Assignment 1

Utrecht University
Statistical Learning and Stochastic Processes 2025-2026

Grigory Reznikov, Pavel Sinitcyn

Deadline: ***12 December 2025, 23:59 CET***

- You may work individually or in pairs. Each participant should submit the solution of the whole assignment (if you work in a pair, one report will be chosen at random), and clearly list all authors in the report.
- Please submit:
  - *report.pdf* – a single PDF (or Markdown) file containing all theoretical solutions and explanations/comments.
  - *code.ipynb* – a single Jupyter notebook with all code for the assignment, clearly separated into sections for each task.
- Reach out on Teams if you have any questions.
- A Q&A session will be held approximately one week before the deadline.
- Please indicate how much time you spent on the assignment when you submit your work.

1 Conte Marlo Sampling

Suppose you are given a uniform distribution $\mathbf{X} \sim \text{Uniform}(0, 1)$ and you want to numerically estimate its mean $\mathbb{E}[X] = \frac{0+1}{2} = 0.5$. A natural way to do so is to use Monte Carlo sampling: draw N samples from $\mathbf{X}$ and average them. Let's however make things more interesting and consider another strategy: still draw N samples from $\mathbf{X}$, but instead of averaging them find their median. Will this strategy work? If yes, which one is better? In this task you will analyze both strategies to answer these questions.

First of all, let's formally define both estimator distributions for a fixed N. For simplicity, let's consider only odd N so the median is always well-defined.

$$\bar{\mathbf{X}}(\mathbf{N}) = \frac{1}{N}\sum_{i=1}^{N} X_i, \quad X_i \sim \text{Uniform}(0, 1)$$
$$\hat{\mathbf{X}}(\mathbf{N}) = \text{median}(X_1, X_2, \ldots, X_N), \quad X_i \sim \text{Uniform}(0, 1)$$

(a) Find the $\mathbb{E}[\bar{X}(N)]$ analytically. Does it really estimate the mean of $\mathbf{X}$?

(b) Find the $Var(\bar{X}(N))$ analytically. Hint: find the variance of a single X_i first.

To analyze $\hat{\mathbf{X}}(\mathbf{N})$, let's find its probability density function (PDF). One way to do so is to estimate a probability of $\hat{\mathbf{X}}(\mathbf{N})$ being in a small interval $[x, x+dx]$. The event of $\hat{\mathbf{X}}(\mathbf{N})$ hitting this interval is equivalent to

- exactly $\frac{N-1}{2}$ samples being less than x,
- exactly 1 sample being in the interval $[x, x + dx]$,
- exactly $\frac{N-1}{2}$ samples being greater than $x + dx$.

For fixed events falling into these three categories the probability is

$$\begin{aligned}\mathbf{P} &= x^{\frac{N-1}{2}} \cdot dx \cdot (1 - (x + dx))^{\frac{N-1}{2}} \\ &= x^{\frac{N-1}{2}} \cdot (1 - x)^{\frac{N-1}{2}} \cdot dx + o(dx)\end{aligned}$$

Also, there are $\binom{N}{\frac{N-1}{2}}$ ways to select which samples are less than x and $N - \frac{N-1}{2} = \frac{N+1}{2}$ ways to select which rest sample is in the interval $[x, x+dx]$, so the total probability of $\hat{\mathbf{X}}(\mathbf{N})$ hitting the interval $[x, x+dx]$ is

$$\mathbf{P}\left[\hat{X}(N) \in [x, x+dx]\right] = \binom{N}{\frac{N-1}{2}} \cdot \frac{N+1}{2} \cdot x^{\frac{N-1}{2}} \cdot (1-x)^{\frac{N-1}{2}} \cdot dx + o(dx)$$

and the PDF of $\hat{\mathbf{X}}(\mathbf{N})$ is

$$f_{\hat{X}(N)}(x) = \binom{N}{\frac{N-1}{2}} \cdot \frac{N+1}{2} \cdot x^{\frac{N-1}{2}} \cdot (1-x)^{\frac{N-1}{2}}, \quad x \in [0,1]$$

(c) Find the $\mathbb{E}[\hat{X}(N)]$ analytically. Does it really estimate the mean of $\mathbf{X}$? Hint: you may either find it directly using the PDF or use the symmetry argument.

