Introduction to Machine Learning in Geosciences GEO371T/GEO395D.1

Clustering

Mrinal K. Sen
Geosciences
UT Austin

October 24, 2023

Content - Clustering

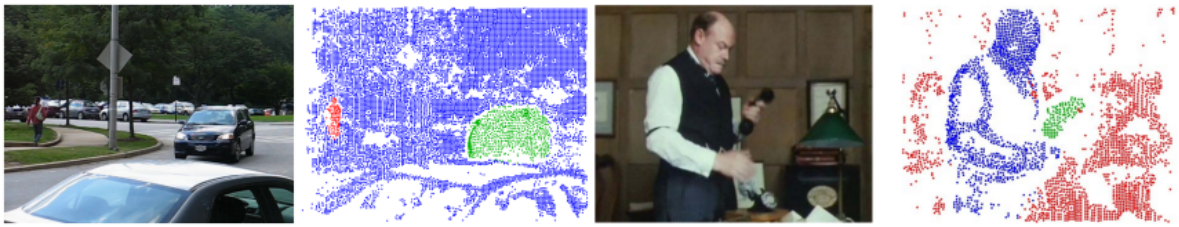
- Introduction
- Clustering Types
- Algorithms
  - Self Organizing Maps (SOM)
  - K-means Clustering
  - Soft K-means Clustering Mixture Models - EM method
  - Hierarchical Clustering
  - Density-based Clustering
- Cluster Validations

Introduction

- Clustering is one of the most widely used techniques for exploratory data analysis.
- Computational biologists cluster genes on the basis of similarities in their expression in different experiments.
- Astronomers cluster stars on the basis of their spatial proximity.
- Geologists cluster rock types based on their physical properties.
- **Clustering is a type of unsupervised learning**

Example1

- Determine groups of people in image above
  - based on clothing styles
  - gender, age, etc

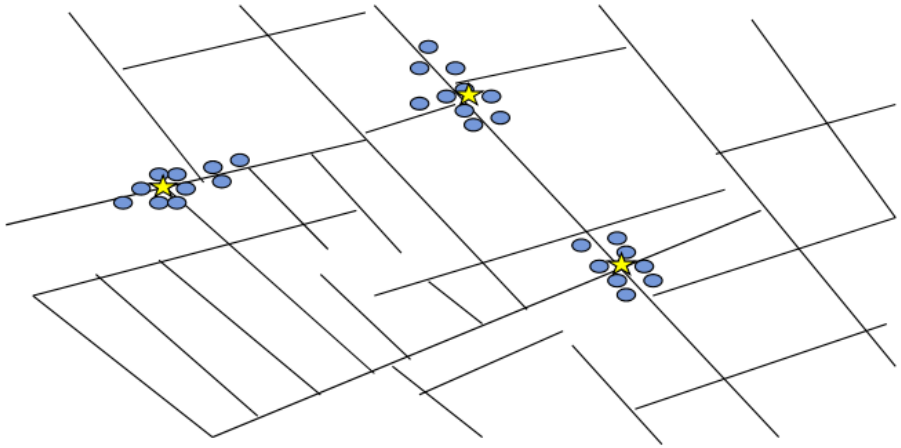


- Determine moving objects in videos

Example2

First (?) Application of Clustering

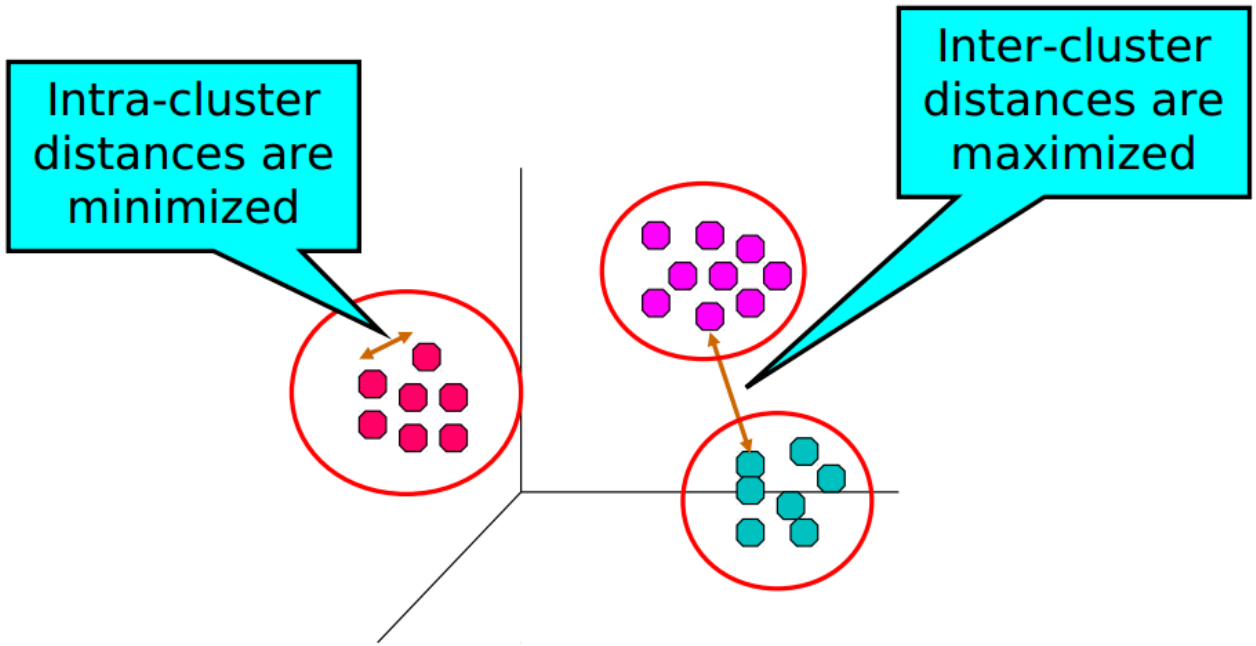
- John Snow, a London physician plotted the location of cholera deaths on a map during an outbreak in the 1850s.
- The locations indicated that cases were clustered around certain intersections where there were polluted wells -- thus exposing both the problem and the solution.



From: Nina Mishra HP Labs

What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



Resource: Lecture Notes

for Chapter 8 Slides by Tan, Steinbach, Kumar adapted by Michael Hahsler

What is Cluster Analysis?

What is Similarity?

Hard to define! *But we know it when we see it*

The real meaning of similarity is a philosophical question. We will take a more pragmatic approach: think in terms of a **distance** (rather than similarity) between random variables.

5

Unsupervised Learning

- **Supervised learning algorithms** have a clear goal: produce desired outputs for given inputs
- Goal of **unsupervised learning algorithms** (no explicit feedback whether outputs of system are correct) less clear.
- Tasks to consider:
  - Reduce dimensionality
  - Find clusters
  - Model data density
  - Find hidden cause
- Key Utility
  - Compress Data
  - Detect Outliers
  - Facilitate other learning.

Resource:Zemel, Urtasun, Fidler (UofT) CSC 411: 12-Clustering

Challenges

- What is a good notion of "similarity"?
- Euclidean distance bad for Image!
- The notion of similarity used can make the same algorithm behave in very different ways and can in some cases be a motivation for developing new algorithms (not necessarily just for clustering algorithms)
- Another question is how to compare different clustering algorithms
  - May have specific methods for making these decisions based on the clustering algorithms used.
  - Can also use performance on down-the-line tasks as a proxy when choosing between different setups

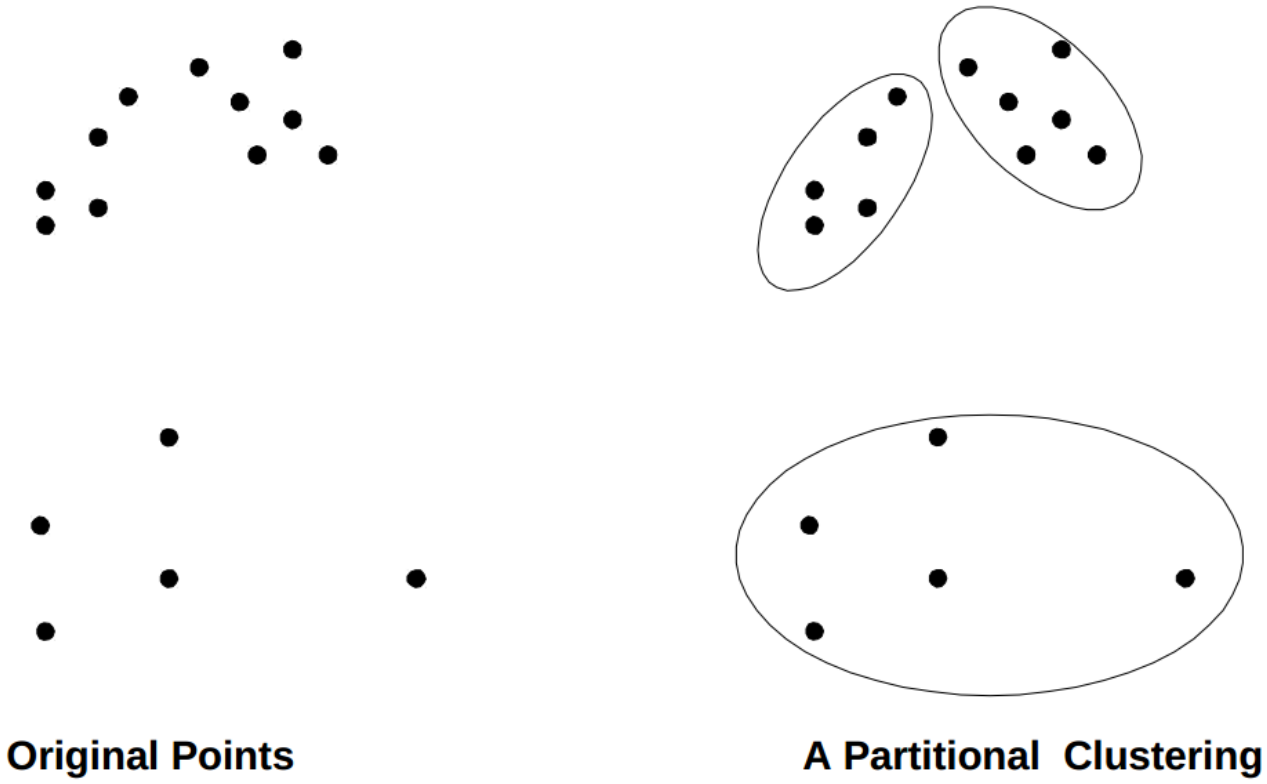
Resource: Kamyar Ghasemipour

Types of Clustering

- A clustering is a set of clusters!
- **Partitional Clustering**
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset.
- **Hierarchical clustering**
  - A set of nested clusters organized as a hierarchical tree

Resource: Lecture Notes for Chapter 8 Slides by Tan, Steinbach, Kumar adapted by Michael Hahsler

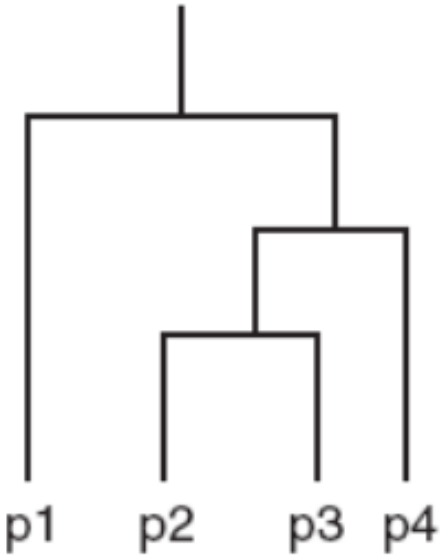
Partitional Clustering



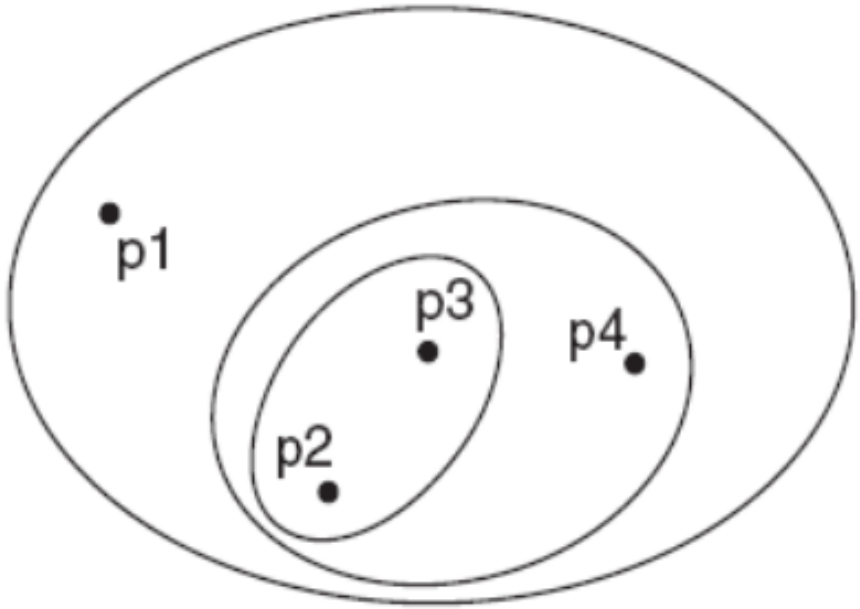
Resource: Lecture Notes

for Chapter 8 Slides by Tan, Steinbach, Kumar adapted by Michael Hahsler

Hierarchical Clustering



(a) Dendrogram.



