CPSC 340: Machine Learning and Data Mining

Density-Based Clustering

Fall 2017

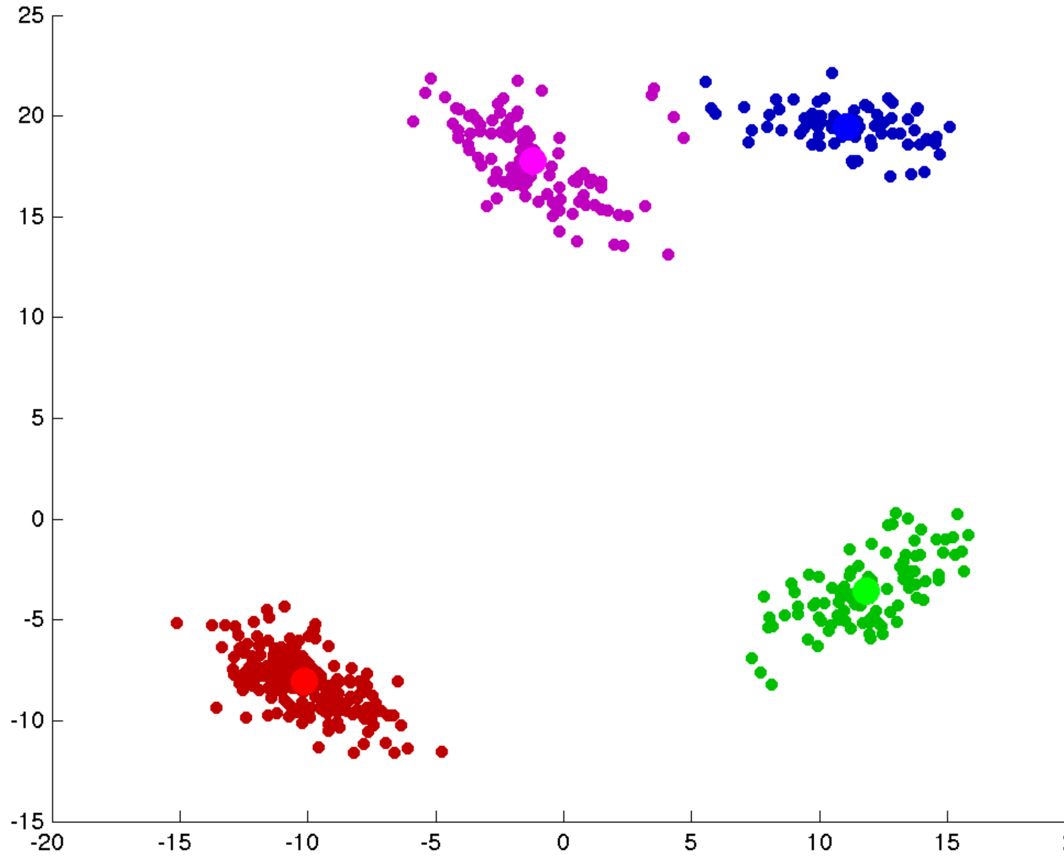
Admin

- Assignment 1 is due Friday.
  - Needs Julie 0.6 (you can use JuliaBox if you can’t get Julia/PyPlot working).
  - There was a bug in the decision tree predict function.
  - There was a minor error in the example_knn.jl function.
  - Follow the assignment guidelines naming convention (a1.zip/a1.pdf).

- Assignment 0 grades posted on Connect?

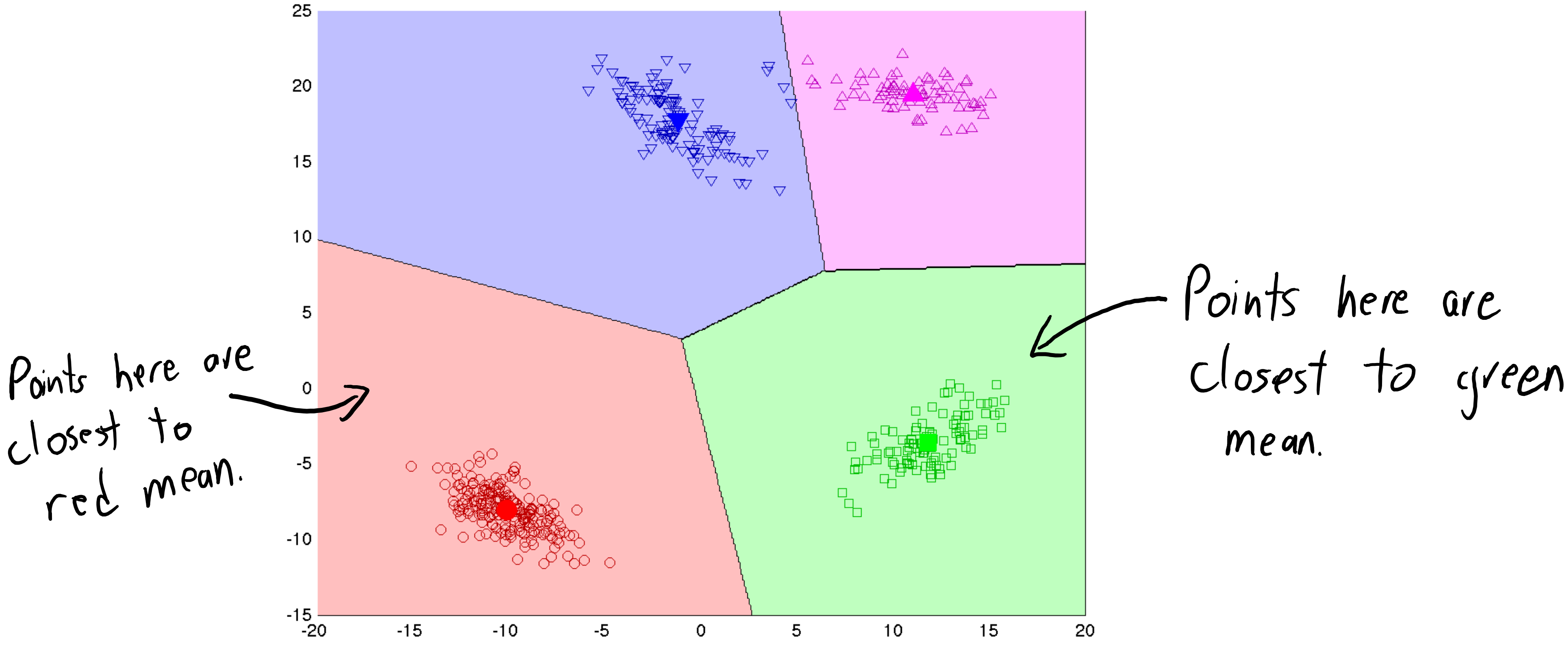
Last Time: K-Means Clustering

- We want to cluster data:
  - Assign objects to groups.
- K-means clustering:
  - Define groups by “means”
  - Assigns objects 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”:

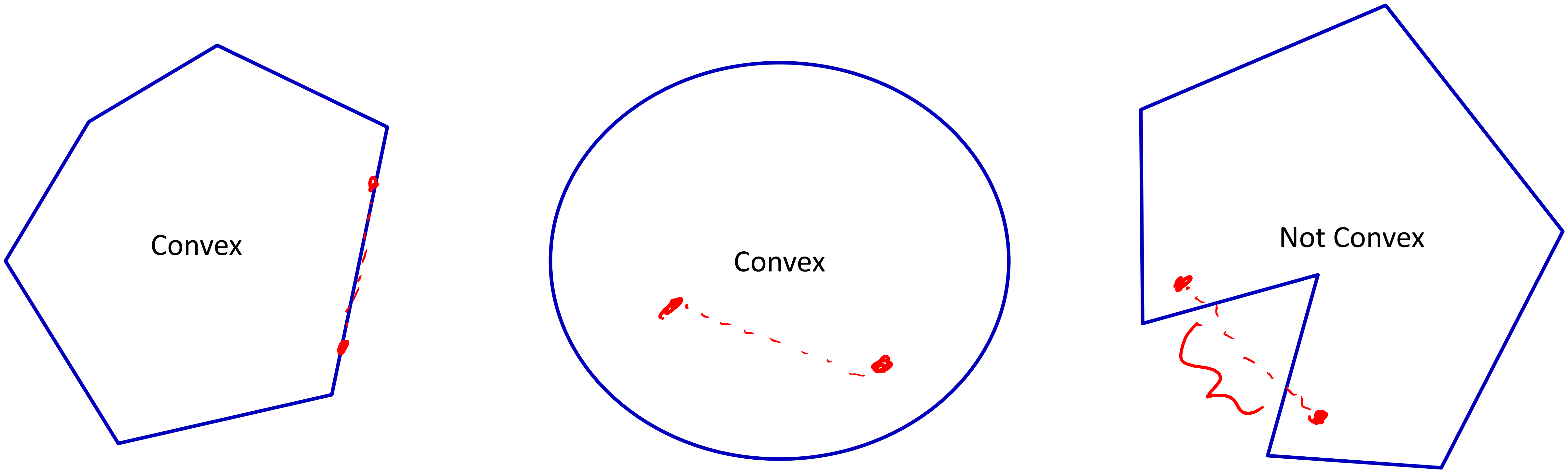


- Observe that the clusters are convex regions.

[Animation]

Convex Sets

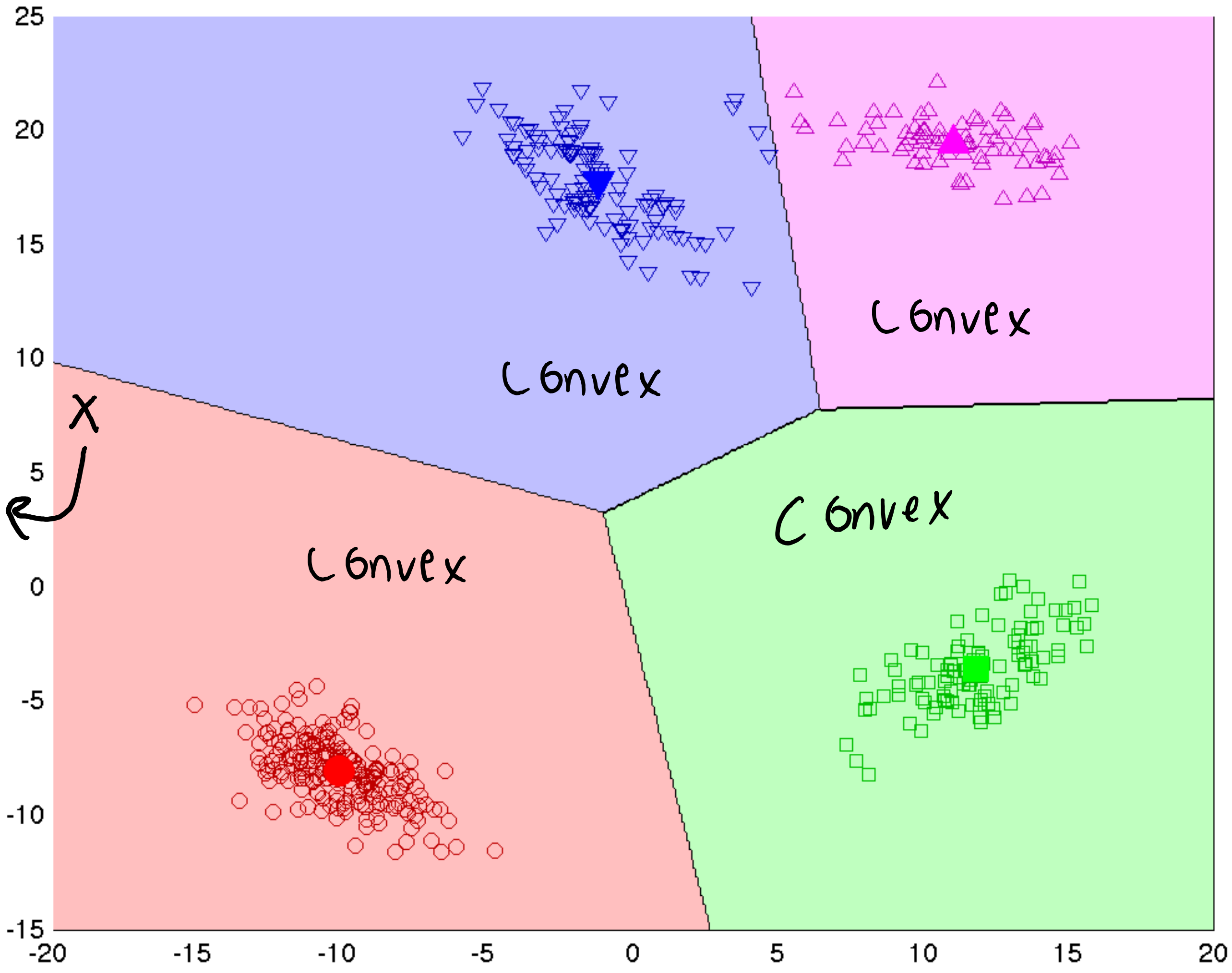
- A set is convex if line between two points in the set stays in the set.



