Gaussian mixtures

Pierre-Alexandre Mattei

http://pamattei.github.io – @pamattei
pierre-alexandre.mattei@inria.fr

MSc Data Science

Recap on Linear Discriminant Analysis

GMM basics

Estimation for unsupervised GMMs: the EM algorithm

Recap on Linear Discriminant Analysis

GMM basics

Estimation for unsupervised GMMs: the EM algorithm

Linear Discriminant Analysis (LDA)

- LDA is a simple generative classifier

[1] $\mathrm{Cat}(\pi)$ means a categorical distribution with proportions $\pi = (\pi_1, ..., \pi_K)$

Linear Discriminant Analysis (LDA)

- LDA is a simple generative classifier
- Every class is assumed to follow a multivariate Gaussian distribution.
- For K classes, the generative model is as follows:

$$c \sim \mathrm{Cat}(\pi)^1,$$

$$x|k \sim \mathcal{N}(\mu_k, \Sigma).$$

Question: Enumerate all the parameters of the LDA model.

[1] Cat(π) means a categorical distribution with proportions $\pi = (\pi_1, ..., \pi_K)$

Linear Discriminant Analysis (LDA)

- LDA is a simple generative classifier
- Every class is assumed to follow a multivariate Gaussian distribution.
- For K classes, the generative model is as follows:

$$c \sim \mathrm{Cat}(\pi)^1,$$

$$x|k \sim \mathcal{N}(\mu_k, \Sigma).$$

Question: Enumerate all the parameters of the LDA model.
Answer: $\mu_1, ..., \mu_K, \Sigma_1, ..., \Sigma_K$.

[1] Cat(π) means a categorical distribution with proportions $\pi = (\pi_1, ..., \pi_K)$

Linear Discriminant Analysis (LDA)

- LDA is a simple generative classifier
- Every class is assumed to follow a multivariate Gaussian distribution.
- For K classes, the generative model is as follows:

$$c \sim \mathrm{Cat}(\pi)^2,$$

$$x|k \sim \mathcal{N}(\mu_k, \Sigma).$$

Question: How do we estimate these parameters?

[2] Cat(π) means a categorical distribution with proportions $\pi = (\pi_1, ..., \pi_K)$

Linear Discriminant Analysis (LDA)

- LDA is a simple generative classifier
- Every class is assumed to follow a multivariate Gaussian distribution.
- For K classes, the generative model is as follows:

$$c \sim \mathrm{Cat}(\pi)^2,$$

$$x|k \sim \mathcal{N}(\mu_k, \Sigma).$$

Question: How do we estimate these parameters?
Answer: Using maximum likelihood estimates that we saw in Lecture 2: for each class k, $\hat{\mu}_k$ is the empirical mean of class k, and $\hat{\Sigma}$ is the average of the empirical covariances of all classes, weighted by their proportions.

[2] Cat(π) means a categorical distribution with proportions $\pi = (\pi_1, ..., \pi_K)$

Linear Discriminant Analysis (LDA)

- LDA is a simple generative classifier
- Every class is assumed to follow a multivariate Gaussian distribution.
- For K classes, the generative model is as follows:

$$c \sim \text{Cat}(\pi)^{3},$$

$$x|k \sim \mathcal{N}(\mu_k, \Sigma).$$

Question: How can we use this model for classification?

[3] Cat(π) means a categorical distribution with proportions $\pi = (\pi_1, ..., \pi_K)$

Linear Discriminant Analysis (LDA)

- LDA is a simple generative classifier
- Every class is assumed to follow a multivariate Gaussian distribution.
- For K classes, the generative model is as follows:

$$c \sim \text{Cat}(\pi)^{3},$$

$$x|k \sim \mathcal{N}(\mu_k, \Sigma).$$

Question: How can we use this model for classification?
Answer: Using Bayes's rule, we can compute the probabilities that a sample x belongs to each class k:

$$p(k|x) = \frac{p(x|k)p(k)}{p(x)}.$$

[3] Cat(π) means a categorical distribution with proportions $\pi = (\pi_1, ..., \pi_K)$

