Mathematical Underpinnings Lab 7 11.04.2024

Task 1

The task is to estimate Kullback-Leibler divergence between two distributions.

a) Implement a function `KL_multivariate_normal` computing Kullback-Leibler divergence between two multivariate normal distribution $p \sim \mathcal{N}(\mu_1, \Sigma_1)$ and $q \sim \mathcal{N}(\mu_2, \Sigma_2)$. Next, implement a function `T_opt` computing logarithm of likelihood ratio of two multivariate normal distributions at x: $\log \frac{p(x)}{q(x)}$.

b) Sample $n = 10000$ observations from $\mathcal{N}(\mu_1, \Sigma_1)$ and n observations from $\mathcal{N}(\mu_2, \Sigma_2)$, where $\mu_1 = (0.2, 0.2, 0.2, 0, 0)$, $\mu_2 = (0, 0, 0.2, 0.2, 0.2)$

$$\Sigma_i = \begin{pmatrix} 1 & \rho_i & \rho_i & \rho_i & \rho_i \\ \rho_i & 1 & \rho_i & \rho_i & \rho_i \\ \rho_i & \rho_i & 1 & \rho_i & \rho_i \\ \rho_i & \rho_i & \rho_i & 1 & \rho_i \\ \rho_i & \rho_i & \rho_i & \rho_i & 1 \end{pmatrix},$$

and $\rho_1 = 0.1$, $\rho_2 = -0.2$. Compute KL divergence using the function `KL_multivariate_normal`. Next, estimate Donsker-Varadhan functional for $\log \frac{p(x)}{q(x)}$ evaluated at sampled x.

c) Implement a loss function based on Donsker-Varadhan representation of Kullback-Leibler divergence. You may use functions: `tf.math.reduce_sum, tf.math.multiply, tf.subtract, tf.math.log, tf.math.multiply, tf.math.exp`.

d) Implement a model with previously defined loss function. Use 25% of your data as validation. As optimization might be unstable, you may consider using smaller `learning_rate`, `tf.keras.callbacks.EarlyStopping` or other methods to improve convergence.

e) Using the trained model estimate up to a constant $\log \frac{p(x)}{q(x)}$ (and compare it with the output of a function) and KL divergence on the training data. Next, check the performance on the newly generated data.

f) Do the same task using Nguyen-Wainwright-Jordan instead of Donsker-Varadhan representation. Compare the results.

Task 2

In this task we will focus on a density estimation in a multivariate case. Let our target distribution p on $[0,1]^5$ be the mixture of *Beta* distributions:

$$p(x) = \frac{1}{2} \prod_{i=1}^{5} Beta(x_i|a, b) + \frac{1}{2} \prod_{i=1}^{5} Beta(x_i|b, a),$$

where $a = 2$, $b = 5$.

a) Implement a function, which for $x = (x_1, x_2, x_3, x_4, x_5)$ returns $p(x)$. Plot a contour plot of the density p for first two dimensions. Fix $x_3 = x_4 = x_5 = 0.5$.

b) Draw a sample of $n = 10000$ observations both from p and q, where q is a uniform distribution on a cube $[0,1]^5$. Estimate the optimal value of

$$\sup_{\Phi > 0} \{\mathbb{E}_p \log \Phi(X) - \mathbb{E}_q \Phi(X) + 1\}$$

based on the sample using the formulas of the densities of the assumed distributions f and q.

c) Train a model that takes as input a joint sample from p and q and the information from which distribution each observation comes, and outputs the optimal Φ. As a loss function use the one given in a bullet point above.

d) Compare the optimal values estimated by the model at sampled points with the values obtained by using the information about the p distribution.

e) Use KDE to estimate p.

f) Compare the performance of the two estimators. First, draw a plot as in a) for both estimators. Next, sample $n = 1000$ observations from p and compute

$$\frac{1}{n}\sum_{x}(p(x) - \hat{p}(x))^2.$$