Clustering

Matt Dickenson

Agenda

- Introduction
- Why Clustering Works
- Popular Approaches
  - DBSCAN
  - K-Means
  - Gaussian EM
- When to Use Clustering
- When Clustering Fails

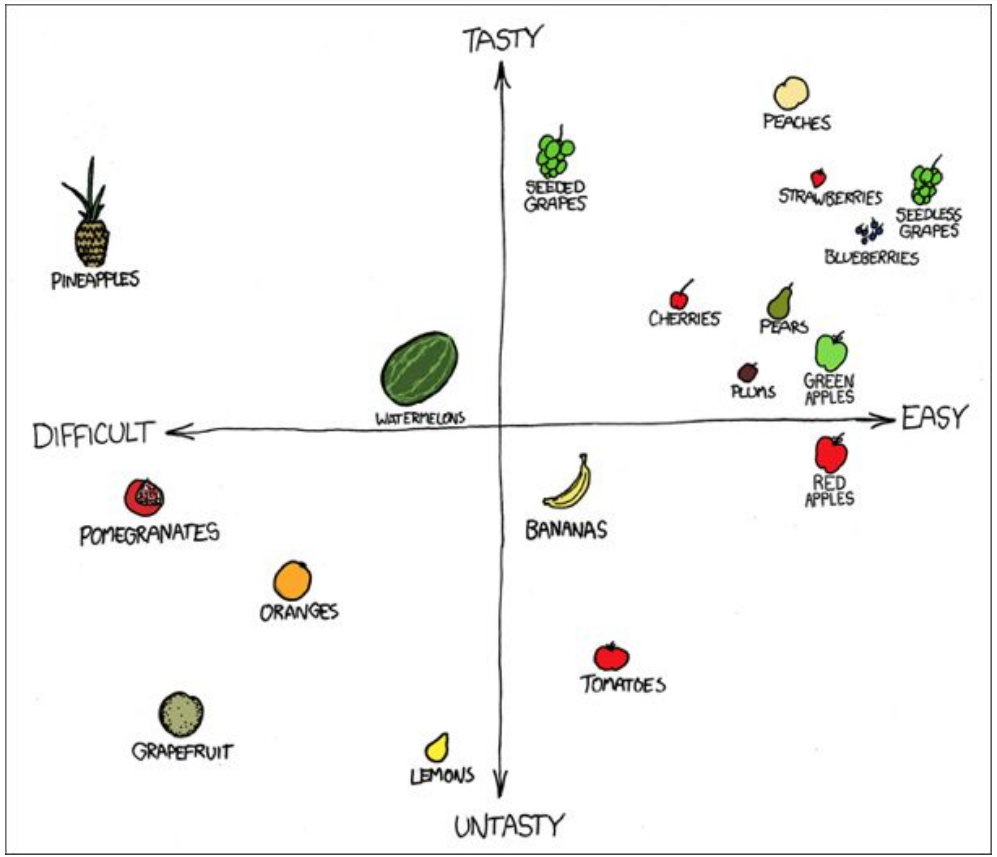
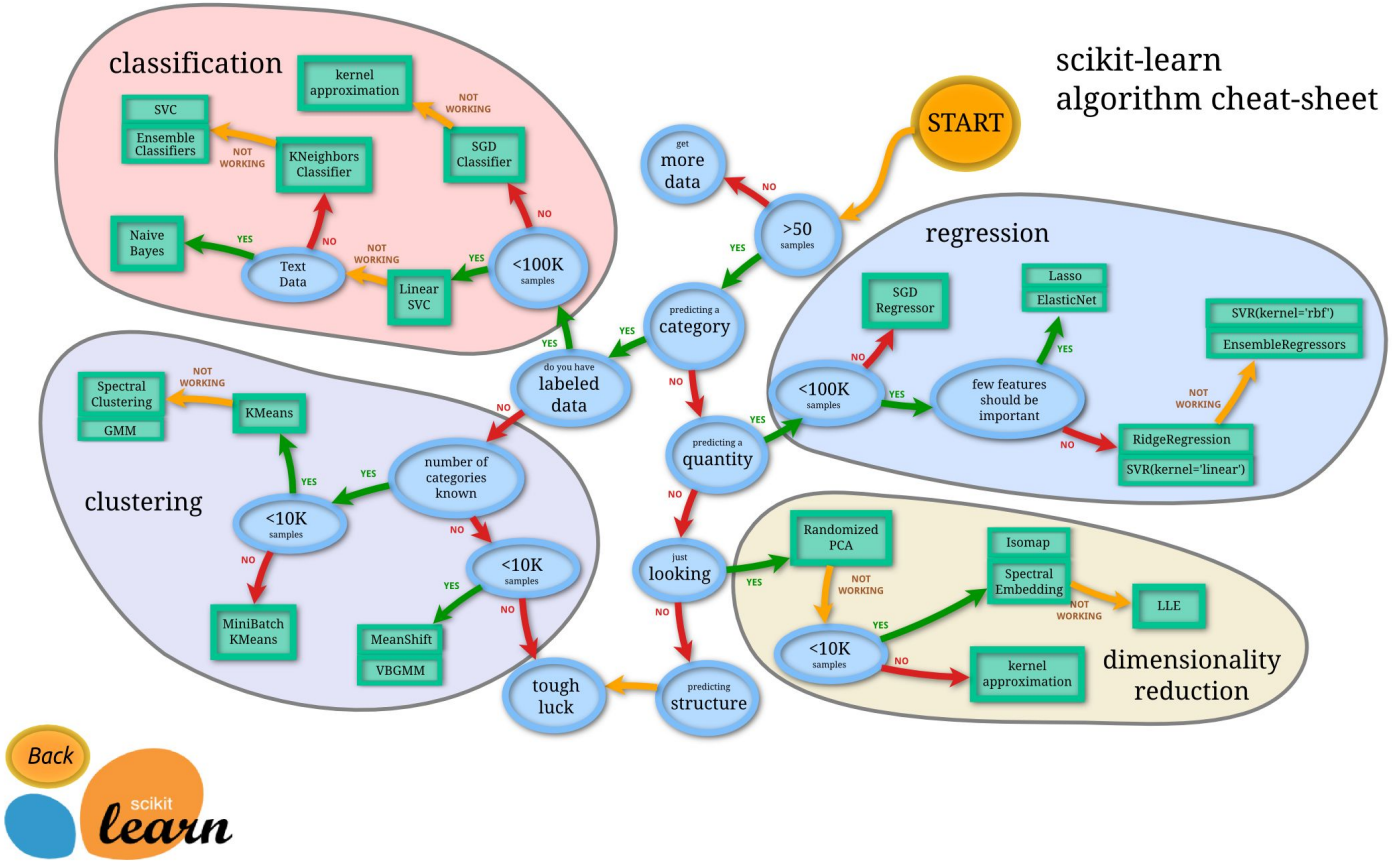
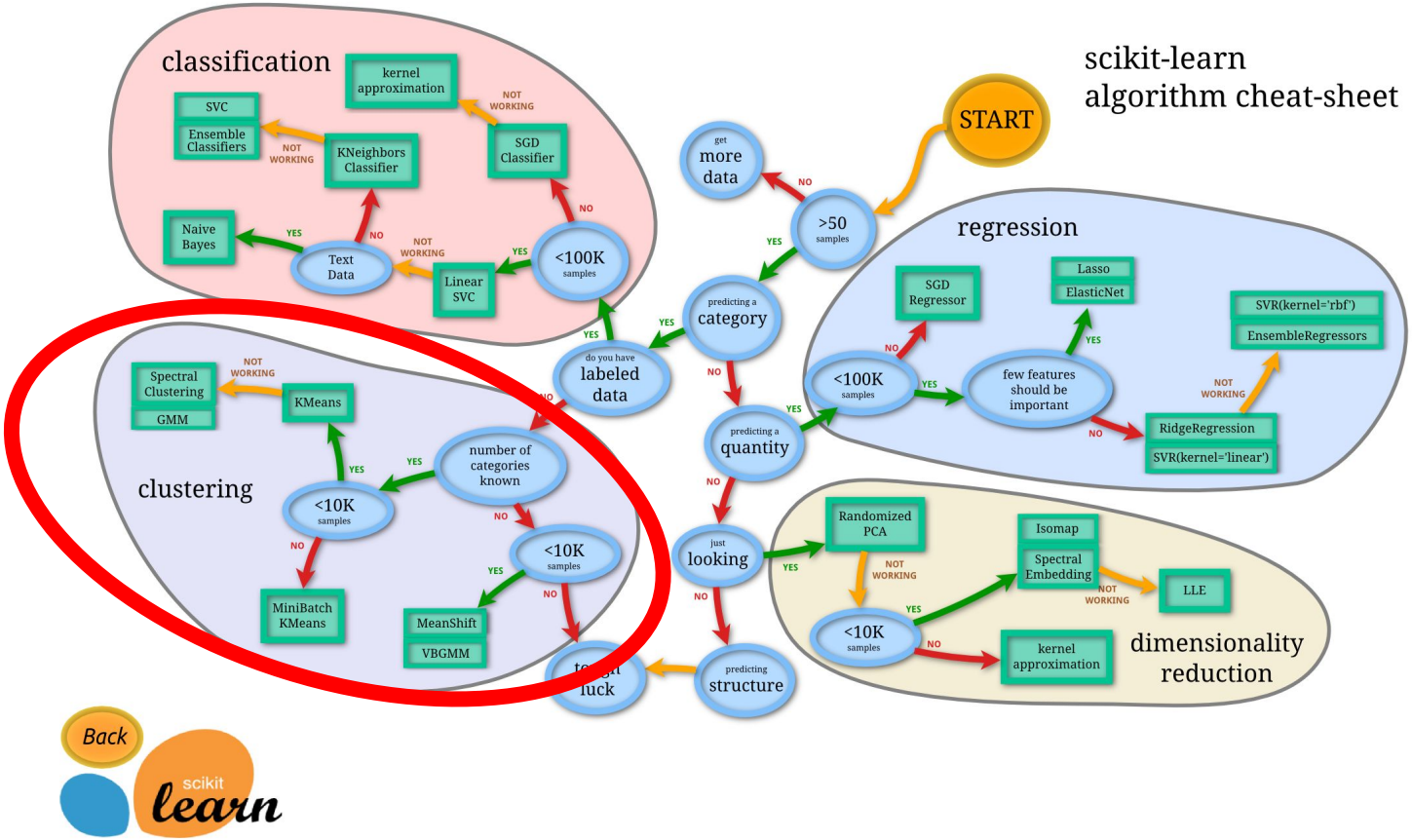


Image: https://xkcd.com/388/

Overview

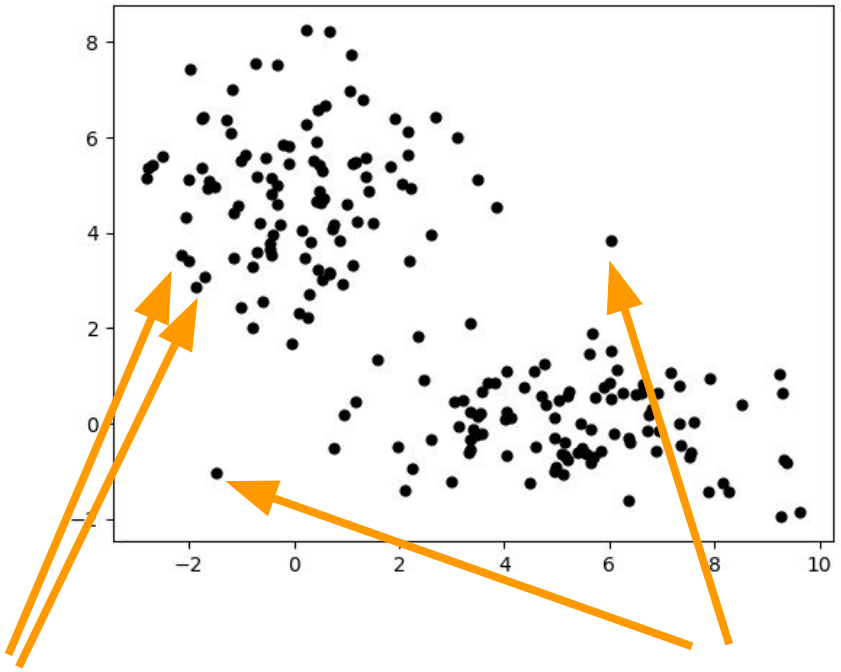

scikit-learn
algorithm cheat-sheet
START
get more data
>50 samples
predicting a category
do you have labeled data
predicting a quantity
just looking
predicting structure
tough luck
classification
kernel approximation
SVC
Ensemble Classifiers
KNeighbors Classifier
SGD Classifier
Naive Bayes
Text Data
Linear SVC
<100K samples
clustering
Spectral Clustering
GMM
KMeans
number of categories known
<10K samples
MiniBatch KMeans
MeanShift
VBGMM
regression
SGD Regressor
Lasso
ElasticNet
<100K samples
few features should be important
SVR(kernel='rbf')
EnsembleRegressors
RidgeRegression
SVR(kernel='linear')
dimensionality reduction
Randomized PCA
Isomap
Spectral Embedding
LLE
<10K samples
kernel approximation
YES
NO
NOT WORKING
Back
scikit learn

Overview


scikit-learn
algorithm cheat-sheet
START
get more data
NO
>50 samples
YES
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
NO
Text Data
NOT WORKING
Linear SVC
YES
<100K samples
NO
predicting a category
YES
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
<100K samples
YES
few features should be important
YES
NOT WORKING
RidgeRegression
NO
SVR(kernel='linear')
do you have labeled data
YES
NO
predicting a quantity
YES
NO
clustering
Spectral Clustering
GMM
NOT WORKING
KMeans
YES
number of categories known
YES
NO
<10K samples
NO
<10K samples
YES
NO
MiniBatch KMeans
MeanShift
VBGMM
just looking
YES
Randomized PCA
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction
NO
predicting structure
Back
scikit learn

Overview

- Unsupervised approach
  - No labeled data
- Group similar points together
  - How to define similarity?



Why Clustering Works

- Often datasets consist of underlying groups

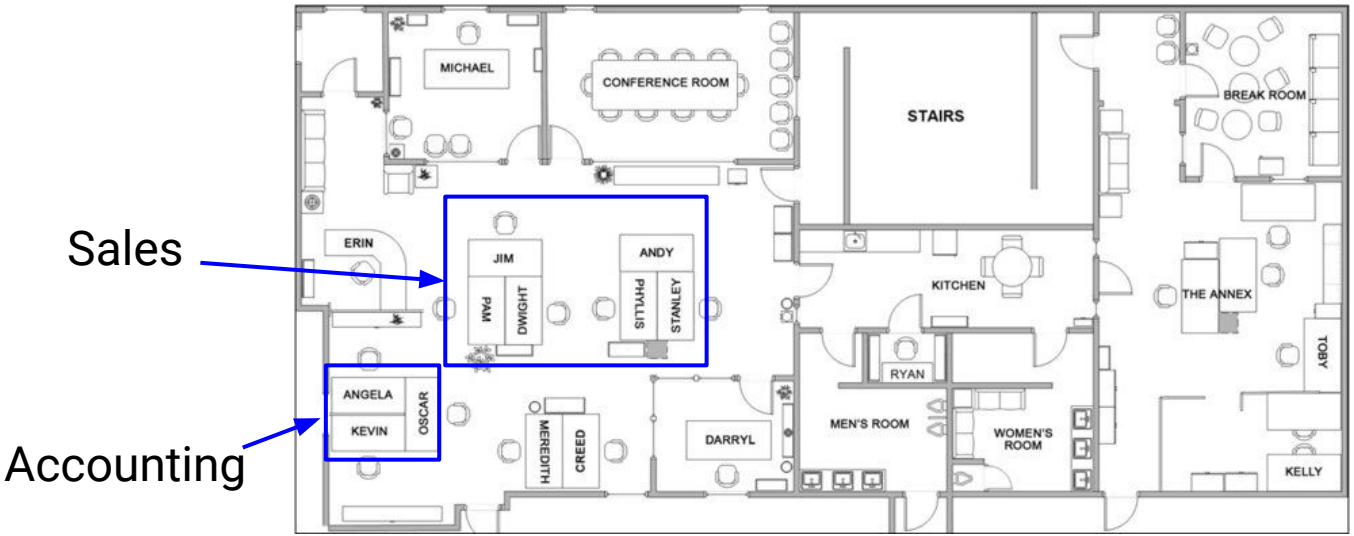
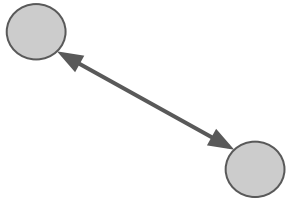


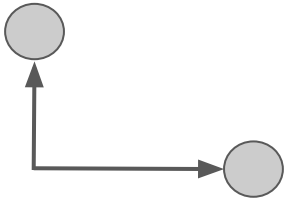
Image: http://theoffice.wikia.com

Why Clustering Works

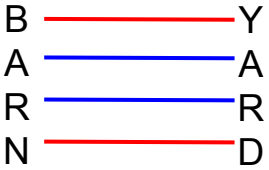
- Often datasets consist of underlying groups
- Usually simple to define similarity as a distance



Euclidean

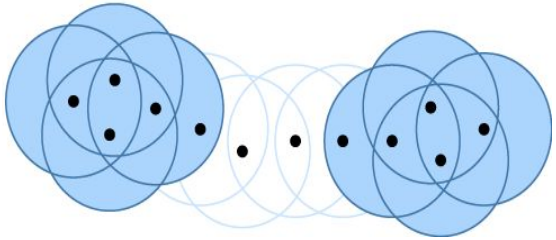


Manhattan

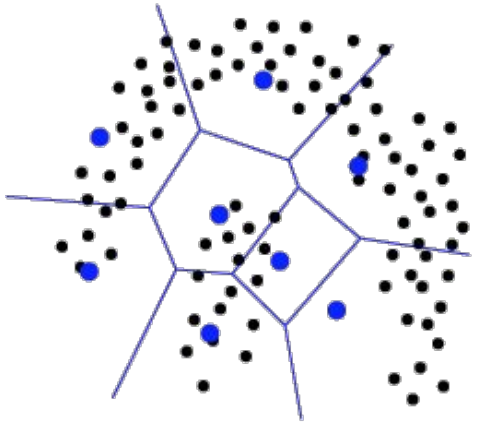


Levenshtein

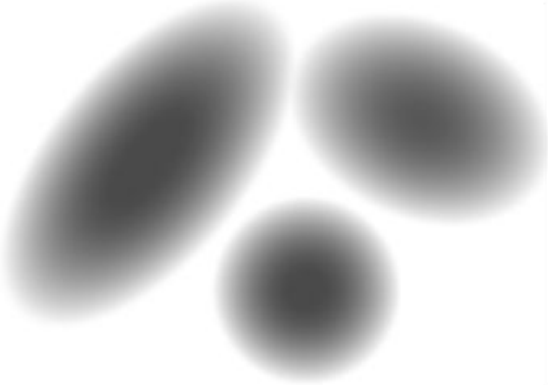
Popular Clustering Algorithms



DBSCAN
neighborhood-based



K-Means
centroid-based



Gaussian E-M
distribution-based

Images: http://shapeofdata.wordpress.com

DBSCAN

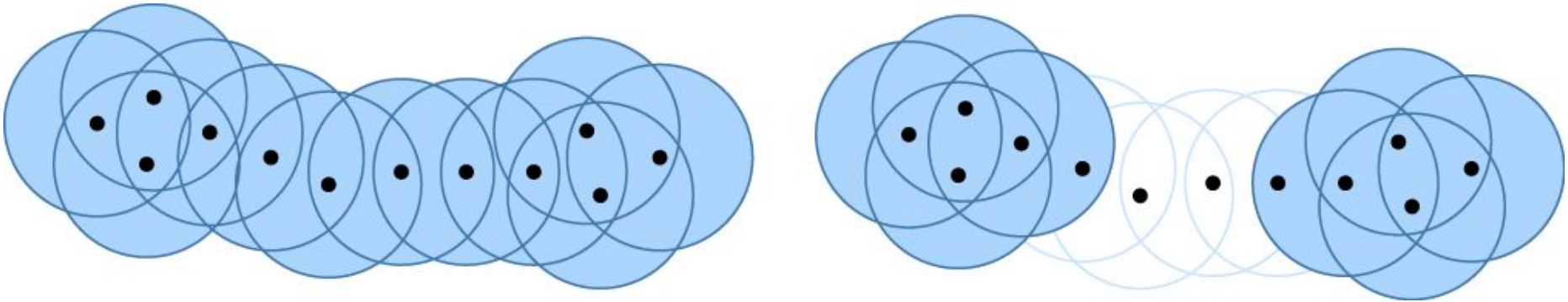


Image: http://shapeofdata.wordpress.com

DBSCAN

“Density-based spatial clustering of applications with noise”

