Mixture Models and GMM (Contd.)

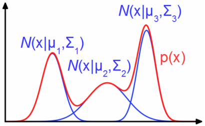
Piyush Rai

Probabilistic Machine Learning (CS772A)

Aug 31, 2017

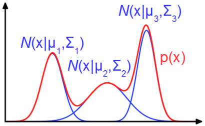
Recap: Gaussian Mixture Model

- A model for data clustering and density estimation
- Assumes data generated from a mixture of K Gaussians with mixing proportions $\pi_1, \ldots, \pi_K$



Recap: Gaussian Mixture Model

- A model for data clustering and density estimation
- Assumes data generated from a mixture of K Gaussians with mixing proportions $\pi_1, \ldots, \pi_K$

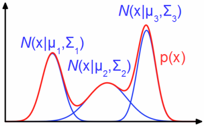


- The prior probability of observation $\boldsymbol{x}_n$ generated from the k-th Gaussian

$$p(\boldsymbol{z}_n = k|\pi) = \pi_k$$

Recap: Gaussian Mixture Model

- A model for data clustering and density estimation
- Assumes data generated from a mixture of K Gaussians with mixing proportions $\pi_1, \ldots, \pi_K$



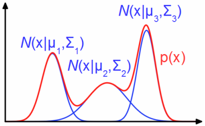
- The prior probability of observation $\boldsymbol{x}_n$ generated from the k-th Gaussian

$$p(\boldsymbol{z}_n = k|\pi) = \pi_k$$

- Same as a multinoulli prior on $\boldsymbol{z}$, i.e., $p(\boldsymbol{z}_n|\pi) = \prod_{k=1}^K \pi_k^{z_{nk}}$

Recap: Gaussian Mixture Model

- A model for data clustering and density estimation
- Assumes data generated from a mixture of K Gaussians with mixing proportions $\pi_1, \ldots, \pi_K$



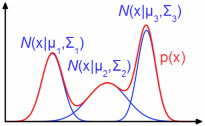
- The prior probability of observation $\boldsymbol{x}_n$ generated from the k-th Gaussian

$$p(\boldsymbol{z}_n = k|\pi) = \pi_k$$

- Same as a multinoulli prior on $\boldsymbol{z}$, i.e., $p(\boldsymbol{z}_n|\pi) = \prod_{k=1}^{K} \pi_k^{z_{nk}}$ (note $z_{nk} = 1$ if $\boldsymbol{z}_n = k$; 0 otherwise)

Recap: Gaussian Mixture Model

- A model for data clustering and density estimation
- Assumes data generated from a mixture of K Gaussians with mixing proportions $\pi_1, \ldots, \pi_K$



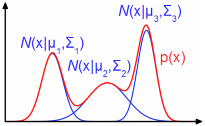
- The prior probability of observation $\boldsymbol{x}_n$ generated from the k-th Gaussian

$$p(\boldsymbol{z}_n = k|\pi) = \pi_k$$

- Same as a multinoulli prior on $\boldsymbol{z}$, i.e., $p(\boldsymbol{z}_n|\pi) = \prod_{k=1}^{K} \pi_k^{z_{nk}}$ (note $z_{nk} = 1$ if $\boldsymbol{z}_n = k$; 0 otherwise)
- If $\boldsymbol{z}_n = k$, we generate $\boldsymbol{x}_n$ from the k-th Gaussian.

Recap: Gaussian Mixture Model

- A model for data clustering and density estimation
- Assumes data generated from a mixture of K Gaussians with mixing proportions $\pi_1, \ldots, \pi_K$



- The prior probability of observation $\boldsymbol{x}_n$ generated from the k-th Gaussian

$$p(\boldsymbol{z}_n = k|\pi) = \pi_k$$

- Same as a multinoulli prior on $\boldsymbol{z}$, i.e., $p(\boldsymbol{z}_n|\pi) = \prod_{k=1}^{K} \pi_k^{z_{nk}}$ (note $z_{nk} = 1$ if $\boldsymbol{z}_n = k$; 0 otherwise)
- If $\boldsymbol{z}_n = k$, we generate $\boldsymbol{x}_n$ from the k-th Gaussian. Thus

$$p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

Recap: Gaussian Mixture Model (GMM)

- The marginal distribution of $\boldsymbol{x}_n$ (requires summing over all possibilities of $\boldsymbol{z}_n$)

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

Recap: Gaussian Mixture Model (GMM)

- The marginal distribution of $\boldsymbol{x}_n$ (requires summing over all possibilities of $\boldsymbol{z}_n$)

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

where $\Theta = (\pi, \{\mu_k, \Sigma_k\}_{k=1}^{K})$

Recap: Gaussian Mixture Model (GMM)

- The marginal distribution of $\boldsymbol{x}_n$ (requires summing over all possibilities of $\boldsymbol{z}_n$)

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

where $\Theta = (\pi, \{\mu_k, \Sigma_k\}_{k=1}^{K})$

- The MLE objective for the GMM will be

$$\sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\Theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

Recap: Gaussian Mixture Model (GMM)

- The marginal distribution of $\boldsymbol{x}_n$ (requires summing over all possibilities of $\boldsymbol{z}_n$)

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

  where $\Theta = (\pi, \{\mu_k, \Sigma_k\}_{k=1}^{K})$

- The MLE objective for the GMM will be

$$\sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\Theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

- Doing MLE on this objective is tricky due to the "$\log \sum$" term

Recap: Gaussian Mixture Model (GMM)

- The marginal distribution of $\boldsymbol{x}_n$ (requires summing over all possibilities of $\boldsymbol{z}_n$)

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

where $\Theta = (\pi, \{\mu_k, \Sigma_k\}_{k=1}^{K})$

- The MLE objective for the GMM will be

$$\sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\Theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

- Doing MLE on this objective is tricky due to the "$\log \sum$" term
  - Parameters get coupled; no closed form solution (iterative methods needed, slow convergence)

Recap: Gaussian Mixture Model (GMM)

- The marginal distribution of $\boldsymbol{x}_n$ (requires summing over all possibilities of $\boldsymbol{z}_n$)

