95-891: Introduction to Artificial Intelligence

Session 5: Unsupervised Learning: From k-means to Gaussian mixture models

David Steier

steier@andrew.cmu.edu

September 9, 2025

Agenda

- Knowledge Check
- K-means clustering
- Principal Component Analysis
- Gaussian Mixture Models
- Expectation Maximization
- Appendix: Semi-supervised learning

Knowledge Check
(aka “Practice Questions for the Quiz”)

- Why did the pace of AI progress increase in the last 15 years or so?
- What’s the difference between the regression and clustering problem types?
- True or False: Regularization reduces variance. Explain your answer
- True or False: A confusion matrix can be used to evaluate both classification and regression/prediction models. Explain your answer.

Supervised vs. Unsupervised Learning

Supervised Learning

- Modelling the dependency of an output (dependent or *target*) variable based on various independent input variables
- **Examples**
  - Logistic regression
  - Decision trees
  - Support Vector Machines

Unsupervised Learning

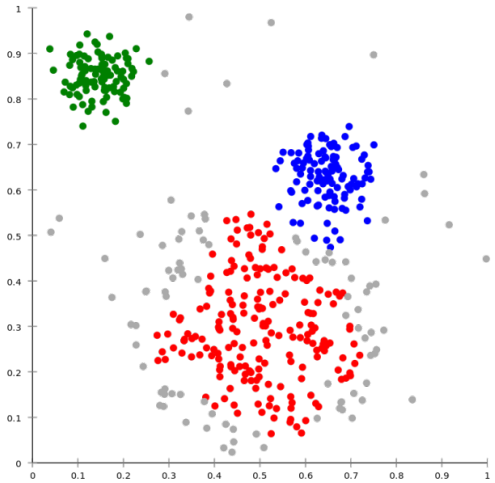
- Identification of patterns and regularities in the data such as groups of entities with similar characteristics or typical correlations without a target variable
- **Examples**
  - Clustering
  - Association rules
  - Dimensionality reduction

Clustering: What Is It?

Clustering: the process of finding a "natural" partition of a dataset

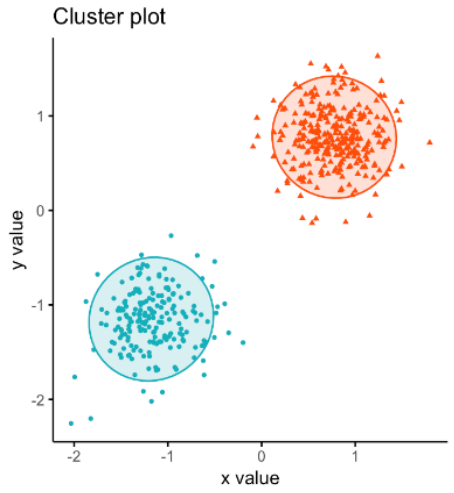
- A commonly-used form of *unsupervised* learning
- There is no single notion of a cluster - there are many clustering algorithms
- Clustering is usually an iterative, judgment-intensive activity

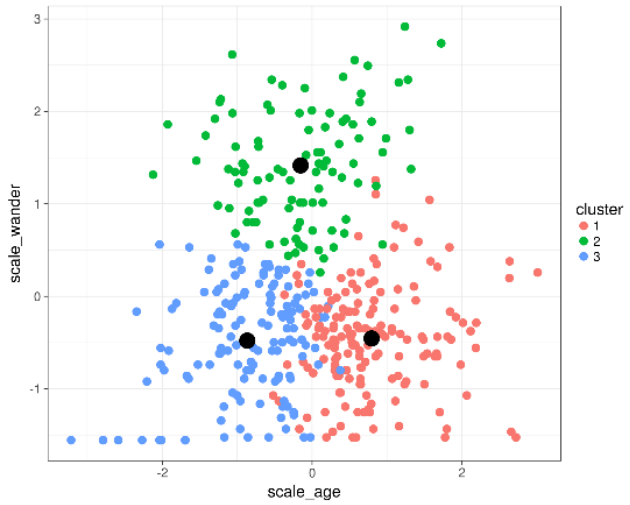
  - A clustering algorithm partitions a dataset into "natural" groups based on a set of variables presented for consideration.
  - Each variable corresponds to a dimension of the data-space that will be partitioned
  - "Distance" in this data space is interpreted as "similarity"
  - Clusters are derived in such a way that items in one cluster are similar to one another; dissimilar from items in other clusters.

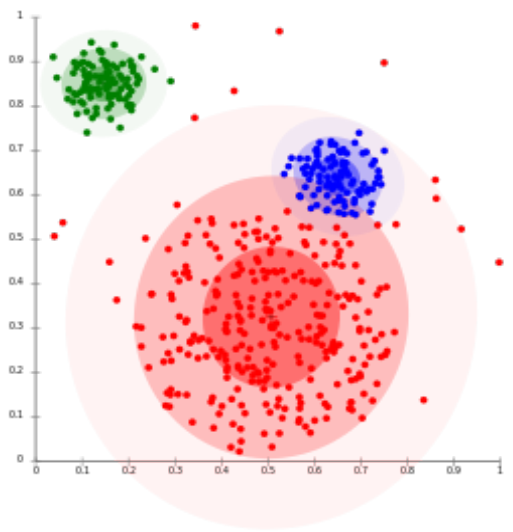


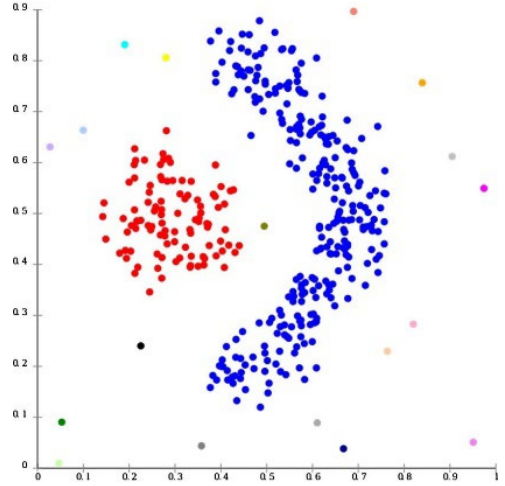
Cluster Properties

- Clusters may have different sizes, shapes, densities
- Clusters may form a hierarchy
- Clusters may be overlapping or disjoint









Clustering Applications

Methods

- K-means
- Hierarchical
- DBScan

Examples

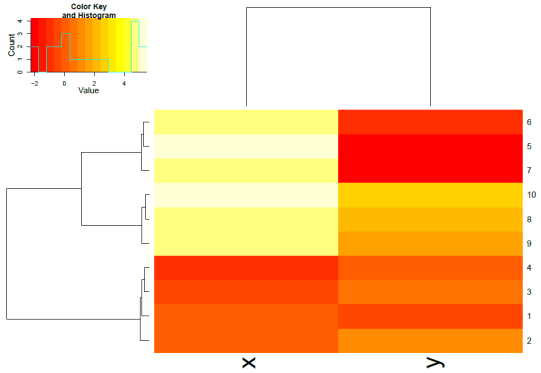
- Customer segmentation
- Molecule search
- Anomaly detection

- Find “natural” clusters and desc
  - Data understanding
- Find useful and suitable groups
  - Data Class Identification
- Find representatives for homogenous groups
  - Data Reduction
- Find unusual data objects
  - Outlier Detection
- Find random perturbations of the data
  - Noise Detection

Types of Clustering Approaches

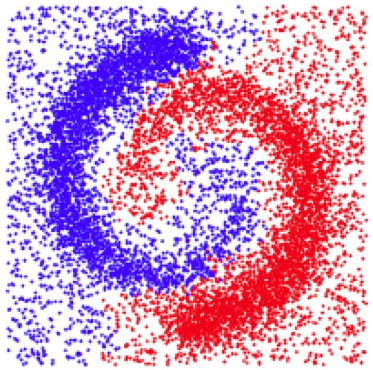
Linkage Based

e.g. Hierarchical Clustering



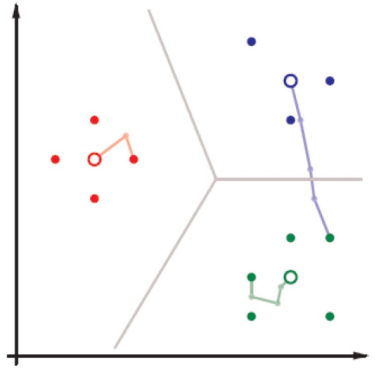
Density based Clustering

e.g. DBSCAN



Clustering by Partitioning

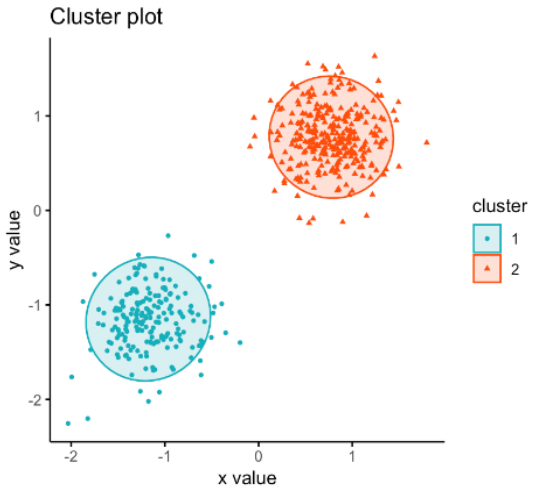
e.g. k-Means



Clustering by Partitioning

Goal:

A (disjoint) partitioning into k clusters with minimal costs



- Local optimization method:
  - choose k initial cluster representatives
  - optimize these representatives iteratively
  - assign each object to its most similar cluster representative
- Types of cluster representatives:
  - Mean of a cluster (*construction of central points*)
  - Median of a cluster (*selection of representative points*)
  - Probability density function of a cluster (*expectation maximization*)

k-Means Clustering

Partition *n* observations into *k* clusters.