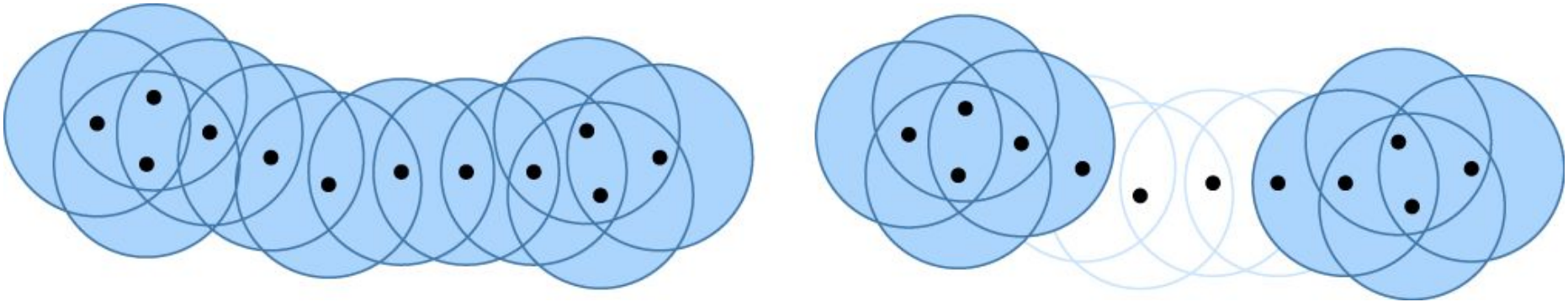
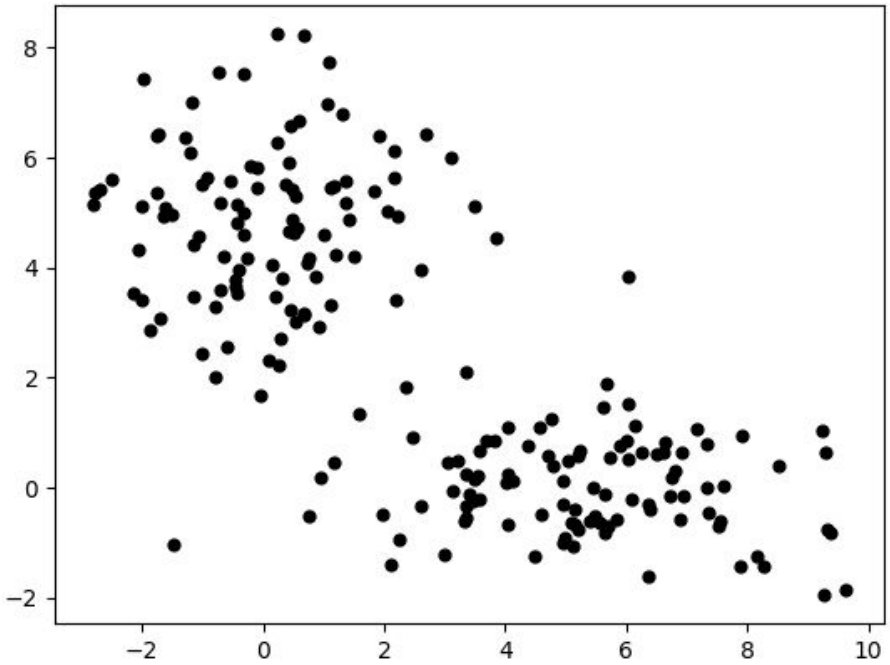


Image: http://shapeofdata.wordpress.com

DBSCAN

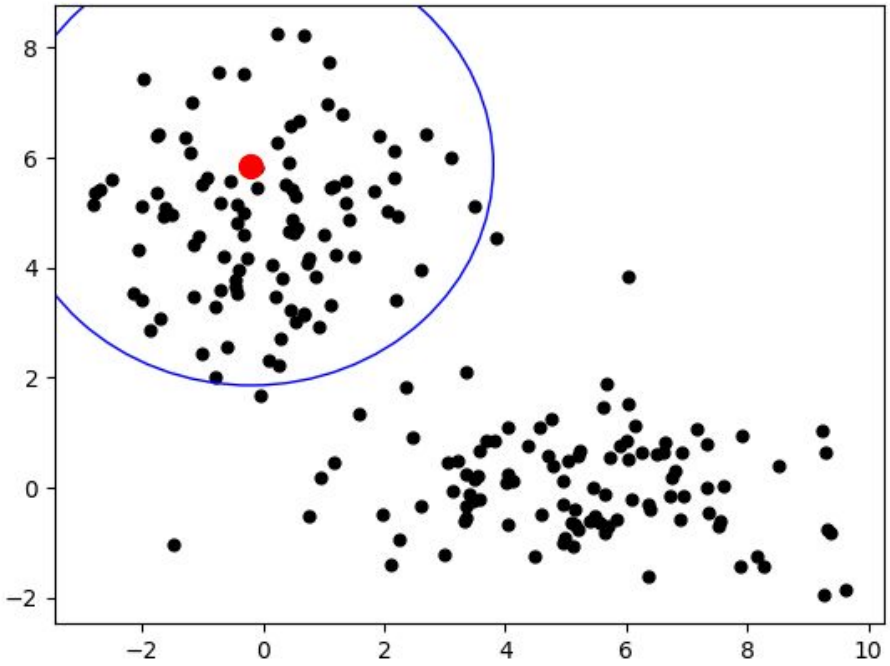
- 2 parameters
  - ε controls the size of a point's "neighborhood"
  - How many neighbors a point must have to be considered "core"
- Result
  - List of core points
  - Assigned points & outliers

DBSCAN



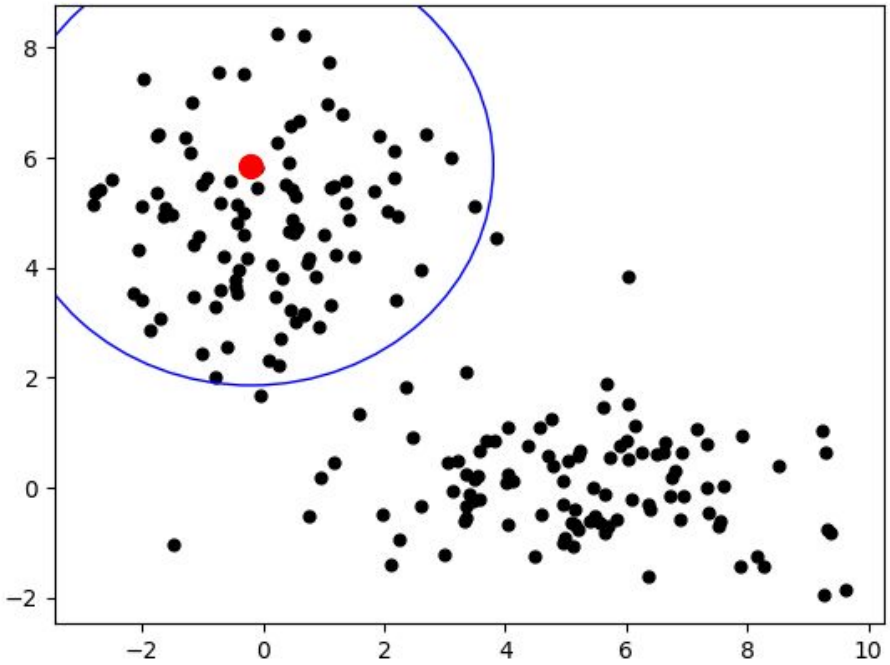
DBSCAN

ε = 4, minPts = 30



DBSCAN

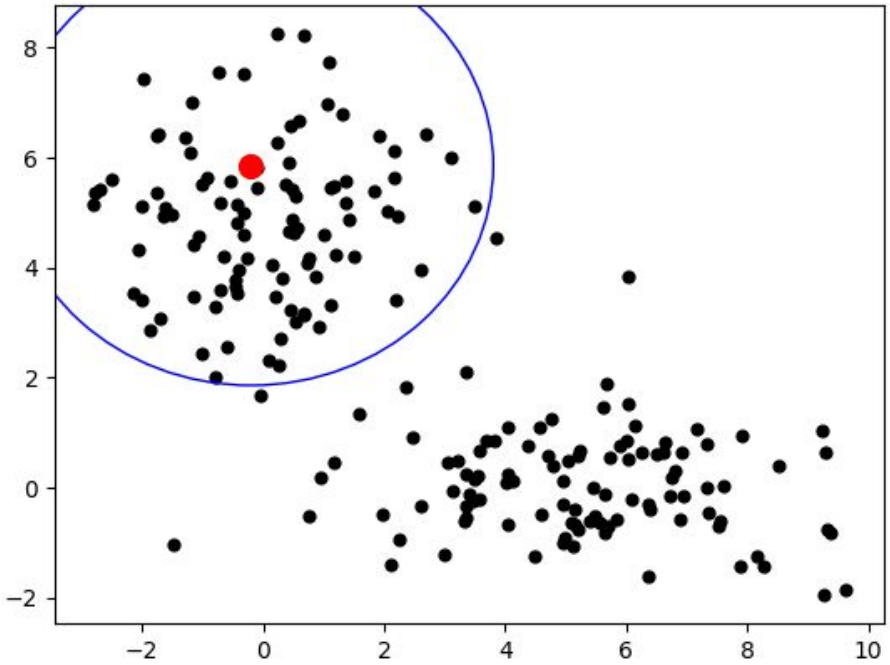
Iterate through points, count neighbors within ε distance



ε = 4, minPts = 30

DBSCAN

If the point has |neighbors| > minPts, consider it “core”

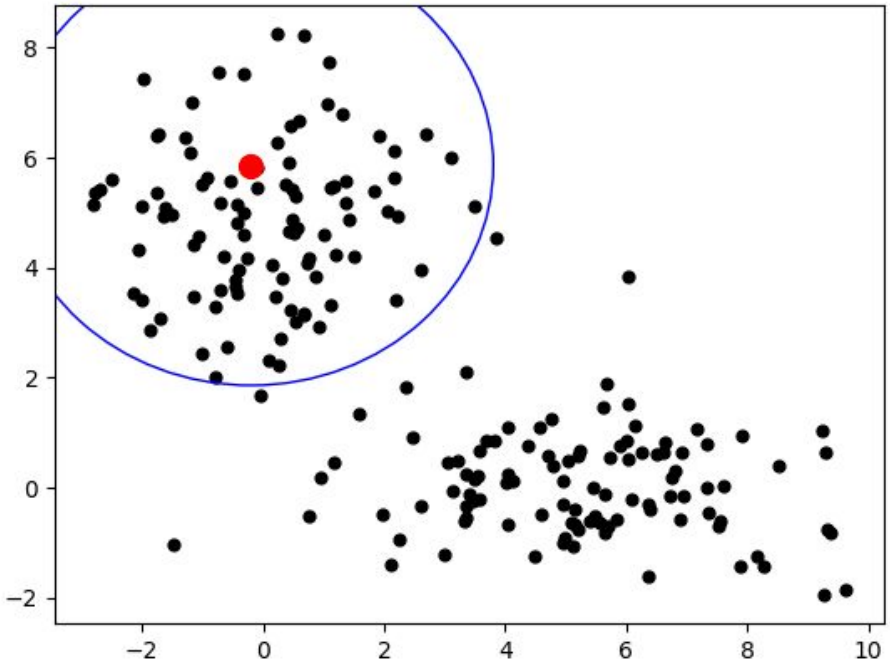


ε = 4, minPts = 30

DBSCAN

If the point has |neighbors| > minPts, consider it “core”

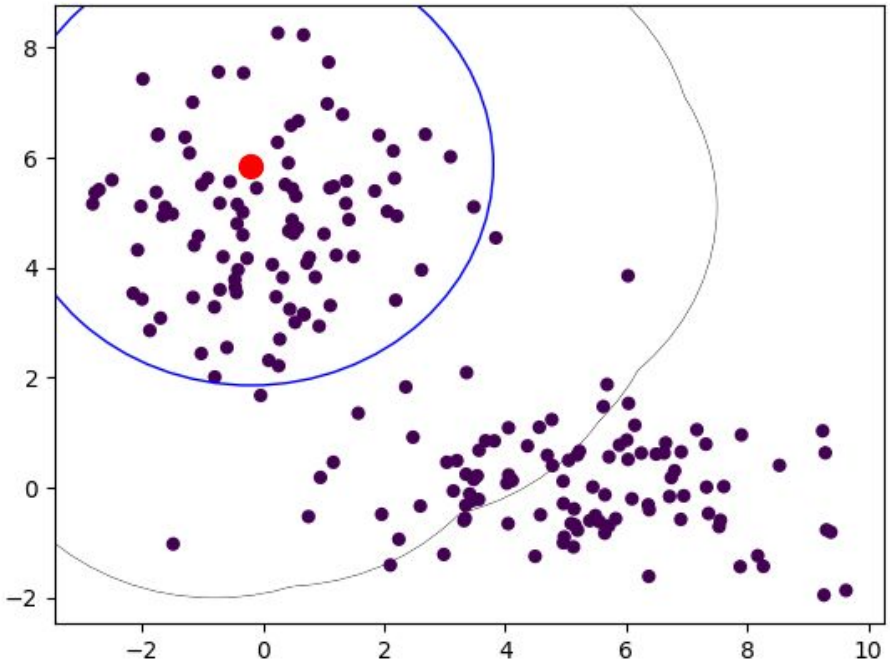
97 neighbors



ε = 4, minPts = 30

DBSCAN

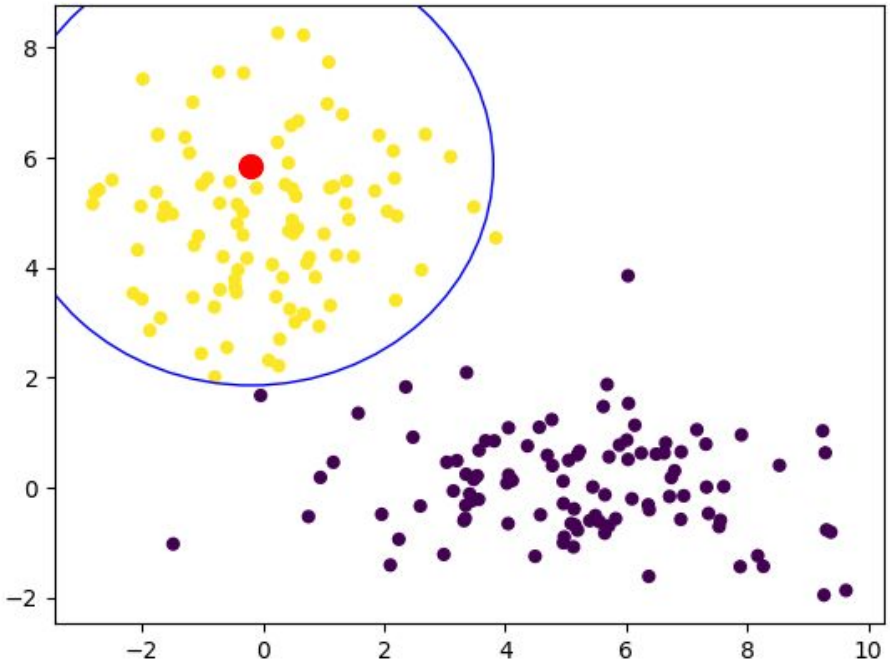
Check neighbors' neighbors for density



ε = 4, minPts = 30

DBSCAN

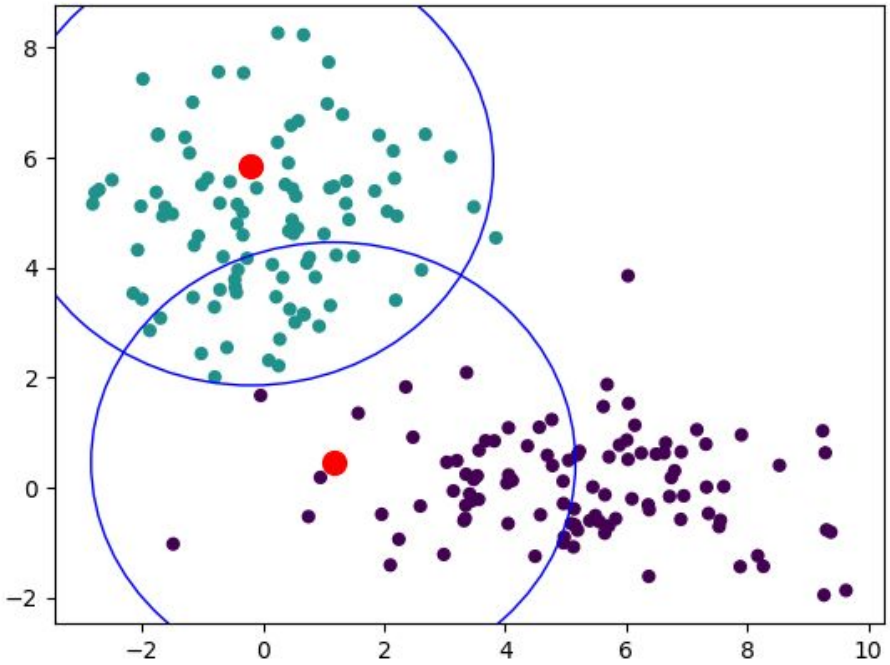
Assign core point's neighbors to its cluster



ε = 4, minPts = 30

DBSCAN

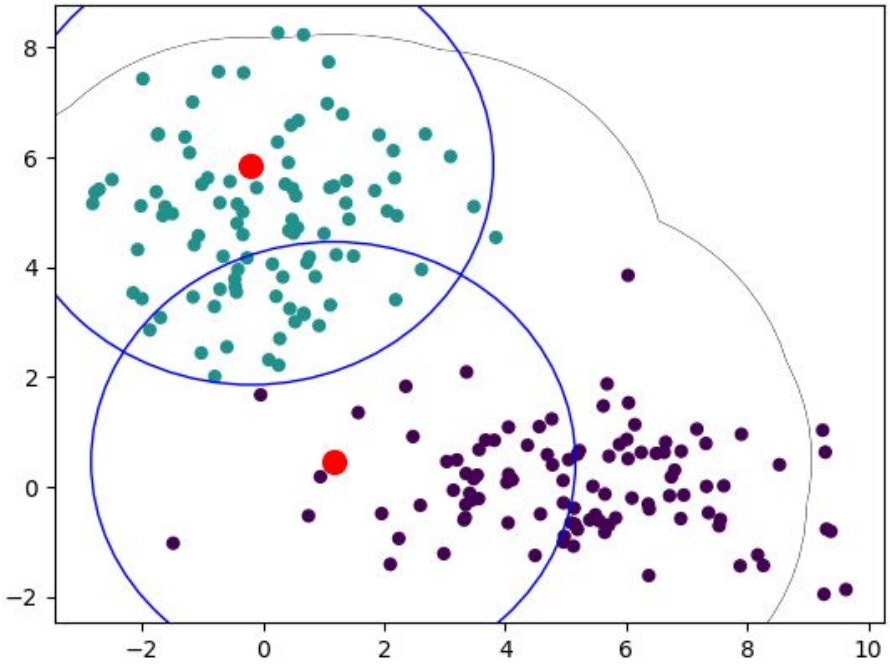
Iterate through points until you find next core point



ε = 4, minPts = 30

DBSCAN

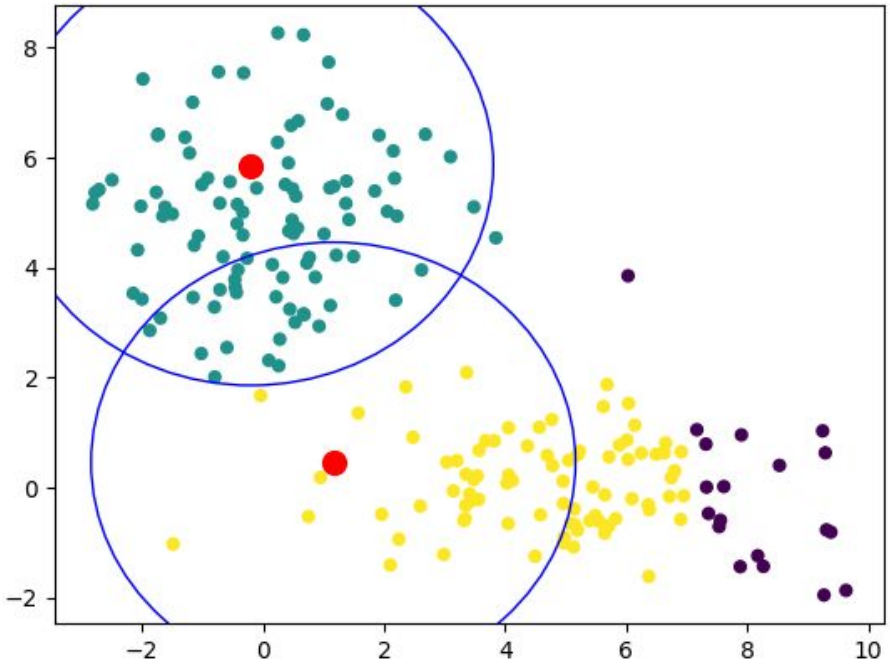
Check neighbors' neighbors for density



ε = 4, minPts = 30

DBSCAN

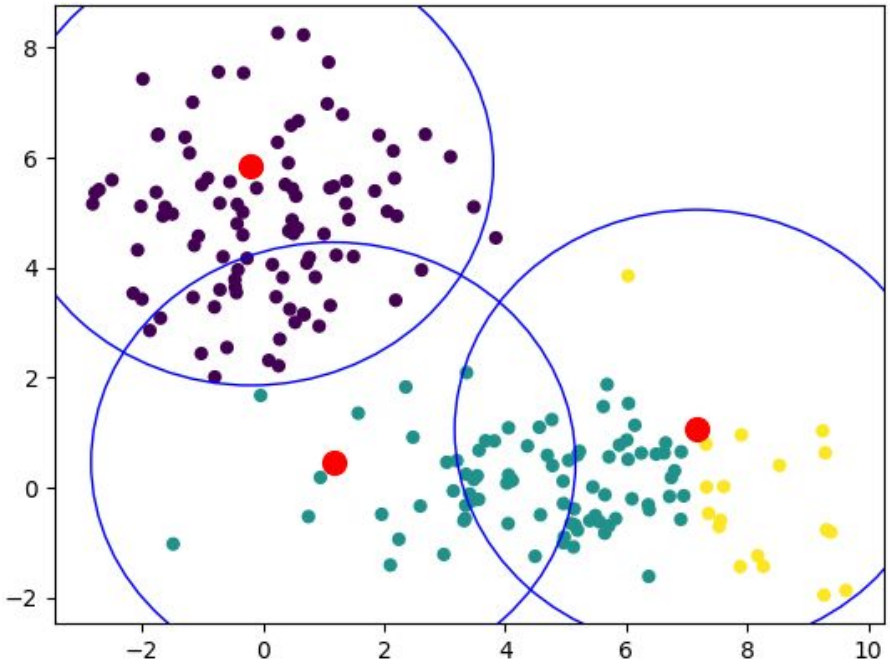
Assign unlabeled neighbors to new core point



