Gaussian mixture models unit:

- Everything you need to know about probability (last week Friday)
- Lab activity: From histograms to Gaussians (Wednesday)
- Mixture models (**today**)
- Expectation-maximization (next week Monday)

Today and next time:

- The density estimation task
- Gaussian mixture models
- The general idea of Expectation-Maximization
- The algorithm

Lessons of this unit

- Gaussian models as representative example of density estimation.
- Mixture models
- Unsupervised learning in data with latent variables
- Expectation-maximization as an iterative algorithm in the absence of a closed-form solution

Gaussian model family:

$$p(x) = \mathcal{N}(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Multivariate Gaussian model family:

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} e^{\frac{-(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}{2}}$$

Gaussian mixture model family:

$$p(x) \text{ or } p(x, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{i=0}^{K-1} \pi_i \mathcal{N}(x \mid \mu_i, \sigma_i)$$

Given (scalar) observations $\boldsymbol{x}$ generated by a process suspected of being comprised of K subprocesses, each with a Gaussian distribution, train a model to predict the probability of observation value x, using the following model family:

$$p(x) \text{ or } p(x, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{i=0}^{K-1} \pi_i \mathcal{N}(x \mid \mu_i, \sigma_i)$$

where π_i is the probability of an observation having come from subprocess i and μ_i and σ_i are the mean and standard deviation, respectively, of subprocess i, and $\mathcal{N}$ is the probability density function for the Gaussian distribution,

$$p(x) = \mathcal{N}(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

That is, find $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\sigma}$ to maximize the likelihood of the training data under this model.

Coming up:

Due Fri, Feb 21:
Submit "Dataset" checkpoint for term project

Recommended sometime:
Read or skim chapter on GMM/EM from Deisenroth et al.
(See Canvas)

Due Tues, Feb 25:
Take GMM quiz

Due Fri, Feb 28:
Do GMM/EM programming assignment

Due Wed, Mar 5:
Read and respond to Urbina et al, "Dual use of AI-powered drug discovery"
(See Canvas)

Also coming sometime...
Textbook and supplemental reading about SVMs

Assume we have N data points, and let n index into the data.
In the **expectation** step, we calculate *responsibility* $r_{n,i}$, a measure of the ith Gaussian's contribution to the probability of datapoint x_n:

$$r_{n,i} = \frac{\pi_i \mathcal{N}(x_n \mid \mu_i, \sigma_i)}{\sum_{j=0}^{K-1} \pi_j \mathcal{N}(x_n \mid \mu_j, \sigma_j)}$$

In the **maximization step**, we recalculate $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ based on the expected values for $r_{n,i}$. N_i is the sum of the contributions of model i to all data points, a proxy for how many data points come from model i:

$$N_i = \sum_{n=0}^{N-1} r_{n,i}$$

Recalculating μ, σ, and π_i:

$$\mu_i^{\text{new}} = \frac{\sum_{n=0}^{N-1} r_{n,i} x_n}{N_i} \qquad \sigma_i^{\text{new}} = \sqrt{\frac{\sum_{n=0}^{N-1} r_{n,i} \left(x_n - \mu_i^{\text{new}}\right)^2}{N_i}} \qquad \pi_i^{\text{new}} = \frac{N_i}{N}$$

We measure the how well the current model fits the data by computing the *log likelihood*:

$$\sum_{n=0}^{N-1} \lg \sum_{i=0}^{K-1} \pi_i \mathcal{N}(x_n \mid \mu_i, \sigma_i)$$

For initial $\boldsymbol{\pi}$, assume all processes are equally likely,

$$\pi_i = \frac{1}{K}$$

For initial $\boldsymbol{\mu}$, either spread the means evenly throughout the training data range,

$$\mu_i = x_{min} + \frac{x_{max} - x_{min}}{k+1} \cdot (i+1)$$

or use random values from the training data range.
For initial $\boldsymbol{\sigma}$, use

$$\sigma_i = \frac{x_{max} - x_{min}}{k^2}$$

.