Ch. 7 Probabilistic Learning

Gaussian Mixture Models

Chen Hu

Mixture of Gaussians

* Unsupervised Learning. We have a training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$ with no labels, we want to find the cluster assignment z for x.

* Model the data by specifying a joint distribution

$$p\left(x^{(i)}, z^{(i)}\right) = p\left(x^{(i)} \middle| z^{(i)}\right) p(z^{(i)})$$

where $z^{(i)} \sim Multinomial(\phi), \phi_j \geq 0, \sum_{j=1}^{k} \phi_j = 1$, and

Probability of choosing cluster j → $\phi_j = p\left(z^{(i)} = j\right)$, $x^{(i)} | (z^{(i)} = j) \sim \mathcal{N}(\mu_j, \Sigma_j)$

k is the number of values $z^{(i)}$ can take on.

Mixture of Gaussians

* Unsupervised Learning. We have a training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$ with no labels, we want to find the cluster assignment z for x.

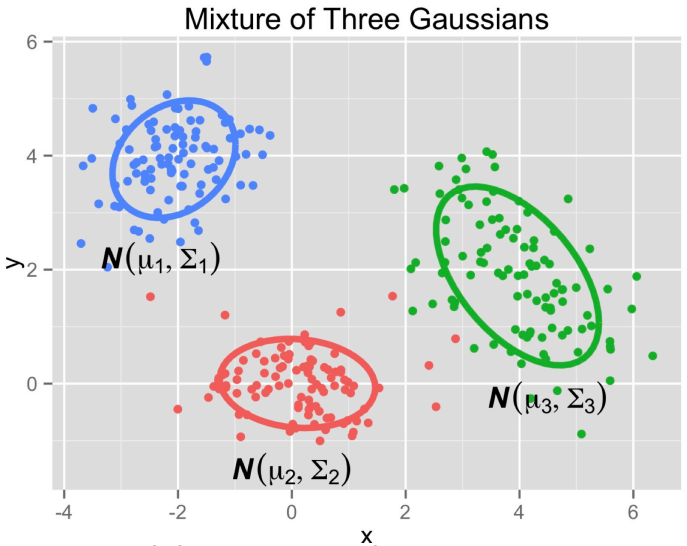
* Model the data by specifying a joint distribution

$$p\left(x^{(i)}, z^{(i)}\right) = p\left(x^{(i)} \middle| z^{(i)}\right) p(z^{(i)})$$

where $z^{(i)} \sim Multinomial(\phi), \phi_j \geq 0, \sum_{j=1}^{k} \phi_j = 1$, and

Probability of choosing cluster j → $\phi_j = p\left(z^{(i)} = j\right)$, $x^{(i)} | (z^{(i)} = j) \sim \mathcal{N}(\mu_j, \Sigma_j)$

k is the number of values $z^{(i)}$ can take on.



blue = # red = # green

$$\phi_{blue} = \phi_{red} = \phi_{green} = \frac{1}{3}$$

$$p\left(x^{(i)} \middle| z^{(i)} = blue\right) = 0.789101$$
$$p\left(x^{(i)} \middle| z^{(i)} = red\right) = 0.000042$$
$$p\left(x^{(i)} \middle| z^{(i)} = \text{green}\right) = 0.000021$$

Mixture of Gaussians

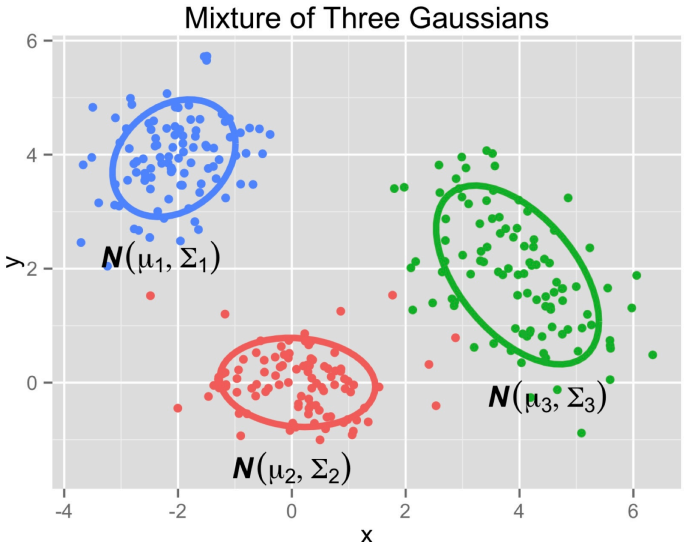
* Unsupervised Learning. We have a training set $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ with no labels, we want to find the cluster assignment z for x.

