



UNIVERSIDADE  
FEDERAL DO CEARÁ

DEPARTAMENTO  
DE COMPUTAÇÃO

# Aprendizagem de Máquina Probabilística

César Lincoln Cavalcante Mattos

2024

# Agenda

## ① Modelos de misturas

## ② Algoritmo Expectation-Maximization

Algoritmo EM para Mistura de Gaussianas (GMMs)

Algoritmo EM para GMMs com estimação MAP

Seleção de modelos para GMM

Algoritmo EM para Mistura de Especialistas

Algoritmo EM como um limiar inferior

Variantes do algoritmo EM

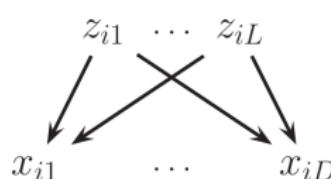
Algoritmo EM para dados faltantes

## ③ Tópicos adicionais

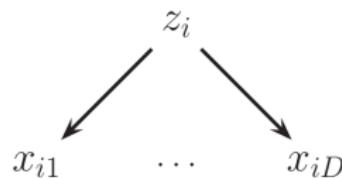
## ④ Referências

# Modelos de variáveis latentes

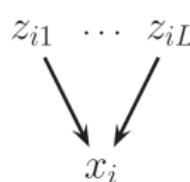
- Uma maneira de modelar variáveis correlacionadas é através de **modelos de variáveis latentes** (*latent variable models*, LVM).
- Considera que as observações foram geradas por “causas” ocultas comuns.
- Pode ser visto como uma maneira de obter um “gargalo” (*bottleneck*) que representa os dados de maneira comprimida.



(a)



(b)



(c)



(d)

# Modelos de misturas

- Considere um modelo gerativo com variáveis latentes discretas:

$$z \sim p(z), \text{ em que } z \in \{1, \dots, K\},$$
$$\mathbf{x} \sim p(\mathbf{x}|z).$$

- Escolhemos uma priori  $p(z) = \text{Cat}(\boldsymbol{\pi})$ , ou seja,  $p(z = k) = \pi_k$ .
- A verossimilhança  $p(\mathbf{x}|z = k)$  relaciona a variável latente  $z$  e a observação  $\mathbf{x}$ .
- Um modelo de mistura pode ser obtido ao marginalizar  $z$ :

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^K p(\mathbf{x}|z = k)p(z = k)$$
$$= \sum_{k=1}^K \pi_k p(\mathbf{x}|z = k),$$

em que  $0 \leq \pi_k \leq 1$ ,  $\sum_{k=1}^K \pi_k = 1$  e  $\boldsymbol{\theta}$  são parâmetros do modelo.

# Modelos de misturas para agrupamento

- Ao usar um modelo de misturas nas observações  $\mathbf{x}_i|_{i=1}^N$  disponíveis, podemos encontrar qual componente (ou grupo/*cluster*) a gerou calculando a posteriori  $p(z_i = k|\mathbf{x}_i, \boldsymbol{\theta})$ .
- No chamado **soft clustering**, computamos a **responsabilidade**  $r_{ik}$  do grupo  $k$  via regra de Bayes:

$$r_{ik} \triangleq p(z_i = k|\mathbf{x}_i, \boldsymbol{\theta}) = \frac{p(z_i = k|\boldsymbol{\theta})p(\mathbf{x}_i|z_i = k, \boldsymbol{\theta})}{\sum_{k'=1}^K p(z_i = k'|\boldsymbol{\theta})p(\mathbf{x}_i|z_i = k', \boldsymbol{\theta})}.$$

- Note que a diferença em relação a um classificador generativo é que  $z_i$  não é observado.
- No caso do chamado **hard clustering**, temos:

$$\hat{z}_i = \arg \max_k r_{ik} = \arg \max_k [\log p(z_i = k|\boldsymbol{\theta}) + \log p(\mathbf{x}_i|z_i = k, \boldsymbol{\theta})].$$

# Mistura de especialistas

- Modelos de misturas também podem ser usados no contexto de modelos discriminantes para regressão ou classificação.
- Uma **mistura de especialistas** (*mixture of experts*, MoE) é obtida a partir de submodelos (especialistas) em certas regiões do espaço de entrada.
- Para  $K$  modelos de regressão linear, temos:

$$p(y_i | \mathbf{x}_i, z_i = k, \boldsymbol{\theta}) = \mathcal{N}(y_i | \mathbf{w}_k^\top \mathbf{x}_i, \sigma_k^2), \quad 1 \leq k \leq K,$$

$$p(z_i | \mathbf{x}_i, \boldsymbol{\theta}) = \text{Cat}(z_i, \mathcal{S}(\mathbf{V}^\top \mathbf{x}_i)),$$

$$p(y_i | \mathbf{x}_i, \boldsymbol{\theta}) = \sum_{k=1}^K p(y_i | \mathbf{x}_i, z_i = k, \boldsymbol{\theta}) p(z_i = k | \mathbf{x}_i, \boldsymbol{\theta}),$$

em que  $p(z_i | \mathbf{x}_i, \boldsymbol{\theta})$  age como uma *gating function* e  $\mathcal{S}(\cdot)$  é uma função softmax parametrizada por  $\mathbf{V}$ .

# Agenda

① Modelos de misturas

② Algoritmo Expectation-Maximization

Algoritmo EM para Mistura de Gaussianas (GMMs)

Algoritmo EM para GMMs com estimação MAP

Seleção de modelos para GMM

Algoritmo EM para Mistura de Especialistas

Algoritmo EM como um limiar inferior

Variantes do algoritmo EM

Algoritmo EM para dados faltantes

③ Tópicos adicionais

④ Referências

# Mistura de Gaussianas

- O modelo de mistura de Gaussianas (*Gaussian mixture model*, GMM) é um dos mais usados para representar dados contínuos:

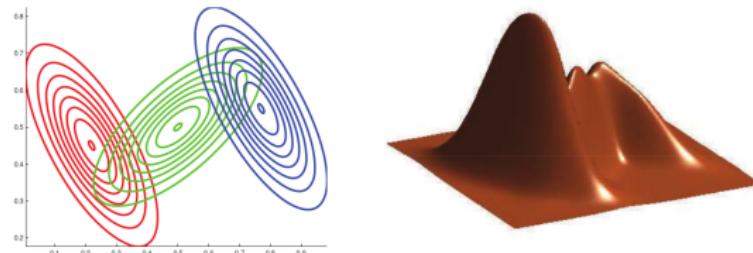
$$p(\mathbf{x}_i | \boldsymbol{\theta}) = \sum_{k=1}^K p(z_i = k) \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

- Alternativamente, como  $z_i$  possui uma distribuição categórica, podemos representá-la por um vetor  $\mathbf{z}_i \in \{0, 1\}^K$  em que  $\sum_k z_{ik} = 1$ :

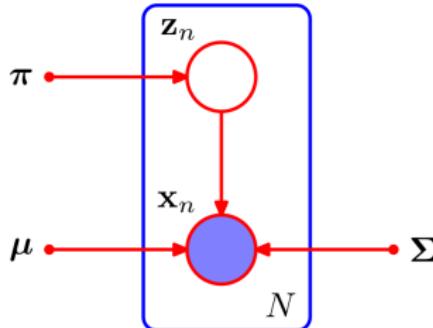
$$\begin{aligned} p(\mathbf{x}_i | \boldsymbol{\theta}) &= \sum_{k=1}^K p(z_{ik} = 1) \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \end{aligned}$$

# Mistura de Gaussianas

- Um GMM com componentes  $D$ -dimensionais suficientes pode aproximar qualquer distribuição contínua em  $\mathbb{R}^D$ .



- As variáveis do modelo e suas relações podem ser representadas por um **modelo gráfico probabilístico**:



# Algoritmo Expectation-Maximization

- ML ou MAP não é aplicável diretamente à GMM (ou qualquer outro LVM), pois as variáveis latentes não são observadas.
- Em geral, a verossimilhança dos dados observados é dada por:

$$p(\mathbf{x}_i | \boldsymbol{\theta}) = \sum_{z_i} p(\mathbf{x}_i | z_i, \boldsymbol{\theta}) p(z_i | \boldsymbol{\theta}),$$

$$p(\mathbf{X} | \boldsymbol{\theta}) = \prod_{i=1}^N \sum_{z_i} p(\mathbf{x}_i | z_i, \boldsymbol{\theta}) p(z_i | \boldsymbol{\theta}),$$

$$\mathcal{L}(\boldsymbol{\theta}) = \log p(\mathbf{X} | \boldsymbol{\theta}) = \sum_{i=1}^N \log \left[ \sum_{z_i} p(\mathbf{x}_i, z_i | \boldsymbol{\theta}) \right],$$

em que todos os parâmetros foram reunidos em  $\boldsymbol{\theta}$ .

- **Problema:** as variáveis latentes não são observadas e  $\mathcal{L}(\boldsymbol{\theta})$  não pode ser calculada.

