \leq \|x_i - w_c\|$ for all 'c'
$\|x_j - w_1\| \leq \|x_j - w_c\|$ for all 'c' } equality for $c=1$

→ Let x_m be an arbitrariy point between x_i and x_j.

→ So we can write it as $x_m = \theta x_i + (1-\theta)x_j$ for some $\theta \in [0,1]$

Then $\|x_m - w_1\| = \|\theta x_i + (1-\theta)x_j - (\theta w_1 + (1-\theta)w_1)\|$ $\quad (w_1 = \theta w_1 + (1-\theta)w_1)$

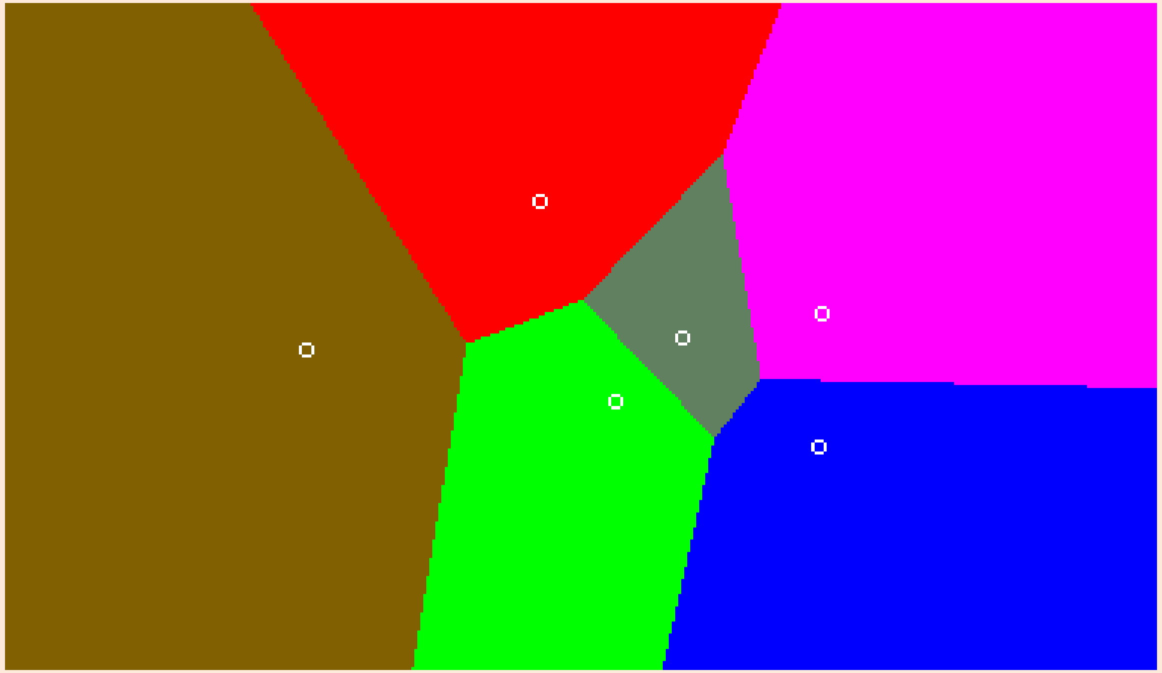
$\leq \|\theta x_i - \theta w_1\| + \|(1-\theta)x_j - (1-\theta)w_1\|$ (triangle inequality)

$= \theta\|x_i - w_1\| + (1-\theta)\|x_j - w_1\|$ (homogenity of norms)

x_i and x_j are in cluster 1 { $\leq \theta\|x_i - w_c\| + (1-\theta)\|x_j - w_c\| = \|x_j - w_c\|$ so x_m is in cluster 1.