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

  where $\Theta = (\pi, \{\mu_k, \Sigma_k\}_{k=1}^{K})$

- The MLE objective for the GMM will be

$$\sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\Theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

- Doing MLE on this objective is tricky due to the "$\log \sum$" term
  - Parameters get coupled; no closed form solution (iterative methods needed, slow convergence)
- An alternating "guess, re-estimate, and repeat until converge" algorithm helps solve such problems in a clean, simple, and efficient way; basically, the Expectation Maximization (EM) algorithm

GMM Parameter Estimation

- MLE for GMM (and LVMs in general) becomes simple if we knew (or "guess") the $\boldsymbol{z}_n$ for each $\boldsymbol{x}_n$

GMM Parameter Estimation

- MLE for GMM (and LVMs in general) becomes simple if we knew (or "guess") the $\boldsymbol{z}_n$ for each $\boldsymbol{x}_n$
- Reason: If $\boldsymbol{z}_n$ is known (suppose $\boldsymbol{z}_n = k$), then the summation over $\boldsymbol{z}_n$ isn't required

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

GMM Parameter Estimation

- MLE for GMM (and LVMs in general) becomes simple if we knew (or "guess") the $\boldsymbol{z}_n$ for each $\boldsymbol{x}_n$
- Reason: If $\boldsymbol{z}_n$ is known (suppose $\boldsymbol{z}_n = k$), then the summation over $\boldsymbol{z}_n$ isn't required

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

- With $\boldsymbol{z}_n$ known, can treat $(\boldsymbol{x}_n, \boldsymbol{z}_n)$ as our "data" and do MLE on $p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta)$, instead of $p(\boldsymbol{x}_n|\Theta)$

GMM Parameter Estimation

- MLE for GMM (and LVMs in general) becomes simple if we knew (or "guess") the $\boldsymbol{z}_n$ for each $\boldsymbol{x}_n$
- Reason: If $\boldsymbol{z}_n$ is known (suppose $\boldsymbol{z}_n = k$), then the summation over $\boldsymbol{z}_n$ isn't required

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

- With $\boldsymbol{z}_n$ known, can treat $(\boldsymbol{x}_n, \boldsymbol{z}_n)$ as our "data" and do MLE on $p(\boldsymbol{x}_n, \boldsymbol{z}_n | \Theta)$, instead of $p(\boldsymbol{x}_n | \Theta)$
  - $p(\boldsymbol{x}_n, \boldsymbol{z}_n | \Theta)$ is known as "complete data likelihood" (CLL) - $\boldsymbol{z}_n$ makes $\boldsymbol{x}_n$ "complete"

GMM Parameter Estimation

- MLE for GMM (and LVMs in general) becomes simple if we knew (or "guess") the $\boldsymbol{z}_n$ for each $\boldsymbol{x}_n$
- Reason: If $\boldsymbol{z}_n$ is known (suppose $\boldsymbol{z}_n = k$), then the summation over $\boldsymbol{z}_n$ isn't required

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

- With $\boldsymbol{z}_n$ known, can treat $(\boldsymbol{x}_n, \boldsymbol{z}_n)$ as our "data" and do MLE on $p(\boldsymbol{x}_n, \boldsymbol{z}_n | \Theta)$, instead of $p(\boldsymbol{x}_n | \Theta)$
  - $p(\boldsymbol{x}_n, \boldsymbol{z}_n | \Theta)$ is known as "complete data likelihood" (CLL) - $\boldsymbol{z}_n$ makes $\boldsymbol{x}_n$ "complete"
  - $p(\boldsymbol{x}_n | \Theta)$ is known as "incomplete data likelihood" (ILL)

GMM Parameter Estimation

- MLE for GMM (and LVMs in general) becomes simple if we knew (or "guess") the $\boldsymbol{z}_n$ for each $\boldsymbol{x}_n$
- Reason: If $\boldsymbol{z}_n$ is known (suppose $\boldsymbol{z}_n = k$), then the summation over $\boldsymbol{z}_n$ isn't required

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

- With $\boldsymbol{z}_n$ known, can treat $(\boldsymbol{x}_n, \boldsymbol{z}_n)$ as our "data" and do MLE on $p(\boldsymbol{x}_n, \boldsymbol{z}_n | \Theta)$, instead of $p(\boldsymbol{x}_n | \Theta)$
  - $p(\boldsymbol{x}_n, \boldsymbol{z}_n | \Theta)$ is known as "complete data likelihood" (CLL) - $\boldsymbol{z}_n$ makes $\boldsymbol{x}_n$ "complete"
  - $p(\boldsymbol{x}_n | \Theta)$ is known as "incomplete data likelihood" (ILL)
- Notation: $\boldsymbol{z}_n = [z_{n1}, z_{n2}, \ldots, z_{nK}]$ has a one-hot representation (note: only a single z_{nk} will be 1)

GMM Parameter Estimation

- MLE for GMM (and LVMs in general) becomes simple if we knew (or "guess") the $\boldsymbol{z}_n$ for each $\boldsymbol{x}_n$
- Reason: If $\boldsymbol{z}_n$ is known (suppose $\boldsymbol{z}_n = k$), then the summation over $\boldsymbol{z}_n$ isn't required

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

- With $\boldsymbol{z}_n$ known, can treat $(\boldsymbol{x}_n, \boldsymbol{z}_n)$ as our "data" and do MLE on $p(\boldsymbol{x}_n, \boldsymbol{z}_n | \Theta)$, instead of $p(\boldsymbol{x}_n | \Theta)$
  - $p(\boldsymbol{x}_n, \boldsymbol{z}_n | \Theta)$ is known as "complete data likelihood" (CLL) - $\boldsymbol{z}_n$ makes $\boldsymbol{x}_n$ "complete"
  - $p(\boldsymbol{x}_n | \Theta)$ is known as "incomplete data likelihood" (ILL)
- Notation: $\boldsymbol{z}_n = [z_{n1}, z_{n2}, \ldots, z_{nK}]$ has a one-hot representation (note: only a single z_{nk} will be 1)
- Denoting all the GMM parameters by Θ, the complete data log-likelihood will be

$$\mathrm{CLL}(\Theta) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n, \boldsymbol{z}_n | \Theta)$$

