Imperial College London

Lecture 13: Gaussian Mixture Models, EM, Model Selection

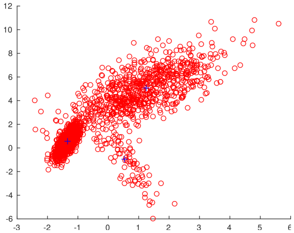
Recommended reading: Bishop, Chapter 1.3, 3.1, 9.2

Duncan Gillies and Marc Deisenroth

Department of Computing
Imperial College London

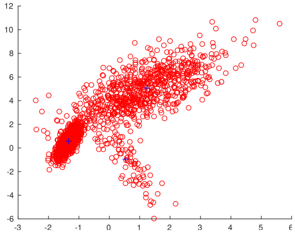
February 17, 2016

Problem Statement



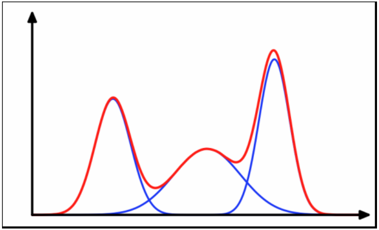
- Often, we are given a set of points whose density we wish to model
- Example: Find mean, variance of a Gaussian
  - ⏩ MLE/MAP estimation

Problem Statement



- Often, we are given a set of points whose density we wish to model
- Example: Find mean, variance of a Gaussian
  - ▸▸ MLE/MAP estimation
- Gaussians (or similarly all other distributions we encountered so far) have very limited modeling capabilities.
  - ▸▸ Mixture models are more flexible

Gaussian Mixtures



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

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

$$\sum_k \pi_k = 1$$

- Individual components are Gaussian distributions
- Each component is weighted by π_k

Parameter Learning for GMMs

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

Parameter Learning for GMMs

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

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

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

Parameter Learning for GMMs

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

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

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

- Problem: We cannot move the log into the sum

  ⏩ Nasty optimization problem

Parameter Learning for GMMs

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

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

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

- Problem: We cannot move the log into the sum
  ▸▸ Nasty optimization problem

▸▸ Iterative scheme (EM Algorithm) for learning parameters

GMM Likelihood

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

1. Likelihood:

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

GMM Likelihood

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

1. Likelihood:

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

2. Log-likelihood:

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

Necessary Optimality Conditions

Learning Objective

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

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

Necessary Optimality Conditions

Learning Objective

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

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

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

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

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

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

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

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

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

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

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:

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

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

In More Detail

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

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

Overall:

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

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

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:

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

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

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

In More Detail

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

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

Overall:

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

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

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

Similarly...

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

$$\frac{\partial L}{\partial \pi_k} = \mathbf{0} \Leftrightarrow \pi_k = \frac{N_k}{N}$$

▶▶Requires Lagrange multipliers

Similarly...

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

$$\frac{\partial L}{\partial \pi_k} = \mathbf{0} \Leftrightarrow \pi_k = \frac{N_k}{N}$$

▶▶Requires Lagrange multipliers

- 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.

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 $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$:

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

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 $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$:

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

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

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

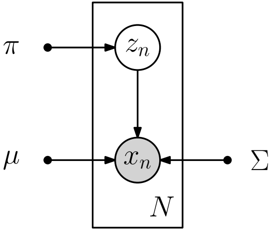
$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^{N} r_{ik} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top$$

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

Example

Demo

The Latent-Variable Perspective



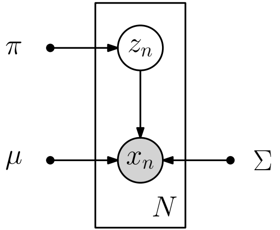
$$p(z_k = 1) = \pi_k, \quad 0 \leqslant \pi_k \leqslant 1, \sum_k \pi_k = 1$$

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

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

- $\boldsymbol{z}_n = (z_1, \ldots, z_K)$ is a discrete latent variable. Exactly one entry of $\boldsymbol{z}_n$ is 1, all others are 0 ⏩ 1-of-K code

