Mathematics for Machine Learning

Prof Willie Brink

Applied Mathematics, Stellenbosch University

Lecture 10: Density Estimation with GMMs

Contents of the module

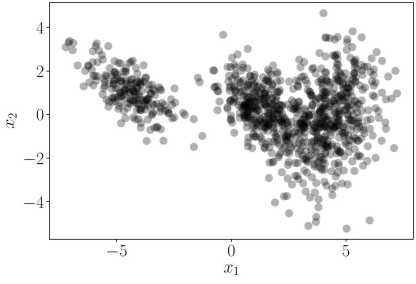
Introduction

The aim here is to represent a given dataset compactly using a probability density function from some parametric family (e.g. a Gaussian distribution).

Useful especially for large datasets.

From a density we can sample, that is, generate new data.

We can also compute the likelihood that a new point comes from the same distribution.



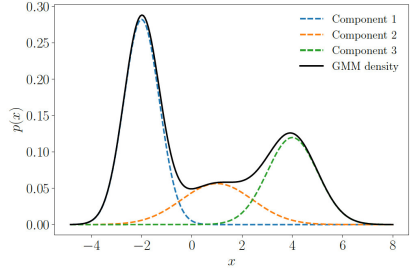
11.1 Gaussian mixture model

A Gaussian mixture model is a linear (convex) combination of K Gaussian distributions:

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad \text{with} \quad \pi_k \in [0,1] \quad \text{and} \quad \sum_{k=1}^{K} \pi_k = 1$$

where $\boldsymbol{\theta} = \{\boldsymbol{x}_k, \boldsymbol{\Sigma}_k, \pi_k : k = 1, \ldots, K\}$ contains all the parameters of the model.

This gives us significantly more flexibility than a single unimodal Gaussian distribution.



Example

$$p(x|\boldsymbol{\theta}) = 0.5\mathcal{N}(x|-2, \tfrac{1}{2}) + 0.2\mathcal{N}(x|1,2) + 0.3\mathcal{N}(x|4,1)$$

11.2 Parameter estimation via maximum likelihood

Given a dataset $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ of points i.i.d. sampled from some distribution $p(\boldsymbol{x})$, our task is to represent the unknown $p(\boldsymbol{x})$ by a GMM with K mixture components.
The idea is to find the maximum likelihood estimate $\boldsymbol{\theta}_{\text{ML}}$ of the GMM parameters.

The data is i.i.d., so $p(\mathcal{X}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{\theta}) = \prod_{n=1}^{N} \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)$

The log-likelihood is $\mathcal{L}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\boldsymbol{\theta}) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

We'd like to find the gradient of $\mathcal{L}$ w.r.t. the model parameters $\boldsymbol{\theta}$, set it to $\mathbf{0}$, and solve for $\boldsymbol{\theta}$. Unfortunately, here we cannot obtain a closed-form solution. Instead we will use an iterative scheme, where the idea is to update one parameter at a time while keeping the others fixed.

The responsibility of the kth mixture component for the n-th data point is defined as

$$r_{n,k} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

$(r_{n,1}, \ldots, r_{n,K})$ is a probability vector; a "soft assignment" of $\boldsymbol{x}_n$ to the K components.

Updating the GMM means: $\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} r_{n,k}\, \boldsymbol{x}_n \quad \text{with} \quad N_k = \sum_{n=1}^{N} r_{n,k}$

Updating the GMM covariances: $\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} r_{n,k} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}$

Updating the GMM miture weights: $\pi_k = \frac{N_k}{N}$

11.3 EM algorithm

Unfortunately, the updates on the previous slide are not a closed-form solution for the parameters of the GMM, because the responsibilities $r_{n,k}$ depend on those parameters.

But they do suggest a simple iterative scheme, called expectation maximisation.

Choose initial values for $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, $\boldsymbol{\pi}_k$, and alternate until convergence between:

E-step: evaluate the responsibilities $r_{n,k}$

M-step: use the updated responsibilities to re-estimate $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k