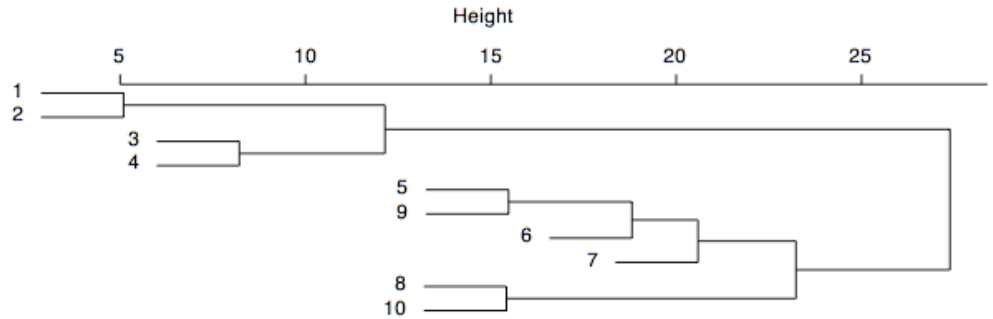
(b) Nested cluster diagram.

Resource:

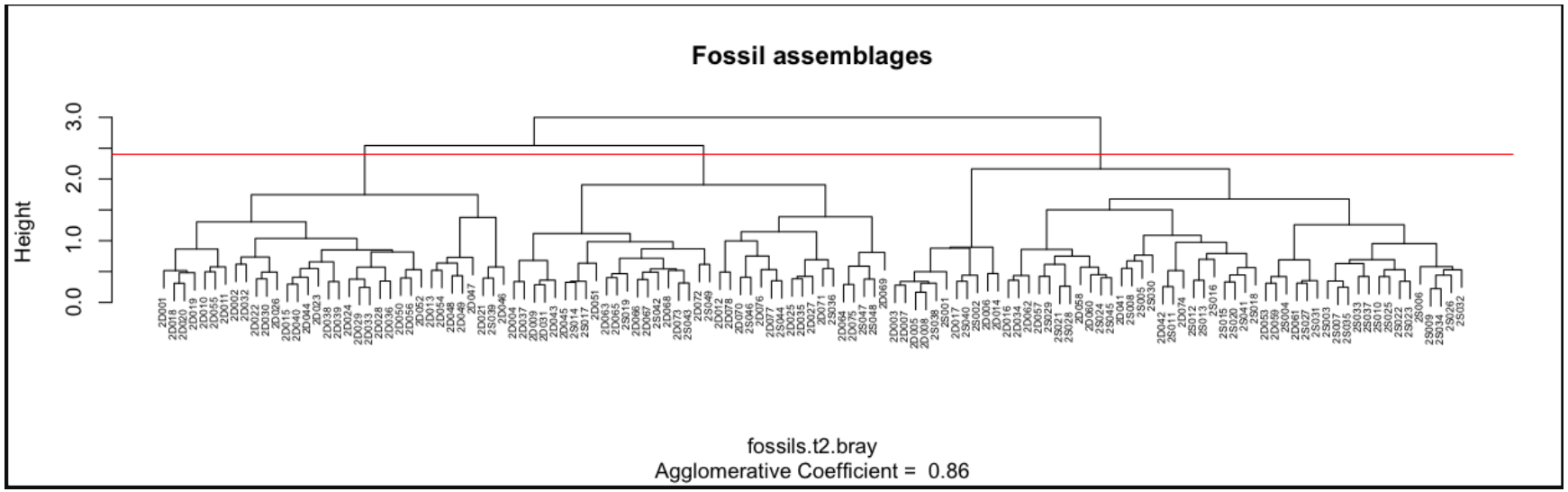
Lecture Notes for Chapter 8 Slides by Tan, Steinbach, Kumar adapted by Michael Hahsler

Dendograms

- The results of the cluster analysis are shown by a dendrogram.
- it lists all of the samples and indicates at what level of similarity any two clusters were joined.
- The x-axis is the measure of the similarity or distance at which clusters join and different programs use different measures on this axis.
- samples 1 and 2 are the most similar and join to form the first cluster (with a similarity or height of 5), followed by samples 3 and 4. The last two clusters to form are 1–2-3–4 and 5–9-6–7-8–10. In some cases, clusters grow by the joining of two smaller clusters, such as the joining of 1–2 and 3–4. In other cases, clusters grow by the addition of single samples, such as the join of 6 with 5–9, followed by the join of 7. The growth of a cluster by the repeated addition of single samples is called chaining.



Dendograms


Fossil assemblages
Height
0.0
1.0
2.0
3.0
fossils.t2.bray
Agglomerative Coefficient = 0.86

Dendograms

An example is presented below that illustrates the relationship between dendrogram and dissimilarity as evaluated between objects with 2 variables. Essentially, the level at which branches merge (relative to the "root" of the tree) is related to their similarity. In the example below it is clear that (in terms of clay and rock fragment content) soils 4 and 5 are more similar to each other than to soil 2. In addition, soils 1 and 3 are more similar to each other than soils 4 and 5 are to soil 2. Recall that in this case pair-wise dissimilarity is based on the Euclidean distance between soils in terms of their clay content and rock fragment content. Therefore proximity in the scatter plot of frock frags vs. clay is directly related to our simple evaluation of "dissimilarity". Inline-comments in the code below elaborate further.

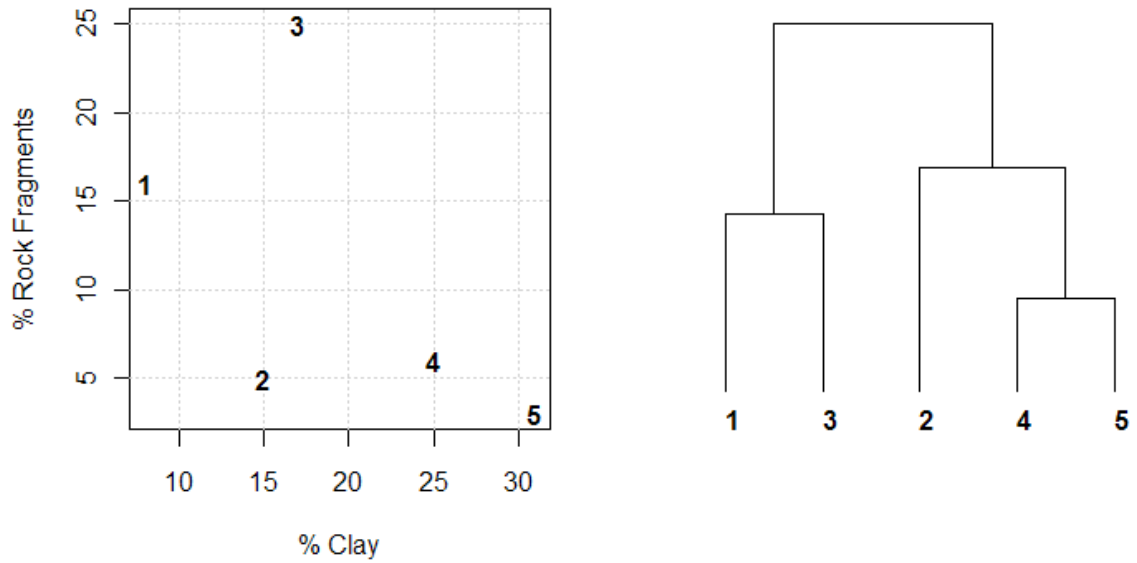


Image: Data to Dendrogram

https://casoilresource.lawr.ucdavis.edu/blog/graphical-explanation-how-interpret-dendrogram/

Linkage-Based Clustering

Linkage-based clustering is probably the simplest and most straightforward paradigm of clustering. These algorithms proceed in a sequence of rounds. They start from the trivial clustering that has each data point as a single-point cluster. Then, repeatedly, these algorithms merge the closest clusters of the previous clustering. Consequently, the number of clusters decreases with each such round. If kept going, such algorithms would eventually result in the trivial clustering in which all of the domain points share one large cluster.

Linkage-Based Clustering

1. Single Linkage clustering, in which the between-clusters distance is defined by the minimum distance between members of the two clusters, namely,

$$D(A,B) \stackrel{\text{def}}{=} \min\{d(x,y) : x \in A,\ y \in B\}$$

2. Average Linkage clustering, in which the distance between two clusters is defined to be the average distance between a point in one of the clusters and a point in the other, namely,

$$D(A,B) \stackrel{\text{def}}{=} \frac{1}{|A||B|} \sum_{x \in A,\ y \in B} d(x,y)$$

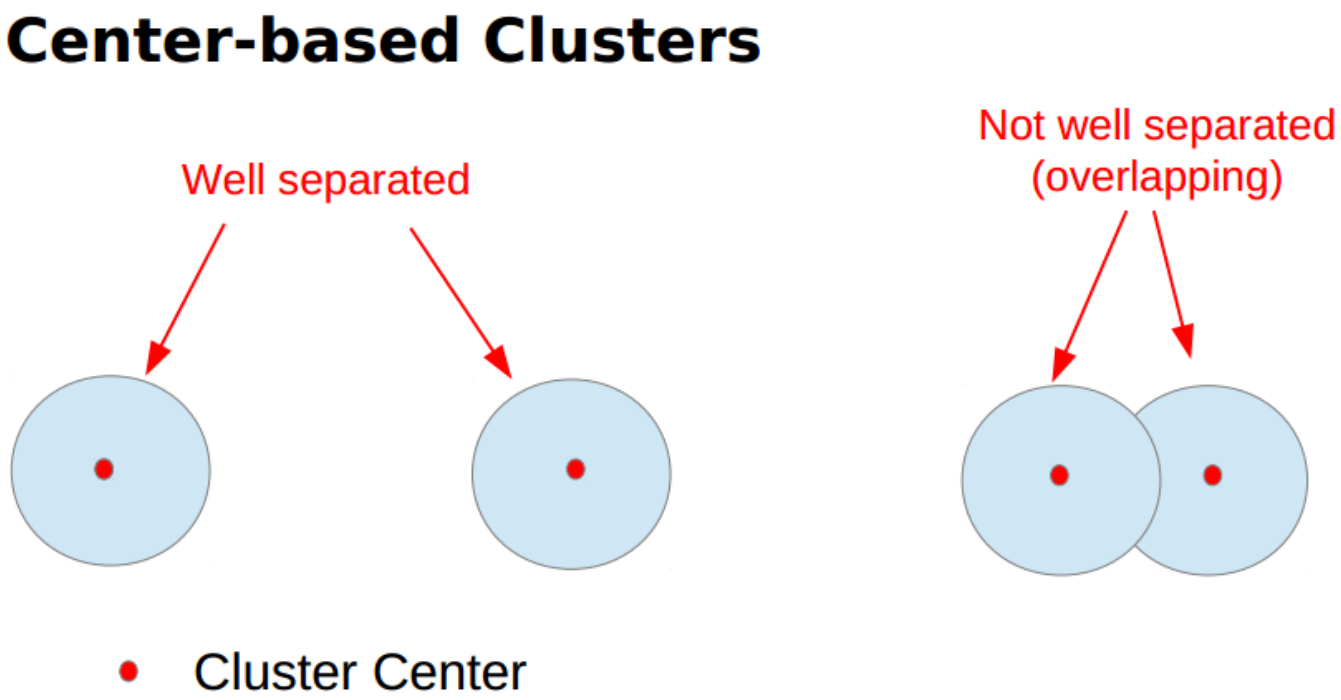
3. Max Linkage clustering, in which the distance between two clusters is defined as the maximum distance between their elements, namely,

$$D(A,B) \stackrel{\text{def}}{=} \max\{d(x,y) : x \in A,\ y \in B\}.$$

Types of Clusters

- Center-based clusters
- Contiguous clusters
- Density-based clusters
- Conceptual clusters
- Described by an Objective Function