# Algoritmo Expectation-Maximization

- **Ideia:** Considerar que os valores  $z_i$  são conhecidos e definir a verossimilhança dos dados completos (conjunta):

$$\mathcal{L}_c(\boldsymbol{\theta}) \triangleq \sum_{i=1}^N \log p(\mathbf{x}_i, z_i | \boldsymbol{\theta}).$$

- **Ideia:** Computar  $\mathbb{E}[\mathcal{L}_c(\boldsymbol{\theta})]$  em relação à sua posteriori:

$$\begin{aligned}\mathbb{E}[\mathcal{L}_c(\boldsymbol{\theta})] &= \sum_{k=1}^K p(z_i = k | \mathbf{X}, \boldsymbol{\theta}) \mathcal{L}_c(\boldsymbol{\theta}) \\ &= \sum_{i=1}^N \sum_{k=1}^K p(z_i = k | \mathbf{X}, \boldsymbol{\theta}) \log p(\mathbf{x}_i, z_i | \boldsymbol{\theta}).\end{aligned}$$

- **Ideia:** Como buscamos uma solução de ML, otimizamos  $\mathbb{E}[\mathcal{L}_c(\boldsymbol{\theta})]$  em função de  $\boldsymbol{\theta}$ :

$$\boldsymbol{\theta}_{ML} = \arg \max \mathbb{E}[\mathcal{L}_c(\boldsymbol{\theta})].$$

# Algoritmo Expectation-Maximization

- **Problema:** Temos uma relação cruzada de dependência entre  $\mathbb{E}[\mathcal{L}_c(\theta)]$  e  $\theta$ .
- **Ideia:** Seguimos uma estratégia iterativa, resultando no algoritmo EM (**Expectation-Maximization**):
  - **Passo E:** O valor esperado na iteração atual  $t$  é dado por:

$$Q(\theta, \theta^{(t-1)}) = \mathbb{E}[\mathcal{L}_c(\theta) | \mathbf{X}, \theta^{(t-1)}],$$

em que explicitou-se a dependência dos dados de treino  $\mathbf{X}$  e dos parâmetros da iteração anterior  $\theta^{(t-1)}$ .

- **Passo M:** O valor atualizado dos parâmetros é obtido via otimização da função auxiliar  $Q(\cdot)$ :

$$\theta^{(t)} = \arg \max_{\theta} Q(\theta, \theta^{(t-1)}).$$

# Algoritmo EM para GMMs

- No caso de uma GMM com  $K$  componentes, temos:

$$\begin{aligned}\mathcal{L}_c(\boldsymbol{\theta}) &= \sum_{i=1}^N \log p(\mathbf{x}_i, z_i | \boldsymbol{\theta}) \\ &= \sum_{i=1}^N \log p(\mathbf{x}_i | z_i, \boldsymbol{\theta}) p(z_i | \boldsymbol{\theta}) \\ &= \sum_{i=1}^N \log \prod_{k=1}^K [\pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]^{\mathbb{I}(z_i=k)} \\ &= \sum_{i=1}^N \sum_{k=1}^K \mathbb{I}(z_i = k) [\log \pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)],\end{aligned}$$

em que o expoente  $\mathbb{I}(z_i = k)$  foi incluído por considerarmos que  $\mathbf{x}_i$  foi observado da  $k$ -ésima componente da mistura.

# Algoritmo EM para GMMs

- A função auxiliar é obtida tomando a esperança com relação a  $z_i$ :

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)}) &= \mathbb{E} \left[ \sum_{i=1}^N \sum_{k=1}^K \mathbb{I}(z_i = k) [\log \pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] \right] \\ &= \sum_{i=1}^N \sum_{k=1}^K \mathbb{E}[\mathbb{I}(z_i = k)][\log \pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] \\ &= \sum_{i=1}^N \sum_{k=1}^K \underbrace{p(z_i = k | \mathbf{x}_i, \boldsymbol{\theta}^{(t-1)})}_{r_{ik}} [\log \pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] \\ &= \sum_{i=1}^N \sum_{k=1}^K r_{ik} \log \pi_k + \sum_{i=1}^N \sum_{k=1}^K r_{ik} \log \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \end{aligned}$$

em que  $r_{ik}$  é o **coeficiente de responsabilidade** da componente  $k$  pela observação  $i$ .

# Algoritmo EM para GMMs

- Portanto, o **passo E** consiste em computar os coeficientes de responsabilidade  $r_{ik}$ :

$$\begin{aligned} r_{ik} \triangleq p(z_i = k | \mathbf{x}_i, \boldsymbol{\theta}^{t-1}) &= \frac{p(\mathbf{x}_i | z_i = k, \boldsymbol{\theta}^{t-1}) p(z_i = k)}{\sum_{k'=1}^K p(\mathbf{x}_i | z_i = k', \boldsymbol{\theta}^{t-1}) p(z_i = k')} \\ &= \frac{\pi_k^{(t-1)} \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k^{(t-1)}, \boldsymbol{\Sigma}_k^{(t-1)})}{\sum_{k'=1}^K \pi_{k'}^{(t-1)} \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_{k'}^{(t-1)}, \boldsymbol{\Sigma}_{k'}^{(t-1)})}. \end{aligned}$$

# Algoritmo EM para GMMs

- No **passo M**, otimizamos  $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)})$  com relação a  $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ .
- Para  $\pi_k$ , como  $\sum_k \pi_k = 1$ , derivamos o Lagrangiano abaixo:

$$\begin{aligned}\frac{\partial}{\partial \pi_k} \left[ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)}) + \lambda \left( 1 - \sum_{k=1}^K \pi_k \right) \right] &= 0 \\ \frac{\partial}{\partial \pi_k} \left[ \sum_{i=1}^N \sum_{k=1}^K r_{ik} \log \pi_k + \lambda \left( 1 - \sum_{k=1}^K \pi_k \right) \right] &= 0 \\ \sum_{i=1}^N \left[ \frac{1}{\pi_k} r_{ik} - \lambda \right] &= 0 \\ \pi_k &= \frac{\sum_{i=1}^N r_{ik}}{N\lambda}.\end{aligned}$$

- Como  $\sum_k \pi_k = 1$ , temos  $\lambda = \frac{\sum_k \sum_i r_{ik}}{N} = 1$  e  $\pi_k = \frac{\sum_{i=1}^N r_{ik}}{N}$ .

# Algoritmo EM para GMMs

- Precisamos ainda otimizar  $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)})$  em relação a  $\boldsymbol{\mu}_k$  e  $\boldsymbol{\Sigma}_k$ :

$$Q(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k^{(t-1)}, \boldsymbol{\Sigma}_k^{(t-1)}) = \sum_{i=1}^N r_{ik} \log \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$
$$\propto \sum_{i=1}^N r_{ik} \left[ -\frac{1}{2} \log |\boldsymbol{\Sigma}_k| - \frac{1}{2} (\mathbf{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k) \right].$$

- A última expressão corresponde à log-verossimilhança de uma Gaussiana em que cada termo é ponderado por  $r_{ik}$ , logo:

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

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

# Algoritmo EM para GMMs

## Resumo do algoritmo

- ① Escolha  $K, \pi_k^{(0)}, \boldsymbol{\mu}_k^{(0)}, \boldsymbol{\Sigma}_k^{(0)}, \forall k$ .
- ② Faça  $t = 1$  e repita até convergir:

- Passo E:

$$r_{ik} \triangleq p(z_i = k | \mathbf{x}_i, \boldsymbol{\theta}^{(t-1)}) = \frac{\pi_k^{(t-1)} \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k^{(t-1)}, \boldsymbol{\Sigma}_k^{(t-1)})}{\sum_{k'=1}^K \pi_{k'}^{(t-1)} \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_{k'}^{(t-1)}, \boldsymbol{\Sigma}_{k'}^{(t-1)})}.$$

- Passo M:

$$\pi_k^{(t)} = \frac{\sum_{i=1}^N r_{ik}}{N},$$

$$\boldsymbol{\mu}_k^{(t)} = \frac{\sum_{i=1}^N r_{ik} \mathbf{x}_i}{\sum_{i=1}^N r_{ik}}, \quad \boldsymbol{\Sigma}_k^{(t)} = \frac{\sum_{i=1}^N r_{ik} \mathbf{x}_i \mathbf{x}_i^\top}{\sum_{i=1}^N r_{ik}} - \boldsymbol{\mu}_k^{(t)} (\boldsymbol{\mu}_k^{(t)})^\top.$$

# Algoritmo K-médias

- Considera-se que  $\pi_k = \frac{1}{K}$ ,  $\Sigma_k = \sigma^2 \mathbf{I}$ ,  $\forall k$ , são fixos.
- Considera-se ainda que  $r_{ik} \triangleq p(z_i = k | \mathbf{x}_i, \boldsymbol{\theta}^{(t-1)}) \approx \mathbb{I}(k = \hat{z}_i)$ , em que  $\hat{z}_i = \arg \max_k p(z_i = k | \mathbf{x}_i, \boldsymbol{\theta}^{(t-1)})$

