Gaussian mixtures

Nathanaël Carraz Rakotonirina

Mathématiques Informatique et Statistique Appliquées (MISA)
Université d'Antananarivo

Mixture models

It is an unsupervised learning approach. A mixture model is a combination of multiple simple distributions.

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{i=1}^{K} \pi_i p_i(\boldsymbol{x})$$

where p_i is the ith component of the mixture and π_i the mixture weights ($0 \leq \pi_i \leq 1$ and $\sum_{i=1}^{K} \pi_i = 1$)
We introduce the latent (hidden) variable z which can take values $1, ..., K$. It specifies which distribution to use to generate the output. We note $p(z = i) = \pi_i$ and $p(\boldsymbol{x}|z = i) = p_i(\boldsymbol{x}) = p(\boldsymbol{x};\boldsymbol{\theta}_i)$.

$$p(z;\boldsymbol{\theta}) = \text{Cat}(z;\boldsymbol{\pi})$$
$$p(\boldsymbol{x}|z = i;\boldsymbol{\theta}) = p(\boldsymbol{x};\boldsymbol{\theta}_i)$$

According to the model, the data is generated as follows : we first sample a specific component z, and then depending on z, we generate the data $\boldsymbol{x}$ (using the corresponding parameter).
We obtain the form of the model by marginalizing out z:

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{i=1}^{K} p(z=i;\boldsymbol{\theta}) p(\boldsymbol{x}|z=i;\boldsymbol{\theta}) = \sum_{i=1}^{K} \pi_i p(\boldsymbol{x};\boldsymbol{\theta}_i)$$

This is difficult because we do not know the value of z.

Gaussian mixture models

A Gaussian mixture model (GMM) is a mixture model with a gaussian base distribution.

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{i=1}^{K} \pi_i \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$$

For large enough K a GMM can approximate any smooth distribution over R^D.

Applications of GMM

- Clustering
- Prior to regularize an inverse problem (image restoration tasks like image denoising, deblurring, inpainting, super-resolution ...)

GMM for clustering

Suppose we know the MLE of the parameters $\hat{\boldsymbol{\theta}} = (\boldsymbol{\pi}, \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\})$, we use Bayes rule to compute the **responsibility** r_{ji} of cluster k for data point $\boldsymbol{x}_j$:

$$r_{ji} = p(z_j = i|\boldsymbol{x}_j; \boldsymbol{\theta}) = \frac{p(z_j = i; \boldsymbol{\theta}) p(\boldsymbol{x}_j|z_j = i; \boldsymbol{\theta})}{\sum_{i'=1}^{K} p(z_j = i'; \boldsymbol{\theta}) p(\boldsymbol{x}_j|z_j = i'; \boldsymbol{\theta})} = \frac{\pi_i p(\boldsymbol{x}_j; \boldsymbol{\theta}_i)}{\sum_{i'=1}^{K} \pi_{i'} p(\boldsymbol{x}_j; \boldsymbol{\theta}_{i'})}$$

Then, we can compute the most probable cluster for x_j as

$$\hat{z}_j = \arg\max_i r_{ji} = \arg\max_i \left(\log p(z_j = i; \boldsymbol{\pi}) + \log p(\boldsymbol{x}_j|z_j = i; \boldsymbol{\theta}) \right)$$

K-means clustering

K-means clustering is a special case where $\boldsymbol{\Sigma}_i = \boldsymbol{I}$ and $\pi_i = 1/K$ so we just have to estimate the means $\boldsymbol{\mu}_i$.

EM algorithm

The **expectation maximization (EM)** algorithm is designed to compute the MLE or MAP parameter estimate for probability models that have missing data and/or hidden variables.
We alternate between two steps:

- **E step**: expectation step, we estimate the hidden variables
- **M step**: maximization step, we compute the MLE

We need to iterate this process since the steps depend on each other. It may converge to a local maximum.

MLE using the EM algorithm

- **E step**: It just computes the responsibility of cluster i for generating x_i using the current parameter estimates:

$$r_{ji}^{(t)} = p(z_j = i|\boldsymbol{x}_j; \boldsymbol{\theta}^{(t)}) = \frac{\pi_i p(\boldsymbol{x}_j; \boldsymbol{\theta}_i^{(t)})}{\sum_{i'=1}^{K} \pi_{i'} p(\boldsymbol{x}_j; \boldsymbol{\theta}_{i'}^{(t)})}$$

- **M step**: The M step maximizes the expected complete data log likelihood :

$$\begin{aligned}
\mathcal{L}^{(t)}(\boldsymbol{\theta}) &= \mathbb{E}\left[\sum_j \log p(z_j; \boldsymbol{\pi}) + \log p(\boldsymbol{x}_j|z_j; \boldsymbol{\theta})\right] \\
&= \mathbb{E}\left[\sum_j \log\left(\prod_i \pi_i^{z_{ij}}\right) + \log\left(\prod_i \mathcal{N}(\boldsymbol{x}_j; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)^{z_{ij}}\right)\right] \\
&= \sum_j \sum_i \mathbb{E}[z_{ij}] \log \pi_i + \sum_j \sum_i \mathbb{E}[z_{ij}] \mathcal{N}(\boldsymbol{x}_j; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \\
&= \sum_j \sum_i r_{ji}^{(t)} \log \pi_i + \sum_j \sum_i r_{ji}^{(t)} \mathcal{N}(\boldsymbol{x}_j; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)
\end{aligned}$$

The parameter estimates are the following:

$$\boldsymbol{\mu}_i^{(t+1)} = \frac{\sum_j r_{ji}^{(t)} \boldsymbol{x}_j}{r_k^{(t)}}$$

$$\boldsymbol{\Sigma}_i^{(t+1)} = \frac{\sum_j r_{ji}^{(t)} (\boldsymbol{x}_j - \boldsymbol{\mu}_i^{(t+1)})(\boldsymbol{x}_j - \boldsymbol{\mu}_i^{(t+1)})^\top}{r_k^{(t)}}$$

where $r_i^{(t)} = \sum_j r_{ji}^{(t)}$ weighted number of points assigned to cluster k.
The mixture weights during the M step:

$$\pi_i^{(t+1)} = \frac{1}{N} \sum_j r_{ji}^{(t)} = \frac{r_i^{(t)}}{N}$$

What's next ?

Explore

- Unidentifiability and label switching
- Bayesian model selection to select K
- MAP estimate of a GMM
- Mixtures of Bernoullis
- Gaussian scale mixtures