(d) Find the $Var(\hat{X}(N))$ analytically. Which distribution has the better variance $\bar{\mathbf{X}}(\mathbf{N})$ or $\hat{\mathbf{X}}(\mathbf{N})$?

Hint: you may use that

$$\int_0^1 x^a (1-x)^b dx = \frac{a!b!}{(a+b+1)!}$$

(e) Simulate both numerical approaches to check your results.

2 Dishonest Casino is Back!

Occasionally, a dishonest casino strikes back! This time, they prepared two games for you to play.

- Game `A` is a biased coin game: with probability 0.505 you lose 1 dollar, and with probability 0.495 you win 1 dollar.
- Game `B` is two biased coins game. If your current capital is a multiple of 3, you play with coin `B1`: with probability $B1_+ = 0.095$ you win 1 dollar, and with probability $B1_- = 0.905$ you lose 1 dollar. If your current capital is not a multiple of 3, you play with coin `B2`: with probability $B2_+ = 0.745$ you win 1 dollar, and with probability $B2_- = 0.255$ you lose 1 dollar.

While the first game is obviously losing in a long run, the second game requires more careful analysis. Note, that the game `B` can be modelled as a Markov chain with 3 states corresponding to the remainder of your current capital when divided by 3.

(a) Let π be the stationary distribution of the Markov chain for the game `B`. Let $E = \pi_0 \cdot (B1_+ - B1_-) + (\pi_1 + \pi_2) \cdot (B2_+ - B2_-)$. Show that if $E < 0$, then the game is losing in a long run and if $E > 0$, then the game is winning in a long run.

(b) Compute the stationary distribution π and the value of E to show that the game `B` is losing in a long run.

(c) Use Monte Carlo simulation to numerically validate that the game `B` is losing in a long run.

Surprisingly, given that both games are losing in a long run, you can still win in a long run by playing them in a proper order! For example, if you play the random game with equal probabilities every time, you will win in a long run. This phenomenon is called Parrondo's paradox.

(d) Construct the Markov chain for the random game. Compute its stationary distribution to see that it is winning in a long run.

(e) Confirm the result of (d) using Monte Carlo simulation.

Another strategy to win in a long run is to play games in some periodic order, for example, `ABBABBABB...`.

(f) Show analytically that the provided periodic strategy is winning in a long run.

(g) Confirm the result of (f) using Monte Carlo simulation.

A natural question arises: what is the optimal (in terms of expected profit) strategy to play in this casino? Turns out that the optimal strategy is always a function $\{0, 1, 2\} \to \{\texttt{A}, \texttt{B}\}$ that basically selects which game to play based on the remainder of your current capital when divided by 3. Both randomized selection and periodic strategies will not outperform the optimal strategy in such form. While we do not ask you to prove this fact, this is an interesting thing to think about!

3 Transmembrane Proteins

Proteins are the workhorses of living cells. They act as molecular machines, builders, sensors, and defenders. Motor proteins are responsible for your muscle movements, pigment-related proteins give flowers their colors, and your personal army of antibodies tirelessly protect you from viruses, bacteria, and cancer cells.

Proteins are the chain of amino acids. You can think of each amino acid as a symbol from a special alphabet (with about 20-21 possible letters). In this assignment we will therefore treat a protein as a string over an alphabet of size 21.

Many proteins live entirely inside the cell, but some are partly exposed to the outside world. They pass through the cell membrane, a thin layer that separates the cell interior from its surroundings. The parts of a protein that lie inside the membrane are called transmembrane domains. In this assignment you will use a hidden Markov model (HMM) to predict which positions in a protein sequence are transmembrane domains.

You are provided with a dataset of human proteins with known transmembrane domains, you can download it here. Each row of the dataset contains a protein sequence (a string of amino acids) and a string of the same length with transmembrane domain annotations (`1` indicates that the corresponding amino-acid is part of a transmembrane domain, `0` indicates that it is not).