GMM Parameter Estimation

- MLE for GMM (and LVMs in general) becomes simple if we knew (or "guess") the $\boldsymbol{z}_n$ for each $\boldsymbol{x}_n$
- Reason: If $\boldsymbol{z}_n$ is known (suppose $\boldsymbol{z}_n = k$), then the summation over $\boldsymbol{z}_n$ isn't required

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

- With $\boldsymbol{z}_n$ known, can treat $(\boldsymbol{x}_n, \boldsymbol{z}_n)$ as our "data" and do MLE on $p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta)$, instead of $p(\boldsymbol{x}_n|\Theta)$
  - $p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta)$ is known as "complete data likelihood" (CLL) - $\boldsymbol{z}_n$ makes $\boldsymbol{x}_n$ "complete"
  - $p(\boldsymbol{x}_n|\Theta)$ is known as "incomplete data likelihood" (ILL)
- Notation: $\boldsymbol{z}_n = [z_{n1}, z_{n2}, \ldots, z_{nK}]$ has a one-hot representation (note: only a single z_{nk} will be 1)
- Denoting all the GMM parameters by Θ, the complete data log-likelihood will be

$$\text{CLL}(\Theta) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\boldsymbol{z}_n)p(\boldsymbol{z}_n)$$

GMM Parameter Estimation

- MLE for GMM (and LVMs in general) becomes simple if we knew (or "guess") the $\boldsymbol{z}_n$ for each $\boldsymbol{x}_n$
- Reason: If $\boldsymbol{z}_n$ is known (suppose $\boldsymbol{z}_n = k$), then the summation over $\boldsymbol{z}_n$ isn't required

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

- With $\boldsymbol{z}_n$ known, can treat $(\boldsymbol{x}_n, \boldsymbol{z}_n)$ as our "data" and do MLE on $p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta)$, instead of $p(\boldsymbol{x}_n|\Theta)$
  - $p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta)$ is known as "complete data likelihood" (CLL) - $\boldsymbol{z}_n$ makes $\boldsymbol{x}_n$ "complete"
  - $p(\boldsymbol{x}_n|\Theta)$ is known as "incomplete data likelihood" (ILL)
- Notation: $\boldsymbol{z}_n = [z_{n1}, z_{n2}, \ldots, z_{nK}]$ has a one-hot representation (note: only a single z_{nk} will be 1)
- Denoting all the GMM parameters by Θ, the complete data log-likelihood will be

$$\mathrm{CLL}(\Theta) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\boldsymbol{z}_n)p(\boldsymbol{z}_n) = \sum_{n=1}^{N} \log \prod_{k=1}^{K} [p(\boldsymbol{x}_n|\boldsymbol{z}_n = k)p(\boldsymbol{z}_n = k)]^{z_{nk}}$$

GMM Parameter Estimation

- The CLL gets further simplified to

$$\text{CLL}(\Theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \log p(\boldsymbol{z}_n = k) p(\boldsymbol{x}_n | \boldsymbol{z}_n = k)$$

GMM Parameter Estimation

- The CLL gets further simplified to

$$\mathrm{CLL}(\Theta) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \log p(\boldsymbol{z}_n = k) p(\boldsymbol{x}_n | \boldsymbol{z}_n = k) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} [\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

GMM Parameter Estimation

- The CLL gets further simplified to

$$\text{CLL}(\Theta) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \log p(\boldsymbol{z}_n = k)p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Simple algebraic form. MLE easy by taking (partial) derivatives w.r.t. each parameter

GMM Parameter Estimation

- The CLL gets further simplified to

$$\text{CLL}(\Theta) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \log p(\boldsymbol{z}_n = k)p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Simple algebraic form. MLE easy by taking (partial) derivatives w.r.t. each parameter
- However, to do so, we need the z_{nk}'s (each a binary r.v.).

GMM Parameter Estimation

- The CLL gets further simplified to

$$\text{CLL}(\Theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \log p(\boldsymbol{z}_n = k) p(\boldsymbol{x}_n | \boldsymbol{z}_n = k) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} [\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Simple algebraic form. MLE easy by taking (partial) derivatives w.r.t. each parameter
- However, to do so, we need the z_{nk}'s (each a binary r.v.). This is where we'll make a guess.

GMM Parameter Estimation

- The CLL gets further simplified to

$$\text{CLL}(\Theta) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\log p(\boldsymbol{z}_n = k)p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Simple algebraic form. MLE easy by taking (partial) derivatives w.r.t. each parameter
- However, to do so, we need the z_{nk}'s (each a binary r.v.). This is where we'll make a guess.
  - Idea: Use posterior expectation of z_{nk} as our guess

GMM Parameter Estimation

- The CLL gets further simplified to

$$\text{CLL}(\Theta) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \log p(\boldsymbol{z}_n = k) p(\boldsymbol{x}_n | \boldsymbol{z}_n = k) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} [\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Simple algebraic form. MLE easy by taking (partial) derivatives w.r.t. each parameter
- However, to do so, we need the z_{nk}'s (each a binary r.v.). This is where we'll make a guess.
  - Idea: Use posterior expectation of z_{nk} as our guess (seems reasonable; will see justification shortly)

GMM Parameter Estimation

- The CLL gets further simplified to

$$\text{CLL}(\Theta) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\log p(\boldsymbol{z}_n = k)p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Simple algebraic form. MLE easy by taking (partial) derivatives w.r.t. each parameter
- However, to do so, we need the z_{nk}'s (each a binary r.v.). This is where we'll make a guess.
    - Idea: Use posterior expectation of z_{nk} as our guess (seems reasonable; will see justification shortly)

$$\mathbb{E}[z_{nk}] = 0 \times p(z_{nk} = 0|\boldsymbol{x}_n) + 1 \times p(z_{nk} = 1|\boldsymbol{x}_n)$$

GMM Parameter Estimation

- The CLL gets further simplified to

$$\text{CLL}(\Theta) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \log p(\boldsymbol{z}_n = k) p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Simple algebraic form. MLE easy by taking (partial) derivatives w.r.t. each parameter
- However, to do so, we need the z_{nk}'s (each a binary r.v.). This is where we'll make a guess.
  - Idea: Use posterior expectation of z_{nk} as our guess (seems reasonable; will see justification shortly)

$$\begin{aligned} \mathbb{E}[z_{nk}] &= 0 \times p(z_{nk} = 0|\boldsymbol{x}_n) + 1 \times p(z_{nk} = 1|\boldsymbol{x}_n) \\ &= p(z_{nk} = 1|\boldsymbol{x}_n) \end{aligned}$$

