Foundations of Machine Learning
African Masters in Machine Intelligence

Density Estimation with Gaussian Mixture Models

Marc Deisenroth

Quantum Leap Africa
African Institute for Mathematical Sciences, Rwanda

Department of Computing
Imperial College London

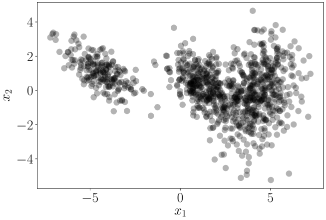
@mpd37
mdeisenroth@aimsammi.org

October 30, 2018

Reading Material

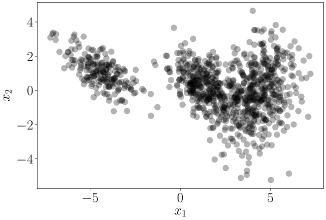
- Mathematics for Machine Learning (Chapter 11): mml-book.com
- Pattern Recognition and Machine Learning (Chapter 9)

Problem Statement



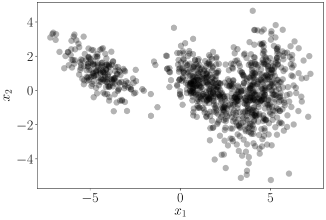
- Density estimation: Given a dataset (unlabeled), find a probability density function from which the data could have plausibly been generated

Problem Statement



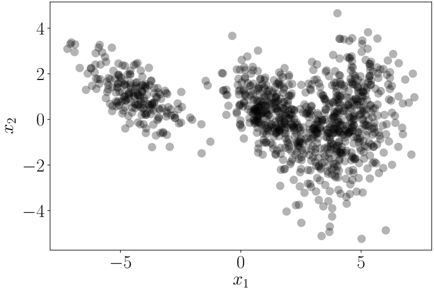
- Density estimation: Given a dataset (unlabeled), find a probability density function from which the data could have plausibly been generated
- Typically: Fix the class/model of densities and find optimal parameters given this class

Problem Statement



- Density estimation: Given a dataset (unlabeled), find a probability density function from which the data could have plausibly been generated
- Typically: Fix the class/model of densities and find optimal parameters given this class
- Example. Class: Gaussian; Find mean and variance
  ▶▶ MLE/MAP estimation

Problem Statement (2)



- Gaussians (or similarly all other distributions we encountered so far) have very limited modeling capabilities: Too simple

  ▶▶ Mixture models are more flexible

Overview

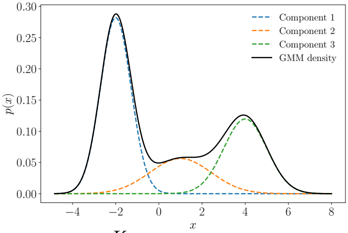
Gaussian Mixture Models

Parameter Learning

Implementation

Probabilistic Perspective

Gaussian Mixture Model



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

$$0 \leqslant \pi_k \leqslant 1$$

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

- Individual components are Gaussian distributions
- Each component is weighted by π_k (mixture weights)

Overview

Gaussian Mixture Models

Parameter Learning

Implementation

Probabilistic Perspective

Parameter Learning for GMMs

- Objective: Maximum likelihood estimate of model parameters $\boldsymbol{\theta}$ given a dataset $\mathcal{X}$
- $\boldsymbol{\theta} := \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k,\ k = 1, \ldots, K\}$
- Maximum likelihood estimate:

Parameter Learning for GMMs

- Objective: Maximum likelihood estimate of model parameters $\boldsymbol{\theta}$ given a dataset $\mathcal{X}$
- $\boldsymbol{\theta} := \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k,\ k = 1, \ldots, K\}$
- Maximum likelihood estimate:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\mathcal{X}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\mathcal{X} \,|\, \boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k\big)$$

$$\stackrel{\log}{=} \arg\max_{\boldsymbol{\theta}} \log \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\mathcal{X} \,|\, \boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k\big)$$

Parameter Learning for GMMs

- Objective: Maximum likelihood estimate of model parameters $\boldsymbol{\theta}$ given a dataset $\mathcal{X}$
- $\boldsymbol{\theta} := \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k,\ k = 1, \dots, K\}$
- Maximum likelihood estimate:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\mathcal{X}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\mathcal{X} \,|\, \boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k\big)$$

$$\stackrel{\log}{=} \arg\max_{\boldsymbol{\theta}} \log \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\mathcal{X} \,|\, \boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k\big)$$

