Review: Gaussian mixture model

Gaussian Mixture Models

A multivariate normal distribution or multivariate Gaussian distribution is a generalization of the one-dimensional Gaussian distribution into muliple dimensions.

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\}$$

The distribution is given by its mean, $\boldsymbol{\mu}$, and covariance, $\boldsymbol{\Sigma}$, matrices. To generate samples from the multivariate normal distribution under python, one could use the numpy.random.multivariate_normal function from numpy.

In statistics, a mixture model is a probabilistic model for density estimation using a mixture distribution. A mixture model can be regarded as a type of unsupervised learning or clustering [wikimixmodel]. Mixture models provide a method of describing more complex propability distributions, by combining several probability distributions. Mixture models can also be used to cluster data.

The Gaussian mixture distribution is given by the following equation [bishop2006]:

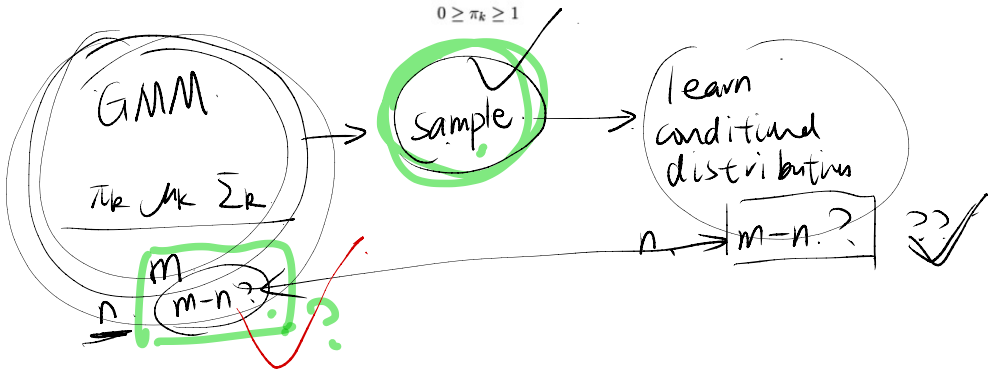
$$p(\mathbf{x}) = \sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$$

Here we have a linear mixture of Gaussian density functions, $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$. The parameters π_k are called mixing coefficients, which must fulfill

$$\sum_{k=1}^{K}\pi_k = 1$$

and given $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) \geq 0$ and $p(\mathbf{x}) \geq 0$ we also have that

$$0 \geq \pi_k \geq 1$$



How to sample from the GMM ?

Sample one point

1. 一共有 K 个 component, π_k

2. X 无论, 采自

$P(X=k) = \pi_k$ for any k. ✓

3. 从 $\mathcal{N}(\mu_k, \Sigma_k)$ 里取一点 ✓

X → random variable, r.v.

realization, realized value

Conditional Distribution of GMM.

The conditional and marginal distributions can be easily derived from the joint probability described by the GMM, see [bishop2006], [rasmussen2006], [sung2004]. Let $\mathbf{x}_A$ and $\mathbf{x}_B$ be jointly Gaussian vectors, and given a GMM with the Guassian distributions $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ with $\boldsymbol{\Lambda}_k = \boldsymbol{\Sigma}_k^{-1}$

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{pmatrix}, \quad \boldsymbol{\mu}_k = \begin{pmatrix} \boldsymbol{\mu}_{kA} \\ \boldsymbol{\mu}_{kB} \end{pmatrix}$$

$$\boldsymbol{\Sigma}_k = \left(\begin{array}{c|c} \boldsymbol{\Sigma}_{kAA} & \boldsymbol{\Sigma}_{kAB} \\ \hline \boldsymbol{\Sigma}_{kBA} & \boldsymbol{\Sigma}_{kBB} \end{array}\right), \quad \boldsymbol{\Lambda}_k = \left(\begin{array}{c|c} \boldsymbol{\Lambda}_{kAA} & \boldsymbol{\Lambda}_{kAB} \\ \hline \boldsymbol{\Lambda}_{kBA} & \boldsymbol{\Lambda}_{kBB} \end{array}\right)$$

Then the marginal distribution can be written as

$$p_k(\mathbf{x}_A) = \int p_k(\mathbf{x}) d\mathbf{x}_B = \mathcal{N}(\mathbf{x}_A|\boldsymbol{\mu}_{kA}, \boldsymbol{\Sigma}_{kAA})$$

and the conditional distribution for each Gaussian component k is given by

$$p_k(\mathbf{x}_A|\mathbf{x}_B) = \frac{p_k(\mathbf{x}_A, \mathbf{x}_B)}{p_k(\mathbf{x}_B)} = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{kA|B}, \boldsymbol{\Lambda}_{kAA}^{-1})$$

$$\boldsymbol{\mu}_{kA|B} = \boldsymbol{\mu}_{kA} - \boldsymbol{\Lambda}_{kAA}^{-1}\boldsymbol{\Lambda}_{kAB}(\mathbf{x}_B - \boldsymbol{\mu}_{kB})$$

and for the whole GMM as

$$p(\mathbf{x}_A|\mathbf{x}_B) = \sum_{k=1}^{K} \pi'_k p_k(\mathbf{x}_A|\mathbf{x}_B), \quad \pi'_k = \frac{\pi_k \mathcal{N}(\mathbf{x}_B|\boldsymbol{\mu}_{kB}, \boldsymbol{\Sigma}_{kBB})}{\sum_k \mathcal{N}(\mathbf{x}_B|\boldsymbol{\mu}_{kB}, \boldsymbol{\Sigma}_{kBB})}$$

