Probabilistic Decision Making VU, WS25
Assignment 1
Probability Spaces, Sigma-Algebras, Distribution Functions

Thomas Wedenig
thomas.wedenig@tugraz.at

Teaching Assistant:	Leon Tiefenböck
Points to achieve:	20 Points
Deadline:	14.11.2025 23:59 (strict, no late submissions allowed)
Hand-in procedure:	You can work in groups of **at most two people**. **Exactly one** team member uploads three files to TeachCenter: **The report (.pdf)**, `sigma_algebra.py`, and `monte_carlo.py`. The first page of the report must be the **cover letter**. Do not rename the two Python files. Do not upload a folder. Do not zip the files. We do not accept submissions via means other than TeachCenter.
Plagiarism:	If detected, we grade *all involved parties* with "Ungültig aufgrund von Täuschung"

General Remarks

Your submission will be graded based on:

- Correctness (Is your code doing what it should be doing? Is your derivation correct?)
- The depth of your interpretations (Usually, only a couple of lines are needed.)
- The quality of your plots (Is everything clearly readable/interpretable? Are axes labeled? …)

Remarks:

- All results (i.e., plots, results of computations) should be included in your PDF report.
- Report all intermediate steps in pen & paper exercises—only presenting the final solution is insufficient.
- Your submission must run with Python 3.11.13 and the package versions listed in `requirements.txt`.
  - Check TeachCenter for instructions to setup a conda environment.
- Do not use any external packages except for the ones listed in `requirements.txt`.
- **Do not modify the function signatures** of the provided functions.
  - i.e., do not edit the function names and inputs
- Do not use Large Language Models (LLMs) to generate any part of your solution.
  - Evident LLM usage will be treated as plagiarism.

Failure to adhere to these rules may result in point deductions.

Task 1 – Probability Spaces [5 Points]

Task 1.1 [1.5 points] For each of the following candidates $(\Omega, \mathcal{F})$, decide whether $\mathcal{F}$ is a σ-algebra on the set Ω. Justify your answer in each case. $\mathcal{P}(A)$ denotes the powerset of A, i.e., the set of all subsets of A.

(a) $\Omega = \{1,2,3,4,5,6\}$, $\mathcal{F} = \{\emptyset, \Omega, \{1,2\}, \{3,4\}, \{5,6\}, \{1,2,3,4\}, \{1,2,5,6\}, \{3,4,5,6\}\}$.

(b) $\Omega = \mathbb{R}$, $\mathcal{F} = \mathcal{P}(\Omega)$.

(c) $\Omega = \mathbb{N}$, $\mathcal{F} = \{A \subseteq \Omega : A \text{ is finite or } \Omega \setminus A \text{ is finite}\}$.

(d) $\Omega = \mathbb{N}$, $\mathcal{F} = \{A \subseteq \Omega : A \text{ is countable or } \Omega \setminus A \text{ is countable}\}$.

Task 1.2 [1.5 points] For each of the following candidates $(\Omega, \mathcal{F})$, decide whether $\mathcal{F}$ is a σ-algebra on the set Ω. Justify your answer in each case. If you think $\mathcal{F}$ is not a σ-algebra on Ω, provide the smallest σ-algebra on Ω that contains all sets $\in \mathcal{F}$.

(a) $\Omega = \{1,2,3,4,5,6\}$, $\mathcal{F} = \{\emptyset, \Omega, \{1,2\}, \{3,4,5,6\}, \{1,3\}, \{2,4,5,6\}\}$.

(b) $\Omega = \{1,2,3,4,5,6\}$, $\mathcal{F} = \{\emptyset, \Omega, \{2,4,6\}, \{1,3,5\}, \{1,2,3\}, \{4,5,6\}\}$.

(c) $\Omega = \{1,2,3,4,5\}$, $\mathcal{F} = \{\emptyset, \Omega, \{1,2\}, \{3,4,5\}, \{1,2,3,4\}, \{5\}, \{1,2,5\}\}$.

Task 1.3 [1 points] For each of the following candidates $(\Omega, \mathcal{F}, \mathbb{P})$, decide whether it is a *valid probability triple*. Justify your answer in each case.

(a) $\Omega = \{1,2,3,4,5\}$, $\mathcal{F} = \mathcal{P}(\Omega)$, $\mathbb{P}(A) = \sum_{n \in A} p_n$, $(p_1, p_2, p_3, p_4, p_5) = (0.2, 0.3, 0.4, 0.1, 0)$.