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

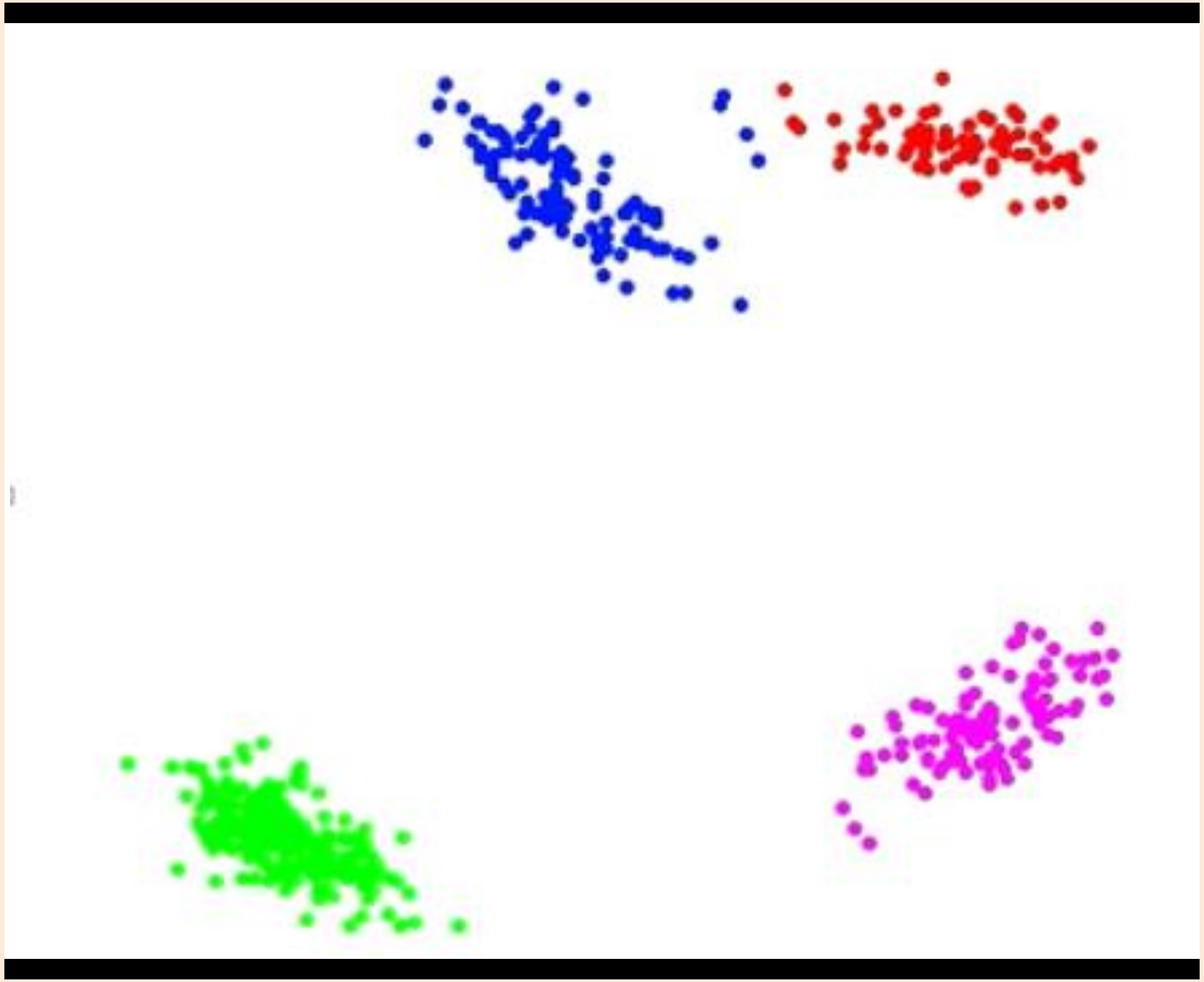
UBClustering Algorithm

- 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 $p(i,j) = \text{count}(x_i \text{ in same cluster as } x_j)/m$.
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.

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.

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.

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

- 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.