Linear Discriminant Analysis (LDA)

Using Bayes's rule, we can compute the probabilities that a sample x belongs to each class k:

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

can be simplified

$$p(k|x) = \frac{\mathcal{N}(x|\mu_k, \Sigma)\pi_k}{p(x)},$$

Linear Discriminant Analysis (LDA)

Using Bayes's rule, we can compute the probabilities that a sample x belongs to each class k:

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

can be simplified

$$p(k|x) = \frac{\mathcal{N}(x|\mu_k, \Sigma)\pi_k}{p(x)},$$

Question: How do we compute $p(x)$?

Linear Discriminant Analysis (LDA)

Using Bayes's rule, we can compute the probabilities that a sample x belongs to each class k:

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

can be simplified

$$p(k|x) = \frac{\mathcal{N}(x|\mu_k, \Sigma)\pi_k}{p(x)},$$

Question: How do we compute $p(x)$?
Using the law of total probability:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma).$$

The marginal distribution of LDA

Using the law of total probability:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k).$$

What is the name of such a density?

The marginal distribution of LDA

Using the law of total probability:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k).$$

What is the name of such a density?
This a called a Gaussian mixture model (GMM) with K components and a common covariance.

The marginal distribution of LDA

Using the law of total probability:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k).$$

What is the name of such a density?
This a called a Gaussian mixture model (GMM) with K components and a common covariance.

LDA is a supervised version of GMM. As we saw in the past two lectures, estimation is quite simple in this supervised case.

The marginal distribution of LDA

Using the law of total probability:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k).$$

What is the name of such a density?
This a called a Gaussian mixture model (GMM) with K components and a common covariance.

LDA is a supervised version of GMM. As we saw in the past two lectures, estimation is quite simple in this supervised case. Today, we will see an unsupervised version of GMM, that can be used for clustering or density estimation. We will see that estimating the parameters is much trickier in this unsupervised setting.

Recap on Linear Discriminant Analysis

GMM basics

Estimation for unsupervised GMMs: the EM algorithm

Unsupervised Gaussian mixture models

We have some data $\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^D$. A Gaussian mixture model with K clusters is a statistical model $(p_\theta, \theta \in \Theta_K)$ with the following density

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

with $\theta = (\pi_1, ..., \pi_K, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_K)$.

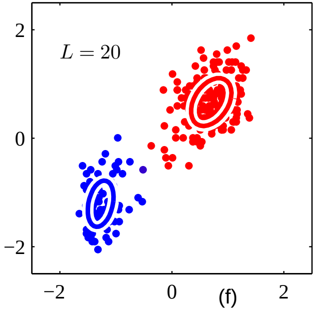

Figure: **Figure 9.8 from Bishop's book**.

Unsupervised Gaussian mixture models

We have some data $\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^D$. A Gaussian mixture model with K clusters is a statistical model $(p_\theta, \theta \in \Theta_K)$ with the following density

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

with $\theta = (\pi_1, ..., \pi_K, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_K)$.
Question: What are the differences between this and LDA?

[4] It is also possible to be unsupervised with a common covariance, or supervised with different covariances...

Unsupervised Gaussian mixture models

We have some data $\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^D$. A Gaussian mixture model with K clusters is a statistical model $(p_\theta, \theta \in \Theta_K)$ with the following density

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

with $\theta = (\pi_1, ..., \pi_K, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_K)$.
Question: What are the differences between this and LDA?

- Here, we don't have access to labels, we only see $\mathbf{x}_1, ..., \mathbf{x}_N$!
- The covariances of the classes are different: $\boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_k$, while there was only a single common covariance in LDA.[4]

[4] It is also possible to be unsupervised with a common covariance, or supervised with different covariances...

Unsupervised Gaussian mixture models

We have some data $\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^D$. A Gaussian mixture model with K clusters is a statistical model $(p_\theta, \theta \in \Theta_K)$ with the following density

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

with $\theta = (\pi_1, ..., \pi_K, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_K)$.
Question: What are the similarities between this and LDA?