3 steps:

1. **Initialization** – *k* initial "means" (centroids) are generated at random
2. **Assignment** – *k* clusters are created by associating each observation with the nearest centroid
3. **Update** – The centroid of each clusters becomes its new mean

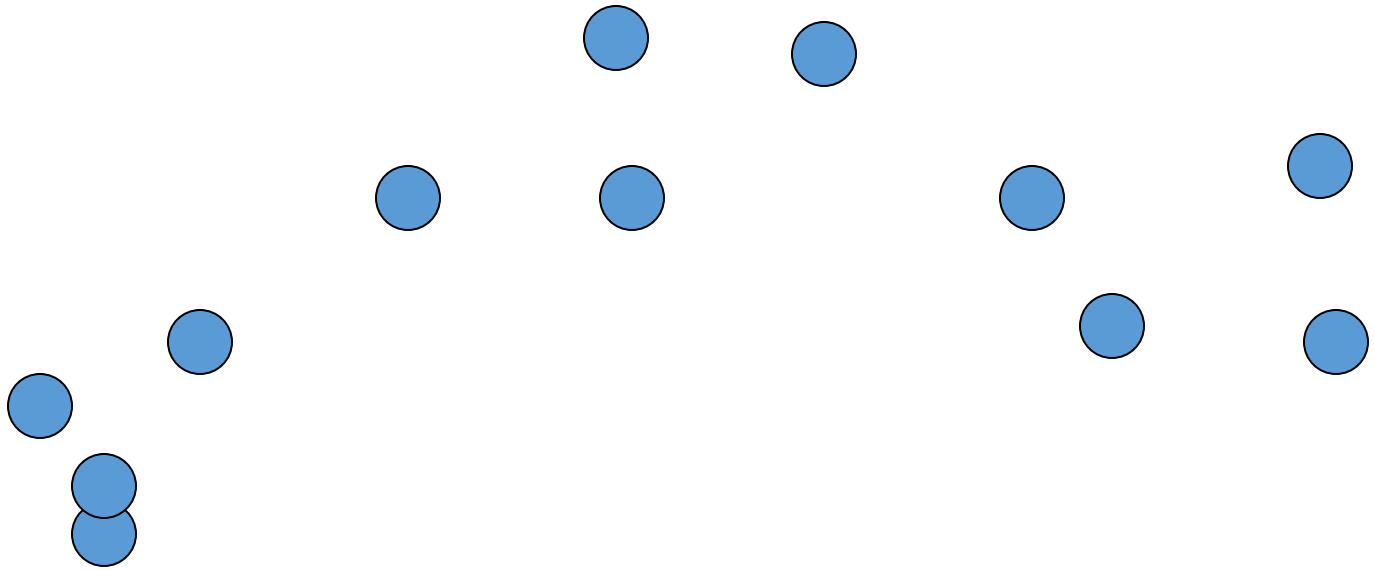
Assignment and Update are repeated iteratively until convergence

The end result is that the sum of squared errors is minimized between points and their respective centroids

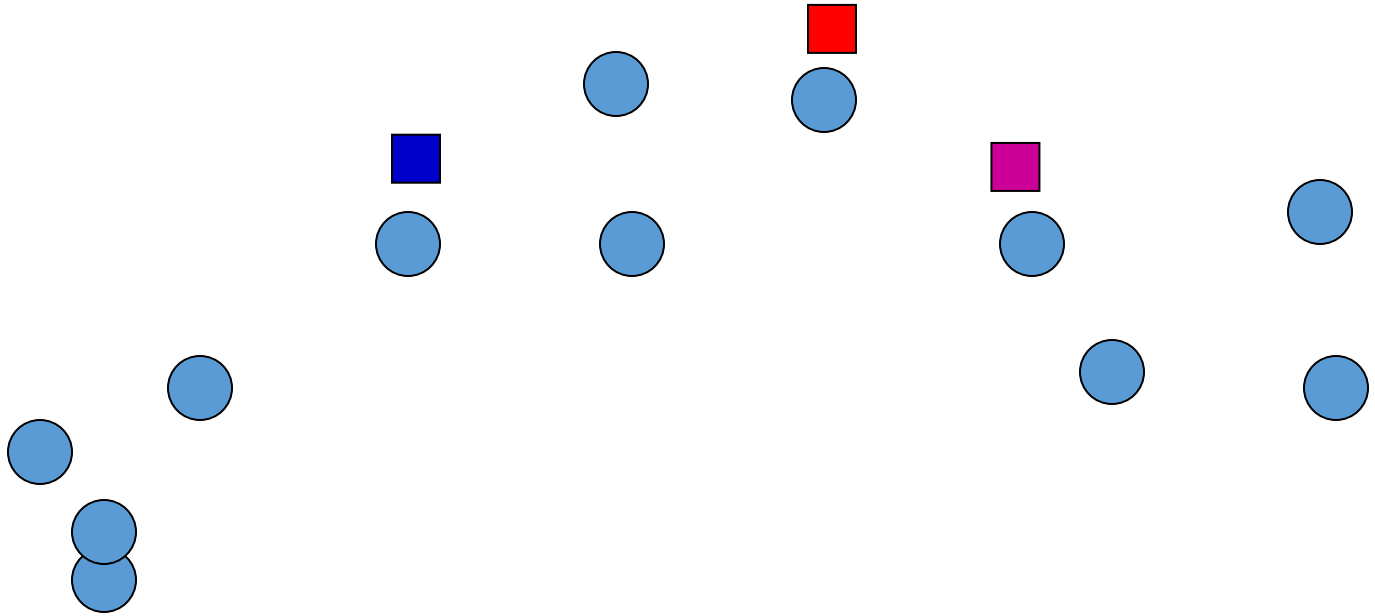
http://benalexkeen.com/k-means-clustering-in-python/

k-Means: An Example

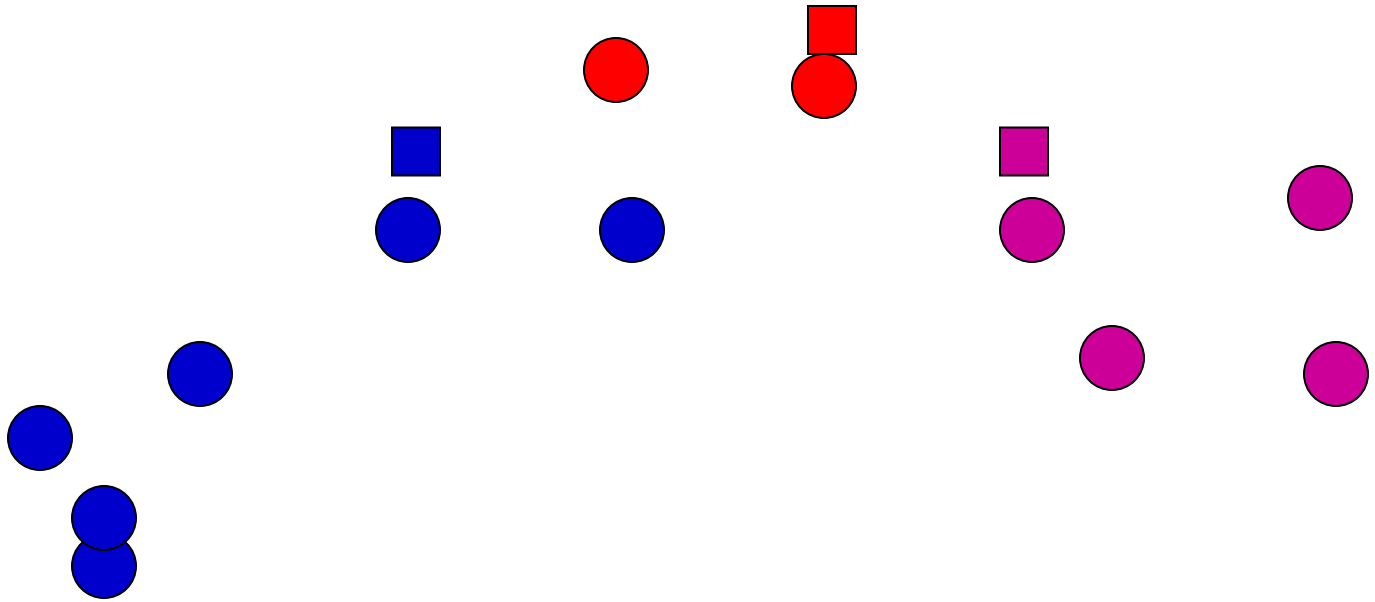
From www.cs.pomona.edu/~dkauchak/classes/f13/cs451-f13/.../lecture31-kmeans.pptx