- Problem: We cannot move the log into the sum
  ▶▶ Difficult optimization problem

Parameter Learning for GMMs

- Objective: Maximum likelihood estimate of model parameters $\boldsymbol{\theta}$ given a dataset $\mathcal{X}$
- $\boldsymbol{\theta} := \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k,\ k = 1, \dots, K\}$
- Maximum likelihood estimate:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\mathcal{X}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\mathcal{X} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)$$

$$\stackrel{\log}{=} \arg\max_{\boldsymbol{\theta}} \log \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\mathcal{X} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)$$

- Problem: We cannot move the log into the sum
  - ▶▶ Difficult optimization problem

▶▶ Iterative scheme (EM Algorithm) for learning parameters

GMM Likelihood

Assume an i.i.d. data set $\mathcal{X} = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ is given, and we want to determine the optimal parameters $\boldsymbol{\theta}^*$ of the GMM via Maximum Likelihood

1. Likelihood:

$$p(\mathcal{X}|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\theta}), \qquad p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)$$

GMM Likelihood

Assume an i.i.d. data set $\mathcal{X} = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ is given, and we want to determine the optimal parameters $\boldsymbol{\theta}^*$ of the GMM via Maximum Likelihood

1. Likelihood:

$$p(\mathcal{X}|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\theta}), \qquad p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)$$

2. Log-likelihood:

$$\log p(\mathcal{X}|\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \underbrace{\sum_{i=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)}_{=:L}$$

Necessary Optimality Conditions

Learning Objective

Find parameters $\boldsymbol{\theta}^*$ that maximize the log-likelihood

$$\log p(\mathcal{X}|\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \underbrace{\sum_{i=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)}_{=:L}$$

Necessary Optimality Conditions

Learning Objective

Find parameters $\boldsymbol{\theta}^*$ that maximize the log-likelihood

$$\log p(\mathcal{X}|\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \underbrace{\sum_{i=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)}_{=:L}$$

$$\frac{\partial L}{\partial \boldsymbol{\mu}_k} = \boldsymbol{0}^\top \iff \sum_{i=1}^{N} \frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \boldsymbol{0}^\top$$

$$\frac{\partial L}{\partial \boldsymbol{\Sigma}_k} = \boldsymbol{0} \iff \sum_{i=1}^{N} \frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}_k} = \boldsymbol{0}$$

$$\frac{\partial L}{\partial \pi_k} = 0 \iff \sum_{i=1}^{N} \frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \pi_k} = 0$$

High-Level Gradients

We need to compute gradients of the form

$$\frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

High-Level Gradients

We need to compute gradients of the form

$$\frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

In general,

$$\frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{p(\boldsymbol{x}_i|\boldsymbol{\theta})} \frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

High-Level Gradients

We need to compute gradients of the form

$$\frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

In general,

$$\frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{p(\boldsymbol{x}_i|\boldsymbol{\theta})} \frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

With

$$p(\boldsymbol{x}_i|\boldsymbol{\theta}) =$$

High-Level Gradients

We need to compute gradients of the form

$$\frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

In general,

$$\frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{p(\boldsymbol{x}_i|\boldsymbol{\theta})} \frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

With

$$p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \sum_{j=1}^{K} \pi_j \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)$$

High-Level Gradients

We need to compute gradients of the form

$$\frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

In general,

$$\frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{p(\boldsymbol{x}_i|\boldsymbol{\theta})} \frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

With

$$p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \sum_{j=1}^{K} \pi_j \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)$$

we get

$$\frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} =$$

High-Level Gradients

We need to compute gradients of the form

$$\frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

In general,

$$\frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{p(\boldsymbol{x}_i|\boldsymbol{\theta})}\frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

With

$$p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \sum_{j=1}^{K} \pi_j \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)$$

we get

$$\frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{j=1}^{K} \pi_j \frac{\partial \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)}{\partial \boldsymbol{\mu}_k}$$