GMM Parameter Estimation

- The CLL gets further simplified to

$$\text{CLL}(\Theta) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \log p(\boldsymbol{z}_n = k) p(\boldsymbol{x}_n | \boldsymbol{z}_n = k) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} [\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Simple algebraic form. MLE easy by taking (partial) derivatives w.r.t. each parameter
- However, to do so, we need the z_{nk}'s (each a binary r.v.). This is where we'll make a guess.
  - Idea: Use posterior expectation of z_{nk} as our guess (seems reasonable; will see justification shortly)

$$\begin{aligned}
\mathbb{E}[z_{nk}] &= 0 \times p(z_{nk} = 0 | \boldsymbol{x}_n) + 1 \times p(z_{nk} = 1 | \boldsymbol{x}_n) \\
&= p(z_{nk} = 1 | \boldsymbol{x}_n) \\
&\propto p(z_{nk} = 1) p(\boldsymbol{x}_n | z_{nk} = 1) \qquad \text{(from Bayes Rule)}
\end{aligned}$$

GMM Parameter Estimation

- The CLL gets further simplified to

$$\text{CLL}(\Theta) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \log p(\boldsymbol{z}_n = k)p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Simple algebraic form. MLE easy by taking (partial) derivatives w.r.t. each parameter
- However, to do so, we need the z_{nk}'s (each a binary r.v.). This is where we'll make a guess.
  - Idea: Use posterior expectation of z_{nk} as our guess (seems reasonable; will see justification shortly)

$$\begin{aligned}
\mathbb{E}[z_{nk}] &= 0 \times p(z_{nk} = 0|\boldsymbol{x}_n) + 1 \times p(z_{nk} = 1|\boldsymbol{x}_n) \\
&= p(z_{nk} = 1|\boldsymbol{x}_n) \\
&\propto p(z_{nk} = 1)p(\boldsymbol{x}_n|z_{nk} = 1) \qquad \text{(from Bayes Rule)} \\
\text{Thus} \quad \mathbb{E}[z_{nk}] &\propto \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad \text{(Posterior prob. that } \boldsymbol{x}_n \text{ is generated by } k\text{-th Gaussian)}
\end{aligned}$$

GMM Parameter Estimation

- The CLL gets further simplified to

$$\text{CLL}(\Theta) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\log p(\boldsymbol{z}_n = k)p(\boldsymbol{x}_n|\boldsymbol{z}_n = k) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Simple algebraic form. MLE easy by taking (partial) derivatives w.r.t. each parameter
- However, to do so, we need the z_{nk}'s (each a binary r.v.). This is where we'll make a guess.
  - Idea: Use posterior expectation of z_{nk} as our guess (seems reasonable; will see justification shortly)

$$\begin{aligned} \mathbb{E}[z_{nk}] &= 0 \times p(z_{nk} = 0|\boldsymbol{x}_n) + 1 \times p(z_{nk} = 1|\boldsymbol{x}_n) \\ &= p(z_{nk} = 1|\boldsymbol{x}_n) \\ &\propto p(z_{nk} = 1)p(\boldsymbol{x}_n|z_{nk} = 1) \qquad \text{(from Bayes Rule)} \\ \text{Thus} \quad \mathbb{E}[z_{nk}] &\propto \pi_k\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad \text{(Posterior prob. that } \boldsymbol{x}_n \text{ is generated by } k\text{-th Gaussian)} \end{aligned}$$

- Note: We can finally normalize $\mathbb{E}[z_{nk}]$ as $\mathbb{E}[z_{nk}] = \frac{\pi_k\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K}\pi_\ell\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell)}$ since $\sum_{k=1}^{K}\mathbb{E}[z_{nk}] = 1$

GMM Parameter Estimation

- The CLL gets further simplified to

$$\text{CLL}(\Theta) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \log p(\boldsymbol{z}_n = k) p(\boldsymbol{x}_n | \boldsymbol{z}_n = k) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} [\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Simple algebraic form. MLE easy by taking (partial) derivatives w.r.t. each parameter
- However, to do so, we need the z_{nk}'s (each a binary r.v.). This is where we'll make a guess.
  - Idea: Use posterior expectation of z_{nk} as our guess (seems reasonable; will see justification shortly)

$$\begin{aligned}
\mathbb{E}[z_{nk}] &= 0 \times p(z_{nk} = 0 | \boldsymbol{x}_n) + 1 \times p(z_{nk} = 1 | \boldsymbol{x}_n) \\
&= p(z_{nk} = 1 | \boldsymbol{x}_n) \\
&\propto p(z_{nk} = 1) p(\boldsymbol{x}_n | z_{nk} = 1) \qquad \text{(from Bayes Rule)} \\
\text{Thus} \quad \mathbb{E}[z_{nk}] &\propto \pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad \text{(Posterior prob. that } \boldsymbol{x}_n \text{ is generated by } k\text{-th Gaussian)}
\end{aligned}$$

- Note: We can finally normalize $\mathbb{E}[z_{nk}]$ as $\mathbb{E}[z_{nk}] = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)}$ since $\sum_{k=1}^{K} \mathbb{E}[z_{nk}] = 1$
- **Wait!** Computing $\mathbb{E}[z_{nk}]$ requires knowing $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ (chicken-and-egg problem ☹)