* Model the data by specifying a joint distribution

$$p\left(x^{(i)}, z^{(i)}\right) = p\left(x^{(i)} \middle| z^{(i)}\right) p(z^{(i)})$$

where $z^{(i)} \sim Multinomial(\phi), \phi_j \geq 0, \sum_{j=1}^{k} \phi_j = 1$, and

Probability of choosing cluster j → $\phi_j = p\left(z^{(i)} = j\right)$, $x^{(i)} | (z^{(i)} = j) \sim \mathcal{N}(\mu_j, \Sigma_j)$



k is the number of values $z^{(i)}$ can take on.

Our model posits that $x^{(i)}$ was generated by randomly choosing $z^{(i)}$ from $\{1, 2, \dots, k\}$, then $x^{(i)}$ was drawn from one of k Gaussians depending on $z^{(i)}$.

Mixture of Gaussians

* Unsupervised Learning. We have a training set $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ with no labels, we want to find the cluster assignment z for x.

* Model the data by specifying a joint distribution

$$p\left(x^{(i)}, z^{(i)}\right) = p\left(x^{(i)} \middle| z^{(i)}\right) p(z^{(i)})$$

$z^{(i)}$ are latent random variables, they are HIDDEN/UNOBSERVED!
This is what makes our problem difficult.

Mixture of Gaussians

The parameters of our model are ϕ, μ and Σ. To estimate them, we can write down the likelihood of our data:

$$\ell(\phi, \mu, \Sigma) = \sum_{i=1}^{m} \log p(x^{(i)};\ \phi, \mu, \Sigma) = \sum_{i=1}^{m} \log \sum_{z^{(i)}=1}^{k} p\left(x^{(i)} \middle| z^{(i)}; \mu, \Sigma\right) p(z^{(i)}, \phi)$$

If we knew what $z^{(i)}$ were, the problem would have been easy, and we can write the likelihood as

$$\ell(\phi, \mu, \Sigma) = \sum_{i=1}^{m} \log p\left(x^{(i)} \middle| z^{(i)}; \mu, \Sigma\right) + \log p(z^{(i)}, \phi)$$

Mixture of Gaussians

$$\ell(\phi, \mu, \Sigma) = \sum_{i=1}^{m} \log p\left(x^{(i)} \middle| z^{(i)}; \mu, \Sigma\right) + \log p(z^{(i)}, \phi)$$

We now maximize the above equation w.r.t. ϕ, μ and Σ, note that we assume for each $x^{(i)}$, we already know which cluster j it belongs to,

$$\phi_j = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\}$$

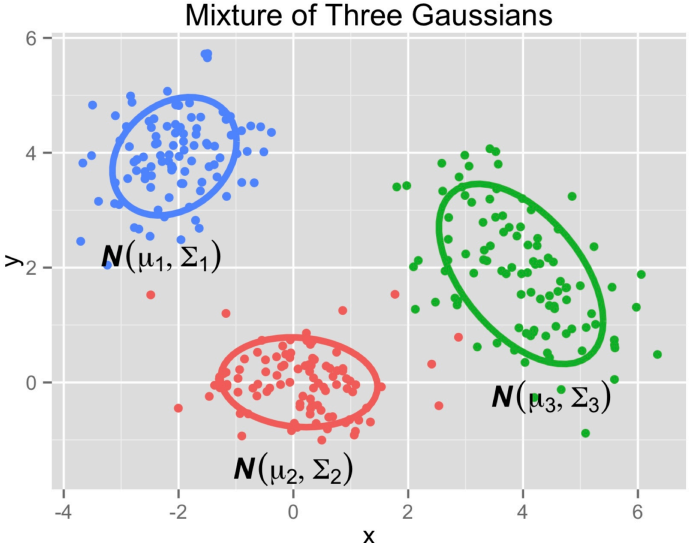
Proportion of samples in the dataset that belong to cluster *j*

Mean of samples in cluster *j*

$$\mu_j = \frac{\sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\}}$$

Covariance matrix of samples in cluster *j*

$$\Sigma_j = \frac{\sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\}}$$



EM (Expectation-Maximization) Algorithm

* A two-step iterative algorithm:
  E-step - "guess" the value of $z^{(i)}$;
  M-step - update the parameters of our model based on guesses.

E-step

For each $i,\ j$, set

$$w_j^{(i)} = p(z^{(i)} = j | x^{(i)};\ \phi, \mu, \Sigma)$$

M-step

Update parameters

$$\phi_j = \frac{1}{m}\sum_{i=1}^{m} w_j^{(i)}$$

$$\mu_j = \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{m} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_j^{(i)}}$$

EM (Expectation-Maximization) Algorithm

* A two-step iterative algorithm:
  E-step - "guess" the value of $z^{(i)}$;
  M-step - update the parameters of our model based on guesses.