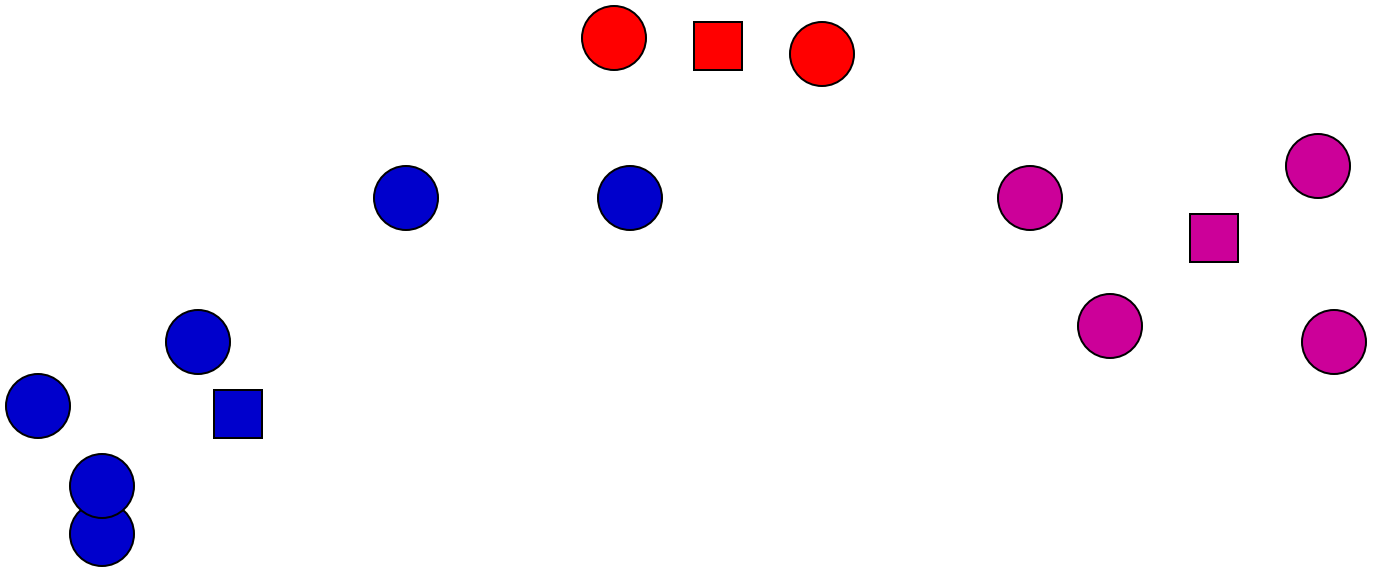
k-Means: Initialize Centers Randomly



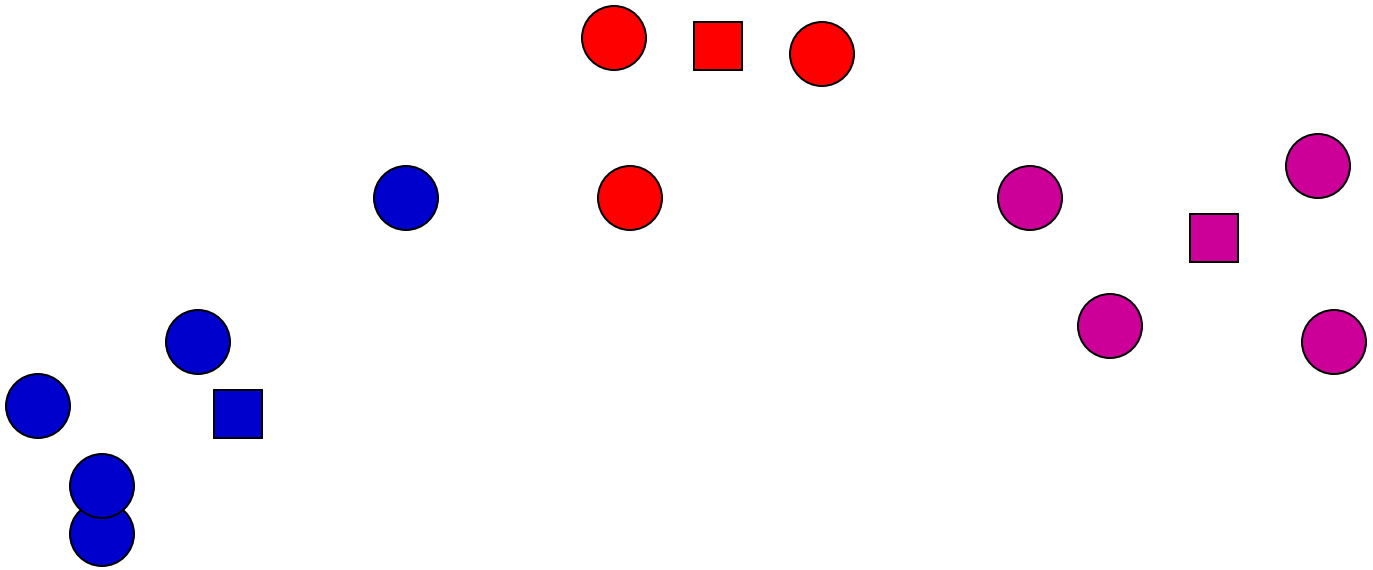
k-Means: Assign Points to Nearest Center



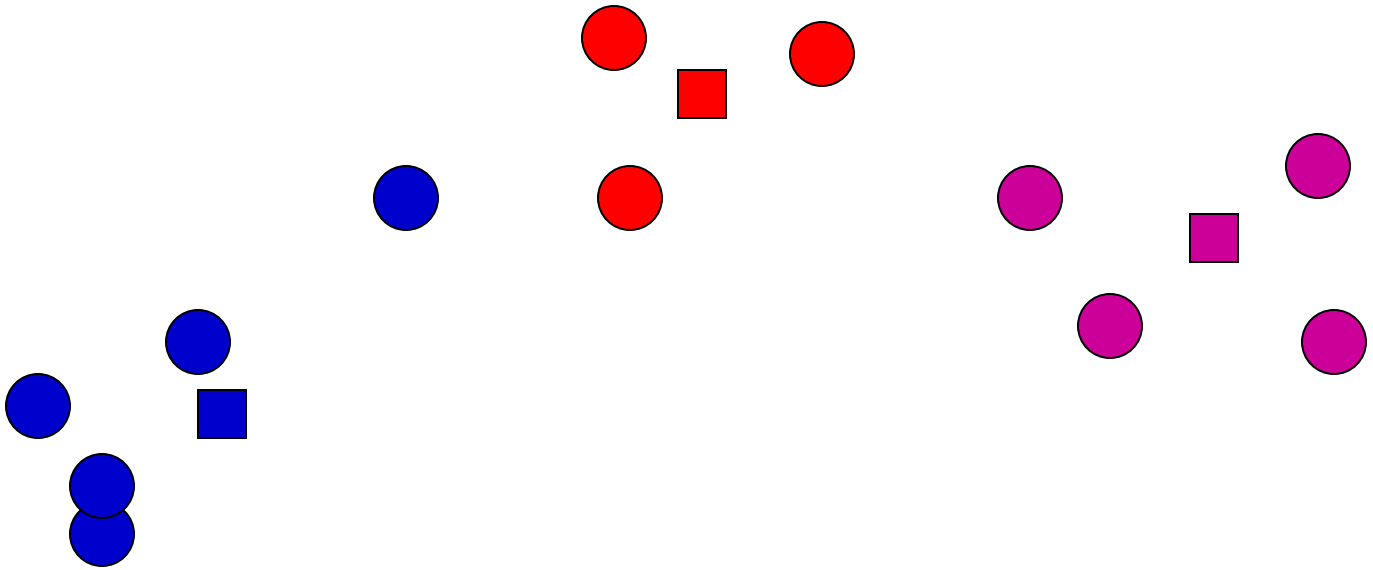
k-Means: Update Centers



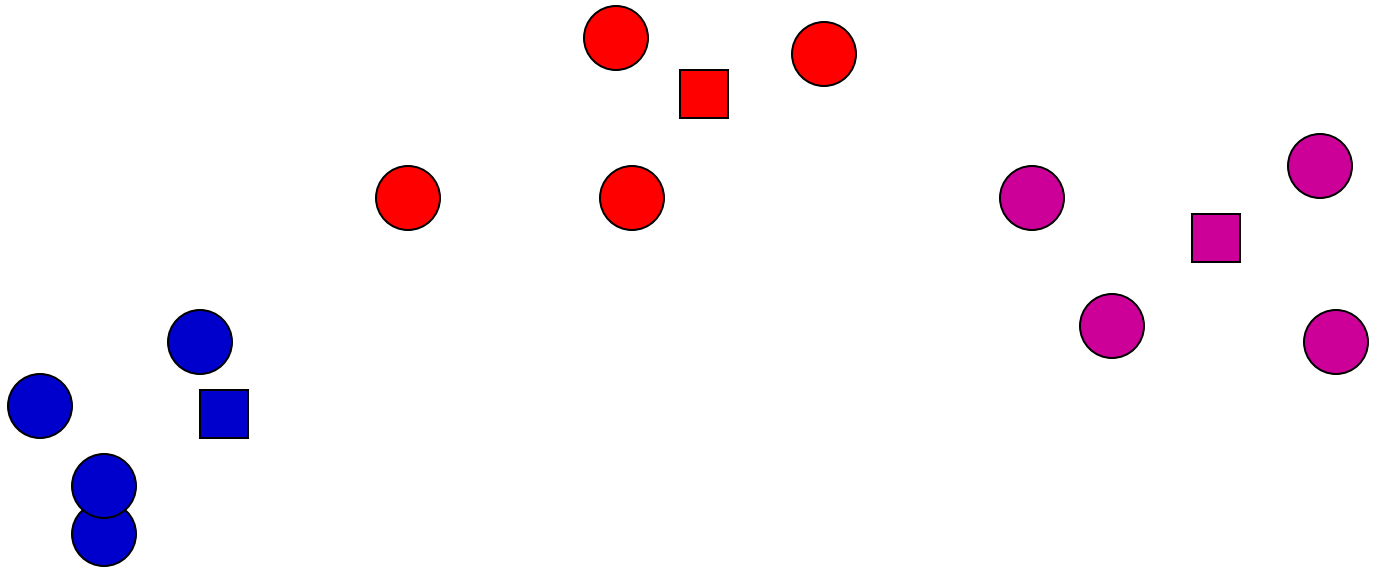
k-Means: Assign Points to Nearest Center