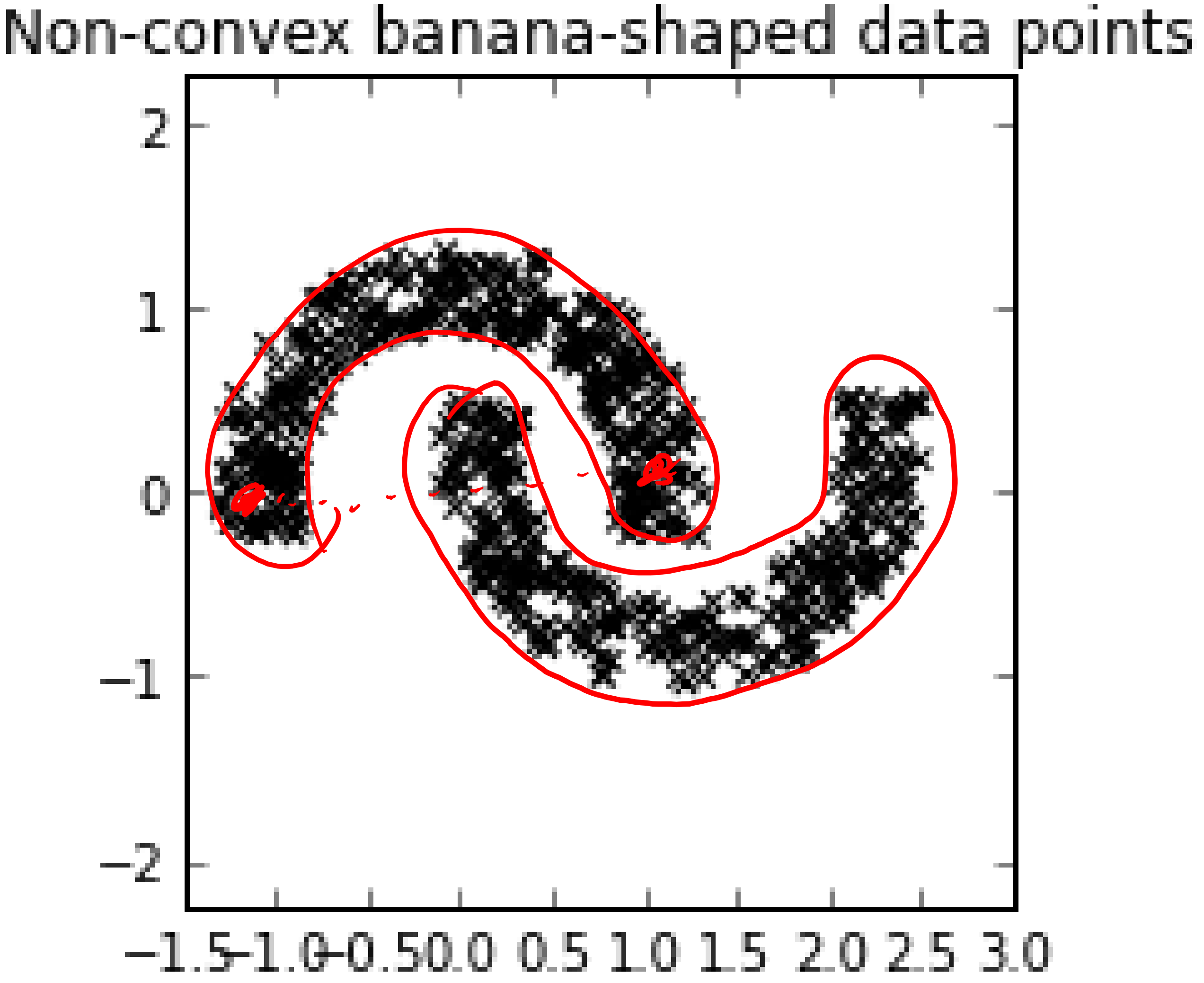
Shape of K-Means Clusters

Issues with shape of K-means clusters:

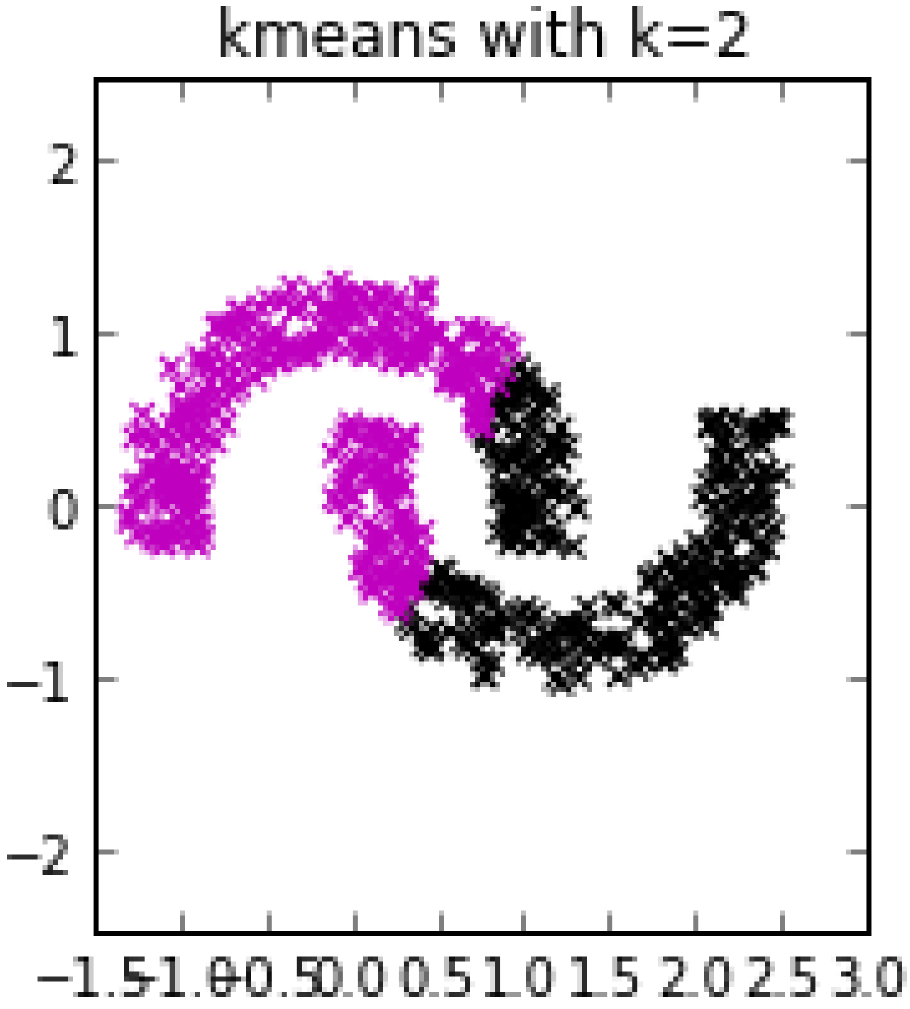
1. Clusters in the data might not be convex.
2. Does this point really belong in red cluster



K-Means with Non-Convex Clusters

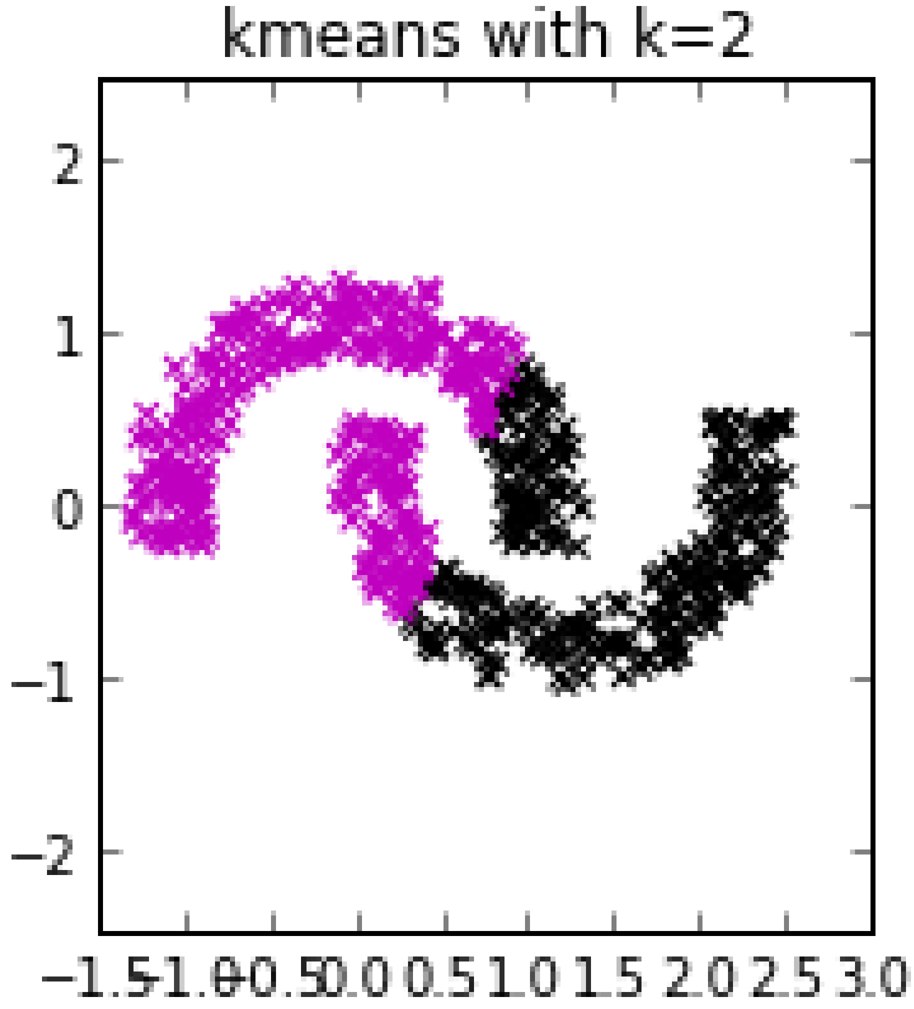


K-Means with Non-Convex Clusters



K-means cannot separate non-convex clusters

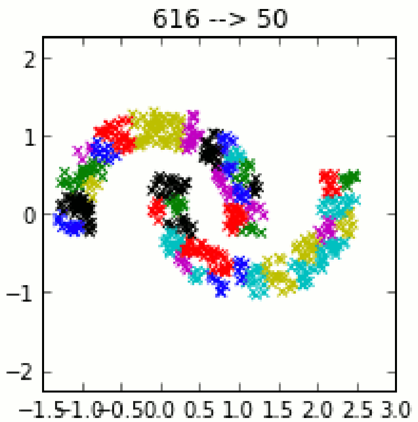
K-Means with Non-Convex Clusters

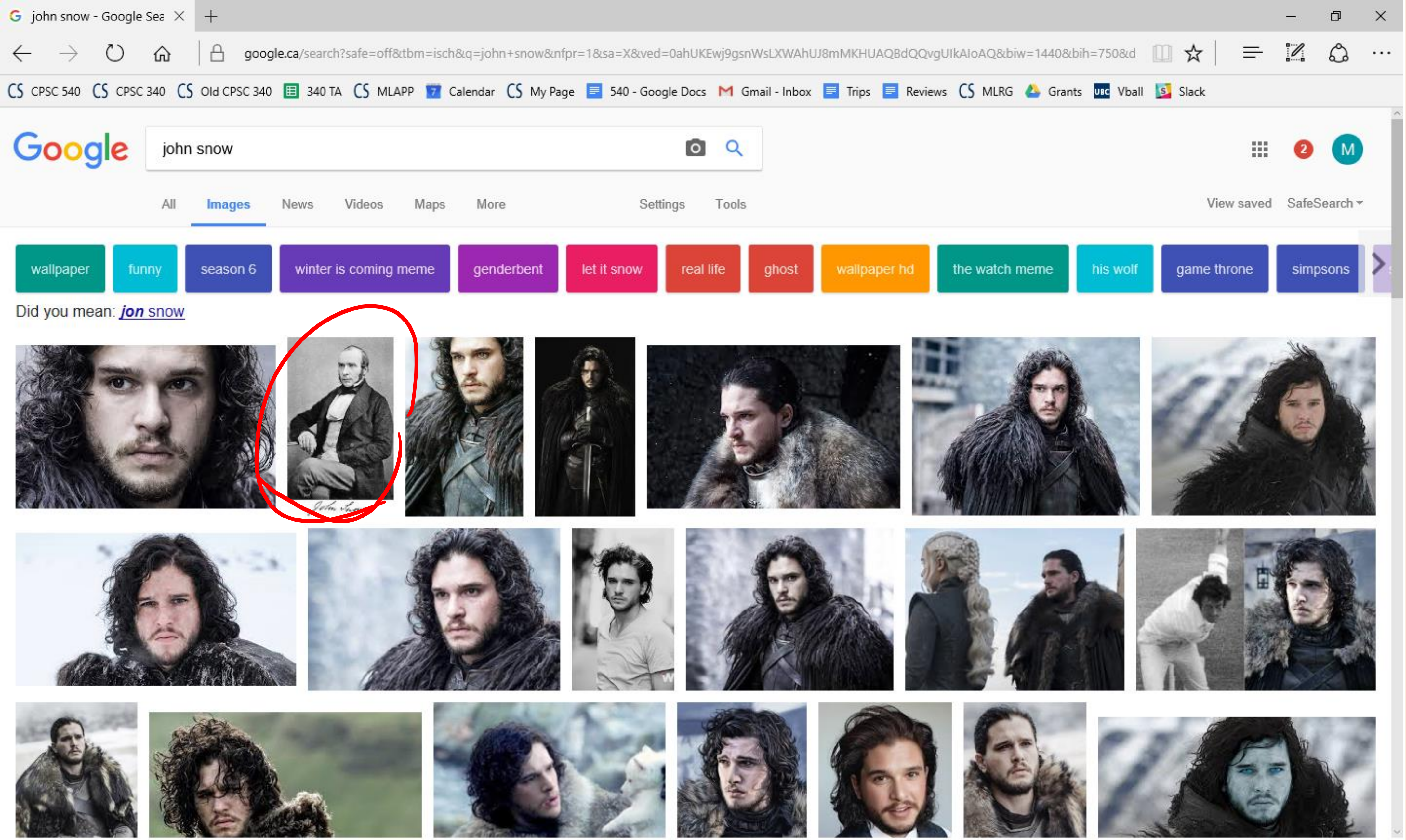


K-means cannot separate non-convex clusters

Though over-clustering can help

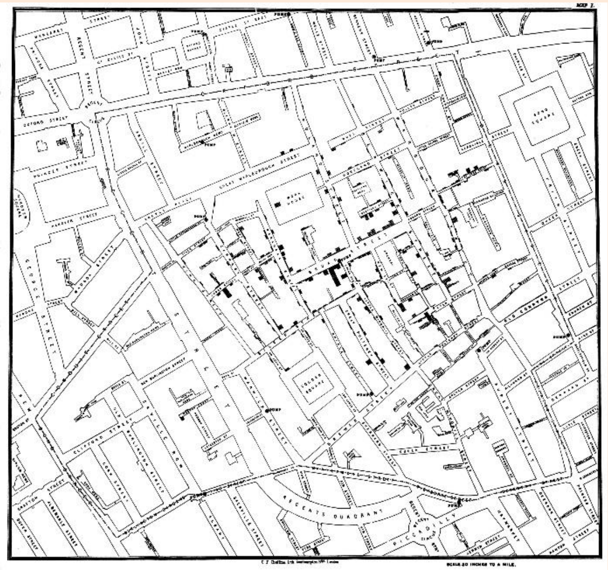
(next class)