In More Detail

$$\frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{j=1}^{K} \pi_j \frac{\partial \mathcal{N}\left(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}{\partial \boldsymbol{\mu}_k} =$$

In More Detail

$$\frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{j=1}^{K} \pi_j \frac{\partial \mathcal{N}\left(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}{\partial \boldsymbol{\mu}_k} = \pi_k \frac{\partial \mathcal{N}\left(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\partial \boldsymbol{\mu}_k}$$

In More Detail

$$\frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{j=1}^{K} \pi_j \frac{\partial \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)}{\partial \boldsymbol{\mu}_k} = \pi_k \frac{\partial \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)}{\partial \boldsymbol{\mu}_k}$$

$$= \pi_k (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)$$

In More Detail

$$\frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{j=1}^{K} \pi_j \frac{\partial \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)}{\partial \boldsymbol{\mu}_k} = \pi_k \frac{\partial \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)}{\partial \boldsymbol{\mu}_k}$$

$$= \pi_k (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)$$

Overall:

$$\frac{\partial L}{\partial \boldsymbol{\mu}_k} = \sum_{i=1}^{N} \frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{i=1}^{N} \frac{1}{p(\boldsymbol{x}_i|\boldsymbol{\theta})} \frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k}$$

In More Detail

$$\frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{j=1}^{K} \pi_j \frac{\partial \mathcal{N}\left(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}{\partial \boldsymbol{\mu}_k} = \pi_k \frac{\partial \mathcal{N}\left(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\partial \boldsymbol{\mu}_k}$$

$$= \pi_k (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \mathcal{N}\left(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$$

Overall:

$$\frac{\partial L}{\partial \boldsymbol{\mu}_k} = \sum_{i=1}^{N} \frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{i=1}^{N} \frac{1}{p(\boldsymbol{x}_i|\boldsymbol{\theta})} \frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k}$$

$$= \sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \underbrace{\frac{\pi_k \mathcal{N}\left(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_j \pi_j \mathcal{N}\left(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}}_{:= r_{ik}}$$

In More Detail

$$\frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{j=1}^{K} \pi_j \frac{\partial \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)}{\partial \boldsymbol{\mu}_k} = \pi_k \frac{\partial \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)}{\partial \boldsymbol{\mu}_k}$$

$$= \pi_k (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)$$

Overall:

$$\frac{\partial L}{\partial \boldsymbol{\mu}_k} = \sum_{i=1}^{N} \frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{i=1}^{N} \frac{1}{p(\boldsymbol{x}_i|\boldsymbol{\theta})} \frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k}$$

$$= \sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \underbrace{\frac{\pi_k \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)}{\sum_j \pi_j \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)}}_{:= r_{ik}} = \sum_{i=1}^{N} r_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} = \boldsymbol{0}^\top$$

In More Detail

$$\frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{j=1}^{K} \pi_j \frac{\partial \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)}{\partial \boldsymbol{\mu}_k} = \pi_k \frac{\partial \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)}{\partial \boldsymbol{\mu}_k}$$

$$= \pi_k (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)$$

Overall:

$$\frac{\partial L}{\partial \boldsymbol{\mu}_k} = \sum_{i=1}^{N} \frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{i=1}^{N} \frac{1}{p(\boldsymbol{x}_i|\boldsymbol{\theta})} \frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k}$$

$$= \sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \underbrace{\frac{\pi_k \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)}{\sum_j \pi_j \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)}}_{:= r_{ik}} = \sum_{i=1}^{N} r_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} = \boldsymbol{0}^\top$$

$$\iff \sum_{i=1}^{N} r_{ik} \boldsymbol{x}_i = \sum_{i=1}^{N} r_{ik} \boldsymbol{\mu}_k$$