## Resumo do algoritmo

- ① Escolha  $K$ ,  $\mu_k^{(0)}$ ,  $\forall k$ .
- ② Faça  $t = 1$  e repita até convergir:
  - Atribuição dos padrões:

$$\hat{z}_i = \arg \max_k p(z_i = k | \mathbf{x}_i, \boldsymbol{\mu}_k^{(t-1)}) = \arg \min_k \left\| \mathbf{x}_i - \boldsymbol{\mu}_k^{(t-1)} \right\|^2.$$

- Atualização dos centróides:

$$\boldsymbol{\mu}_k^{(t)} = \frac{\sum_i \mathbb{I}(\hat{z}_i = k) \mathbf{x}_i}{\sum_i \mathbb{I}(\hat{z}_i = k)} = \frac{1}{N_k} \sum_{i: \hat{z}_i = k} \mathbf{x}_i.$$

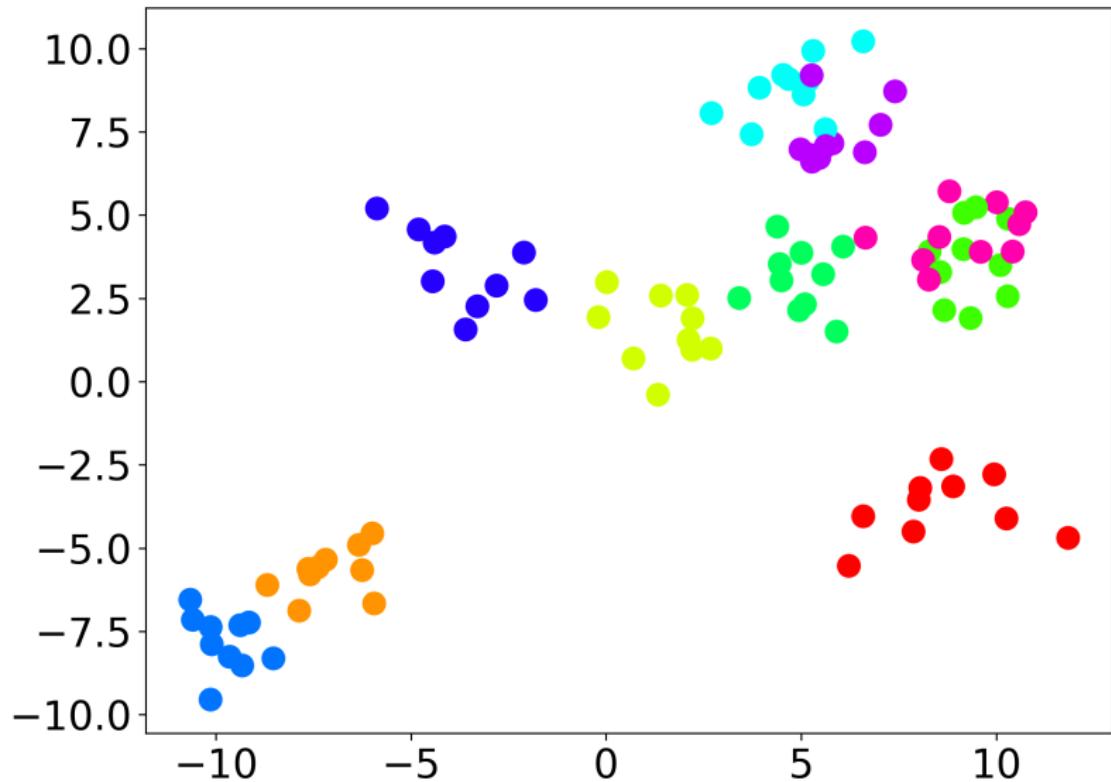
# Algoritmo K-médias

- Podemos executar o algoritmo K-médias com diferentes inicializações  $\mu_k^{(0)}$  para os centróides e escolher aquela com menor **erro de reconstrução**:

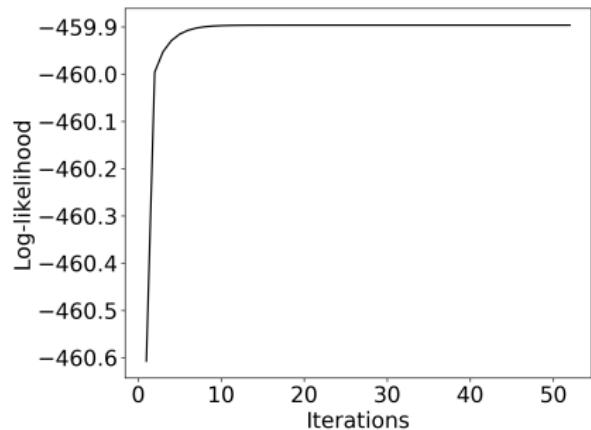
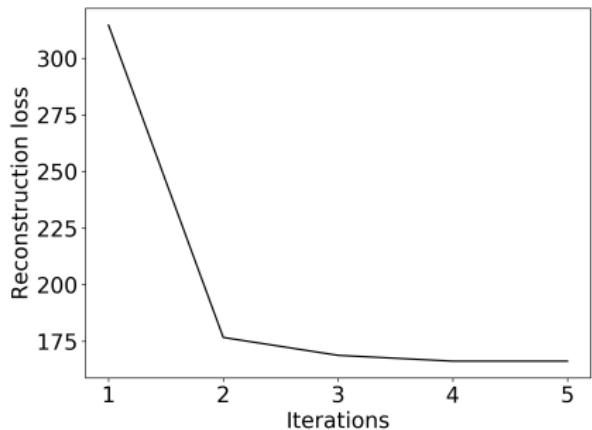
$$E_{\text{rec.}} = \sum_{k=1}^K \sum_{i:\hat{z}_i=k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$$

- O K-médias++ é uma heurística eficiente para a inicialização:
  - ① Escolha o primeiro centróide aleatoriamente entre os dados.
  - ② Repita  $K - 1$  vezes: escolha o próxima centróide entre os dados com probabilidade proporcional à distância ao seu centróide mais próximo.
- Podemos usar o algoritmo K-médias para inicializar as componentes do GMM:
  - Os vetores de média são iniciados com os centróides.
  - As matrizes de covariância são iniciadas com as matrizes de covariância estimadas dos grupos encontrados.

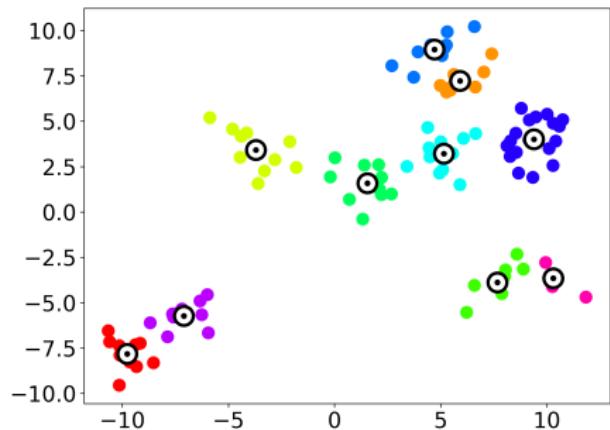
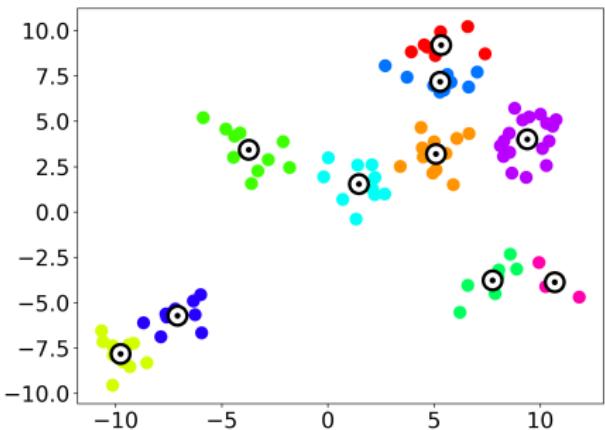
# K-médias e GMM