The simplest HMM for this task has two hidden states: `I` for being inside the membrane and `O` for being outside the membrane. The observations are the 21 amino acids. The intuition behind this HMM is that amino acids have different frequencies inside and outside the membrane.

Note that training of such HMM does not require Baum-Welch algorithm, since you know the hidden states for each training sequence. You can therefore rely on the Monte Carlo-styled parameter estimation that we used for regular Markov models during tutorials.

(a) Train the HMM on the provided dataset. To do so, shuffle the data and split it into 80% training and 20% test sets. Estimate the HMM parameters using the training set.

(b) Implement the Viterbi algorithm to predict transmembrane domains in protein sequences. Evaluate your predictions on the test

set using precision and recall metrics. Since the model is very simple, do not expect high scores. 70+% precision and 50+% recall is already a good result!

One of the issues with the above HMM is that it cannot model the length of transmembrane domains. To fix this, we can use a more complex HMM with multiple `I` hidden states: `I1`, `I2`, ..., `Ik`, where `I1` represents the first amino acid inside the membrane, `I2` represents the second, and so on to `Ik` which represents the k-th and all subsequent amino acids inside the membrane. When transmembrane domain ends, the model transitions back to `O` state.

(c) Implement this extended HMM with $k = 20$ internal states. Train it on the same training set and evaluate on the same test set. You should observe an improvement in precision.

(d) Try various values of k. How does it affect the performance?

You may see that even with the extended HMM, performance is still not very high. State of the art HMMs for transmembrane domain prediction use more complex architectures with additional hidden states (for example, whether non-membrane parts are inside or outside of the cell) to achieve better quality.

4 Gamma-ray Burst

Gamma rays are the most energetic form of electromagnetic radiation. They are produced in the most extreme environments in the universe, such as supernova explosions, surfaces of neutron stars and regions around black holes. Modern space telescopes constantly register individual γ-ray photons hitting their detectors over time. In typical conditions, these photon arrivals can be modelled as a Poisson process, that is, the number of photons counted in non-overlapping time intervals are independent Poisson random variables.

However, sometimes telescopes register sudden bursts of γ-ray photons that last from milliseconds to several hours. During these bursts, the rate of photon detections increases and then decreases back to the background level. Statistically, we can describe this phenomenon as a Poisson process with piecewise-constant rate: typically low background rate λ_{bg}, then a higher burst rate $\lambda_{burst} > \lambda_{bg}$, and finally back to the background rate λ_{bg}.

In this problem you will work with a **synthetic** dataset that mimics a telescope registering γ-ray photons over time. You can download the dataset here. This dataset contains two columns: `t`, the start time of each bin in seconds and `counts` which is the number of photons detected in that bin. Each bin has a width of 0.5 seconds. The dataset was generated according to the statistical model described above: $\lambda_{burst} > \lambda_{bg}$ were selected, than a burst interval $[t_{start}, t_{end}]$ were chosen, and finally counts in each bin were drawn from the corresponding Poisson distribution. Note, that the dataset is "toy", real telescope data will have much larger λ.

(a) Use any algorithm of your choice to estimate the t_{start} and t_{end}. Plot the data and visually verify that your estimates are correct.

Now let's try to estimate the parameters of the underlying Poisson process: λ_{bg} and λ_{burst}. To do so, we will use maximum likelihood estimation (MLE). The likelihood of observing counts with selected model parameters is

$$
\begin{aligned}
L(\lambda_{bg}, \lambda_{burst}) &= \prod_i P(c_i | \lambda_{bg}, \lambda_{burst}) \\
&= \prod_{i: t_i < t_{start} \text{ or } t_i > t_{end}} \frac{e^{-\lambda_{bg}} \lambda_{bg}^{c_i}}{c_i!} \cdot \prod_{i: t_{start} \le t_i \le t_{end}} \frac{e^{-\lambda_{burst}} \lambda_{burst}^{c_i}}{c_i!}
\end{aligned}
$$

where c_i is the number of counts in bin i starting at time t_i.

(b) Find the MLE estimates for λ_{bg} and λ_{burst} analytically or numerically.

Now let's try to statistically verify that a burst actually happened. To do so, we will use the null hypothesis that no burst happened, that is, $\lambda_{bg} = \lambda_{burst} = \lambda$.