[5] we often say "cluster" data in the unsupervised case, and "classify" data in the supervised case.

Unsupervised Gaussian mixture models

We have some data $\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^D$. A Gaussian mixture model with K clusters is a statistical model $(p_\theta, \theta \in \Theta_K)$ with the following density

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

with $\theta = (\pi_1, ..., \pi_K, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_K)$.
Question: What are the similarities between this and LDA?

- The classes are Gaussian: $\mathbf{x}|k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
- One can cluster[5] data using Bayes's rule $p_\theta(k|\mathbf{x}) = p_\theta(\mathbf{x}|k)\pi_k / p_\theta(\mathbf{x})$

[5] we often say "cluster" data in the unsupervised case, and "classify" data in the supervised case.

The parameters of a Gaussian mixture model

Consider the model

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

with $\theta = (\pi_1, ..., \pi_K, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_K)$.

What are the constraints on the parameters? In other words, in which space the proportions, means, and covariances live?

The proportions and means

The proportions must sum to one! So

$$(\pi_1, ..., \pi_K) \in \Delta_K,$$

with $\Delta_K = \{\mathbf{t} \in \mathbb{R}^K, t_1 + ... + t_K = 1\}$. This set is usually called the simplex.

The means $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K$ are simply vectors in $\mathbb{R}^D$.

The covariances $\mathbf{\Sigma}_1, ..., \mathbf{\Sigma}_K$ must be symmetric positive definite matrices. This means that they are symmetric and all their eigenvalues are strictly positive.

Recap on Linear Discriminant Analysis

GMM basics

Estimation for unsupervised GMMs: the EM algorithm

The nasty likelihood of GMMs

The goal is now to estimate our parameters $\theta = (\pi_1, ..., \pi_K, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_K)$. A natural approach would be to maximise the log-likelihood function

$$\ell(\theta) = \sum_{n=1}^{N} \log p_\theta(x_n) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right).$$

The nasty likelihood of GMMs

The goal is now to estimate our parameters $\theta = (\pi_1, ..., \pi_K, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_K)$. A natural approach would be to maximise the log-likelihood function

$$\ell(\theta) = \sum_{n=1}^{N} \log p_\theta(x_n) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right).$$

Unfortunately, this function is quite nasty, and difficult to optimise. In particular, there are no closed-form maximisers. Here, we will see a quite simple and clever way of iteratively maximising $\ell(\theta)$ called the EM algorithm.

The nasty likelihood of GMMs

The goal is now to estimate our parameters $\theta = (\pi_1, ..., \pi_K, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_K)$. A natural approach would be to maximise the log-likelihood function

$$\ell(\theta) = \sum_{n=1}^{N} \log p_\theta(x_n) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right).$$

Unfortunately, this function is quite nasty, and difficult to optimise. In particular, there are no closed-form maximisers. Here, we will see a quite simple and clever way of iteratively maximising $\ell(\theta)$ called the EM algorithm. There are many ways to expose this very classical algorithm.

Here, we will follow Sections 9.2.1 and 9.2.2 of Bishop's book. We may see other perspectives on EM later in the course.

The nasty likelihood of GMMs

A few comments about why it is hard to maximise the likelihood of GMMs. For simplicity, let's look at the "simple" case with $K = 2$ classes:

- It's clear that the likelihood of $\theta = (\pi_1, \pi_2, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2)$ and $\tilde{\theta} = (\pi_2, \pi_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_2, \boldsymbol{\Sigma}_1)$ will be the same. However, there two parameters θ $\tilde{\theta}$ may be very far away in parameter space! This means that the likelihood may have many local optima!

The nasty likelihood of GMMs

A few comments about why it is hard to maximise the likelihood of GMMs. For simplicity, let's look at the "simple" case with $K = 2$ classes:

- A GMM can completely overfit a single data point. More precisely, there are some singularities in the likelihood: if $\boldsymbol{\mu}_1 = \mathbf{x}_1$ and $\boldsymbol{\Sigma}_1 = \varepsilon \mathbf{I}$, then $\ell(\theta) \longrightarrow \infty$ when $\varepsilon \longrightarrow 0$. The model completely overfits $\mathbf{x}_1$.