GMM Parameter Estimation: The Alternating Approach

Can solve the chicken-and-egg problem by taking an alternating approach

GMM Parameter Estimation: The Alternating Approach

Can solve the chicken-and-egg problem by taking an alternating approach

1. Assume $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$ is known.

GMM Parameter Estimation: The Alternating Approach

Can solve the chicken-and-egg problem by taking an alternating approach

1. Assume $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$ is known. Estimate the required expectations of the latent variables

$$\mathbb{E}[z_{nk}]$$

GMM Parameter Estimation: The Alternating Approach

Can solve the chicken-and-egg problem by taking an alternating approach

1. Assume $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$ is known. Estimate the required expectations of the latent variables

$$\mathbb{E}[z_{nk}] = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)}$$

GMM Parameter Estimation: The Alternating Approach

Can solve the chicken-and-egg problem by taking an alternating approach

1. Assume $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$ is known. Estimate the required expectations of the latent variables

$$\mathbb{E}[z_{nk}] = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)} \qquad (\text{for } n = 1, \ldots, N,\ k = 1, \ldots, K)$$

GMM Parameter Estimation: The Alternating Approach

Can solve the chicken-and-egg problem by taking an alternating approach

1. Assume $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$ is known. Estimate the required expectations of the latent variables

$$\mathbb{E}[z_{nk}] = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)} \qquad \text{(for } n = 1, \ldots, N,\ k = 1, \ldots, K\text{)}$$

2. Using $\mathbb{E}[z_{nk}]$ computed in step 1, maximize the expected CLL objective w.r.t. $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$

GMM Parameter Estimation: The Alternating Approach

Can solve the chicken-and-egg problem by taking an alternating approach

1. Assume $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$ is known. Estimate the required expectations of the latent variables

$$\mathbb{E}[z_{nk}] = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)} \qquad \text{(for } n = 1, \ldots, N,\ k = 1, \ldots, K\text{)}$$

2. Using $\mathbb{E}[z_{nk}]$ computed in step 1, maximize the <u>expected</u> CLL objective w.r.t. $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}[z_{nk}][\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

GMM Parameter Estimation: The Alternating Approach

Can solve the chicken-and-egg problem by taking an alternating approach

1. Assume $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$ is known. Estimate the required expectations of the latent variables

$$\mathbb{E}[z_{nk}] = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)} \qquad (\text{for } n = 1, \ldots, N,\ k = 1, \ldots, K)$$

2. Using $\mathbb{E}[z_{nk}]$ computed in step 1, maximize the expected CLL objective w.r.t. $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}[z_{nk}][\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

.. this will give ML estimates for parameters $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$. Details on the next slides.

GMM Parameter Estimation: The Alternating Approach

Can solve the chicken-and-egg problem by taking an alternating approach

1. Assume $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$ is known. Estimate the required expectations of the latent variables

$$\mathbb{E}[z_{nk}] = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)} \qquad (\text{for } n = 1, \ldots, N, \ k = 1, \ldots, K)$$

2. Using $\mathbb{E}[z_{nk}]$ computed in step 1, maximize the expected CLL objective w.r.t. $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}[z_{nk}][\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

.. this will give ML estimates for parameters $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$. Details on the next slides.

The algorithm alternates between step 1 and 2 until convergence

GMM Parameter Estimation: The Alternating Approach

Can solve the chicken-and-egg problem by taking an alternating approach

1. Assume $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$ is known. Estimate the required expectations of the latent variables

$$\mathbb{E}[z_{nk}] = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)} \qquad (\text{for } n = 1, \ldots, N,\ k = 1, \ldots, K)$$

2. Using $\mathbb{E}[z_{nk}]$ computed in step 1, maximize the expected CLL objective w.r.t. $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$

$$\mathcal{L} = \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}[z_{nk}][\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

.. this will give ML estimates for parameters $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$. Details on the next slides.

The algorithm alternates between step 1 and 2 until convergence

This is an example of the more general **Expectation Maximization** (EM) algorithm.

GMM Parameter Estimation: The Alternating Approach

Can solve the chicken-and-egg problem by taking an alternating approach

1. Assume $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$ is known. Estimate the required expectations of the latent variables

$$\mathbb{E}[z_{nk}] = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)} \qquad (\text{for } n = 1, \ldots, N,\ k = 1, \ldots, K)$$

2. Using $\mathbb{E}[z_{nk}]$ computed in step 1, maximize the expected CLL objective w.r.t. $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}[z_{nk}][\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

.. this will give ML estimates for parameters $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$. Details on the next slides.

The algorithm alternates between step 1 and 2 until convergence

This is an example of the more general **Expectation Maximization** (EM) algorithm. EM can be used for MLE/MAP in probabilistic models **that contain latent variables** making standard MLE/MAP hard

GMM Parameter Estimation

- Given $\mathbb{E}[z_{nk}] = \gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell)}$, the expected complete data log-lik.

GMM Parameter Estimation

- Given $\mathbb{E}[z_{nk}] = \gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell)}$, the expected complete data log-lik.

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} [\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

GMM Parameter Estimation

- Given $\mathbb{E}[z_{nk}] = \gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell)}$, the expected complete data log-lik.

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} [\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Taking derivatives w.r.t. $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, $\forall k = 1, \ldots, K$

GMM Parameter Estimation

- Given $\mathbb{E}[z_{nk}] = \gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)}$, the expected complete data log-lik.

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} [\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Taking derivatives w.r.t. $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, $\forall k = 1, \ldots, K$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = \frac{\partial}{\partial \boldsymbol{\mu}_k} \sum_{n=1}^{N} \gamma_{nk} \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = 0 \qquad \text{(note: constants w.r.t. } \mu_k \text{ can be ignored)}$$

GMM Parameter Estimation

- Given $\mathbb{E}[z_{nk}] = \gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^K \pi_\ell \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell)}$, the expected complete data log-lik.

$$\mathcal{L} = \sum_{n=1}^N \sum_{k=1}^K \gamma_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Taking derivatives w.r.t. $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, $\forall k = 1, \ldots, K$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = \frac{\partial}{\partial \boldsymbol{\mu}_k} \sum_{n=1}^N \gamma_{nk} \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = 0 \qquad \text{(note: constants w.r.t. } \mu_k \text{ can be ignored)}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k} = \frac{\partial}{\partial \boldsymbol{\Sigma}_k} \sum_{n=1}^N \gamma_{nk} \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = 0 \qquad \text{(note: constants w.r.t. } \boldsymbol{\Sigma}_k \text{ can be ignored)}$$

GMM Parameter Estimation

- Given $\mathbb{E}[z_{nk}] = \gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell)}$, the expected complete data log-lik.