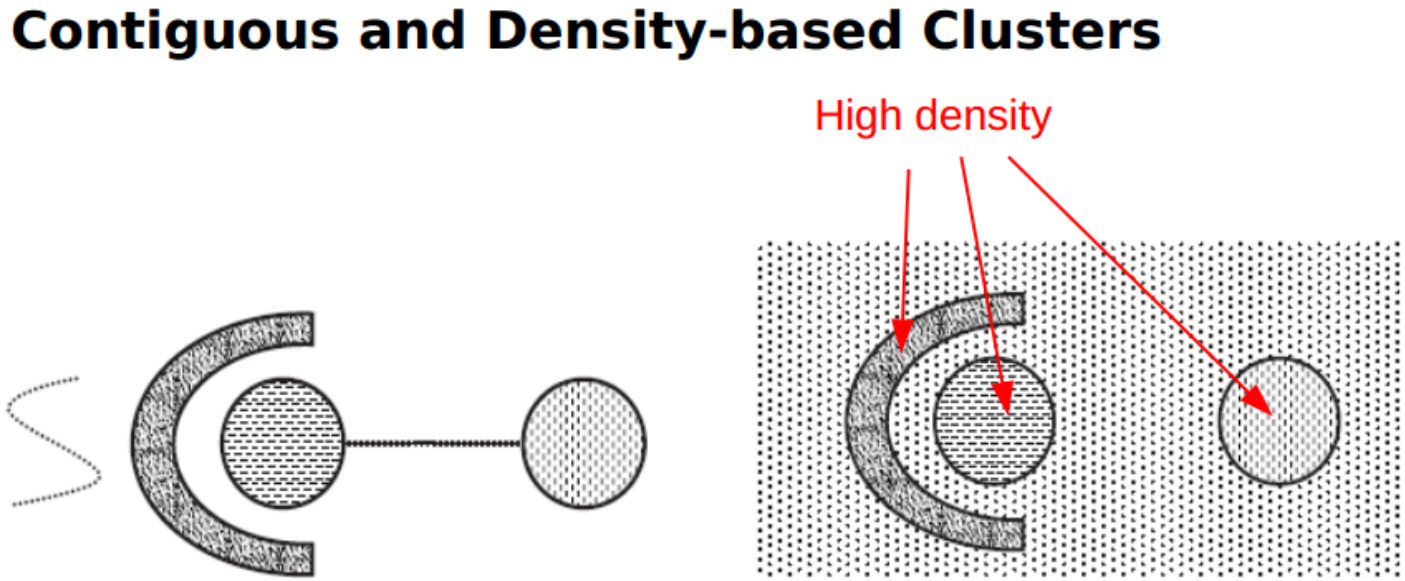
Center-based Clustering



A cluster is a set of objects such that an object in a cluster is closer (more similar) to the “center” of a cluster, than to the center of any other cluster

The center of a cluster is often a **centroid**, the average of all the points in the cluster, or a **medoid**, the most “representative” point of a cluster

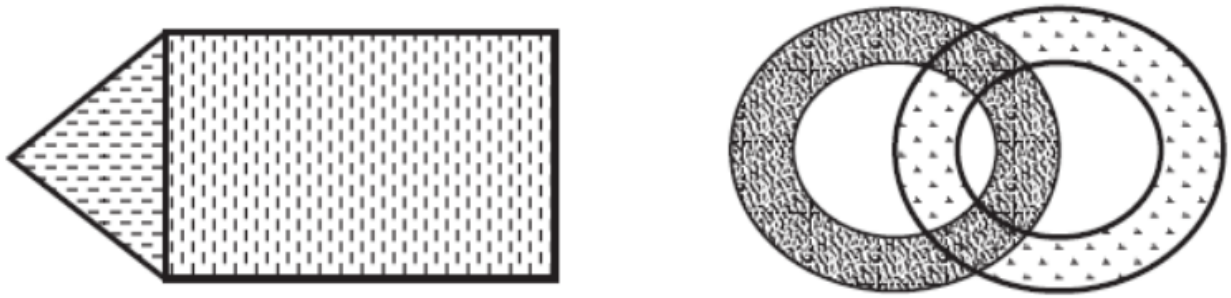
Contiguous and Density-based Clusters



(c) Contiguity-based clusters. Each point is closer to at least one point in its cluster than to any point in another cluster.

(d) Density-based clusters. Clusters are regions of high density separated by regions of low density.

Conceptual Clusters



(e) Conceptual clusters. Points in a cluster share some general property that derives from the entire set of points. (Points in the intersection of the circles belong to both.)

Conceptual clusters are hard to detect since they are often not:

- Center-based
- Contiguity-based
- Density-based

Objective Functions

The best clustering minimizes or maximizes an objective function.

- Example: Minimize the Sum of Squared Errors

$$SSE = \sum_{i=1}^{K} \sum_{\boldsymbol{x} \in C_i} \|\boldsymbol{x} - \boldsymbol{m_i}\|^2$$

  - $\boldsymbol{x}$ is a data point in cluster C_i, $\boldsymbol{m_i}$ is the center for cluster C_i as the mean of all points in the cluster and ||.|| is the L2 norm (= Euclidean distance).

Problem: Enumerate all possible ways of dividing the points into clusters and evaluate the `goodness' of each potential set of clusters by using the given objective function. (NP Hard)

Objective Functions

Global objective function

- Typically used in **partitional clustering** (k-means uses SSE)
- **Mixture Models** assume that the data is a 'mixture' of a number of parametric statistical distributions (e.g., a mixture of Gaussians).

Local objective function

- **Hierarchical clustering** algorithms typically have local objectives
- **Density-based clustering** is based on local density estimates
- **Graph based approaches.** Graph partitioning and shared nearest neighbors

We will talk about the objective functions when we talk about individual clustering algorithms.

SOM

- A form of unspervised neural network
- Used for visualization and exploratory data analysis of high dimensional datasets. Also can be used for clustering.
- The self-organizing map (SOM) (Kohonen, 2001) is one of the most effective unsupervised pattern recognition techniques!
- Kohonen (1982) presented his clustering and dimensionality reduction methods as a "self-organizing process" whereby a "simple network of adaptive physical elements" is made to resonate in a particular way with externally provided signals (a "primary event space") and tried to link these ideas with the functionality of the human brains (Murtagh, 1995).
- This popular neural network method was initially used in biology and computer science for data mining purposes.

SOM

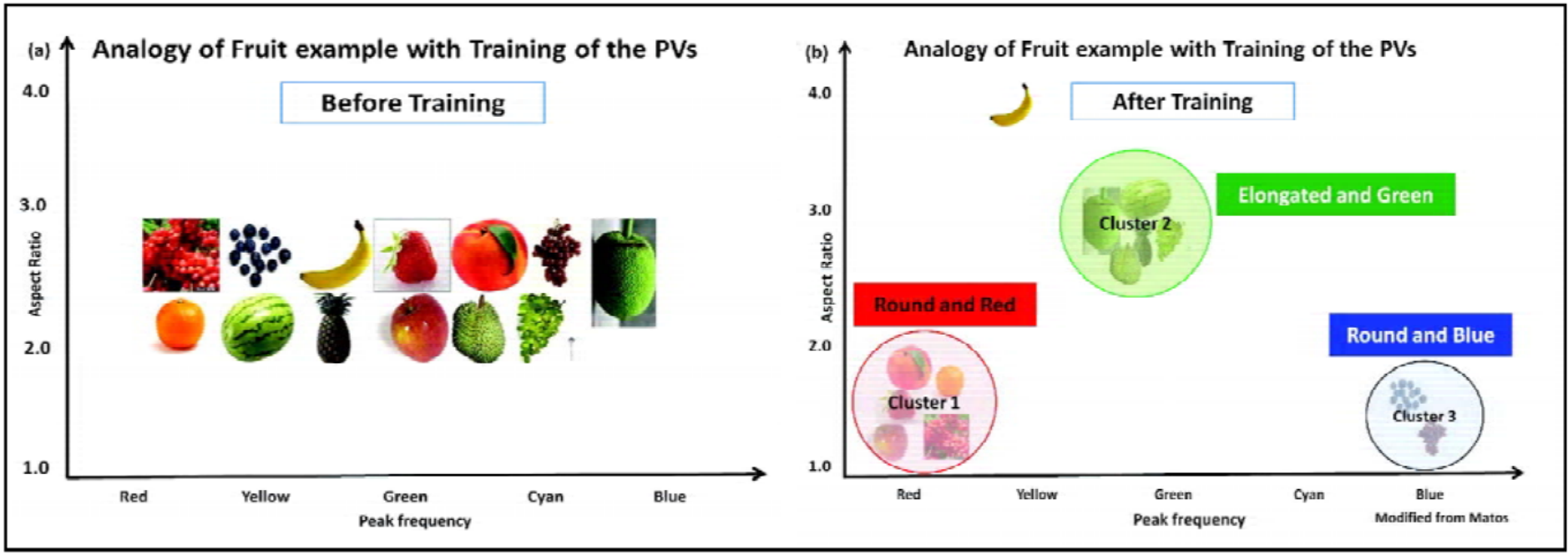


Fig.1 Example of different fruits illustrating unsupervised clustering analysis. (a) The unorganized fruits before training; (b) Clustering of the fruits after training considering their two properties (color and aspect ratio).

https://www.spgindia.org/geohorizon/dec2010/automatic.pdf

SOM

- SOM closely relates to vector quantization methods (Haykin, 1999).
- Input: J vectors in a N dimensional vector space (features)

$$\mathbf{x}_j = [x_{j1}, x_{j2}, ..., x_{jN}]^T$$

- The objective of the algorithm is to organize the input dataset into a geometric structure called the SOM.
- SOM consists of neurons or Prototype Vectors (PVs) organized by a lower dimension grid, usually 2D, which are representative of the input data and have the same dimension of the input features.
- Considering 2D SOM represented by P prototype vectors

$$\mathbf{m}_i = [m_{i1}, m_{i2}, ..., m_{iN}]^T, \quad i = 1, 2, .., P$$

SOM

- After assigning the PVs, we train these PVs (i.e., changing their individual weight values) in accordance to the input data. After SOM neighborhood training these PVs become clusters representing different classes of the input dataset.
- Figure 1(a) shows some fruits that have different properties. Let us consider two of their properties: their aspect ratio (shape) and peak frequency (color) and group these fruits according to these properties.
- After training, the fruits are arranged in three major groups; round and red fruits, elongated and green, round and blue, as shown in Figure 1(b).
- Now if we draw an analogy with a 2D SOM grid map, PVs represent the fruits; the shape and the color of the fruits are the attributes (features) of the PVs and these different attributes help to arrange them into different groups of clusters.

SOM

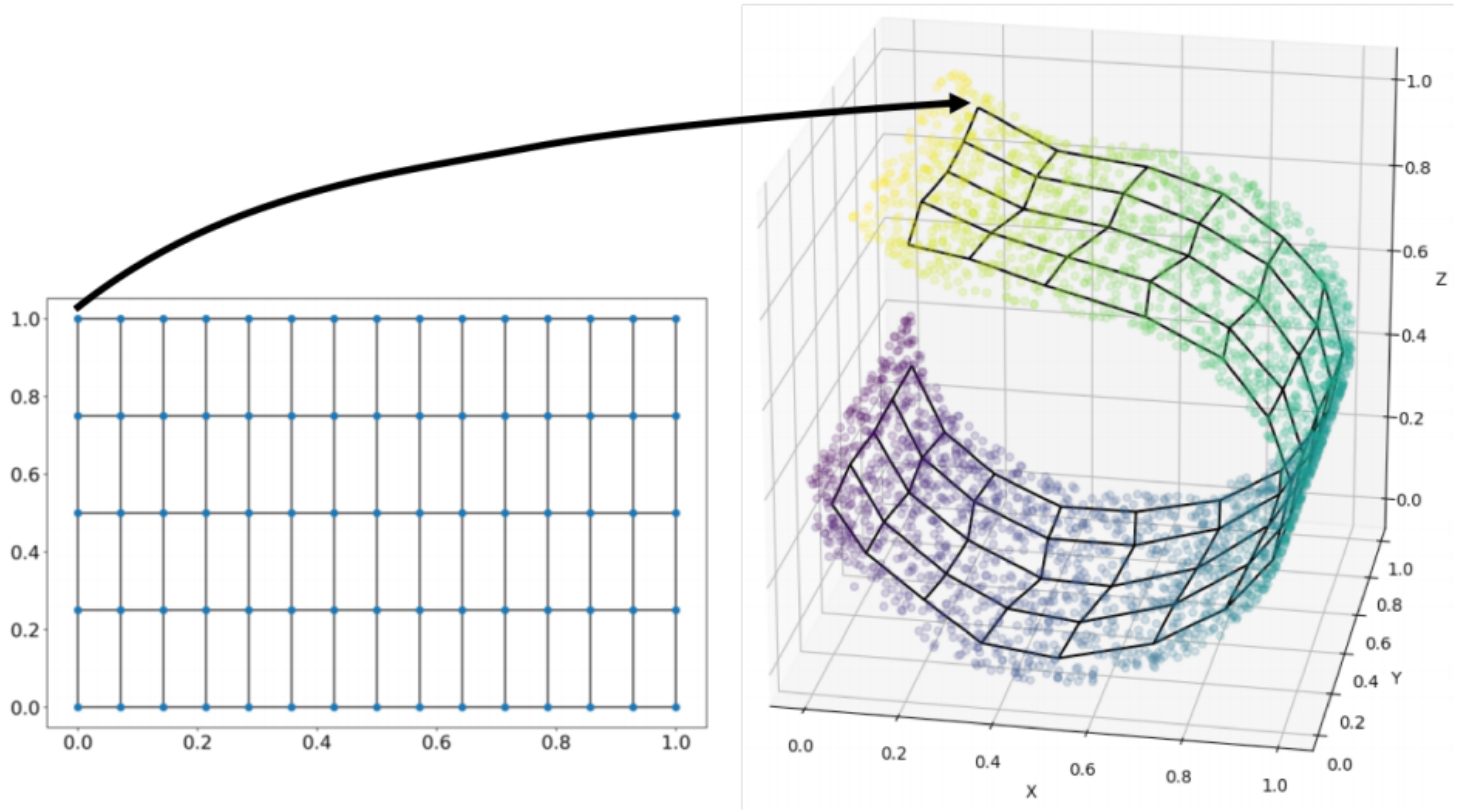


Figure 1: On the left: A plot of a 15×5 lattice in the lattice space, $\mathbb{R}^2$, the black lines indicate neighbors and the blue points indicate the location of the node in lattice space. On the right: a plot in the input data space, $\mathbb{R}^3$, of a SOM trained with the lattice on the left, fit to 3-D data that has a non-linear relationship. The circles are data points, the color of each data point is its position along a non-linear manifold. The arrow shows the connection between a node's lattice vector at $(1, 0)$ in the lattice space and weight vector at $(0.17, 0.88, 0.94)$ in the data space.