ε = 4, minPts = 30

DBSCAN

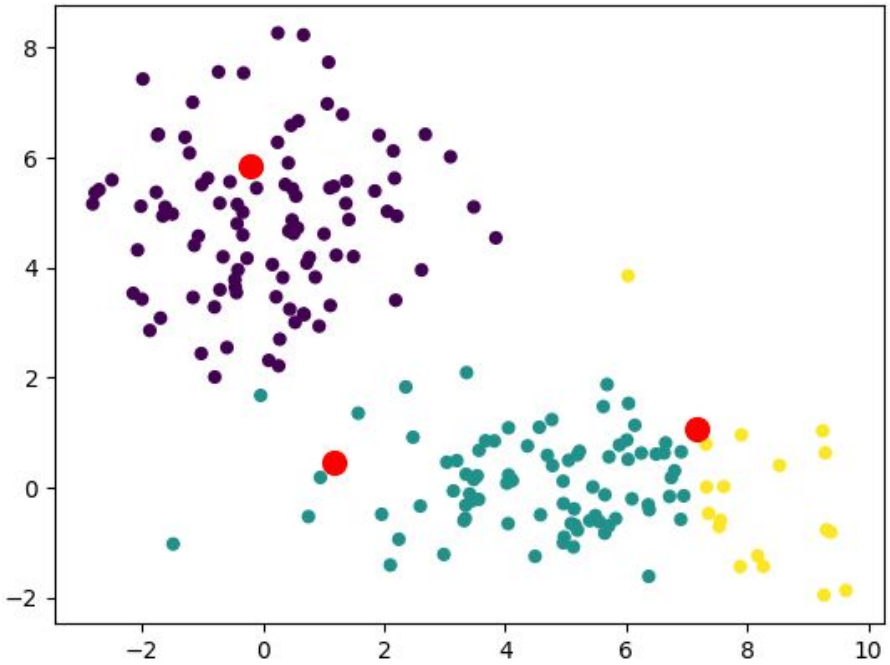
Continue until all points have been checked



ε = 4, minPts = 30

DBSCAN

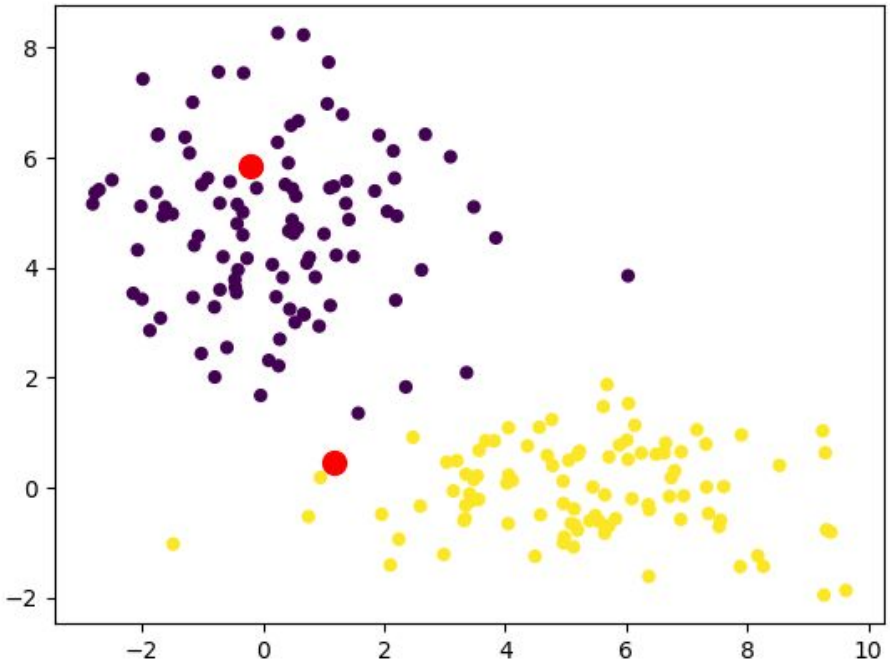
Result: 3 clusters



ε = 4, minPts = 30

DBSCAN

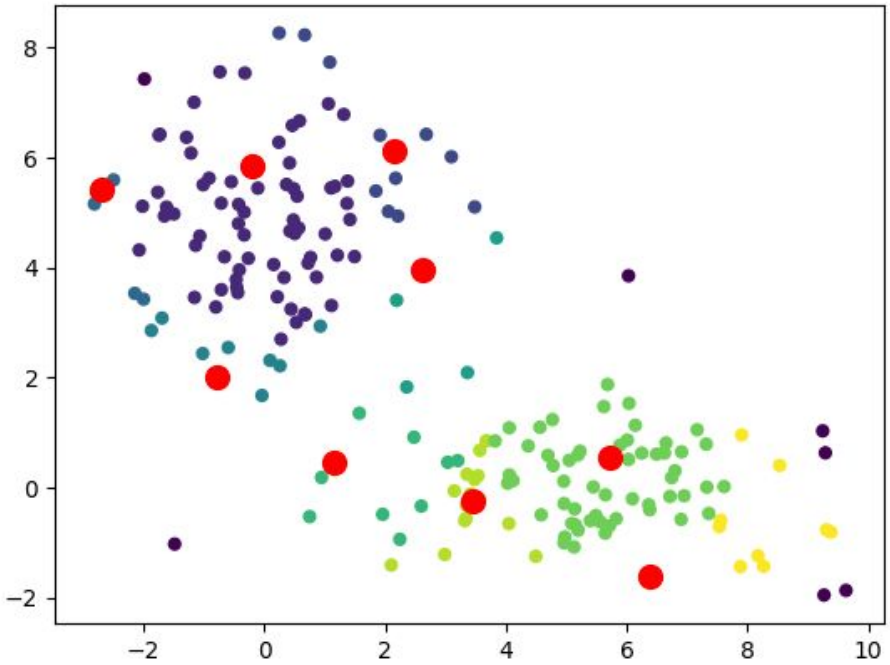
Result: 2 clusters



ε = 5, minPts = 30

DBSCAN

Result: 9 clusters



ε = 2, minPts = 10

DBSCAN

- Pros
  - Don’t have to specify number of clusters
  - Fairly robust to outliers
  - Handles non-spherical clusters
- Cons
  - Non-deterministic
    - For neighbors of two core points, cluster assignment depends on order that data is processed
  - Non-probabilistic
    - Returns cluster assignments without uncertainty/probability
    - Tuning parameters can be difficult

K-Means

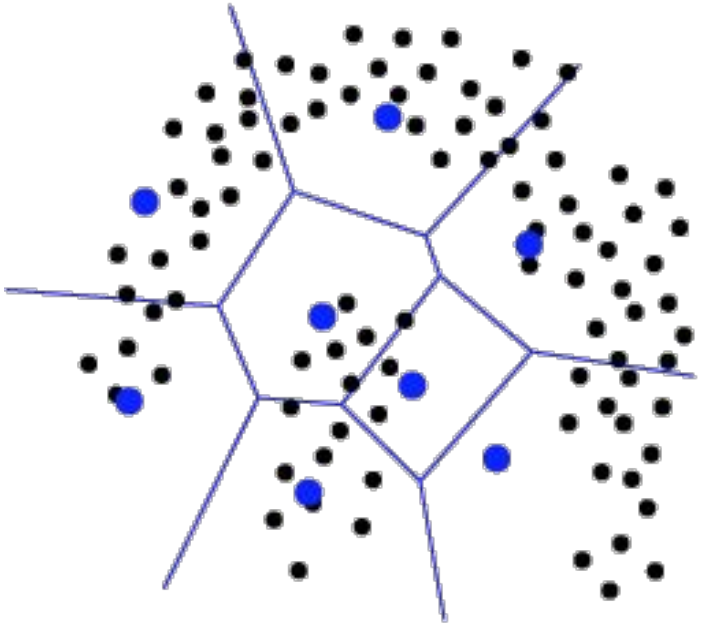
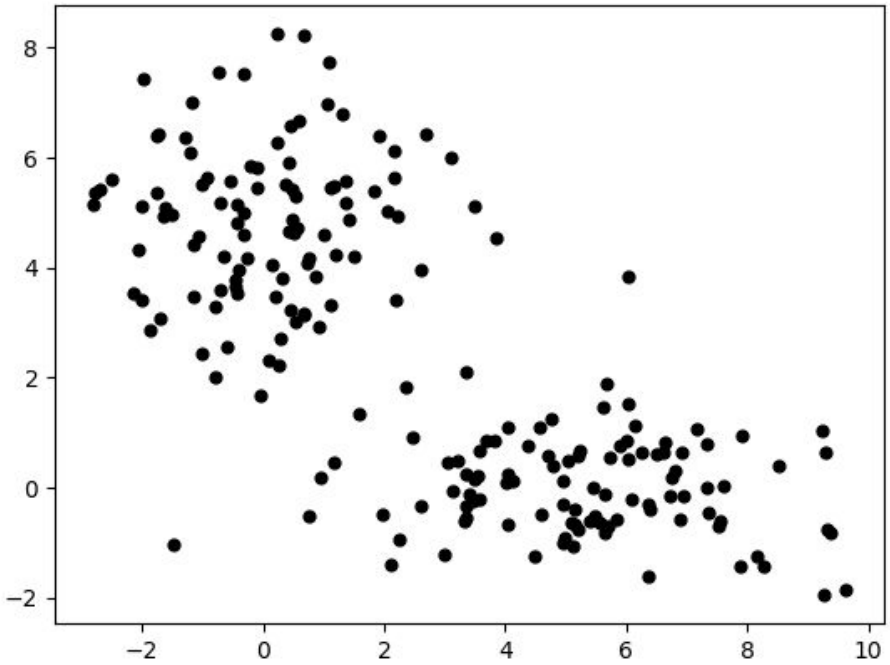


Image: http://shapeofdata.wordpress.com

K-Means

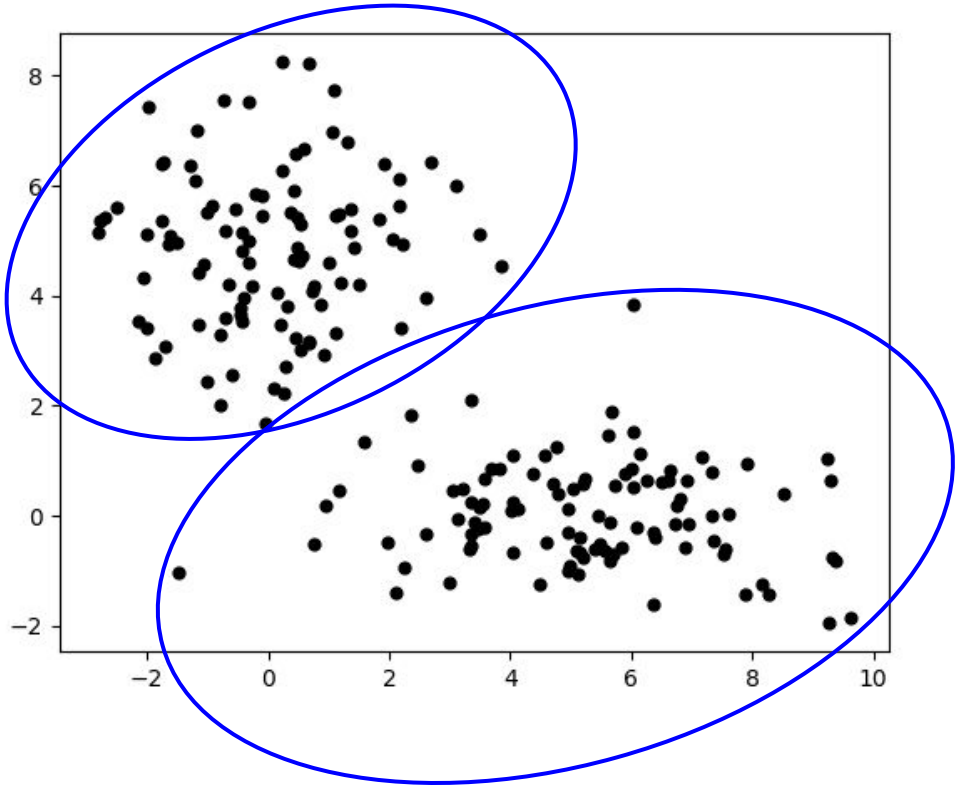
- 1 parameter, *k*, controls number of clusters
- Distance function
  - Typically Euclidean distance in feature space
- Result: list of centroids and point labels
- Caveat: there are many ways to implement K-Means

