Problem Set 2
ECE 685D Fall 2020

Instructor: Vahid Tarokh
ECE Department, Duke University

Due: 8:59:59 a.m. EST on Sept. 22, 2020

Problem 1: Conditional Multivariate Gaussian Distribution

Let $\mathbf{x}$ denote a D-dimensional multivariate Gaussian random vector with mean vector $\boldsymbol{\mu} \in \mathbb{R}^D$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$ with probability density function (PDF) denoted as $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and given by

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

Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ denote a partition of $\mathbf{x}$ such that $\mathbf{x}_1 \in \mathbb{R}^p$ and $\mathbf{x}_2 \in \mathbb{R}^{D-p}$. Derive the conditional PDF of $\mathbf{x}_2$ given $\mathbf{x}_1 = \mathbf{a}$ where $\mathbf{a}$ is a constant vector in $\mathbb{R}^p$ and provide brief explanation of each derivation step. Compute the mean vector and the covariance matrix of the conditional PDF of $\mathbf{x}_2 \in \mathbb{R}^2$ given $\mathbf{x}_1 = (1, 1)$ when $\mathbf{x}$ follows a multivariate Gaussian distribution with

$$\boldsymbol{\mu} = (1, 1, 1, 2), \quad \boldsymbol{\Sigma} = \begin{pmatrix} \mathbf{I}_2 & 0.25 \times \mathbf{I}_2 \\ 0.25 \times \mathbf{I}_2 & 1.25 \times \mathbf{I}_2 \end{pmatrix},$$

where $\mathbf{I}_2$ is the identity matrix of dimension 2×2.

Problem 2: Gaussian Mixture Model

Recall from the lecture slides that the PDF of a D-dimensional Gaussian Mixture Model (GMM) with K Gaussian densities is given by

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

where $\pi_k \geq 0$ are the mixing coefficients satisfying $\sum_{k=1}^{K} \pi_k = 1$, whereas $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ denote the mean vector and the covariance matrix of the k-th Gaussian density.

Write a Python program for generating 10^3 2-dimensional random samples (i.e., $D = 2$) from a GMM with the following parameters: $K = 4$, $\{\pi_1, \pi_2, \pi_3, \pi_4\} = \{1/8, 1/8, 1/4, 1/2\}$, $\boldsymbol{\mu}_1 = (0, 0)$, $\boldsymbol{\mu}_2 = (0, 2)$, $\boldsymbol{\mu}_3 = (2, 0)$, $\boldsymbol{\mu}_4 = (2, 2)$ and covariance matrices

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.2 \end{pmatrix}, \quad \boldsymbol{\Sigma}_2 = \begin{pmatrix} 0.1 & -0.15 \\ -0.15 & 0.3 \end{pmatrix}, \quad \boldsymbol{\Sigma}_3 = \begin{pmatrix} 0.3 & 0.05 \\ 0.05 & 0.3 \end{pmatrix}, \quad \boldsymbol{\Sigma}_4 = 0.15 \times \mathbf{I}_2.$$

Write a Python program to fit a GMM to the data generated in part 1.1 with varying number of components $K = \{1, 2, 3, 4, 5, 6, 7\}$. For each estimated model compute the log-likelihood function given the generated data (see lecture slides) and plot it with respect to K. Note: to perform the fitting, you can use the class mixture.GMM from the sklearn package and the score method to compute the log-likelihood.

Problem 3: Bayesian linear regression

Suppose we want to estimate the location of an object (x_1 and x_2 coordinates) in a large field with size 2 km $\times$ 2 km. To this end, we are going to use N sensors which can estimate the coordinates. Our prior knowledge of the coordinates of the location is a 2-dimensional random Gaussian vector given by $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu_0}, \boldsymbol{\Sigma_0})$. In addition, the measurement of each sensor, y_i is a random vector distributed as $y_i \sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{\Sigma_y})$ for $i = 1, 2, \ldots, N$.

a What is the distribution of coordinates of the object after observing the sensor measurements. Please provide all the details including the pdf, and all the necessary moments information.