LLNL

report 2019 by Ponmalai and Kamath LLNL-TR-791165

SOM

Algorithm 1: Main Idea Behind the SOM Training Algorithm

initialize lattice nodes;
initialize weight vectors;
$N \leftarrow$ Iteration count;
for $i \leftarrow 1$ **to** N **do**
    $x \leftarrow$ pick a random point in the dataset;
    $c \leftarrow$ the lattice node closest to x;
    move the weight vector of c closer to x;
    move the weight vectors of the neighbors of c slightly closer to x;
end

LLNL report 2019 by Ponmalai and Kamath LLNL-TR-791165

SOM

a problem and decreases with each iteration t; r_b and r_i are the

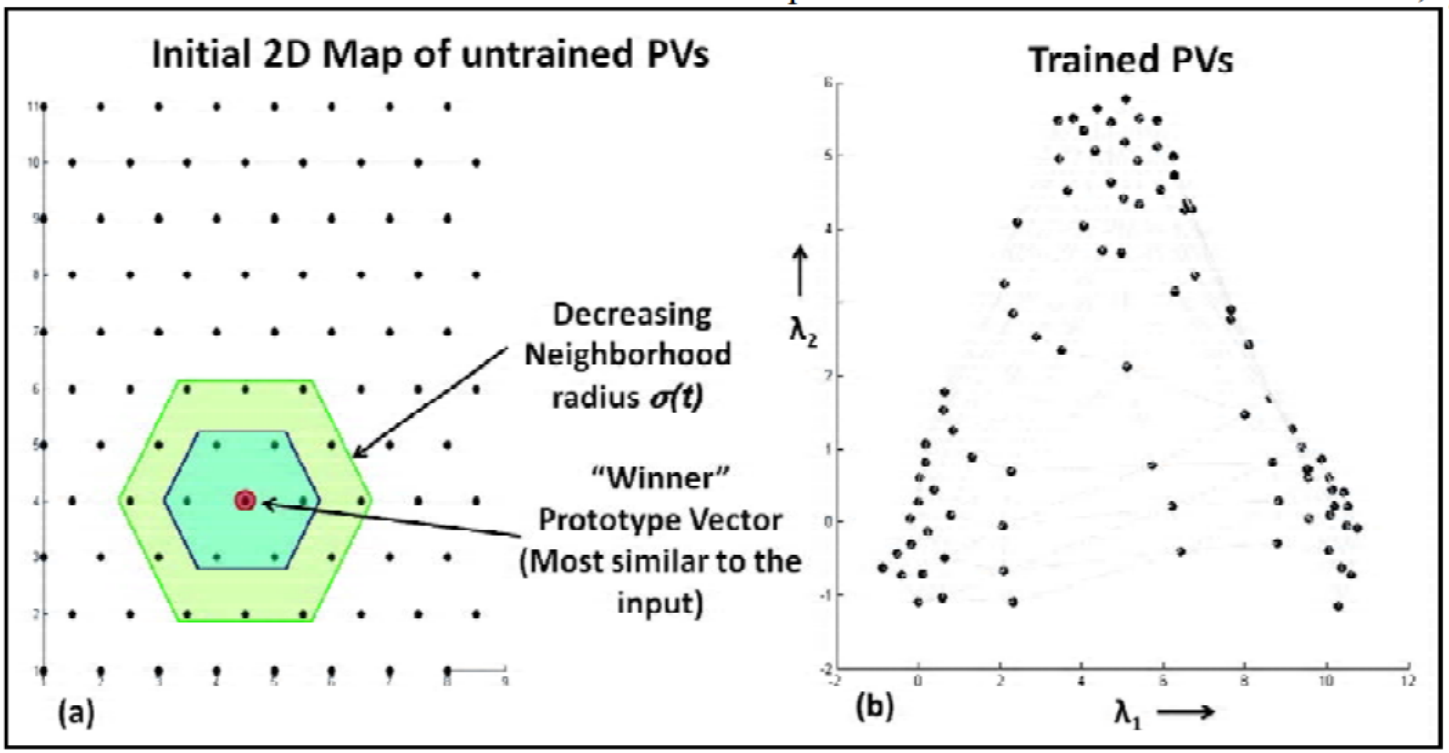


Fig. 3 2D map of the PVs (a) before and (b) after the training. Around the "Winner PV" the larger green hexagon represents a neighborhood at an early iteration while the smaller blue hexagon represents a neighborhood at a later iteration. The results can be interpreted as forming three clusters, thus classifying the input data properly.

GEOHORIZONS December 2010/8

SOM

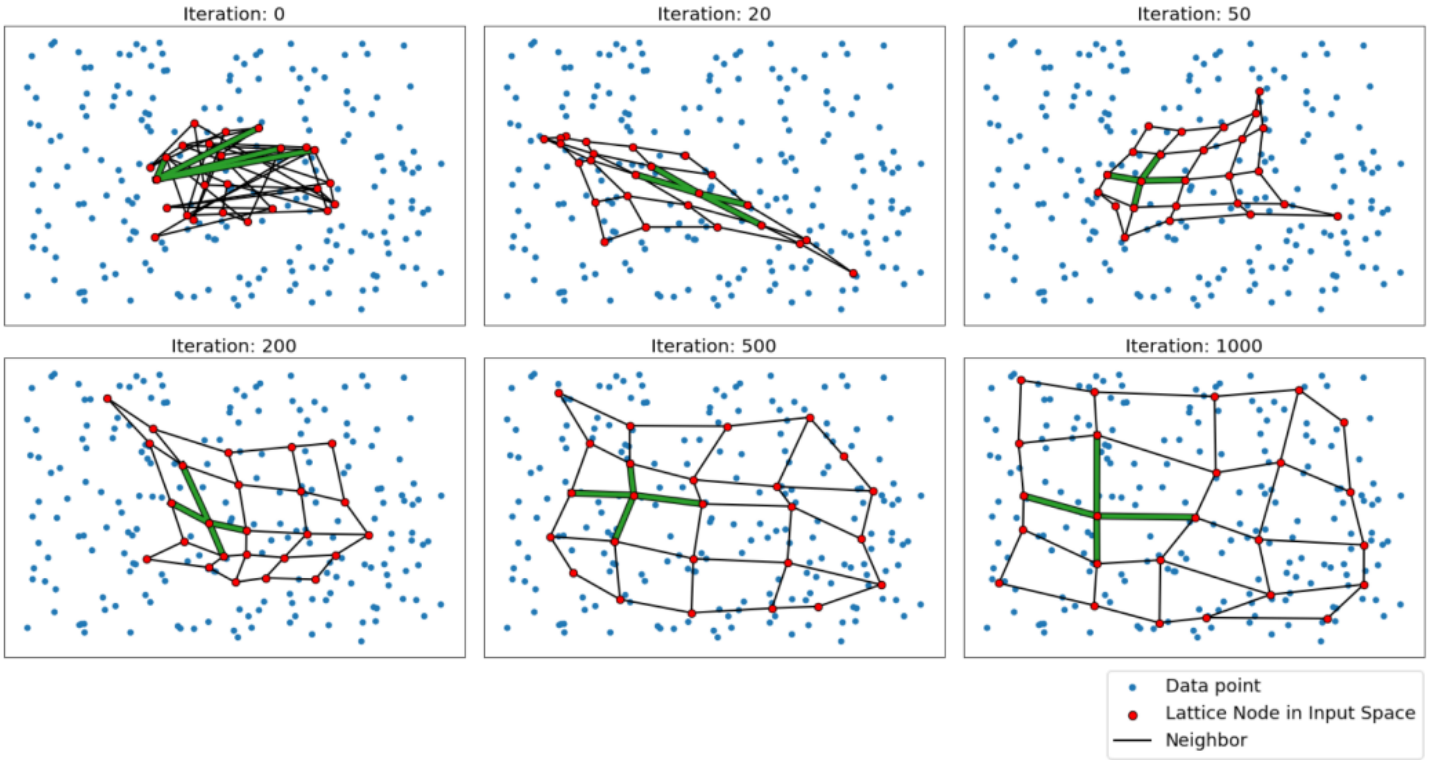


Figure 2: The training of a 5×5 SOM on uniform random points for 2000 iterations. A more in-depth look at SOM training can be found in Section 3. The green segments highlight the neighbors of one lattice node.

SOM

- We can use a trained SOM to understand how different clusters of data are distributed through the data space.
- The number of nodes in a SOM lattice can have a huge effect on the resulting map. Kohonen's [11] advice is to use $5\sqrt{n}$ nodes where n is the number of data points. If a rectangular map is used, Kohonen also suggests that the optimal ratio of height to width of the lattice is equal to the ratio of the two largest eigenvalues of the autocorrelation matrix.
- Input data should be rescaled to ensure each feature has the same scale. In practice we have found it simplest to rescale each feature to be between 0 and 1. Outliers that are not representative of the dataset should also be removed.
- Neighborhood Function Choices:

- $h_{i,j}$ should attain its maximum when $\|l_i - l_j\| = 0$ and be symmetric about the winning node j.
- As $\|l_i - l_j\|$ increases, $h_{i,j}$ should decrease.

$$h_{i,j} \to 0 \text{ as } \|l_i - l_j\| \to \infty$$

Any function that meets these requirements is a valid neighborhood function, below are some examples. These functions are visualized in Figure 5.

- Gaussian - $h_{i,c(x)} = \exp\left(-\frac{\|l_i - l_{c(x)}\|}{2\sigma(k)^2}\right)$
- Step/Bubble - $h_{i,c(x)} = \begin{cases} 1 & \|l_i - l_{c(x)}\| \leq \sigma(k) \\ 0 & \text{otherwise} \end{cases}$
- Triangle - $h_{i,c(x)} = \begin{cases} 1 - \frac{\|l_i - l_{c(x)}\|}{\sigma(k)} & \|l_i - l_{c(x)}\| \leq \sigma(k) \\ 0 & \text{otherwise} \end{cases}$
- Gaussian with cutoff - $h_{i,c(x)} = \begin{cases} \exp\left(-\frac{\|l_i - l_{c(x)}\|}{2\sigma(k)^2}\right) & \|l_i - l_{c(x)}\| \leq \sigma(k) \\ 0 & \text{otherwise} \end{cases}$

SOM

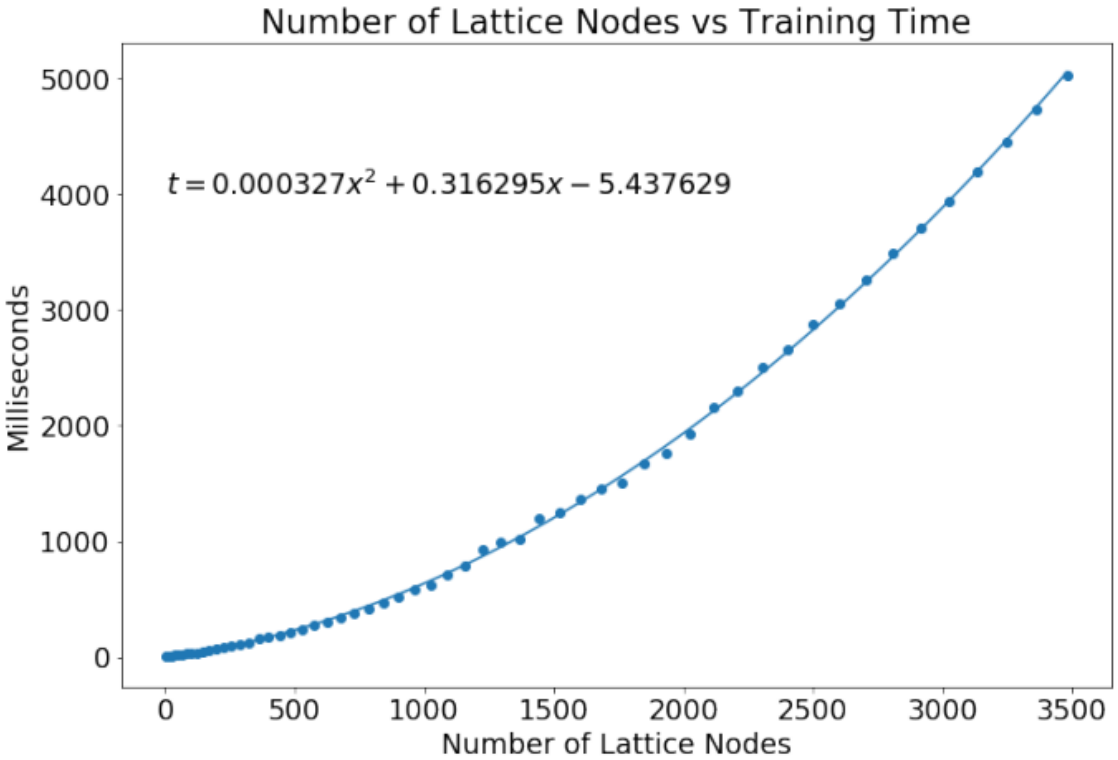


Figure 6: Training time for 100 iterations of SOMs with varying numbers of lattice nodes. A quadratic was fit to the data and the equation of the quadratic is shown.

