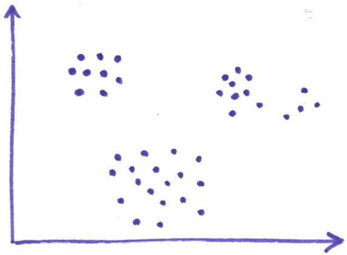
Supervised and unsupervised learning

Supervised learning: input data $\rightarrow$ output data.

Unsupervised learning: input data and nothing else.

Clustering problem



This problem is very hard to formalize. What exactly is a *cluster*? How to determine the number of clusters in the data? Are the data clustered at all? How to compare two clustering algorithms?

K-means clustering

K-means clustering aims to cluster the dataset $\{\mathbf{x}_i\}$ into K clusters $\{S_k\}$, each represented by a vector $\boldsymbol{\mu}_k$, to minimize the following loss function:

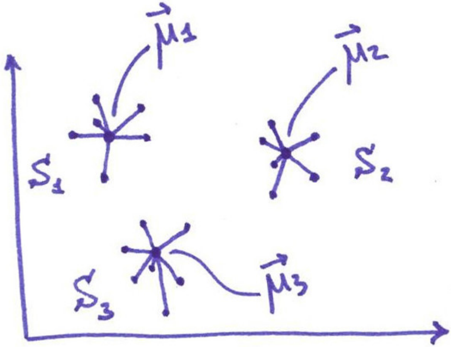
$$\mathcal{L} = \sum_{k=1}^{K} \sum_{i \in S_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2.$$

Alternatively, this can be written as

$$\mathcal{L} = \sum_{k=1}^{K} \sum_{i=1}^{n} r_{ik} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2,$$

where $r_{ik} = 1$ if $\mathbf{x}_i \in S_k$ and 0 otherwise.

K-means loss function



Minimizing the K-means loss

$$\mathcal{L} = \sum_k \sum_{i \in S_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 = \sum_{k=1}^{K} \sum_{i=1}^{n} r_{ik} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$$

Not analytically solvable. Not convex. Gradient descent can be messy.

Alternative approach (Lloyd's algorithm): iteratively optimize over r_{ik} and over $\boldsymbol{\mu}_k$.

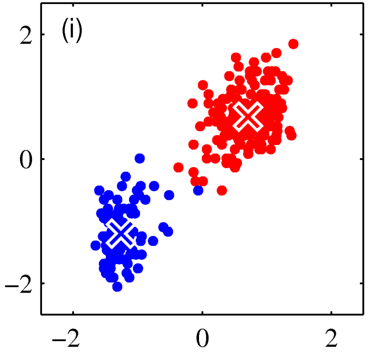
- For fixed $\boldsymbol{\mu}_k$: assign each point $\mathbf{x}_i$ to the nearest cluster center.

$$i \in S_k \text{ if } k = \arg\min_j \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2.$$

- For fixed r_{ik}:

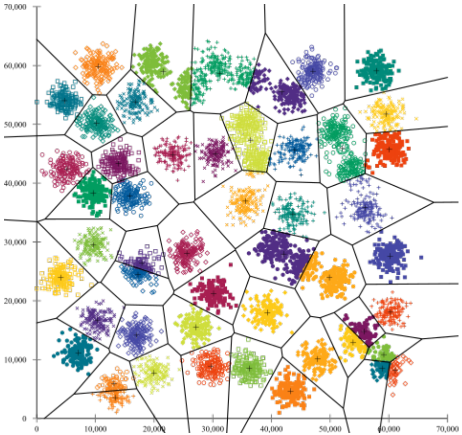
$$\boldsymbol{\mu}_k = \frac{1}{|S_k|} \sum_{i \in S_k} \mathbf{x}_i.$$

Illustration of the Lloyd's algorithm



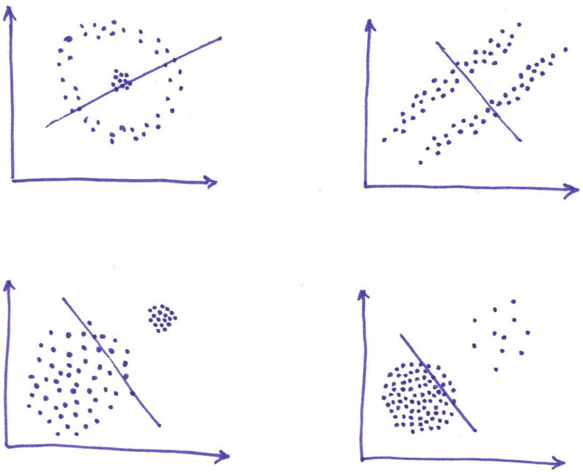
Bishop, *Pattern Recognition and Machine Learning*

Local minima



https://stats.stackexchange.com/questions/133656

Drawbacks of K-means



Gaussian mixture model (GMM)

Gaussian mixture:

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

Intuitively, we could use the same iterative approach as in the Lloyd's algorithm for K-means:

- Assign each point to the 'nearest' Gaussian component (cluster). Here 'nearest' means 'with the highest posterior' $\pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.
- Update the parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k)$ of each Gaussian.