K-Means



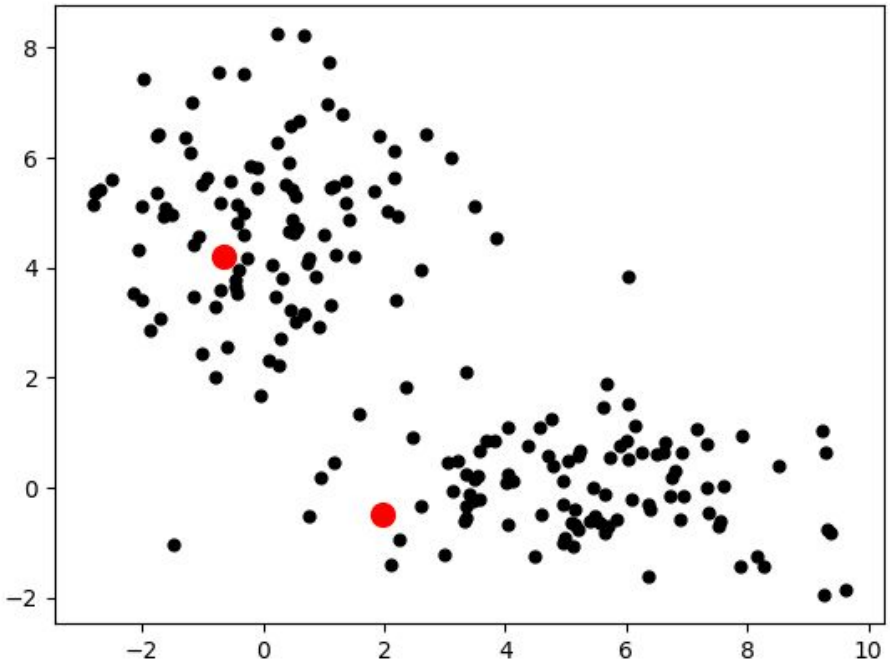
K-Means

Hypothesis: k=2



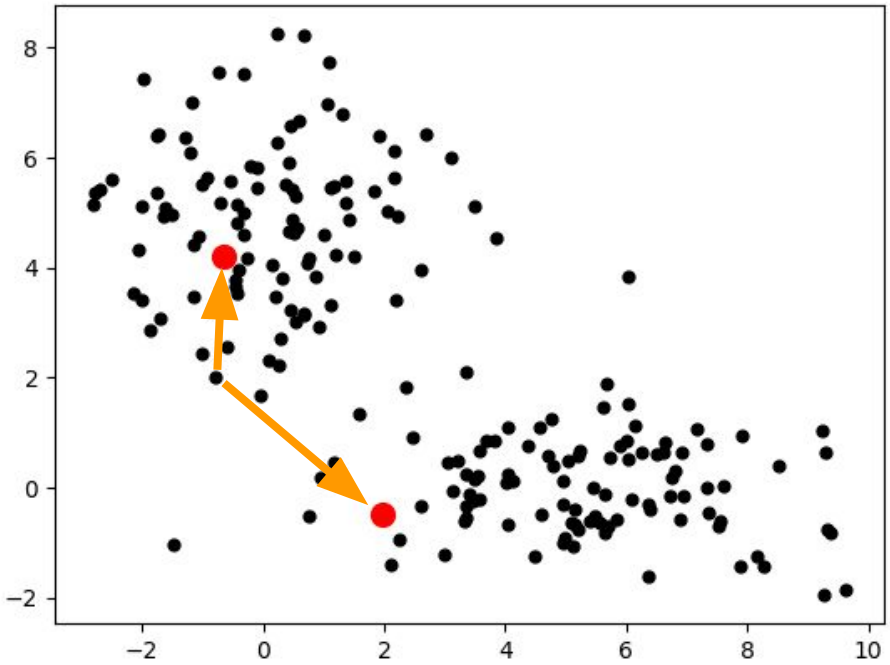
K-Means

Pick *k* random data points as initial centroids



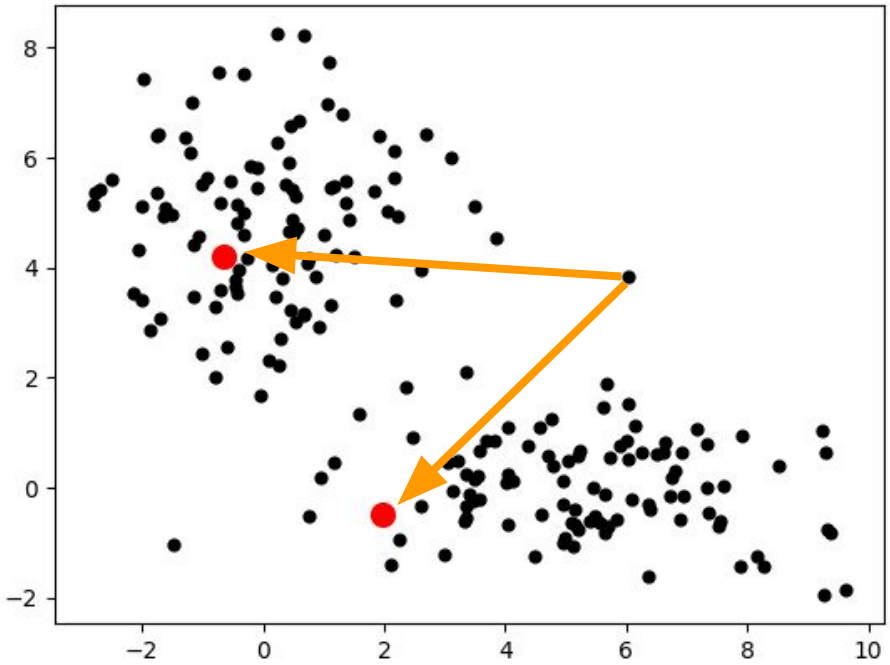
K-Means

Compute the distance of each point from each centroid



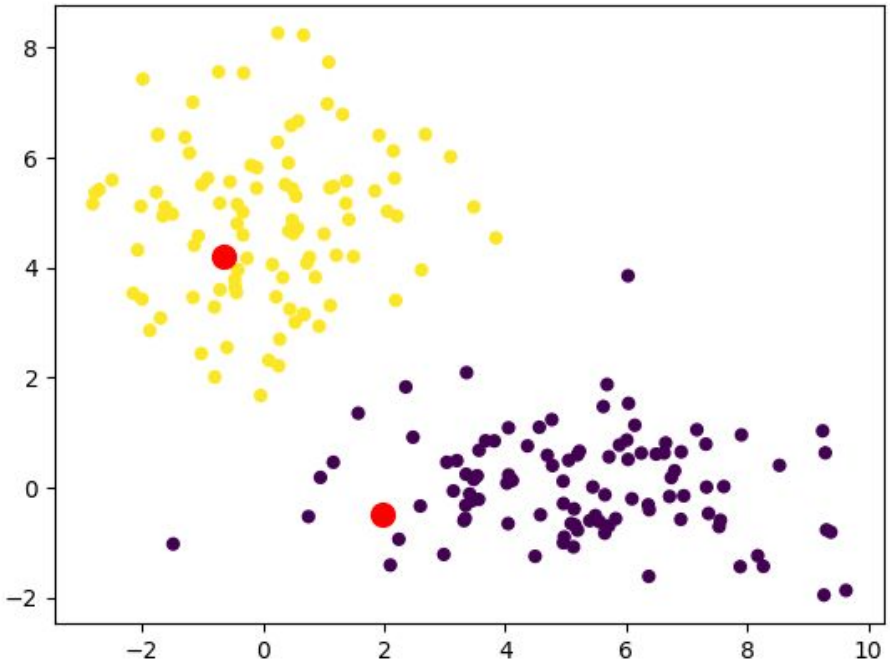
K-Means

Compute the distance of each point from each centroid



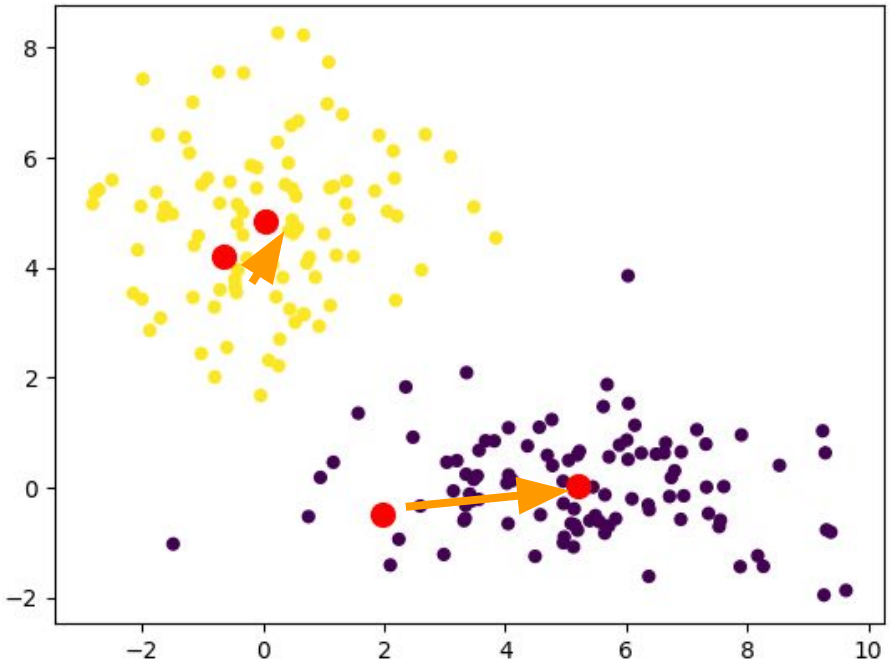
K-Means

Assign each point to the closest cluster



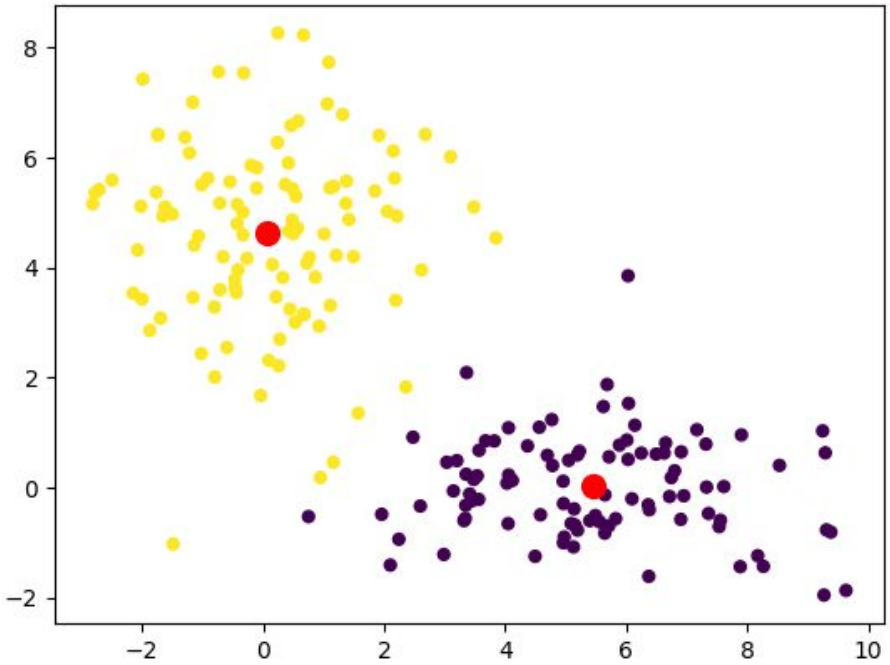
K-Means

Update centroids to the mean of new clusters



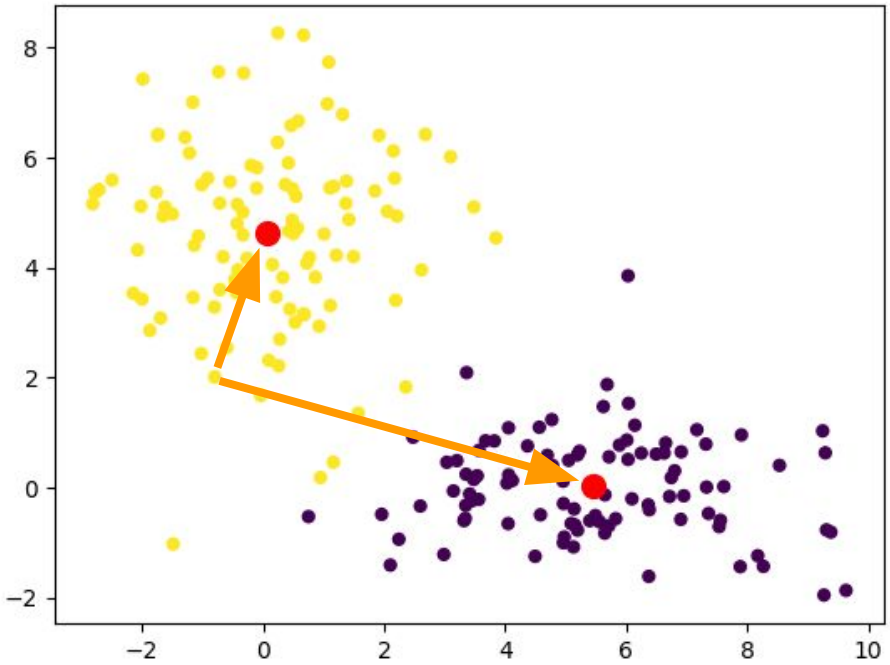
K-Means

Update centroids to the mean of new clusters



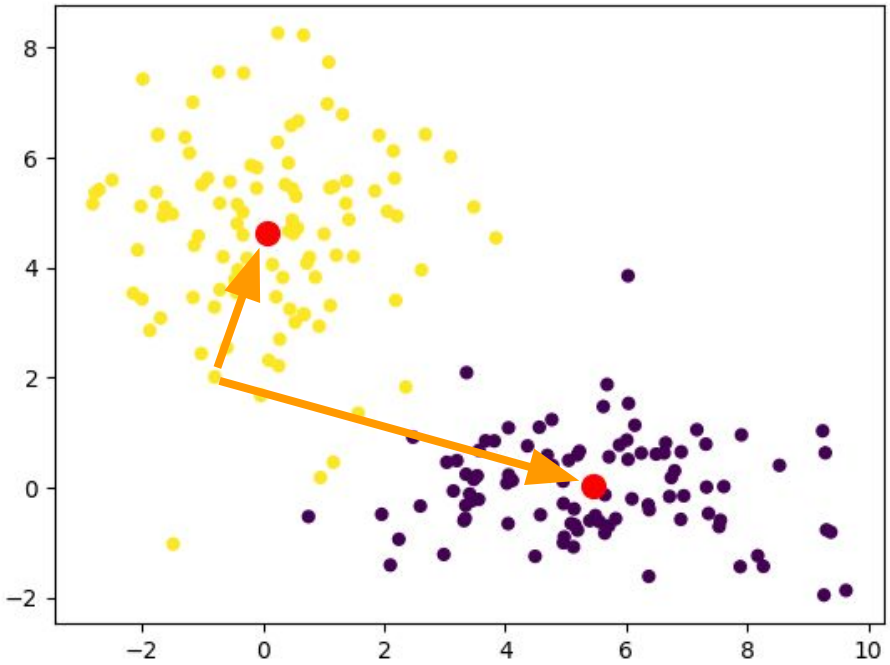
K-Means

Repeat until clusters converge



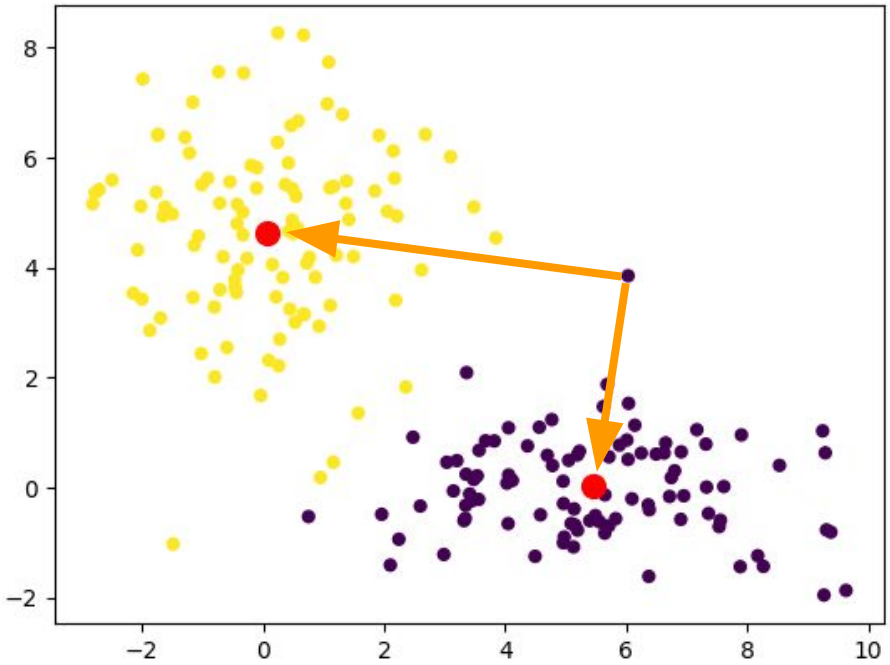
K-Means

Repeat until clusters converge ← little/no change between iters



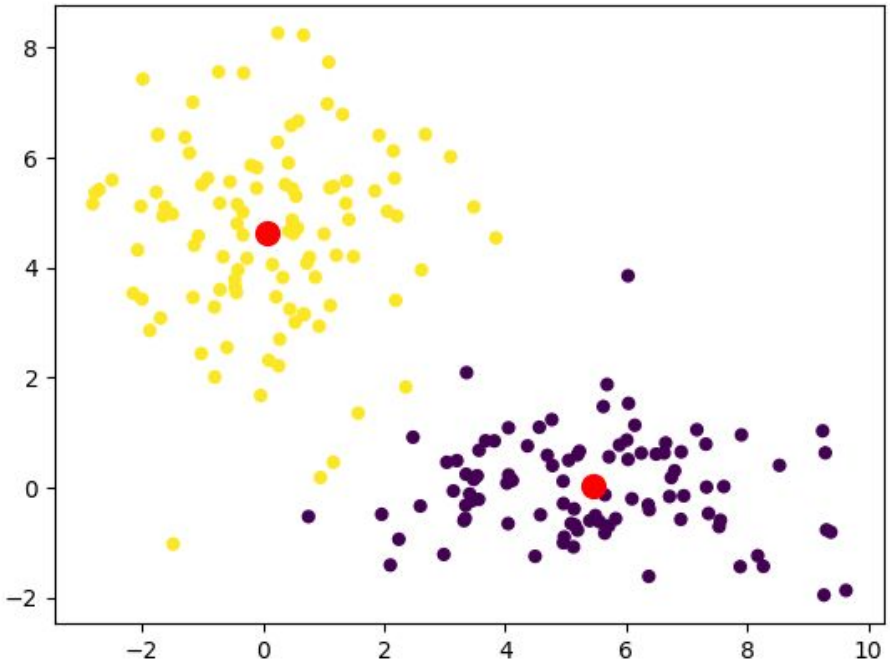
K-Means

Repeat until clusters converge



K-Means

Repeat until clusters converge



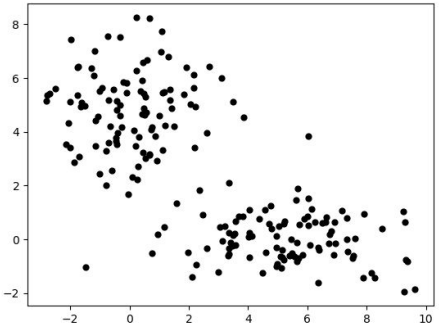
K-Means

- Pros
  - Quick to implement (and usually quick to run)
  - Simple, intuitive algorithm

- Cons
  - Can be difficult to choose *k* in practice
  - Clusters will all have the same radius
  - Sensitive to outliers

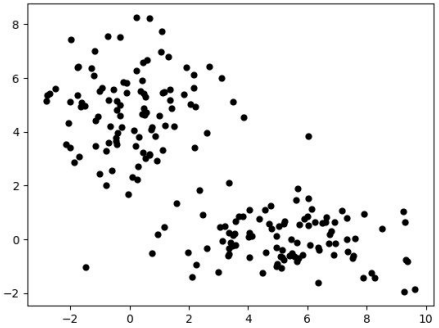
How do I choose a good value of k?

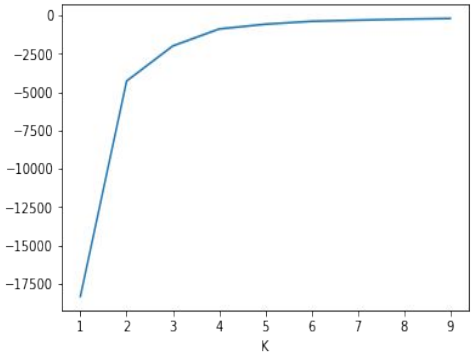
- Visually inspect the data
  - This is impractical for high-dimensional data



How do I choose a good value of k?

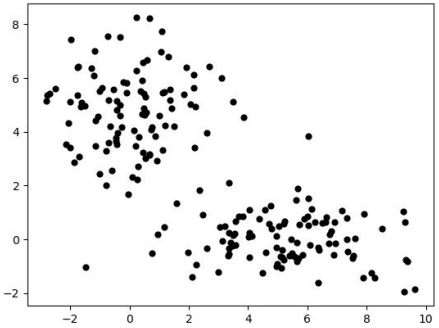
- Visually inspect the data
  - This is impractical for high-dimensional data
- Try several different values
  - Which value of k minimizes the total error?
  - Increasing k will always decrease the total error
  - Pick the value that gives the greatest gain (“elbow method”)

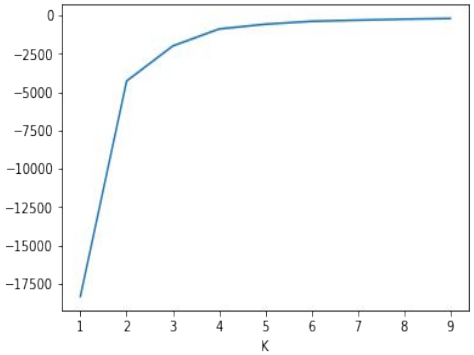




How do I choose a good value of k?

- Visually inspect the data
  - This is impractical for high-dimensional data
- Try several different values
  - Which value of *k* minimizes the total error?
  - Increasing *k* will always decrease the total error
  - Pick the value that gives the greatest gain (“elbow method”)
- Other metrics
  - Silhouette score (more detail next week)





Expectation-Maximization

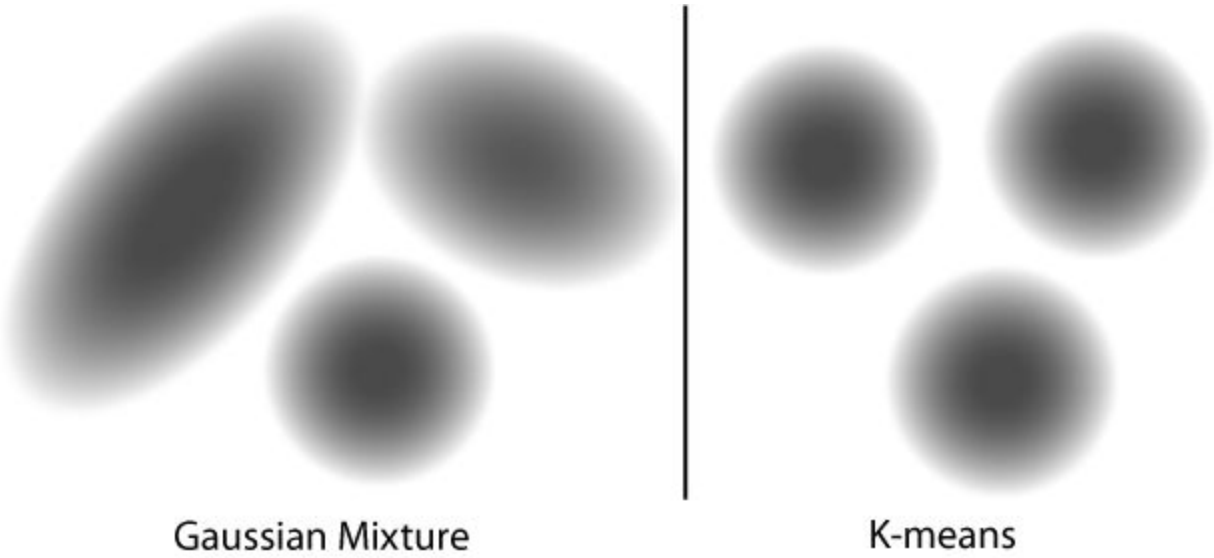
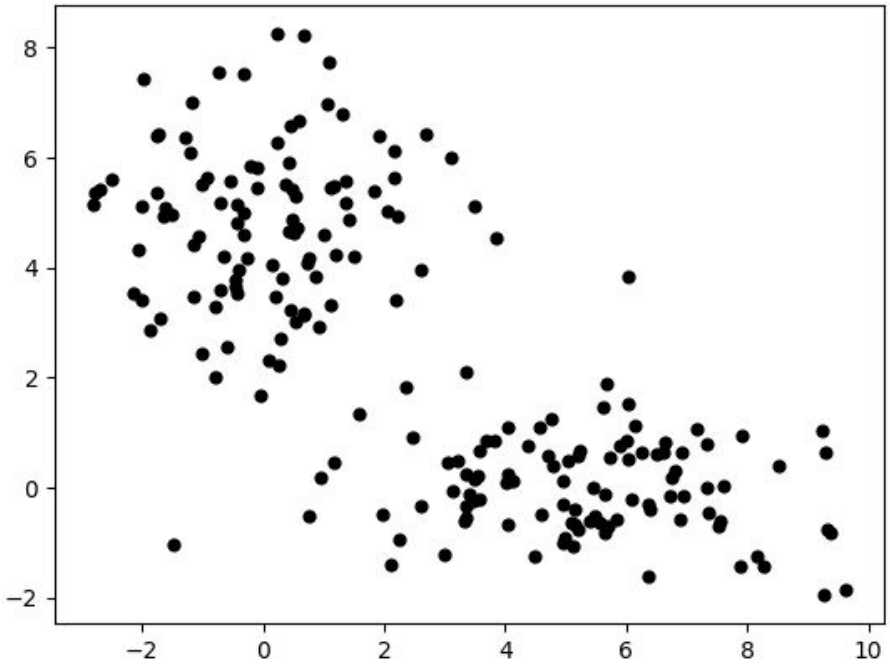


Image: http://shapeofdata.wordpress.com

Expectation-Maximization

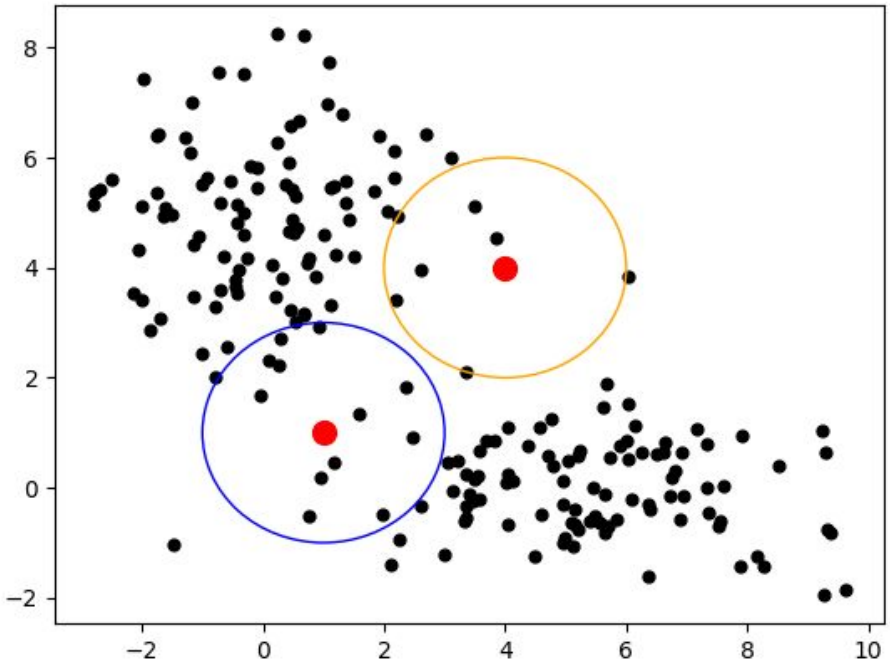
- You can use EM with a variety of distributions
- We will use it with a mixture of Gaussians (GMM)
- 2 parameters per cluster
  - μ represents cluster center
  - σ represents cluster shape & scale (standard deviation)
- Result
  - Parameters for final clusters
  - Labels for each point (“soft” assignments)