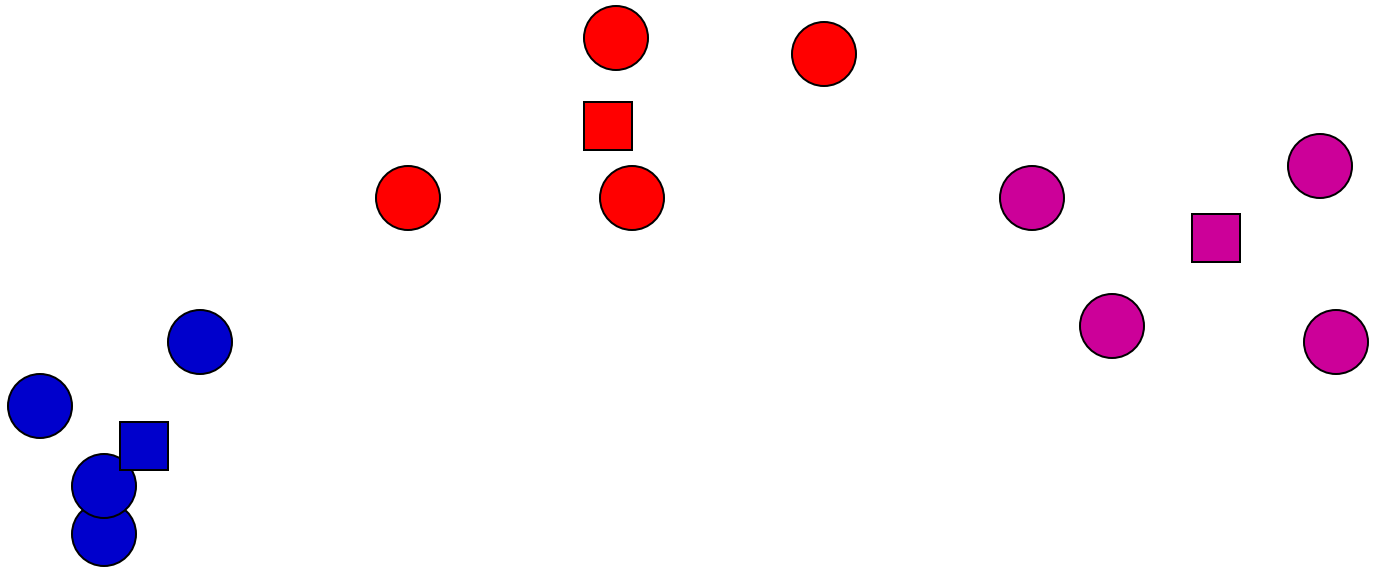
k-Means: Update Centers



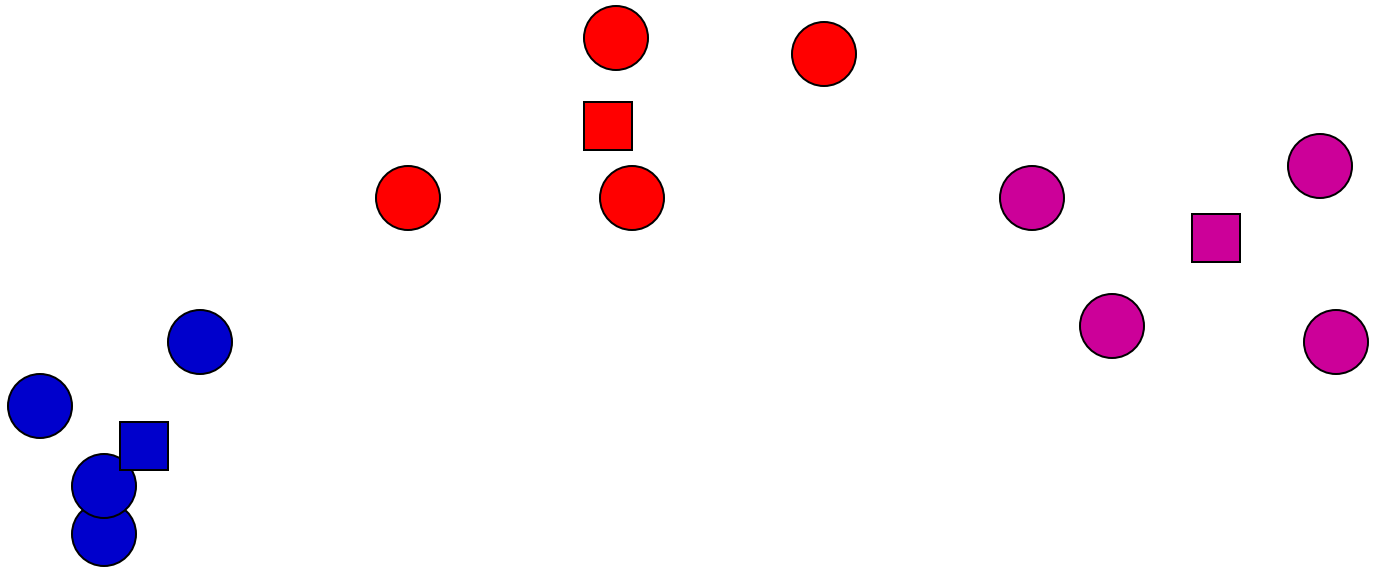
k-Means: Assign Points to Nearest Center



k-Means: Update Centers



k-Means: Assign Points to Nearest Center



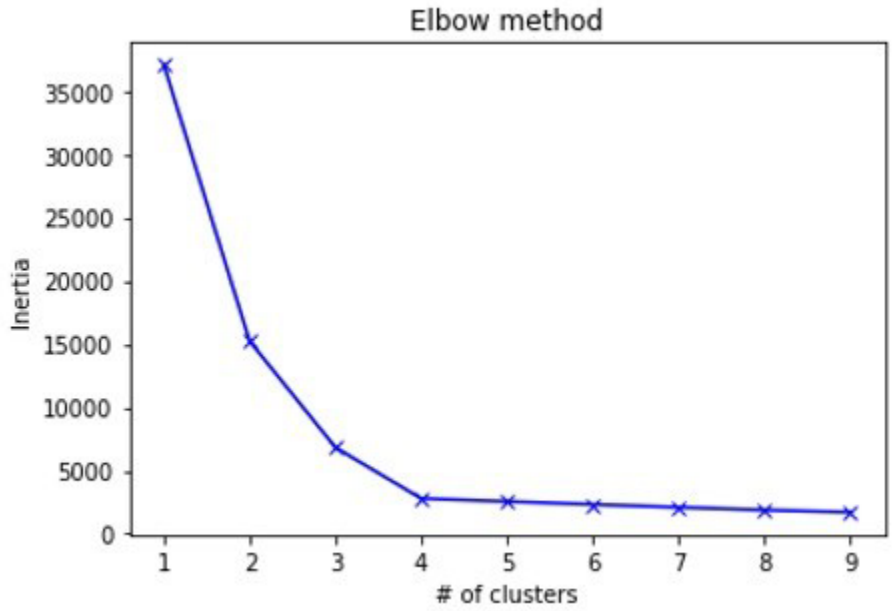
No changes: Done

k-Means Summary

- **Advantages:**
  - Relatively efficient
  - Simple implementation
- **Weaknesses:**
  - Often terminates at a local optimum
  - Applicable only when mean is defined (what about categorical data?)
  - Need to specify k, the number of clusters, in advance
  - Unable to handle noisy data and outliers
  - Not suitable to discover clusters with non-convex shapes

Elbow Method for Choosing *k*

- Inertia is the average distance between the points in the cluster and a centroid

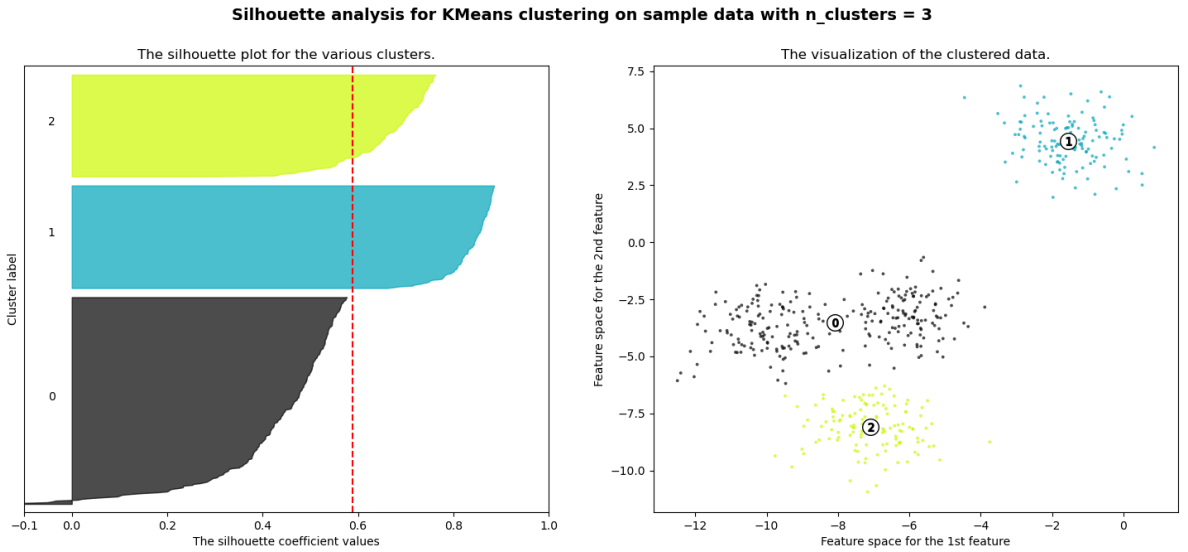


M. Eslamijam, “Customer segmentation: How machine learning makes marketing smart,” December 28, 2020, https://bdtechtalks.com/2020/12/28/machine-learning-customer-segmentation/

Silhouette Method for Choosing k (k=3)

- Silhouette coefficients measure separation of points from other clusters +1 far away, 0 on the decision boundary, -1 might be in wrong cluster