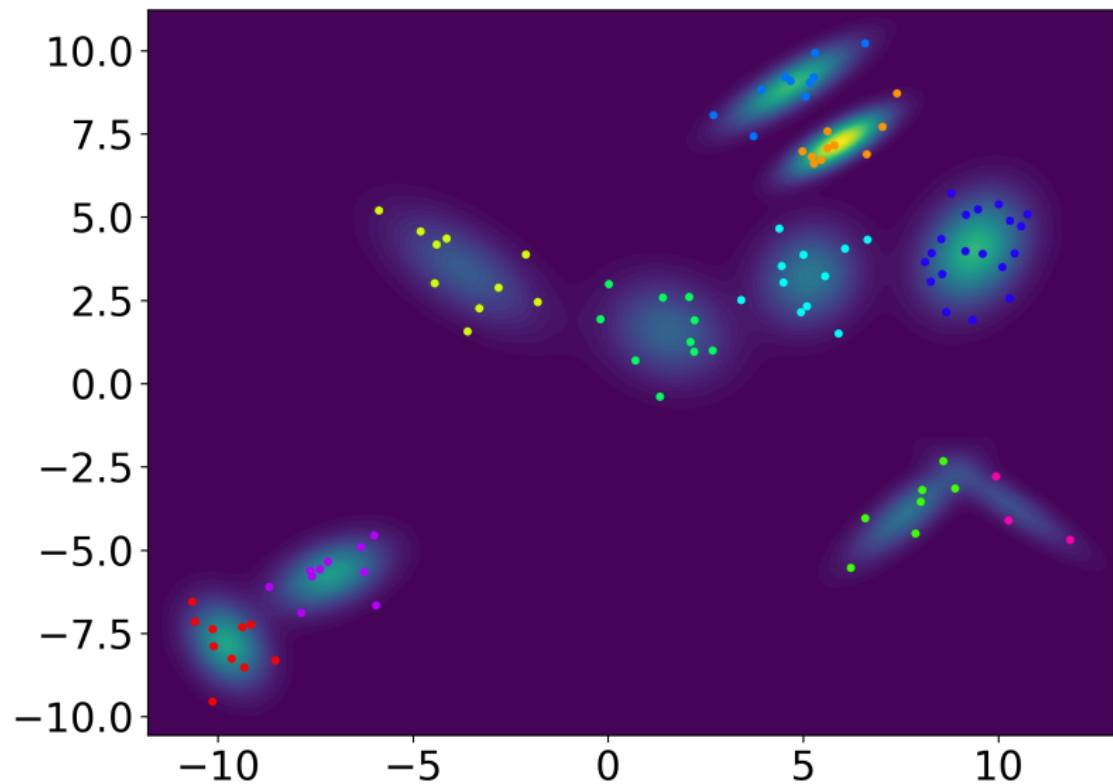
# K-médias (esquerda) e GMM (direita)



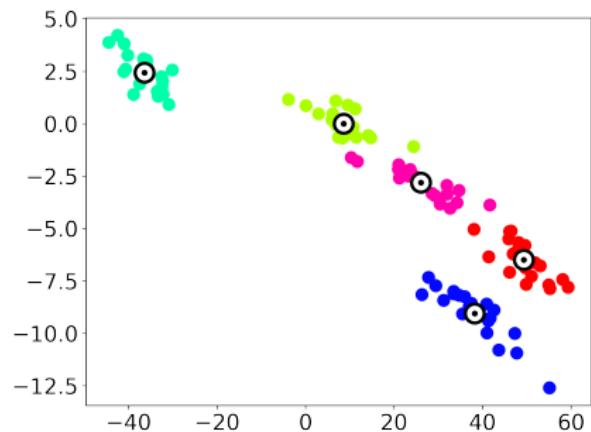
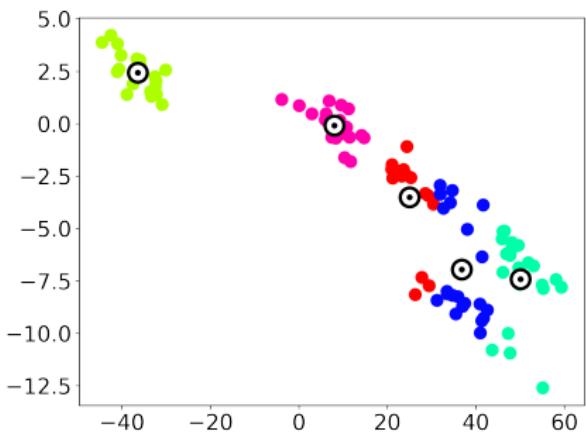
# K-médias (esquerda) e GMM (direita)



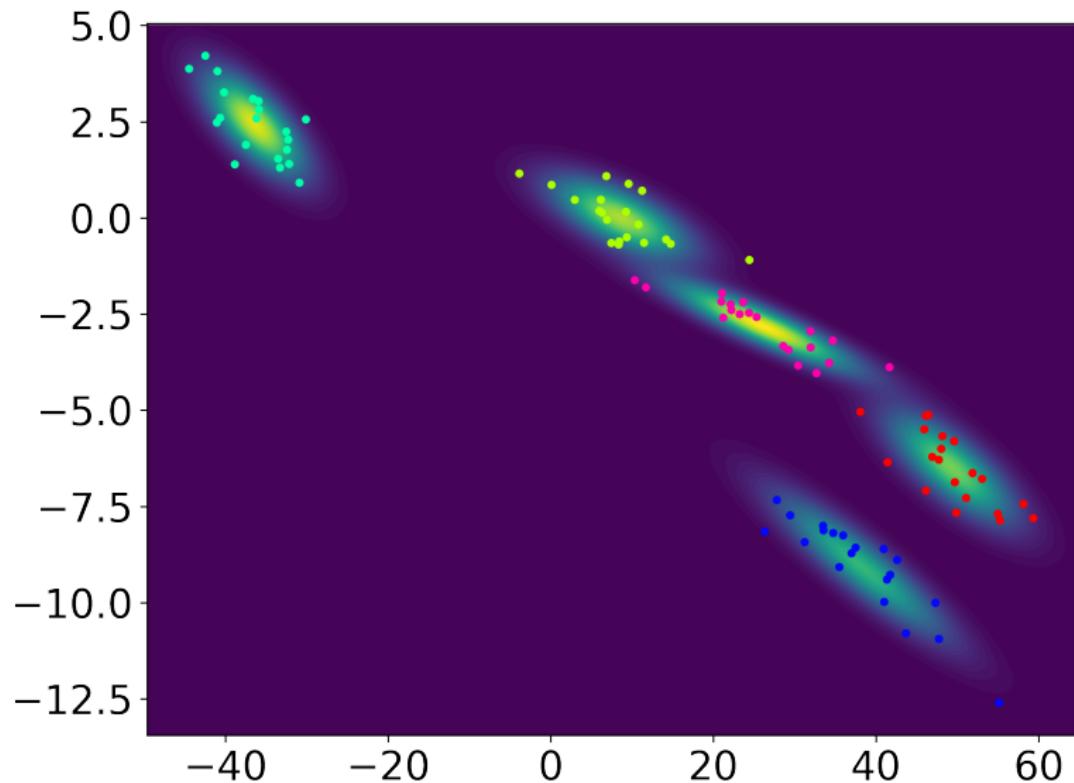
# GMM



# K-médias (esquerda) e GMM (direita)



# GMM



# Algoritmo EM para GMMs com solução MAP

- Podemos obter uma solução MAP com o algoritmo EM considerando prioris para os parâmetros do modelo.
- A função auxiliar seria dada por:

$$Q'(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)}) = \sum_{i=1}^N \sum_{k=1}^K r_{ik} \log \pi_k + \sum_{i=1}^N \sum_{k=1}^K r_{ik} \log \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ + \underbrace{\log p(\boldsymbol{\pi}) + \sum_{k=1}^K \log p(\boldsymbol{\theta}_k)}_{\text{termos das prioris}}.$$

- Menor chance de ocorrer singularidades na otimização, em que uma Gaussiana é colocada exatamente sobre uma observação.
- Para as probabilidades  $\boldsymbol{\pi}$  das componentes da mistura, a distribuição de Dirichlet é conjugada:

$$p(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi} | \boldsymbol{\alpha}).$$

# Algoritmo EM para GMMs com solução MAP

- Para as médias  $\mu_k$  e as covariâncias  $\Sigma_k$ , escolhemos a priori conjugada em que média e covariância são dependentes:

$$\begin{aligned} p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) &= p(\boldsymbol{\mu}_k | \boldsymbol{\Sigma}_k)p(\boldsymbol{\Sigma}_k) \\ &= \text{NIW}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k | \mathbf{m}_0, \kappa_0, \nu_0, \mathbf{S}_0) \\ &\triangleq \mathcal{N}(\boldsymbol{\mu}_k | \boldsymbol{\mu}_0, \frac{1}{\kappa_0} \boldsymbol{\Sigma}_k) \text{IW}(\boldsymbol{\Sigma}_k | \mathbf{S}_0, \nu_0), \end{aligned}$$

onde definimos a distribuição Normal-inverse-Wishart (NIW).

- Escolhas usuais para os hiperparâmetros a partir dos dados são  $\mathbf{S}_0 = \frac{1}{K^{2/D}} \mathbf{S}_{\bar{x}}$ ,  $\nu_0 = D + 2$ ,  $\boldsymbol{\mu}_0 = \bar{\mathbf{x}}$ ,  $\kappa_0 = 0.01$  (ou  $\kappa_0 \rightarrow 0$ ), em que  $\bar{\mathbf{x}}$  e  $\mathbf{S}_{\bar{x}}$  são a média e a covariância amostral.
- Somente o passo M do algoritmo EM será modificado!