In More Detail

$$\frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{j=1}^{K} \pi_j \frac{\partial \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)}{\partial \boldsymbol{\mu}_k} = \pi_k \frac{\partial \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)}{\partial \boldsymbol{\mu}_k}$$

$$= \pi_k (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)$$

Overall:

$$\frac{\partial L}{\partial \boldsymbol{\mu}_k} = \sum_{i=1}^{N} \frac{\partial \log p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{i=1}^{N} \frac{1}{p(\boldsymbol{x}_i|\boldsymbol{\theta})} \frac{\partial p(\boldsymbol{x}_i|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k}$$

$$= \sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \underbrace{\frac{\pi_k \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)}{\sum_j \pi_j \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)}}_{:= r_{ik}} = \sum_{i=1}^{N} r_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} = \boldsymbol{0}^\top$$

$$\iff \sum_{i=1}^{N} r_{ik} \boldsymbol{x}_i = \sum_{i=1}^{N} r_{ik} \boldsymbol{\mu}_k \iff \boldsymbol{\mu}_k^* = \frac{\sum_{i=1}^{N} r_{ik} \boldsymbol{x}_i}{\sum_{i=1}^{N} r_{ik}} = \frac{1}{N_k} \sum_{i=1}^{N} r_{ik} \boldsymbol{x}_i$$

Similarly...

$$\frac{\partial L}{\partial \boldsymbol{\Sigma}_k} = \mathbf{0} \iff \boldsymbol{\Sigma}_k^* = \frac{1}{N_k} \sum_{i=1}^{N} r_{ik} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top$$

$$\frac{\partial L}{\partial \pi_k} = \mathbf{0}^\top \iff \pi_k^* = \frac{N_k}{N}$$

▸▸Requires Lagrange multipliers

▸▸ See Chapter 11 of "Mathematics for Machine Learning" for details

Similarly...

$$\frac{\partial L}{\partial \boldsymbol{\Sigma}_k} = \mathbf{0} \iff \boldsymbol{\Sigma}_k^* = \frac{1}{N_k} \sum_{i=1}^{N} r_{ik} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top$$

$$\frac{\partial L}{\partial \pi_k} = \mathbf{0}^\top \iff \pi_k^* = \frac{N_k}{N}$$

▸▸Requires Lagrange multipliers

▸▸ See Chapter 11 of "Mathematics for Machine Learning" for details

- Bad news: These results do not constitute a closed-form solution of the parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$ of the mixture model because the responsibilities r_{ik} depend on those parameters in a complex way.

Similarly...

$$\frac{\partial L}{\partial \boldsymbol{\Sigma}_k} = \mathbf{0} \iff \boldsymbol{\Sigma}_k^* = \frac{1}{N_k} \sum_{i=1}^{N} r_{ik} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top$$

$$\frac{\partial L}{\partial \pi_k} = \mathbf{0}^\top \iff \pi_k^* = \frac{N_k}{N}$$

▸▸ Requires Lagrange multipliers

▸▸ See Chapter 11 of "Mathematics for Machine Learning" for details

- Bad news: These results do not constitute a closed-form solution of the parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$ of the mixture model because the responsibilities r_{ik} depend on those parameters in a complex way.
- Good news: Results suggest a simple iterative scheme for finding a solution to the MLE problem: Compute responsibilities and then update one parameter at a time while keeping the other ones fixed ▸▸ **Expectation Maximimation** algorithm

Overview

Gaussian Mixture Models

Parameter Learning

Implementation

Probabilistic Perspective

Expectation Maximization (EM) Algorithm

- Iterative scheme for learning parameters in mixture models and latent-variable models
  1. Choose initial values for $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$
  2. Until convergence, alternate between
     - **E-step:** Evaluate the responsibilities r_{ik} (posterior probability of data point i belonging to mixture component k)
     - **M-step:** Use the updated responsibilities to re-estimate the parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$
- Every step in the EM algorithm increases the likelihood function
- Convergence: Check log-likelihood or the parameters

Implementation

1. Initialize $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$

Implementation

1. Initialize $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$
2. **E-step:** Evaluate responsibilities for every data point $\boldsymbol{x}_i$ using current parameters $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$:

$$r_{ik} = \frac{\pi_k \mathcal{N}\left(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \, \boldsymbol{\Sigma}_k\right)}{\sum_j \pi_j \mathcal{N}\left(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \, \boldsymbol{\Sigma}_j\right)}$$

Implementation

1. Initialize $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$
2. **E-step:** Evaluate responsibilities for every data point $\boldsymbol{x}_i$ using current parameters $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$:

$$r_{ik} = \frac{\pi_k \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \, \boldsymbol{\Sigma}_k\big)}{\sum_j \pi_j \mathcal{N}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \, \boldsymbol{\Sigma}_j\big)}$$