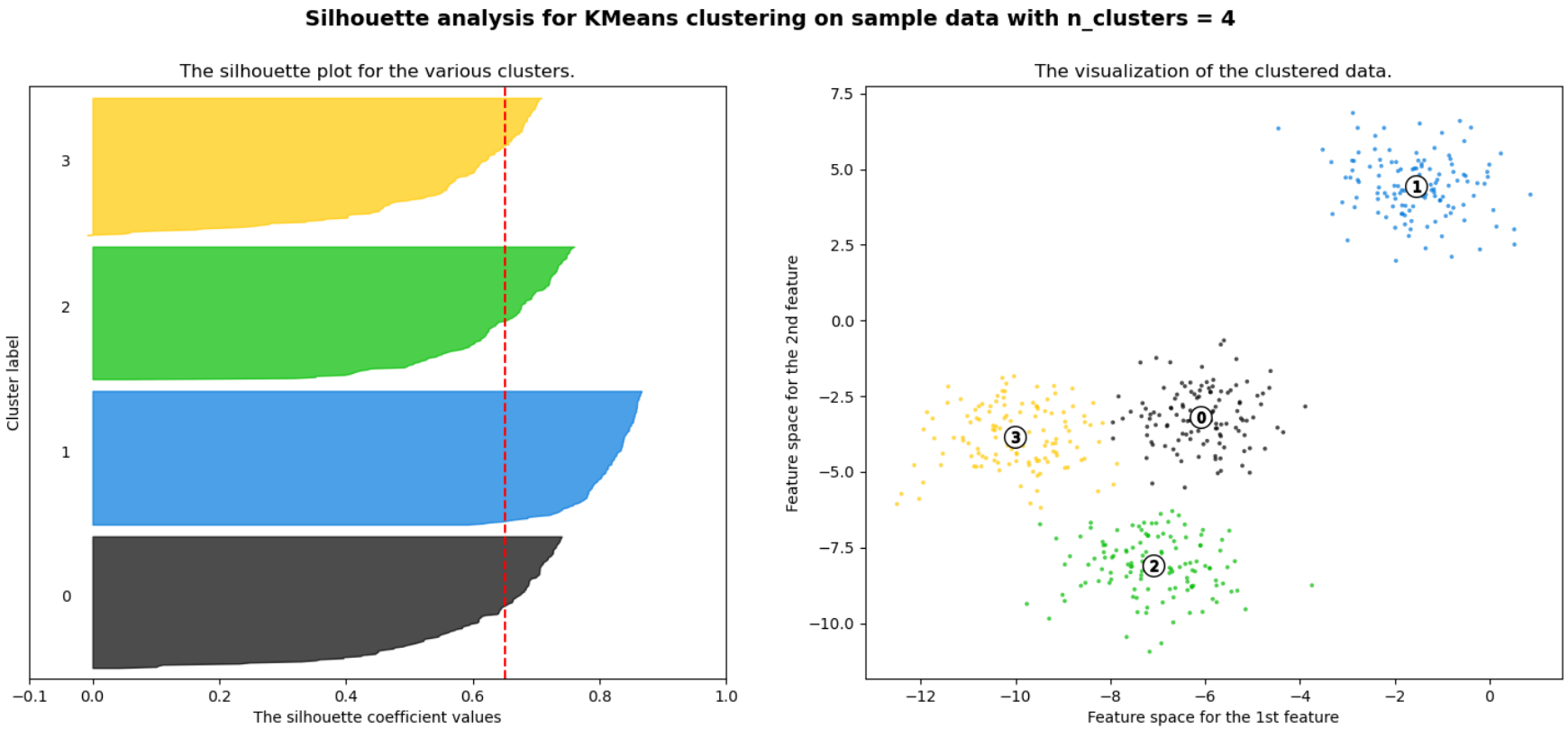
For k=3, avg silhouette coefficient (red dashed line) is **.588**



Sci-kit learn documentation, “Selecting the number of clusters with silhouette analysis on KMeans clustering”, https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Silhouette Method for Choosing k (k=4)

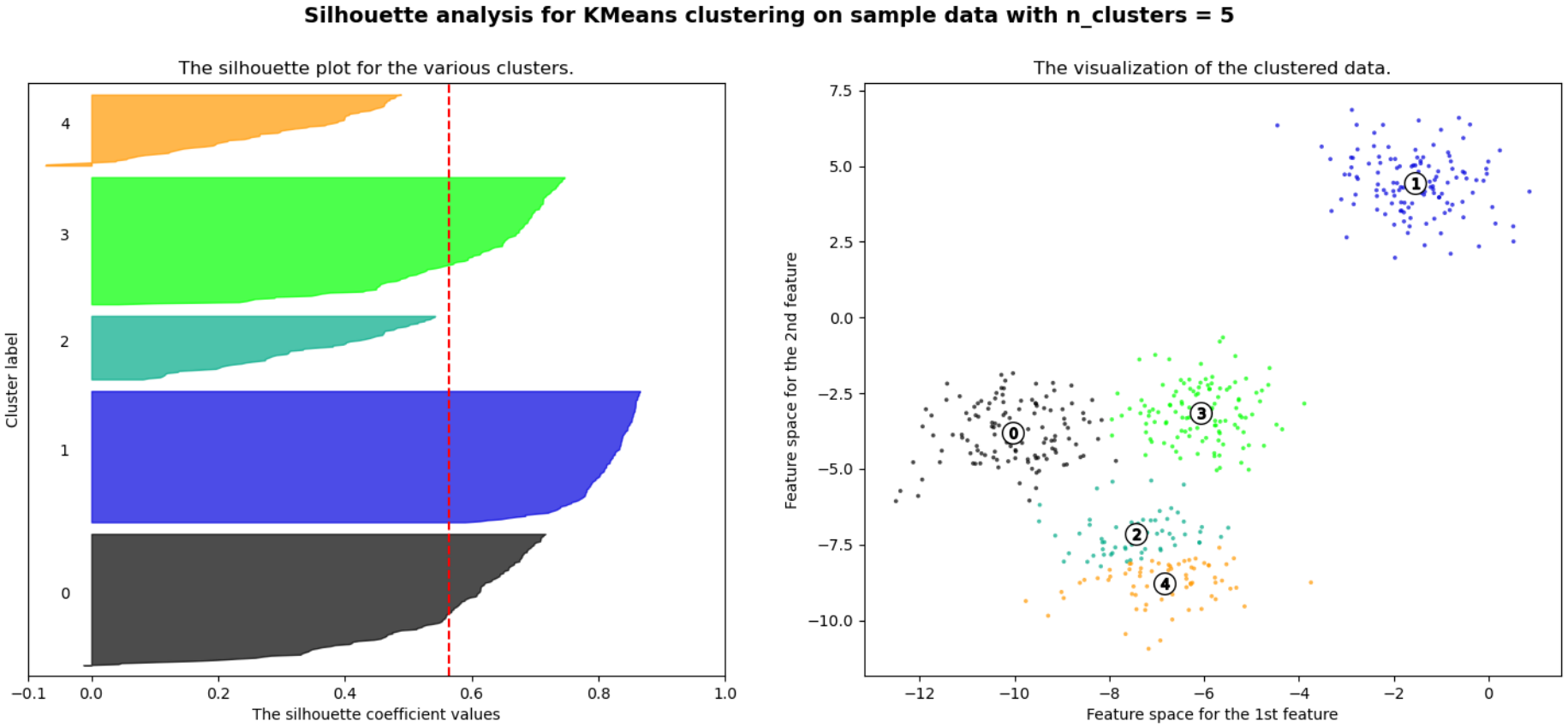
For k=4, avg silhouette coefficient (red dashed line) is **.651**



Sci-kit learn documentation, “Selecting the number of clusters with silhouette analysis on KMeans clustering”, https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Silhouette Method for Choosing k (k=5)

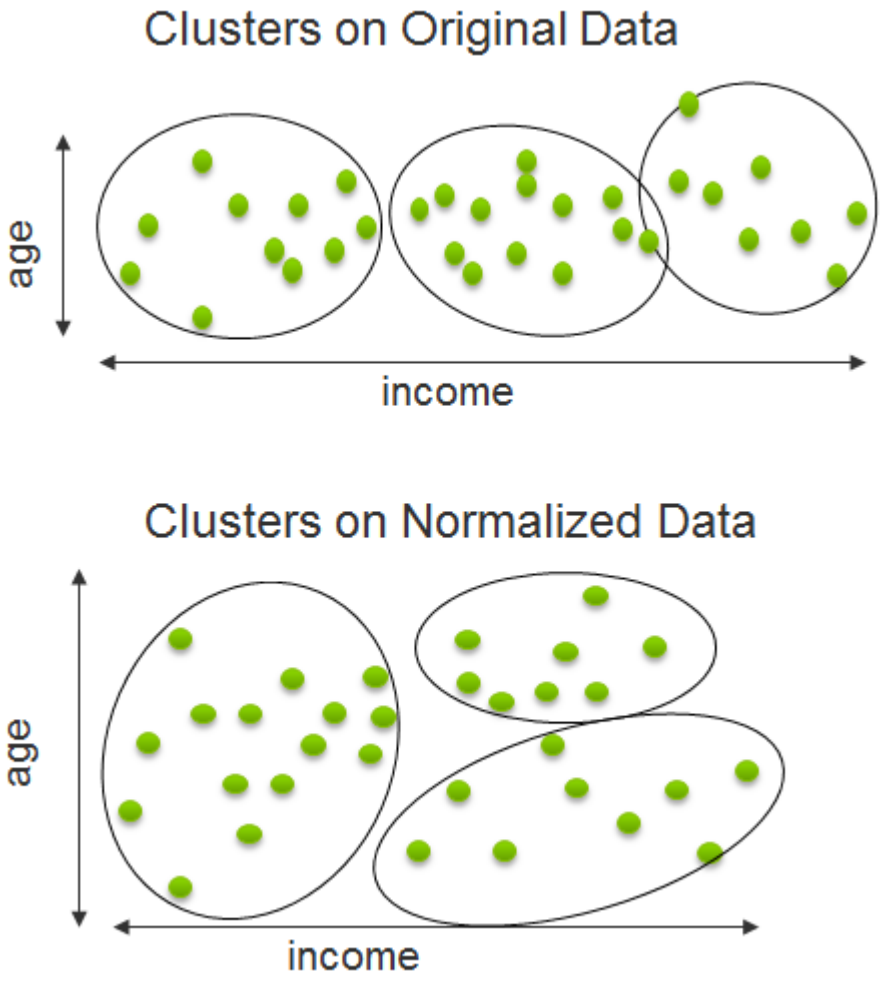
For k=5, avg silhouette coefficient (red dashed line) is **.566**



Sci-kit learn documentation, “Selecting the number of clusters with silhouette analysis on KMeans clustering”, https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