$$\mathcal{L} = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Taking derivatives w.r.t. $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, $\forall k = 1, \ldots, K$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = \frac{\partial}{\partial \boldsymbol{\mu}_k} \sum_{n=1}^{N} \gamma_{nk} \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = 0 \qquad \text{(note: constants w.r.t. } \mu_k \text{ can be ignored)}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k} = \frac{\partial}{\partial \boldsymbol{\Sigma}_k} \sum_{n=1}^{N} \gamma_{nk} \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = 0 \qquad \text{(note: constants w.r.t. } \boldsymbol{\Sigma}_k \text{ can be ignored)}$$

- For each k, it's a "weighted" version of the MLE for the multivar. Gaussian $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

GMM Parameter Estimation

- Given $\mathbb{E}[z_{nk}] = \gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^{K}\pi_\ell \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell)}$, the expected complete data log-lik.

$$\mathcal{L} = \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)]$$

- Taking derivatives w.r.t. $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, $\forall k = 1,\ldots,K$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = \frac{\partial}{\partial \boldsymbol{\mu}_k}\sum_{n=1}^{N}\gamma_{nk}\log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) = 0 \qquad \text{(note: constants w.r.t. } \mu_k \text{ can be ignored)}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k} = \frac{\partial}{\partial \boldsymbol{\Sigma}_k}\sum_{n=1}^{N}\gamma_{nk}\log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) = 0 \qquad \text{(note: constants w.r.t. } \boldsymbol{\Sigma}_k \text{ can be ignored)}$$

- For each k, it's a "weighted" version of the MLE for the multivar. Gaussian $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$
- Can also solve for π_k likewise (subject to contraint $\sum_{k=1}^{K}\pi_k = 1$)

GMM Parameter Estimation

- Given $\mathbb{E}[z_{nk}] = \gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{\ell=1}^K \pi_\ell \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell)}$, the expected complete data log-lik.

$$\mathcal{L} = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)]$$

- Taking derivatives w.r.t. $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, $\forall k = 1,\ldots,K$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = \frac{\partial}{\partial \boldsymbol{\mu}_k}\sum_{n=1}^{N} \gamma_{nk} \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) = 0 \qquad \text{(note: constants w.r.t. } \mu_k \text{ can be ignored)}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k} = \frac{\partial}{\partial \boldsymbol{\Sigma}_k}\sum_{n=1}^{N} \gamma_{nk} \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) = 0 \qquad \text{(note: constants w.r.t. } \boldsymbol{\Sigma}_k \text{ can be ignored)}$$

- For each k, it's a "weighted" version of the MLE for the multivar. Gaussian $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$
- Can also solve for π_k likewise (subject to contraint $\sum_{k=1}^K \pi_k = 1$)
- Derivations are a bit tedious (but straightforward). I will provide a note.

GMM Parameter Estimation

- Suppose $N_k = \sum_{n=1}^{N} \gamma_{nk}$

GMM Parameter Estimation

- Suppose $N_k = \sum_{n=1}^{N} \gamma_{nk}$ is the effective number of observations assigned to Gaussian k

GMM Parameter Estimation

- Suppose $N_k = \sum_{n=1}^{N} \gamma_{nk}$ is the effective number of observations assigned to Gaussian k
  - Note that each $\boldsymbol{x}_n$ gets "soft" assignment $\gamma_{nk} \in (0, 1)$ to the k-th Gaussian

GMM Parameter Estimation

- Suppose $N_k = \sum_{n=1}^{N} \gamma_{nk}$ is the <u>effective number of observations</u> assigned to Gaussian k
  - Note that each $\boldsymbol{x}_n$ gets "soft" assignment $\gamma_{nk} \in (0,1)$ to the k-th Gaussian (also $\sum_{k=1}^{K} \gamma_{nk} = 1$)

GMM Parameter Estimation

- Suppose $N_k = \sum_{n=1}^{N} \gamma_{nk}$ is the effective number of observations assigned to Gaussian k
  - Note that each $\boldsymbol{x}_n$ gets "soft" assignment $\gamma_{nk} \in (0, 1)$ to the k-th Gaussian (also $\sum_{k=1}^{K} \gamma_{nk} = 1$)
- The final expressions for updates of $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$

GMM Parameter Estimation

- Suppose $N_k = \sum_{n=1}^{N} \gamma_{nk}$ is the effective number of observations assigned to Gaussian k
  - Note that each $\boldsymbol{x}_n$ gets "soft" assignment $\gamma_{nk} \in (0,1)$ to the k-th Gaussian (also $\sum_{k=1}^{K} \gamma_{nk} = 1$)
- The final expressions for updates of $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x}_n$$

GMM Parameter Estimation

- Suppose $N_k = \sum_{n=1}^{N} \gamma_{nk}$ is the effective number of observations assigned to Gaussian k
  - Note that each $\boldsymbol{x}_n$ gets "soft" assignment $\gamma_{nk} \in (0,1)$ to the k-th Gaussian (also $\sum_{k=1}^{K} \gamma_{nk} = 1$)
- The final expressions for updates of $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$

$$
\begin{aligned}
\boldsymbol{\mu}_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x}_n \\
\boldsymbol{\Sigma}_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top
\end{aligned}
$$

GMM Parameter Estimation

- Suppose $N_k = \sum_{n=1}^{N} \gamma_{nk}$ is the effective number of observations assigned to Gaussian k
  - Note that each $\boldsymbol{x}_n$ gets "soft" assignment $\gamma_{nk} \in (0,1)$ to the k-th Gaussian (also $\sum_{k=1}^{K} \gamma_{nk} = 1$)
- The final expressions for updates of $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$

$$\begin{aligned}
\boldsymbol{\mu}_k &= \frac{1}{N_k}\sum_{n=1}^{N} \gamma_{nk}\boldsymbol{x}_n \\
\boldsymbol{\Sigma}_k &= \frac{1}{N_k}\sum_{n=1}^{N} \gamma_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \\
\pi_k &= \frac{N_k}{N}
\end{aligned}$$