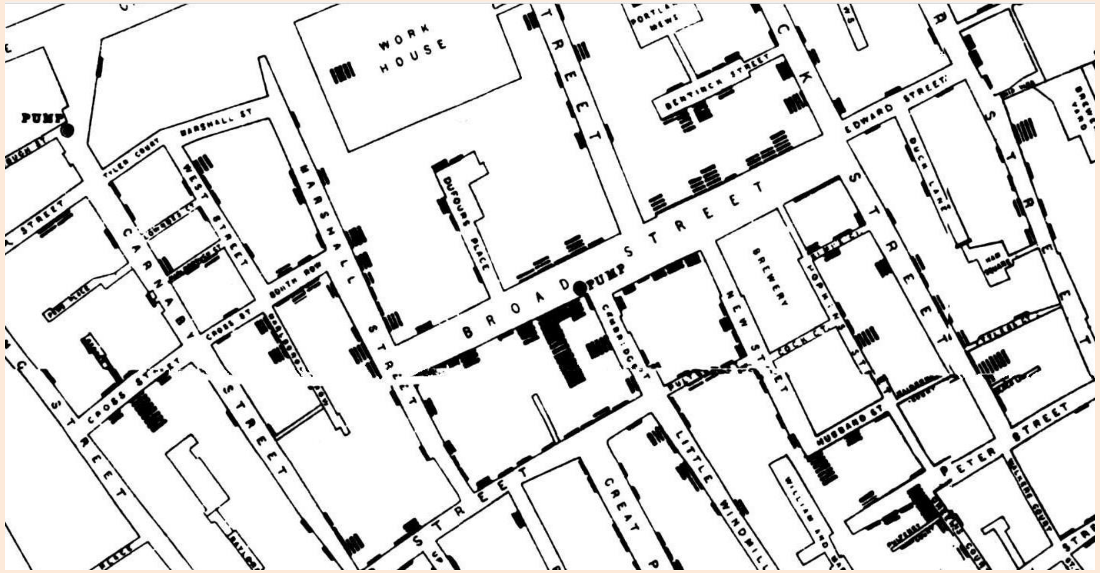



john snow - Google Sea
google.ca/search?safe=off&tbm=isch&q=john+snow&nfpr=1&sa=X&ved=0ahUKEwj9gsnWsLXWAhUJ8mMKHUAQBdQQvgUIkAIoAQ&biw=1440&bih=750&d
CPSC 540
CPSC 340
Old CPSC 340
340 TA
MLAPP
Calendar
My Page
540 - Google Docs
Gmail - Inbox
Trips
Reviews
MLRG
Grants
Vball
Slack
Google
john snow
All
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

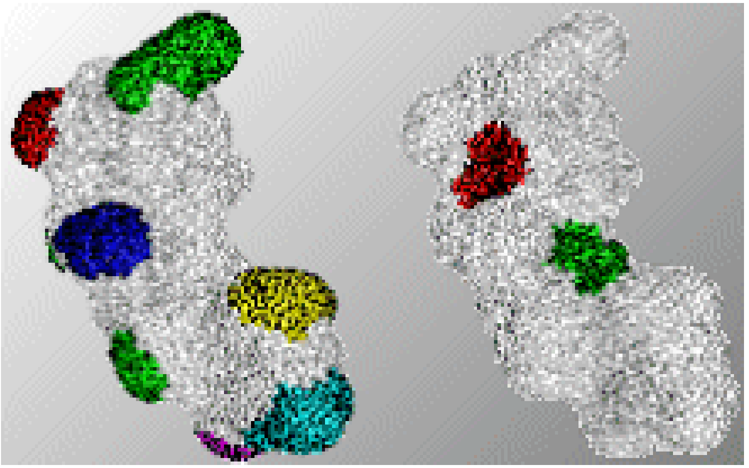
- Density-based clustering:
  - Clusters are defined by "dense" regions.
  - Objects in non-dense regions don't get clustered.
    - Not trying to "partition" the space.

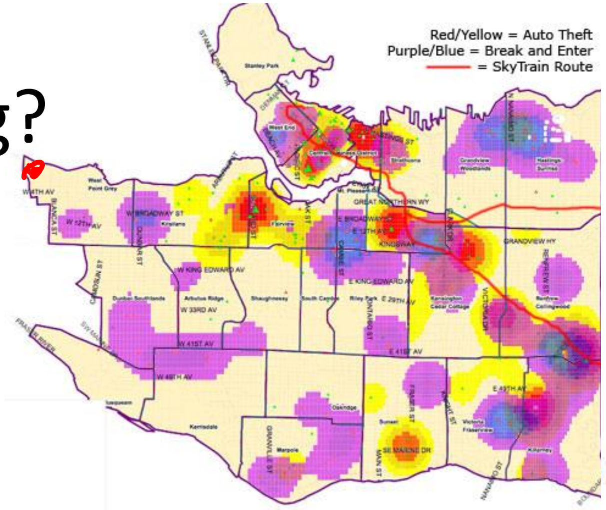
- Clusters can be non-convex:
  - Elephant clusters affected by vegetation, mountains, rivers, water access, etc.
- It's a non-parametric clustering method:
  - No fixed number of clusters 'k'.
  - Clusters can become more complicated with more data.

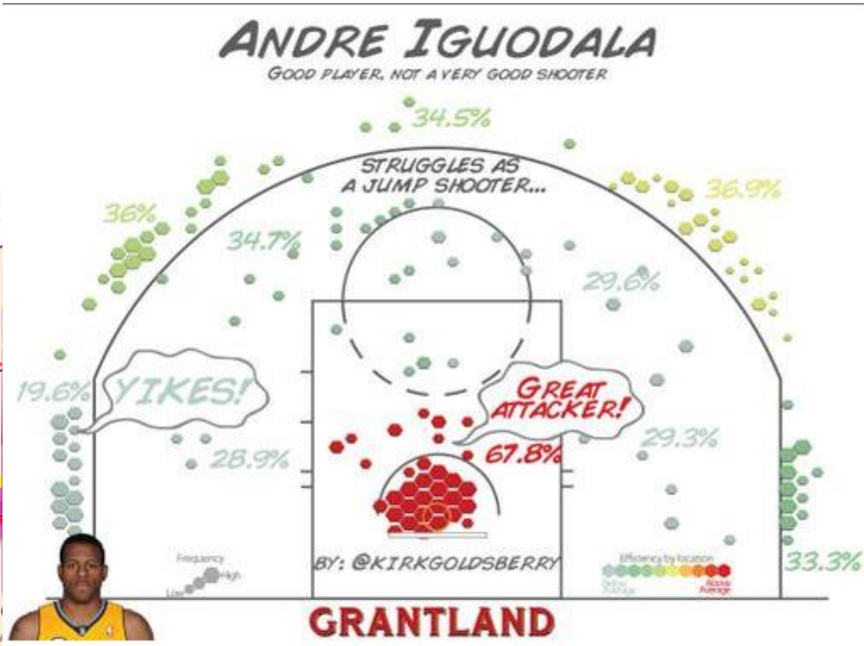
http://www.defenders.org/elephant/basic-facts

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?



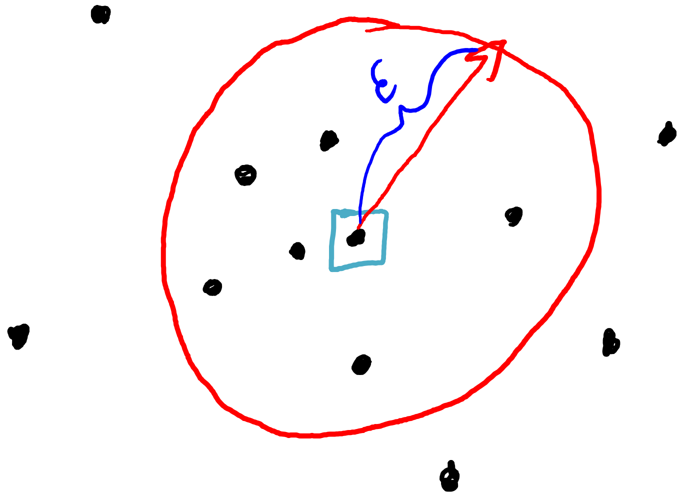




https://en.wikipedia.org/wiki/Cluster_analysis
https://www.flickr.com/photos/dbarefoot/420194128/
http://letsgowarriors.com/replacing-jarrett-jack/2013/10/04/
http://www.dbs.informatik.uni-muenchen.de/Forschung/KDD/Clustering/

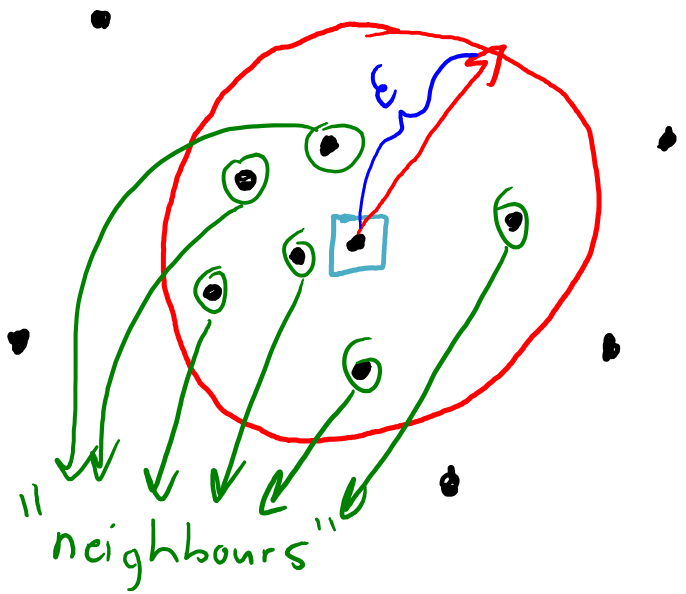
Density-Based Clustering

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



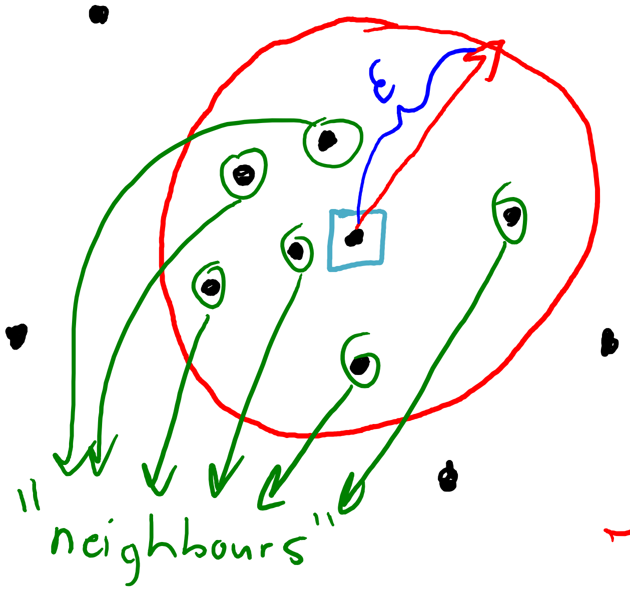
Density-Based Clustering

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



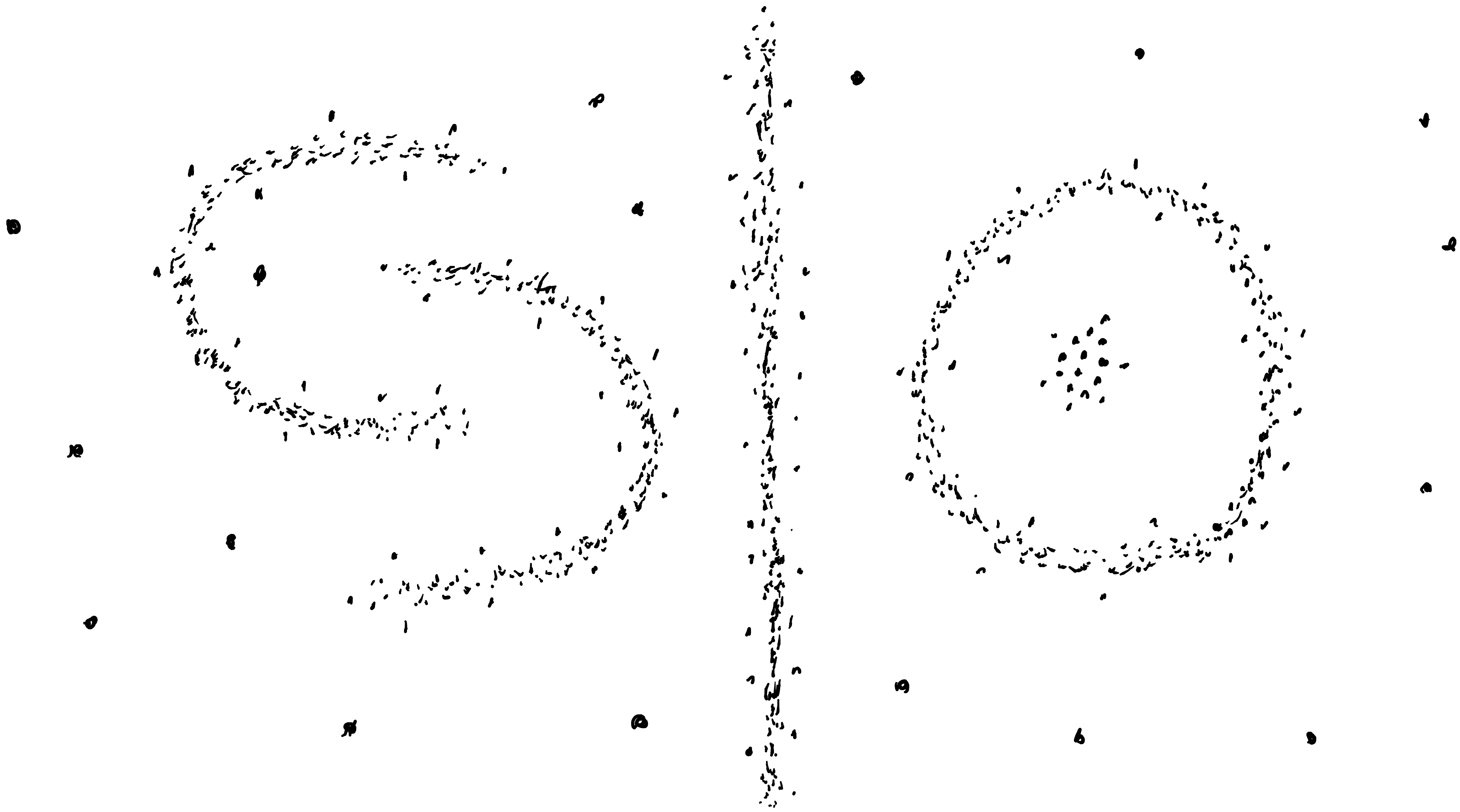
Density-Based Clustering

- 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.