LLNL

report 2019 by Ponmalai and Kamath LLNL-TR-791165

SOM: U Matrix

- A common way to display SOMs is the heat map of U-matrix. The U-matrix value of a particular node is the minimum/maximum/average distance between the node and its closest neighbors.
- U-matrix (unified distance matrix) representation of the Self-Organizing Map visualizes the distances between the neurons.
- The distance between the adjacent neuons is calculated and presented with different colorings between the adjacent nodes.
- A dark coloring between the neurons corresponds to a large distance and thus a gap between the codebook values in the input space. A light coloring between the neurons signifies that the codebook vectors are close to each other in the input space.
- Light areas can be thought as clusters and dark areas as cluster separators.
- This can be a helpful presentation when one tries to find clusters in the input data without having any a priori information about the clusters.

SOM

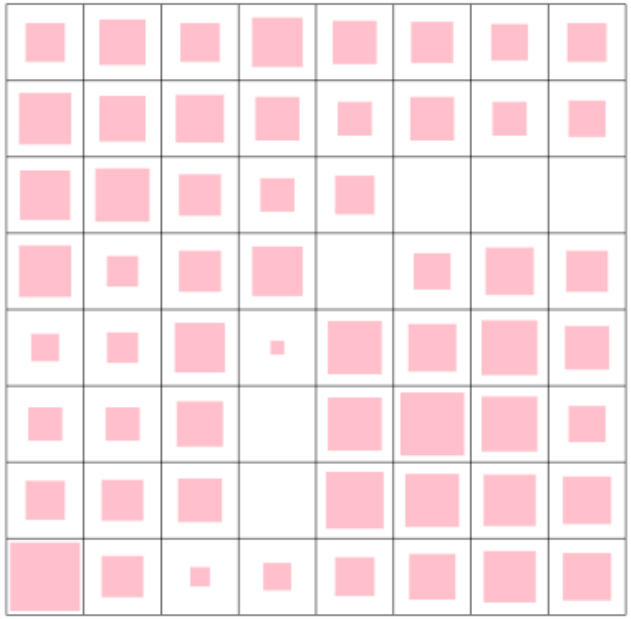


Figure 5. Frequency map of neurons in an 8 x 8 SOM grid.

https://algobeans.com/2017/11/02/self-organizing-map/

SOM

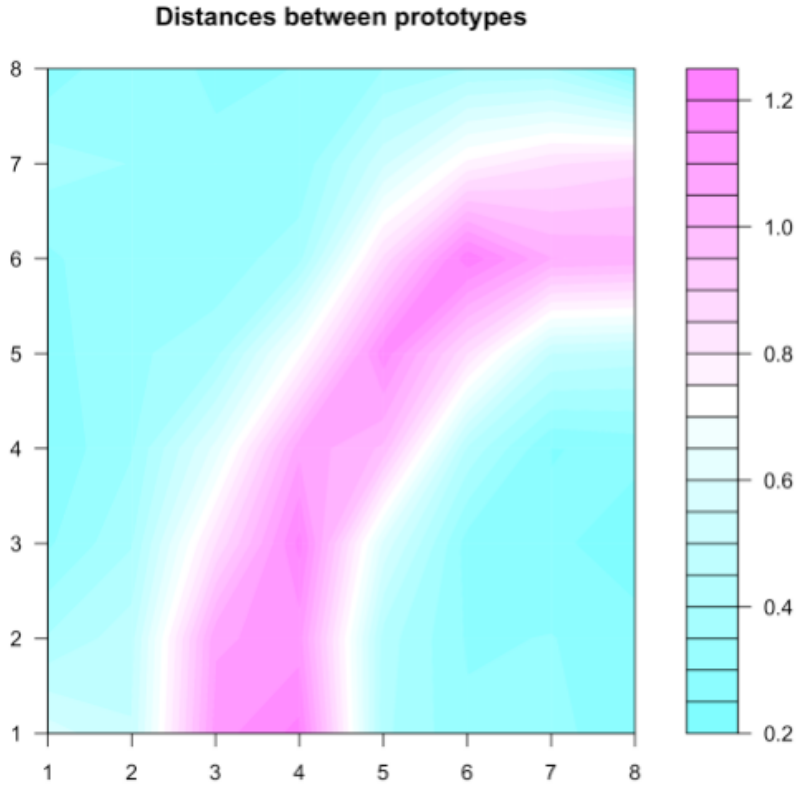


Figure 6. U-matrix showing similarity and dissimilarity between neurons. Useful for identifying clusters.

https://algobeans.com/2017/11/02/self-organizing-map/

K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified

k-Means

input: $\mathcal{X} \subset \mathbb{R}^n$; Number of clusters k
initialize: Randomly choose initial centroids $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$
repeat until convergence
$\forall i \in [k]$ set $C_i = \{\mathbf{x} \in \mathcal{X} : i = \text{argmin}_j \|\mathbf{x} - \boldsymbol{\mu}_j\|\}$
(break ties in some arbitrary manner)
$\forall i \in [k]$ update $\boldsymbol{\mu}_i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}$

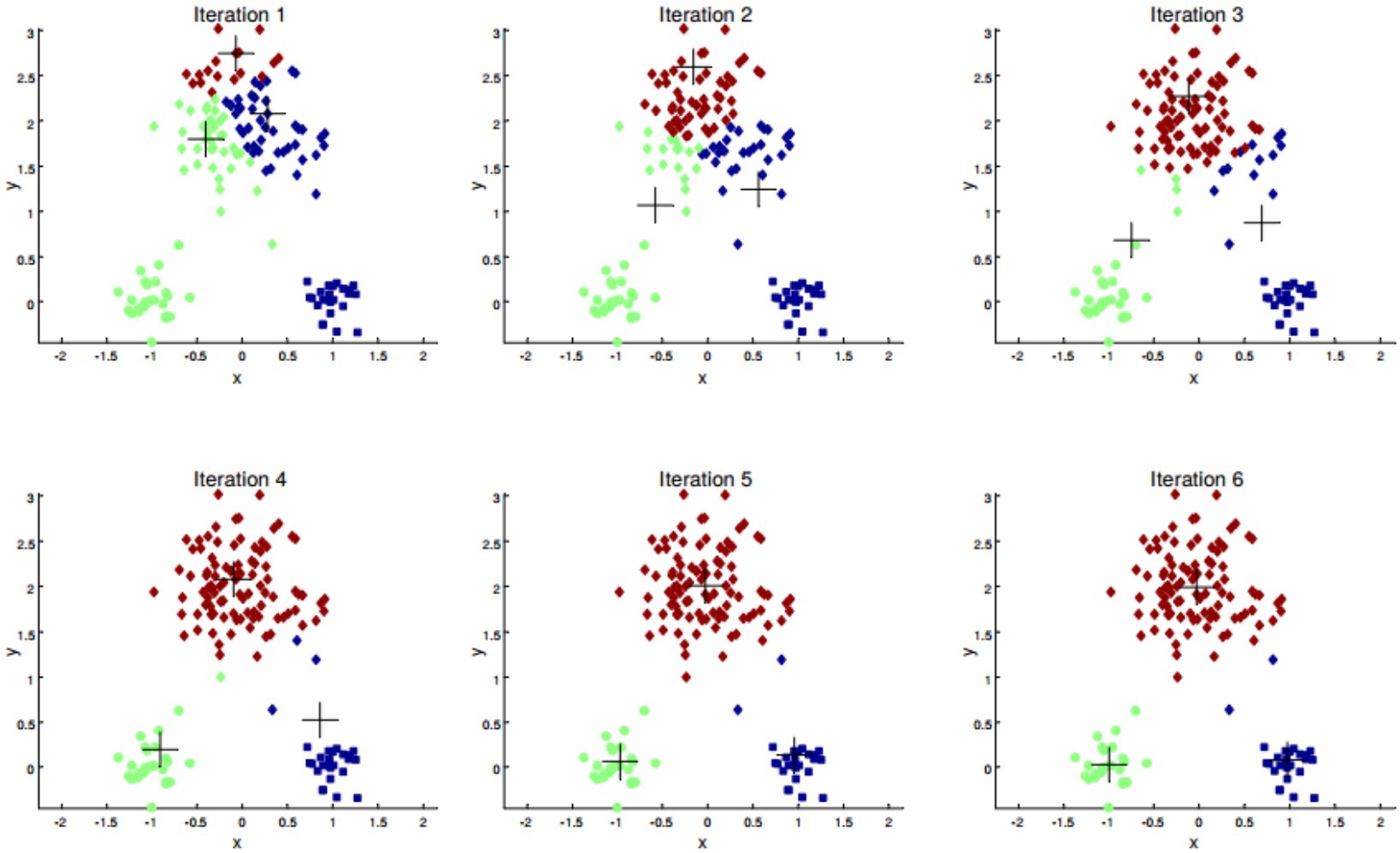
K-means

- Divide a data set $\chi = \{\mathbf{x}_t; t = 1, ..., T\}$ into k groups, each represented by its centroid denoted by $\mu_j, j = 1, ..., k$.
- Two tasks:
  - Determine k centroids $\{\mu_1, ..., \mu_k\}$,
  - Assign each $\mathbf{x}_t$ to one of the centroids.
- Centroid $\mathbf{c}_j \in \{\mu_1, ..., \mu_k\}$.
- Minimiize the sum of the squared errors

K-means

- Initial Centroids are often chosen randomly. Clusters produced vary from one run to another.
- The centroid is the mean of the points in the cluster.
- 'Closeness' is measured by Euclidean distance
- K-means will converge (points stop changing assignment) typically in the first few iterations. Often the stopping condition is changed to 'Until relatively few points change clusters'
- Complexity is O(n * k * I * d)
  - n = number of points, K = number of clusters,
  - I = number of iterations, d = number of attributes.

K-means Example



Lecture

Notes for Chapter 8: Slides by Tan, Steinbach, Kumar adapted by Michael Hahsler

K-means: Choosing Initial Centroids

- Multiple runs
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids the ones that are far away from each other.

K-means: Processing

- Pre-processing
  - Normalize the data (e.g., scale to unit standard deviation)
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split 'loose' clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are 'close' and that have relatively low SSE
- K-means has problems when clusters are of differing Sizes, and Densities
- K-means has problems when the data contains outliers.

Gaussian Mixture Models

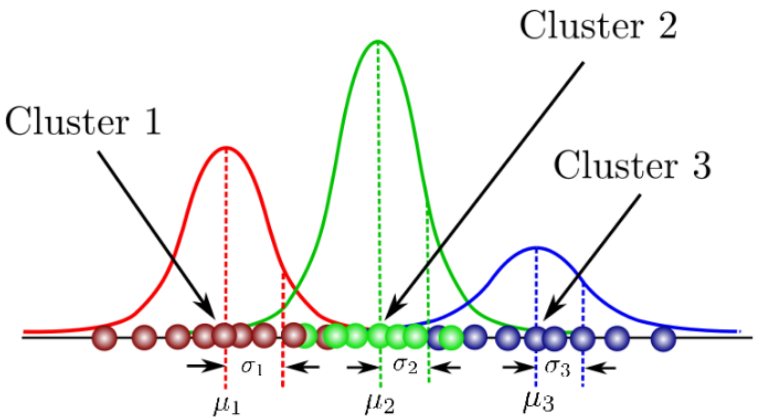
- K-means is a **hard clustering method**.
- It will associate each point to one and only one cluster.
- There is no uncertainty measure or probability that tells us how much a data point is associated with a specific cluster.
- We may want to use a soft clustering rather than a hard one! Mixture Models (GMM) attempt to do that.

Multi-variate Gaussian

$$N(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

Gaussian Mixture Models

- A Gaussian Mixture is a function of several Gaussians, each identified by $k \in 1, \ldots, K$, where K is the number of clusters of our dataset. Each Gaussian k in the mixture is comprised of the following parameters:
  - A mean μ that defines its centre.
  - A covariance Σ that defines its width. This would be equivalent to the dimensions of an ellipsoid in a multivariate scenario.
  - A mixing probability *wi* that defines how big or small the Gaussian function will be.



https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95

Gaussian Mixture Models

- A GMM is a linear weighted sum of K Gaussian densities:

$$p(\mathbf{x}) = \sum_{k=1}^{K} w_k N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$w_k \equiv Pr(\text{mix} = k)$$

$$N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = -\frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right)$$

$$\sum_{k=1}^{K} w_k = 1$$

.