# Algoritmo EM para GMMs com solução MAP

## Resumo do algoritmo

- ① Escolha  $K, \alpha_k, \boldsymbol{\mu}_0, \kappa_0, \nu_0, \mathbf{S}_0, \pi_k^{(0)}, \boldsymbol{\mu}_k^{(0)}, \Sigma_k^{(0)}, \forall k$ .
- ② Faça  $t = 1$  e repita até convergir:

- Passo E:

$$r_{ik} \triangleq p(z_i = k | \mathbf{x}_i, \boldsymbol{\theta}^{(t-1)}) = \frac{\pi_k^{(t-1)} \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k^{(t-1)}, \boldsymbol{\Sigma}_k^{(t-1)})}{\sum_{k'=1}^K \pi_{k'}^{(t-1)} \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_{k'}^{(t-1)}, \boldsymbol{\Sigma}_{k'}^{(t-1)})}.$$

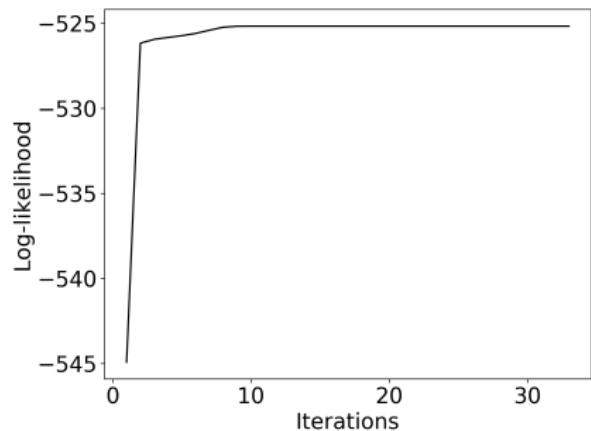
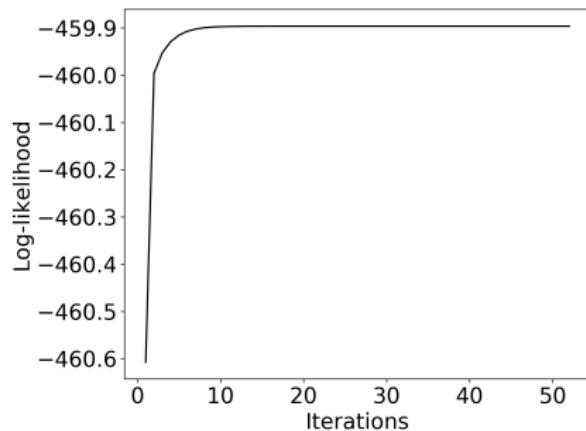
- Passo M:

$$\pi_k^{(t)} = \frac{\alpha_k - 1 + \sum_i r_{ik}}{N - K + \sum_k \alpha_k},$$

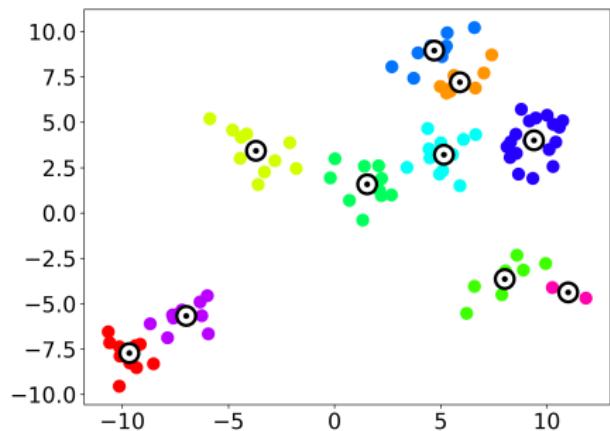
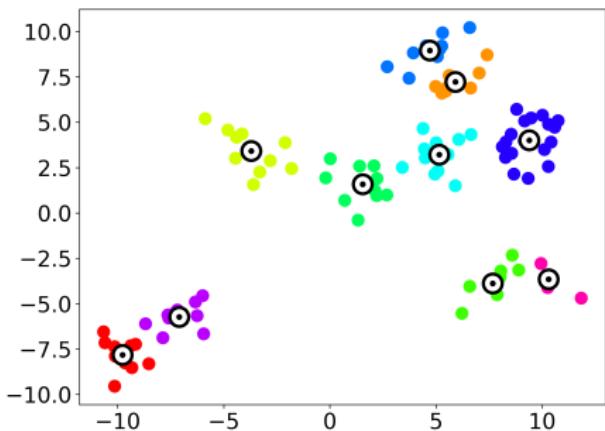
$$\bar{\mathbf{x}}_k \triangleq \frac{\sum_i r_{ik} \mathbf{x}_i}{\sum_i r_{ik}}, \quad \boldsymbol{\mu}_k^{(t)} = \frac{\kappa_0 \mathbf{m}_0 + \bar{\mathbf{x}}_k \sum_i r_{ik}}{\kappa_0 + \sum_i r_{ik}},$$

$$\boldsymbol{\Sigma}_k^{(t)} = \frac{\mathbf{S}_0 + \sum_i r_{ik} (\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^\top + \frac{\kappa_0 \sum_i r_{ik}}{\kappa_0 + \sum_i r_{ik}} (\bar{\mathbf{x}}_k - \mathbf{m}_0)(\bar{\mathbf{x}}_k - \mathbf{m}_0)^\top}{\nu_0 + D + 2 + \sum_i r_{ik}}.$$

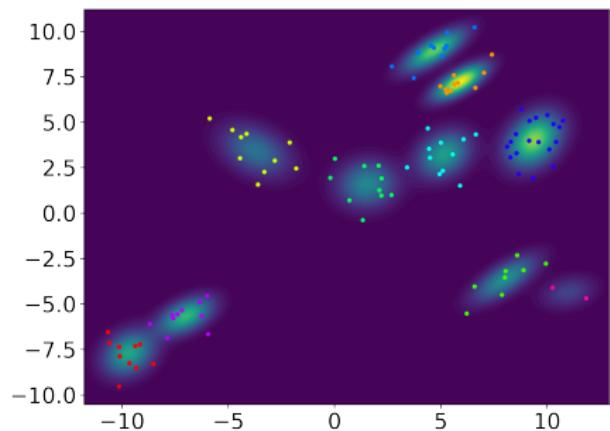
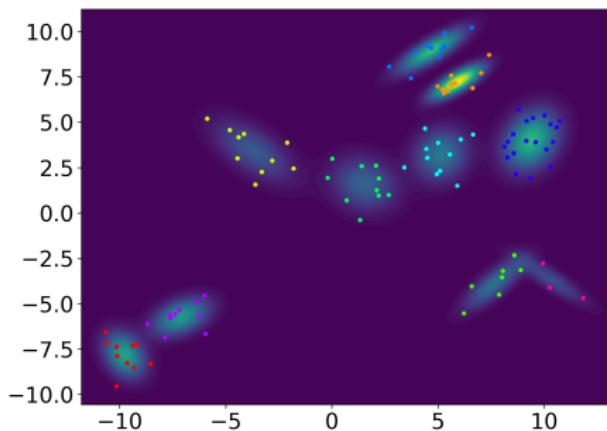
# GMM-ML (esquerda) e GMM-MAP (direita)



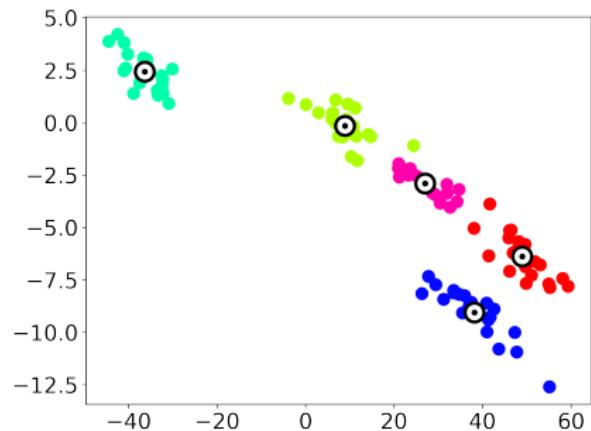
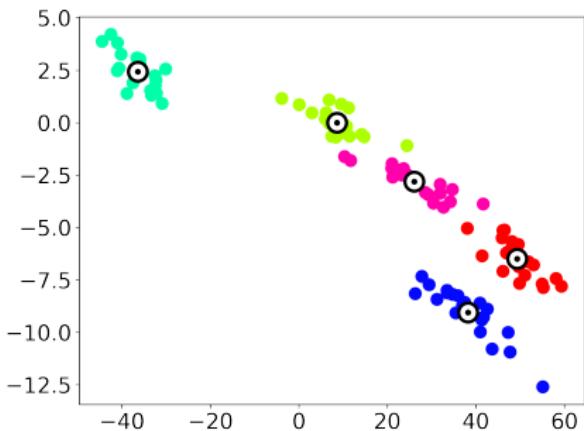
# GMM-ML (esquerda) e GMM-MAP (direita)