(b) $\Omega = \mathbb{N} \setminus \{0\}$, $\mathcal{F} = \mathcal{P}(\Omega)$, $\mathbb{P}(A) = \sum_{n \in A} \frac{1}{2^n}$.

(c) $\Omega = \mathbb{N} \setminus \{0\}$, $\mathcal{F} = \mathcal{P}(\Omega)$, $\mathbb{P}(A) = \sum_{n \in A} \frac{1}{n^2}$.

(d) $\Omega = [0,1]$, $\mathcal{F} = \mathcal{P}(\Omega)$, $\mathbb{P}(A) = 0.6 \cdot \delta_0(A) + 0.4 \cdot \delta_1(A)$ with $\delta_x(A) := \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases}$

Task 1.4 [1 points] Let $(\Omega, \mathcal{F}, \mathbb{P})$ and $(\Omega, \mathcal{F}, \mathbb{Q})$ be two probability triples, i.e., Ω is a non-empty sample space, $\mathcal{F}$ is a σ-algebra on Ω, and $\mathbb{P}, \mathbb{Q} : \mathcal{F} \to [0,1]$ are probability measures.

(a) Assume that

$$\mathbb{P}(A) = \mathbb{Q}(A) \qquad \forall A \in \mathcal{F} \text{ where } \mathbb{P}(A) \leq 1/2.$$

Prove the following statement:

$$\mathbb{P}(A) = \mathbb{Q}(A) \qquad \forall A \in \mathcal{F}.$$

(b) Assume now instead that

$$\mathbb{P}(A) = \mathbb{Q}(A) \qquad \forall A \in \mathcal{F} \text{ where } \mathbb{P}(A) < 1/2.$$

Do we still have $\mathbb{P}(A) = \mathbb{Q}(A)$ $\forall A \in \mathcal{F}$? If you think so, prove this. If you believe the opposite, provide a counterexample, i.e., a concrete example of Ω, $\mathcal{F}$, $\mathbb{P}$ and $\mathbb{Q}$ that do not satisfy this.

Task 2 – Sigma-Algebras [10 Points]

Task 2.1 [3.5 points] In `sigma_algebra.py`, implement the function `is_sigma_algebra`, which takes a finite set Ω (sample space) and a set of sets E as input and which returns `True` if and only if E is a sigma-algebra over Ω. Do not change the function signature of `is_sigma_algebra(omega: Set, E: List[Set])`[1]. Test your implementation by writing at least 5 tests in `run_tests()`. Using big $\mathcal{O}$ notation, briefly discuss the computational complexity of your implementation of `is_sigma_algebra` w.r.t. the input size $|E|$ (in your analysis, you can assume that $|\Omega|$ is constant).

Hints:

- Also make sure you check $E \subseteq \mathcal{P}(\Omega)$, where $\mathcal{P}(\cdot)$ denotes the powerset of the input
- The python type `set` allows the usual mathematical set operations, for example:
  - `S1.union(S2)` returns the union of sets `S1` and `S2`
  - `S1 - S2` returns the set difference of sets `S1` and `S2`
  - `S1.issubset(S2)` returns `True` if `S1` is a subset of `S2`

Task 2.2 [4.5 points] In `sigma_algebra.py`, implement the function `complete_sigma_algebra`, which takes a finite set Ω (sample space) and a set of sets E as input and returns the smallest sigma-algebra on Ω that contains all sets in E. If a set in E is not a subset of Ω, it must not be included in the resulting sigma-algebra. Test your implementation by writing at least 5 tests in `run_tests()`.

Task 2.3 [1 point] Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two sigma-algebras over the set Ω. Define

$$\mathcal{F} := \mathcal{F}_1 \cap \mathcal{F}_2$$

Is $\mathcal{F}$ a sigma-algebra over Ω? Explain your reasoning.

Task 2.4 [1 point] Let $\mathcal{F}_1, \mathcal{F}_2$ be two sigma-algebras over Ω_1, Ω_2 respectively. Define

$$\mathcal{F} := \mathcal{F}_1 \cup \mathcal{F}_2$$

Is $\mathcal{F}$ a sigma-algebra over $\Omega_1 \cup \Omega_2$? Explain your reasoning.