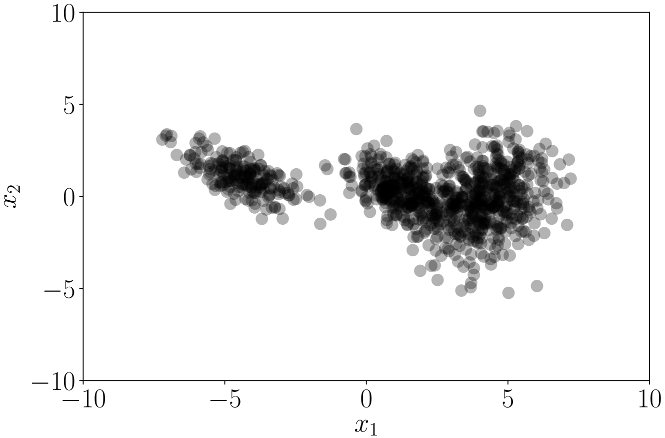
3. **M-step:** Re-estimate parameters $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ using the current responsibilities r_{ik} (from E-step):

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum\nolimits_{i=1}^{N} r_{ik} \boldsymbol{x}_i$$

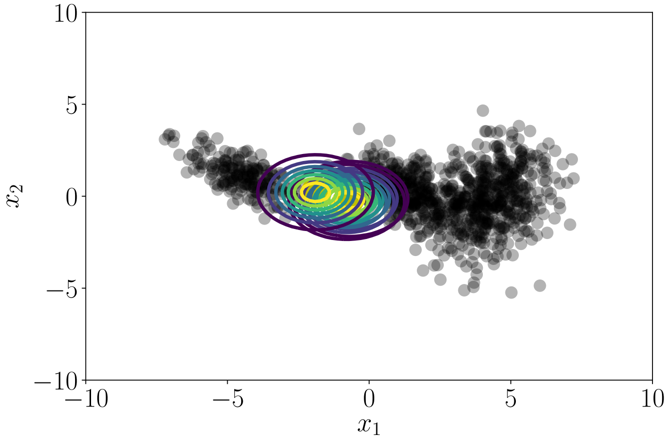
$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum\nolimits_{i=1}^{N} r_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top$$

$$\pi_k = \frac{N_k}{N}$$

Example

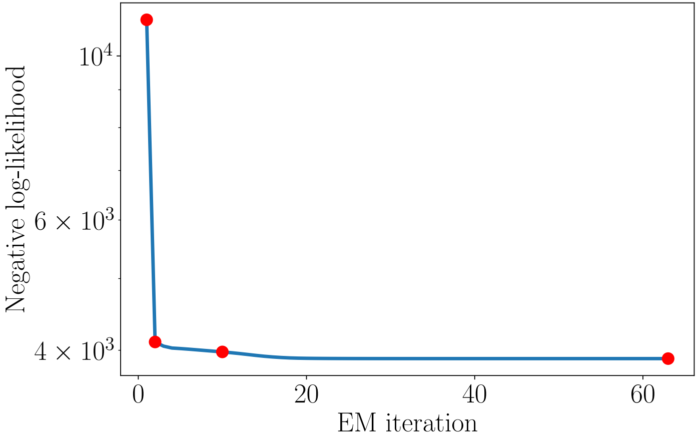


Example



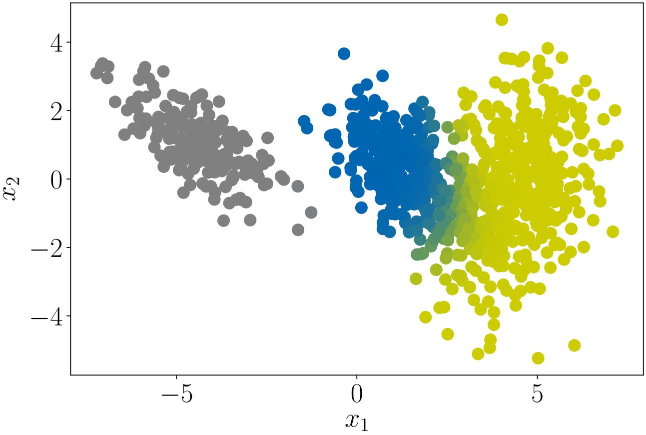
Example

Example (2)