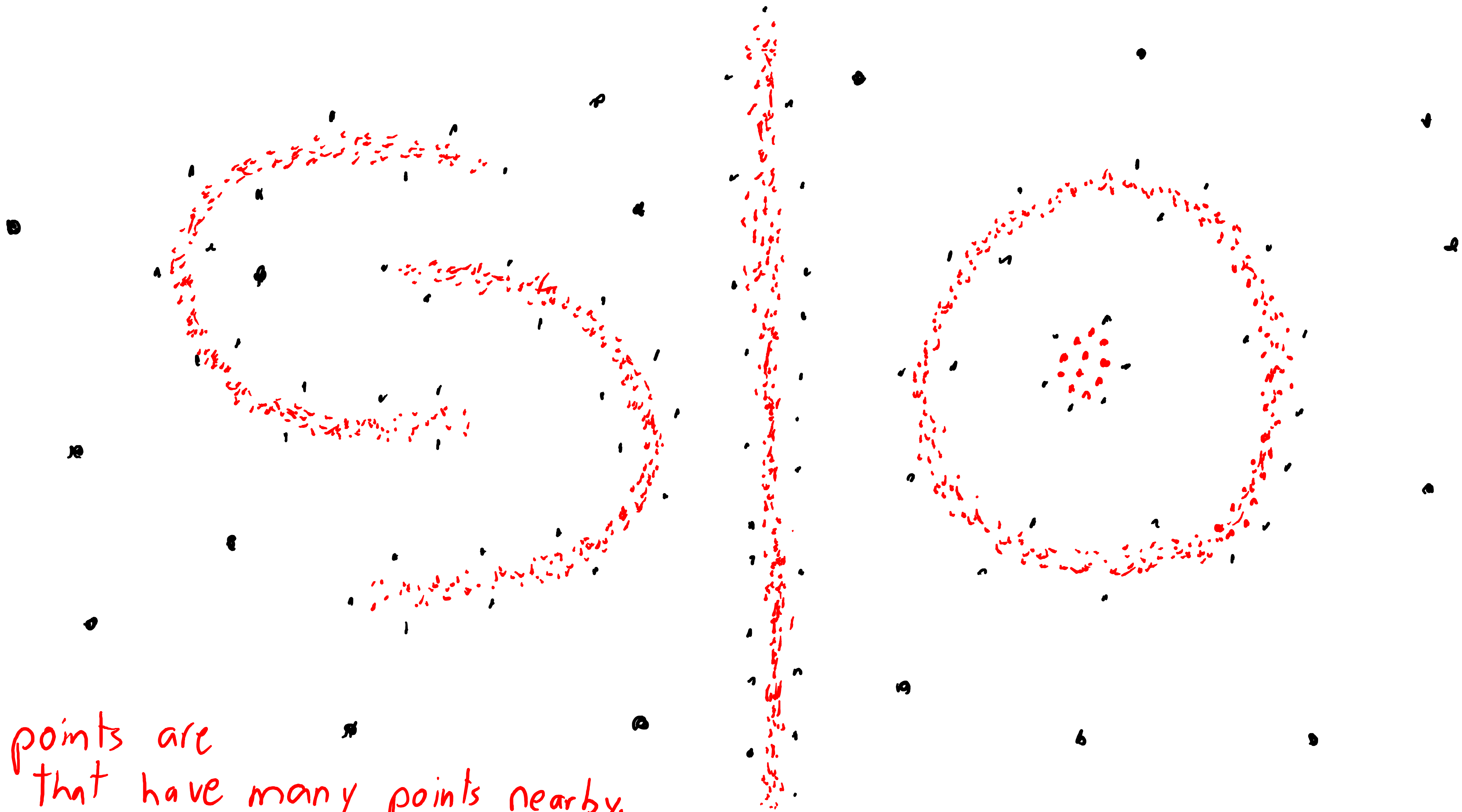


E.g., if minNeighbours = 3 then this is a "core" point since 6 points are "neighbours"

Density-Based Clustering

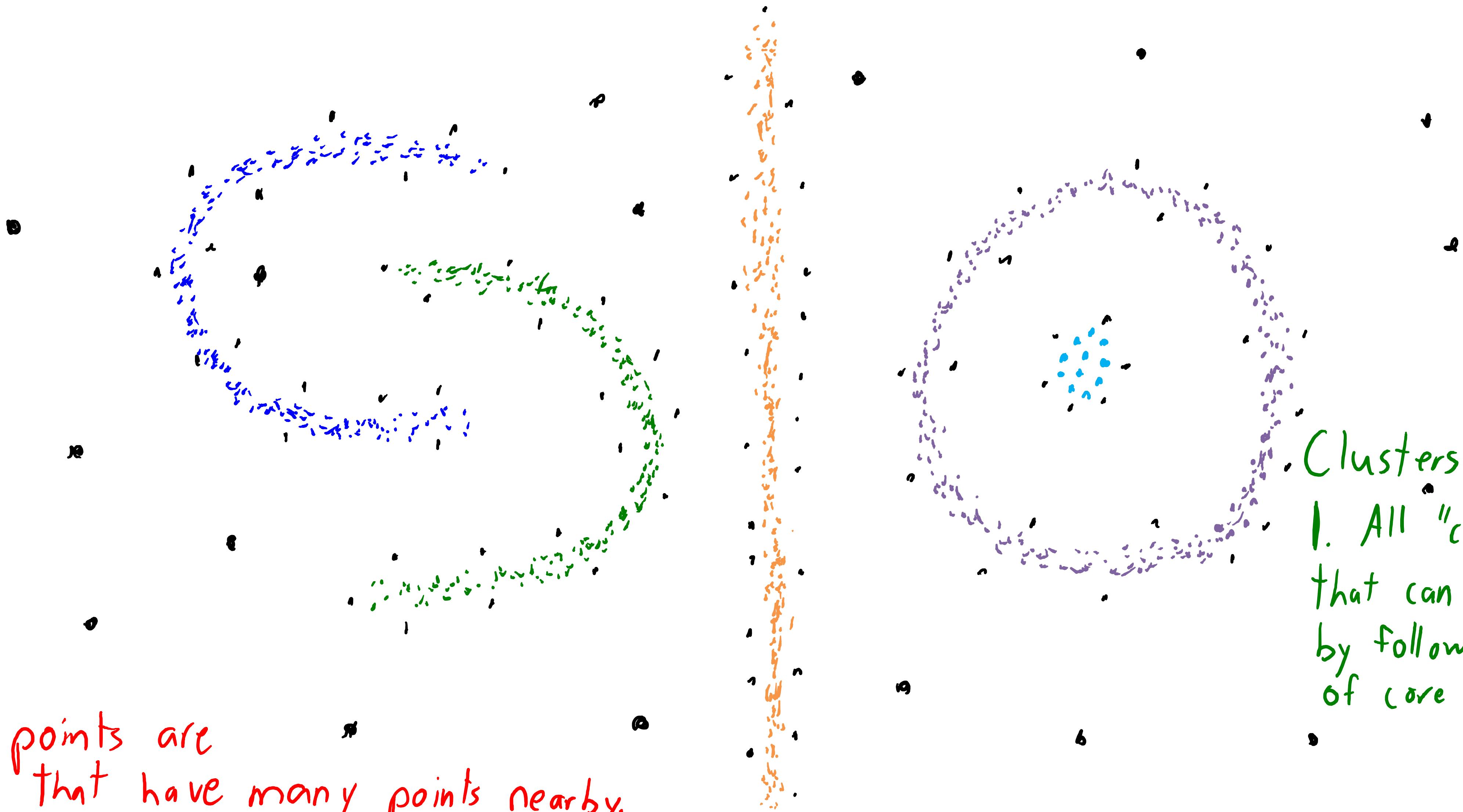


Density-Based Clustering



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

Density-Based Clustering

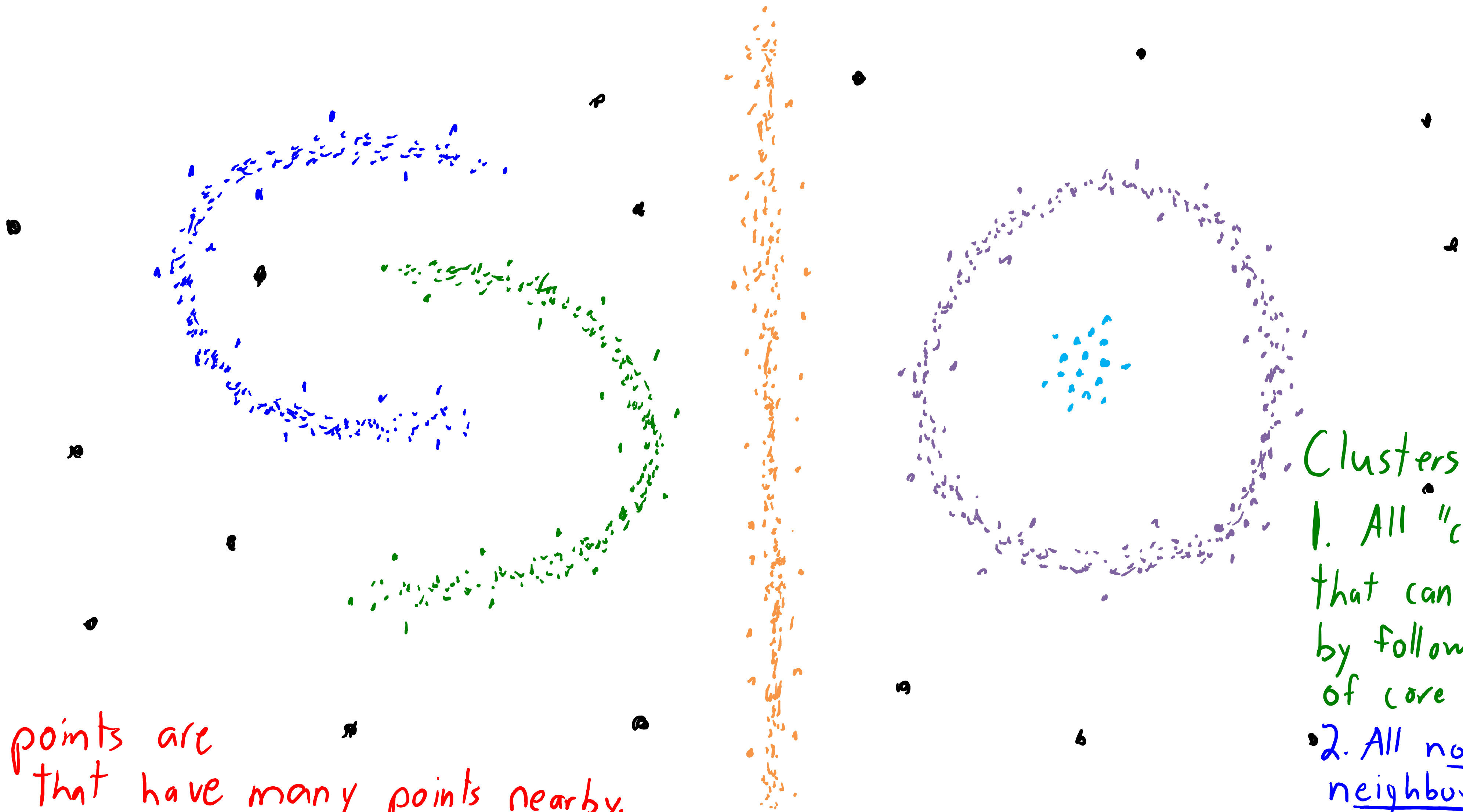


Clusters contain:

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

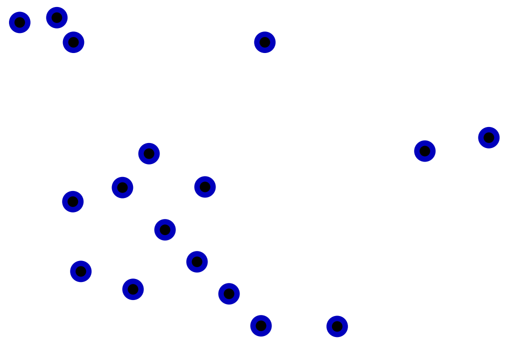


"Core" points are points that have many points nearby.