(c) Under the null hypothesis, find the MLE estimate for λ.

For a given dataset, we will use the likelihood ratio test to compare the null hypothesis (no burst) versus the alternative hypothesis (burst happened). The test statistic is

$$T = -2 \ln \frac{L(\hat{\lambda})}{L(\hat{\lambda}_{bg}, \hat{\lambda}_{burst})}$$

where $\hat{\lambda}$ is the MLE estimate under the null hypothesis and $(\hat{\lambda}_{bg}, \hat{\lambda}_{burst})$ are the MLE estimates under the alternative hypothesis.

(d) Compute the value of the test statistic T for the provided dataset.

To perform the likelihood ratio test, we will generate synthetic datasets under the null hypothesis and compute the test statistic for each of them.

(e) Generate 1000 synthetic datasets under the null hypothesis using the MLE estimate $\hat{\lambda}$ from (c). For each dataset compute the test statistic T. Plot the histogram of the resulting test statistics. Which conclusion can you make about the presence of a burst in the original dataset?

5 Ising Model

In this task you will use Metropolis-Hastings algorithm to sample configurations of the 2D Ising model that has significant importance in physics.

The Ising model is a simple mathematical model of ferromagnetism: each point of a grid represents an atom whose «spin» can be either up $(+1)$ or down (-1). You can think of each spin as representing the orientation of a tiny magnetic dipole, like a miniature bar magnet with a north and south pole. When many neighboring atoms have the same spin, their magnetic fields reinforce each other, and material develops a net magnetization, that is, becomes a large magnet. However, if neighboring spins are misaligned, they tend to cancel each other out so the material does not exhibit magnetism.

The internal energy of an Ising system is defined in terms of how neighboring spins interact. For a configuration of spins $\{s_i\}, s_i \in \{-1, +1\}$, the energy is given by

$$E = -J \sum_{\langle i,j \rangle} s_i s_j$$

where J is the interaction strength and summation is over all pairs of **neighboring** spins. For the case of a 2D square lattice, each spin has at most four neighbors. In this problem, we always use $J = 1$, so energy is just

$$E = -\sum_{\langle i,j \rangle} s_i s_j$$

Note that energy is minimized when all spins are equal. The system «prefers» states with lower energy, so the state with equal spins is both energetically preferable and demonstrates maximum magnetization.

However, physical systems are not always found in their lowest energy state. At non-zero temperature T, the atoms constantly jiggle due to thermal energy, so the system explores multiple configurations over time. For a given temperature T, the probability of observing a configuration with some energy E is proportional to the exponent of negative energy:

$$P(E) \propto e^{-E/kT}$$

where k is the Boltzmann constant. For simplicity, we set $k = 1$.

(a) Implement the Metropolis-Hastings algorithm to sample configurations of the 2D Ising model at a given temperature T. Draw the resulting configurations for the 16×16 lattices at temperatures $T = 1.0, 1.5, 2.0, 3.0, 5.0, 10.0$. You should observe that at lower temperatures spins tend to be clustered, while at higher temperatures spins are close to random.

(b) Validate your implementation. Consider all possible $2^{2\times 2} = 16$ configurations of a 2×2 Ising model and find their probabilities at temperature $T = 1.0$. Use Monte Carlo sampling to estimate the probabilities of your MCMC sampler and compare them to the exact values.

The Curie point is a temperature at which a ferromagnetic material loses its magnetic properties. This can be easily verified experimentally but in this task we will try to observe it computationally. To do so, consider a magnetization of a system

$$M = \frac{1}{N} \left| \sum_i s_i \right|$$

where N is the size of the lattice.

Higher magnetization means that more spins are aligned (and the material is magnetic), while lower magnetization means that spins are disordered and magnetic properties are lost.

(c) Using your MCMC implementation, estimate the average magnetization of a 16×16 Ising model at temperatures ranging from $T = 1.0$ to $T = 5.0$ with a step of 0.1. Plot the resulting magnetization as a function of temperature and identify the Curie point.

(d) Try various lattice sizes. Does the Curie point change with the size?