Gaussian Mixture Models

- GMM with 3 mixtures ($K = 3$):

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

$$p(\mathbf{x}) = \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

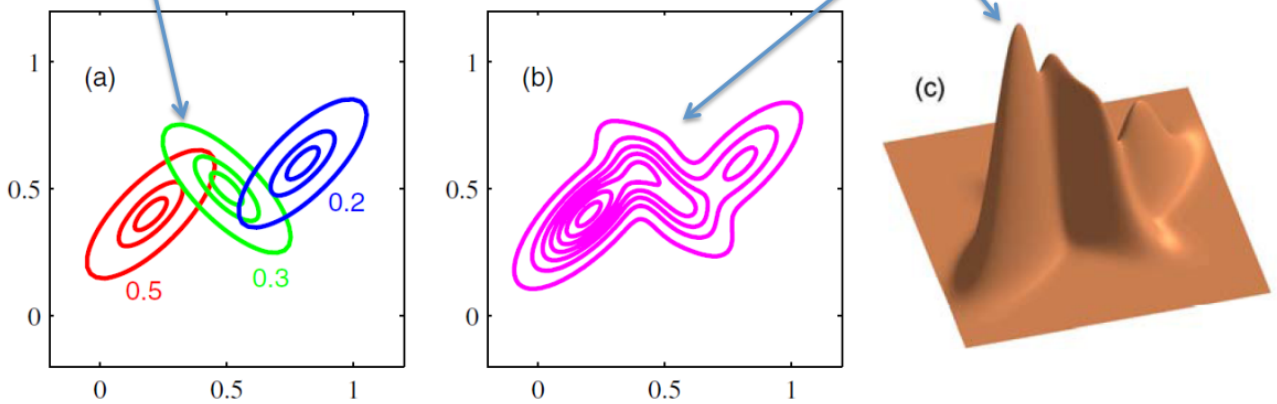
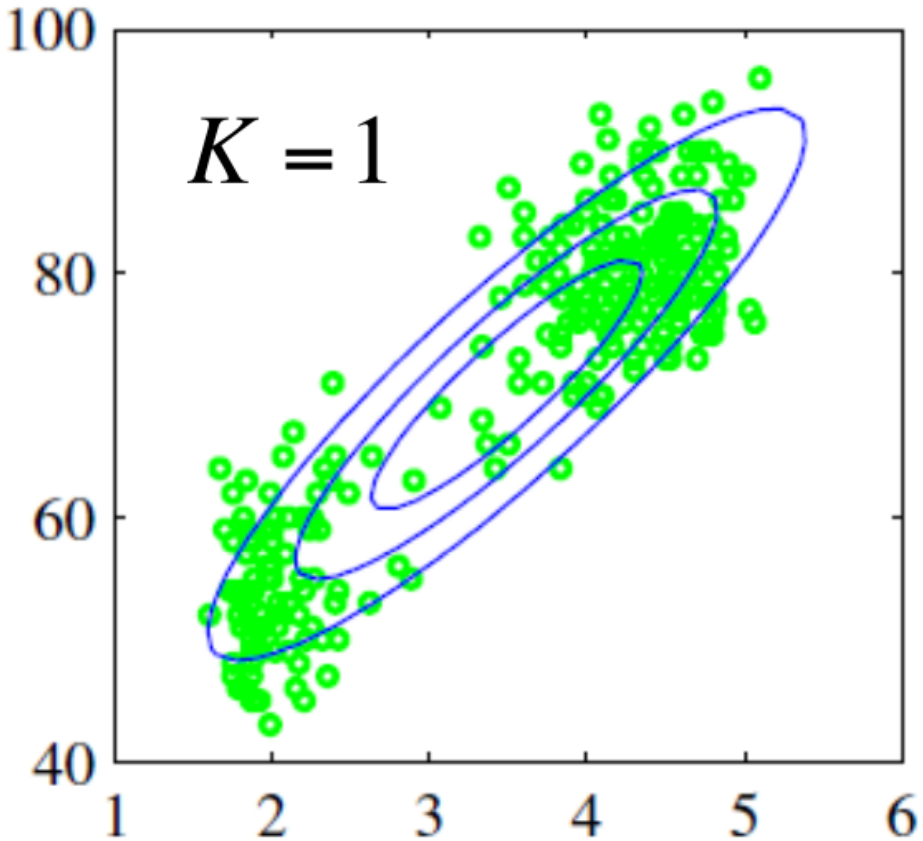
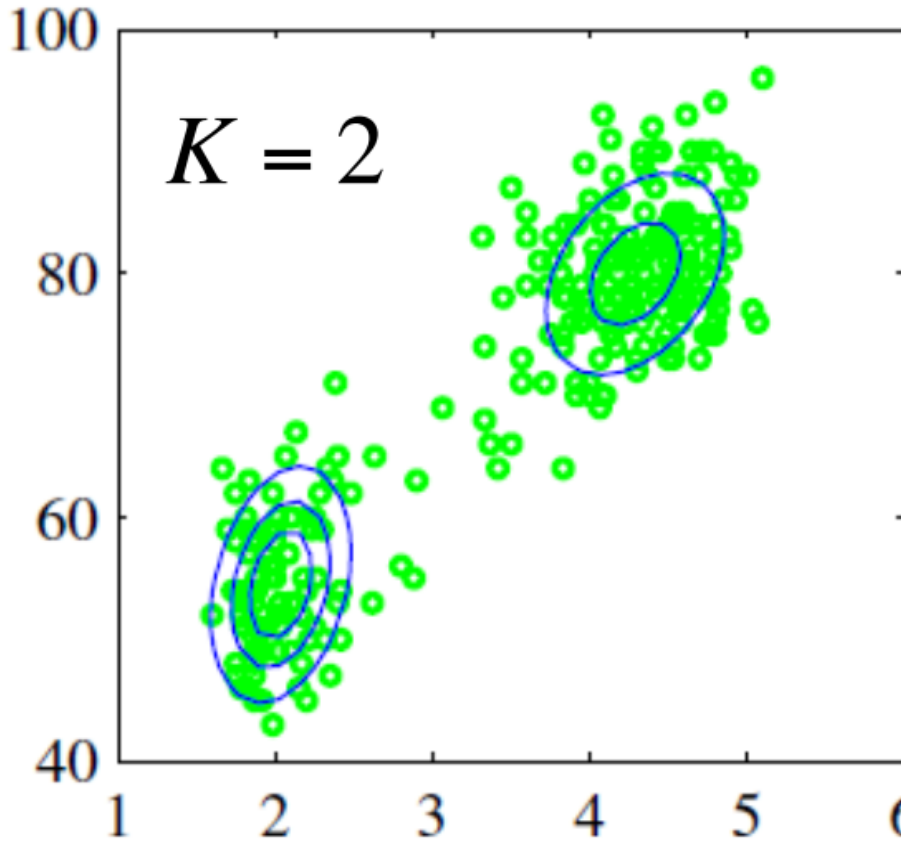


Figure 2.23 Illustration of a mixture of 3 Gaussians in a two-dimensional space. (a) Contours of constant density for each of the mixture components, in which the 3 components are denoted red, blue and green, and the values of the mixing coefficients are shown below each component. (b) Contours of the marginal probability density $p(\mathbf{x})$ of the mixture distribution. (c) A surface plot of the distribution $p(\mathbf{x})$.

http://www.eie.polyu.edu.hk/mwmak

GMM Clustering





http://www.eie.polyu.edu.hk/mwmak

Gaussian Mixtures

- A probabilistic approach to clustering is to assume that the data is generated from a set of K simple distributions, in our current case, Gaussians:

$$p(\mathbf{x}) = \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

  w_k is the probability that a sample is drawn from k-th mixture component, $\sum_{k=1}^{K} w_k = 1,\ w_k > 0$.

- Equivalently, you may also consider this to be introducing a hidden variable $z \in \{1, 2, ..., K\}$, indicating that each component comes from

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(z = k) p(\mathbf{x}|z = k);$$

$$p(z = k) = w_k,\ p(\mathbf{x}|z = k) = \mathcal{N}(\mu_k, \Sigma_k).$$

Gaussian Mixtures

- Given data $D = \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$.
- Goal: estimate the parameters of the mixture $(w_k, \mu_k, \Sigma_k);\ k = 1, 2, ..., K$
- We use MLE (or MAP)

$$\log\ P(D|w, \mu, \Sigma) = \sum_i \log \sum_k p_k(z_i = k) p_{\mu,\Sigma}(\mathbf{x}_i|z_i = k).$$

Gaussian Mixtures

Variance Parameters

- Full covariance: Σ_k is arbitrary for each class (clusters are general ellipsoids)
- Shared Full covariance: Σ_k is arbitrary but same for each class (all clusters have same ellipsoid shape,
- Diagonal (aka Naive Bayes): Σ_k is a diagonal matrix (clusters are axis-aligned ellipsoids)
- Shared Diagonal: Σ_k is a diagonal matrix but same for each class (all clusters have same axis-aligned ellipsoid shape)
- Spherical: $\Sigma_k = \sigma_k \, I$
- Shared Spherical: $\Sigma_k = \sigma \, I$

Mixture Parameters

- Equal Weights: $w_k = 1/K$.
- Arbitrary Weights

K Means

- K-means can be seen as trying to find the values of hidden variables z_i(cluster assignments), and variances are shared spherical, $\Sigma_k = \sigma\, I$ (the value of σ doesn't change the optimal $\mathbf{z}$, andμ, so we can just assume it's fixed).

$$\arg\max_{\mathbf{z},\mu} \log P(D, \mathbf{z}|w, \mu,) = \arg\max_{\mathbf{z},\mu} \sum \log(p_w(z_i))p_{\mu,}(\mathbf{x}_i|z_i))$$

$$= \arg\min_{\mathbf{z},\mu} \sum_i ||\mu_{z_i} - \mathbf{x}_i||^2.$$

Gaussian Mixtures - EM

- Given data $D = \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$.
- Goal: estimate the parameters of the mixture $(w_k, \mu_k, \Sigma_k);\ k = 1, 2, ..., K$
- We use MLE (or MAP)

$$\log\ P(D|w, \mu, \Sigma) = \sum_i \log \sum_k p_k(z_i = k) p_{\mu,\Sigma}(\mathbf{x}_i|z_i = k).$$

- ML does not work here as there is no closed form solution.
- Parameters can be calculated using **Expectation Maximization (EM) technique**

Gaussian Mixtures - EM

Latent Variable (Weight): Posterior Probability

- We can think of the mixing coefficients as prior probabilities for the components.
- For a given value of 'x', we can evaluate the corresponding posterior probabilities, called responsibilities!
- From Bayes' Rule:

$$\gamma_k(\mathbf{x}) = p(k|\mathbf{x}) = \frac{p(k)p(\mathbf{x}|k)}{p(\mathbf{x})} = \frac{w_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} w_j \mathcal{N}(\mathbf{x}|\mu_j, \Sigma_j)}$$

Gaussian Mixtures - EM Algorithm

- Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters comprising the means and covariances of the components and the mixing coefficients).
- Initialize the means μ_j , covariances Σ_j and mixing coefficients w_j, and evaluate the initial value of the log likelihood.
- **E Step:** Evaluate the responsibilities using the current parameter values

$$\gamma_k(\mathbf{x}) = \frac{w_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} w_j \mathcal{N}(\mathbf{x}|\mu_j, \Sigma_j)}$$

Gaussian Mixtures - EM Algorithm

- **M Step**: Re-estimate the parameters using the current responsibilities

$$\mu_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

$$\Sigma_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)(\mathbf{x}_n - \mu_j)(\mathbf{x}_n - \mu_j)^T}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

$$w_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(\mathbf{x}_n).$$

- Evaluate the likelihood

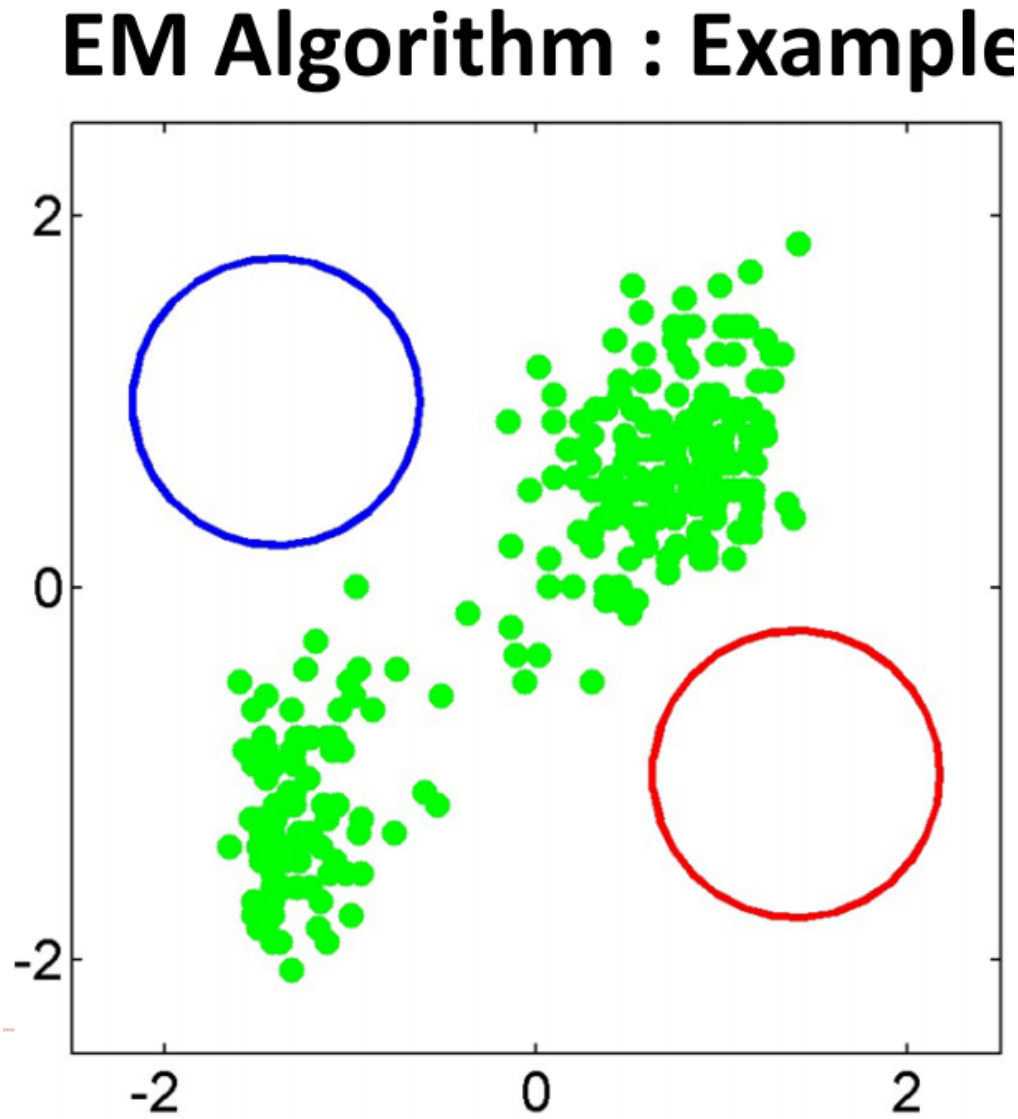
$$\log\ p(D|\mu.\Sigma, w) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k) \right\}$$