The Latent-Variable Perspective



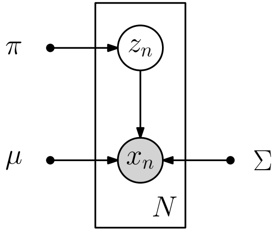
$$p(z_k = 1) = \pi_k, \quad 0 \leqslant \pi_k \leqslant 1, \sum_k \pi_k = 1$$

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

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

- $\boldsymbol{z}_n = (z_1, \dots, z_K)$ is a discrete latent variable. Exactly one entry of $\boldsymbol{z}_n$ is 1, all others are 0 ⏩ 1-of-K code
- For every observed data point $\boldsymbol{x}_n$ there is a corresponding latent variable $\boldsymbol{z}_n$, which indicates which mixture component generated $\boldsymbol{x}_n$

The Latent-Variable Perspective



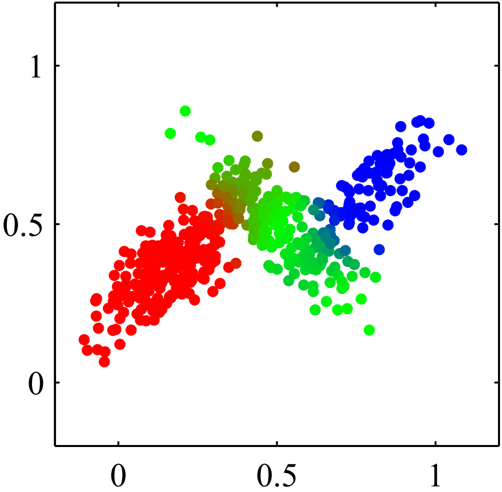
$$p(z_k = 1) = \pi_k, \quad 0 \leqslant \pi_k \leqslant 1, \sum_k \pi_k = 1$$

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

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

- $\boldsymbol{z}_n = (z_1, \ldots, z_K)$ is a discrete latent variable. Exactly one entry of $\boldsymbol{z}_n$ is 1, all others are 0 ⏩ 1-of-K code
- For every observed data point $\boldsymbol{x}_n$ there is a corresponding latent variable $\boldsymbol{z}_n$, which 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.

Visualizing the Responsibilities



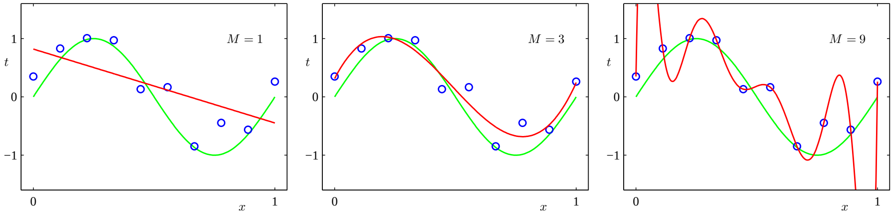
From PRML (Bishop, 2006)

EM with Latent Variables (see CO-495)

- Latent-variable perspective gives rise to a general EM algorithm for maximum likelihood parameter estimation (regression, classification, dimensionality reduction, density estimation, ...), see Dempster et al., (1977)
- EM iteratively maximizes a lower bound on the log likelihood $\log p(\boldsymbol{X}|\boldsymbol{\theta})$
- At the same time, EM iteratively minimizes the KL divergence $\mathrm{KL}(q(\boldsymbol{Z})||p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta}))$ between an approximate posterior $q(\boldsymbol{Z})$ and the true (but unknown) posterior distribution $p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})$

Model Selection

Model Selection

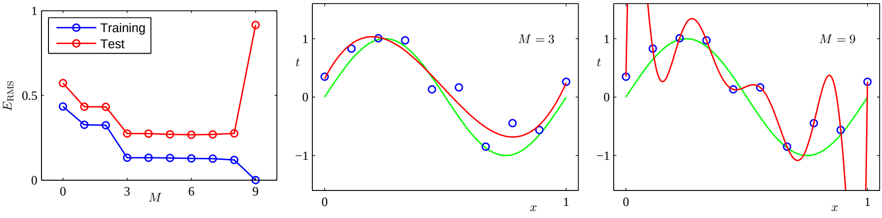


From PRML (Bishop, 2006)