- Negative log likelihood never increases

Visualizing the Responsibilities



- Soft assignments of data points between obvious clusters

Overview

Gaussian Mixture Models

Parameter Learning

Implementation

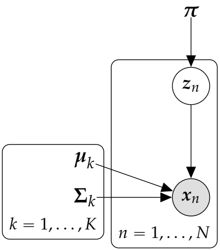
Probabilistic Perspective

Probabilistic Perspective

- We will be very explicit about how data is generated for a given set of model parameters
- Justification of some ad-hoc choices we made earlier (e.g., definition of responsibilities)
- Interpretation of some model parameters as prior/posterior probabilities
- Can be used for a principled derivation of the EM algorithm (which generally allows for maximum likelihood estimation in latent variable models)

  ▶▶ Not covered here. See Bishop (2006) for more details

Probabilistic Perspective on one Slide



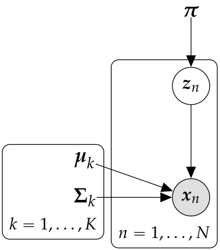
$$p(z_{nk} = 1) = \pi_k, \quad 0 \leqslant \pi_k \leqslant 1, \quad \sum_{k=1}^K \pi_k = 1$$

$$p(\boldsymbol{x}_n | z_{nk} = 1) = \mathcal{N}\left(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$$

$$p(\boldsymbol{x}_n) = \sum_{k=1}^K p(z_{nk} = 1) p(\boldsymbol{x}_n | z_{nk} = 1)$$

$$= \sum_{k=1}^K \pi_k \mathcal{N}\left(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$$

Probabilistic Perspective on one Slide



$$p(z_{nk} = 1) = \pi_k, \quad 0 \leqslant \pi_k \leqslant 1, \quad \sum_{k=1}^{K} \pi_k = 1$$

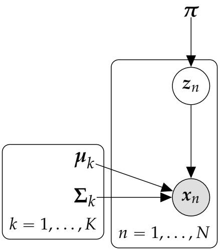
$$p(\boldsymbol{x}_n | z_{nk} = 1) = \mathcal{N}\big(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k, \, \boldsymbol{\Sigma}_k\big)$$

$$p(\boldsymbol{x}_n) = \sum_{k=1}^{K} p(z_{nk} = 1) p(\boldsymbol{x}_n | z_{nk} = 1)$$

$$= \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k, \, \boldsymbol{\Sigma}_k\big)$$

- $\boldsymbol{z}_n = (z_{n1}, \dots, z_{nK})$ is a discrete latent variable. Exactly one entry of $\boldsymbol{z}_n$ is 1, all others are 0 ▸▸ 1-of-K code/One-hot encoding
- For every data point $\boldsymbol{x}_n$ there is a corresponding latent variable $\boldsymbol{z}_n$ that indicates which mixture component generated $\boldsymbol{x}_n$

Probabilistic Perspective on one Slide



$$p(z_{nk} = 1) = \pi_k, \quad 0 \leqslant \pi_k \leqslant 1, \quad \sum_{k=1}^{K} \pi_k = 1$$

$$p(\boldsymbol{x}_n | z_{nk} = 1) = \mathcal{N}\big(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)$$

$$p(\boldsymbol{x}_n) = \sum_{k=1}^{K} p(z_{nk} = 1) p(\boldsymbol{x}_n | z_{nk} = 1)$$

$$= \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)$$

- $\boldsymbol{z}_n = (z_{n1}, \ldots, z_{nK})$ is a discrete latent variable. Exactly one entry of $\boldsymbol{z}_n$ is 1, all others are 0 ▸▸ 1-of-K code/One-hot encoding
- For every data point $\boldsymbol{x}_n$ there is a corresponding latent variable $\boldsymbol{z}_n$ that indicates which mixture component generated $\boldsymbol{x}_n$
- Posterior $p(z_k = 1 | \boldsymbol{x}_i) = r_{ik}$ corresponds to the "responsibility" (see earlier) that mixture component k generated data point i.