Likelihood in GMM

Gaussian mixture:

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

Log-likelihood:

$$\mathcal{L} = \sum_{i=1}^{n} \log \left[\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right],$$

where $\mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \ldots \exp\left(-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)\right)$.

Set the derivative with respect to $\boldsymbol{\mu}_k$ to zero:

$$\sum_{i=1}^{n} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{z_{ik}} \cancel{\boldsymbol{\Sigma}_k^{-1}}(\mathbf{x}_i - \boldsymbol{\mu}_k) = 0.$$

Likelihood in GMM

Set the derivative with respect to $\boldsymbol{\mu}_k$ to zero:

$$\sum_{i=1}^{n} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{z_{ik}} \cancel{\boldsymbol{\Sigma}_k^{-1}}(\mathbf{x}_i - \boldsymbol{\mu}_k) = 0.$$

$$\sum_{i=1}^{n} z_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k) = 0.$$

$$\boldsymbol{\mu}_k = \frac{\sum z_{ik}\mathbf{x}_i}{\sum z_{ik}}.$$

This is a *weighted* mean of all points.

Very similar derivation shows that $\boldsymbol{\Sigma}_k$ should be the weighted covariance matrix, and $\pi_k = \sum z_{ik}/n$.

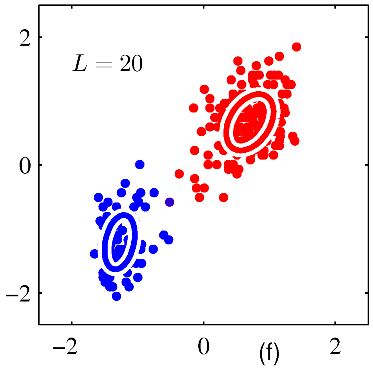
Expectation-maximization (EM)

Expectation-maximization algorithm iteratively alternates between updating $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$ and updating z_{ik}:

- **E-step:** compute the posterior probability z_{ik} for each point to be in each Gaussian component.
- **M-step:** update the parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k)$ of each Gaussian using weighted averages.

EM is a very generic algorithm to optimize likelihood in probabilistic models with *latent variables*. (In GMMs, latent variables are true class memberships.) E-step computes posterior over latent variables, conditioned on the parameters of the model. M-step optimizes the parameters, conditioned on the latent variables.

Illustration of the EM



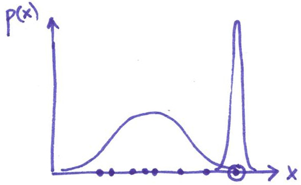
Bishop, *Pattern Recognition and Machine Learning*

Divergence in GMM

Gaussian mixture:

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

The likelihood can diverge if $\boldsymbol{\mu}_k = \mathbf{x}_i$ for some i and $\boldsymbol{\Sigma}_k \to \mathbf{0}$.



In practice: if one of the Gaussians starts 'collapsing' during EM towards a degenerate solution, do something (e.g. randomly reset its mean and covariance matrix).

EM vs. gradient descent

- Both EM and gradient descent are iterative algorithms.
- Both can converge to a local minimum.
- EM does not need a learning rate.
- In EM, all parameters are automatically meaningful after each step without imposing constraints (such as π_k summing to 1, or all $\mathbf{\Sigma}_k$ being positive-definite).

GMM vs. K-means

Similar to what we discussed about LDA, one can constrain $\mathbf{\Sigma}_k$ in a GMM to be shared between classes, or diagonal, or spherical.

A GMM with shared spherical covariance matrix $\mathbf{\Sigma}_k = \sigma^2\mathbf{I}$ is very closely related to K-means. The main difference is that K-means performs *hard* cluster assignments in the 'E-step', whereas GMM performs *soft* cluster assignments. If $\sigma^2 \to 0$, GMM converges to K-means.

Note: in practical implementations it can be convenient to initialize GMM with a K-means solution.

GMM vs. K-means