MAD 2024-25, Assignment 5

Bulat Ibragimov, Sune Darkner

hand in until: 03.01.2025 at 9:00

General comments: The assignments in MAD must be completed and written individually. You are allowed (and encouraged) to discuss the exercises in small groups. If you do so, you are required to list your group partners in the submission. The report must be written completely by yourself.

Sampling

Exercise 1 (Sampling from a Gaussian distribution, 2 points). Lets consider the case of a univariate Gaussian (Normal) distributed random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ with density $p(x|\mu, \sigma^2)$. We will use sampling to estimate expectation values for some selected functions of X, specifically we will consider the mean $E(X)$, kurtosis $\mathrm{Kurt}(X) = E\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] - 3$, and the entropy $H(X) = E[-\log(p(x|\mu, \sigma^2))]$. Here we use the Gaussian distribution, because we can easily sample directly from it, and we can prove analytically that these expectations are $E(X) = \mu$, $\mathrm{Kurt}(X) = 0$, and $H(X) = \ln(\sigma\sqrt{2\pi e})$ (where ln refers to the natural logarithm and $e = \exp(1)$), which we can use to investigate the quality of our sampling-based estimates.

a) (1 point) Implement a Python function that performs Monte Carlo integration (for any choice of μ and σ^2) of the three expectation functions mean, kurtosis, and entropy. We recommend that you use either `numpy.random.randn` or `numpy.random.Generator.normal`.

b) (1 point) Use your Python function to get approximate values for the three expectation functions stated above for the Gaussian distribution with parameter values $\mu = 2$ and $\sigma^2 = 4$. For each expectation function make a plot of the estimated value as a function of the number of samples N using an appropriate range of N.

Deliverables. a) Include your function as a code snippet in the report, b) include the three plots and comment on what you see from these.

Exercise 2 (Rejection sampling, 5 points). Lets consider a univariate random variable $X \in \mathbb{R}$ which has a distribution for which we only know the probability density function (PDF) up to a normalization constant, i.e. the PDF is proportional to

$$p(x) \propto \exp\left(-|x-2|\right) \ . \tag{1}$$

We will use sampling to estimate expectation values for some functions of X as in the previous question. But this time, we will consider the mean $E(X)$, variance $\mathrm{Var}(X) = E[(X - E(X))^2]$, and skewness $\mathrm{Skew}(X) = E\left[\left(\frac{X-E(X)}{\sqrt{\mathrm{Var}(X)}}\right)^3\right]$. But since we are not sure which distribution it is, we will be using rejection sampling using a Gaussian distribution as proposal distribution $q(x) = \mathcal{N}(\mu, \sigma^2)$.

a) (1 point) Plot the unnormalized distribution $\tilde{p}(x) = \exp\left(-|x-2|\right)$ and the proposal $k \cdot q(x) = k \cdot \mathcal{N}(\mu, \sigma^2)$ in the same graph, and argue what are appropriate values for μ, σ, and k in order to use the proposal distribution in a rejection sampling algorithm. Can you choose the parameters such that $k \cdot q(x) \geq \tilde{p}(x)$ for all x?

b) (2 points) Write a Python function that implements the rejection sampling algorithm for the distribution described by (1) using your choice of parameters μ, σ, and k. We recommend that you use either `numpy.random.randn` or `numpy.random.Generator.normal` as well as `numpy.random.Generator.random`.

c) (1 point) Implement a Python function that performs Monte Carlo integration of the three expectation functions mean, variance, and skewness by using samples generated by the function from b).

d) (1 point) Use your Python functions to get approximate values for the three expectation functions stated above for the distribution described by (1). For each expectation function make a plot of the estimated value as a function of the number of samples N using an appropriate range of N.

Deliverables. a) Include the plot of $\tilde{p}(x)$ and $k \cdot q(x)$ as well as arguments for and the values you choose for μ, σ, and k, b) include your function as a code snippet in the report, c) include your function as a code snippet in the report, d) include the three plots and comment on what you see from these.

Clustering

Exercise 3 (**Clustering, 9 points).** Please solve the tasks from *Clustering_L*14*_student_version.ipynd*, in particular:

a) (2 points) Distance metric implementation.
b) (3 points) The silhouette score implementation.
c) (4 points) k-Means implementation.

Deliverables. a-c) include your code spinets, plots and explanations into the report.