Prior $p(\boldsymbol{z})$

- $\pi_k = p(z_k = 1)$ is the (prior) probability that the kth mixture component generates a data point $\boldsymbol{x}$
- $\boldsymbol{\pi} := [\pi_1, \ldots, \pi_K]^\top, \;\; \sum_k \pi_k = 1$
- $p(\boldsymbol{z}) = \boldsymbol{\pi}$ is a probability vector of length K

Generative Process

Ancestral sampling from a GMM (generative process) is simple:

$$z^{(i)} \sim p(\boldsymbol{z}) \qquad \text{Select mixture component}$$

$$\boldsymbol{x}_i \sim p(\boldsymbol{x}|z^{(i)} = 1) \qquad \text{Draw sample from this component}$$

Discard sampled $z^{(i)}$ and end up with valid data samples $\boldsymbol{x}_i$ from the GMM

Likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$

- $\boldsymbol{\theta} := \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k : \; k = 1, \ldots, K\}$ contains all model parameters
- Likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ does not depend on latent variables
  ▶▶ Marginalize out latent variable $\boldsymbol{z}$:

$$
\begin{aligned}
p(\boldsymbol{x}|\boldsymbol{\theta}) &= \sum_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{z}) p(\boldsymbol{z}|\boldsymbol{\theta}) \\
&= \sum_{k=1}^{K} p(\boldsymbol{x}|\boldsymbol{\theta}, z_k = 1) p(z_k = 1|\boldsymbol{\theta}) \\
&= \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{\mu}_k, \, \boldsymbol{\Sigma}_k\big)
\end{aligned}
$$

Likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$

- $\boldsymbol{\theta} := \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k : \; k = 1, \ldots, K\}$ contains all model parameters
- Likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ does not depend on latent variables

  ▶▶ Marginalize out latent variable $\boldsymbol{z}$:

$$
\begin{aligned}
p(\boldsymbol{x}|\boldsymbol{\theta}) &= \sum_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{z}) p(\boldsymbol{z}|\boldsymbol{\theta}) \\
&= \sum_{k=1}^{K} p(\boldsymbol{x}|\boldsymbol{\theta}, z_k = 1) p(z_k = 1|\boldsymbol{\theta}) \\
&= \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{\mu}_k, \, \boldsymbol{\Sigma}_k\big)
\end{aligned}
$$

▶▶ Classical GMM likelihood function

Posterior $p(\boldsymbol{z}|\boldsymbol{x})$

- Bayes' theorem:

$$p(\boldsymbol{z} \mid \boldsymbol{x}) = \frac{p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})}{p(\boldsymbol{x})}$$

Posterior $p(\boldsymbol{z}|\boldsymbol{x})$

- Bayes' theorem:

$$p(\boldsymbol{z} \mid \boldsymbol{x}) = \frac{p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})}{p(\boldsymbol{x})}$$

- Posterior distribution

Posterior $p(\boldsymbol{z}|\boldsymbol{x})$

- Bayes' theorem:

$$p(\boldsymbol{z} \mid \boldsymbol{x}) = \frac{p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})}{p(\boldsymbol{x})}$$

- Posterior distribution

Posterior $p(\boldsymbol{z}|\boldsymbol{x})$

- Bayes' theorem:

$$p(\boldsymbol{z}\,|\,\boldsymbol{x}) = \frac{p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})}{p(\boldsymbol{x})}$$

- Posterior distribution

$$p(z_k = 1\,|\,\boldsymbol{x}) = \frac{p(\boldsymbol{x}\,|\,z_k = 1)p(z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\boldsymbol{x}\,|\,z_j = 1)} = \frac{\pi_k \mathcal{N}\big(\boldsymbol{x}\,|\,\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\big(\boldsymbol{x}\,|\,\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)}$$

Posterior $p(\boldsymbol{z}|\boldsymbol{x})$

- Bayes' theorem:

$$p(\boldsymbol{z} \,|\, \boldsymbol{x}) = \frac{p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})}{p(\boldsymbol{x})}$$

- Posterior distribution

$$p(z_k = 1 \,|\, \boldsymbol{x}) = \frac{p(\boldsymbol{x} \,|\, z_k = 1)p(z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\boldsymbol{x} \,|\, z_j = 1)} = \frac{\pi_k \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)}$$

▸▸ "Responsibility" of the kth mixture component for data point $\boldsymbol{x}$
▸▸ Posterior probability that the kth mixture component generated data point $\boldsymbol{x}$

Some Comments

- If we make "hard assignments" of data points to clusters (responsibilities r_{ik} or posteriors are either 0 or 1), we get *K*-means clustering

Some Comments

- If we make "hard assignments" of data points to clusters (responsibilities r_{ik} or posteriors are either 0 or 1), we get K-means clustering
- Overfitting as with other maximum likelihood approaches (what does overfitting in GMMs look like?)