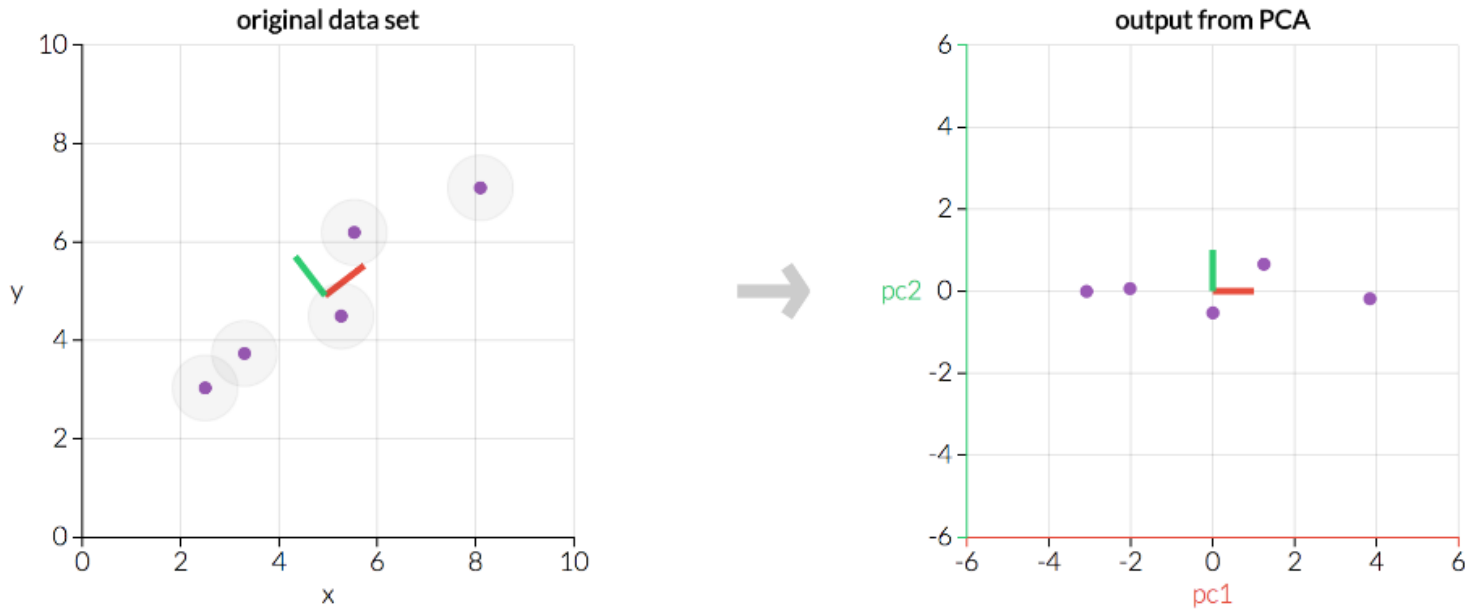
Importance of Normalization in Clustering

- Clustering works best on normalized data, as two fields with drastically different value spreads can produce unintended results.
- For example, if you want to cluster on age and income, the algorithm may assign more importance to income as it has a broader range.
- A popular normalization method is to obtain the standard deviation of values for each field and use that statistic to calculate a Z-score.



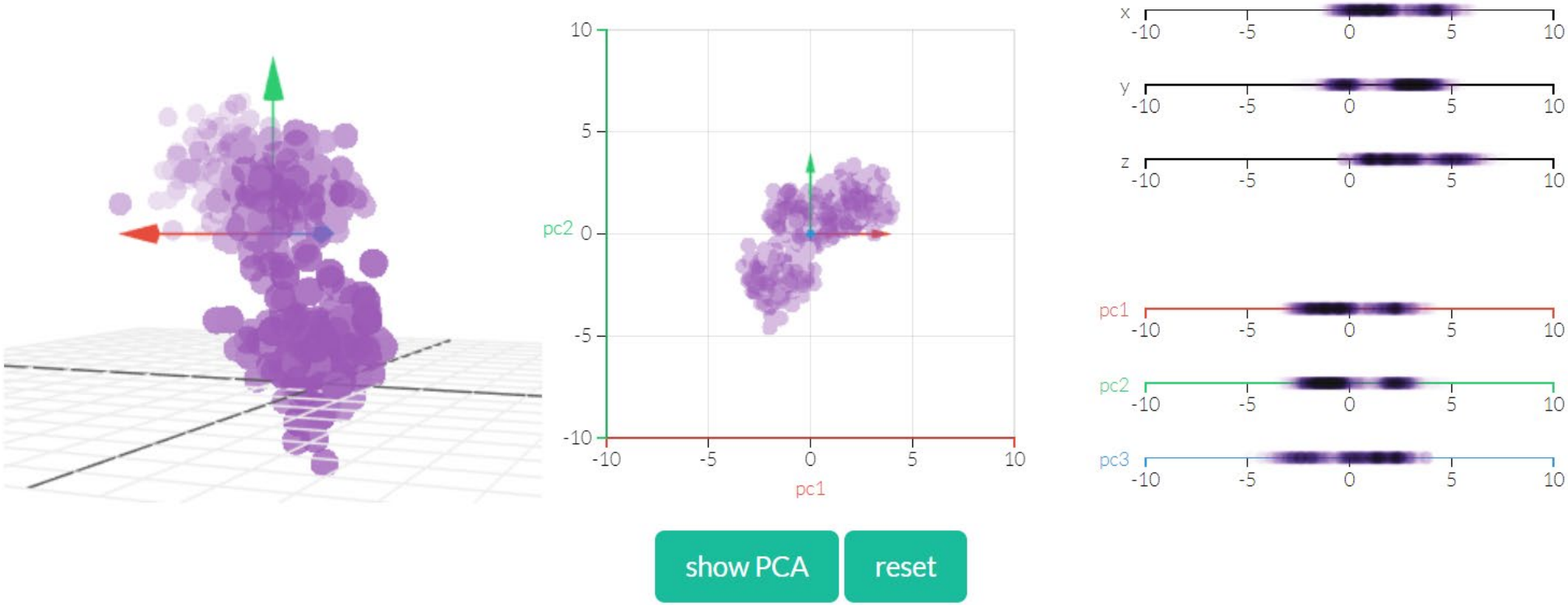
Principal Component Analysis: What Is It?

- Principal Component Analysis (PCA) is a technique which transforms a number of **possibly correlated** variables into a smaller number of **linearly uncorrelated** variables.



Brems, M.,"A One-Stop shop for Principal Component Analysis," April 17, 2018 https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c

PCA As Dimensionality Reduction



http://setosa.io/ev/principal-component-analysis/

Applying PCA in 17 Dimensions

	England	Wales	Scotland	N Ireland
Cheese	105	103	103	66
Carcass meat	245	227	242	267
Other meat	685	803	750	586
Fish	147	160	122	93
Fats and oils	193	235	184	209
Sugars	156	175	147	139
Fresh potatoes	720	874	566	1033
Fresh Veg	253	265	171	143
Other Veg	488	570	418	355
Processed potatoes	198	203	220	187
Processed Veg	360	365	337	334
Fresh fruit	1102	1137	957	674
Cereals	1472	1582	1462	1494
Beverages	57	73	53	47
Soft drinks	1374	1256	1572	1506
Alcoholic drinks	375	475	458	135
Confectionery	54	64	62	41

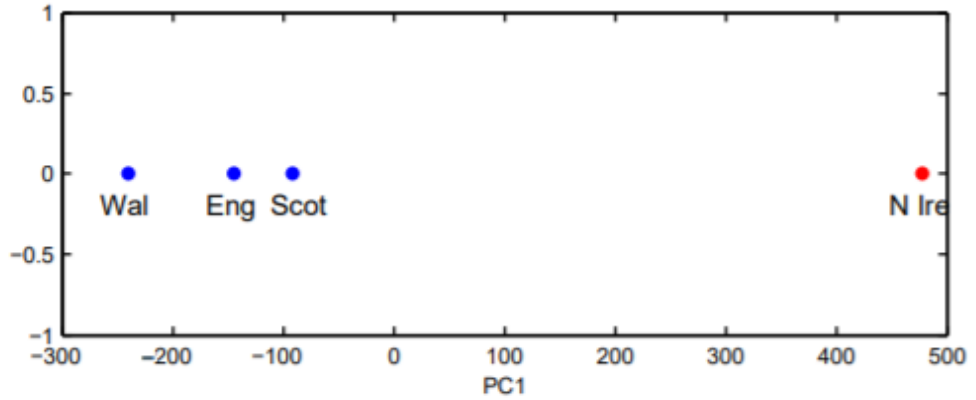
Table 1: UK food consumption in 1997 (g/person/week). Source: DEFRA website

Richardson, M., "Principal Component Analysis," May 2009, http://www.dsc.ufcg.edu.br/~hmg/disciplinas/posgraduacao/rn-copin-2014.3/material/SignalProcPCA.pdf

Score Plots for First and Second Principal Components

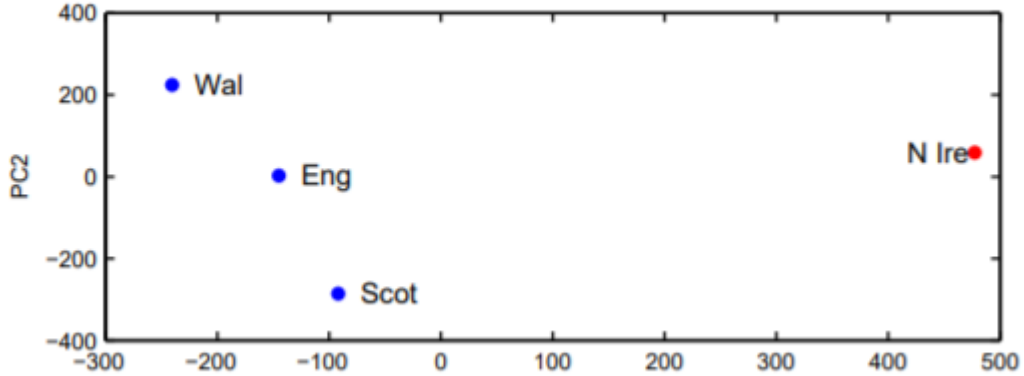
- First component explains **67%** of the variance

Figure 1: Projections onto first principal component (1-D space)