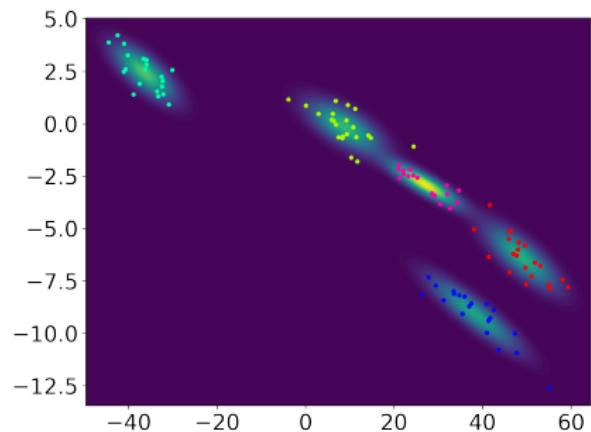
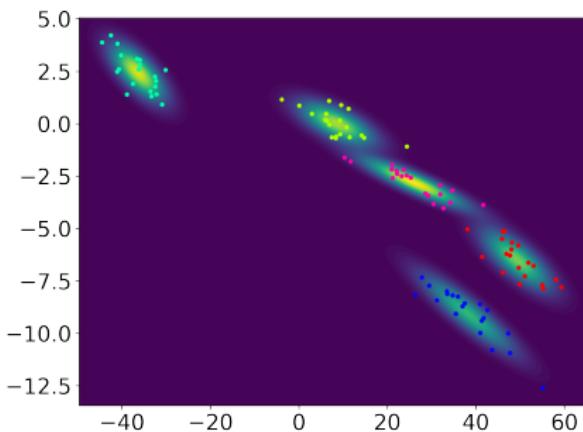
# GMM-ML (esquerda) e GMM-MAP (direita)



# GMM-ML (esquerda) e GMM-MAP (direita)



# GMM-ML (esquerda) e GMM-MAP (direita)



# Seleção de modelos para GMM

- Além dos parâmetros do GMM ou outro modelo de mistura, precisamos escolher o número  $K$  de componentes.
- Duas abordagens práticas:
  - Calcular o **BIC (Bayesian Information Criterion)** ou o **AIC (Akaike Information Criterion)** para diferentes valores de  $K$ :

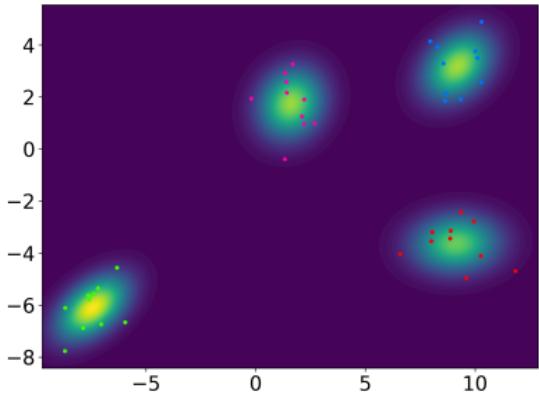
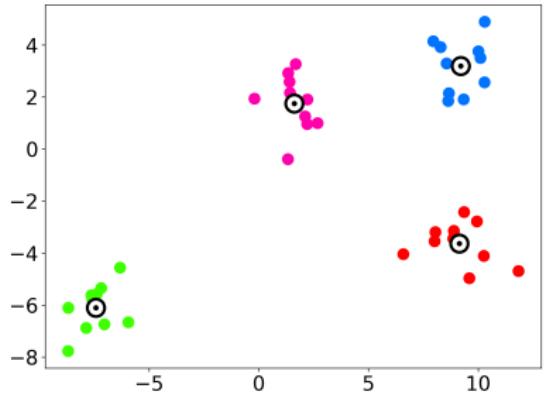
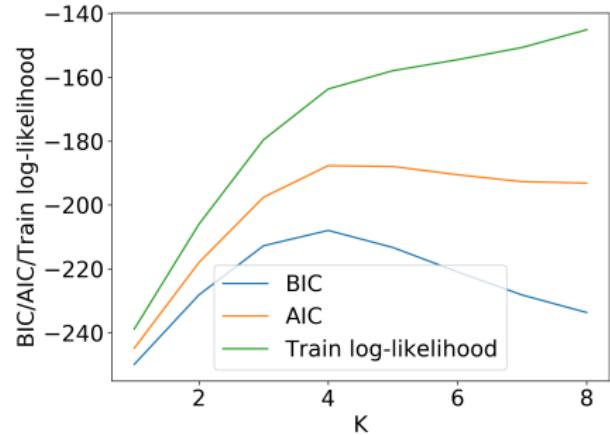
$$\text{BIC}(\mathcal{D}|K) \triangleq \log p(\mathcal{D}|\hat{\theta}) - \frac{1}{2}M \log N,$$

$$\text{AIC}(\mathcal{D}|K) \triangleq \log p(\mathcal{D}|\hat{\theta}) - M,$$

em que  $M$  é o número de parâmetros e  $N$  é o total de observações.

- Calcular a log-verossimilhança  $\log p(\mathcal{D}|\hat{\theta})$  em **dados não usados no treinamento** para diferentes valores de  $K$ .

# Seleção de modelos via AIC/BIC



# Algoritmo EM para Mistura de Especialistas

- No caso de uma mistura de especialistas representados por modelos lineares, temos a seguinte log-verossimilhança para os dados completos:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)}) = \sum_{i=1}^N \sum_{k=1}^K r_{ik} \log[\pi_{ik} \mathcal{N}(y_i | \mathbf{w}_k^\top \mathbf{x}_i, \sigma_k^2)],$$
$$\pi_{ik} \triangleq \mathcal{S}(\mathbf{V}^\top \mathbf{x}_i)_k,$$
$$r_{ik} \propto \pi_{ik}^{(t-1)} \mathcal{N}(y_i | \mathbf{x}_i^\top \mathbf{w}_k^{(t-1)}, (\sigma_k^{(t-1)})^2).$$

- O passo E (cálculo dos  $r_{ik}$ ) continua o mesmo, apenas trocando  $\pi_k$  por  $\pi_{ik}$ .

# Algoritmo EM para Mistura de Especialistas

- O passo M com relação aos parâmetros  $\boldsymbol{w}_k$  e  $\sigma_k^2$  é dado por:

$$\begin{aligned} Q(\boldsymbol{\theta}_k, \boldsymbol{\theta}^{(t-1)}) &= \sum_{i=1}^N r_{ik} \log \mathcal{N}(y_i | \boldsymbol{w}_k^\top \boldsymbol{x}_i, \sigma_k^2) \\ &= \sum_{i=1}^N r_{ik} \left\{ -\frac{1}{\sigma_k^2} (y_i - \boldsymbol{w}_k^\top \boldsymbol{x}_i)^2 \right\} \end{aligned}$$

- A expressão acima corresponde a um problema de mínimos quadrados ponderado, como no IRLS, tendo solução na forma:

$$\begin{aligned} \boldsymbol{w}_k &= (\boldsymbol{X}^\top \boldsymbol{R}_k \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{R}_k \boldsymbol{y}, \\ \boldsymbol{R}_k &= \text{diag}(r_{1k}, r_{2k}, \dots, r_{Nk}). \end{aligned}$$

- Para a variância, temos:

$$\sigma_k^2 = \frac{\sum_{i=1}^N r_{ik} (y_i - \boldsymbol{w}_k^\top \boldsymbol{x}_i)^2}{\sum_{i=1}^N r_{ik}}$$

# Algoritmo EM para Mistura de Especialistas

- No caso do parâmetro  $\mathbf{V}$  da função softmax, temos a seguinte componente:

$$Q(\mathbf{V}, \boldsymbol{\theta}^{(t-1)}) = \sum_{i=1}^N \sum_{k=1}^K r_{ik} \log \pi_{ik}.$$

- Como  $\pi_{ik} = \mathcal{S}(\mathbf{V}^\top \mathbf{x}_i)_k$ , a expressão acima corresponde à função custo de um modelo de regressão logística multiclasse.
- A única diferença de uma regressão softmax é o uso de "soft-labels"  $r_{ik}$  em vez de saídas 1-hot-encoding.