Sometimes, we have to make high-level decisions about the model we want to use:

- Number of components in a mixture model
- Network architecture of (deep) neural networks
- Type of kernel in a support vector machine
- Degree of a polynomial in a regression problem

Test vs Training Error

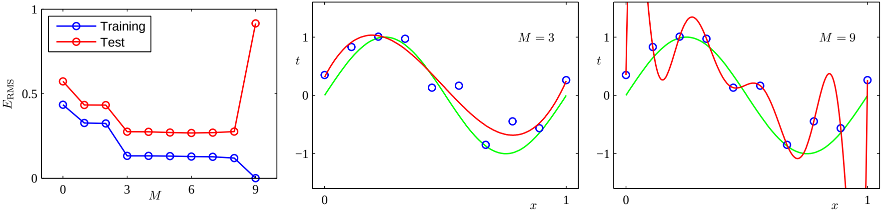


From PRML (Bishop, 2006)

General problem:

- Model fits training data perfectly, but may not do well on test data ⏩ Overfitting (especially with MLE)

Test vs Training Error

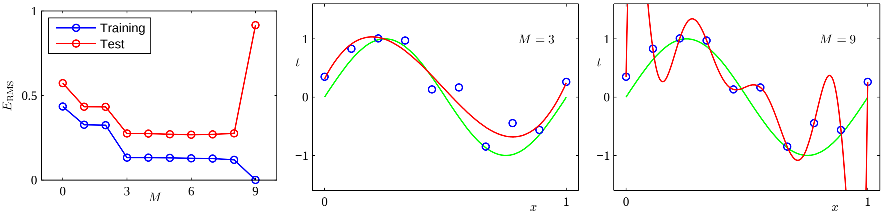


From PRML (Bishop, 2006)

General problem:

- Model fits training data perfectly, but may not do well on test data ⏩ Overfitting (especially with MLE)
- Training performance $\neq$ test performance, but we are largely interested in test performance

Test vs Training Error

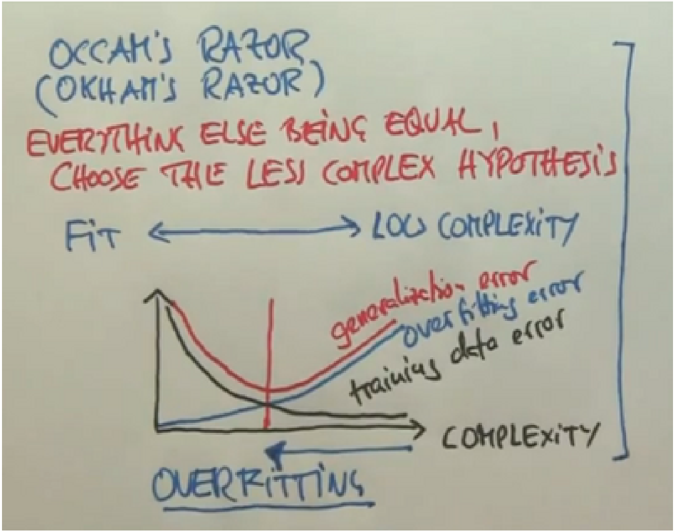


From PRML (Bishop, 2006)

General problem:

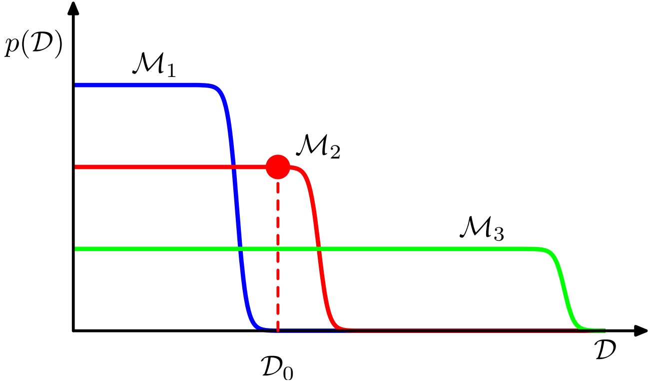
- Model fits training data perfectly, but may not do well on test data ⏩ Overfitting (especially with MLE)
- Training performance $\neq$ test performance, but we are largely interested in test performance
- Need mechanisms for assessing how a model generalizes to unseen test data ⏩ Model selection