b What is your predicted location based on the sensor measurements? What is the MLE estimate?

c Assume that $N = 10$, $\boldsymbol{\mu_0} = [0.5, 0.5]^T$, $\boldsymbol{\Sigma_0} = 0.1\boldsymbol{I_2}$, $\boldsymbol{x} = [1.5, 1.5]^T$, and $\boldsymbol{\Sigma_y} = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.1 \end{bmatrix}$. Plot the contour of prior and posterior distributions.

d Repeat part c, with $\boldsymbol{\Sigma_0} = 10\boldsymbol{I_2}$, with $N = 10$. Now let $\boldsymbol{\Sigma_0} = 0.1\boldsymbol{I_2}$ and $N = 100$, and repeat part c.

e What is your conclusion from part d?

Problem 4: Bias-variance trade-off

Generate a training data set of 5 samples $\mathbf{x} = (x_1, \ldots, x_5)$ placed between 0 and 1 on a regular grid with step 0.2, together with the corresponding real-valued targets $\mathbf{t} = (t_1, \ldots, t_5)$ such that $t_i = \sin(2\pi x_i) + \cos(4\pi x_i)$. Also, in the same manner as above, generate a test data set of 100 regularly placed samples in $(0, 1)$ with step 0.01. Using a built in method from any python package of choice that supports polynomial fitting (such as the polyfit method from numpy), fit a polynomial model to the training data of varying degrees $M = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$. Plot the sum-of-squares error with respect to M computed on both training and test data sets.

Fix the polynomial model to degree $M = \{14\}$. Using L_2 regularization, penalizing the sum of squares of the model weights (as in the lecture slides), fit a regularized model with regularization parameter $\lambda \in [10^{-2}, 50]$. Plot the sum-of-squares error with respect to $\ln \lambda$ computed on both training and test data sets. Note: you are free to use any python package of your choice.

Problem 5: Minimizing Minkowski Loss

Let $p(\mathbf{x}, t)$ denote the joint PDF of real-valued input vector $\mathbf{x} \in \mathbb{R}^D$ and real-valued target variable $t \in \mathbb{R}$. Let

$$\mathbb{E}[L] = \int \int L(t, y(\mathbf{x})) p(\mathbf{x}, t) d\mathbf{x} dt,$$

denote the expected loss with respect to a model $y(\mathbf{x})$ and some loss function $L(t, y(\mathbf{x}))$. Show that for $L(t, y(\mathbf{x})) = |t - y(\mathbf{x})|$, $\mathbb{E}[L]$ is minimized by the median of the conditional distribution $p(t|\mathbf{x})$.

Let

$$L(t, y(\mathbf{x})) = \begin{cases} 0 & |t - y(\mathbf{x})| \leq \delta \\ 1 & |t - y(\mathbf{x})| > \delta \end{cases}, \quad \delta > 0$$

denote the hit-or-miss loss function. Show that $\mathbb{E}[L]$ is minimized by the mode of the conditional distribution $p(t|\mathbf{x})$.

Problem 6: Nonlinear Basis Functions

Let GMM-1 denote a 2-dimensional GMM with parameters $K = 1$, $\mu = (0, 0)$ and covariance matrix $\mathbf{\Sigma} = 0.1 \times \mathbf{I}_2$. Similarly, let GMM-2 denote a 2-dimensional GMM with parameters $K = 2$, $\pi_1 = \pi_2 = 0.5$, $\boldsymbol{\mu}_1 = (-1, -1), \boldsymbol{\mu}_2 = (1, 1)$ and covariance matrices $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = 0.1 \times \mathbf{I}_2$. Generate 100 samples from GMM-1 and 200 samples from GMM-2. Plot the data using different colors for the GMMs. Could you find a single line that separates the two data sets?

Let

$$\phi_1(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T(\mathbf{x}-\boldsymbol{\mu})\right) \text{ and } \phi_2(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T(\mathbf{x}-\boldsymbol{\mu}_1)\right),$$

be two 2-dimensional Gaussian basis functions. Define the transformation $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}))$ and plot the transformed data from GMM-1 and GMM-2. What do you observe?