Gaussian EM



Gaussian EM

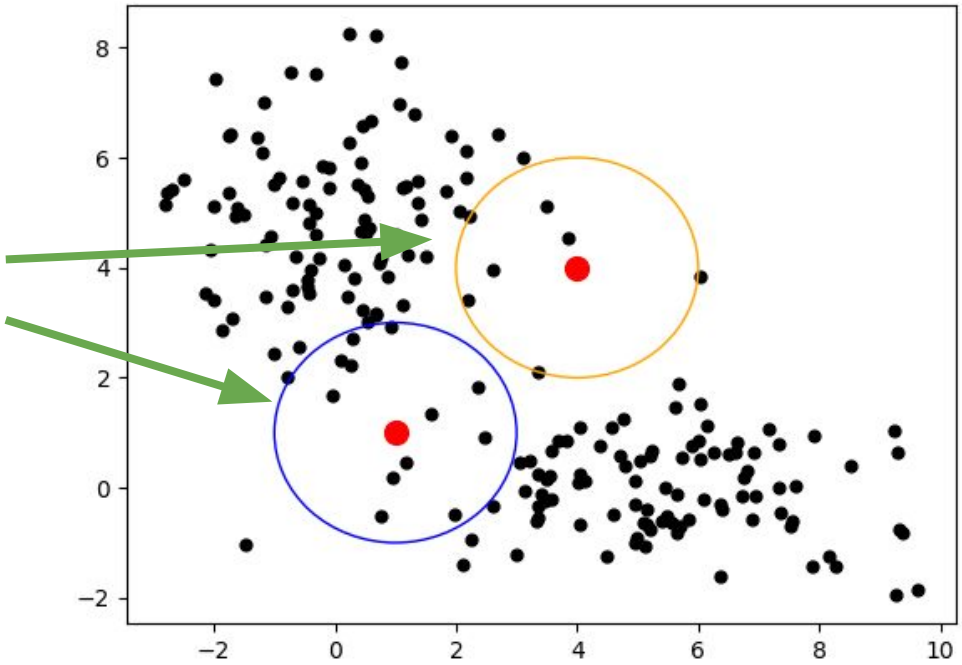
Make initial guesses of parameter values



Gaussian EM

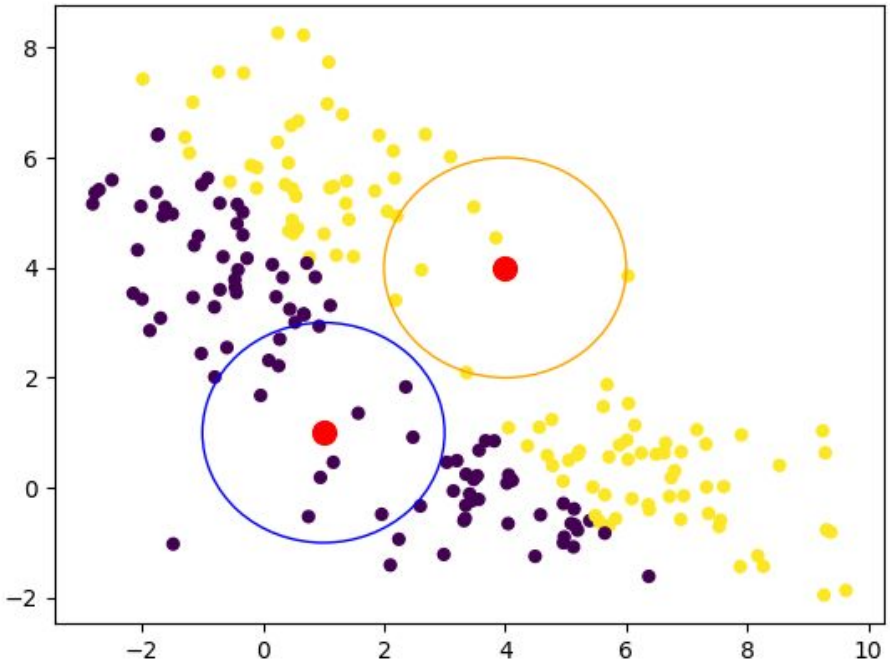
Make initial guesses of parameter values

intentionally
bad guesses!



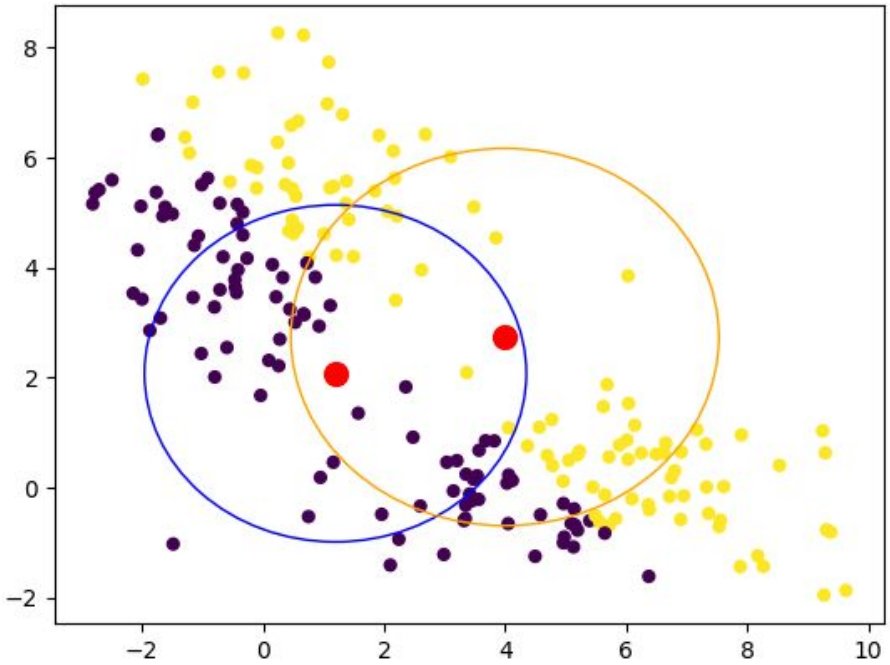
Gaussian EM

Assign each point to its most likely cluster (E-step)



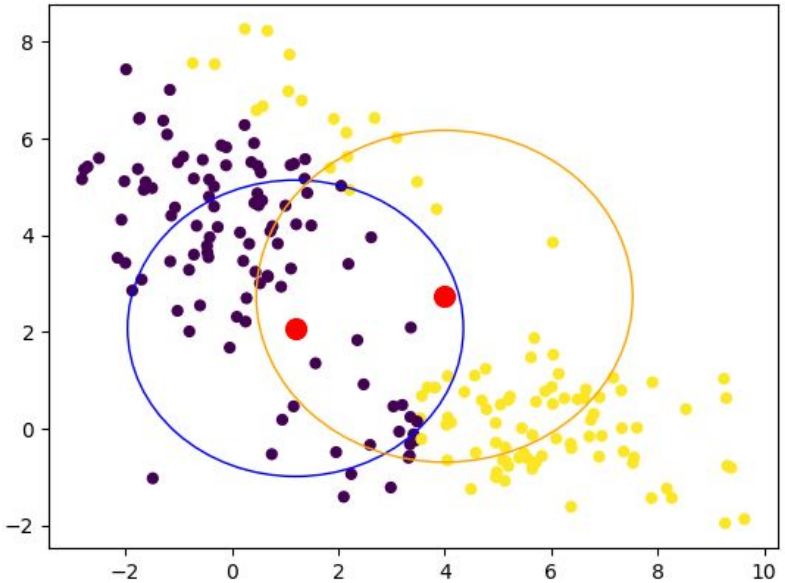
Gaussian EM

Update cluster parameters (M-step)