# Algoritmo EM para Mistura de Especialistas

## Resumo do algoritmo

- ① Escolha  $K$ ,  $\mathbf{V}^{(0)}$ ,  $\mathbf{w}_k^{(0)}$ ,  $(\sigma_k^2)^{(0)}$ ,  $\forall k$ .
- ② Faça  $t = 1$  e repita até convergir:

- Passo E:

$$\pi_{ik}^{(t-1)} = \mathcal{S}((\mathbf{V}^{(t-1)})^\top \mathbf{x}_i)_k,$$
$$r_{ik} = \frac{\pi_{ik}^{(t-1)} \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k^{(t-1)}, \boldsymbol{\Sigma}_k^{(t-1)})}{\sum_{k'=1}^K \pi_{ik'}^{(t-1)} \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_{k'}^{(t-1)}, \boldsymbol{\Sigma}_{k'}^{(t-1)})}.$$

- Passo M:

$$\mathbf{w}_k^{(t)} = (\mathbf{X}^\top \mathbf{R}_k \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{R}_k \mathbf{y}, \quad \mathbf{R}_k = \text{diag}(r_{1k}, \dots, r_{Nk}),$$
$$(\sigma_k^2)^{(t)} = \frac{\sum_i r_{ik} (y_i - (\mathbf{w}_k^{(t)})^\top \mathbf{x}_i)^2}{\sum_i r_{ik}}, \quad \mathbf{V}^{(t)} \leftarrow \text{"soft" softmax.}$$

## Algoritmo EM como um limiar inferior

- Reescrevemos a log-verossimilhança dos dados observados de um LVM com variáveis latentes  $\mathbf{z}_i$  discretas (como uma mistura):

$$\mathcal{L}(\boldsymbol{\theta}) \triangleq \sum_{i=1}^N \log p(\mathbf{x}_i | \boldsymbol{\theta}) = \sum_{i=1}^N \log \left[ \sum_{\mathbf{z}_i} p(\mathbf{x}_i, \mathbf{z}_i | \boldsymbol{\theta}) \right].$$

- Considere uma distribuição arbitrária  $q(\mathbf{z}_i)$ :

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^N \log \left[ \sum_{\mathbf{z}_i} q(\mathbf{z}_i) \frac{p(\mathbf{x}_i, \mathbf{z}_i | \boldsymbol{\theta})}{q(\mathbf{z}_i)} \right].$$

- **Problema:** A expressão acima é difícil de trabalhar por causa do log fora do somatório.

## Algoritmo EM como um limiar inferior

- Ideia: Usamos a **desigualdade de Jensen**, definida por:

$$f \left( \sum_{i=1}^N \lambda_i \mathbf{x}_i \right) \leq \sum_{i=1}^N \lambda_i f(\mathbf{x}_i),$$

para uma função  $f(\cdot)$  convexa qualquer.

- A função  $\log(\cdot)$  é côncava, então podemos usar o inverso da desigualdade de Jensen em  $\mathcal{L}(\boldsymbol{\theta})$ :

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \sum_{i=1}^N \log \left[ \sum_{\mathbf{z}_i} q(\mathbf{z}_i) \frac{p(\mathbf{x}_i, \mathbf{z}_i | \boldsymbol{\theta})}{q(\mathbf{z}_i)} \right] \\ &\geq \sum_i \sum_{\mathbf{z}_i} q(\mathbf{z}_i) \log \frac{p(\mathbf{x}_i, \mathbf{z}_i | \boldsymbol{\theta})}{q(\mathbf{z}_i)} \\ &\geq \sum_i \mathbb{E}_{q(\mathbf{z}_i)} [\log p(\mathbf{x}_i, \mathbf{z}_i | \boldsymbol{\theta})] + \mathbb{H}[q(\mathbf{z}_i)] \triangleq \mathcal{L}(\boldsymbol{\theta}, q),\end{aligned}$$

em que  $\mathbb{H}[q(\mathbf{z}_i)] = -\mathbb{E}_{q(\mathbf{z}_i)} [\log q(\mathbf{z}_i)]$  é um termo de entropia.

## Algoritmo EM como um limiar inferior

- A escolha da distribuição  $q$  (agora um “parâmetro”) deve maximizar o limiar inferior (lower bound) obtido:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, q) &= \sum_i \sum_{\mathbf{z}_i} q(\mathbf{z}_i) \log \frac{p(\mathbf{x}_i, \mathbf{z}_i | \boldsymbol{\theta})}{q(\mathbf{z}_i)} \\ &= \sum_i \sum_{\mathbf{z}_i} q(\mathbf{z}_i) \log \frac{p(\mathbf{z}_i | \mathbf{x}_i, \boldsymbol{\theta}) p(\mathbf{x}_i | \boldsymbol{\theta})}{q(\mathbf{z}_i)} \\ &= \sum_i \sum_{\mathbf{z}_i} q(\mathbf{z}_i) \log \frac{p(\mathbf{z}_i | \mathbf{x}_i, \boldsymbol{\theta})}{q(\mathbf{z}_i)} + \sum_i \sum_{\mathbf{z}_i} q(\mathbf{z}_i) \log p(\mathbf{x}_i | \boldsymbol{\theta}) \\ &= - \sum_i \text{KL}(q(\mathbf{z}_i) \| p(\mathbf{z}_i | \mathbf{x}_i, \boldsymbol{\theta})) + \underbrace{\sum_i \log p(\mathbf{x}_i | \boldsymbol{\theta})}_{\mathcal{L}(\boldsymbol{\theta})},\end{aligned}$$

em que usamos a definição da divergência de Kullback-Leibler:

$$\text{KL}(q(\mathbf{z}) \| p(\mathbf{z})) = \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z})}.$$

## Algoritmo EM como um limiar inferior

- A divergência de KL é sempre não-negativa e somente igual a zero para distribuições idênticas.
- Teremos  $\mathcal{L}(\boldsymbol{\theta}, q) = \mathcal{L}(\boldsymbol{\theta})$  para  $q(\mathbf{z}_i) = p(\mathbf{z}_i | \mathbf{x}_i, \boldsymbol{\theta})$ .
- Como os parâmetros  $\boldsymbol{\theta}$  são desconhecidos, usamos sua estimativa  $\boldsymbol{\theta}^{(t-1)}$  até a iteração anterior.
- Assim, fazemos  $q^{(t)}(\mathbf{z}_i) = p(\mathbf{z}_i | \mathbf{x}_i, \boldsymbol{\theta}^{(t-1)})$ , que é o resultado do **passo E** do algoritmo EM:

$$\mathcal{L}(\boldsymbol{\theta}, q^{(t)}) = \sum_i \mathbb{E}_{q^{(t)}(\mathbf{z}_i)} [\log p(\mathbf{x}_i, \mathbf{z}_i | \boldsymbol{\theta})] + \mathbb{H}[q^{(t)}(\mathbf{z}_i)].$$

- O primeiro termo de  $\mathcal{L}(\boldsymbol{\theta}, q^{(t)})$  é a log-verossimilhança esperada dos dados completos.
- O segundo termo independe de  $\boldsymbol{\theta}$ , logo o **passo M** torna-se:

$$\boldsymbol{\theta}^{(t)} = \arg \max_{\boldsymbol{\theta}} \sum_i \mathbb{E}_{q^{(t)}(\mathbf{z}_i)} [\log p(\mathbf{x}_i, \mathbf{z}_i | \boldsymbol{\theta})].$$

## Algoritmo EM como um limiar inferior

- Como usamos  $q^{(t)}(\mathbf{z}_i) = p(\mathbf{z}_i | \mathbf{x}_i, \boldsymbol{\theta}^{(t-1)})$ , o termo KL some:

$$\mathcal{L}(\boldsymbol{\theta}^{(t-1)}, q^{(t)}) = - \sum_i \underbrace{\text{KL}(q^{(t)}(\mathbf{z}_i) \| p(\mathbf{z}_i | \mathbf{x}_i, \boldsymbol{\theta}^{(t-1)}))}_{0} + \mathcal{L}(\boldsymbol{\theta}^{(t-1)}),$$

$$\mathcal{L}(\boldsymbol{\theta}^{(t-1)}, q^{(t)}) = \mathcal{L}(\boldsymbol{\theta}^{(t-1)}).$$

- A cada passo E do algoritmo EM o limiar inferior “toca”  $\mathcal{L}(\boldsymbol{\theta}^{(t-1)})$ , então maximizá-lo equivale a maximizar  $\mathcal{L}(\boldsymbol{\theta}^{(t-1)})$ .
- Cada atualização do algoritmo EM monotonicamente incrementa a log-verossimilhança dos dados observados:

$$\mathcal{L}(\boldsymbol{\theta}^{(t)}) \geq \mathcal{L}(\boldsymbol{\theta}^{(t)}, q^{(t)}) \geq \mathcal{L}(\boldsymbol{\theta}^{(t-1)}, q^{(t)}) = \mathcal{L}(\boldsymbol{\theta}^{(t-1)}).$$

# Variantes do algoritmo EM

- Os passos E e M do algoritmo EM são dados por:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)}) = \mathbb{E}[\mathcal{L}_c(\boldsymbol{\theta}) | \mathcal{D}, \boldsymbol{\theta}^{(t-1)}] = \sum_i \mathbb{E}_{q^{(t)}(\mathbf{z}_i)} [\log p(\mathbf{x}_i, \mathbf{z}_i | \boldsymbol{\theta})],$$

$$\boldsymbol{\theta}^{(t)} = \arg \max_{\boldsymbol{\theta}} [Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)}) + \mathbb{H}[q^{(t)}(\mathbf{z}_i)]],$$

em que o termo  $\mathbb{H}[q^{(t)}(\mathbf{z}_i)]$  pode ser ignorado no algoritmo original por não depender de  $\boldsymbol{\theta}$ .