Occam's Razor



From `crowfly.net`

Occam's Razor (2)



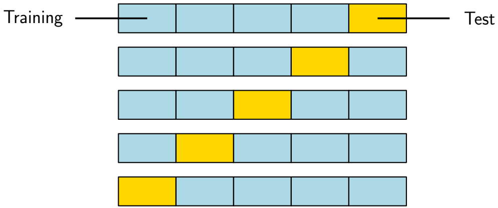
From PRML (Bishop, 2006)

- Choose the simplest model that explains the data reasonably well

Cross Validation

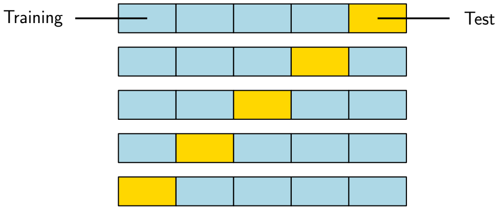
- Partition your training data into L subsets
- Train the model on $L - 1$ subsets
- Evaluate the model on the other subset

Cross Validation



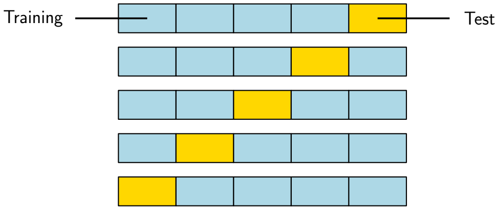
- Partition your training data into L subsets
- Train the model on $L - 1$ subsets
- Evaluate the model on the other subset
- To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.

Cross Validation



- Partition your training data into L subsets
- Train the model on $L - 1$ subsets
- Evaluate the model on the other subset
- To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.
- Train many models, compare test error

Cross Validation



- Partition your training data into L subsets
- Train the model on $L - 1$ subsets
- Evaluate the model on the other subset
- To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.
- Train many models, compare test error

Number of training runs increases with the number of partitions

Information Criteria

- Correct for the bias of MLE by addition of a penalty term to compensate for the overfitting of more complex models (with lots of parameters)

Information Criteria

- Correct for the bias of MLE by addition of a penalty term to compensate for the overfitting of more complex models (with lots of parameters)
- Akaike Information Criterion (Akaike 1974):

$$\ln p(\boldsymbol{x}|\boldsymbol{\theta}_{\mathrm{ML}}) - M$$

where M is the number of model parameters

Information Criteria

- Correct for the bias of MLE by addition of a penalty term to compensate for the overfitting of more complex models (with lots of parameters)
- Akaike Information Criterion (Akaike 1974):

$$\ln p(\boldsymbol{x}|\boldsymbol{\theta}_{\mathrm{ML}}) - M$$

  where M is the number of model parameters
- Bayesian Information Criterion/MDL (Schwarz 1978) (for exponential family distributions):

$$\ln p(\boldsymbol{x}) = \ln \int p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \approx \ln p(\boldsymbol{x}|\boldsymbol{\theta}_{\mathrm{ML}}) - \frac{1}{2} M \ln N$$

  where N is the number of data points and M is the number of parameters.

Information Criteria

- Correct for the bias of MLE by addition of a penalty term to compensate for the overfitting of more complex models (with lots of parameters)
- Akaike Information Criterion (Akaike 1974):

$$\ln p(\boldsymbol{x}|\boldsymbol{\theta}_{\mathrm{ML}}) - M$$

  where M is the number of model parameters
- Bayesian Information Criterion/MDL (Schwarz 1978) (for exponential family distributions):

$$\ln p(\boldsymbol{x}) = \ln \int p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \approx \ln p(\boldsymbol{x}|\boldsymbol{\theta}_{\mathrm{ML}}) - \frac{1}{2}M \ln N$$

  where N is the number of data points and M is the number of parameters.
- BIC penalizes model complexity more heavily than AIC.