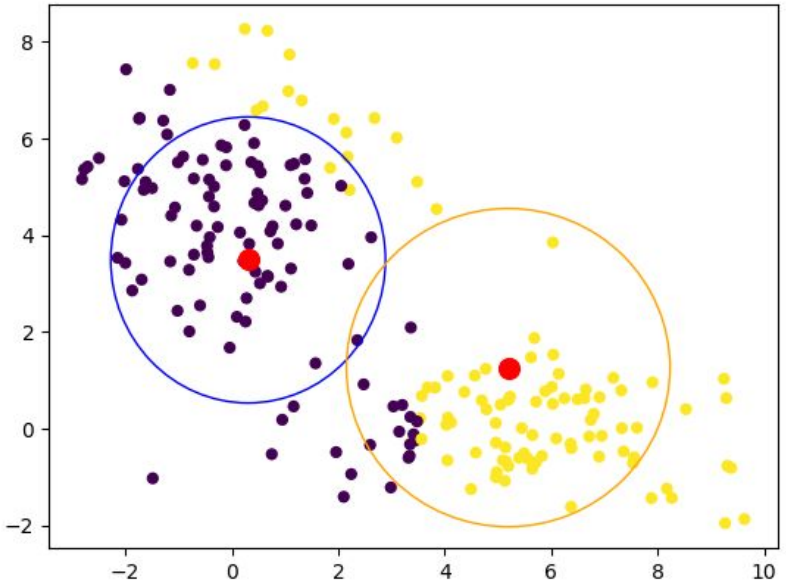


Gaussian EM

Repeat E-Step



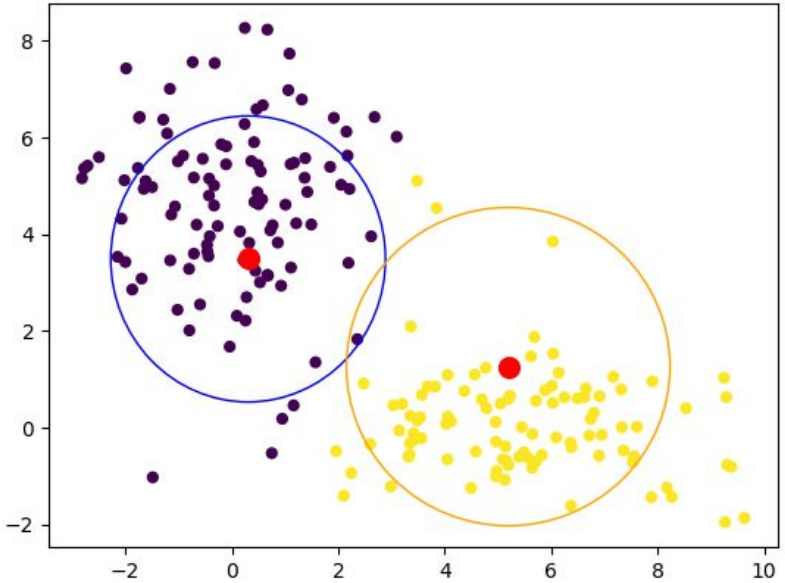
Repeat M-Step



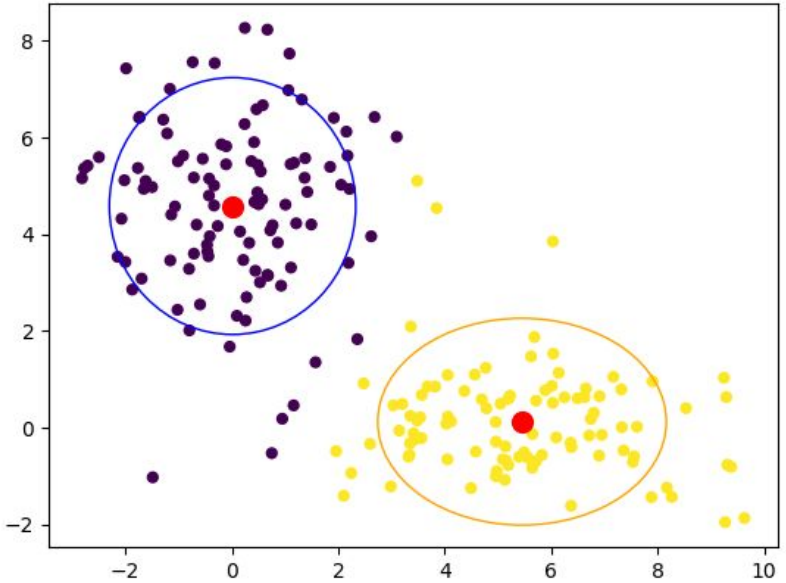
iteration 2

Gaussian EM

Repeat E-Step



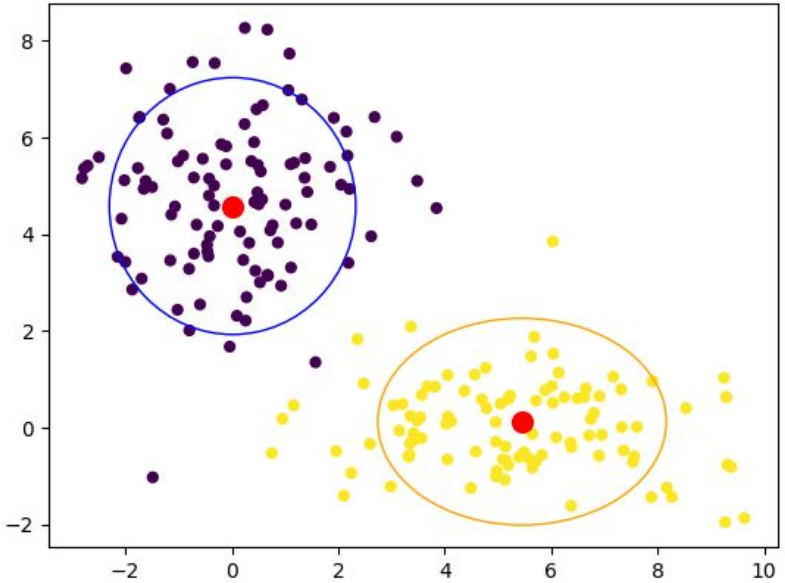
Repeat M-Step



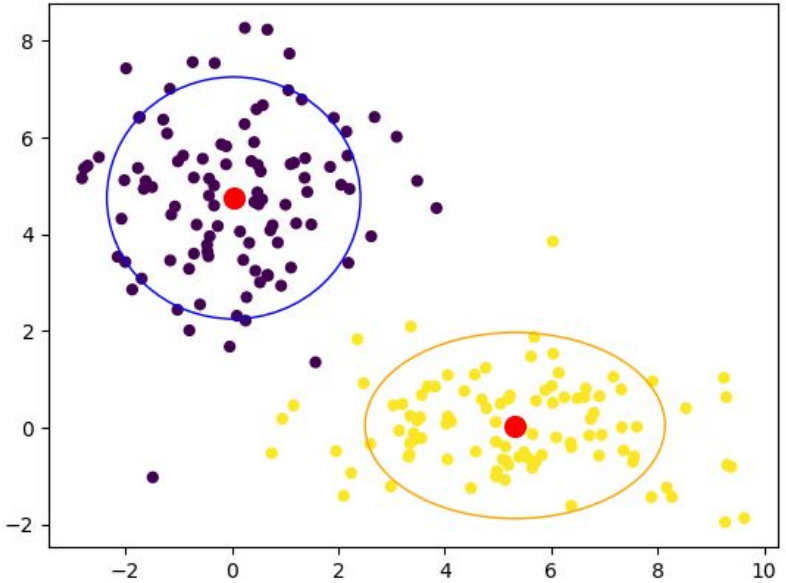
iteration 3

Gaussian EM

Repeat E-Step



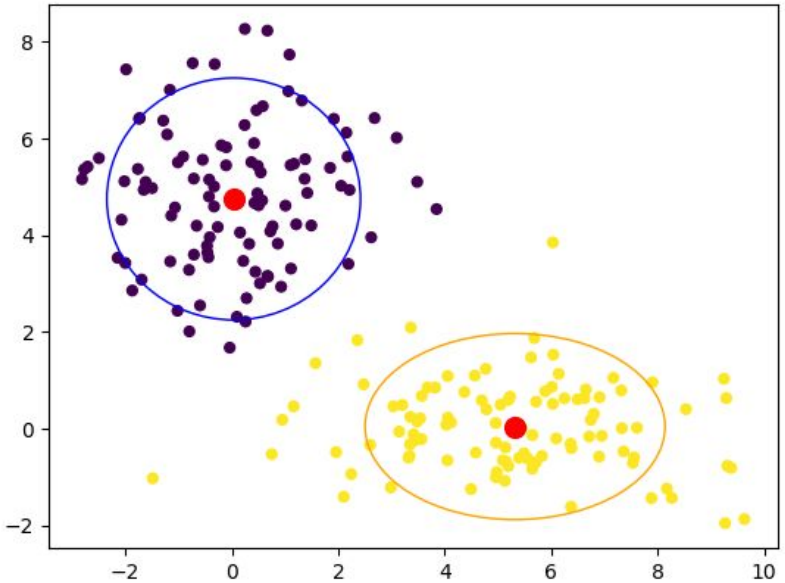
Repeat M-Step



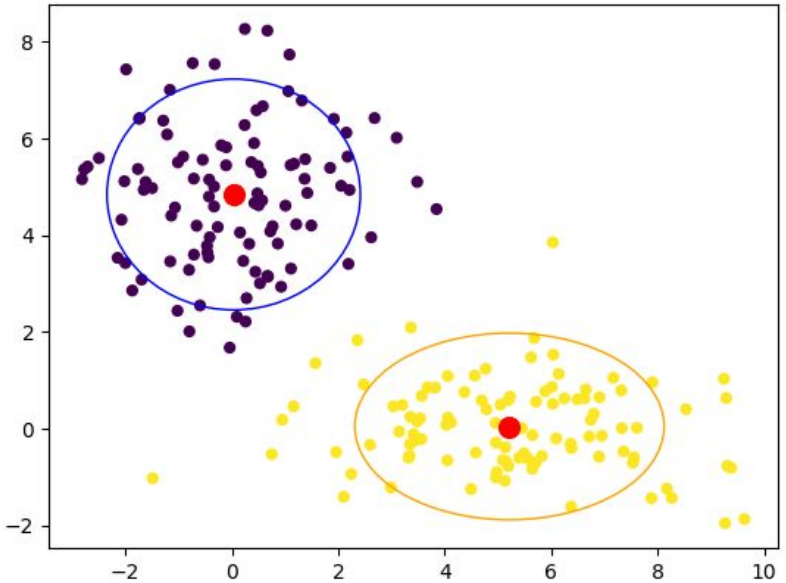
iteration 4

Gaussian EM

Repeat E-Step



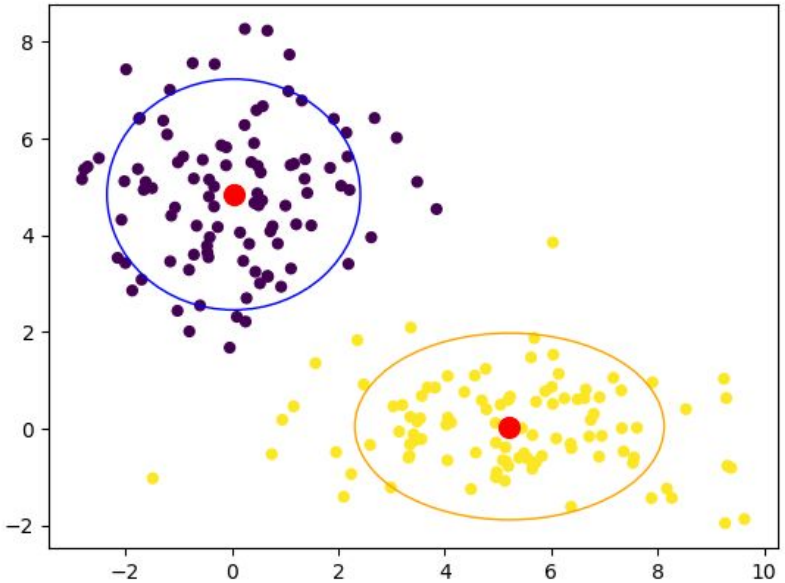
Repeat M-Step



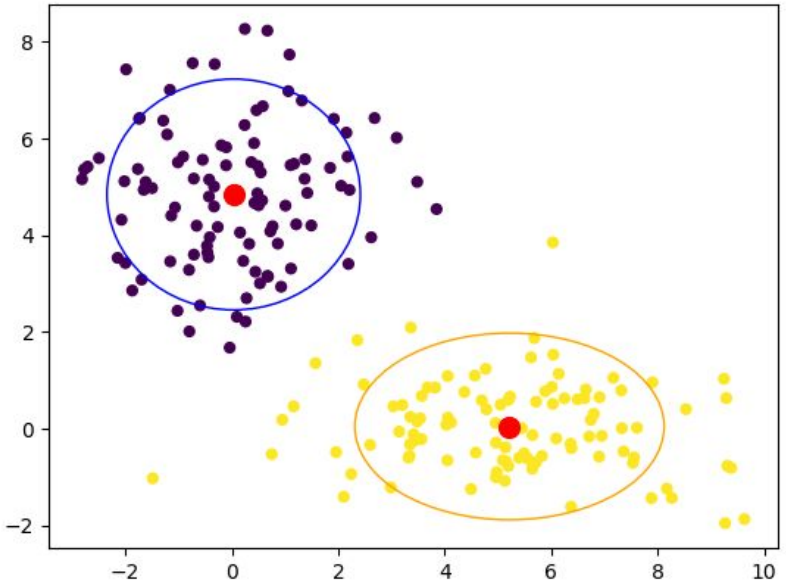
iteration 5

Gaussian EM

Repeat E-Step



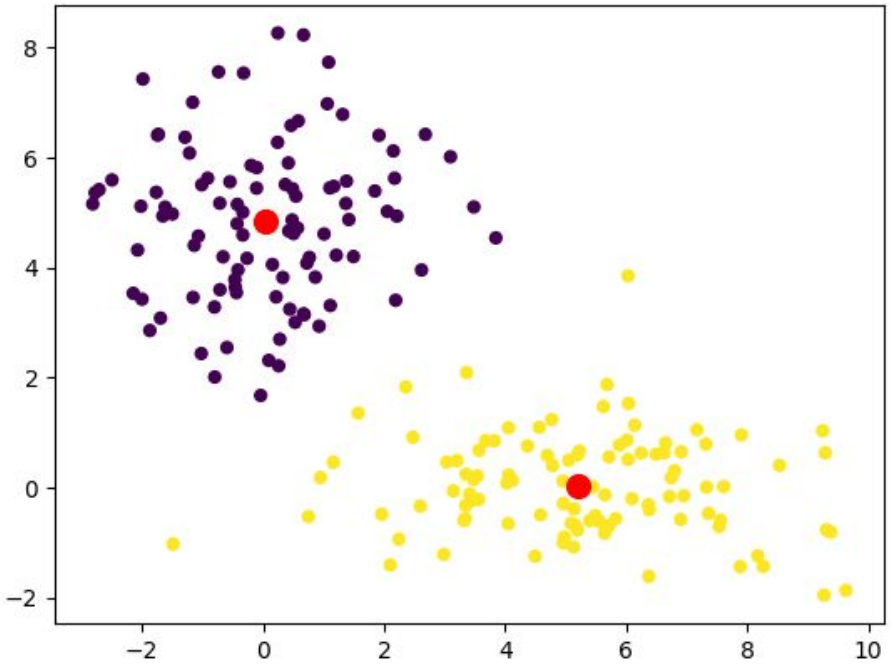
Repeat M-Step



iteration 6

Gaussian EM

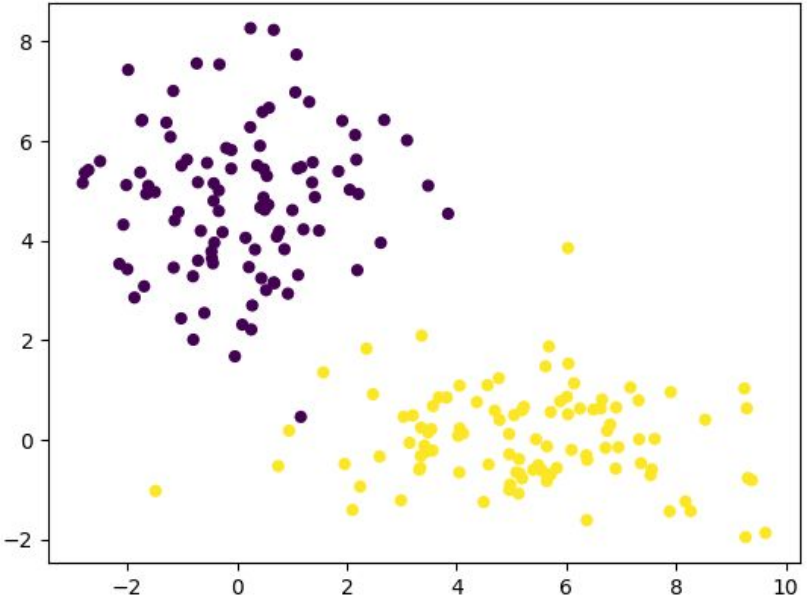
Continue until clusters converge



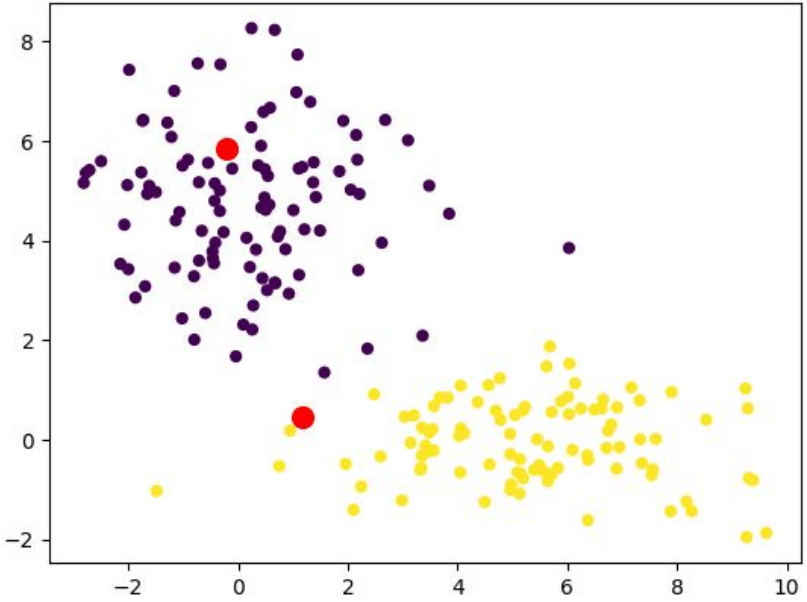
Expectation-Maximization

- Pros
  - Each cluster is a statistical distribution
  - “Soft” assignment with probabilistic uncertainty
  - Robust to poor initial clusters
  - Clusters can be different sizes
  - Very flexible approximation
- Cons
  - Must choose number of clusters in advance
  - Distance function is less flexible
  - Somewhat sensitive to outliers

Comparison

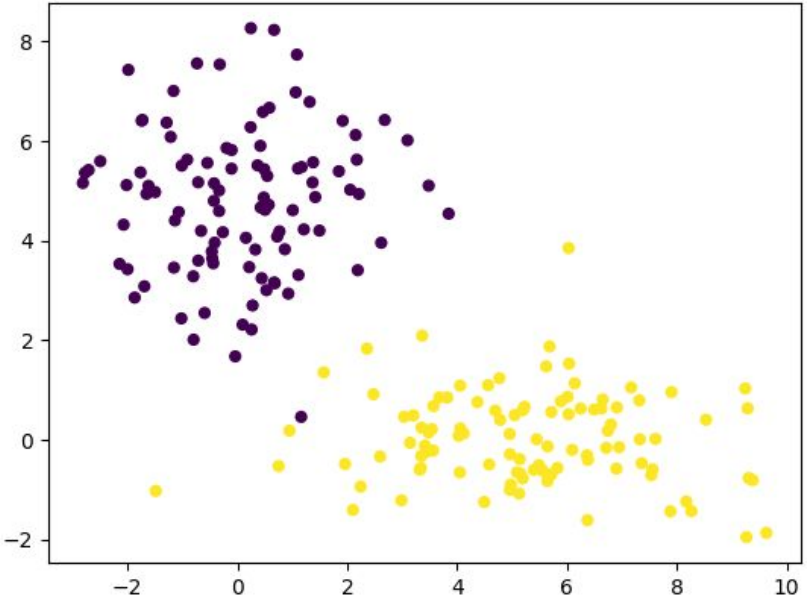


Truth

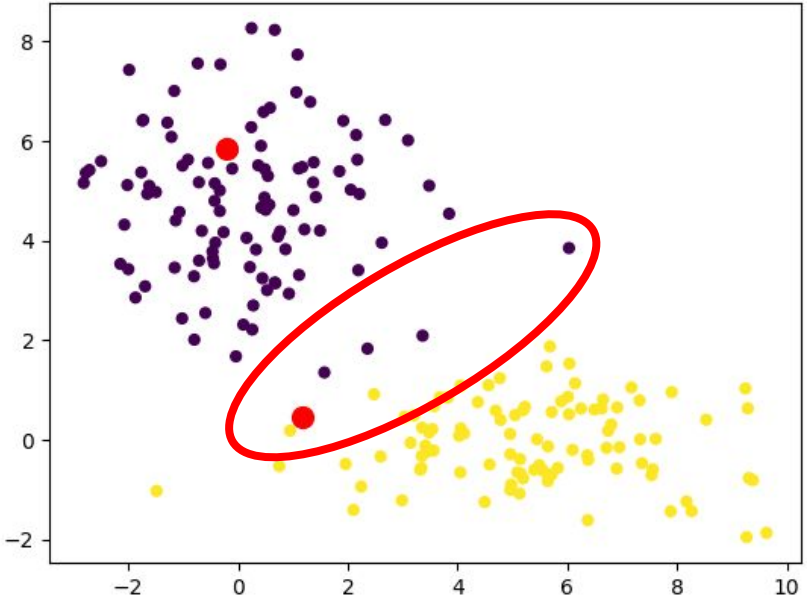


DBSCAN

Comparison

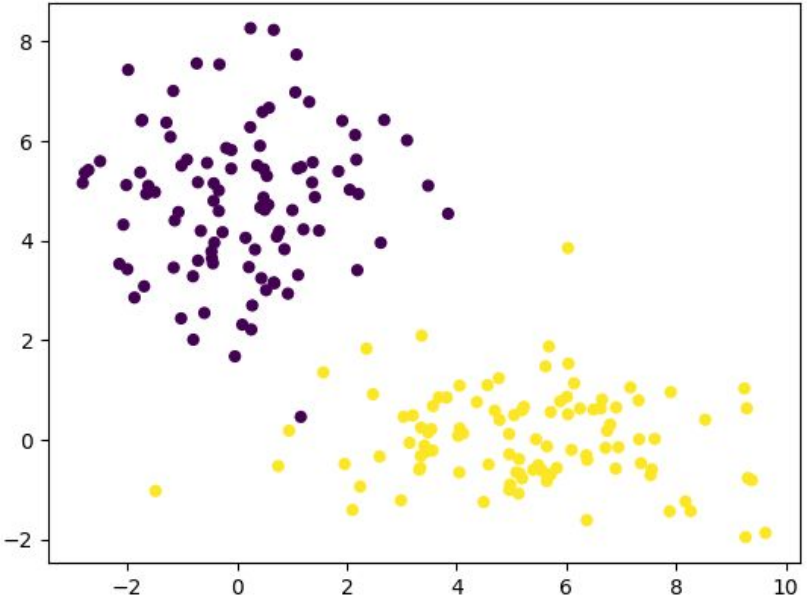


Truth

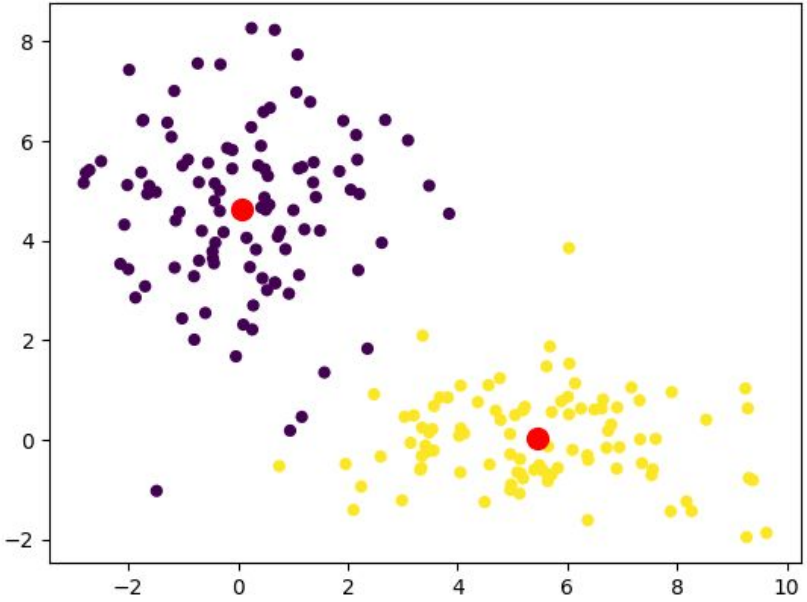


DBSCAN

Comparison

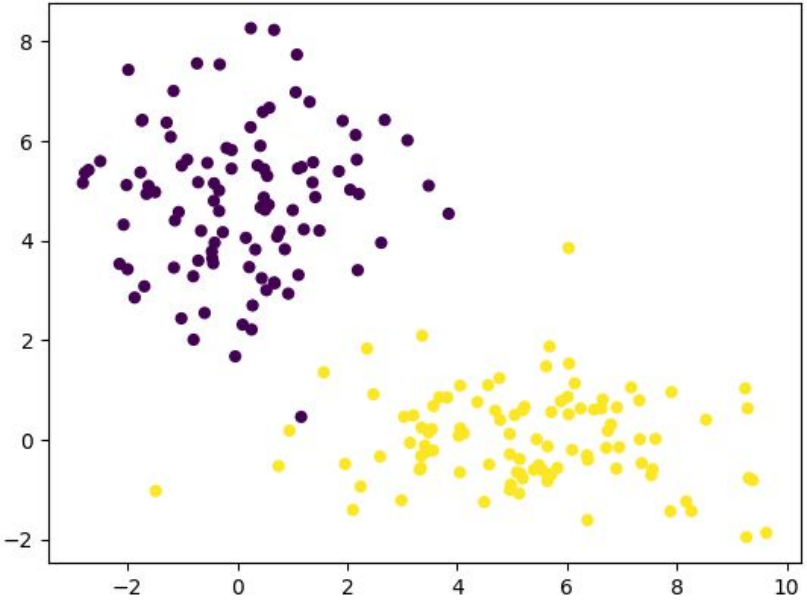


Truth

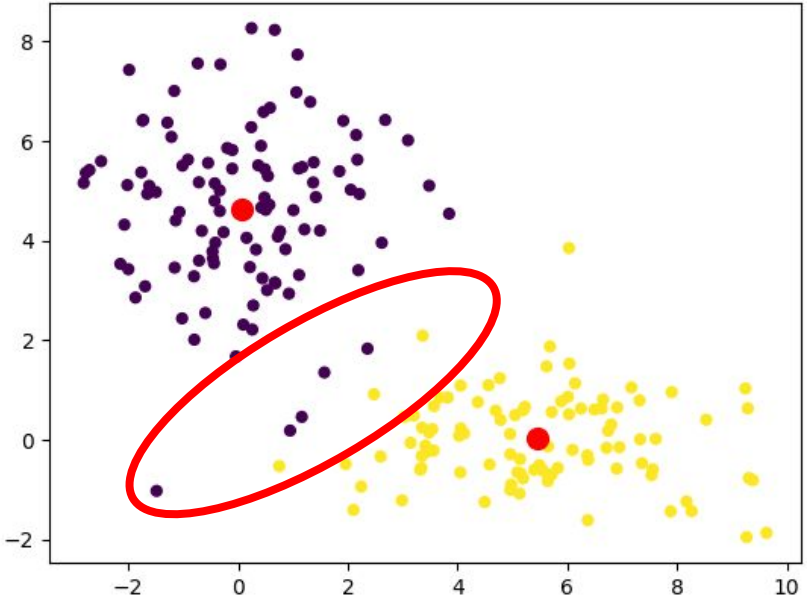


K-Means

Comparison

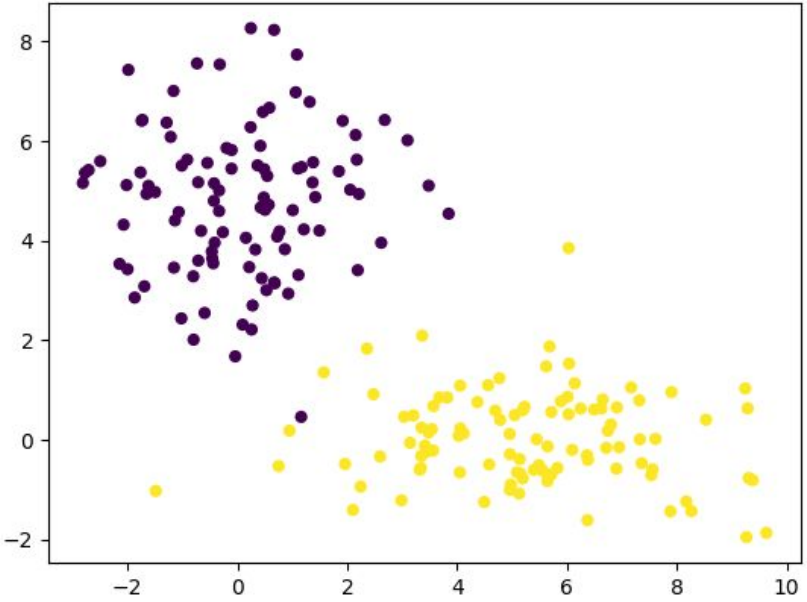


Truth

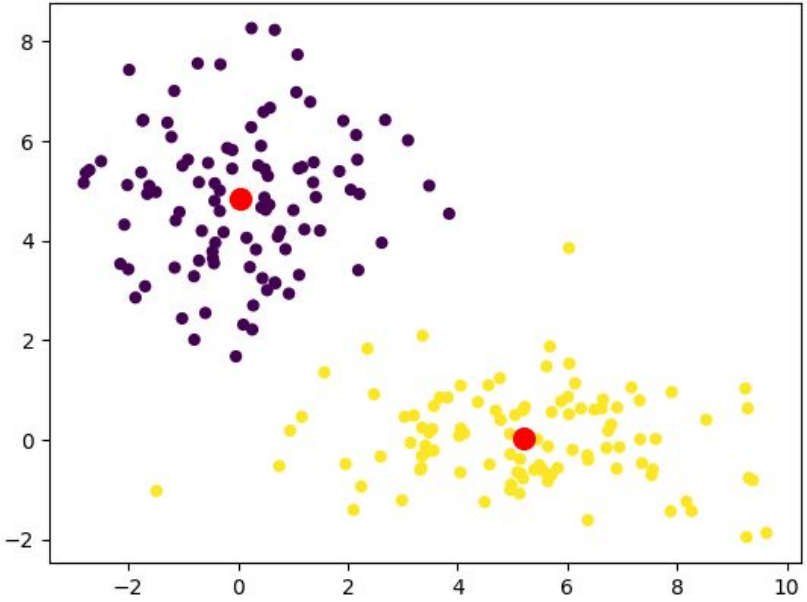


K-Means

Comparison

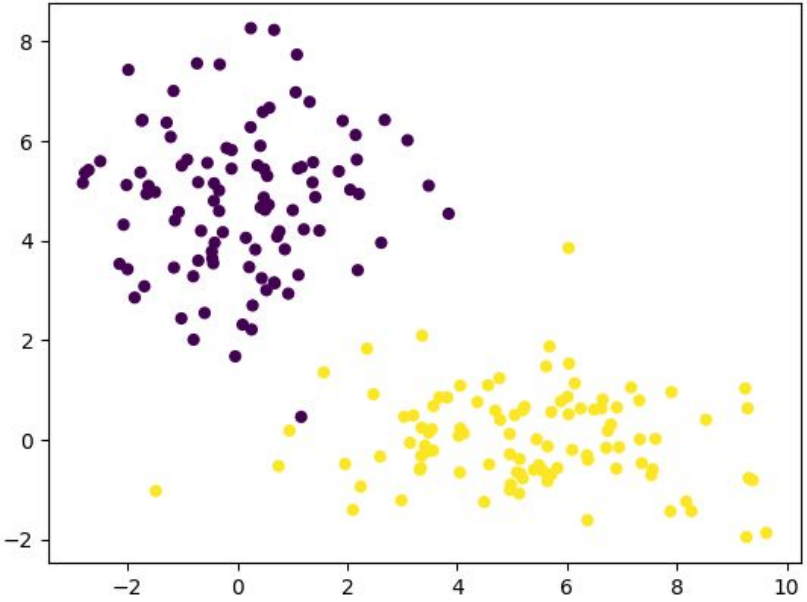


Truth

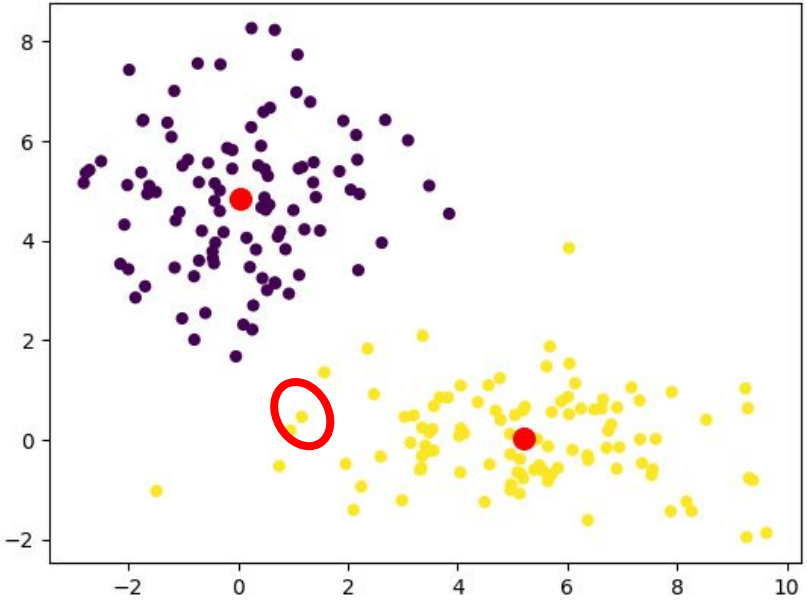


Gaussian EM

Comparison



Truth

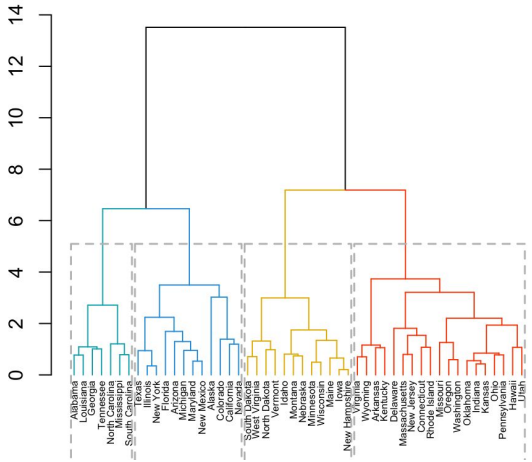


Gaussian EM

Choosing a clustering algorithm

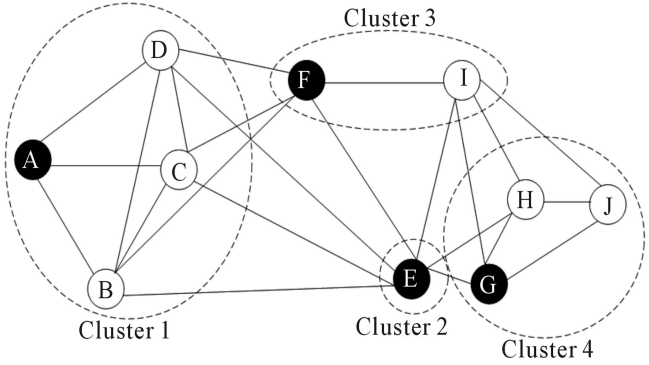
- DBSCAN
  - You have a good sense of what an appropriate ε threshold is
  - You are concerned about outliers
- K-Means
  - You want to get started very quickly
  - You have time to try different values of *k*
- Expectation-Maximization
  - You want to convey uncertainty about your predictions
  - You know the number of clusters or have time to try several values