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

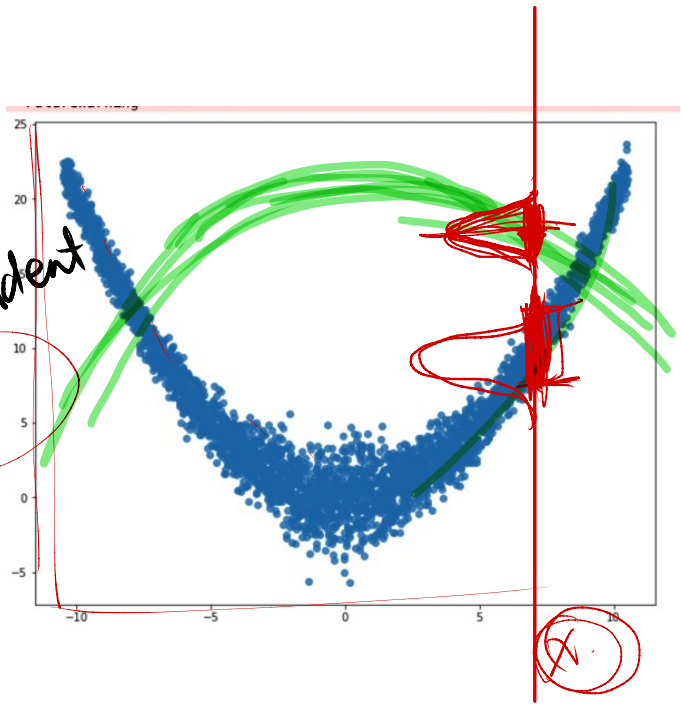
Code ← (HW).

$$P(X_{m+1:n} = x_{m+1:n} \mid X_{1:m} = x_{1:m}) \quad \text{(pdf)}$$

$$= \frac{P(X_{m+1:n} = x_{m+1:n}, X_{1:m} = x_{1:m})}{P(X_{1:m} = x_{1:m})}$$

$$= \text{Normal}$$

Review of last class.



[X, Y]
↓ ↓
condition dependent

condition X.

Model: Given condition $X = x$

the distribution of Y is modeled

with a normal distribution $N(\mu_x, \sigma_x^2)$

Neural network(f) Input: condition $\in \mathbb{R}$

output: $\mu_x = f_1(x)$ $\sigma_x^2 = f_2(x)$

$f: \mathbb{R} \to \mathbb{R}^2$

$x \to 0$ $\quad 0 \to f_1(x)$

softplus $\to f_2(x)$

Dataset: $\{(x_i, y_i)\}_{i=1}^{n}$

Training =

$$\min -\sum_{i=1}^{n} \log \mathcal{N}\left(y^i \mid f_1(x^i), f_2(x^i)\right)$$

$\mathcal{N}(f_1(x^i), f_2(x^i))$.

$\mathcal{N}(y \mid \mu, \sigma^2)$ is the value of pdf at y,

normal distribution with mean μ and variance σ^2.

Mixture density networks.

Mixture Density Networks (MDNs) combine conventional neural networks with a mixture density model for the purpose of estimating conditional distributions $p(\boldsymbol{y}|\boldsymbol{x})$ (Bishop, 1994). In particular, the parameters of the unconditional mixture distribution $p(\boldsymbol{y})$ are outputted by the neural network, which takes the conditional variable $\boldsymbol{x}$ as input.

For our purpose, we employ a Gaussian Mixture Model (GMM) with diagonal covariance matrices as density model. The conditional density estimate $\hat{p}(\boldsymbol{y}|\boldsymbol{x})$ follows as weighted sum of K Gaussians

$$\hat{p}(\boldsymbol{y}|\boldsymbol{x}) = \sum_{k=1}^{K} w_k(\boldsymbol{x};\theta)\mathcal{N}(\boldsymbol{y}|\mu_k(\boldsymbol{x};\theta), \sigma_k^2(\boldsymbol{x};\theta)) \quad (7)$$

wherein $w_k(\boldsymbol{x};\theta)$ denote the weight, $\mu_k(\boldsymbol{x};\theta)$ the mean and $\sigma_k^2(\boldsymbol{x};\theta)$ the variance of the k-th Gaussian component. All the GMM parameters are governed by the neural network with parameters θ and input $\boldsymbol{x}$. It is possible to use a GMM with full covariance matrices Σ_k by having the neural network output the lower triangular entries of the respective Cholesky decompositions $\Sigma_k^{1/2}$ (Tansey et al., 2016). However, we choose diagonal covariance matrices in order to avoid the quadratic increase in the neural network's output layer size as the dimensionality of $\mathcal{Y}$ increases.

The mixing weights $w_k(\boldsymbol{x};\theta)$ must resemble a categorical distribution, i.e. it must hold that $\sum_{k=1}^{K} w_k(\boldsymbol{x};\theta) = 1$ and $w_k(\boldsymbol{x};\theta) \geq 0\ \forall k$. To satisfy the conditions, the softmax function is used.

$$w_k(\boldsymbol{x}) = \frac{\exp(a_k^w(\boldsymbol{x}))}{\sum_{i=1}^{K}\exp(a_i^w(\boldsymbol{x}))} \quad (8)$$

In that, $a_k^w(\boldsymbol{x}) \in \mathbb{R}$ denote the logit scores emitted by the neural network. Similarly, the standard deviations $\sigma_k(\boldsymbol{x})$ must be positive. To ensure that the respective neural network satisfy the non-negativity constraint, a sofplus non-linearity is applied:

$$\sigma_k(\boldsymbol{x}) = \log\left(1 + exp(a_k^{\sigma}(\boldsymbol{x}))\right) \quad (9)$$

Since the component means $\mu_k(\boldsymbol{x};\theta)$ are not subject to such restrictions, we use a linear layer without non-linearity for the respective output neurons.