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)

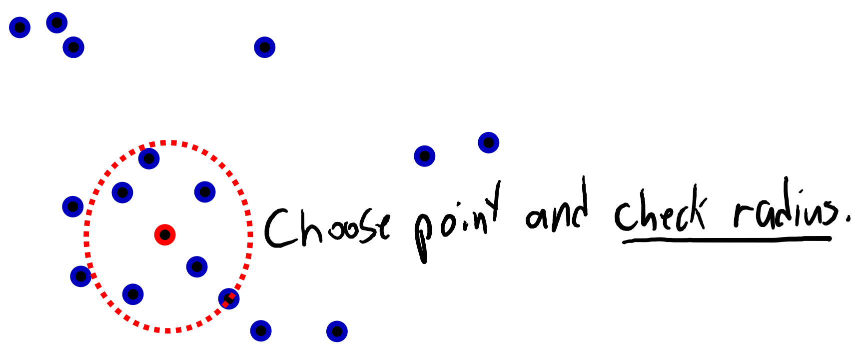
Density-Based Clustering (Example)

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



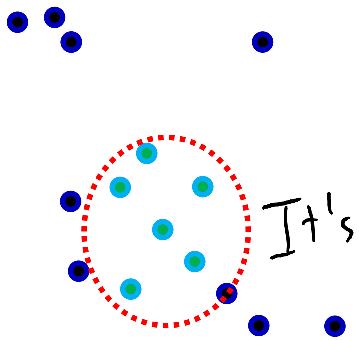
Density-Based Clustering (Example)

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



Density-Based Clustering (Example)

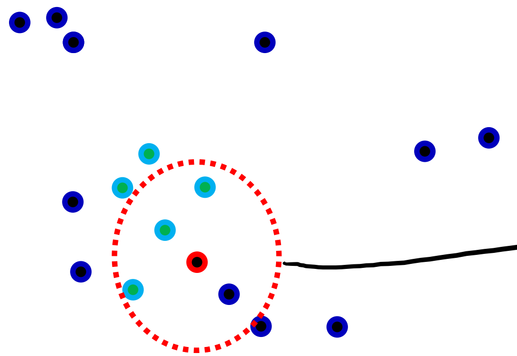
- Each "core" point defines a cluster:
  - Consisting of "core" point and all its "neighbours".
- Merge clusters if "core" points are "neighbours" of each other.



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

Density-Based Clustering (Example)

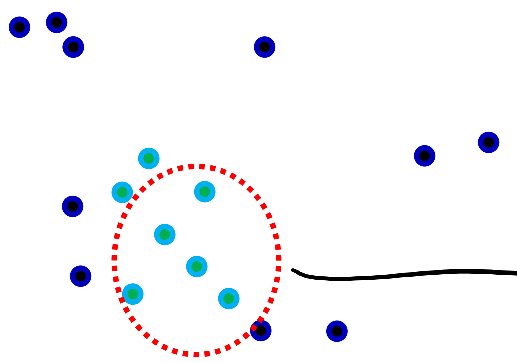
- Each “core” point defines a cluster:
  - Consisting of “core” point and all its “neighbours”.
- Merge clusters if “core” points are “neighbours” of each other.



Now test if a newly-added point is a "core" point.

Density-Based Clustering (Example)

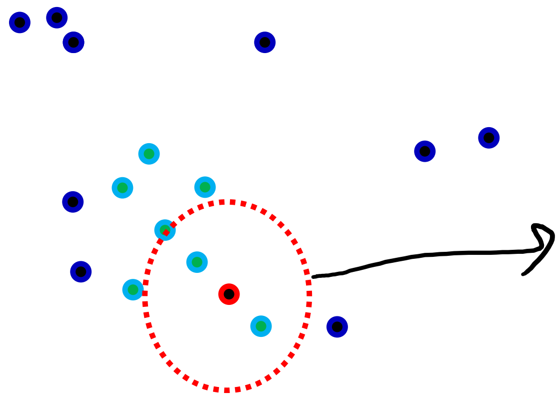
- Each "core" point defines a cluster:
  - Consisting of "core" point and all its "neighbours".
- Merge clusters if "core" points are "neighbours" of each other.



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

Density-Based Clustering (Example)

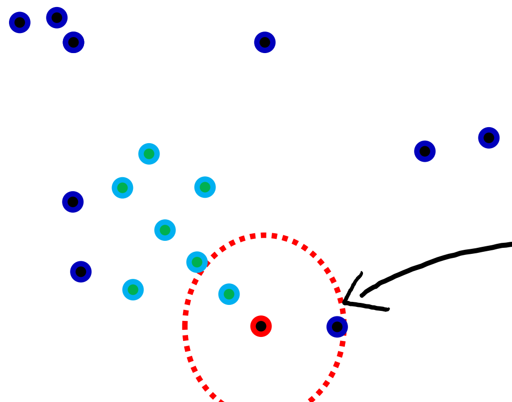
- Each “core” point defines a cluster:
  - Consisting of “core” point and all its “neighbours”.
- Merge clusters if “core” points are “neighbours” of each other.



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

Density-Based Clustering (Example)

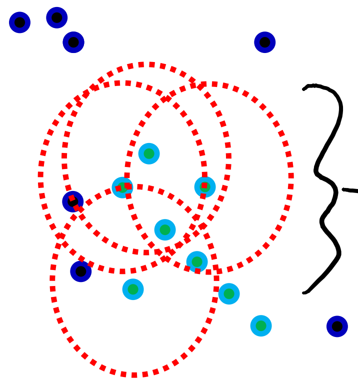
- Each “core” point defines a cluster:
  - Consisting of “core” point and all its “neighbours”.
- Merge clusters if “core” points are “neighbours” of each other.



Not a "core" point, so <u>do not add</u> its neighbour.

Density-Based Clustering (Example)

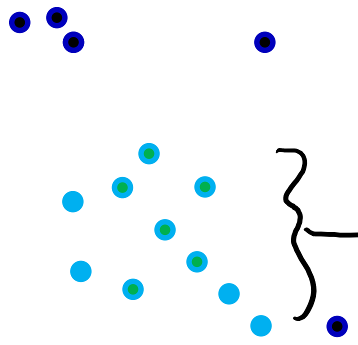
- Each "core" point defines a cluster:
  - Consisting of "core" point and all its "neighbours".
- Merge clusters if "core" points are "neighbours" of each other.



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

Density-Based Clustering (Example)

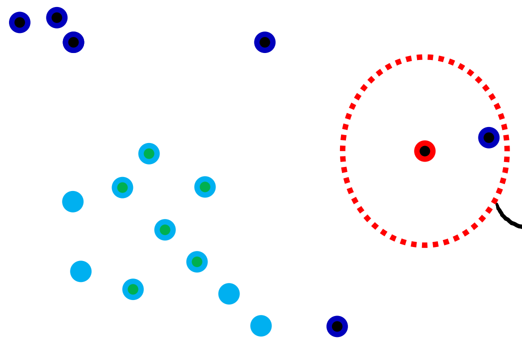
- Each "core" point defines a cluster:
  - Consisting of "core" point and all its "neighbours".
- Merge clusters if "core" points are "neighbours" of each other.



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

Density-Based Clustering (Example)

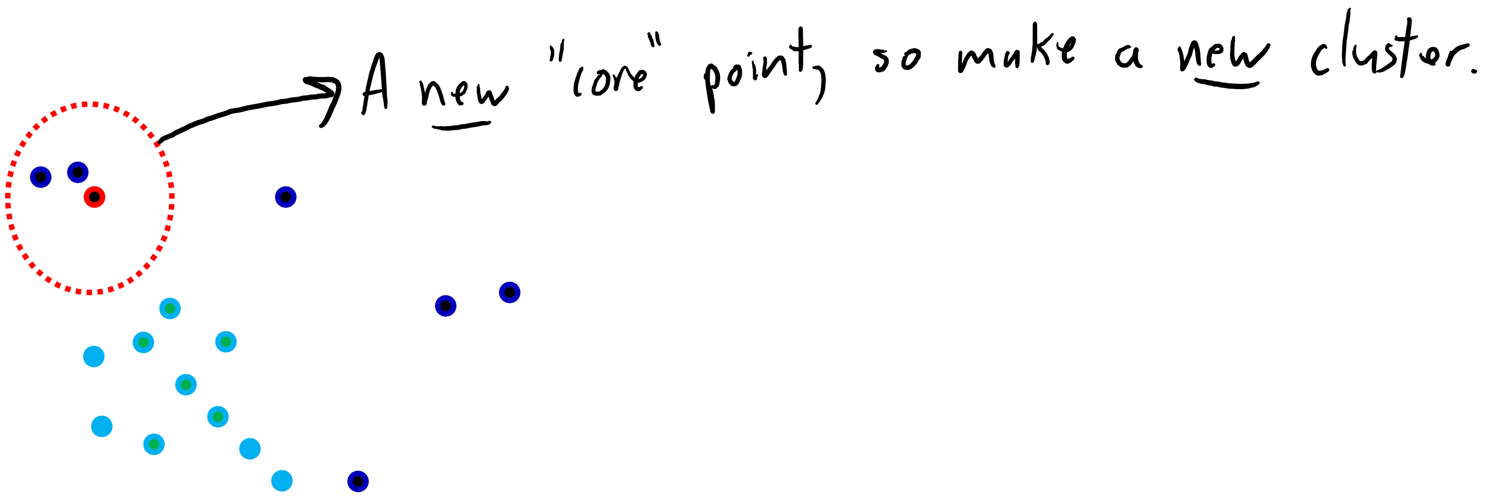
- Each "core" point defines a cluster:
  - Consisting of "core" point and all its "neighbours".
- Merge clusters if "core" points are "neighbours" of each other.



Not enough points in radius => not a "core" point.
Don't assign cluster

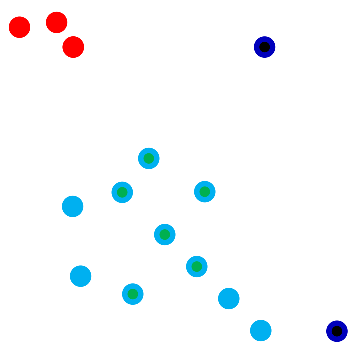
Density-Based Clustering (Example)

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



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

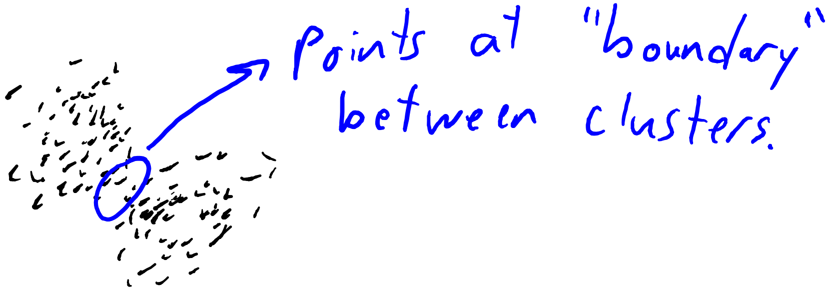
- For each example x_i:
  - If x_i is already assigned to a cluster, do nothing.
  - Test whether x_i is a 'core' point (≥ minNeighbours examples within 'ε').
    - If x_i is not core point, do nothing (this could be an outlier).
    - If x_i is a core point, "expand" cluster.
- "Expand" cluster function:
  - Assign all x_j within distance 'ε' of core point x_i to cluster.
  - For each newly-assigned neighbour x_j that is a core point, "expand" cluster.

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.
  - 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

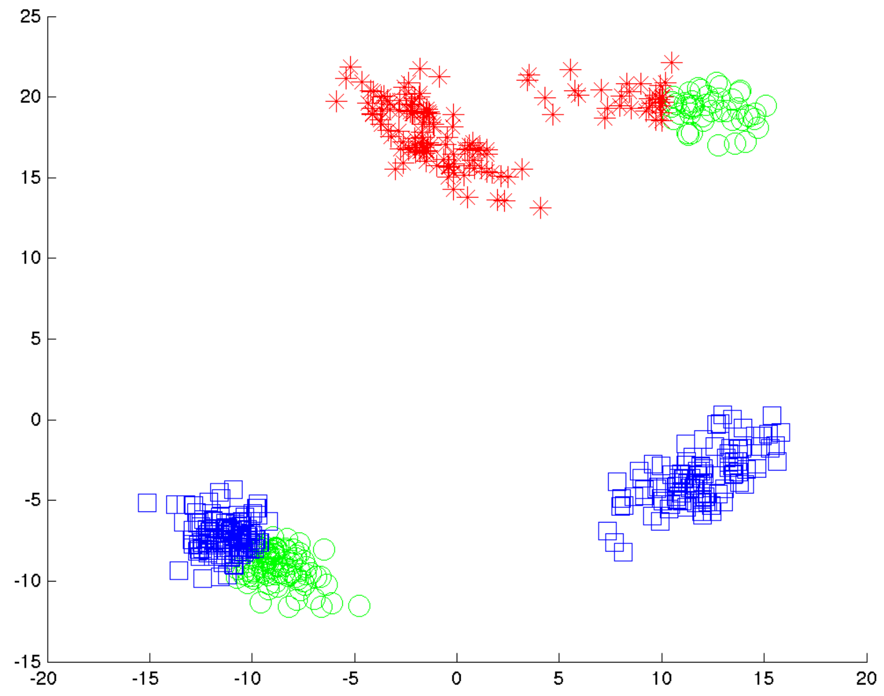
Multiple random runs of K means

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

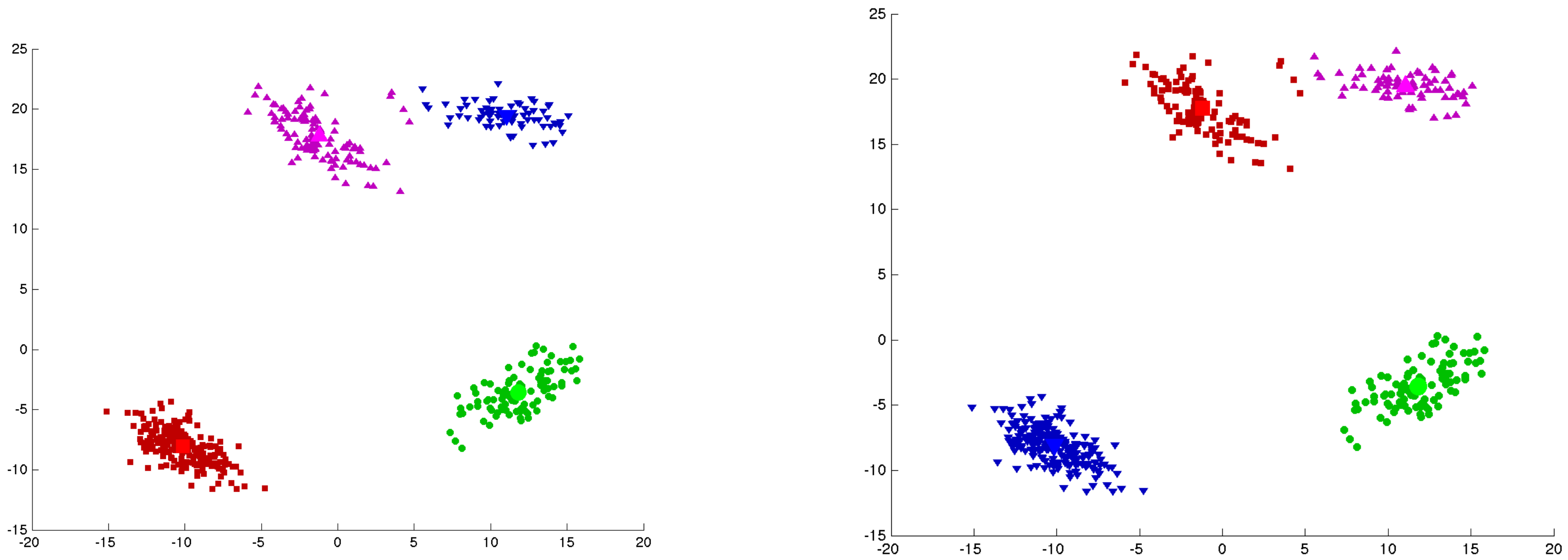
- E.g., run k-means 20 times and then cluster using the mode of each $\hat{y}_i$.
- Normally, averaging across models doing different things is good.