Bayesian Model Comparison

- Place a prior $p(M)$ on the class of models

[1]When would the integral be tractable?

Bayesian Model Comparison

- Place a prior $p(M)$ on the class of models
- Given a training set $\mathcal{D}$, we compute the posterior distribution over models as

$$p(M_i|\mathcal{D}) \;\propto\; p(M_i)p(\mathcal{D}|M_i)$$

which allows us to express a preference for different models

[1] When would the integral be tractable?

Bayesian Model Comparison

- Place a prior $p(M)$ on the class of models
- Given a training set $\mathcal{D}$, we compute the posterior distribution over models as

$$p(M_i|\mathcal{D}) \propto p(M_i)p(\mathcal{D}|M_i)$$

  which allows us to express a preference for different models
- Model evidence (marginal likelihood):

$$p(\mathcal{D}|M_i) = \int p(\mathcal{D}|\boldsymbol{\theta}_{M_i})p(\boldsymbol{\theta}_{M_i}|M_i)d\boldsymbol{\theta}_{M_i}$$

[1]When would the integral be tractable?

Bayesian Model Comparison

- Place a prior $p(M)$ on the class of models
- Given a training set $\mathcal{D}$, we compute the posterior distribution over models as

$$p(M_i|\mathcal{D}) \;\propto\; p(M_i)p(\mathcal{D}|M_i)$$

which allows us to express a preference for different models

- Model evidence (marginal likelihood):

$$p(\mathcal{D}|M_i) = \int p(\mathcal{D}|\boldsymbol{\theta}_{M_i})p(\boldsymbol{\theta}_{M_i}|M_i)d\boldsymbol{\theta}_{M_i}$$

- Bayes factor for comparing two models: $p(\mathcal{D}|M_1)/p(\mathcal{D}|M_2)$

[1] When would the integral be tractable?

Bayesian Model Comparison

- Place a prior $p(M)$ on the class of models
- Given a training set $\mathcal{D}$, we compute the posterior distribution over models as

$$p(M_i|\mathcal{D}) \;\propto\; p(M_i)p(\mathcal{D}|M_i)$$

which allows us to express a preference for different models

- Model evidence (marginal likelihood):

$$p(\mathcal{D}|M_i) = \int p(\mathcal{D}|\boldsymbol{\theta}_{M_i})p(\boldsymbol{\theta}_{M_i}|M_i)d\boldsymbol{\theta}_{M_i}$$

- Bayes factor for comparing two models: $p(\mathcal{D}|M_1)/p(\mathcal{D}|M_2)$
- Integral often intractable[1]

[1] When would the integral be tractable?

Bayesian Model Averaging

- Place a prior $p(M)$ on the class of models
- Instead of selecting the "best" model, integrate out the corresponding model parameters $\boldsymbol{\theta}_M$ and average over all models $M_i, i = 1, \ldots, L$

$$p(\mathcal{D}) = \sum_{i=1}^{L} p(M_i) \underbrace{\int p(\mathcal{D}|\boldsymbol{\theta}_{M_i}) p(\boldsymbol{\theta}_{M_i}|M_i) d\boldsymbol{\theta}_{M_i}}_{=p(\mathcal{D}|M_i)}$$

Bayesian Model Averaging

- Place a prior $p(M)$ on the class of models
- Instead of selecting the "best" model, integrate out the corresponding model parameters $\boldsymbol{\theta}_M$ and average over all models $M_i, i = 1, \ldots, L$

$$p(\mathcal{D}) = \sum_{i=1}^{L} p(M_i) \underbrace{\int p(\mathcal{D}|\boldsymbol{\theta}_{M_i}) p(\boldsymbol{\theta}_{M_i}|M_i) d\boldsymbol{\theta}_{M_i}}_{=p(\mathcal{D}|M_i)}$$

- Computationally expensive
- Integral often intractable (still...)

References I

[1] H. Akaike. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

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

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[4] R. M. Neal and G. E. Hinton. A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. MIT Press, 1999.

[5] G. E. Schwarz. Estimating the Dimension of a Model. *Annals of Statistics*, 6(2):461–464, 1978.