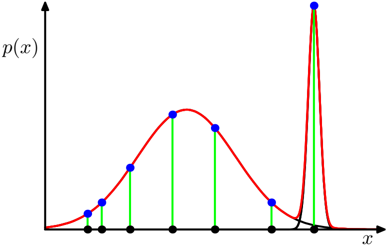

Figure: Figure 9.7 from Bishop's book. The GMM completely overfits the largest point.

A first approach: computing gradients

So, how do we maximise $\ell(\theta)$? A first idea would be to try to solve the equation

$$\nabla_\theta \ell(\theta) = 0$$

that characterises local optima of $\ell(\theta)$. Let's try.

A first approach: computing gradients

So, how do we maximise $\ell(\theta)$? A first idea would be to try to solve the equation

$$\nabla_{\theta}\ell(\theta) = 0$$

that characterises local optima of $\ell(\theta)$. Let's try. So, θ is composed of

means, covariances, and proportions... Let's start with the means. Try to compute $\nabla_{\boldsymbol{\mu}_k}\ell(\theta)$ for all k!

Computing gradients for the means

Try to compute $\nabla_{\boldsymbol{\mu}_k} \ell(\theta)$ for all k! Let's recall that

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

Computing gradients for the means

Try to compute $\nabla_{\boldsymbol{\mu}_k} \ell(\theta)$ for all k! Let's recall that

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

Hints.

- $\nabla_{w_k} \log(w_1 + \ldots + w_K) = \frac{w_k}{w_1 + \ldots + w_K}$
- $\nabla_{\boldsymbol{\mu}} \log \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$

Computing gradients for the means

Try to compute $\nabla_{\boldsymbol{\mu}_k}\ell(\theta)$ for all k! Let's recall that

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

Hints.

- $\nabla_{w_k} \log(w_1 + \ldots + w_K) = \frac{w_k}{w_1+\ldots+w_K}$
- $\nabla_{\boldsymbol{\mu}} \log \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$

  Combining the two and the chain rule gives:

$$\nabla_{\boldsymbol{\mu}_k}\ell(\theta) = \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

Computing gradients for the means

The chain rule gives

$$\nabla_{\boldsymbol{\mu}_k} \ell(\theta) = \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k).$$

Question: Can we simplify the weird quotient $\frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=\mathbf{1}}^{K} \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$?

Computing gradients for the means

The chain rule gives

$$\nabla_{\boldsymbol{\mu}_k}\ell(\theta) = \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k).$$

Question: Can we simplify the weird quotient $\frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=\mathbf{1}}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$?

Since $\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = p(\mathbf{x}_n|k)$, we can see that

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

Computing gradients for the means

The chain rule gives

$$\nabla_{\boldsymbol{\mu}_k} \ell(\theta) = \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k).$$

Question: Can we simplify the weird quotient $\frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$?

Since $\mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = p(\mathbf{x}_n | k)$, we can see that

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

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

Computing gradients for the means

The chain rule gives

$$\nabla_{\boldsymbol{\mu}_k} \ell(\theta) = \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k).$$

with

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

$p(k|\mathbf{x}_n)$ are called the responsabilities, or posterior probabilities. They just

depend on n and K, and will be denoted $\gamma(z_{nk})$ (as in Bishop, where $\mathbf{z}_n = (z_{n1}, ..., z_{nK})$ is a 1-hot encoded version of the classes), or τ_{nk} (Charles likes this notation).