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

Label Switching Problem

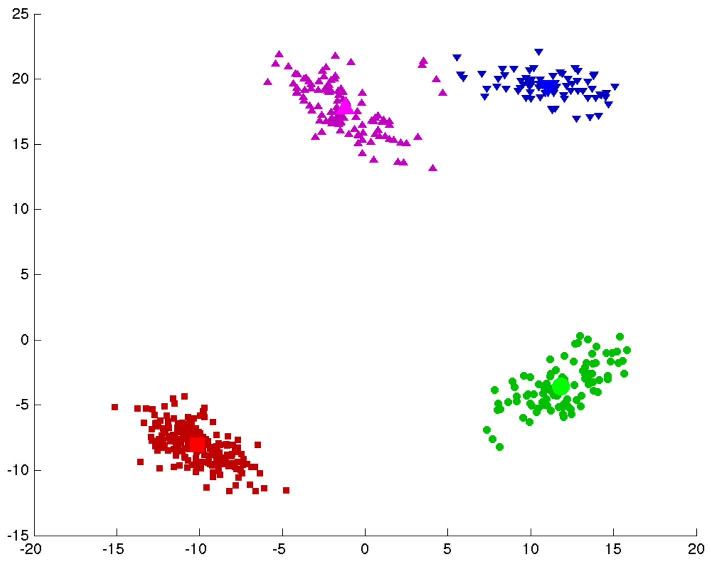
- This doesn't work because of "label switching" problem:
  - The cluster labels $\hat{y}_i$ are meaningless.
  - We could get same clustering with permuted labels:

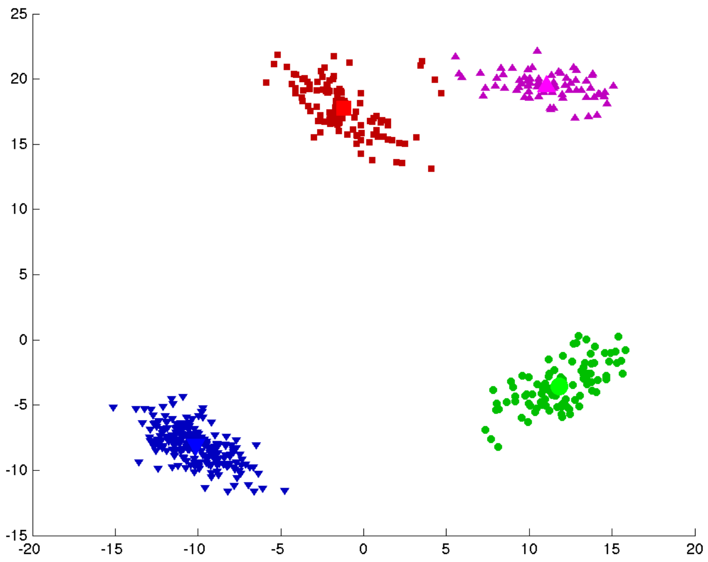


  - All $\hat{y}_i$ become equally likely as number of initializations increases.

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.





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

Summary

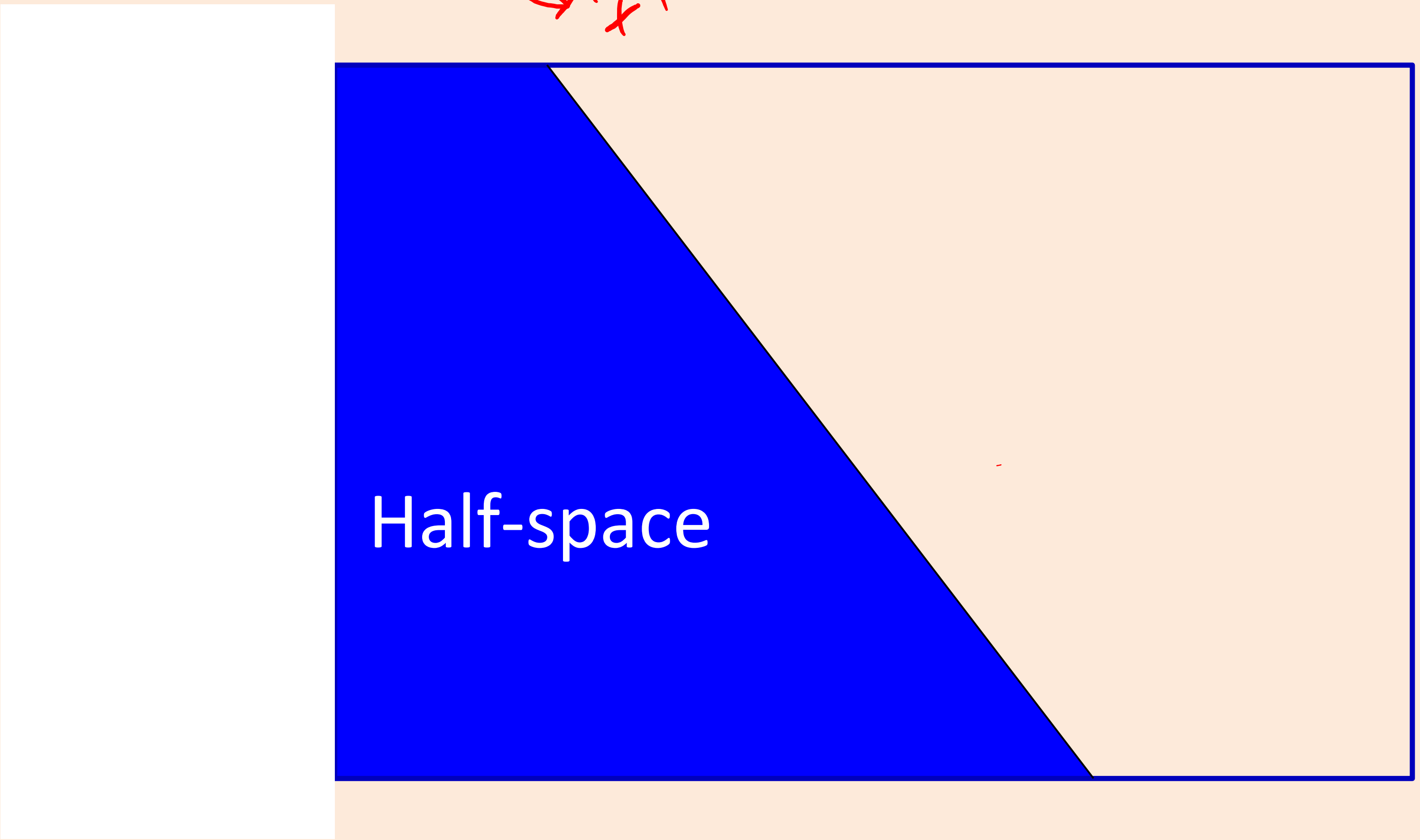
- Shape of K-means clusters:
  - Partitions space into convex sets.
- Density-based clustering:
  - "Expand" and "merge" dense regions of points to find clusters.
  - Not sensitive to initialization or outliers.
  - Useful for finding non-convex connected clusters.
- Ensemble clustering: combines multiple clusterings.
  - Can work well but need to account for label switching.

- Next time:
  - Discovering the tree of life.

Why are k-means clusters convex?

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

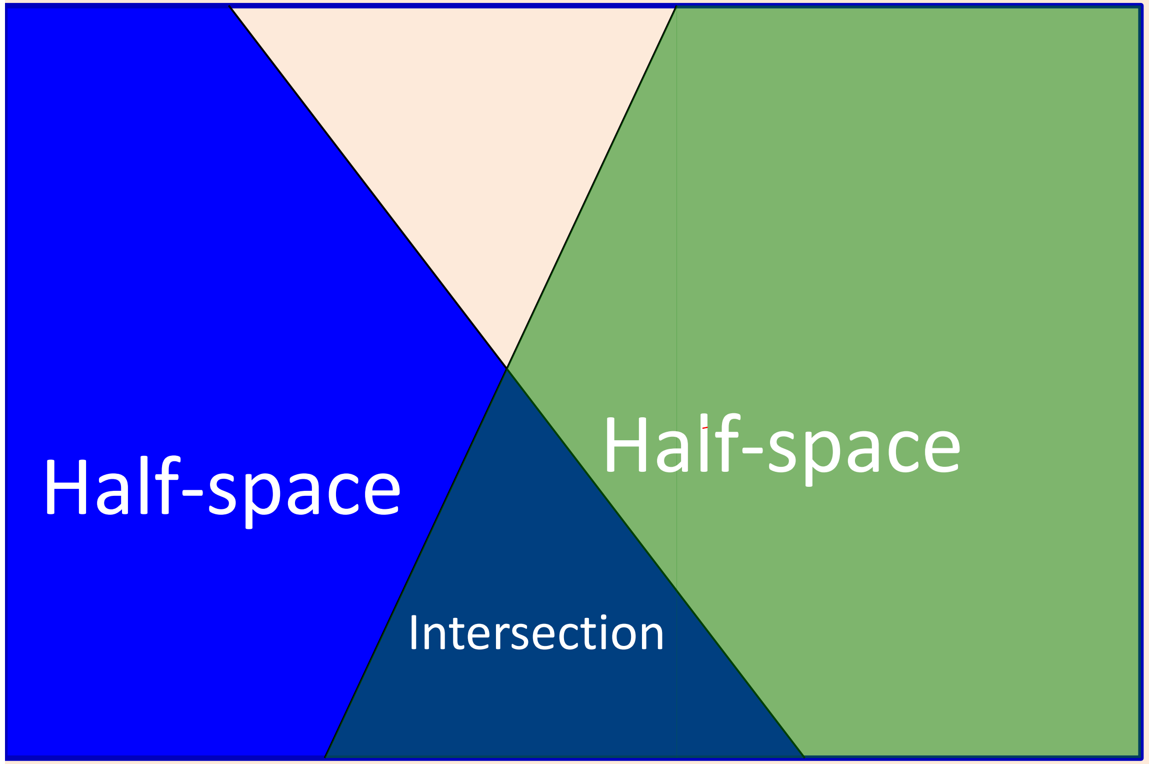
Half-space is Set of points 'x' satifying a linear inequality, like $\sum_{j=1}^{d} a_j x_j \leq b$



Why are k-means clusters convex?

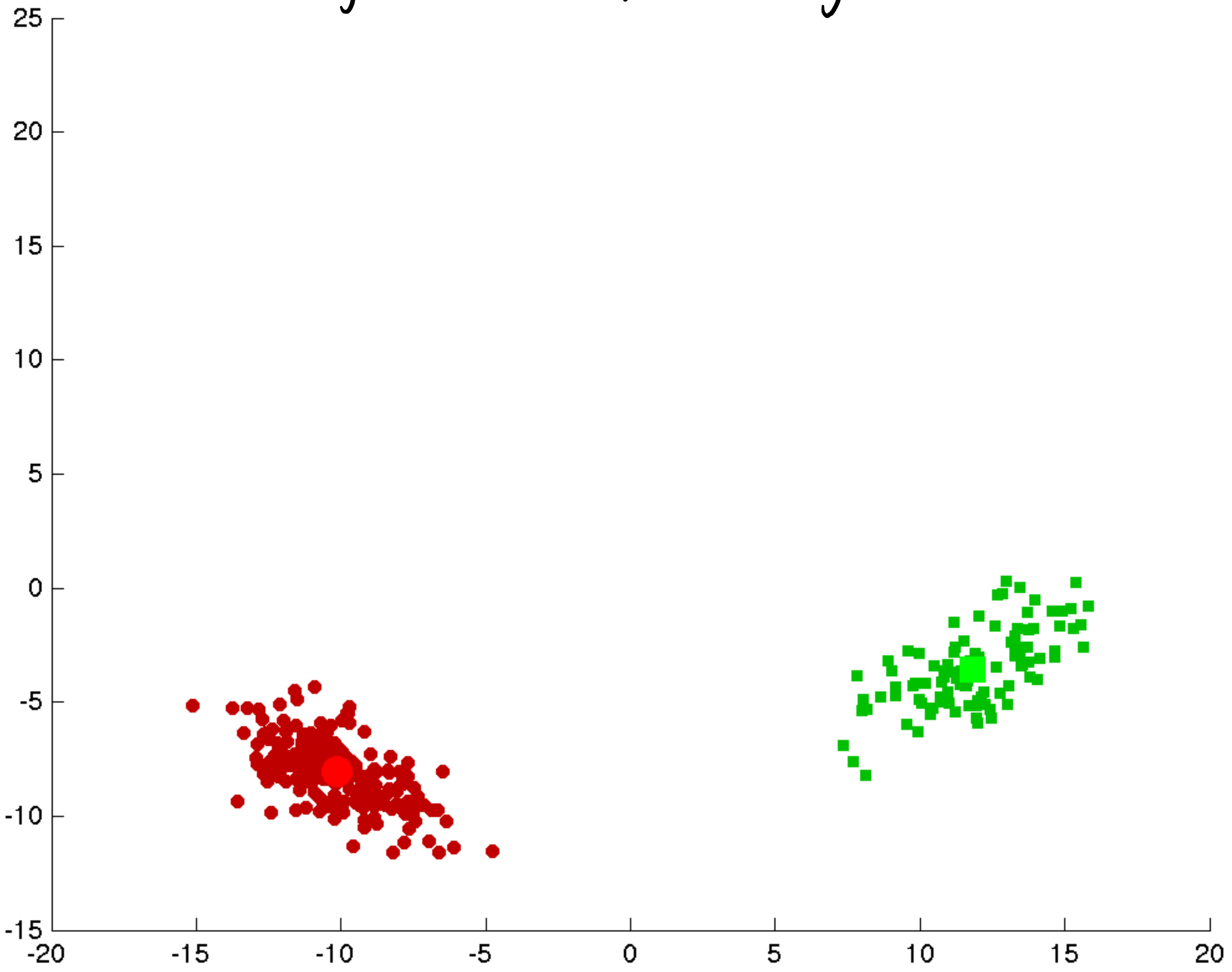
- K-means clusters are formed by the intersection of half-spaces.