Other Clustering Approaches



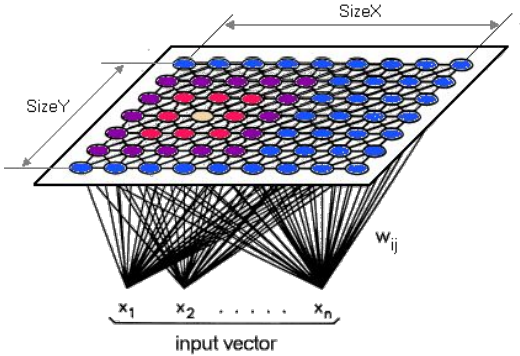
Hierarchical

http://www.sthda.com/sthda/RDoc/figure/clustering/cluster-analysis-in-r-hierarchical-clustering2-1.png



Graph-Based

https://hemberg-lab.github.io/scRNA.seq.course/figures/graph_network.jpg



Neural Model

http://www.pitt.edu/~is2470pb/Spring05/FinalProjects/Group1a/tutorial/kohonen1.gif

Other Clustering Approaches

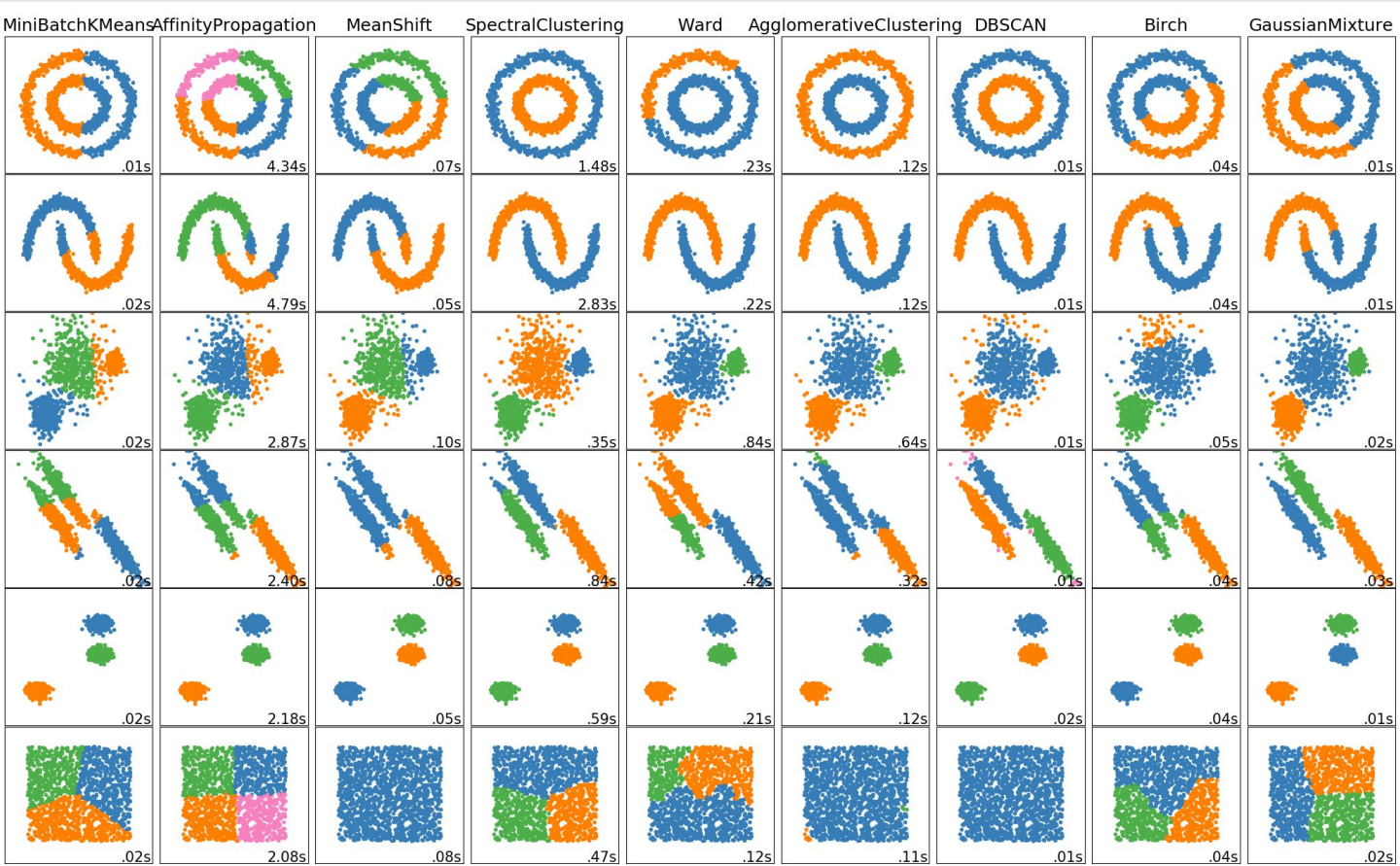
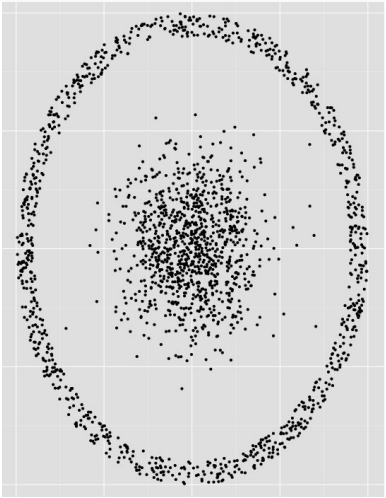
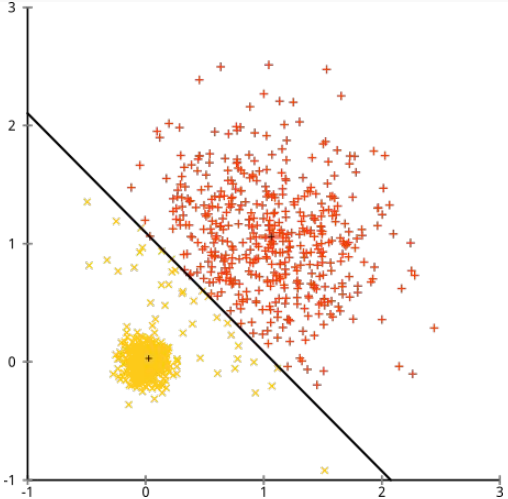


Image: http://scikit-learn.org/stable/modules/clustering.html

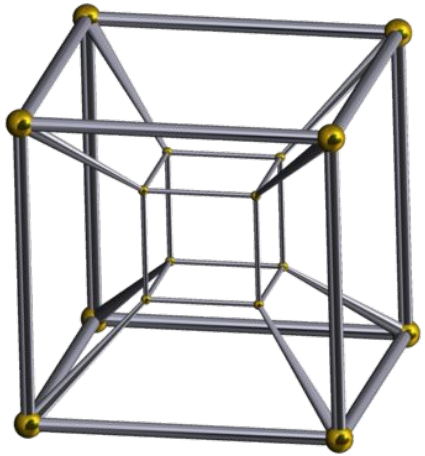
When Clustering Can Fail



non-spherical



different densities



high-dimensional data

	non-spherical	different densities	high-dimensional data
DBSCAN	✔	✘	✘
K-Means	✘	OK	✘
Gaussian EM	✘	✔	OK

http://hameddaily.blogspot.com/2015/03/when-not-to-use-gaussian-mixtures-model.html

https://stats.stackexchange.com/questions/133656/how-to-understand-the-drawbacks-of-k-means

https://upload.wikimedia.org/wikipedia/commons/thumb/a/a2/Schlegel_wireframe_8-cell.png/400px-Schlegel_wireframe_8-cell.png

When to Use Clustering

- When you...
  - suspect your data consists of underlying groups
  - cannot collect labels for classification
  - can define “similar” points via a distance metric

Resources

- Geometric view of clustering: https://shapeofdata.wordpress.com/category/clustering/
- K-Means
  - https://www.datascience.com/blog/k-means-clustering
  - https://www.naftaliharris.com/blog/visualizing-k-means-clustering/
  - Elbow method: https://dataskeptic.com/blog/episodes/2016/the-elbow-method
- DBSCAN
  - https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/
  - http://lineardigressions.com/episodes/2017/11/19/dbscan
- EM
  - http://hameddaily.blogspot.com/2015/03/when-not-to-use-gaussian-mixtures-model.html
  - https://pythonmachinelearning.pro/clustering-with-gaussian-mixture-models/
- Other clustering approaches
  - http://lineardigressions.com/episodes/2016/2/29/t-sne-reduce-your-dimensions-keep-your-clusters
- TF Example: http://playground.tensorflow.org
- Image compression: http://scikit-learn.org/stable/auto_examples/cluster/plot_color_quantization.html

Thanks!

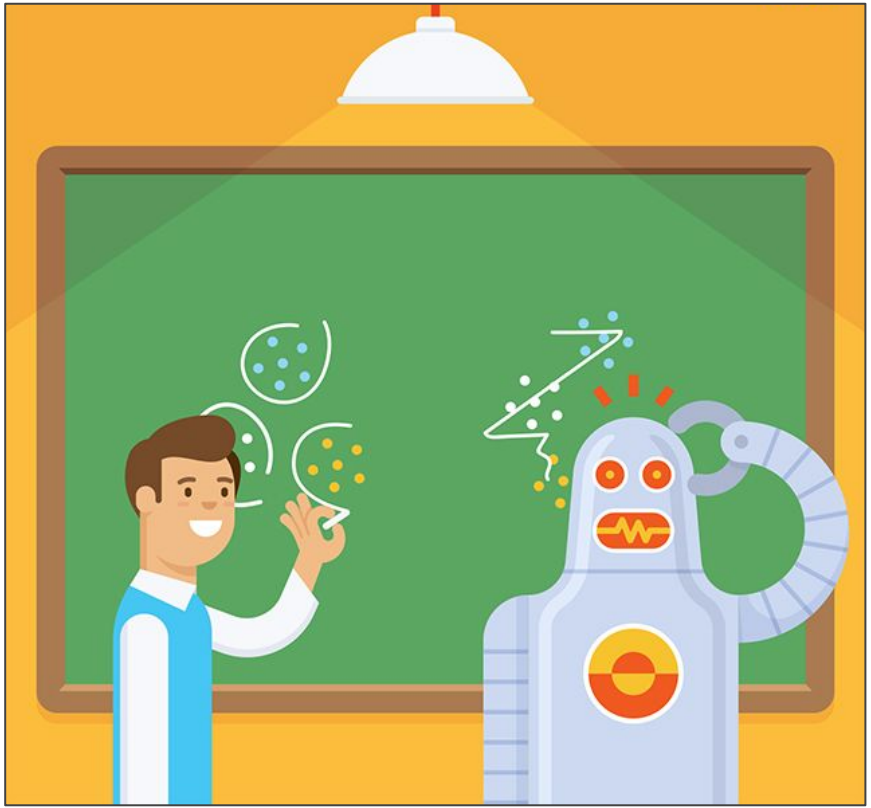


Image: https://toptal.io/

Lab Session

Recap

- Clustering is an unsupervised approach
- DBSCAN
  - Intuitive parameters
  - Difficult to control number of clusters
- K-Means
  - Easy to control number of clusters
  - Sensitive to outliers
- Gaussian EM
  - Probabilistic cluster assignment

Evaluating Clustering Models

- How can we find the “best” parameters?
  - Grid Search
- How can we evaluate cluster quality without labeled data?
  - Silhouette Score
- Code Lab

Grid Search

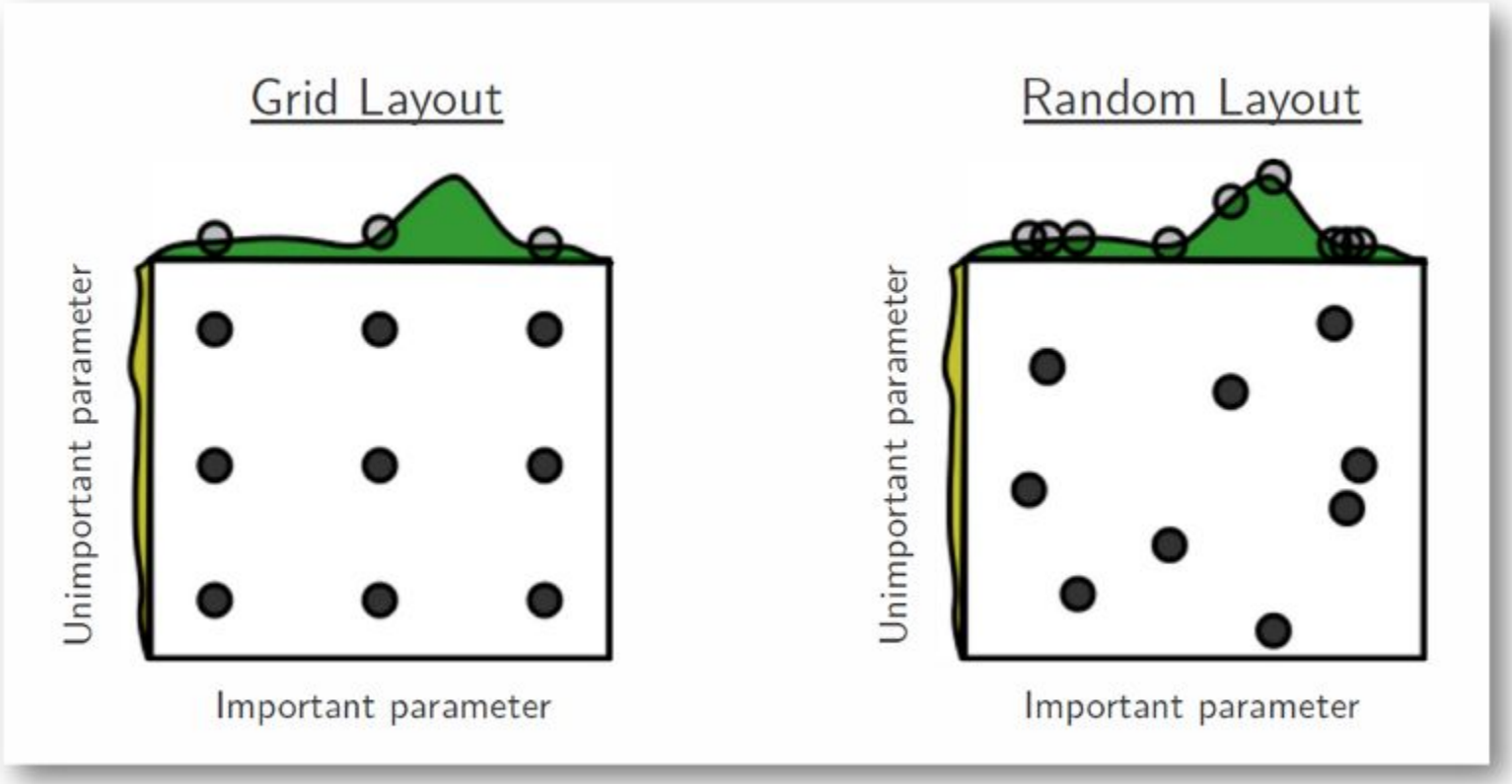
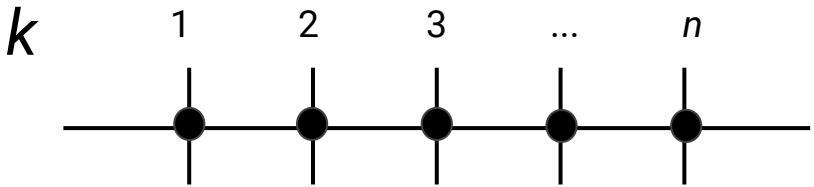
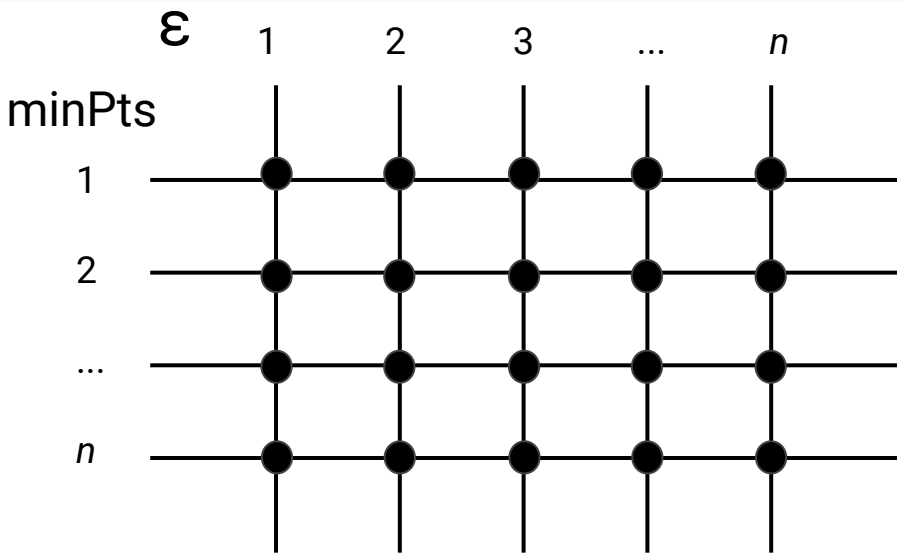


Image: http://jmlr.csail.mit.edu/papers/volume13/bergstra12a/bergstra12a.pdf

Grid Search



K-Means



DBSCAN

Grid Search

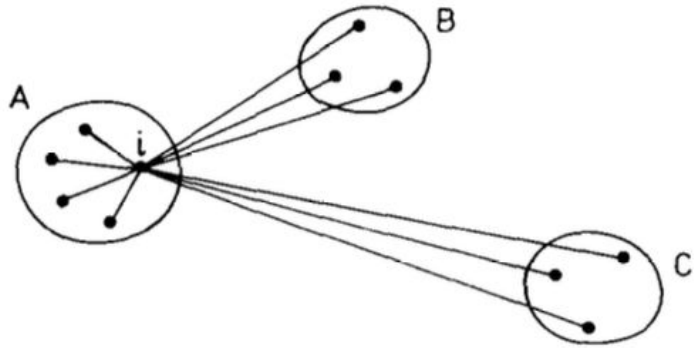
- Pros
  - Very simple
  - Works well with a small number parameters
- Cons
  - Assumes all parameters are equally important
    - Not always true!
    - When this assumption is violated, consider random search