- First two components explains **97%** of the variance

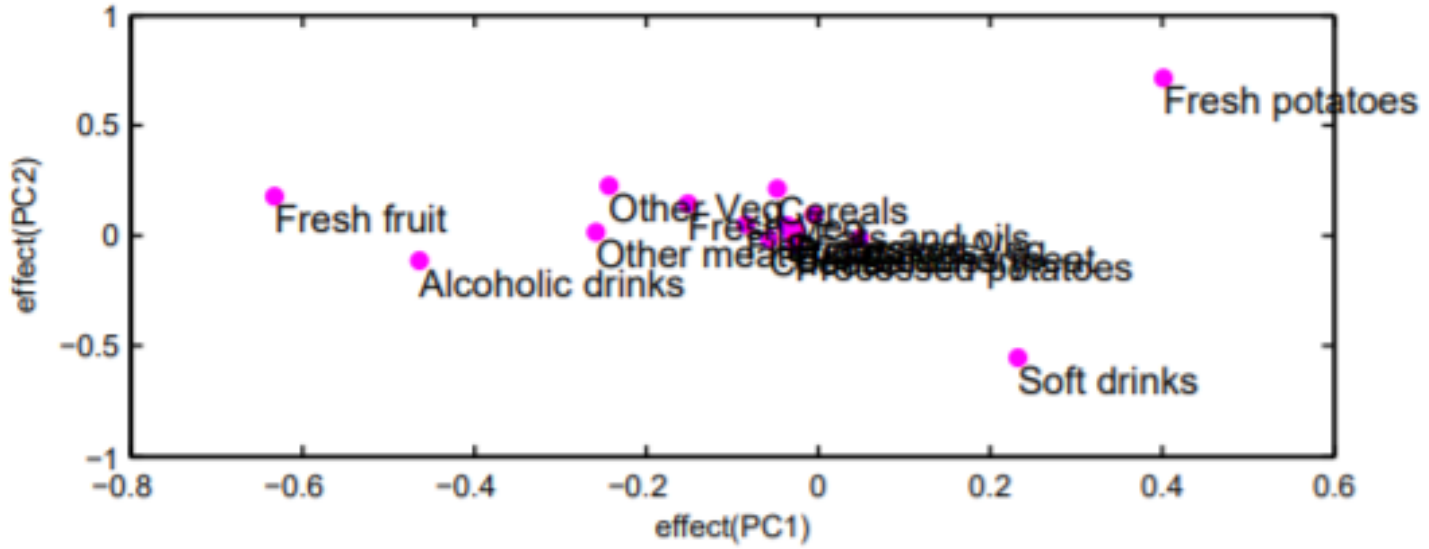
Figure 2: Projections onto first 2 principal components (2-D space)



Richardson, M., "Principal Component Analysis," May 2009, http://www.dsc.ufcg.edu.br/~hmg/disciplinas/posgraduacao/rn-dopin-2014.3/material/SignalProcPCA.pdf

Load Plot Shows Contributions of Original Variables

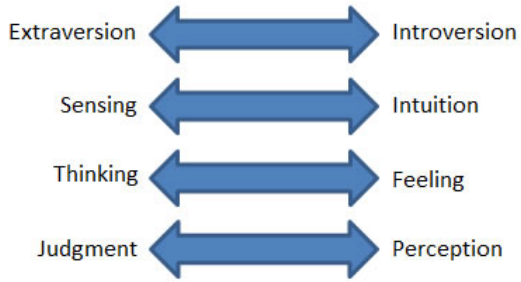
Figure 4: Load plot



Richardson, M., "Principal Component Analysis," May 2009, http://www.dsc.ufcg.edu.br/~hmg/disciplinas/posgraduacao/rn-copin-2014.3/material/SignalProcPCA.pdf

Principal Components Analysis: When to Use It

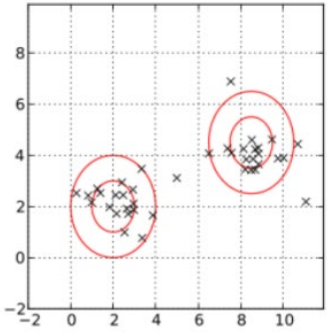
- In exploratory data analysis: To understand the structure behind a large noisy data set with many variables: Most of the variance is captured in the first few principal components
  - Used in Myers-Briggs to reduce answers to 75 questions to 4 dimensions



- To decide how many clusters to create (the value of *k* in *k*-means clustering)
- To compare the output of clustering algorithms
- As a preprocessing step to reduce the dimensionality of a data set before modeling (e.g., as input to regression)

Motivation for Gaussian Mixture Models

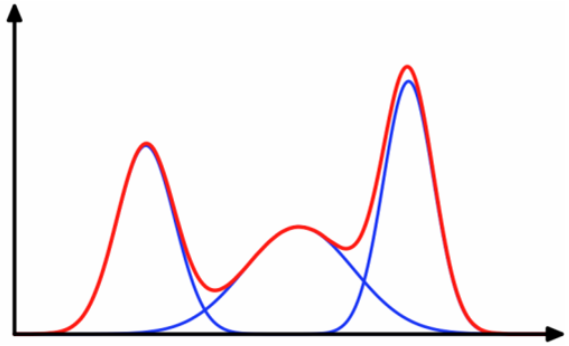
- Clustering, but rather than having hard assignments into clusters like k-means, we have soft assignments where points are members of a cluster with some probability.
- We assume that the data is coming by sampling from a mixture of Gaussians
- Each data point could have been generated by any of the distributions with a corresponding probability.

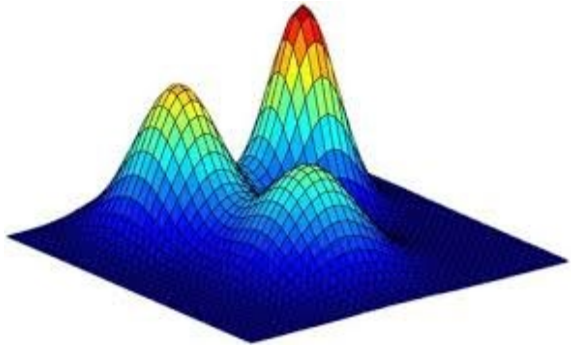


- In effect, each distribution has some ‘responsibility’ for generating a particular data point.

https://brilliant.org/wiki/gaussian-mixture-model/

Gaussian Mixture Models (GMM)





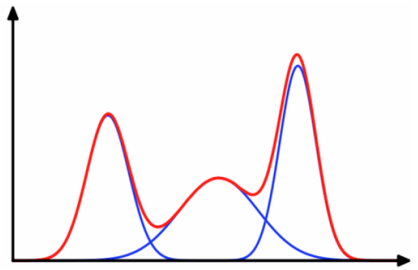
- The overall distribution is a weighted sum of component Gaussians ("regimes")
  - Mixture components: Individual smooth probability density functions
  - Mixture weights: The probability of being associated with each of the given components

http://scikit-learn.org/stable/modules/mixture.html

GMMs as Generators

- Assume *n* training data points and *k* components, each of which is a Gaussian distribution.
- Each component can be thought of as hidden process contributing to the overall mixture model

$$P(\mathbf{x}) = \sum_{i=1}^{k} P(C=i)\, P(\mathbf{x} \mid C=i)$$



- Learning problem is to find the means and covariances of each of the *k* components and the weights of each component so that the mixture model is most likely to ***generate*** the observed data set.

Russell & Norvig, 2020, p. 739

Unsupervised Clustering Recovers GMMs

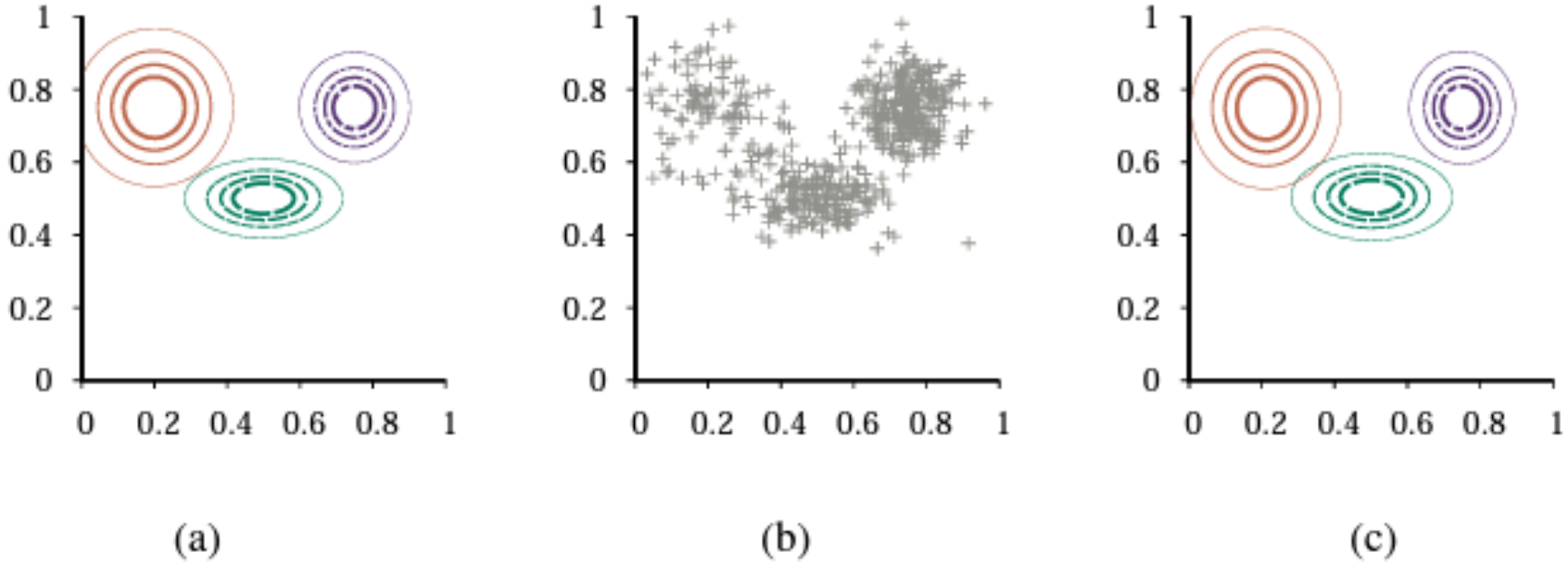
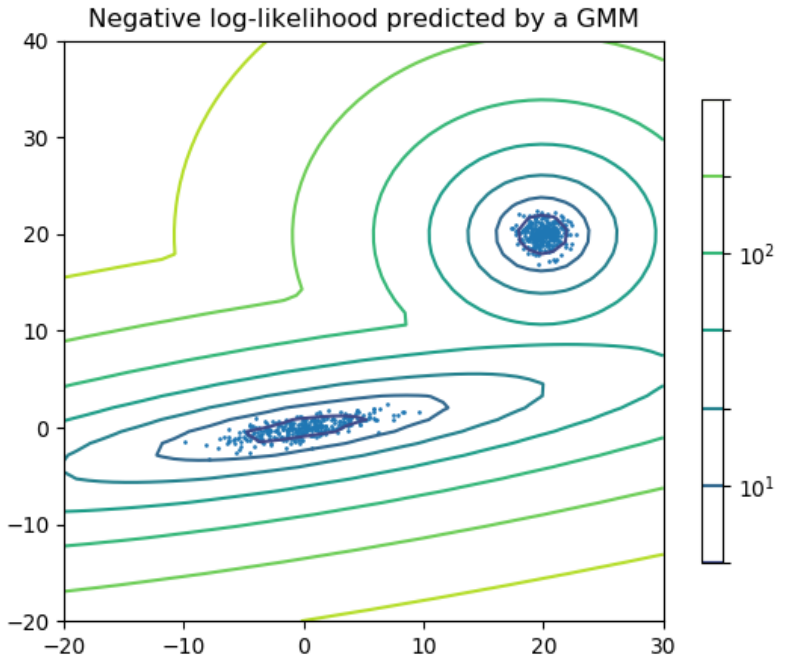


Figure 20.12 (a) A Gaussian mixture model with three components; the weights (left-to-right) are 0.2, 0.3, and 0.5. (b) 500 data points sampled from the model in (a). (c) The model reconstructed by EM from the data in (b).