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



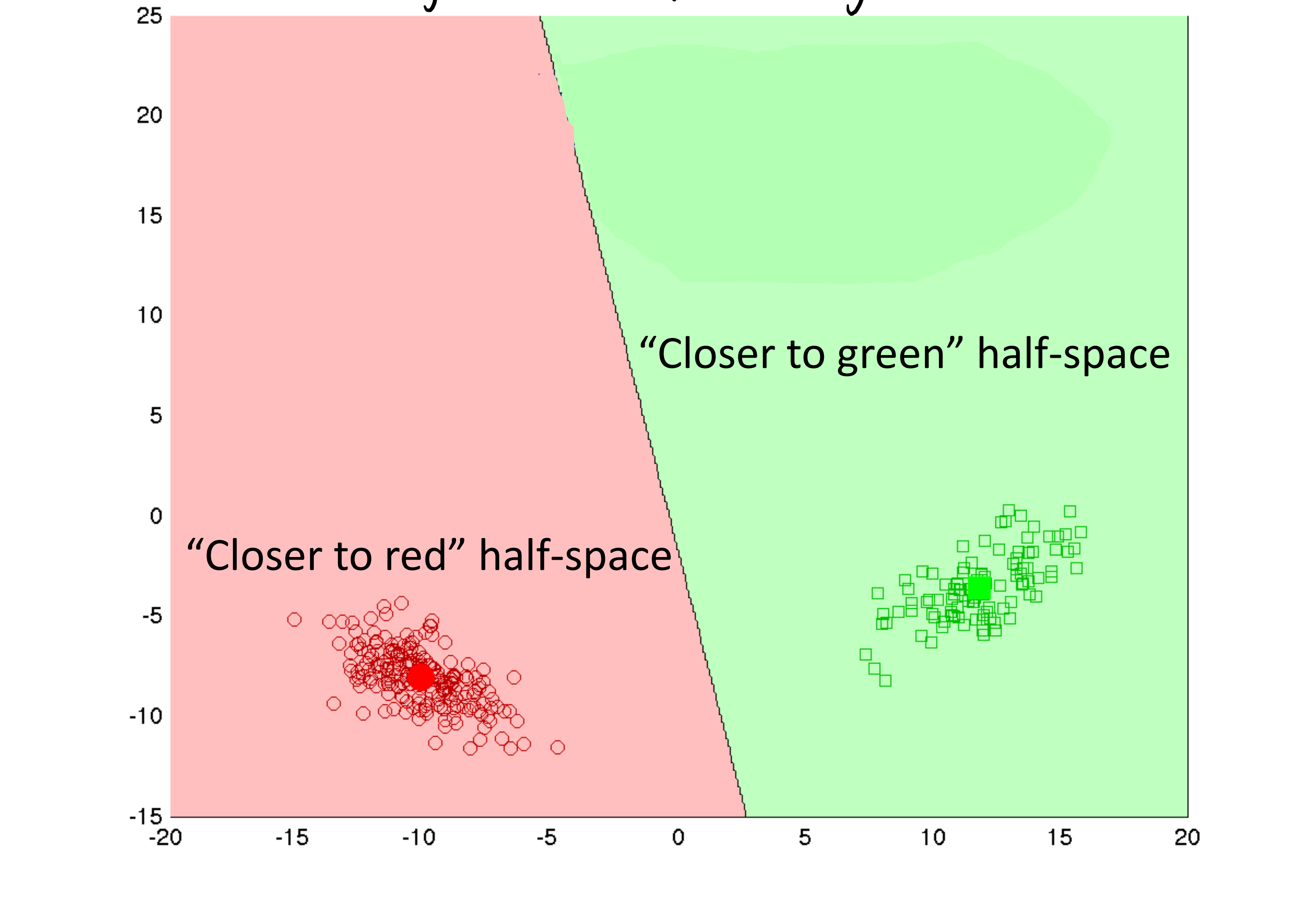
Why are k-means clusters convex?

Which regions are put in green cluster?



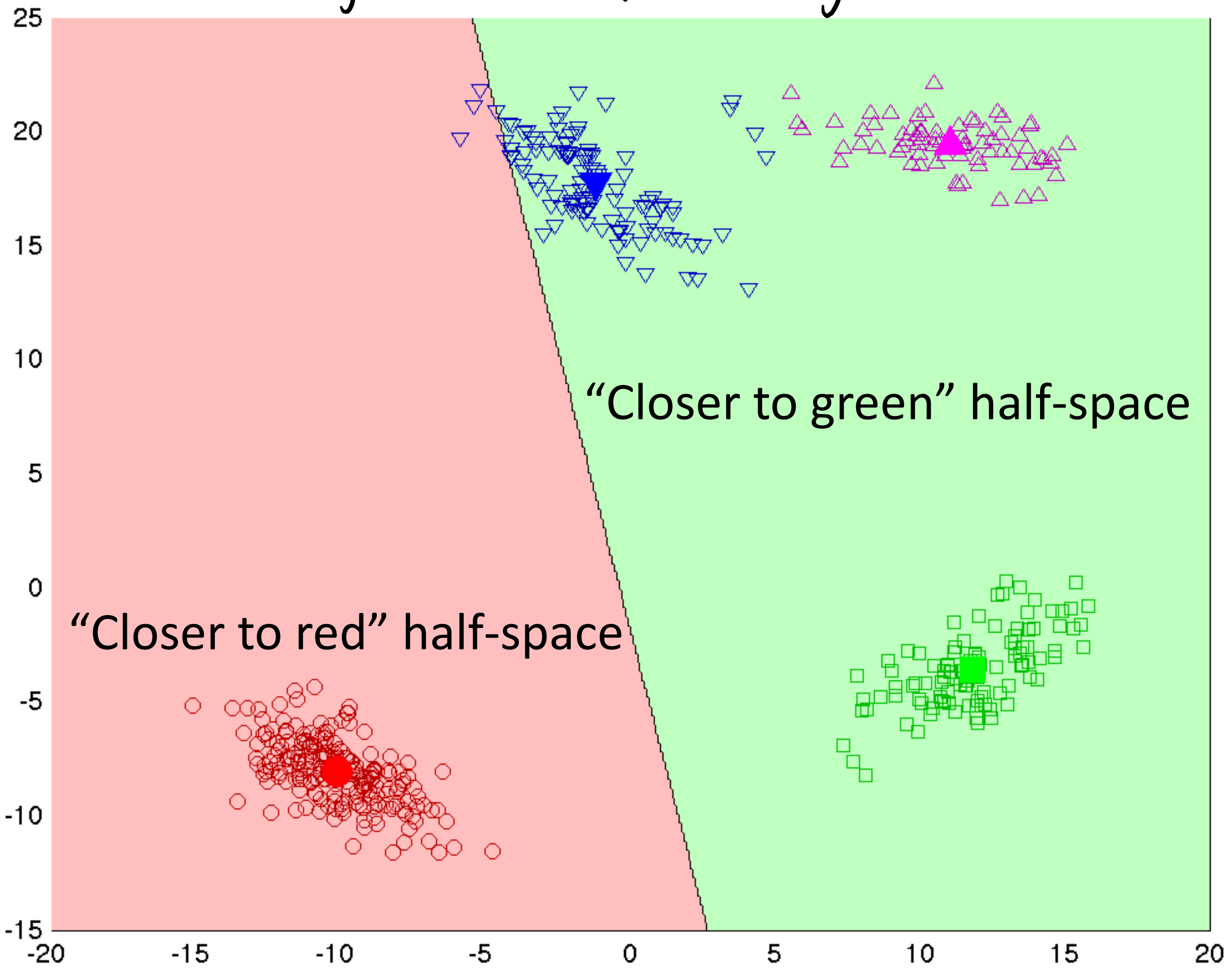
Why are k-means clusters convex?

Which regions are put in green cluster?



Why are k-means clusters convex?

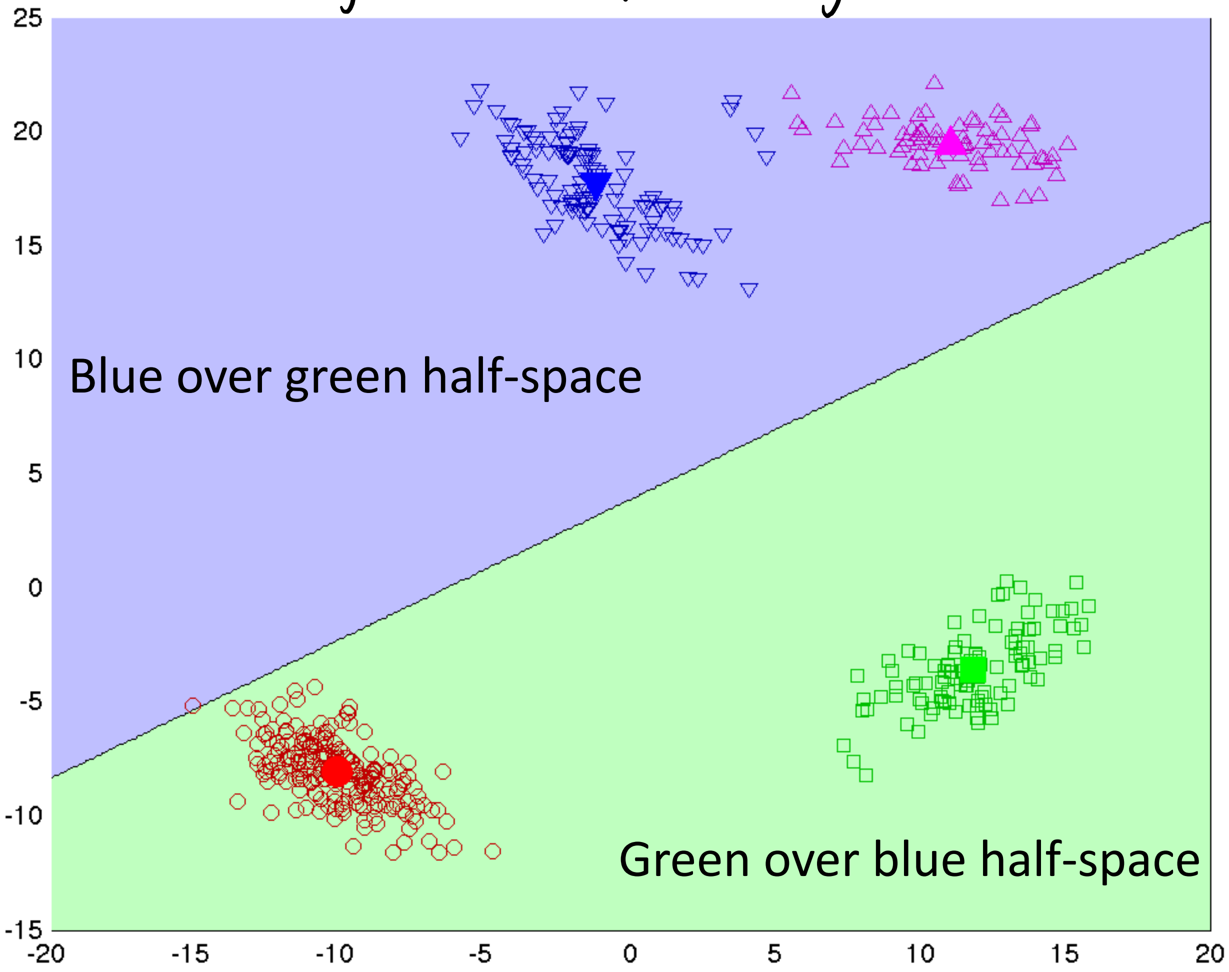
Which regions are put in green cluster?



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

Why are k-means clusters convex?