GMM Parameter Estimation

- Suppose $N_k = \sum_{n=1}^{N} \gamma_{nk}$ is the effective number of observations assigned to Gaussian k
  - Note that each $\boldsymbol{x}_n$ gets "soft" assignment $\gamma_{nk} \in (0,1)$ to the k-th Gaussian (also $\sum_{k=1}^{K} \gamma_{nk} = 1$)
- The final expressions for updates of $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$

$$
\begin{aligned}
\boldsymbol{\mu}_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x}_n \\
\boldsymbol{\Sigma}_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \\
\pi_k &= \frac{N_k}{N}
\end{aligned}
$$

- Note that μ_k, Σ_k updates are similar to multivar. Gaussian MLE equations

GMM Parameter Estimation

- Suppose $N_k = \sum_{n=1}^{N} \gamma_{nk}$ is the effective number of observations assigned to Gaussian k
  - Note that each $\boldsymbol{x}_n$ gets "soft" assignment $\gamma_{nk} \in (0,1)$ to the k-th Gaussian (also $\sum_{k=1}^{K} \gamma_{nk} = 1$)
- The final expressions for updates of $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$

$$
\begin{aligned}
\boldsymbol{\mu}_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x}_n \\
\boldsymbol{\Sigma}_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \\
\pi_k &= \frac{N_k}{N}
\end{aligned}
$$

- Note that μ_k, Σ_k updates are similar to multivar. Gaussian MLE equations
- Each $\boldsymbol{x}_n, n = 1, \ldots, N$ contributes to each $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ update but fractionally (based on γ_{nk})

Summary: EM for Learning GMM

- Initialize the parameters $\Theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ randomly, or using K-means

Summary: EM for Learning GMM

- Initialize the parameters $\Theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ randomly, or using K-means
- Iterate until convergence (e.g., when $\log p(\boldsymbol{x}|\Theta)$ ceases to increase or Θ doesn't change by much)

Summary: EM for Learning GMM

- Initialize the parameters $\Theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ randomly, or using K-means
- Iterate until convergence (e.g., when $\log p(\boldsymbol{x}|\Theta)$ ceases to increase or Θ doesn't change by much)
    - Given Θ, compute each expectation z_{nk} (posterior probability of $z_{nk} = 1$), $\forall n, k$

Summary: EM for Learning GMM

- Initialize the parameters $\Theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ randomly, or using K-means
- Iterate until convergence (e.g., when $\log p(\boldsymbol{x}|\Theta)$ ceases to increase or Θ doesn't change by much)
  - Given Θ, compute each expectation z_{nk} (posterior probability of $z_{nk} = 1$), $\forall n, k$

$$\gamma_{nk} = \mathbb{E}[z_{nk}] \propto \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad \text{(and re-normalize s.t. } \sum_{k=1}^K \gamma_{nk} = 1\text{)}$$

Summary: EM for Learning GMM

- Initialize the parameters $\Theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ randomly, or using K-means
- Iterate until convergence (e.g., when $\log p(\boldsymbol{x}|\Theta)$ ceases to increase or Θ doesn't change by much)
  - Given Θ, compute each expectation z_{nk} (posterior probability of $z_{nk} = 1$), $\forall n, k$

$$\gamma_{nk} = \mathbb{E}[z_{nk}] \propto \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad \text{(and re-normalize s.t. } \sum_{k=1}^{K} \gamma_{nk} = 1\text{)}$$

  - Given "responsibilities" $\gamma_{nk} = \mathbb{E}[z_{nk}]$, and $N_k = \sum_{n=1}^N \gamma_{nk}$, update $\Theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ as

Summary: EM for Learning GMM

- Initialize the parameters $\Theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ randomly, or using K-means
- Iterate until convergence (e.g., when $\log p(\boldsymbol{x}|\Theta)$ ceases to increase or Θ doesn't change by much)
  - Given Θ, compute each expectation z_{nk} (posterior probability of $z_{nk} = 1$), $\forall n, k$

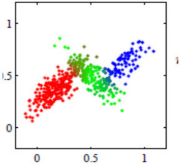
$$\gamma_{nk} = \mathbb{E}[z_{nk}] \propto \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad \text{(and re-normalize s.t. } \sum_{k=1}^K \gamma_{nk} = 1\text{)}$$

  - Given "responsibilities" $\gamma_{nk} = \mathbb{E}[z_{nk}]$, and $N_k = \sum_{n=1}^N \gamma_{nk}$, update $\Theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ as

$$\begin{aligned}
\boldsymbol{\mu}_k &= \frac{1}{N_k}\sum_{n=1}^N \gamma_{nk}\boldsymbol{x}_n \\
\boldsymbol{\Sigma}_k &= \frac{1}{N_k}\sum_{n=1}^N \gamma_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \\
\pi_k &= \frac{N_k}{N}
\end{aligned}$$

GMM: Some Important Aspects

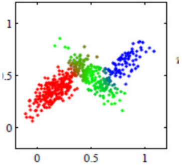
- GMM learns a probabilitic ("soft") clustering as opposed to hard clustering (e.g., K-means)



Soft clustering learned
by a Gaussian mixture model

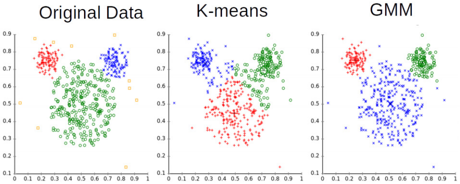
GMM: Some Important Aspects

- GMM learns a probabilitic ("soft") clustering as opposed to hard clustering (e.g., K-means)



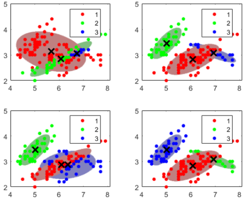
Soft clustering learned by a Gaussian mixture model

- GMM doesn't assume the clusters to be spherical and equi-sized as opposed to K-means (recall that each Gaussian has a specific covariance which can control the shape of that cluster)



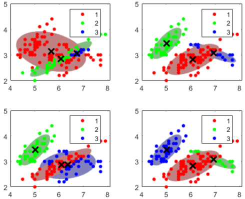
GMM: Some Important Aspects

- GMM, just like K-means, can be sensitive to initialization (EM only converges to a local optima)



GMM: Some Important Aspects

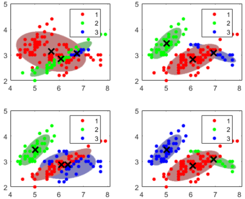
- GMM, just like K-means, can be sensitive to initialization (EM only converges to a local optima)



- Learning full covariances can be tricky: $\mathcal{O}(D^2)$ params to learn and can overfit.