Some Comments

- If we make "hard assignments" of data points to clusters (responsibilities r_{ik} or posteriors are either 0 or 1), we get K-means clustering
- Overfitting as with other maximum likelihood approaches (what does overfitting in GMMs look like?)
- Bayesian extensions exist with priors on the parameters

Some Comments

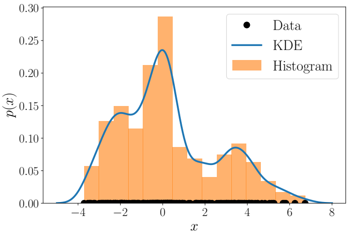
- If we make "hard assignments" of data points to clusters (responsibilities r_{ik} or posteriors are either 0 or 1), we get K-means clustering
- Overfitting as with other maximum likelihood approaches (what does overfitting in GMMs look like?)
- Bayesian extensions exist with priors on the parameters
- Choosing the number of components can be done via model selection or a Bayesian prior (Dirichlet process, Görür (2007))

Some Comments

- If we make "hard assignments" of data points to clusters (responsibilities r_{ik} or posteriors are either 0 or 1), we get K-means clustering
- Overfitting as with other maximum likelihood approaches (what does overfitting in GMMs look like?)
- Bayesian extensions exist with priors on the parameters
- Choosing the number of components can be done via model selection or a Bayesian prior (Dirichlet process, Görür (2007))
- EM can generally be used for parameter learning in latent variable models (e.g., Ghahramani & Roweis (1999)) or reinforcement learning (e.g., Barber (2012))

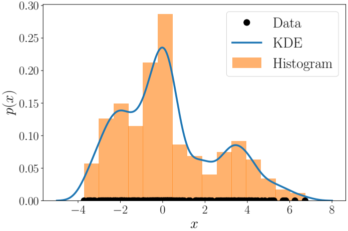
  ▶▶ Latent variable perspective is useful to derive EM in a principled way

Other Density Estimation Methods



- Histograms (Pearson1895)

Other Density Estimation Methods

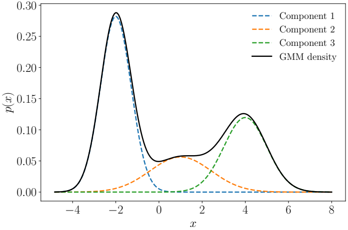


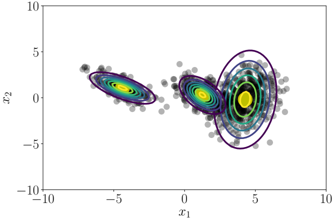
- Histograms (Pearson1895)
- Kernel density estimation (Rosenblatt 1956)

$$p(\boldsymbol{x}) = \frac{1}{Nh} \sum_{n=1}^{N} k\left(\frac{\boldsymbol{x} - \boldsymbol{x}_n}{h}\right), \quad h > 0\,,$$

$$k(\cdot) \geqslant 0, \quad \int k(\boldsymbol{x}) d\boldsymbol{x} = 1$$

Summary





- Density estimation with Gaussian mixture models
- No closed-form solution to maximum likelihood estimation
- EM algorithm for an iterative solution
- Latent variable perspective

References I

[1] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer-Verlag, 2006.

[3] Z. Ghahramani and S. T. Roweis. Learning Nonlinear Dynamical Systems using an EM Algorithm. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11. The MIT Press, 1999.

[4] D. Görür. *Nonparametric Bayesian Discrete Latent Variable Models for Unsupervised Learning*. PhD thesis, Technische Universität Berlin, 2007.

[5] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA, USA, 2012.

[6] K. Pearson. Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 186:343–414, 1895.

[7] M. Rosenblatt. Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.