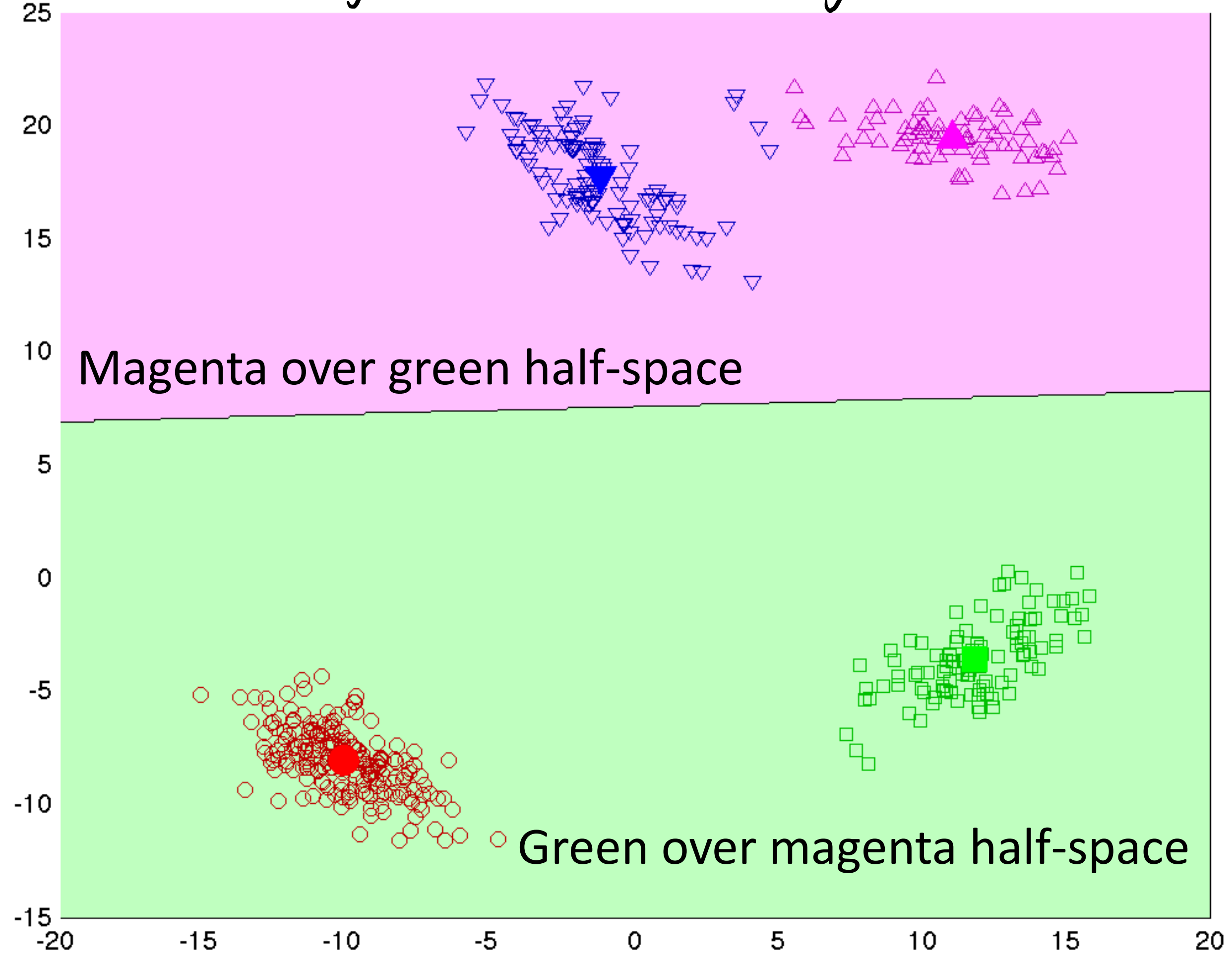
Which regions are put in green cluster?



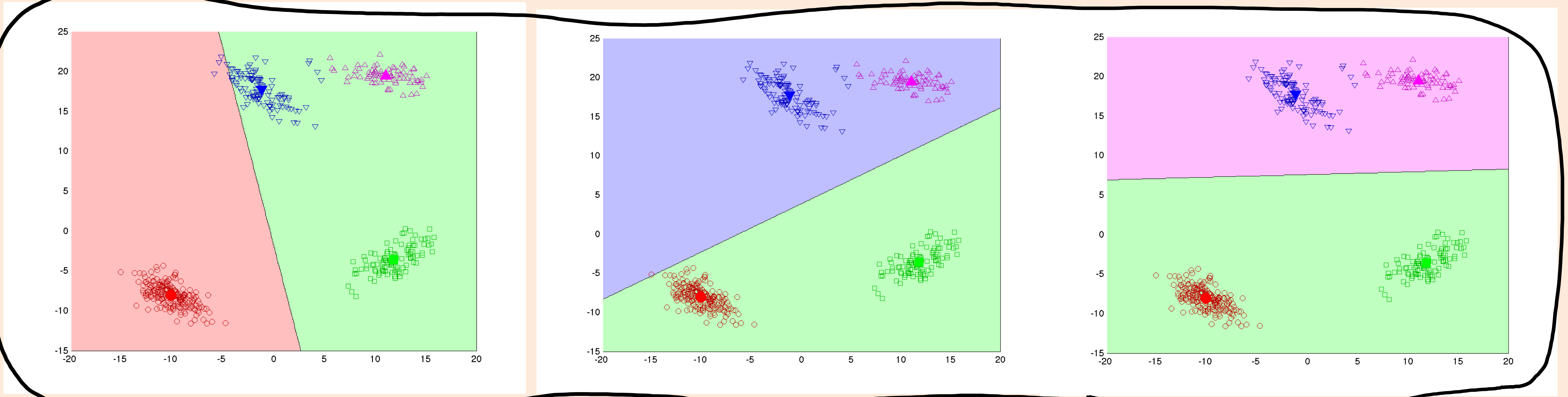
Blue vs. green decision defines different half-spaces.

Why are k-means clusters convex?

Which regions are put in green cluster?

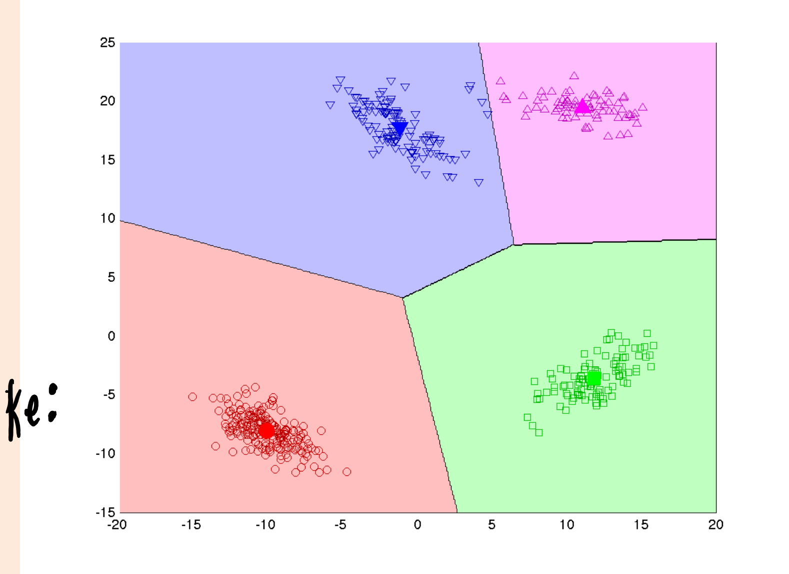


Why are k-means clusters convex?



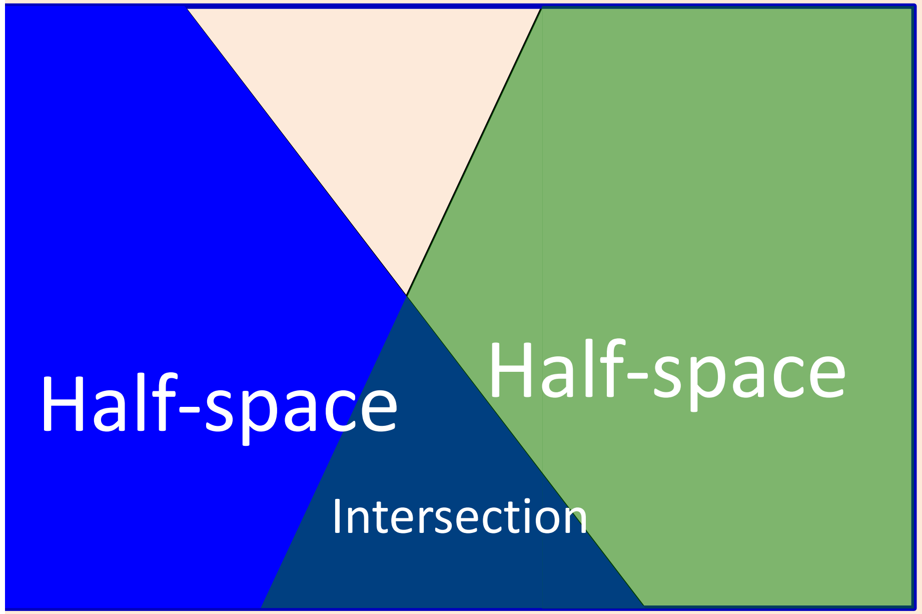
Green "cluster" is the intersection of these three half-spaces.

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

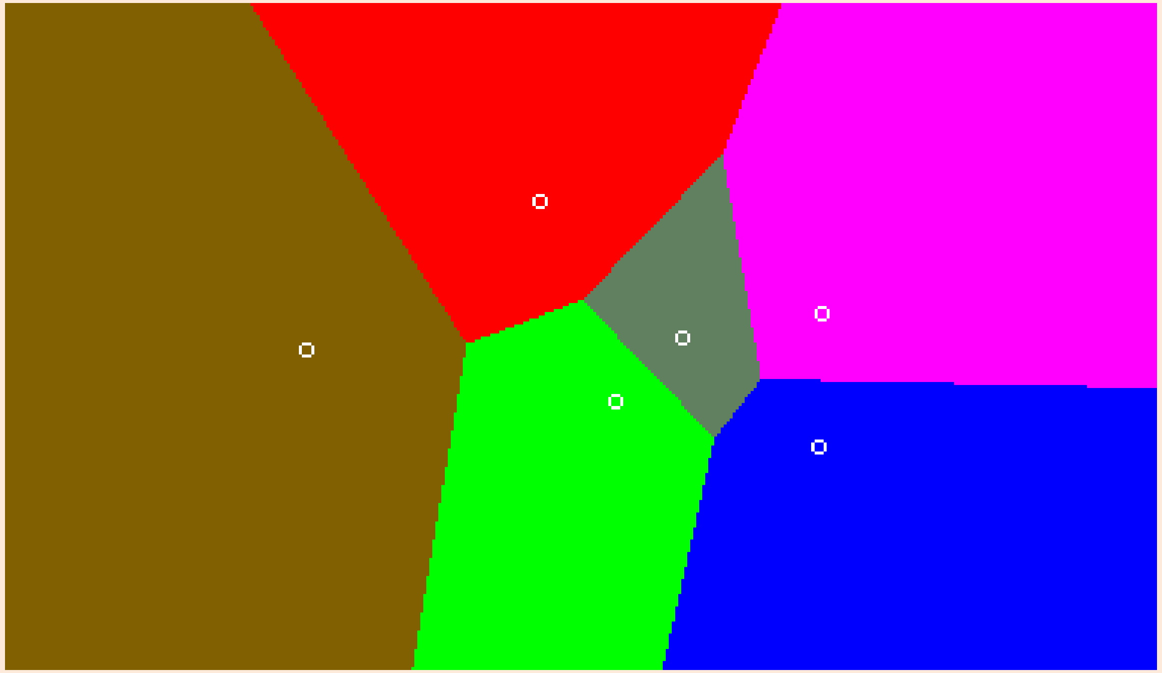
$\leq \|\theta x_i - \theta w_1\| + \|(1-\theta)x_j - (1-\theta)w_1\|$ (triangle inequality)

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

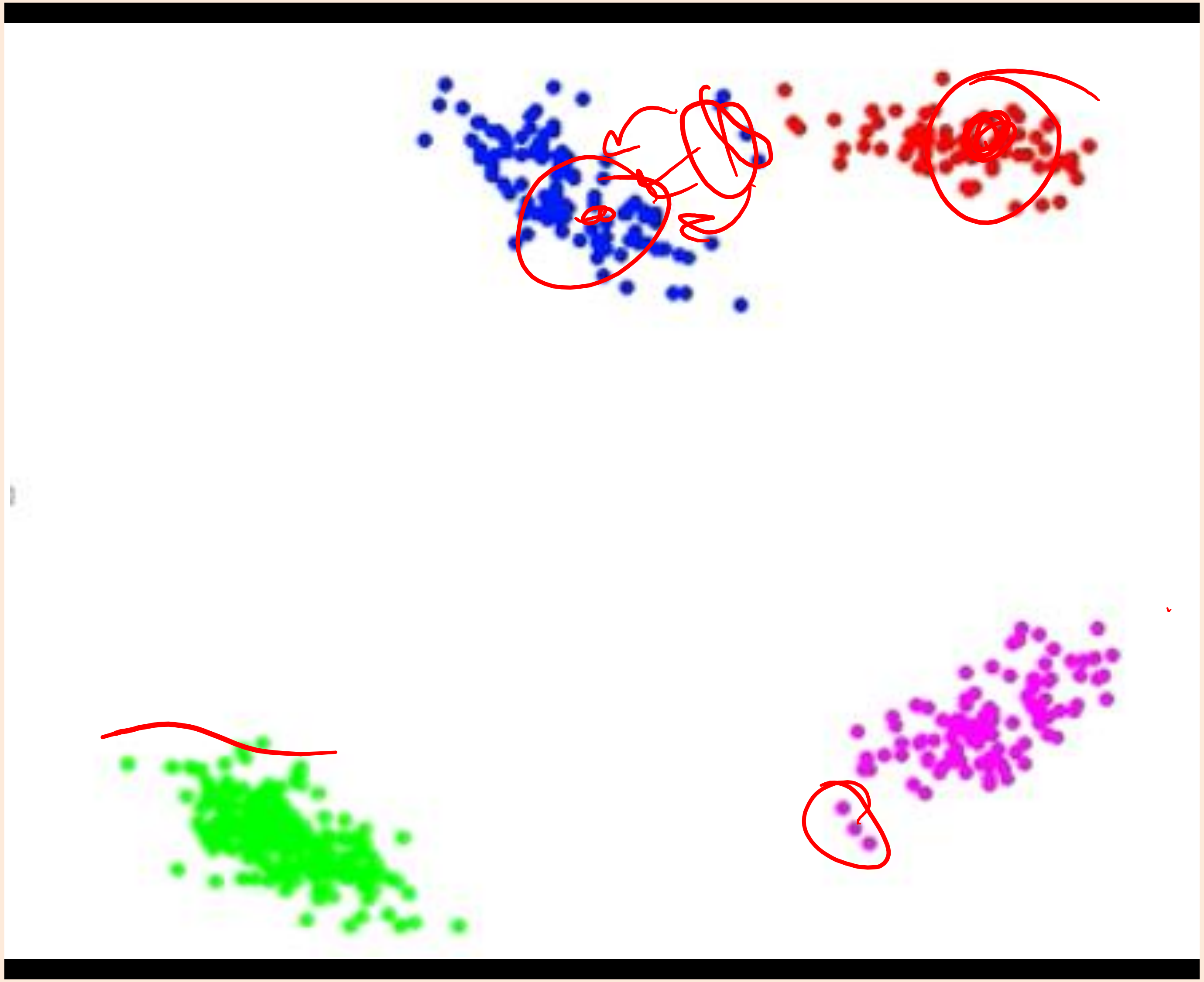
UBClustering Algorithm

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

1. Run k-means with 'm' different random initializations.
2. For each object 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.