GMM: Some Important Aspects

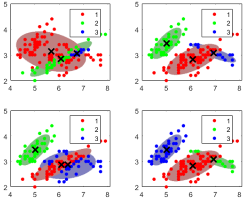
- GMM, just like K-means, can be sensitive to initialization (EM only converges to a local optima)



- Learning full covariances can be tricky: $\mathcal{O}(D^2)$ params to learn and can overfit. Some remedies

GMM: Some Important Aspects

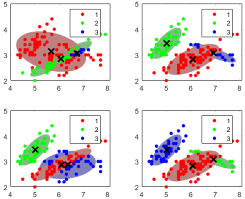
- GMM, just like K-means, can be sensitive to initialization (EM only converges to a local optima)



- Learning full covariances can be tricky: $\mathcal{O}(D^2)$ params to learn and can overfit. Some remedies
  - Assume the covariance matrix to be diagonal or spherical (but results might be poorer)

GMM: Some Important Aspects

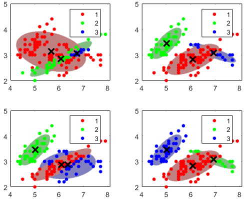
- GMM, just like K-means, can be sensitive to initialization (EM only converges to a local optima)



- Learning full covariances can be tricky: $\mathcal{O}(D^2)$ params to learn and can overfit. Some remedies
  - Assume the covariance matrix to be diagonal or spherical (but results might be poorer)
  - Use priors on the parameters and do MAP estimation for parameters in the M step of EM

GMM: Some Important Aspects

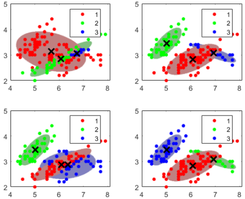
- GMM, just like K-means, can be sensitive to initialization (EM only converges to a local optima)



- Learning full covariances can be tricky: $\mathcal{O}(D^2)$ params to learn and can overfit. Some remedies
  - Assume the covariance matrix to be diagonal or spherical (but results might be poorer)
  - Use priors on the parameters and do MAP estimation for parameters in the M step of EM
  - Can do fully Bayesian inference for GMM

GMM: Some Important Aspects

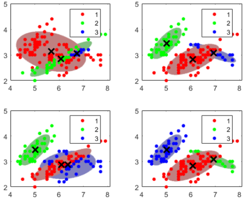
- GMM, just like K-means, can be sensitive to initialization (EM only converges to a local optima)



- Learning full covariances can be tricky: $\mathcal{O}(D^2)$ params to learn and can overfit. Some remedies
  - Assume the covariance matrix to be diagonal or spherical (but results might be poorer)
  - Use priors on the parameters and do MAP estimation for parameters in the M step of EM
  - Can do fully Bayesian inference for GMM
    - Helps learn K using marginal likelihood or nonparametric Bayesian methods

GMM: Some Important Aspects

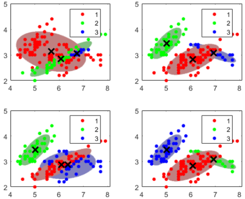
- GMM, just like K-means, can be sensitive to initialization (EM only converges to a local optima)



- Learning full covariances can be tricky: $\mathcal{O}(D^2)$ params to learn and can overfit. Some remedies
  - Assume the covariance matrix to be diagonal or spherical (but results might be poorer)
  - Use priors on the parameters and do MAP estimation for parameters in the M step of EM
  - Can do fully Bayesian inference for GMM
    - Helps learn K using marginal likelihood or nonparametric Bayesian methods
    - Helps learn the hyperparameters

GMM: Some Important Aspects

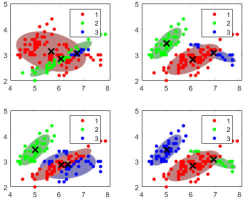
- GMM, just like K-means, can be sensitive to initialization (EM only converges to a local optima)



- Learning full covariances can be tricky: $\mathcal{O}(D^2)$ params to learn and can overfit. Some remedies
  - Assume the covariance matrix to be diagonal or spherical (but results might be poorer)
  - Use priors on the parameters and do MAP estimation for parameters in the M step of EM
  - Can do fully Bayesian inference for GMM
    - Helps learn K using marginal likelihood or nonparametric Bayesian methods
    - Helps learn the hyperparameters
    - More on these aspects later..

GMM: Some Important Aspects

- GMM, just like K-means, can be sensitive to initialization (EM only converges to a local optima)



- Learning full covariances can be tricky: $\mathcal{O}(D^2)$ params to learn and can overfit. Some remedies
  - Assume the covariance matrix to be diagonal or spherical (but results might be poorer)
  - Use priors on the parameters and do MAP estimation for parameters in the M step of EM
  - Can do fully Bayesian inference for GMM
    - Helps learn K using marginal likelihood or nonparametric Bayesian methods
    - Helps learn the hyperparameters
    - More on these aspects later..
  - Use low-rank Gaussians for each mixture component (mixture of factor analyzers)

Mixture Models: Applications beyond Clustering

- Mixture models is a general framework for modeling grouped data

Mixture Models: Applications beyond Clustering

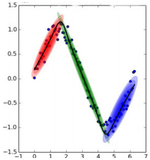
- Mixture models is a general framework for modeling grouped data
- Not limited only to clustering or density estimation

Mixture Models: Applications beyond Clustering

- Mixture models is a general framework for modeling grouped data
- Not limited only to clustering or density estimation
- Mixture of Experts: Each "expert" is a probabilistic supervised learning model $p(y|\boldsymbol{x}, \theta_k)$
  - Overall model is a convex combination of the experts

$$p(y|\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x}) p(y|\boldsymbol{x}, \theta_k)$$

  - Enables learning rich models (e.g., nonlinear reg.) from simpler models (e.g., linear reg.)



Next Class: The General EM Algorithm