- Note que para obter o melhor limiar inferior, tivemos que fazer  $q^{(t)}(\mathbf{z}_i) = p(\mathbf{z}_i | \mathbf{x}_i, \boldsymbol{\theta}^{(t-1)})$ .
- No entanto, a distribuição  $p(\mathbf{z}_i | \mathbf{x}_i, \boldsymbol{\theta}^{(t-1)})$  pode não ser tratável/analítica e requerer inferência aproximada no passo E.
- Além disso, o passo M pode não admitir solução analítica.

# Variantes do algoritmo EM

- **EM variacional:** Escolhemos uma distribuição  $q^{(t)}(\mathbf{z}_i)$  parametrizada e a otimizamos em conjunto com os outros parâmetros do modelo.
- **Monte Carlo EM (MCEM):** Geramos amostras via aproximação de Monte Carlo (como Markov Chain Monte Carlo, MCMC) para computar a esperança em relação a  $p(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\theta}^{(t-1)})$ .
- **EM estocástico:** Semelhante ao MCEM, mas admitindo amostragens estocásticas de  $p(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\theta}^{(t-1)})$ .
- **EM generalizado:** Quando o passo M não é analítico, podemos fazer atualizações parciais nos parâmetros, por exemplo via gradiente ascendente.

# Algoritmo EM para dados faltantes

- Considere a situação em que os dados disponíveis  $\mathbf{X}$  possuem uma parte visível,  $\mathbf{X}_v = \{x_{id} : O_{id} = 1\}$ , e uma parte faltante,  $\mathbf{X}_h = \{x_{id} : O_{id} = 0\}$ , em que  $O_{id} = 1$  indica que a dimensão  $d$  do padrão  $i$  é observada.
- Considerando a hipótese “missing at random” (o dado faltante independe do seu valor, mas depende dos observados), temos:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \sum_{i=1}^N \log p(\mathbf{x}_{iv}, |\boldsymbol{\theta}) \\ &= \sum_{i=1}^N \log \left[ \sum_{\mathbf{x}_{ih}} p(\mathbf{x}_{iv}, \mathbf{x}_{ih} | \boldsymbol{\theta}) \right].\end{aligned}$$

- Como  $\mathcal{L}(\boldsymbol{\theta})$  é difícil de otimizar diretamente, podemos usar o algoritmo EM.

## Algoritmo EM para dados faltantes

- Considere a tarefa de estimar os parâmetros de uma Gaussiana multivariada com dados faltantes.
- A log-verossimilhança esperada dos dados completos será:

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)}) &= \mathbb{E} \left[ \sum_{i=1}^N \log \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}, \boldsymbol{\Sigma}) \right] \\ &\propto -\frac{N}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} \sum_i \mathbb{E}[(\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu})] \\ &\propto -\frac{N}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} \text{Tr}(\boldsymbol{\Sigma}^{-1} \sum_i \mathbb{E}[(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top]) \\ &\propto -\frac{N}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} \text{Tr}(\boldsymbol{\Sigma}^{-1} \mathbf{A}), \\ \mathbf{A} &\triangleq \sum_i (\mathbb{E}[\mathbf{x}_i \mathbf{x}_i^\top] - 2\boldsymbol{\mu}_i \mathbb{E}[\mathbf{x}_i]^\top + \boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top). \end{aligned}$$

- Para concluir o passo E, precisamos computar  $\mathbb{E}[\mathbf{x}_i]$  e  $\mathbb{E}[\mathbf{x}_i \mathbf{x}_i^\top]$ .

# Algoritmo EM para dados faltantes

- Pela propriedade de condicionamento da Gaussiana, temos:

$$\mathbf{x}_{ih} | \mathbf{x}_{iv}, \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{m}_i, \mathbf{S}_i),$$

$$\mathbf{m}_i = \boldsymbol{\mu}_h + \boldsymbol{\Sigma}_{hv} \boldsymbol{\Sigma}_v^{-1} (\mathbf{x}_{iv} - \boldsymbol{\mu}_v),$$

$$\mathbf{S}_i = \boldsymbol{\Sigma}_h - \boldsymbol{\Sigma}_{hv} \boldsymbol{\Sigma}_v^{-1} \boldsymbol{\Sigma}_{vh},$$

$$\text{em que } \boldsymbol{\mu}^{(t-1)} = [\boldsymbol{\mu}_v, \boldsymbol{\mu}_h]^\top \text{ e } \boldsymbol{\Sigma}^{(t-1)} = \begin{bmatrix} \boldsymbol{\Sigma}_v & \boldsymbol{\Sigma}_{vh} \\ \boldsymbol{\Sigma}_{hv} & \boldsymbol{\Sigma}_h \end{bmatrix}.$$

- Assim:

$$\mathbb{E}[\mathbf{x}_i] = [\mathbb{E}[\mathbf{x}_{ih}]; \mathbf{x}_{iv}] = [\mathbf{m}_i; \mathbf{x}_{iv}],$$

$$\begin{aligned} \mathbb{E}[\mathbf{x}_i \mathbf{x}_i^\top] &= \mathbb{E} \left[ \begin{bmatrix} \mathbf{x}_{ih} \\ \mathbf{x}_{iv} \end{bmatrix} [\mathbf{x}_{ih}^\top, \mathbf{x}_{iv}^\top] \right] \\ &= \begin{bmatrix} \mathbb{E}[\mathbf{x}_{ih} \mathbf{x}_{ih}^\top] & \mathbb{E}[\mathbf{x}_{ih}] \mathbf{x}_{iv}^\top \\ \mathbf{x}_{iv} \mathbb{E}[\mathbf{x}_{ih}]^\top & \mathbf{x}_{iv} \mathbf{x}_{iv}^\top \end{bmatrix} = \begin{bmatrix} \mathbf{S}_i + \mathbf{m}_i \mathbf{m}_i^\top & \mathbf{m}_i \mathbf{x}_{iv}^\top \\ \mathbf{x}_{iv} \mathbf{m}_i^\top & \mathbf{x}_{iv} \mathbf{x}_{iv}^\top \end{bmatrix}, \end{aligned}$$

$$\text{em que usamos } \mathbb{E}[\mathbf{x} \mathbf{x}^\top] = \text{cov}[\mathbf{x}] + \mathbb{E}[\mathbf{x}] \mathbb{E}[\mathbf{x}]^\top.$$

# Algoritmo EM para dados faltantes

- No passo M precisamos resolver a equação  $\nabla Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)}) = \mathbf{0}$ .
- Seguimos as regras de atualização usuais para a Gaussiana:

$$\boldsymbol{\mu}^{(t)} = \frac{1}{N} \sum_i^N \mathbb{E}[\mathbf{x}_i],$$
$$\boldsymbol{\Sigma}^{(t)} = \left( \frac{1}{N-1} \sum_i^N \mathbb{E}[\mathbf{x}_i \mathbf{x}_i^\top] \right) - \boldsymbol{\mu}^{(t)} (\boldsymbol{\mu}^{(t)})^\top.$$

- Note que poderíamos incluir prioris para os parâmetros e obter uma solução MAP.

# Agenda

① Modelos de misturas

② Algoritmo Expectation-Maximization

Algoritmo EM para Mistura de Gaussianas (GMMs)

Algoritmo EM para GMMs com estimação MAP

Seleção de modelos para GMM

Algoritmo EM para Mistura de Especialistas

Algoritmo EM como um limiar inferior

Variantes do algoritmo EM

Algoritmo EM para dados faltantes

③ Tópicos adicionais

④ Referências

# Tópicos adicionais

- Mistura de Bernoullis.
- Mistura de distribuições t de Student.
- Algoritmo EM incremental.
- Mistura hierárquica de especialistas.

# Agenda

① Modelos de misturas

② Algoritmo Expectation-Maximization

Algoritmo EM para Mistura de Gaussianas (GMMs)

Algoritmo EM para GMMs com estimação MAP

Seleção de modelos para GMM

Algoritmo EM para Mistura de Especialistas

Algoritmo EM como um limiar inferior

Variantes do algoritmo EM

Algoritmo EM para dados faltantes

③ Tópicos adicionais

④ Referências

# Referências bibliográficas

- **Cap. 11 - MURPHY, Kevin P. Machine learning: a probabilistic perspective, 2012.**
- **Cap. 9 - BISHOP, Christopher M. Pattern recognition and machine learning, 2006.**