$z^{(i)}$ are known

$$\phi_j = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\}$$

$$\mu_j = \frac{\sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} \mathbf{1}\{z^{(i)} = j\}}$$

M-step

Update parameters

$$\phi_j = \frac{1}{m}\sum_{i=1}^{m} w_j^{(i)}$$

$$\mu_j = \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{m} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_j^{(i)}}$$

EM (Expectation-Maximization) Algorithm

* A two-step iterative algorithm:
  E-step - "guess" the value of $z^{(i)}$;
  M-step - update the parameters of our model based on guesses.

E-step

For each $i,\ j$, set

$$w_j^{(i)} = p\left(z^{(i)} = j \middle| x^{(i)};\ \phi, \mu, \Sigma\right)$$

Calculate the posterior probability of $z^{(i)}$ using Bayes rule

$$p\left(z^{(i)} = j \middle| x^{(i)};\ \phi, \mu, \Sigma\right) = \frac{p\left(z^{(i)} = j \middle| \phi\right) p\left(x^{(i)} \middle| z^{(i)};\ \mu, \Sigma\right)}{\sum_{l=1}^{k} p\left(x^{(i)} \middle| z^{(i)} = l;\ \mu, \Sigma\right) p(z^{(i)} = l;\ \phi)}$$

M-step

$$\phi_j = \frac{1}{m} \sum_{i=1}^{m} w_j^{(i)}$$

$$\mu_j = \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}}$$

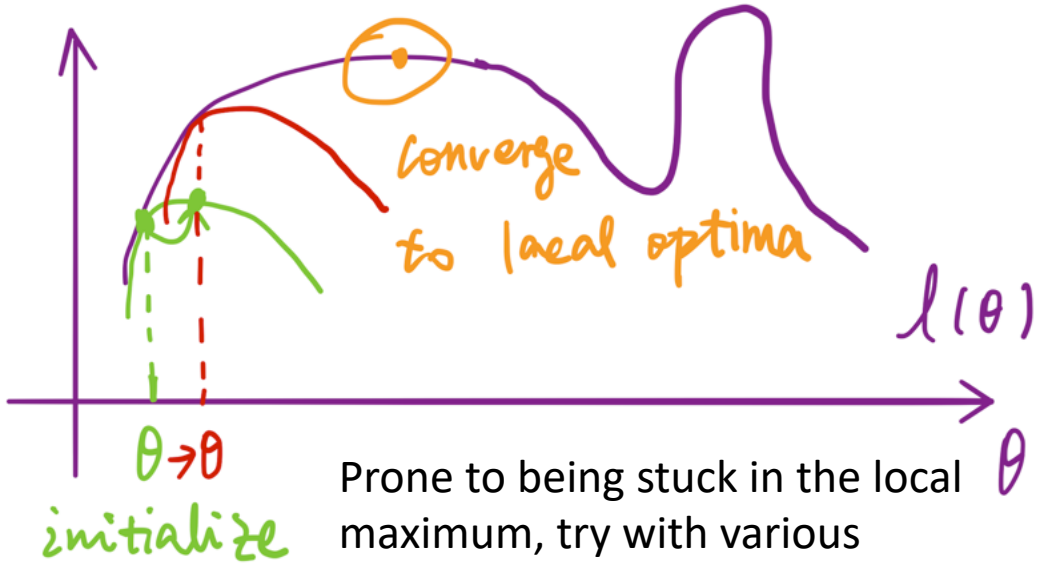
$$\Sigma_j = \frac{\sum_{i=1}^{m} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_j^{(i)}}$$

EM (Expectation-Maximization) Algorithm

* Jensen's Inequality – the big picture of EM algorithm

Let f be a convex function, and let X be a random variable, then

$$E[f(X)] \geq f(E[X])$$



Prone to being stuck in the local maximum, try with various initial values.

E-step - construct a lower bound for the log likelihood (the green curve);

M-step - find the max on the lower bound, then move parameters to there;

E-step - construct a new lower bound for the log likelihood (the red curve);

…

Converge at the orange local optima.

Information Criteria

* Aikake Information Criterium

$$AIC = \ln(largest\ likelihood) - (\#\ parameters)$$

* Bayesian Information Criterium

$$BIC = 2\ln(largest\ likelihood) - (\#\ parameters)\ln(\#\ examples)$$

Both favor simple models.