GMM training - Expectation Maximization or EM

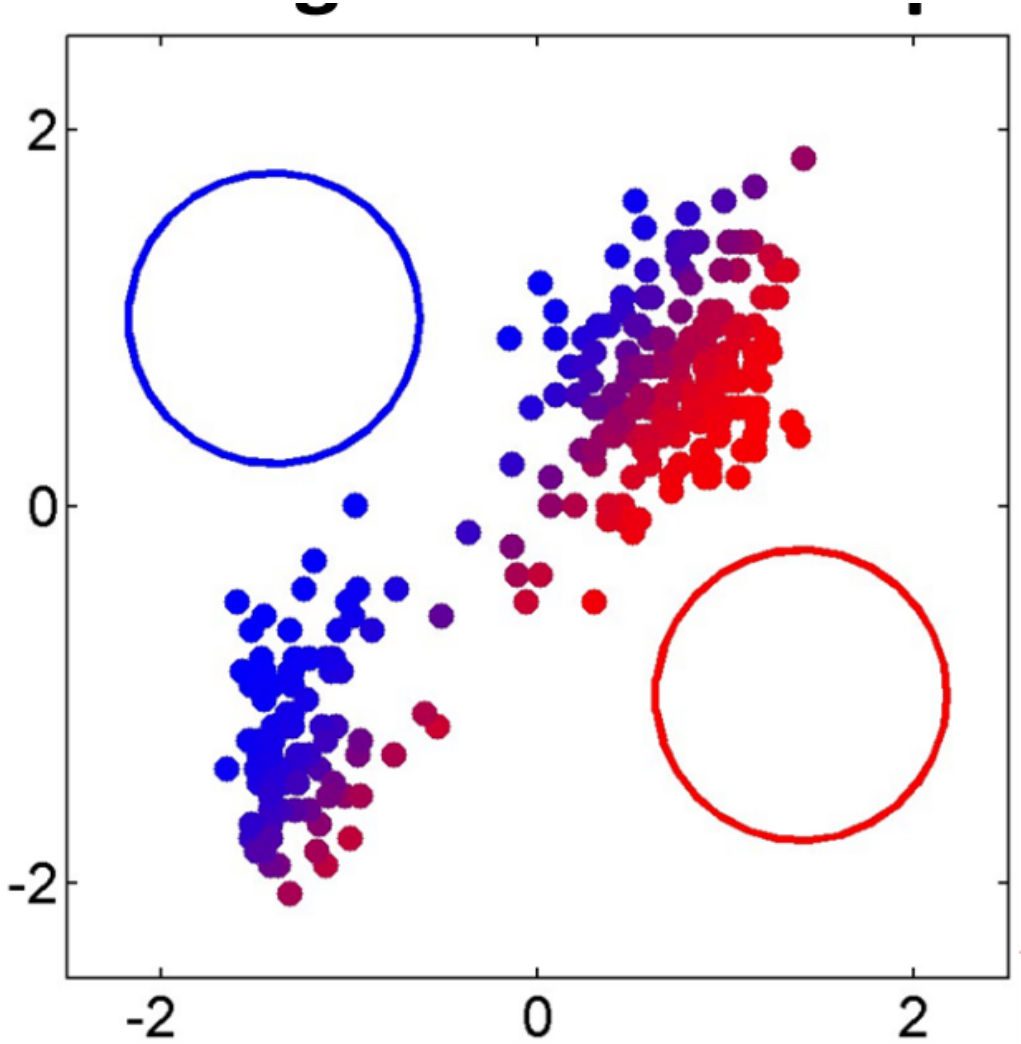
- In summary, the EM algorithm iteratively performs the following:
  - **Initialization**: Randomly select K samples from $\mathcal{X}$ and assign them to $\{\boldsymbol{\mu}_k\}_{k=1}^K$. Set $w_k = \frac{1}{K}$ and $\boldsymbol{\Sigma}_k = \mathbf{I}$, where $k = 1, \ldots, K$.
  - **E-Step**: Find the posterior distribution of the latent (unobserved) variables, given the observed data and the current estimate of the parameters;
  - **M-Step**: Re-estimates the parameters to maximize the likelihood of the observed data, under the assumption that the distribution found in the E-step is correct.
- The iterative process guarantees to increases the true likelihood or leaves it unchanged (if a local maximum has already been reached).

http://www.eie.polyu.edu.hk/mwmak

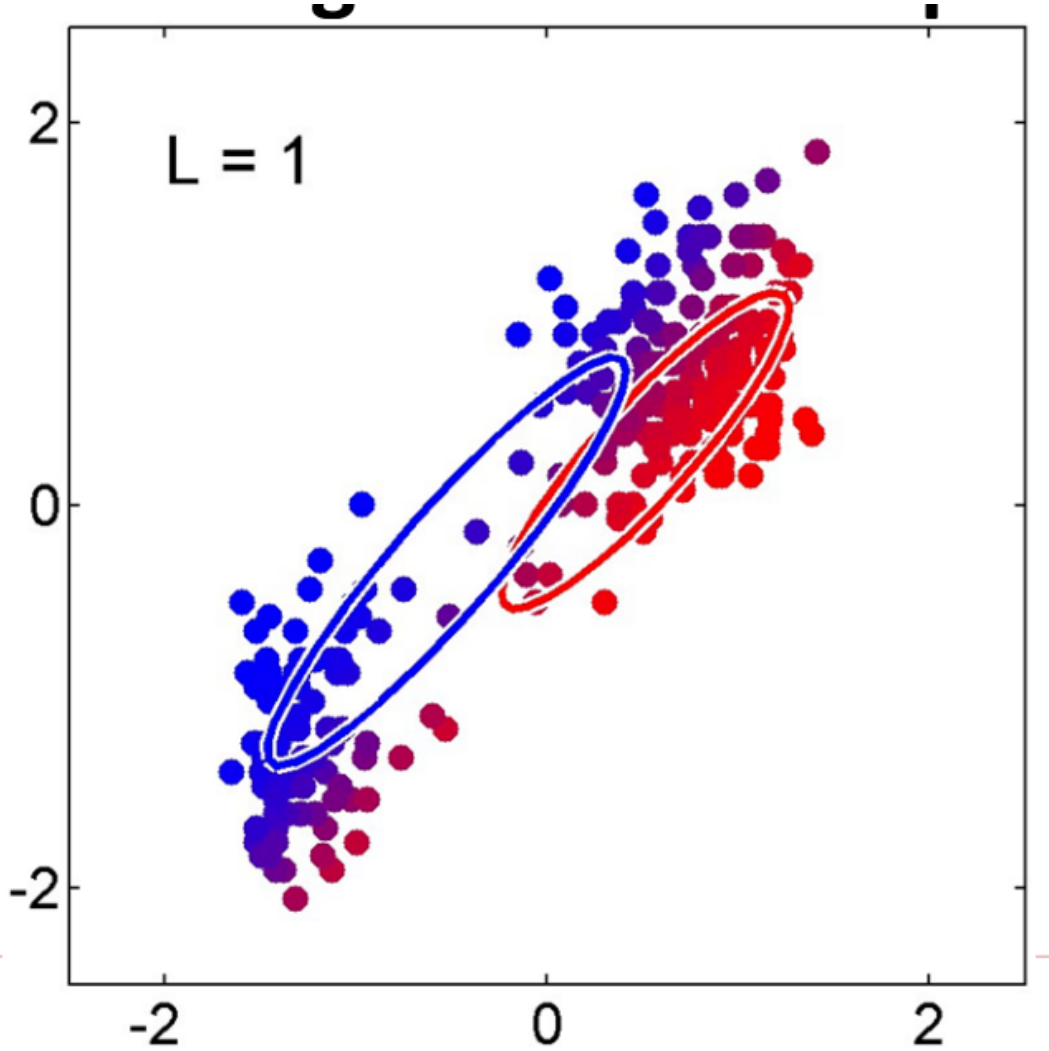
GMM training - Expectation Maximization or EM



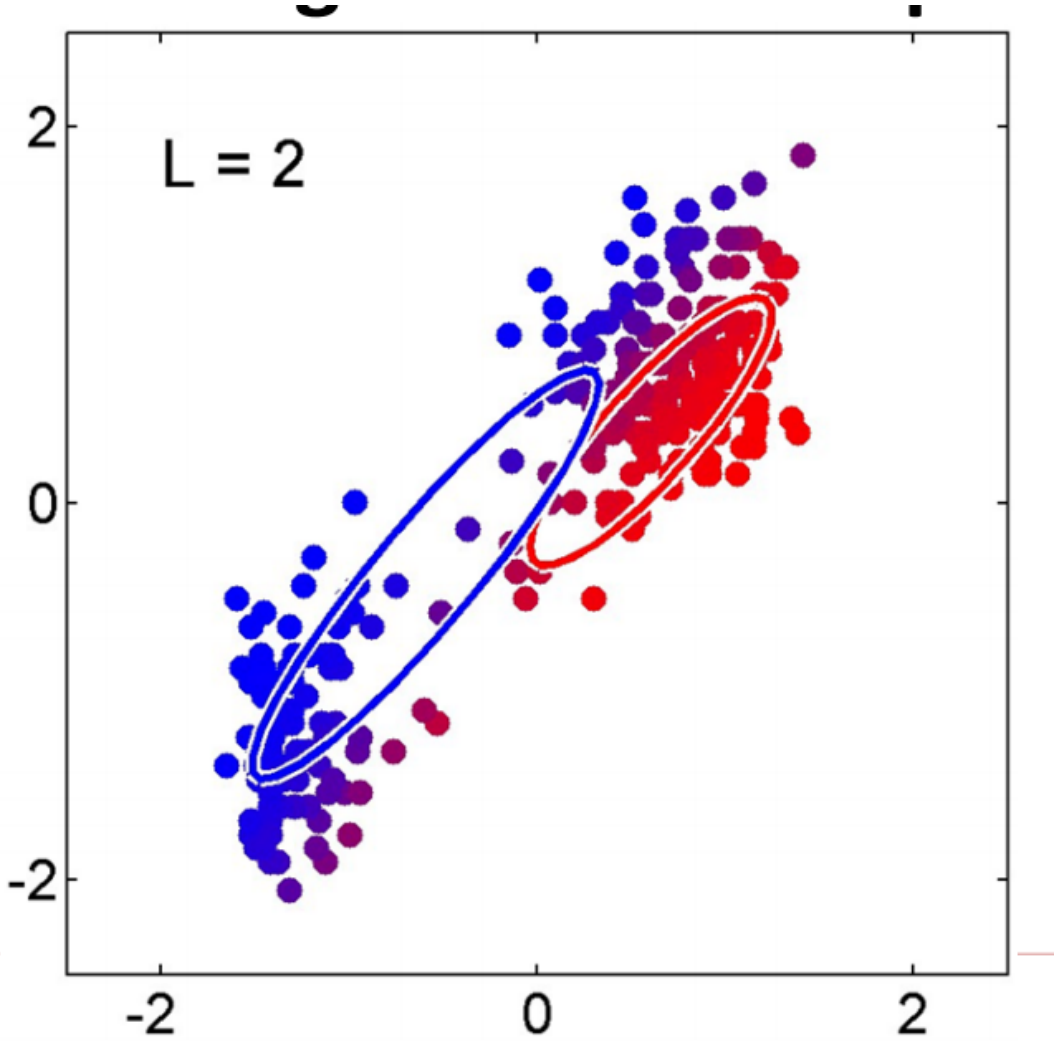
GMM training - Expectation Maximization or EM