Solving the gradient equations for the means

We want to solve $\nabla_{\boldsymbol{\mu}_k} \ell(\theta) = 0$. According to the previous slide, this means

$$\nabla_{\boldsymbol{\mu}_k} \ell(\theta) = \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) = 0.$$

Solving the gradient equations for the means

We want to solve $\nabla_{\boldsymbol{\mu}_k} \ell(\theta) = 0$. According to the previous slide, this means

$$\nabla_{\boldsymbol{\mu}_k} \ell(\theta) = \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) = 0.$$

By multiplying by $\boldsymbol{\Sigma}_k$, we get

$$\sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k) = 0,$$

and finally

Solving the gradient equations for the means

We want to solve $\nabla_{\boldsymbol{\mu}_k} \ell(\theta) = 0$. According to the previous slide, this means

$$\nabla_{\boldsymbol{\mu}_k} \ell(\theta) = \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) = 0.$$

By multiplying by $\boldsymbol{\Sigma}_k$, we get

$$\sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k) = 0,$$

and finally

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n, \text{ where } N_k = \sum_{n=1}^{N} \gamma(z_{nk}).$$

Solving the gradient equations for the means

We want to solve $\nabla_{\boldsymbol{\mu}_k} \ell(\theta) = 0$. According to the previous slide, this means

$$\nabla_{\boldsymbol{\mu}_k} \ell(\theta) = \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) = 0.$$

By multiplying by $\boldsymbol{\Sigma}_k$, we get

$$\sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k) = 0,$$

and finally

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n, \text{ where } N_k = \sum_{n=1}^{N} \gamma(z_{nk}).$$

N_k can be interpreted as the effective number of points in cluster k. The estimate of $\boldsymbol{\mu}_k$ is a weighted mean of the data set, where the weights are the responsabilities of cluster k.

Solving the gradient equations for the covariances

We want to solve $\nabla_{\boldsymbol{\Sigma}_k} \ell(\theta) = 0$. We won't do the detailed math here, but exactly like the estimate of $\boldsymbol{\mu}_k$ was a weighted version of the empirical mean, the estimate of $\boldsymbol{\Sigma}_k$ is a weighted version of the empirical covariance:

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T.$$

Solving the gradient equations for the covariances

We want to solve $\nabla_{\boldsymbol{\Sigma}_k} \ell(\theta) = 0$. We won't do the detailed math here, but exactly like the estimate of $\boldsymbol{\mu}_k$ was a weighted version of the empirical mean, the estimate of $\boldsymbol{\Sigma}_k$ is a weighted version of the empirical covariance:

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T.$$

Now we need to also take care of the proportions $\pi_1, ..., \pi_K$. Using standard constrained optimisation techniques (see Bishop, p.436) shows that the proportion of class k is just the average responsability of cluster k:

$$\pi_k = \frac{N_K}{N}.$$

Towards the EM algorithm

Ok, so what have we done? We solved three different equations for all k

$$\nabla_{\boldsymbol{\mu}_k} \ell(\theta) = 0$$

$$\nabla_{\boldsymbol{\Sigma}_k} \ell(\theta) = 0$$

$$\nabla_{\pi_k} \ell(\theta) = 0.$$

The EM algorithm will just be cycling through these three formulas in an iterative fashion!

How do we do this in practice?

Towards the EM algorithm

We saw that the responsabilities $\gamma(z_{nk}) = p(k|\mathbf{x}_n)$ were crucial in all three formulas. We should probably start by computing these then.

Towards the EM algorithm

We saw that the responsabilities $\gamma(z_{nk}) = p(k|\mathbf{x}_n)$ were crucial in all three formulas. We should probably start by computing these then.
This will be the

$$\text{E-Step: For all } n, k, \text{ compute } \gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

Towards the EM algorithm

We saw that the responsabilities $\gamma(z_{nk}) = p(k|\mathbf{x}_n)$ were crucial in all three formulas. We should probably start by computing these then.
This will be the

$$\text{E-Step: For all } n, k, \text{ compute } \gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

Once we have the responsabilities, we can cycle through our three equations that will collectively constitute our M-step:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

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

After each iteration of the EM algorithm (one E step followed by one M step), we can monitor convergence by looking at the evolution of the value of the log-lik

Towards the EM algorithm

After each iteration of the EM algorithm (one E step followed by one M step), we can monitor convergence by looking at the evolution of the value of the log-likelihood (that will be hopefully be maximised).
Now we'll try to implement EM!