[1] While E is mathematically a set of sets, we represent it here with a `list` of `sets`. The reason is that `set` is not *hashable* in Python, so we cannot construct sets of sets in Python (e.g. `set([set([5])])` throws an error).

Task 3 – Distribution Functions [5 Points]

Task 3.1 [3 points] Let

$$\mathcal{N}(x; \mu, \sigma^2) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

denote the univariate *Gaussian* probability density function (pdf) with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$.

Show that the product of two Gaussian pdfs is again an (unnormalized) Gaussian, i.e.,

$$\mathcal{N}(x; a, \sigma_a^2)\,\mathcal{N}(x; b, \sigma_b^2) = Z \cdot \mathcal{N}(x; \mu_{ab}, \sigma_{ab}^2) \quad \text{with} \quad \mu_{ab} = \frac{\sigma_b^2 a + \sigma_a^2 b}{\sigma_a^2 + \sigma_b^2} \quad \text{and} \quad \sigma_{ab}^2 = \frac{\sigma_a^2 \sigma_b^2}{\sigma_a^2 + \sigma_b^2},$$

where $Z = \int \mathcal{N}(x; a, \sigma_a^2)\,\mathcal{N}(x; b, \sigma_b^2)\,\mathrm{d}x$ is the *partition function*, which can also be written as evaluating a different Gaussian at the point a: $Z = \mathcal{N}(a; b, \sigma_a^2 + \sigma_b^2)$. Clearly show all steps in your derivation. Your final result should exactly recover the equation $\mathcal{N}(x; a, \sigma_a^2)\,\mathcal{N}(x; b, \sigma_b^2) = \mathcal{N}(a; b, \sigma_a^2 + \sigma_b^2)\,\mathcal{N}(x; \mu_{ab}, \sigma_{ab}^2)$ with μ_{ab}, σ_{ab}^2 as above.

Hints:

- Write down the product using the definition above and make use of the rules of the exp function to transform the product into a single exponential form
- After some algebraic manipulation, transform the exponent such that the term x^2 has no leading coefficient in the numerator, which should allow you to read off μ_{ab} and σ_{ab}^2
- Complete the square in the exponent. The quantity that is added to the resulting quadratic will be related to $\mathcal{N}(a; b, \sigma_a^2 + \sigma_b^2)$
- **Remark:** Here, we are multiplying two Gaussian *density functions* (pointwise). This is *not the same* as multiplying the corresponding Gaussian random variables.

Task 3.2 [2 points] Let $p : \mathbb{R} \to \mathbb{R}$ denote the probability density function of a continuous random variable X, given as

$$p(x) = \begin{cases} \frac{1}{2}x & 0 < x < 2 \\ 0 & \text{otherwise} \end{cases}$$

1. The corresponding *cumulative distribution function* (CDF) is defined as

   $$F(x) = \int_{-\infty}^{x} p(z)\; dz.$$

   Write down $F(x)$ as a simple function of x that does not involve an integral (i.e., solve the definite integral).

2. Compute $\mathbb{P}_X([-0.5, 0.5] \cup [1.5, 2])$ using only the CDF F and the properties of a probability measure.

3. Compute $\mathbb{P}_X(\{1\})$. Is it the same as $p(1)$?

4. Prove or refute: $\mathbb{P}_X([0, 1]) = \mathbb{P}_X([0, 1))$.

5. Analytically compute $\mathbb{E}_X[X]$.

6. We can sample from $p(x)$ using the *inverse transform sampling* trick: First, sample $u \sim \text{Unif}([0, 1])$ and then, compute $x = F^{-1}(u)$ where F^{-1} denotes the inverse of F. The result x is a proper sample from p. Write down $F^{-1}(u)$ and implement this sampling procedure in `monte_carlo.py` (function `F_inv`).

7. Use this sampling procedure to estimate $\mathbb{E}_X[X]$ via *Monte Carlo*: For all $N \in \{100, 200, 300, \ldots, 10000\}$, compute the sample mean $\hat{\mathbb{E}}_N[X] := \frac{1}{N}\sum_{i=1}^{N} x_i$ where x_i are i.i.d. samples from p. Plot the sample mean as a function of N (i.e., N is shown on the x-axis, and the corresponding sample mean on the y-axis). Draw a horizontal line at the true expectation. Include this plot in your report.