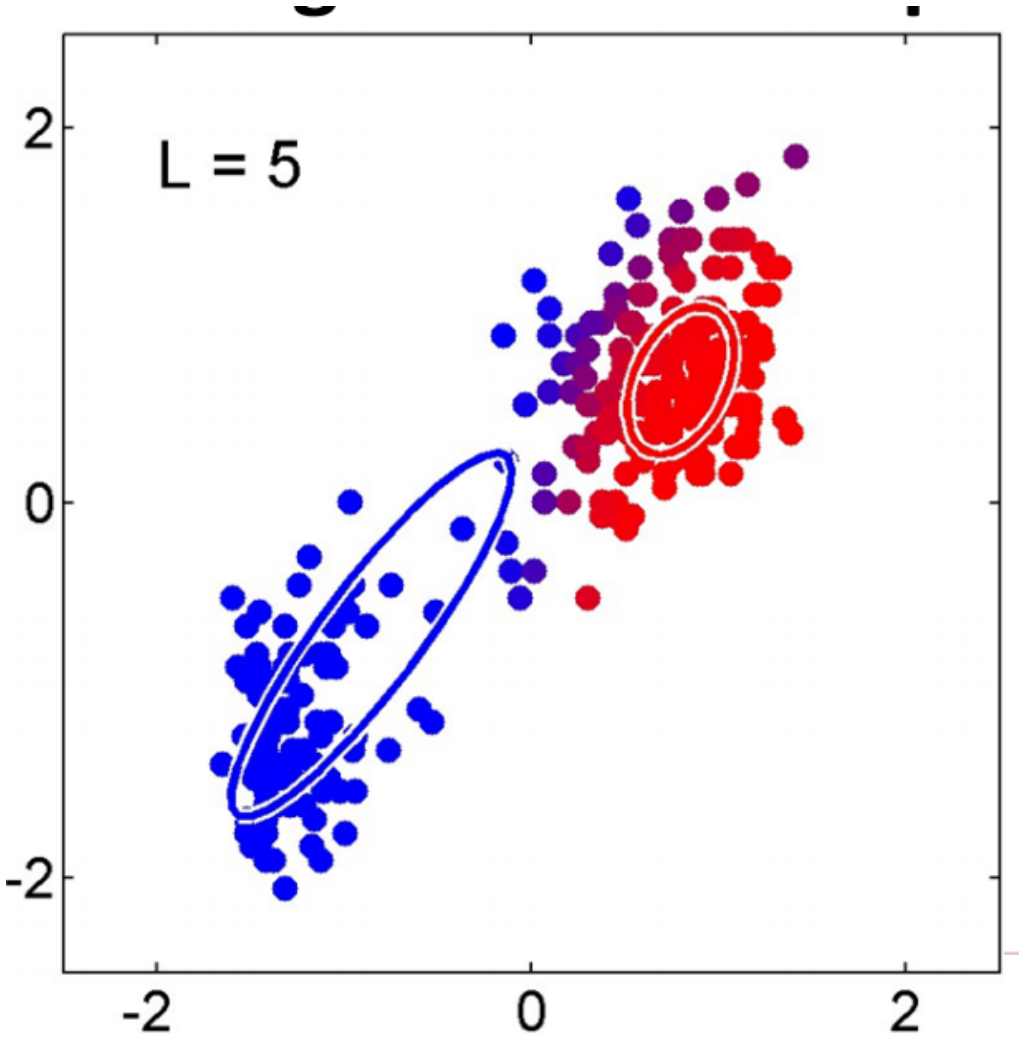
GMM training - Expectation Maximization or EM



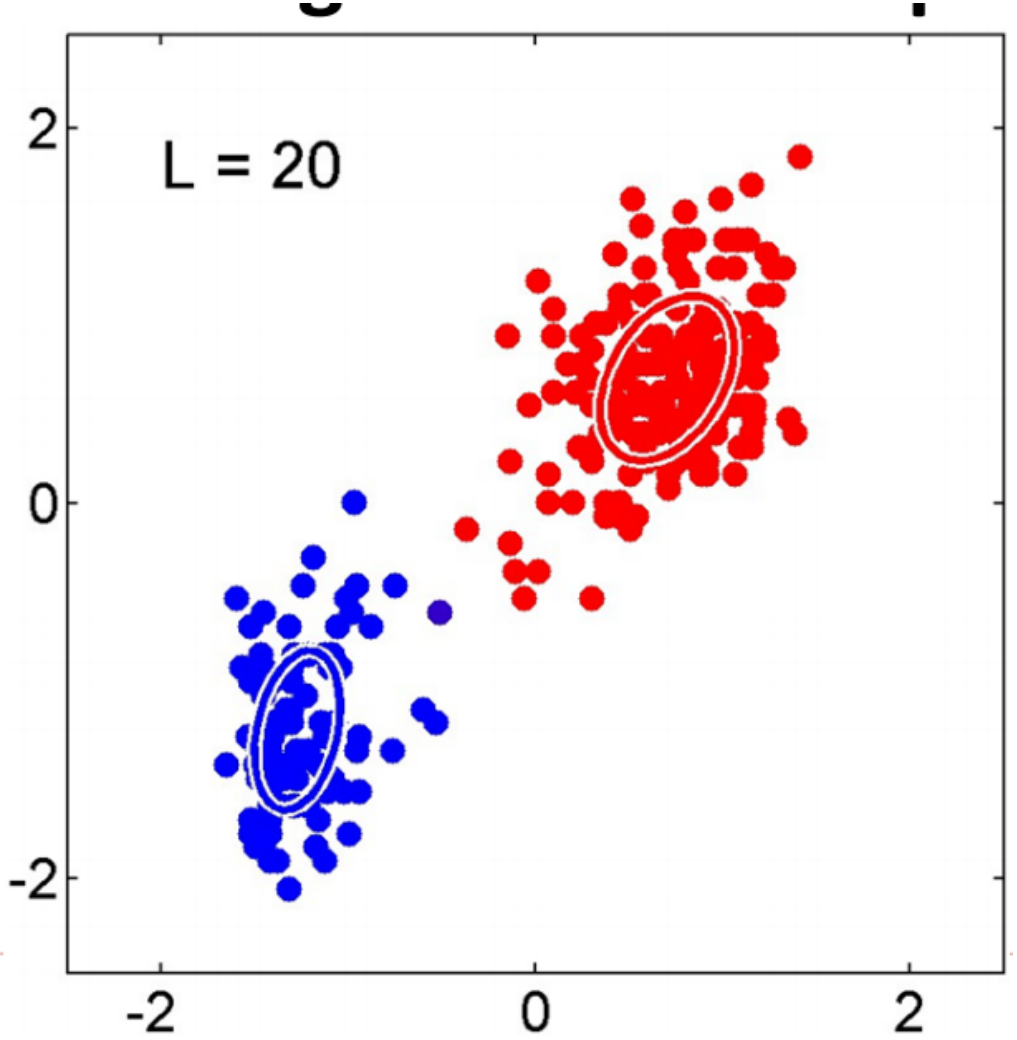
GMM training - Expectation Maximization or EM



GMM training - Expectation Maximization or EM



GMM training - Expectation Maximization or EM

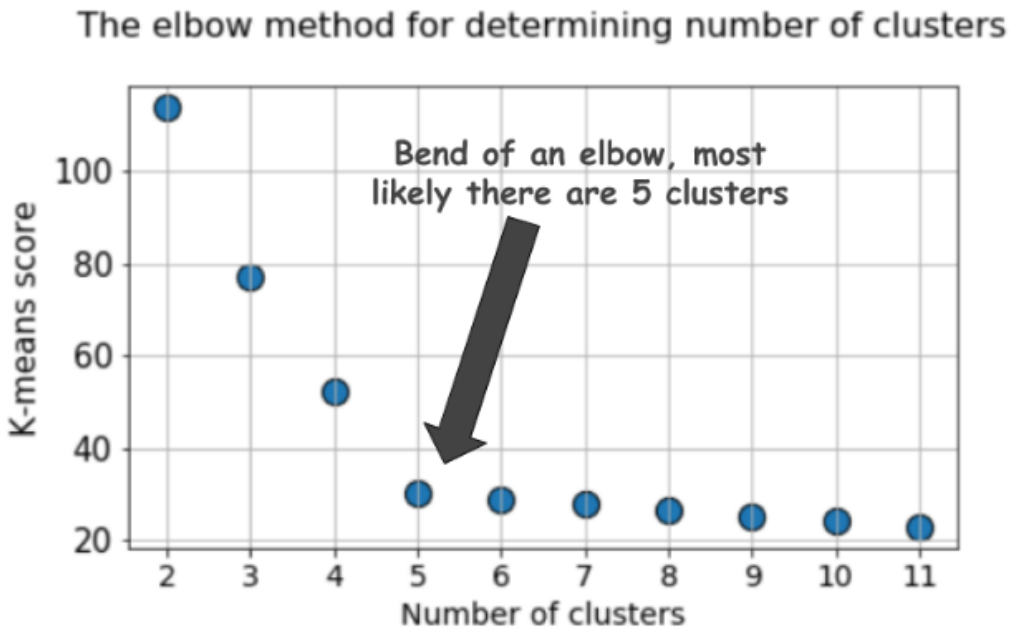


Cluster Validity

- For supervised classification we have a variety of measures to evaluate how good our model is: Accuracy, precision, recall
- For cluster analysis, the analogous question is how to evaluate the "goodness" of the resulting clusters?
- "clusters are in the eyes of the beholder!"
- Then why do we want to evaluate them?
- To avoid finding patterns in noise
- To compare clustering algorithms
- To compare two sets of clusters
- To compare two clusters

Elbow Method

- The most common approach
- It involves running the algorithm multiple times over a loop, with an increasing number of cluster choice
- plotting a clustering score as a function of the number of clusters.
- The score is, in general, a measure of the input data on the k-means objective function i.e. some form of intra-cluster distance relative to inner-cluster distance.

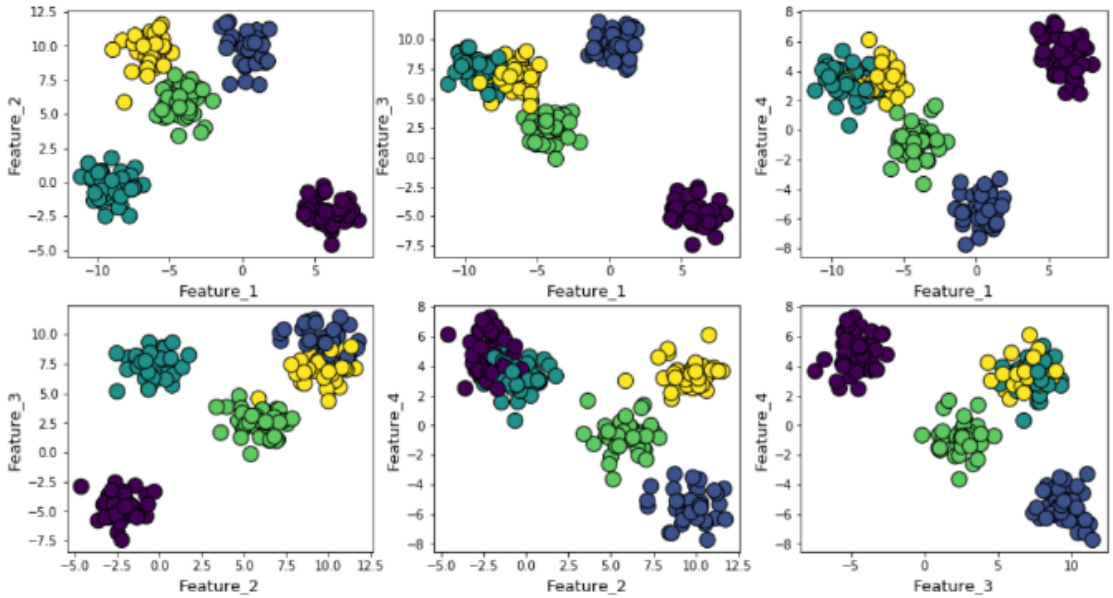


Silhouette coefficient

- The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample.
- The Silhouette Coefficient for a sample is (b - a) / max(a, b). To clarify, b is the distance between a sample and the nearest cluster that the sample is not a part of.
- We can compute the mean Silhouette Coefficient over all samples and use this as a metric to judge the number of clusters.

Example

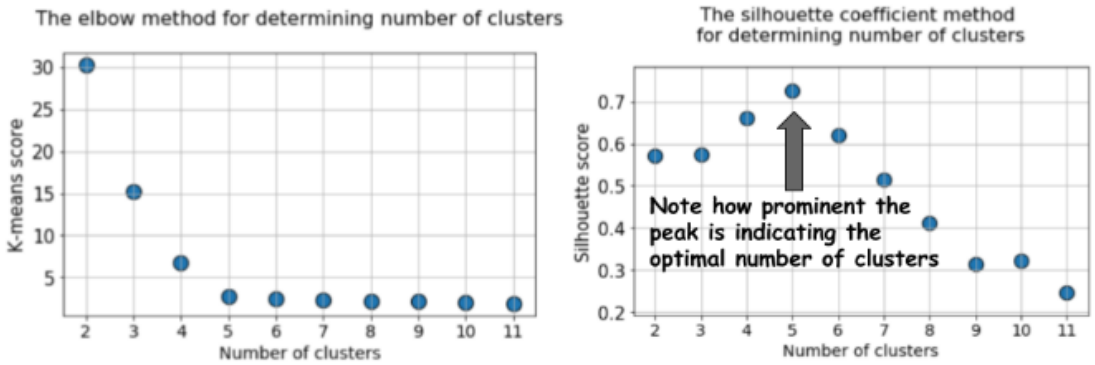
- 4 feature dimensions and five clusters generated at random!



https://towardsdatascience.com/clustering-metrics-better-than-the-elbow-method-6926e1f723a6

Example

- 4 feature dimensions and five clusters generated at random!



The difference could not be starker. The mean silhouette coefficient increases up to the point when $k=5$ and then sharply decreases for higher values of k i.e. **it exhibits a clear peak at $k=5$,** which is the number of clusters the original dataset was generated with.

https://towardsdatascience.com/clustering-metrics-better-than-the-elbow-method-6926e1f723a6