Russell & Norvig, 2020, p. 739

Learning a GMM via Expectation Maximization

- The EM algorithm is an iterative algorithm that aims for the maximum likelihood parameter estimation for generating GMM
- Each iteration of the EM is guaranteed not to decrease the likelihood
- The EM algorithm works as follows:
  1. Start with an initial estimate of the means and the covariances of the Gaussian components
  2. Calculate probabilities of each data point belonging to each Gaussian component
     - If no improvement in expected value, stop (convergence reached).
  3. For each Gaussian component, re-estimate its mean and covariance and component weight using the data that has been assigned to it
  4. Go to Step 2



Russell & Norvig, 2020, p. 740

E-Step and M-Step in Expectation Maximization

- Expectation Step (**E-Step**): Compute the probabilities p_{ij} that component *i* generated data point *j*
- Maximization Step (**M-Step**): Using the probabilities, compute the new
  - Mean for each component: Sum of
    - Probability * data point / # points in the component
  - Covariance for each component: Sum of
    - Probability * difference from the mean * difference from the mean transposed / # points in the component
  - Weight for each component:
    - # points in component / total # points

$$p_{ij} = P(C = i \mid \mathbf{x}_j)$$

$$\boldsymbol{\mu}_i \leftarrow \sum_j p_{ij}\mathbf{x}_j / n_i$$

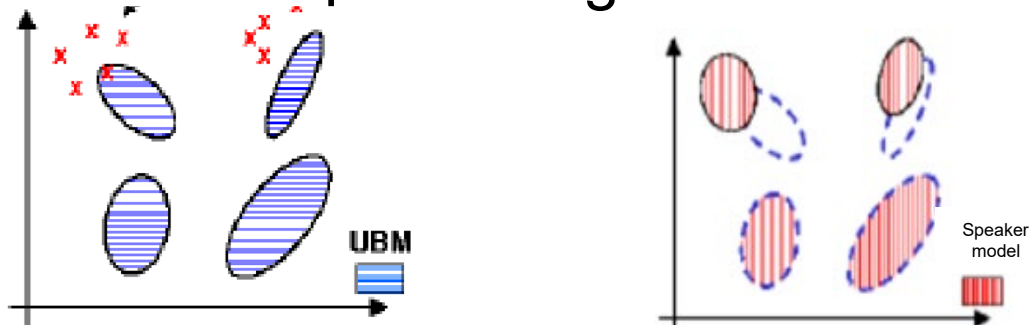
$$\boldsymbol{\Sigma}_i \leftarrow \sum_j p_{ij}(\mathbf{x}_j - \boldsymbol{\mu}_i)(\mathbf{x}_j - \boldsymbol{\mu}_i)^\top / n_i$$

$$w_i \leftarrow n_i / N$$

Russell & Norvig, 2020, p. 740

GMMs for Speaker Identification

- GMMs have been widely used for audio processing and speaker identification problems
  - Acoustic signal is decomposed into k-dimensional vectors and modeled as a GMM
- In speaker identification, target speaker data may be severely limited, but large amounts of data from speakers of no interest is freely available
  - 2 hours of data from speakers of no interest are used to train a universal background model (UBM)
  - The resulting UBM is then adapted using the 2-min voice cut from the target speaker



W. J. J. Roberts, Y. Ephraim, and H. W. Sabrin, "Speaker classification using composite hypothesis testing and list decoding," *IEEE Trans. Speech and Audio Proc.*, Vol. 13, pp. 211-219, 2005.

Conclusion

- Techniques for unsupervised learning without a target variable includes *k*-means clustering
  - Elbow and silhouette methods can be used to determine value of *k*
- Expectation Maximization can be used to estimate the parameters of Gaussian Mixture Models , a process similar to k-means but for clusters defined as probability distributions rather than by centroids

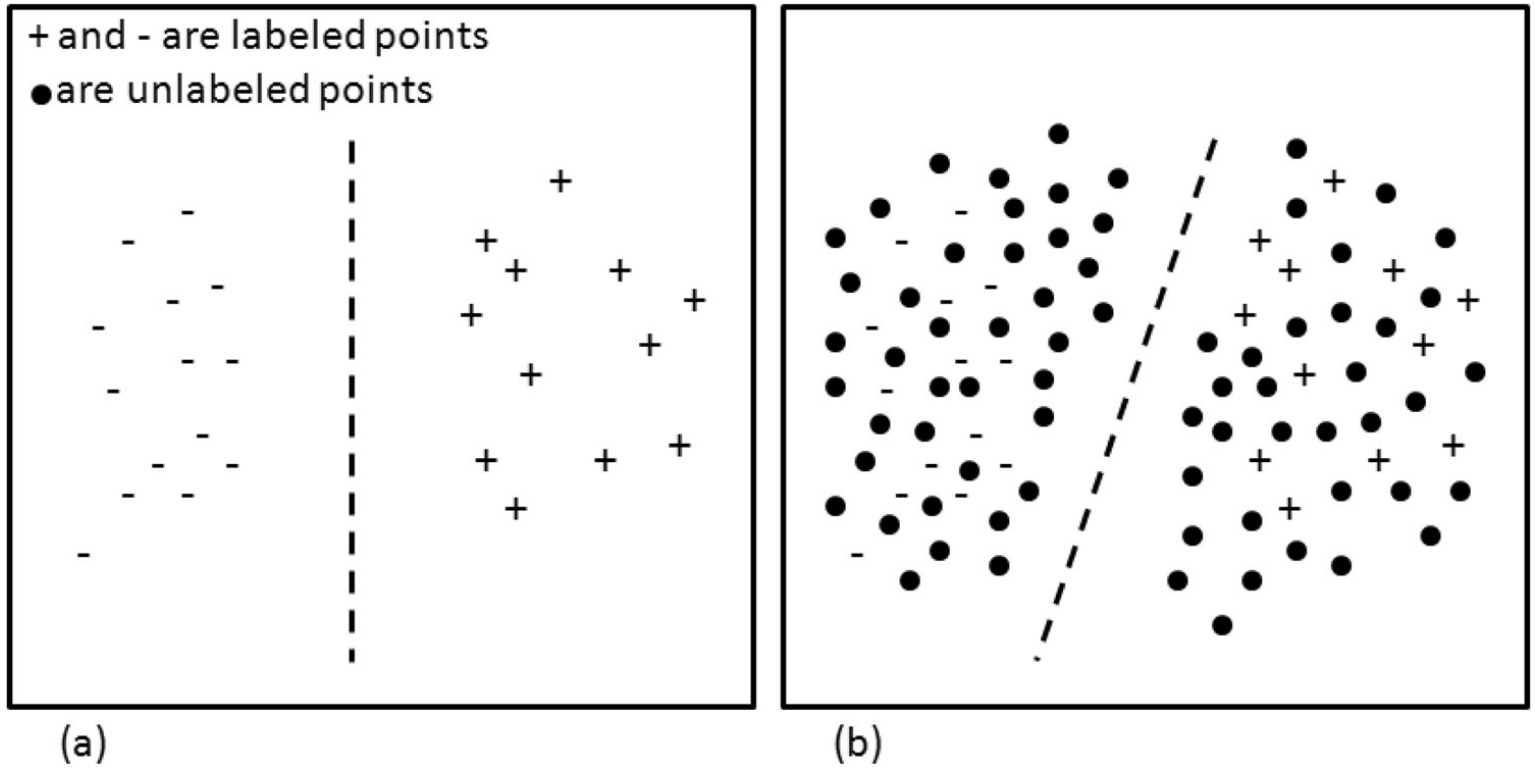
- **Next class (September 11): Reinforcement learning**
  - Also: **Quiz #1**

Appendix: Semi-Supervised Learning

- Extracting relevant features from data is difficult
- Unlabeled data is cheap and abundant
- Labeling examples is a time-intensive task requiring expertise
  - Labeling CT and MRI images requires skilled physician time
  - For Switchboard speech corpus, 400 hours annotation time for each hour of speech
- Labeling data may require infeasible or unethical experiments

Key question: Can we use unlabelled data to learn better models than using just the labelled data?

Semi-Supervised Learning Example



Mohammad Peikari, Sherine Salama, Sharon Nofech-Mozes & Anne L. Martel , “A Cluster-then-label Semi-supervised Learning Approach for Pathology Image Classification”, *Scientific Reports* volume 8, Article number: 7193 (2018) , https://www.nature.com/articles/s41598-018-24876-0

Self-Training

- Assume L is set of labeled instances, U are unlabeled instances
- Repeat
  - Create model from training on L
  - For each instance in U
    - Classify it based on current model
    - If classification probability greater than some threshold, add instance to L and delete it from U
- Until no instances in U were classified with high probability

https://towardsdatascience.com/self-training-classifier-how-to-make-any-algorithm-behave-like-a-semi-supervised-one-2958e7b54ab7

Notes on Self-Training

- Can use any classifier (SVM, logistic regression, decision tree, etc.)
- Probability threshold is arbitrary. Despite requiring high probability, some instances will be classified incorrectly, and this may decrease performance of algorithm.
- Instead of specifying probability threshold, can specify that only the k highest confidence members of U will be added
- Many other semi-supervised learning methods, e.g. co-training (see appendix)