There are many other advanced parameter search methods, too

Silhouette Score

- Revisiting the question “how do I choose *k*?”
- General scoring method for clustering algorithms
  - Works for models other than K-Means too

Silhouette Score



- *a(i)*: average dissimilarity of *i* to all other objects in cluster A
- *d(i, X)*: average dissimilarity between *i* and objects in cluster X
- $b(i) = \min_{X \neq A} d(i, X)$

Image: doi:10.1016/0377-0427(87)90125-7

Silhouette Score

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

- *a*: mean distance between a sample & all other points in the **same cluster**
- *b*: mean distance between a sample & all other points in the **next nearest cluster**

Silhouette Score

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

$$-1 \leq s(i) \leq 1$$

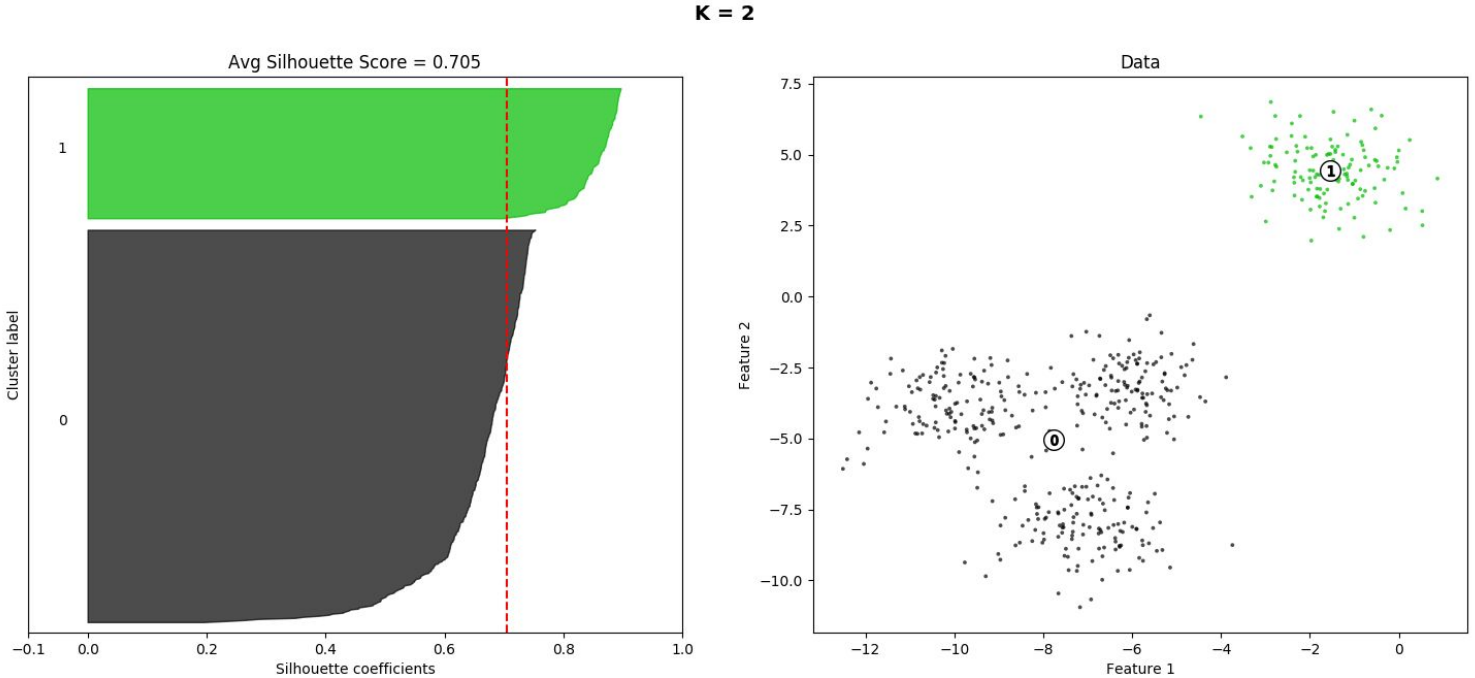
- Only defined for n>1 and k>1

Silhouette Score

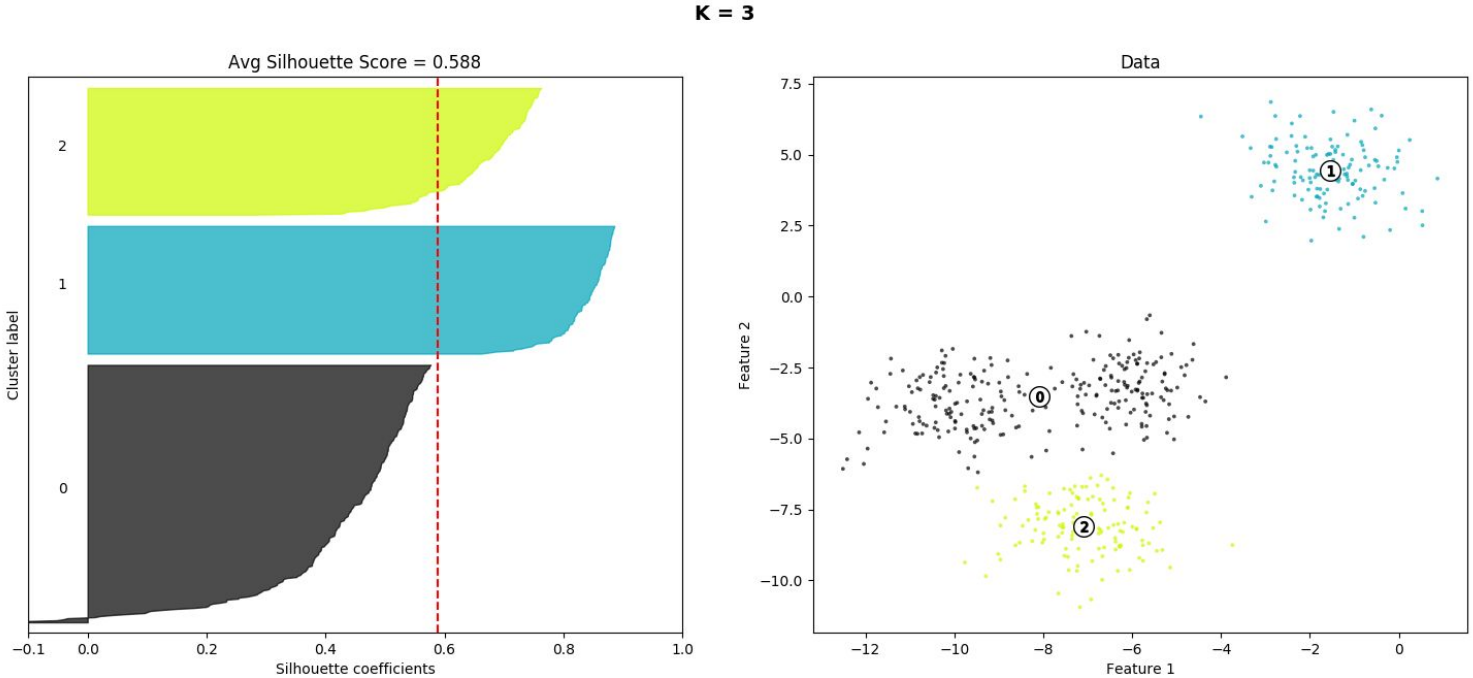
$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

- Close to 1: $a \ll b$
  - Dissimilarity *within* clusters $\ll$ smallest dissimilarity *between* clusters
  - Objects are well-clustered
- Close to 0: $a \cong b$
  - This object could belong to either of two clusters without affecting quality
- Close to -1: $b \ll a$
  - This object is very different from the others in its cluster
  - Likely assigned to the wrong cluster

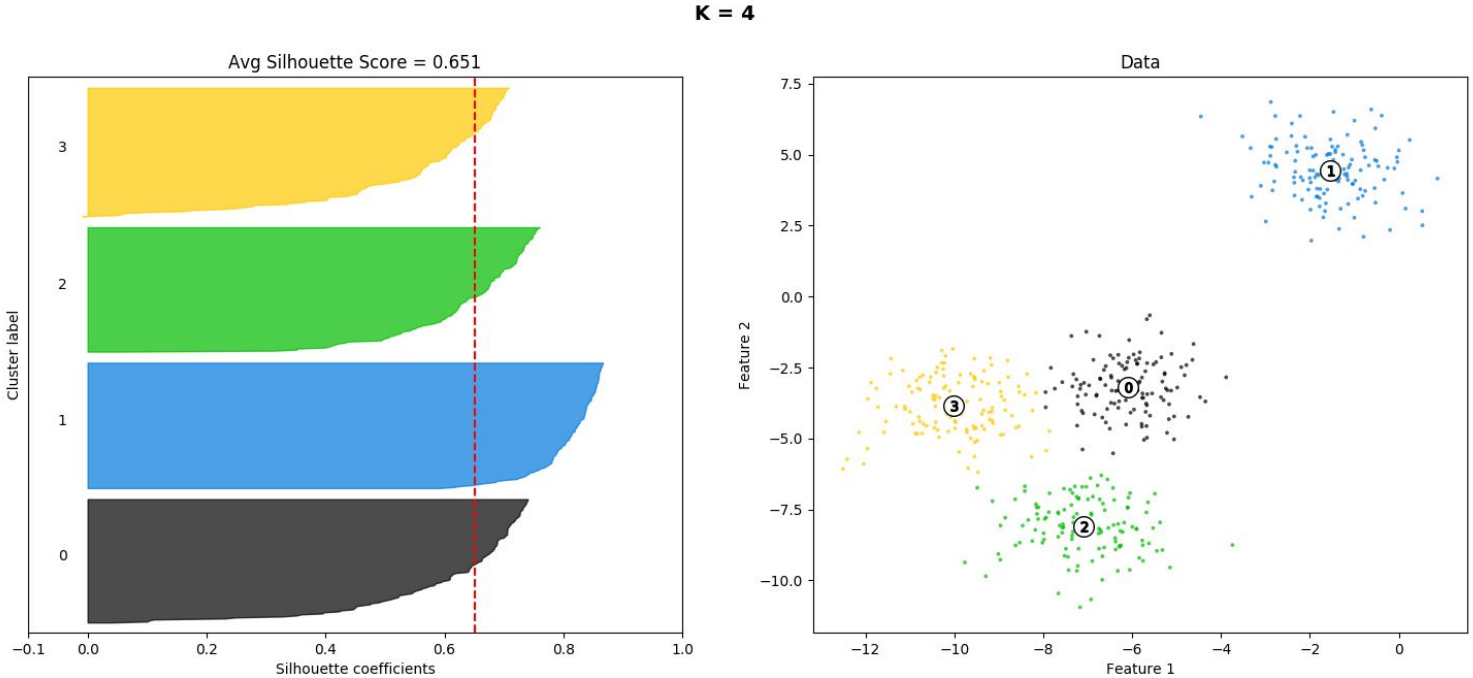
Silhouette Score



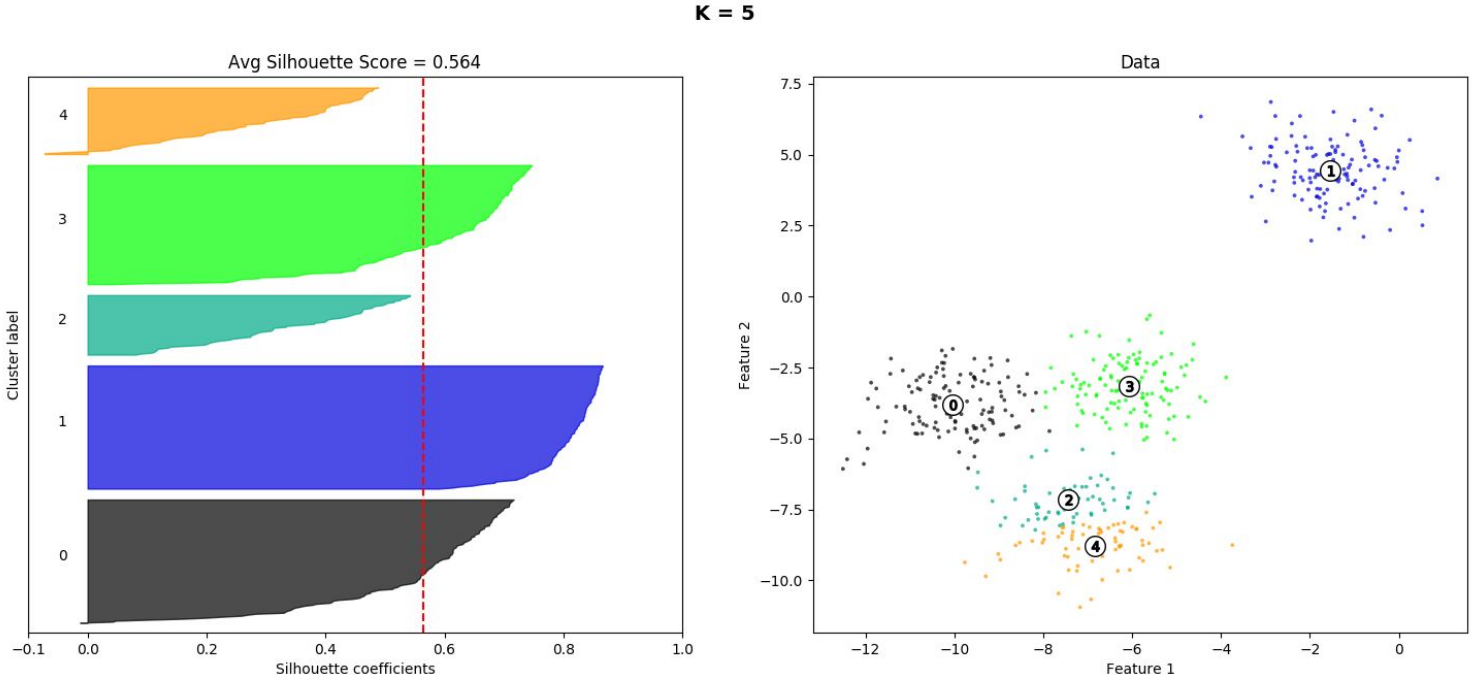
Silhouette Score



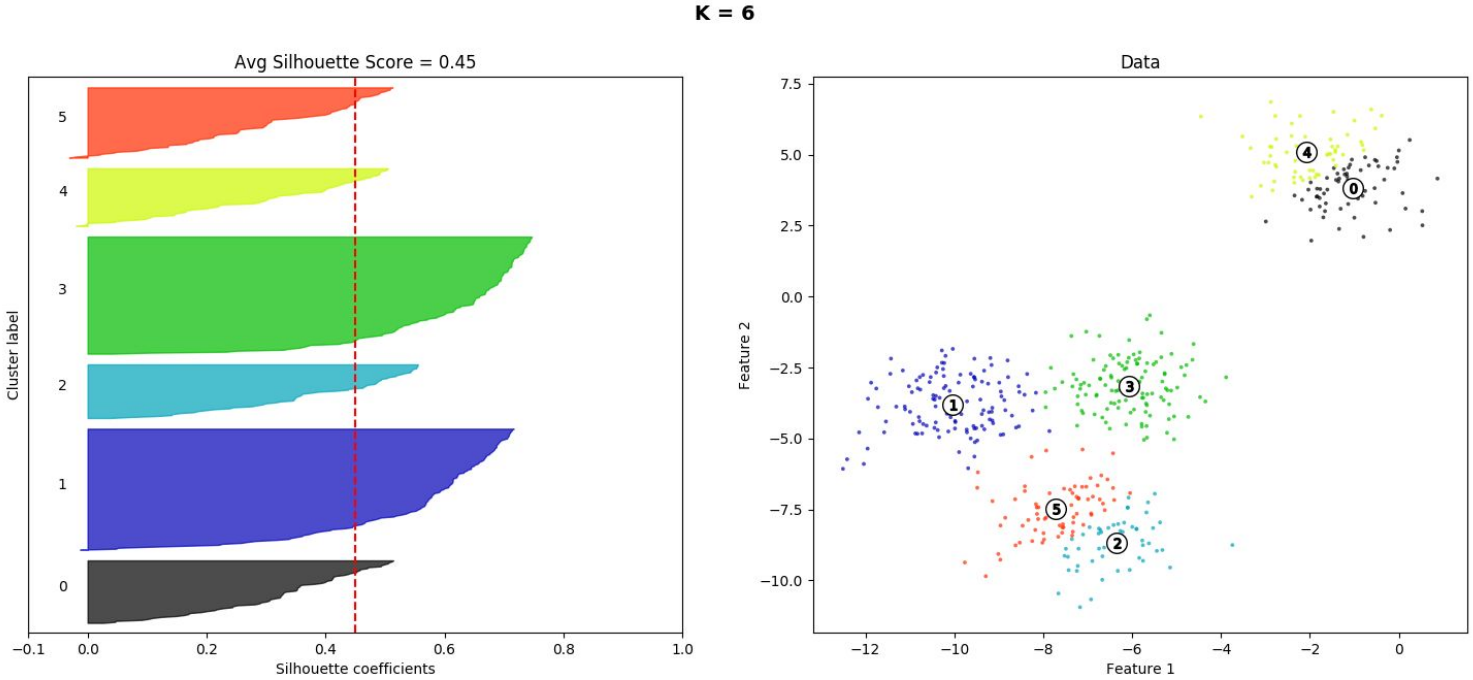
Silhouette Score



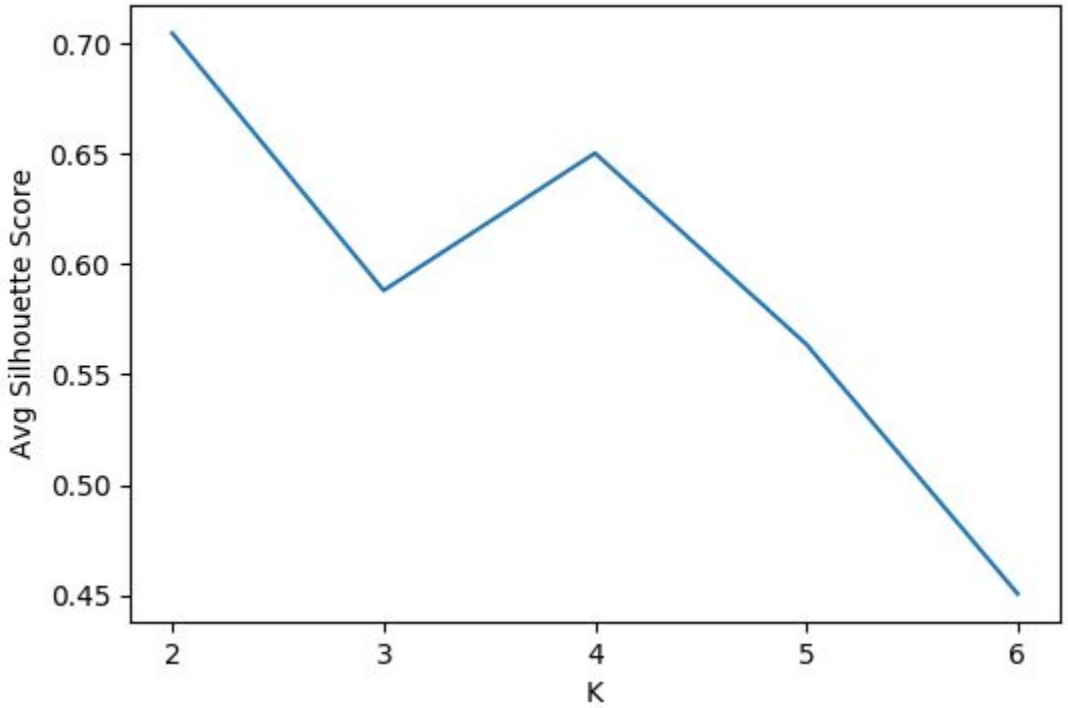
Silhouette Score



Silhouette Score



Silhouette Score



Silhouette Score

- Pros
  - Simple to evaluate and compare
  - Allows comparison of different clustering methods
  - Tends to correlate with qualitative judgment of “good” clusters
- Cons
  - Can be sensitive to data scaling
    - Scale your features to equal “importance”
  - Average silhouette score can mask differences between clusters
  - Biased toward many small clusters

Many other cluster quality metrics exist, too

Resources

- Parameter search
  - https://medium.com/rants-on-machine-learning/smarter-parameter-sweeps-or-why-grid-search-is-plain-stupid-c17d97a0e881
  - https://blog.sigopt.com/posts/evaluating-hyperparameter-optimization-strategies
- Silhouette score
  - http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html#sphx-glr-auto-examples-cluster-plot-kmeans-silhouette-analysis-py
  - http://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient
  - https://kapilddatascience.wordpress.com/2015/12/08/k-mean-clustering-using-silhouette-analysis-with-example-part-3/
  - https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set